

# S3 Portfolio GP

Tim Meijvogel S3 DB01



# Inhoud

<b>Opdracht</b>	<b>4</b>
<b>Web application</b>	<b>4</b>
Frontend	4
Backend	4
Mijn Taak	5
<b>Agile method</b>	<b>6</b>
Scrum	6
Andere methodes	6
Kanban	6
Waterval	7
<b>Cultural differences and ethics</b>	<b>8</b>
Cultuur	8
Wat is cultuur?	8
Wat zijn bekende dimensies van culturele verschillen?	8
Kun je voorbeelden geven van culturele verschillen die je in je studie of leven hebt ervaren? Hoe verklaar je deze verschillen?	9
Wat is jouw cultuur?	9
What did you do to improve the group communications with respect to cultural differences? (indien toepasselijk)	9
Ethiek	10
Wat is ethiek in software-engineering?	10
Waarom is ethiek belangrijk in software engineering?	10
Wat moet je als software engineer doen om ethische aspecten in je werk aan te pakken?	10
Hoe weet je dat jouw ethische overwegingen overeenkomen met die van andere software-engineers?	10
Welke ethische aspecten spelen een rol in mijn project?	11
Voorzie jij ethische conflicten die door uw software worden veroorzaakt? Wat voor soort?	11
Kun jij iets doen om deze ethische conflicten te minimaliseren?	11
<b>Requirements and Design</b>	<b>12</b>
User stories	12
Technische keuzes en architectuur	13
Structuur	13
Backend	13
Frontend	13
Database	14
Model	14
UI	14
Testen	15
Design	15

<b>Business processes</b>	<b>17</b>
Waarom heb ik dit zo ingedeeld:	17
Niet digitaal	18
Digitaal	18
<b>Professional</b>	<b>19</b>
Wat werd er van ons verwacht	19
Team communicatie	19
Communicatie met de stakeholders	20
Progressie van je team bijhouden	21
Vragen voor feedback	22
Research	23

# Opdracht

Dit semester hadden weer een Groepsproject naast ons individuele project. Onze opdracht was het maken van een restaurant bestelsysteem, kunt hier meer over lezen via deze link: <https://github.com/fontys-group3>

Ik heb aan dit project gewerkt met Kaan Gögcay, Mike van den Hoef, Cas Esselink en David Chang.

## Web application

Om dit leerdoel aan te tonen moesten we hetzelfde doen als bij het individuele project. We moesten een full stack web applicatie maken met een javascript framework frontend en de backend moest een OO taal zijn.

### Frontend

We hebben 3 verschillende frontends gemaakt, de eerste was het menu met een winkelwagen, de tweede was een frontend voor de personeelsleden en de derde was een management portal. Iedereen heeft iets aan de frontend gedaan. Kaan en David hebben gewerkt aan het menu, Mike heeft gewerkt aan de personeels frontend en cas heeft gewerkt aan een minimale management portal. Ik heb gewerkt aan de winkelwagen in het menu.

### Backend





Wij hebben gekozen om te werken met een microservice architectuur en de taal die is gebruikt om de backend te maken was java. Kaan, Cas, Mike en David hebben gewerkt aan de backend.

## Mijn Taak

Mijn taak was het maken van de winkelwagen in de frontend. Dit hield in dat ik verantwoordelijk was voor het opslaan van een bestelling in cookies en deze door te sturen naar de backend. Aangezien ik vrijwel geen ervaring had met react of een javascript framework en niet goed was in het bouwen van een frontend betekende dit dat ik ben begonnen met het zoeken naar informatie en tutorials. Hierna ben ik begonnen door te oefenen met react. Uiteindelijk heb ik met de hulp van Kaan en Mike een winkelwagen gemaakt met de functionaliteiten die werden verwacht. Dit deel van het project moest constant bijgewerkt worden en was dus nooit af. Ook was het mijn taak om E2E testen te schrijven voor de frontend. Dit heb ik gedaan voor de frontend die Mike heeft gemaakt en ik was begonnen met het schrijven van tests voor de menu pagina.

Bestelling voor tafel 5 ✕

4 items

	Product	Price	Qty	Total	Actions
	Rosbief Salade	€2,49	1	€2,49	✕
	Bladerdeeg Hapje	€5,49	1	€5,49	✕
	Cheesecake	€4,99	3	€14,97	✕
	Sinas	€3,59	2	€7,18	✕
TOTAAL				€30,13	

Bestel

```
describe("redirect works correctly", () => {
  it("redirects to kitchen correctly", () => {
    cy.visit("/");
    cy.get('button').contains('Kitchen').click();
    cy.url().should('include', '/kitchen')
  })

  it("redirects to bar correctly", () => {
    cy.visit("/");
    cy.get('button').contains('Bar').click();
    cy.url().should('include', '/bar')
  })
})
```

# Agile method

**Agile** method: You are aware of most popular agile methods and their underlying agile principles. You are able to implement the process of your software project according to a chosen methodology.

## Scrum

Scrum werkt als volgt:

Er zijn verschillende zelforganiserende teams die bestaan uit verschillende rollen. De persoon die dit proces begeleidt is de scrum master. Deze persoon zorgt ervoor dat het team optimaal kan presteren. De developers en de scrum master werken samen met product owner, die als tussenpersoon dient tussen de klant en developers. De teams van developers werken in sprints, dit is een tijdsperiode van maximaal 1 maand. De sprint begint met een sprint planning (wat gaan we doen in de aankomende sprint?), en eindigt met een sprint review (dit is het moment dat de developers een (deel)product laten zien dat zij hebben gemaakt en feedback ontvangen van de stakeholders) en een retrospective (hierbij wordt gekeken we naar hoe de samenwerking verliep en hoe dit verbeterd kan worden).

Voor ons groepsproject hebben we gebruik gemaakt van scrum. In onze groep hadden we een scrum master, de scrum master leidde onze dagelijkse stand ups en stand downs. We hadden 5 sprints, deze sprints duurde 3 weken. Na elke sprint hadden we een sprint oplevering, hier kregen we feedback van de product owners en plande we in wat er de volgende sprint moest gebeuren. Na deze opleveringen hadden we een retrospectief, hierbij keken we naar wat hoe onze samenwerking verliep en hoe we dit konden verbeteren.

