**Software Architectuur Document (SAD)**

Project: BestelMar

Projectteam: Rik

Versie: 0.2

Versiedatum: 26 November 2019

Status: Concept

# Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Wijzigingen | Auteur | Datum |
| 0.1 | Opzet document | Nico Kuijpers | 14 dec 2017 |
| 0.2 | Beschrijving specificatie van interfaces aangepast. | Rik Peeters | 26 Nov 2019 |
| 0.3 | Beschrijving C2, C3, C4 en specificatie van interfaces aangepast. | Rik Peeters | 26 Nov 2019 |

Inhoud

Documenthistorie 2

H1 Inleiding 4

H2 Systeem Context (C1) 5

H3 Containers en technologiekeuzes (C2) 6

H4 Componenten (C3) 7

H5 Klassendiagrammen en sequence diagrammen (C4) 8

H6 Persistentie per component 9

H7 Specificatie van interfaces 10

# H1 Inleiding

In dit document wordt de globale technische implementatie en architectuur van de applicatie uiteengezet. Voor het domeinmodel en functionele requirements verwijzen we u naar het analysedoucment waarin deze worden beschreven.

De applicatie gaat dienen als een order systeem voor bijv. bioscopen en cafés. De applicatie moet het makkelijker maken voor bijv. in een bioscoop een drankje te bestellen. Wanneer een klant iets besteld zal dit live worden gedeeld met de medewerkers. Dit zorg voor een nog fijnere bestel ervaring en draagt bij aan een innovatieve manier van horeca.

De live communicatie tussen het backend (API & Web Sockets) en front-end (Vue) loopt via web sockets, overige frequente communicatie verloopt via de REST API

# H2 Systeem Context (C1)

In dit hoofdstuk wordt de context van de applicatie beschreven aan de hand van een context diagram.

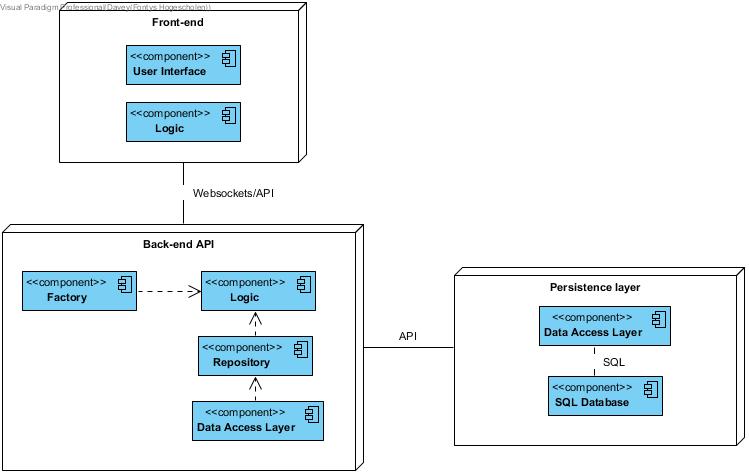
*Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving*

# H3 Containers en technologiekeuzes (C2)

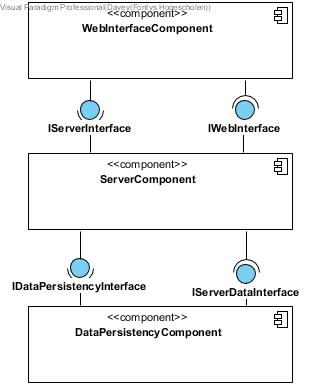
De BestelMar applicatie maakt gebruik van 3 containers. Als front-end wordt Vue js als technologie gebruikt. Hier is voor gekozen omdat met dit framework makkelijk realtime web-applicaties ontwikkeld kunnen worden.

Het back-end communiceert met het front-end m.b.v. websockets om data realtime weer te geven en maakt gebruik van een meerlaags architectuur. De minder belangrijke en frequente communicatie kan via REST interfaces verlopen. Het back-end communiceert vanuit de Data Access Layer met de Persistence Layer m.b.v. REST interfaces. Hier is voor gekozen om de database veilig te stellen door er een extra laag aan toe te voegen. Ook is het hierdoor makkelijker om, indien van overbelasting van het back-end, meerdere back-ends te gebruiken.

