Thijs de Rooij & Taoufik Sadaane

Brabotica 2.0 - Versie 0.1

Proftaakdocument

AVD Back-end development en databases

Inhoudsopgave

[Versiehistorie 2](#_Toc64657401)

[1 Algemene beschrijving 3](#_Toc64657402)

[2 Aanpak 4](#_Toc64657403)

[2.1 Planning 4](#_Toc64657404)

[2.2 Afspraken 5](#_Toc64657405)

[2.3 Eisen en wensen 5](#_Toc64657406)

[3 Functionele benodigdheden 7](#_Toc64657407)

# Versiehistorie

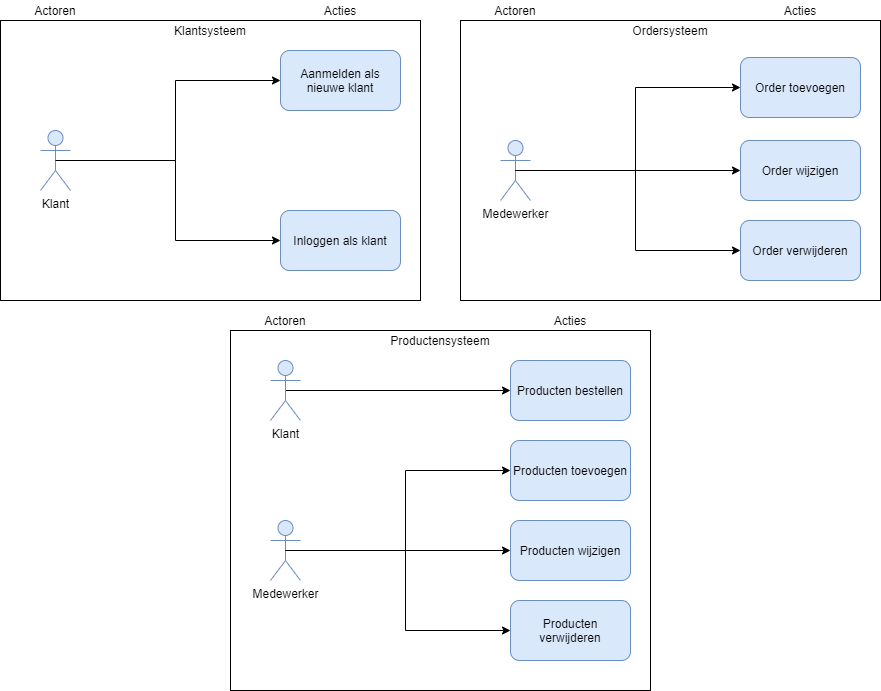
|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur** | **Aanpassingen** |
| 0.1 | 14-02-2021 | Thijs de Rooij | Opzet van het initiële document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1 Algemene beschrijving

De database applicatie van Brabotica 2.0 moet het mogelijk maken voor klanten en medewerkers om op een makkelijke manier gebruik te maken van de webshop van Brabotica 2.0. De volgende subsystemen zijn aanwezig in de database applicatie:

1. Een klantensysteem waarin klanten zich kunnen aanmelden en waar ze kunnen inloggen;
2. Een productensysteem waarin klanten producten kunnen bestellen en medewerkers producten kunnen toevoegen, wijzigen en verwijderen;
3. Een ordersysteem waarin medewerkers orders kunnen toevoegen, wijzigen en verwijderen.

In het schema hieronder zijn de verschillende acties weergegeven en wie toegang hebben tot deze acties.



Figuur 1: Schema algemene beschrijving

# 2 Aanpak

In dit hoofdstuk staat beschreven hoe deze proftaak wordt uitgewerkt en wat nodig is om deze proftaak tot een goed einde te brengen. Het eerste deelhoofdstuk beschrijft de planning van de proftaak. In het tweede deelhoofdstuk staan de afspraken die gemaakt zijn binnen de proftaak. Het laatste deelhoofdstuk beschrijft de eisen en wensen welke gesteld zijn aan de proftaak.