## Andere methodes

Ik beschrijf hier kort 2 andere agile methodes.

### Kanban

Kanban is te verdelen in 6 stappen:

1. Visualiseren

Bij deze stap worden het werkproces en werkafspraken gevisualiseerd. Op deze manier kun je bijvoorbeeld per processtap de bijbehorende afspraken toevoegen.

2. Limiteer de hoeveelheid werk

De Kanban Methode maakt gebruik van een Pull model. Dit betekent dat je pas nieuw werk aan het proces toevoegt als er capaciteit vrijkomt. Je stelt een limiet op de Work in Progress.

### 3. management van de workflow

Bij deze stap wordt de workflow geoptimaliseerd. Er is een goede workflow als items op je kanban bord nauwelijks stil komen te staan. Je meet een goede flow af aan de doorlooptijd of levertijd. Hoe korter, hoe beter de flow is. Mocht er een item zijn dat te lang stil ligt, dan ga je kijken of dit item goed is opgezet.

### 4. Maak afspraken expliciet

Bij deze stap maak je duidelijk wanneer een item af is. Dit wordt gedaan, zodat je niet te lang door blijft werken aan een item dat al voldoet aan wat er door de product owners wordt verwacht.

### 5. Implementeren van feedback

Bij kanban wordt er gebruik gemaakt van feedback loops. Deze feedback loops kunnen maandelijks, wekelijks en zelfs dagelijks zijn. Bij deze meetings wordt op een vast moment een vast onderwerp besproken.

### 6. Verbetering

Bij kanban is het einddoel niet volledig duidelijk, de kanban methode gaat uit van de huidige situatie. Het team en het product verbeteren geleidelijk.

## **Waterval**

Waterval project management is een sequentieel, lineair proces voor projectmanagement. Het proces bestaat uit verschillende fasen. De volgende fase begint pas als de huidige fase helemaal af is en elke voltooiing is definitief, met de watervalmethode kun je niet terugkeren naar een vorige fase. De enige manier om een fase opnieuw te doorlopen, is door opnieuw te beginnen bij fase één.

De fasen die je doorloopt bij een waterval proces zijn:

#### 1. Requirements

Het opstellen van requirements is de eerste stap in het proces. Je moet zorgen dat je alle requirements aan het begin gedefinieerd hebt, aangezien je niet meer terug kan zodra de fase voorbij is.

#### 2. Design

Hierbij is te denken aan class diagrams, architectuur modellen en flowcharts.

#### 3. Implementatie

Dit is wanneer het team begint met het bouwen van de applicatie, dit wordt gedaan aan de hand van de vorige 2 stappen.

#### 4. Verificatie

Zodra de programmeurs klaar zijn met coderen, kan het product worden getest. Testers zoeken naar problemen en rapporteren hierover. Als er zich ernstige problemen voordoen, moet je mogelijk terug naar fase 1 om je project te evalueren.

#### 5. Levering/Implementatie

In deze fase is het product gereed en levert het team het eindproduct aan voor implementatie of publicatie.

## 6. Onderhoud

Het product is aan de klant geleverd en is in gebruik. Als er zich problemen voordoen, moet het team mogelijk patches en updates ontwikkelen om de problemen te verhelpen. Ook nu geldt dat bij grote problemen het team mogelijk terug moet naar fase 1.

# Cultural differences and ethics

## Cultuur

### **Wat is cultuur?**

Cultuur is voor elk land anders ook al zullen in sommige gevallen de verschillen niet groot zijn. Het gaat hier om normen en waarden, de voeding, eetgewoonten, kleding, godsdienst en muziek en dans.

### **Wat zijn bekende dimensies van culturele verschillen?**

Er zijn een aantal bekende dimensies, 4 hiervan zijn:

Power distance:

Deze dimensie gaat over de gelijkheid in een samenleving. Het geeft de houding van de cultuur weer ten opzichte van ongelijkheden in een samenleving. Power distance wordt weergegeven als de mate waarin de minder machtige leden binnen een land verwachten en accepteren dat de macht ongelijk verdeeld is.

Uncertainty avoidance:

Deze dimensie laat zien hoeveel een samenleving risico's tolereert. Wil je controle hebben over de toekomst of laat je het gewoon gebeuren?

Individualism-collectivism:

Dit is de mate van onderlinge afhankelijkheid in een samenleving. Het heeft te maken met de vraag of het zelfbeeld van mensen wordt gedefinieerd door de termen 'ik' of 'wij'. Geef je prioriteit aan je eigen prestaties of aan de van de groep. Het tegenovergestelde van individualisme is collectivisme.

Masculinity-femininity:

Wanneer aan samenleving mannelijk (masculin) is wordt de samenleving gedreven door competitie, prestatie en succes. Hierbij wordt succes bepaald door de beste in een bepaald veld. Dit is een waardesysteem dat op school begint en gedurende het hele bedrijfsleven te zien is. Het tegenovergestelde hiervan is een feministische samenleving. Dit houdt in dat de dominante waarden in de samenleving zorgen voor anderen en kwaliteit van leven zijn. Deze dimensie is een samenleving waar kwaliteit van leven het teken van succes is.



**Kun je voorbeelden geven van culturele verschillen die je in je studie of leven hebt ervaren? Hoe verklaar je deze verschillen?**

Tijdens mijn studie heb ik hier vrijwel niets van gemerkt, Het enig voorbeeld dat kan noemen is dat ik een aantal mensen ben tegengekomen die geen nederlands spraken. Op mijn werk is het enige verschil dat ik zie dat sommige meiden een hoofddoek dragen. Normaal mag niemand tijdens werk iets op hun hoofd dragen, maar omdat dit een keuze is gemaakt die draait om religie zijn deze mensen uitzonderingsgevallen.

