|  |  |
| --- | --- |
|  | **Boodschappenservice Bas Brengt Boodschappen** |

**Het bedrijf**

De heer Van der Heijden begint een boodschappenservice. Het bedrijf bezorgt in de omgeving Rotterdam. Om dat te kunnen doen hebben ze een groot magazijn in Ommoord. Daarin worden artikelen bewaard (bijvoorbeeld potten met aardbeienjam). Op het moment worden alleen niet bederfelijke etenswaren en non food artikelen aangeboden, Bas heeft nog geen koeling om snel bederfelijke artikelen te kunnen bewaren.

De klanten doen een verkooporder (bestelling) telefonisch bij de verkopers van Bas.

De magazijnmedewerkers zoeken de bestelde artikelen op en stoppen ze in een plastic tas. Daarna worden de tassen aan de fietsbezorgers gegeven en die brengen de artikelen naar de klanten toe.

Na verloop van tijd raken de artikelen op en moeten ze met behulp van een inkooporder bij de leveranciers besteld.

Wanneer die geleverd worden zorgt de magazijnmeester er voor dat de artikelen in de juiste stellingen bewaard worden.

BBB verdient geld doordat ze de artikelen goedkoper inkopen dan dat ze verkocht worden.

**Eisen project**

1. Product is voorzien van een **naam van de auteur**.
2. Elke sprint bevat een ontwerp voor de layout van de userinterfase (wireframe) en functionaliteit: userstorie en class diagram;
3. Alle code is **OOP**.
4. Voor elke method is een unittest (PHPUNIT) vereist.
5. Gebruik **PDO** en dus geen Mysqli.

**Afspraken:**

1. Sprints kunnen **niet herkanst** worden.
2. Sprints die **te laat** ingeleverd worden niet meer nagekeken en leveren dus geen punten op tenzij dit van te voren is gemeld.
3. Deze opdracht dien je **individueel** te maken en in te leveren. Uiteraard mag je met elkaar samenwerken om van te leren en bijvoorbeeld elkaars code te reviewen.
4. Indien code wordt **gekopieerd** van een andere student krijgt zowel degene die de code levert als de student die de kopie gebruikt een onvoldoende. Om te voorkomen dat je spullen gekopieerd worden maak je een private repository en geef je toegang tot de docent(en).

**Gesprekken met het personeel Must Have Could/Should Have Won’t Have**

***Bas van der Heijden (CEO)***

Er moet een programma gemaakt worden waarmee de inkopers, magazijnmeester, magazijnmedewerkers, bezorgers en de verkopers van BBB kunnen werken. Zij leggen in de andere gesprekken uit wat ze nodig hebben om hun werk goed te kunnen doen.

Het is mooi als ze bij de start van het programma een hoofdmenu krijgen te zien. Dan kunnen ze klikken op een link om bij een ander scherm te komen waar ze alleen de keuzen vinden die zij nodig hebben om hun werk goed te kunnen doen, dit is een *must have*.

Het zou nog mooier zijn als de gebruikers kunnen inloggen en dan automatisch bij het scherm terechtkomen dat ze nodig hebben, dat is een *could have*.

Elke pagina van het programma moet beginnen met ons logo en de volledige naam van het bedrijf. Op de footer komt dan ons inbelnummer te staan met de melding dat het een gratis nummer is dat alleen vanuit Nederland gebeld kan worden. De verkoper vertelt wat ons [telefoonnummer](#telefoonnummer) is. De andere *must have*  is dat elk soort werknemer een scherm krijgt met de mogelijkheden die voor dat werk nodig zijn.

***Rosita Soeltaansingh-Badri – afdelingshoofd ICT***

De menu’s waar Bas het over gehad heeft moeten met een extern css gemaakt worden. De applicatie moet gemaakt worden in PHP en zoveel mogelijk in OOP zijn. Ik heb ook nagedacht over de tabellen die je nodig hebt. Die staan verderop in de [datadictionary](#_Datadictionary). In ieder geval hebben we de tabellen ARTIKELEN, LEVERANCIERS,KLANTEN,VERKOOPORDERS en INKOOPORDERS nodig. Uiteraard mag je de database verbeteren en aanvullen.

Met de database moet een PDO-connect gemaakt worden om toegang tot de database te krijgen, dat is ook een *must have*.

Voor elke tabel moet een CRUD gemaakt worden, dat is een *must have*.

***Jaap Schaap – verkoper***

Ik wordt door de klanten gebeld op telefoonnummer 0800.11.11.216. Dat is een gratis nummer. De klant vraagt dan bijvoorbeeld om twee potten aardbeienjam. Meestal geeft de klant het artikelid door. Ik kijk dan in de database of er nog twee potten op voorraad zijn. Daarvoor wil ik in ieder geval kunnen zoeken op artId, dat is een must have. Het is ook handig als ik kan zoeken op artikelomschrijving, dat is een *could have*. Als het artikel op voorraad is verander ik de aanwezige voorraad, ik haal twee potten aardbeienjam af van het aantal in voorraad. Ik denk dat de magazijnmeester deze handelingen ook wil kunnen uitvoeren, misschien kunnen we samen deze programma’s gebruiken.