## 2.1 Planning

In de planning staat begrepen wat elke week gebeurt om de proftaak tot een succesvol einde te brengen. De planning is opgedeeld in weken om een concreet overzicht te geven welke taken wanneer worden verricht.

|  |  |  |
| --- | --- | --- |
| **Week** | **Lesstof** | **Proftaak werkzaamheden** |
| 1 | Inleiding relationele datapersistentie | Inlezen proftaakomschrijving |
| 2 | Gegevensmodelering | Vormen groepen en bepalen eerste opzet documenten |
| 3 | *Vakantie* | Thijs: Maken eerste opzet proftaakdocument |
| Taoufik: Maken eerste opzet ERD |
| 4 | P&OC |  |
| 5 | Data integriteit  Werken met SQL |  |
| 6 | Fysiek ontwerp  Triggers, Constraints & Procedures  Over indices |  |
| 7 | P&OC |  |
| 8 | Statistische functies  Joins  Normaliseren  PHP & MySQL |  |
| 9 | *Assessments* |  |
| 10 | *Assessments* |  |

## 2.2 Afspraken

Binnen de proftaak dienen een afspraken gemaakt te worden om ervoor te zorgen dat het eindproduct van gewenste kwaliteit is. In de volgende tabel is te vinden welke afspraken zijn gemaakt. Als nieuwe afspraken worden gemaakt wordt de tabel aangevuld.

|  |  |
| --- | --- |
| **Afspraak** | **Motivatie** |
| Engels is de voertaal in de code en de verschillende modellen | Dit is makkelijker, omdat belangrijke sleutelwoorden in de code en in de modellen ook Engelstalig zijn |
| Namen van tabellen zijn altijd in het meervoud | In tabellen staat een hoeveelheid data, dus is de meervoudsvorm logisch |
| Namen van tabellen zijn altijd met een kleine letter | Om geen verwarring te veroorzaken met waarden in de tabellen |
| Namen van kolommen zijn altijd in het enkelvoud | Kolommen geven informatie over een enkele eigenschap van een entiteit |
| Namen van kolommen zijn altijd met een kleine letter | Om geen verwarring te veroorzaken met waarden in de tabellen |
| Programma om diagrammen en modellen te maken is Draw.io | Dit programma is gratis en toegankelijk voor iedereen |
| Voor het beheren van code wordt Github gebruikt | Github is gratis en is een handige manier om alles overzichtelijk te houden |

## 2.3 Eisen en wensen

|  |  |  |
| --- | --- | --- |
| **Nummer** | **Specificatie** | **Toelichting** |
| **Klantensysteem** | | |
| 1 | Klanten moeten een account kunnen aanmaken | Om orders met producten succesvol naar klanten te kunnen sturen moeten klanten in staat zijn om een account aan te maken |
| 2 | Klanten moeten de gegevens van hun account aan kunnen passen | Dit moet voor klanten mogelijk zijn omdat de gegevens van een klant kunnen veranderen. Denk hierbij bijvoorbeeld aan een verhuizing |
| 3 | Klanten moeten hun account kunnen verwijderen | Het kan voorkomen dat klanten niet meer voornemens zijn om producten te kopen bij Brabotica 2.0, of ze willen niet meer dat hun gegevens bewaart worden |
| **Ordersysteem** | | |
| 1 | Het moet voor klanten mogelijk om een order toe te voegen aan het systeem | Wanneer een klant iets besteld heeft moet de database uitgebreid worden met de desbetreffende order |
| 2 | Klanten moeten een order kunnen annuleren | Het kan voorkomen dat de klant zich bedenkt na het plaatsen van een order. Deze zou dan geannuleerd moeten kunnen worden. Belangrijk is dat alleen orders van de desbetreffende klant geannuleerd kunnen worden |
| 3 | Medewerkers moeten orders van klanten kunnen inzien | Dit is noodzakelijk, want om een order succesvol te kunnen laten verlopen is het voor een medewerker nodig om te zien wat er besteld is |
| 4 | Medewerkers moeten orders kunnen toevoegen | Het kan soms voorkomen dat een klant niet in staat is om een order te plaatsen via het internet. Hierdoor is het voor een medewerker noodzakelijk om handmatig een order te kunnen invoeren |
| 5 | Medewerkers moeten een order kunnen wijzigen | Het kan voorkomen dat bepaalde producten niet voorradig zijn en een vervangend product geleverd moet worden. In dat geval moet de medewerker de order kunnen wijzigen |
| 6 | Medewerkers moeten een order kunnen verwijderen | Soms kan het voorkomen dat een order geplaatst is en niet aan de betalingsverplichting is voldaan. In dit geval moet een medewerker de order kunnen verwijderen |
| **Productensysteem** | | |
| 1 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 3 Functionele benodigdheden