**Wat is jouw cultuur?**

Cultuur gaat voor mij vooral over waarden en normen, over wat goed en fout is. Ik ben van mening dat ik een nederlandse cultuur heb, dit houd bijvoorbeeld niet in dat ik al het eten dat kenmerkend is voor Nederland lekker vind. Dit houd in dat de normen en waarden die in Nederland normaal zijn, voor een groot deel voor mij ook gelden. Een voorbeeld is de manier waarop ik iemand begroet, dit doe ik door de persoon aan te kijken en hallo te zeggen, als het een speciale gelegenheid schud ik ook nog de hand. Als ik dit vergelijk met een land als Japan, waar ze meerdere manieren hebben om iemand te begroeten op basis van wie het is. Verder ben ik zelf niet gelovig, maar heb ik wel respect voor mensen die dit wel zijn, ik probeer ze niet te overtuigen dat wat zij geloven fout is. Wel vind ik het vervelend als dit andersom gebeurt.

Wat voor mij dus belangrijk is, is dat ik mensen behandel zoals ik behandeld zou willen worden.

**What did you do to improve the group communications with respect to cultural differences? (indien toepasselijk)**

Ik kan geen voorbeelden noemen uit onze groep. Mijn ervaring was dat iedereen elkaar respectvol behandelde en we met elkaar omgingen zoals wij zelf ook behandeld zouden willen worden.

# Ethiek

## **Wat is ethiek in software-engineering?**

De ethiek van software-engineering kan vanuit twee richtingen worden bekeken. Ten eerste kan het de activiteit beschrijven dat je je bezighoudt met de gevolgen/impact die je programma kan hebben op andere mensen en Slechte gevolgen vermijd.

Ten tweede kan het worden gebruikt om een verzameling principes te beschrijven, richtlijnen, of ethische aspecten die ons leiden naar verantwoorde keuzes..

## **Waarom is ethiek belangrijk in software engineering?**

De impact van software die wordt geschreven wordt steeds groter, alles gebeurt tegenwoordig automatisch. Mensen kunnen hun baan verliezen door software die wordt geschreven door software engineers. Als software engineers is het niet ons doel om mensen hun baan te laten verliezen maar hun banen makkelijker te maken. Een voorbeeld hiervan is een restaurant dat overstapt naar een applicatie. Als deze applicatie de klant volledig zelfstandig laat bestellen, heeft een ober alleen nog de taak om eten te brengen en borden op te halen. Hierdoor zijn er minder obers nodig en kunnen mensen hun baan verliezen. Wat dus een betere oplossing was, is om de ober een tablet te geven waarmee die gemakkelijk bestellingen op kan nemen, die meteen naar de keuken worden gestuurd.

## **Wat moet je als software engineer doen om ethische aspecten in je werk aan te pakken?**

Als software engineer is het belangrijk dat je je opdrachtgevers duidelijk uitlegt wat de impact van het programma dat jij gaat maken op zijn bedrijf heeft. Dit betekent zowel de goede als slechte gevolgen. Software moet alleen goed gekeurd worden als deze veilig is, voldoet aan de specificaties, de juiste tests doorstaat en de kwaliteit van leven, de privacy of het milieu niet slecht beïnvloed. Het uiteindelijke effect van het werk moet het algemeen belang zijn.

## **Hoe weet je dat jouw ethische overwegingen overeenkomen met die van andere software-engineers?**

Een goede plek om te kijken is de ethische code of conduct van software engineering. Verder kun je met je team overleggen over wie waar staat op het gebied van ethiek. Wanneer iemands overwegingen anders zijn dan die van jou, wil deze persoon waarschijnlijk de applicatie anders laten werken dan wat jij in gedachte had. Om deze reden is het belangrijk dat jij en je teams ethische overwegingen overeenkomen.

## **Welke ethische aspecten spelen een rol in mijn project?**

### **Voorzie jij ethische conflicten die door uw software worden veroorzaakt? Wat voor soort?**

Het grootste ethische aspect dat speelt in onze groep is dat we een proces dat door mensen wordt uitgevoerd, gaan automatiseren. Het werk dat wij automatiseren is een groot deel van de taken die een ober uitvoert. Door onze applicatie zou een ober niet meer naar een tafel hoeven gaan om een bestelling te noteren en deze naar de keuken te brengen. Het enige dat nog moet gebeuren is dat een ober het eten naar de tafel hoeft te brengen en borden ophaalt. Een groot deel van de taken van een ober worden overgenomen. Een restaurant app kan dus zorgen dat bij een bestaand restaurant mensen ontslagen worden, omdat er minder personeel nodig zal zijn. Als het een compleet nieuw restaurant is, kan er niemand ontslagen worden. Wel zorgt het er bij een nieuw restaurant voor dat er minder banen beschikbaar zijn. De impact bij een nieuw restaurant zal dus kleiner zijn dan bij een bestaand restaurant. Maar alsnog zorgt ons programma voor minder werk beschikbaarheid. Ethisch gezien is dit een slechte applicatie, omdat er een grote kans is dat er een aantal obers wordt ontslagen of er een aantal banen nooit beschikbaar worden, terwijl er andere manieren waren om dit effect te verminderen of misschien zelf te vermijden.

### **Kun jij iets doen om deze ethische conflicten te minimaliseren?**

Dit ligt aan de app die de product owner van ons verwacht. In ons geval werd er verwacht dat er een bestelling geplaatst kon worden vanaf de tafel. Dit kan geïnterpreteerd worden op twee manieren.


1. Een klant kan vanaf de tafel een bestelling doen.
2. Een ober kan vanaf de tafel een bestelling doen.

Als wij de ethische conflicten minimaal wilden houden hadden wij waarschijnlijk voor de tweede optie moeten gaan. Maar toen wij aan het project begonnen hebben wij ons dit niet gerealiseerd en hebben optie 1 gemaakt. Ook al zal de tweede optie ook een effect hebben op het aantal medewerkers dat nodig zal zijn, het effect zal minder groot zijn dan bij optie 2.

# Requirements and Design

Voordat we begonnen met het bouwen van de applicatie hebben we user stories gemaakt. Deze hadden we nodig om te laten zien wat we wilde realiseren. Bij onze eerste sprint oplevering hebben we het gehad over deze user stories en de belangrijkste eruit gehaald. In de komende sprint opleveringen hebben we aan de hand van deze user stories aangegeven waar we aan zouden werken.