Met een ander programma (dat is al af en daar hoef je niets aan te doen, dat is een *won’t have* 😉) geef ik aan de magazijnmedewerkers door dat ze het artikel uit de stellingen van het magazijn moeten halen en in een tas moeten doen die later aan de bezorgers wordt doorgegeven.

Als het om een nieuwe klant gaat moet ik hem kunnen toevoegen aan de database. Soms verhuist een klant, daarom moet ik ook zijn gegevens kunnen aanpassen. Verder wil ik kunnen zoeken op klantid. Als ik een bepaalde klant niet kan vinden dan wil ik wel dat ik ook een overzicht met alle klanten kan zien. Wanneer iemand geen klant meer wil zijn moet ik hem uit het bestand kunnen halen. Dat zijn ook *must haves.*  Een *could have* is dat ik graag wil kunnen zoeken op de klantnaam of op zijn postcode.

Ik moet ook de verkooporders bijhouden, daar geldt eigenlijk hetzelfde voor, ik moet een verkooporder kunnen maken, ze allemaal kunnen inzien, wijzigen en verwijderen. Verder wil ik kunnen zoeken op het id van de verkooporder (*must haves*). Eventueel is het fijn als ik een verkooporder op de datum of het klantnummer kan laten zoeken (*could haves*).

***Andesh Hari – magazijnmeester***

Als er een nieuw artikel wordt opgenomen in het assortiment moet ik dat kunnen toevoegen aan de database. Verder moet ik het af en toe kunnen aanpassen, soms verandert de prijs. Er worden ook artikelen uit de verkoop gehaald, die moet ik dus ook kunnen verwijderen uit de database. *Must haves.*

Om te kunnen inventariseren wil ik een artikellijst kunnen zien waarop alle artikelen staan. Verder wil ik ook elk artikel kunnen vinden op artikelid, dat is ook een *must have*. Eventueel wil ik de artikelen ook kunnen vinden door de omschrijving van het artikel te zoeken, dat is een *could have*.

***Leonie de Heer - magazijnmedewerkster***

Eigenlijk hoef ik niet zoveel te weten, ik krijg picklisten via het andere programma dat al gemaakt is en waar jullie niets aan hoeven te doen (*won’t have*).

Maar ik vind het wel fijn als ik een overzicht van alle artikelen kan krijgen met hun locaties, dan hoef ik niet steeds te zoeken naar een artikel. Verder zou ik het fijn vinden om een artikel te kunnen zoeken op artikelid (*must haves*). Als dat ook nog kan op artikelomschrijving ben ik helemaal gelukkig (*could have*).

Verder wil ik de verkooporderstatus kunnen veranderen wanneer ik het artikel ga picken en wanneer ik het aan de bezorger geef.

***Redouan Sanu - bezorger***

De magazijnmedewerker geeft me een tas met een klantnummer. Ik moet dan op mijn telefoon uitzoeken wat het adres van die klant is, dat doe ik met behulp van het klantid, dat is echt een *must have*. Daarna breng ik de tas naar de klant toe, daarvoor moet ik de status van de verkooporder kunnen aanpassen. De klant heeft al betaald bij het doen van de bestelling, maar gelukkig krijg ik meestal toch een fooitje.

***Peter de Jager – inkoper***

Gelukkig worden er heel veel artikelen verkocht bij BBB. Iedere keer als een artikel opraakt moet ik het bijbestellen bij onze leveranciers. Ik doe de bestellingen altijd per email. Zodra ik een bestelling gedaan heb zet ik hem bij de inkooporders.

Mijn werk begint met het bekijken van een overzicht van alle artikelen die besteld moeten worden (*must have*). Daarvoor heeft elk artikel een aanwezige voorraad en een minimum aantal in voorraad. Zodra er minder dan het minimumaantal in voorraad is ga ik bestellen. Ik mail naar de leverancier en ik maak een nieuwe verkooporder aan (ook een *must have*). Ik kan niet zomaar iets bijbestellen, ik moet eerst uitrekenen hoeveel we nodig hebben. Daarvoor kijk ik naar het maximum aantal in voorraad. Er is altijd nog wel wat in voorraad, de bestelhoeveelheid wordt dan het maximum aantal – wat nog in voorraad is.

In het magazijn is er bijvoorbeeld plaats voor 100 potten aardbeienjam (maximumvoorraad). Er moet besteld worden als er minder dan 10 potten aanwezig zijn (minimumvoorraad), op die manier is Bas niet uitverkocht voordat de nieuwe voorraad geleverd is. De bestelhoeveelheid is dan 100 – 10 = 90 potten. Als die geleverd zijn is de stelling weer vol.

Je begrijpt dat ik een overzicht van alle artikelen moet hebben en ik wil kunnen zoeken op een bepaald artikelnummer. Verder wil ik erg graag een overzicht hebben waarop alleen de artikelen staan die ik moet bestellen, dat zijn drie *must haves*.