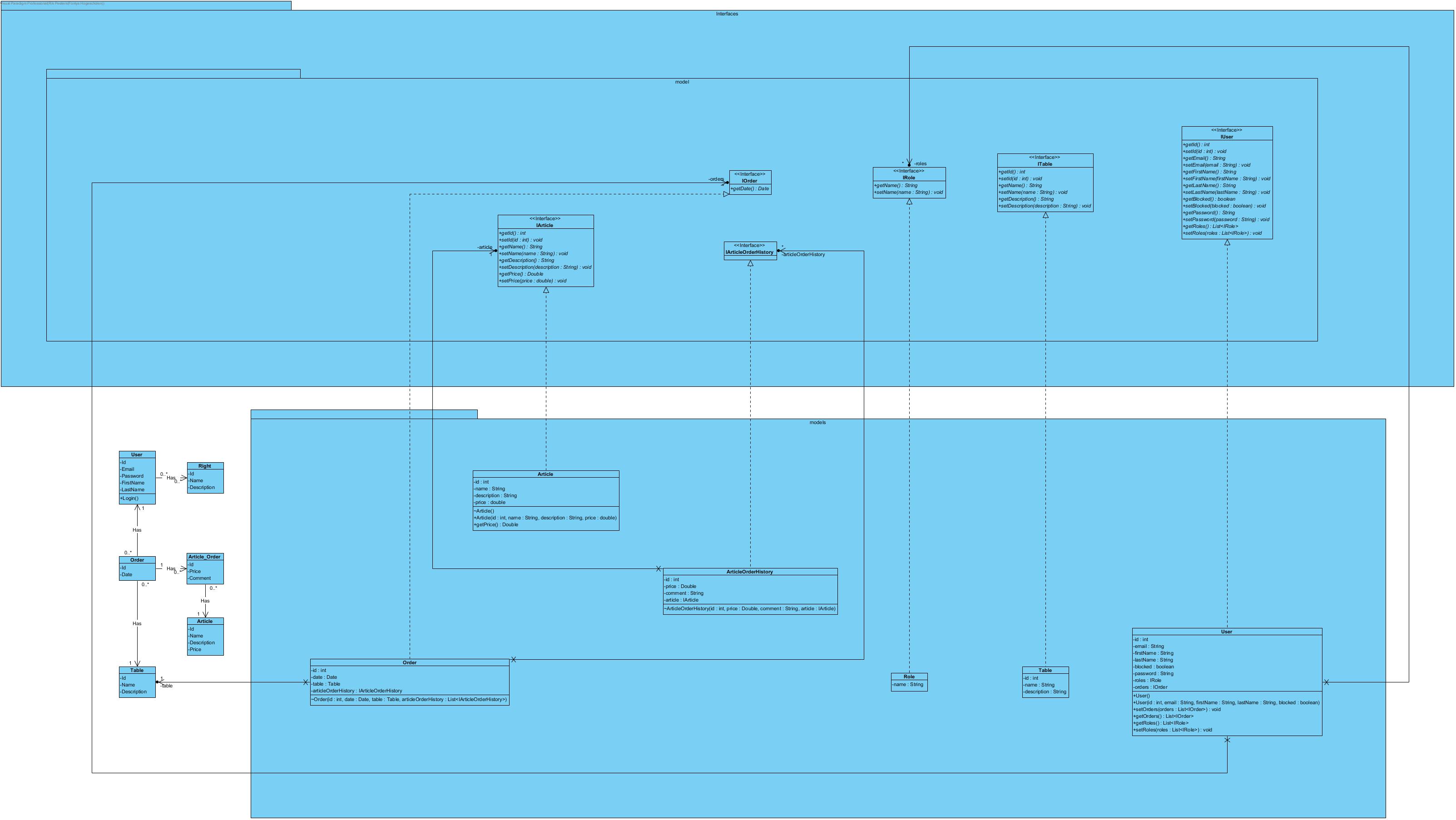


# H4 Componenten (C3)

De BestelMar applicatie bestaat uit 3 componenten. De WebInterface wordt gebruikt door de gebruiker en communiceert met het server component. Het server component communiceert met het DataPersistency component. Het DataPersistency component bevat de SQL-database. De database is dus losgekoppeld van de server.

**

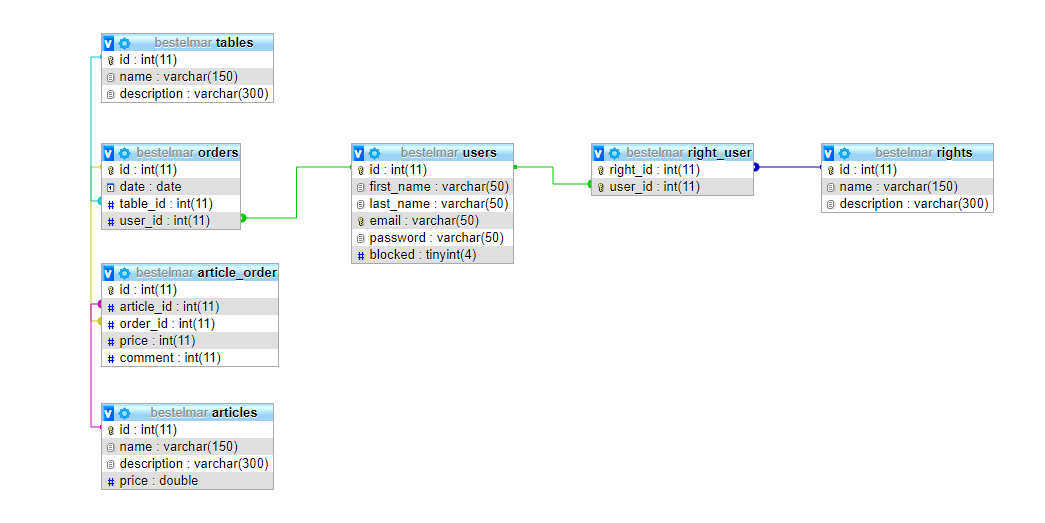
# H5 Klassendiagrammen (C4)

**

# H6 Persistentie per component

De applicatie zal persistentie hebben deze wordt gerealiseerd doormiddel van een mysql database. De objecten die worden opgeslagen staan in het klassendiagram. Hieronder is een ERD-diagram te vinden van de relaties tussen de verschillende tabellen.

ERD Diagram



# H7 Specificatie van interfaces

De implementatie van de article is volgens de crud standaard alle opeenvolgende crud objecten volgen dezelfde endpoints standaard.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Protocol | URL | Returns |
| GetArticleList | Get | /article | List of Articles |
| GetArticle | Get | /article/{id} | Found article |
| CreateArticle | Post | /article | Created article |
| UpdateArticle | Put | /article/{id} | Updated article |
| DeleteArticle | Delete | /article/{id} | Deleted article |