Een deel van de feedback die we over onze userstories hebben ontvangen:

**Timsel** 07-03-2022

bij bepaalde userstories is het niet duidelijk wanneer iets is afgerond dit geldt voor de stories met woorden zoals snel/sneller en efficient. kijk naar betere verwoordingen zoals automatisering

kijk naar ethiek, mensen worden waarschijnlijk ontslagen door deze applicatie.

personeelslid kan specifieker, want er zijn meerdere takken

story 6 moet gespecificeerd.

"zodat" voelt geforceerd, kan weggelaten worden bij bepaalde stories.

story 9: "in de toekomst" is nu al, dus dit moet gespecificeerd. Zorg dat dit concreet is voor de klant.

story 9: management info, hoeft niet recent.

story 12: woord "acties" is niet duidelijk. wat is een actie voor ons? wat zien wij als een fout?

voor het schrijven naar een database gebruik login.

story toevoegen: personeelslid kan een afgeronde bestelling herstellen.

story toevoegen: hoe openen we een sessie?

story toevoegen: manier van bestellen in userstories. (qr)

sessies herstellen zodat een klant zijn bestelling niet opnieuw hoeft te doen.

## User stories

We hebben onze user stories en non functional requirements opgezet in jira, dit is een project management tool. We hebben de user stories met een groen icoontje gemarkeerd en de non functional requirements met een blauw icoontje. Dit zag er zo uit:

▼ SP Sprint 4 25 mei – 22 jun. (7 issues)

	SP-10 Als klant wil ik op het menu zien welke producten nog op voorraad zijn, zodat het duidelijk is wat ik wel en niet kan bestellen.
	SP-8 Als klant wil ik via de applicatie kunnen betalen, zodat de betaling geautomatiseerd verloopt.
	SP-9 Als manager wil ik een management portal van alle restaurants.
	SP-63 Als restaurant wil ik dat de bestel website alleen vanuit het restaurant beschikbaar is, zodat mensen niet vanuit huis het proces verstoren.
	SP-62 Als restaurant wil ik dat de sessie bewaard blijft, zodat een klant geen betaling kan ontlopen
	SP-68 Documentatie gezet in onze organisatie
	SP-65 Maken Business proces

# Technische keuzes en architectuur

## Structuur

Wij hebben voor onze applicatie gekozen om met microservices te werken. De reden dat we deze structuur hebben gekozen was omdat, het voor een makkelijk uitbreidbare applicatie zorgt, door deze structuur konden we ons groepje opdelen in kleinere groepjes die elk aan een service werkte en wanneer er een service een bug zou hebben of niet werkte kon de rest van het team doorwerken. Dit waren voordelen waar wij gebruik van wilde maken.

## Backend

Voor onze backend hadden we de keus tussen verschillende OO talen. We hebben verschillende opties bekeken en zijn uiteindelijk op 2 talen uitgekomen. Dit waren C# en Java. Aangezien we microservices gebruikte zouden we ze beide kunnen gebruiken, maar we hebben de keus gemaakt om alleen java te gebruiken. Dit hebben we gedaan, omdat we allemaal al met C# hadden gewerkt en we iets nieuws wilde oppakken. Andere redenen waren dat Java op C# leek en we dus niet te veel tijd kwijt zouden zijn aan het leren van Java en het feit dat Java cross platform is. Dit betekent dat ons programma zal werken op computers die Microsoft Windows, Linux, macOS, en voor oude versies van Solaris runnen.

## Frontend

Voor onze frontend moesten we een javascript framework kiezen. De 3 waar we naar hebben gekeken waren React, Vue en Angular.

Uiteindelijk hebben we gekozen voor React. Deze keus hebben we gemaakt omdat, React de meest gebruikte javascript framework is en er dus veel documentatie over is. Aangezien de andere frameworks ook veel documentatie hebben was deze reden niet goed genoeg. Dus we hebben ook gekeken naar welke taal beginner vriendelijk was, van de 3 frameworks waren React en Vue het meest beginner vriendelijk. Voor ons was de keus nu tussen Vue en React.

We wilde ook gebruik maken van typescript, typescript is een superset van javascript. Typescript is een OO taal, aangezien we ervaring hebben met het werken met OO talen dachten wij dat dit een betere match zou zijn dan javascript. Andere voordelen zijn het feit dat typescript gebruik maakt van type annotations en dat typescript een compiler language is. Voor ons zorgde dit dat we gestructureerder codeerde en bugs meteen aan konden pakken.

## Database

Voor de keuze van de database hebben we een klein onderzoek gedaan naar relational en non-relational databases.

De voordelen van relationeel waren dat: alle data wordt opgeslagen in tables, de data moet gespecificeerd zijn in de rows en columns. Dit maakt een relationele database overzichtelijk. Op deze manier wordt dubbele data vermeden. Een nadeel is dat wanneer je de structuur van een table verandert dit kan leiden tot verlies van data. Dit is waar een non-relational database handig is. Aangezien data niet in een specifiek formaat wordt opgeslagen, kun je de table aanpassen zonder data te verliezen. Een nadeel van de non-relational database is dat je bepaalde queries niet uit kan voeren, dit geldt voor queries die afhankelijk zijn van relaties met andere tables.

Aangezien wij onze tables niet constant hoeven te veranderen en we een bestelling aan een klant willen kunnen koppelen is het voor ons dus duidelijk dat we een relationele database moeten kiezen. Ook kunnen we dubbele gegevens voorkomen. Een scenario waarin we dubbele data zouden hebben zou zijn bij 10 klanten die dezelfde bestelling plaatsen. Dit zou erg onhandig zijn in NoSQL, omdat je dan hetzelfde stuk data 10 keer volledig zou hebben uitgeschreven.

Na dit onderzoek zijn hebben we gekozen om MySQL te gebruiken, aangezien we hier de meeste ervaring mee hadden.