Als *should have* heb ik de wens om een bestellijst te krijgen, dat is een overzicht waarop de artikelnummers staan met het bestelaantal en het emailadres van de leverancier, maar dan alleen van de artikelen die ook echt besteld moeten worden.

Omdat ik ook alles moet bijhouden van de inkooporders wil ik die kunnen toevoegen, allemaal kunnen zien, wijzigen, verwijderen en kunnen zoeken op inkooporderid. Ook dit zijn *must haves*.

Verder ben ik ook verantwoordelijk voor het leveraniersbestand. Ik moet ze kunnen toevoegen, een overzicht kunnen afdrukken, wijzigen, verwijderen en zoeken op het leveranciersid. Dit zijn voor mijn werk ook echte *must haves*.

Een leuke *should have* is om te kunnen zoeken op leveranciersnaam.

**Dit project is opgedeeld in 14 user stories.**

1. Aanmaken tabellen project BAS en vullen met voorbeeld data.

2. Voeg een nieuwe klant toe. (INSERT)  
3. Artikelen aanmaken(INSERT)  
4. Verkooporders inzien (SELECT)  
5. Artikelen inzien (SELECT)  
6. Artikelen in een verkooporder plaatsen (INSERT/JOIN)

7. Maak een inkooporder mogelijk.(INSERT/JOIN)

8. Orderstatus bijwerken (SELECT/UPDATE)

9. Zoeken op een klant id (SELECT)

10. Klanten, leveranciers en producten verwijderen (DELETE)

11. Verkooporders en inkooporders verwijderen (DELETE)

12. Artikel bijwerken (UPDATE)

13. Klant gegevens bijwerken (UPDATE)

14. Verkooporder bijwerken (UPDATE)

# Datadictionary

Hier onder volgt een overzicht van de minimaal aantal tabellen.  
Aanvullingen of verbeteringen zijn uiteraard toegestaan.

Tabel artikel

|  |  |  |
| --- | --- | --- |
| artId | Int | Primaire unieke sleutel, autonummering, not Null |
| artOmschrijving | Varchar 12 lettertekens | De naam van het artikel, not Null |
| artInkoop | Decimal 3 cijfers, 2 decimalen | Inkoopprijs |
| artVerkoop | Decimal 3 cijfers, 2 decimalen | Verkoopprijs |
| artVoorraad | Int | Aantal artikelen op voorraad, not Null |
| artMinVoorraad | Int | Aantal artikelen dat minimaal op voorraad moet zijn, not Null |
| artMaxVoorraad | Int | Aantal artikelen dat maximaal op voorraad mag zijn, not Null |
| artLocatie | Int | Het nummer van de stelling waarin het artikel te vinden is |

Tabel leverancier

|  |  |  |
| --- | --- | --- |
| levId | Int | Primaire unieke sleutel, autonummering, not Null |
| levNaam | Varchar 15 lettertekens | De bedrijfsnaam, not Null |
| levContact | Varchar 20 lettertekens | Voor- en achternaam contactpersoon |
| levEmail | Varchar 30 lettertekens | Emailadres van de contactpersoon, not Null |
| levAdres | Varchar 30 lettertekens | Straat en huisnummer leverancier |
| levPostcode | Varchar 6 lettertekens |  |
| levWoonplaats | Varchar 25 lettertekens |  |

Tabel inkooporder

|  |  |  |
| --- | --- | --- |
| inkOrdId | Int | Primaire unieke sleutel, autonummering, not Null |
| levId | Int | Primaire samengestelde sleutel, samen met artId uniek |
| artId | int | Primaire samengestelde sleutel, samen met levId uniek |
| inkOrdDatum | Date | Datum waarop de inkooporder is gedaan |
| inkOrdBestAantal | int | Het aantal artikelen dat besteld is. |
| inkOrdStatus | Boolean | Is het artikel al geleverd? |

Tabel verkooporder

|  |  |  |
| --- | --- | --- |
| verkOrdId | Int | Primaire unieke sleutel, autonummering, not Null. |
| klantId | Int | Primaire samengestelde sleutel, samen met artId uniek. |
| artId | int | Primaire samengestelde sleutel, samen met klantId uniek. |
| verkOrdDatum | Date | Datum waarop het artikel is gekocht. |
| verkOrdBestAantal | int | Het aantal artikelen dat besteld is. |
| verkOrdStatus | Int | 1 = genoteerd in deze tabel  2 = magazijnmedewerker verzamelt het artikel (picking)  3. tas met artikel is bij de bezorger  4. tas met artikel is afgeleverd bij de klant  Not null, standaardwaarde 1 |

Tabel klant

|  |  |  |
| --- | --- | --- |
| klantId | Int | Primaire unieke sleutel, autonummering, not Null |
| klantNaam | Varchar 20 lettertekens | Voor- en achternaam van de klant |
| klantEmail | Varchar 30 lettertekens | Emailadres van de klant, not Null |
| klantAdres | Varchar 30 lettertekens | Straat en huisnummer klant, not Null |
| klantPostcode | Varchar 6 lettertekens |  |
| klantWoonplaats | Varchar 25 lettertekens |  |