## Model

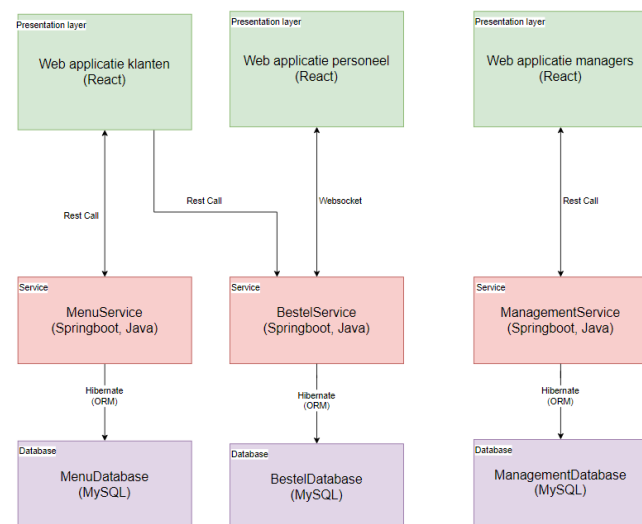
Dit is het model dat we hebben gemaakt om te laten zien hoe onze applicatie in elkaar zit. Wat hier te zien is, is dat we in totaal 3 frontends hebben, een voor de klant een voor de medewerkers en een management portal. We hebben voor elke frontend ook een aparte service gemaakt. Elke service heeft ook zijn eigen database.

Voor de klanten applicatie hebben we een menu gemaakt met een winkelwagen, dit is de frontend.

Deze stuurt de bestelling van de klant naar de bestelservice (backend van de personeels applicatie), ook haalt de klanten frontend het volledige menu op dat in de database is opgeslagen via een rest call.

De applicatie voor de medewerkers maakt gebruik van websockets. Zodra de bestelling van de klant binnenkomt wordt dit via de backend meteen naar de frontend en database gestuurd.

De management portal is nog niet geïmplementeerd, maar moest gebruik maken van een rest call wanneer een bestelling was afgerond bij de personeels applicatie om deze vervolgens op te slaan in de database en weer te geven in de portal.



## UI

### Testen

Voor de UI testen wilde ik gebruik maken van E2E testen, deze keuze heb ik gemaakt, omdat onze applicatie veel afhangt van user input en dit kun je mocken met dit soort tests. Om deze tests te schrijven heb ik gebruik gemaakt van Cypress. Cypress leek mij een goede tool om te gebruiken omdat, het met elke front-end framework werkt en het runt de tests in een browser. Dit zorgde ervoor dat ik in de testomgeving bepaalde acties uit kon voeren en deze makkelijk aan mijn testen toe kon voegen. Het enige nadeel dat waar ik tegenaan liep met cypress was dat de cypress tests niet voorbij github actions kwam, juist omdat ze een frontend nodig hadden.

Ik heb een aantal tests geschreven, In deze tests mock ik user input om op bepaalde knoppen te drukken en te kijken of ik de verwachte uitkomst krijg. Deze zagen er als volgt uit.

```
describe("redirect works correctly", () => {
  it("redirects to kitchen correctly", () => {
    cy.visit("/");
    cy.get('button').contains('Kitchen').click();
    cy.url().should('include', '/kitchen')
  })

  it("redirects to bar correctly", () => {
    cy.visit("/");
    cy.get('button').contains('Bar').click();
    cy.url().should('include', '/bar')
  })
})
```

### Design

Voor het design, wist ik van te voren al hoe ik het er ongeveer uit wilde laten zien. Ik heb hier een scherm schets van gemaakt. Ik ben hierna begonnen met het maken van kaartjes voor een product, dit kaartje moest dus bestaan uit een afbeelding, een manier om producten toe te voegen/te verwijderen en het totaal bedrag. Pas toen ik doorhad hoe ik met react deze aparte componenten samen kon voegen tot 1 component kon ik verder.


The sketch shows a shopping cart interface with the following elements:

- Four rows of input fields, each with a plus (+) icon, a minus (-) icon, and a delete (X) icon.
- The last row shows a quantity of 2 and a price of €5.
- Below the rows is a label "Totaal €5".
- At the bottom is a button labeled "Betaal".

Een van de eerste versies van de winkelwagen was een aparte applicatie voordat deze werd geïntegreerd in het menu. deze zag er zo uit.

## Bestelling voor tafel 0

2 items




Burgor

-

1

+

€ 10



Burgor

-

1

+

€ 10

TOTAAL





€ 10

Bestel

Zoals hier is te zien was de basis van het design er al, maar dit had verder geen functionaliteit. Verder waren de kaartjes te groot en paste het op deze manier niet bij het menu. Om het meer op de scherm schets te laten lijken hebben we de kaartjes kleiner gemaakt, zodat deze in de overlay paste die we lieten zien wanneer iemand vanaf het menu op de winkelwagen klikte. Verder hebben we de kaartjes op een nettere manier gescheiden van elkaar, een prijs van het product toegevoegd dat per product is en een delete actie die het kaartje, ongeacht van de hoeveelheid producten deze bevat verwijderd.

## Bestelling voor tafel 5

4 items

	Product	Price	Qty	Total	Actions
	Rosbief Salade	€2,49	<div><div>-</div><div>1</div><div>+</div></div>	€2,49	<div>×</div>
	Bladerdeeg Hapje	€5,49	<div><div>-</div><div>1</div><div>+</div></div>	€5,49	<div>×</div>
	Cheesecake	€4,99	<div><div>-</div><div>3</div><div>+</div></div>	€14,97	<div>×</div>
	Sinas	€3,59	<div><div>-</div><div>2</div><div>+</div></div>	€7,18	<div>×</div>
TOTAAL				€30,13	

Bestel

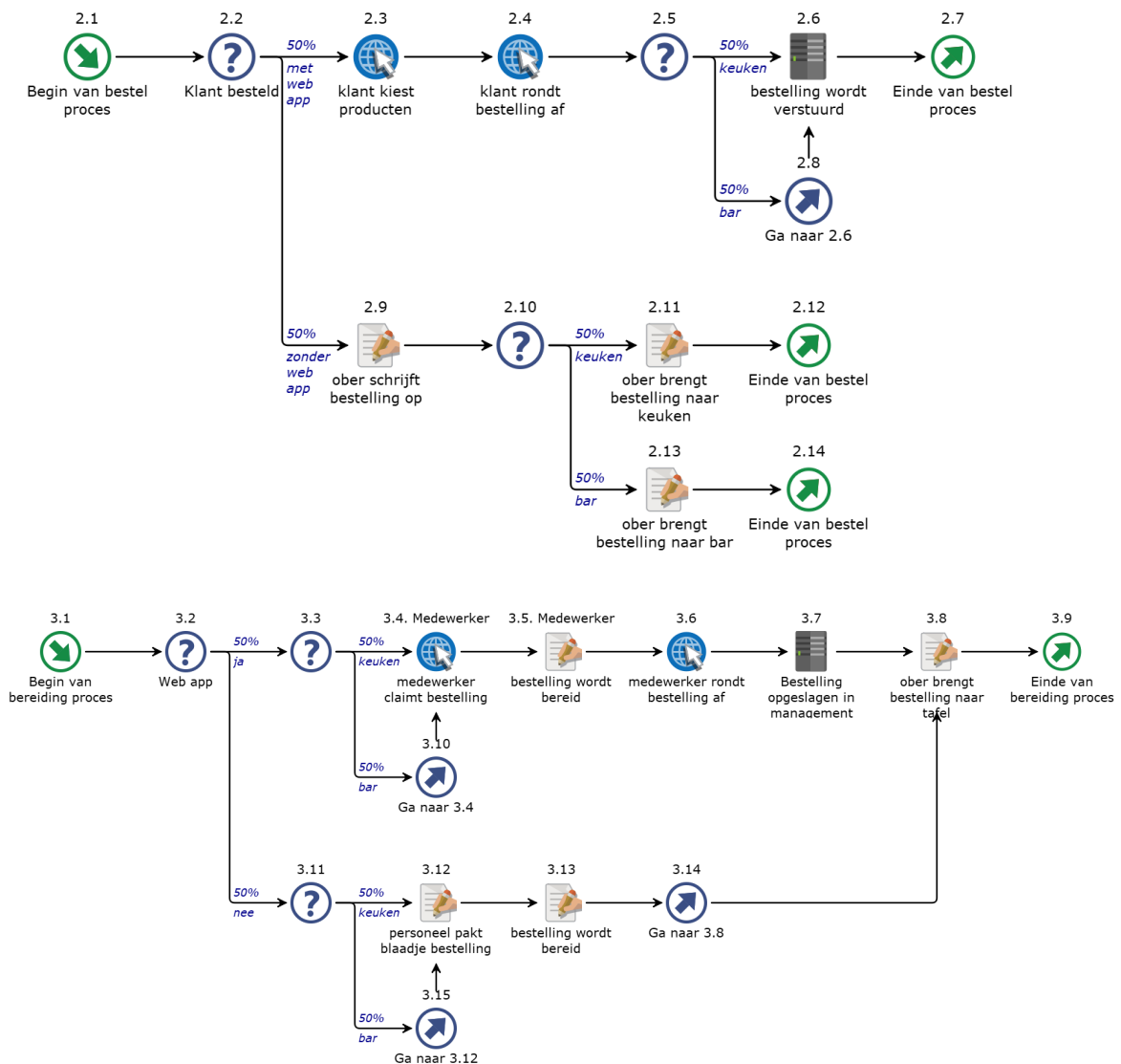


# Business processes

Ik heb voor het groepsproject een business proces gemaakt, dit proces bestaat uit 3 subprocessen. deze subprocessen zijn het verloop van het bestellen, de manier waarop de werknemers bestellingen claimen en het afronden van een sessie. Ook heb ik ervoor gezorgd dat er een vergelijking is tussen het gebruik van onze app en het verloop van dit proces zonder een applicatie.

## Waarom heb ik dit zo ingedeeld:

De reden dat ik dit zo heb ingedeeld is omdat, dit de processen zijn die onze app makkelijker zou maken in vergelijking met restaurants die geen gebruik maken van applicaties. Ook kan ik in deze 3 subprocessen laten zien hoe onze app dit volledige proces kan verbeteren of punten visualiseren waar het beter zou werken zonder de app, mocht dit het geval zijn. (de eerste keuze is altijd met of zonder web app)





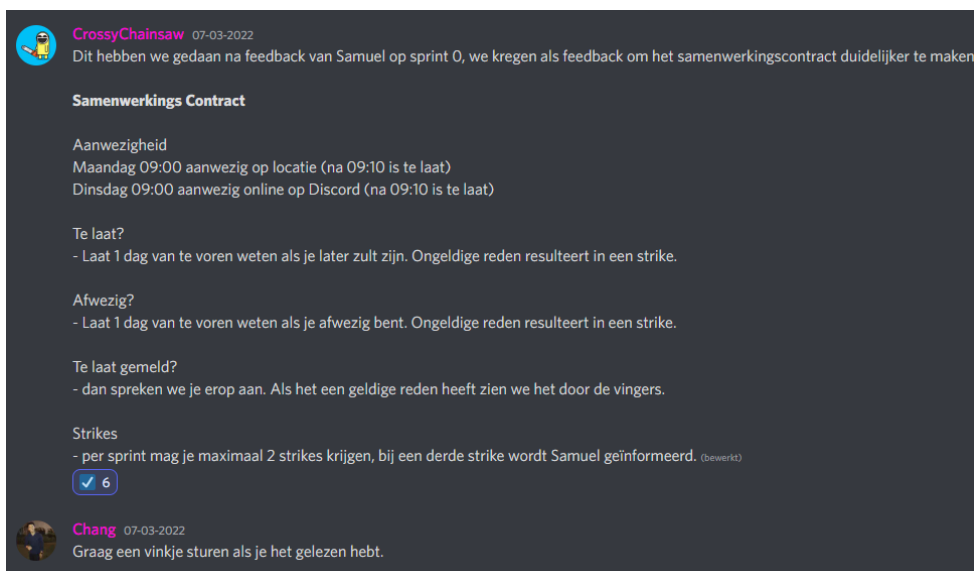
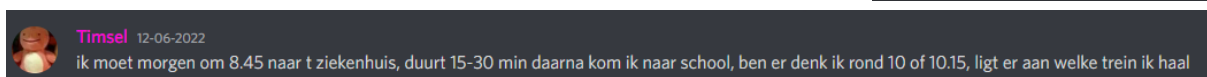
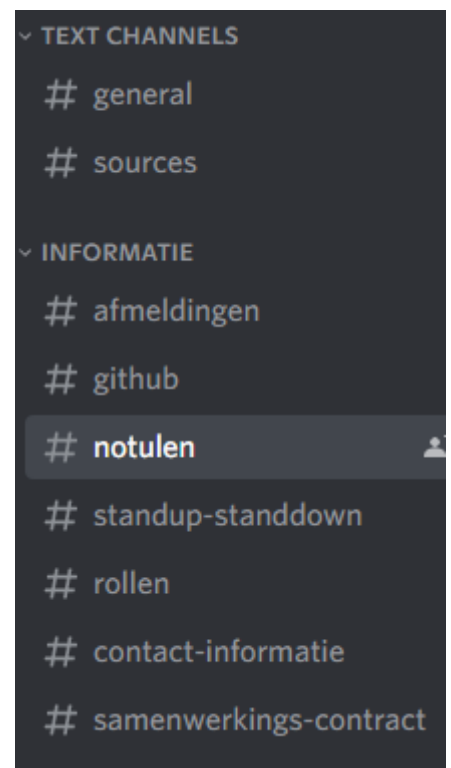
# Professional

## Wat werd er van ons verwacht

**Professional manner:** You develop software as a team effort according to a prescribed software methodology and following team agreements. You are able to track your work progress and communicate your progress with the team. You actively ask and apply feedback from stakeholders and advise them on the most optimal technical and design (architectural) solutions. You choose and substantiate solutions for a given problem.


## Team communicatie

Het platform dat ons team heeft gekozen om over te communiceren was Discord. We hebben onze eigen Discord server opgezet met aparte kanalen. Dit zorgde er voor dat we alles gescheiden konden houden en alles dat belangrijk was terug konden vinden. In deze server hebben wij ook ons samenwerkingscontract gezet, hierin hadden we regels over aanwezigheid gezet, aangezien aan het begin van het semester sommige leden van onze groep niet kwamen opdagen of uit het niets vertrokken. Om zeker te weten dat iedereen het had gelezen en het eens was met dit contract hebben we het getekend met een vinkje onder het bericht. Dit vinkje moest minimaal 6 keer gebruikt zijn onder het bericht. Om ons af te melden hebben we ook een apart kanaal gemaakt, zodat we de rest van de groep tijdig konden laten weten dat we later waren of niet konden komen. Dit zorgde er ook voor dat deze berichten niet in de weg kwamen in andere kanalen.



## Communicatie met de stakeholders






Naast het contact dat ons groepje had met de stakeholders tijdens de sprint opleveringen hebben we ook contact met ze gezocht via emails. We hebben deze taak aan Kaan gegeven, wanneer hij aanwezig was. Via mails hebben we een aantal dingen laten weten, zoals wanneer iemand van ons groepje stopte of we onze sprintplanning moesten aanpassen.



Gögcay, Kaan K.S.

Aan: sander.klijnsen@infosupport.com; Jeroen Uittenbosch

CC: Hoef, Mike M.J.A. van den; Esselink, Cas C.; Meijvogel, Tim T.; Angelov, Samuil S.













Ma 9-5-2022 10:12

Beste InfoSupport,

We hebben net helaas te horen gekregen dat David gaat stoppen met het semester. Hierdoor willen we onze sprintplanning een beetje aanpassen. We hebben besloten de volgende user story eruit te halen:  
*SP-9 Als manager wil ik een management portal van alle restaurants.*  
en de volgende user story nemen we mee:  
*SP-8 Als klant wil ik via de applicatie kunnen betalen, zodat de betaling geautomatiseerd verloopt.*


Dus onze sprint planning ziet er als volgt uit.


	SP-1 Als klant wil ik eten bestellen via de applicatie, zodat de bestelling geautomatiseerd verloopt.	 20	IN PROGRESS	
	SP-3 Als klant wil ik een overzichtelijk menu.	 40	IN PROGRESS	
	SP-10 Als klant wil ik op het menu zien welke producten nog op voorraad zijn, zodat het duidelijk is wat ik wel en nie...	40	TO DO	
	SP-8 Als klant wil ik via de applicatie kunnen betalen, zodat de betaling geautomatiseerd verloopt.	40	PRODUCT BACKLOG	


Met vriendelijke groet,

Groep 3

...

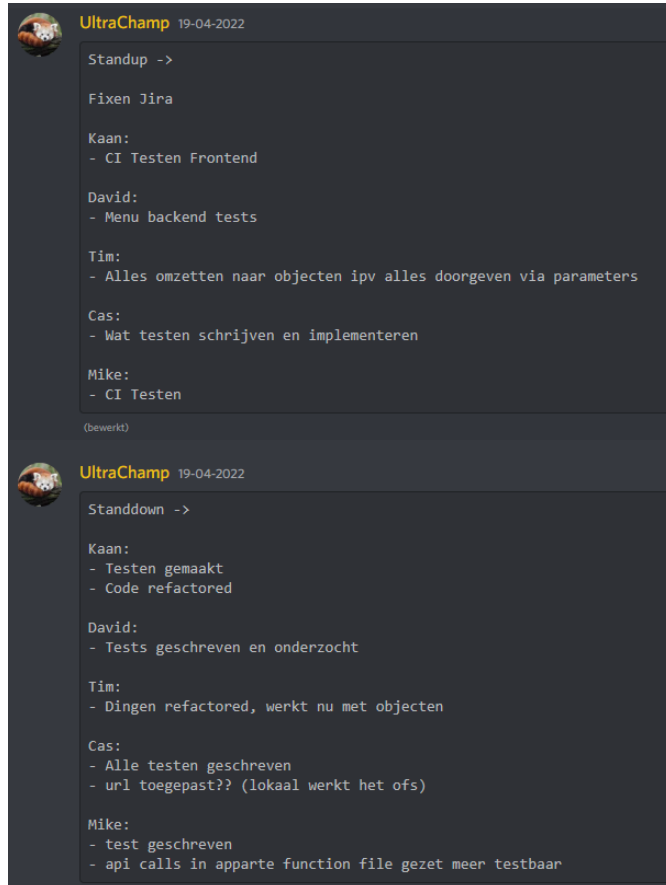
 Beantwoorden

 Allen beantwoorden

 Doorsturen

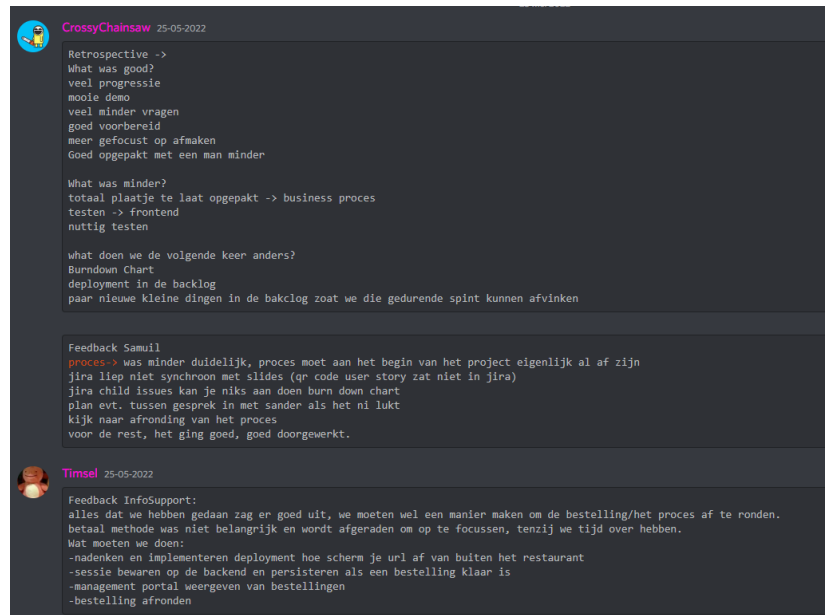
## Progressie van je team bijhouden

De vraag “wat zijn de andere in mijn groep aan het doen?” kwam vrijwel niet voor in ons groepje. Dit komt door onze stand ups, stand downs en onze jira bord. We hebben onze stand ups gebruikt om te zien wat iemand de vorige keer heeft gedaan en om te horen wat die persoon voor de dag heeft gepland. Dit werd gevisualiseerd in ons jira bord. Bij de stand down keken we naar wat er nieuw was gemaakt en of die persoon zijn doel had bereikt. Ook voor deze sessie hebben we een apart discord channel gemaakt waarin we dit bijhielden.



# Vragen voor feedback

We hebben de meeste feedback ontvangen tijdens onze sprint opleveringen, zowel van de stakeholders als onze docent. Ook hiervoor hadden wij een aparte Discord channel aangemaakt, waarin we dit op een gestructureerde manier bij konden houden. Na elke sprint oplevering hebben we deze feedback bekeken en verwerkt. Ook was het soms nodig om feedback te vragen buiten de sprint opleveringen, dit ging dan via een mail die een van ons verstuurde.

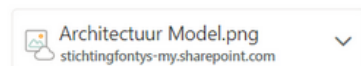


Gögcay, Kaan K.S.

Aan: sander.klijzen@infosupport.com <Sander.Klijzen@infosupport.com>; Jeroen Uittenbosch  
CC: Hoef, Mike M.J.A. van den; Meijvogel, Tim T.; Angelov, Samuil S.; Chang, David D.J.; Esselink, Cas C.



Ma 25-4-2022 13:44



Beste Heren,

We zouden graag willen weten wat jullie van het huidige architectuurmodel vinden, in het architectuur model is het volgende veranderd,

- We hebben de eureka server eruit gehaald
- We hebben nu duidelijkere connecties staan tussen applicaties
- Minder services

Met vriendelijke groet,

Groep 3

## Research

Ik heb dit semester 2 research reports gemaakt, een moest gaan over security en bij de andere mocht ik zelf kiezen wat mijn onderwerp zou zijn. De 2 hoofdvragen die ik heb bedacht zijn: **“Waarom zou je een authenticatie service gebruiken in plaats van een eigen login systeem?”** en **“Welke variatie van het elo systeem is geschikt voor mijn game?”**. Voor deze research reports heb ik me vooral aan 4 stappen gehouden. Deze stappen waren het bedenken en beantwoorden van een hoofdvraag, het bedenken en beantwoorden van deelvragen, mijn informatie valideren en het maken van een conclusie. Ook heb ik gekeken naar het DOT framework. Ik heb een aantal categorieën gebruikt van het domein library. De categorieën die voor mij belangrijk waren, waren vooral “best good and bad practices” (Organise activities aimed at sharing experiences. Depending on your topic, many best, good and bad practices may be available.) en “available product analysis” (Identify existing solutions that may solve the problem (or a part thereof) you are trying to fix with your solution. Decide if it is worth the effort to recreate their work, or whether it is better to simply buy it from them or embed their work in yours.). Dit komt omdat, ik bij mijn research naar elo veel gedeelde meningen tegen kwam onder spelers van veel games. Dit laat zien dat veel game developers dezelfde goede en foute keuzes maken die ik steeds weer tegen kwam. Door deze twee categorieën ben ik uiteindelijk tot mijn conclusie gekomen.

De research reports zijn te vinden in mijn organisatie, op de hoofdpagina staat er bij de inhoud een link naar beide. of vind ze hier:

Elo research: <https://github.com/S3IP/Portfolio/blob/main/Research/EloResearch.pdf>

Security research:

<https://github.com/S3IP/Portfolio/blob/main/Research/Security%20Research.pdf>