Dokumentation

EAI „ShopStantly“

Eduart Bunjaku, Lars Lutz, Robin Widmer, Jan Müller

Inhalt

[2 Ausgangslage 3](#_Toc532724162)

[3 Fazit 3](#_Toc532724163)

[4 Datenmodellierung 4](#_Toc532724164)

# Ausgangslage

Die Aufgabe dieses Projekts ist es vier unabhängige Microservices miteinander zu verbinden. Folgende Microservices wurden für die Erreichung des Ziels erstellt: E-Shop, Inventar, Versand und Zahlung.

Das Ziel ist über einen Chatbot eine Bestellung aufzugeben und der Roboter eine Nachricht an den E-Shop sendet. Somit sollte der Rest automatisiert im Hintergrund laufen und lediglich eine Bestätigung mit der Tracking und Rechnungsnummer zurücksenden, welche vom Bot ausgegeben werden.

Für die Zielerreichung wurden folgende Hilfsmittel verwendet:

**Github/Githubdesktop**

Wurde verwendet als Repository und Speicherort für Dokumentation und andere Dateien wie UseCase Diagramme von Enterprise Architect.

**Heroku**

Heroku dient als Cloudplatform, um den Code zu speichern wie auch auszuführen.

**H2**

Wurde als Datenbank verwendet.

**Spring Boot**

Dient als Verbindungsstück für die einzelnen Microservices.

**Eclipse**

Wurde als IDE verwendet.

**MySQL Wrokbench**

Das ERD wie auch Vorgänger Konzepte der Datenbankstruktur wurden in der MySQL Workben erstellt.

**DialogFlow**

Wurde benutzt um den ChatBot zu programmieren.

**Enterprise Architect**

Wurde verwendet, um die Use Cases zu erstellen.

# Datenmodellierung

Das Entity-Relationship-Diagram (ERD) wurde mit “MySQL-Workbench” erstellt. Es gibt zu jedem Microservice eine eigene Datenbank.

Datenbanken:

* E-Shop
* Inventar
* Versand
* Zahlung

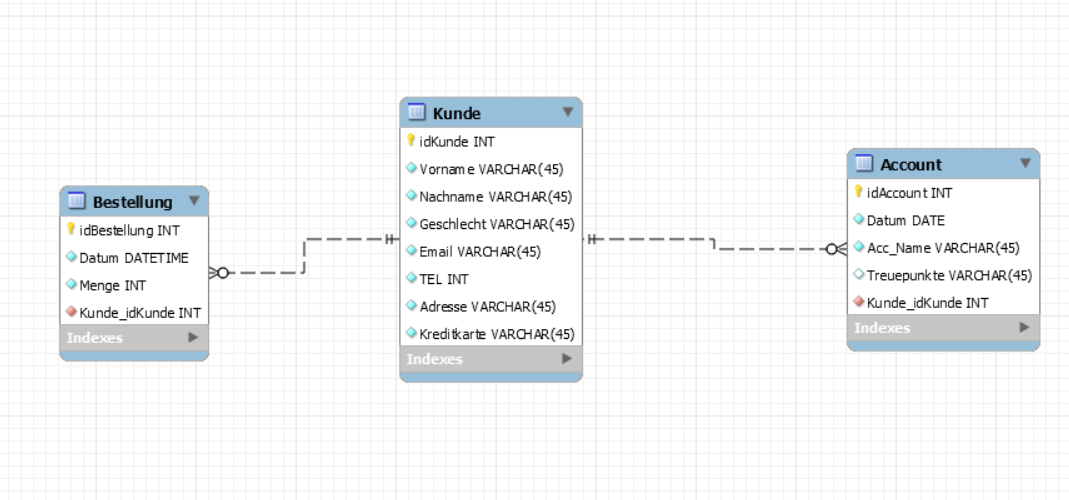


Abbildung : e-Shop

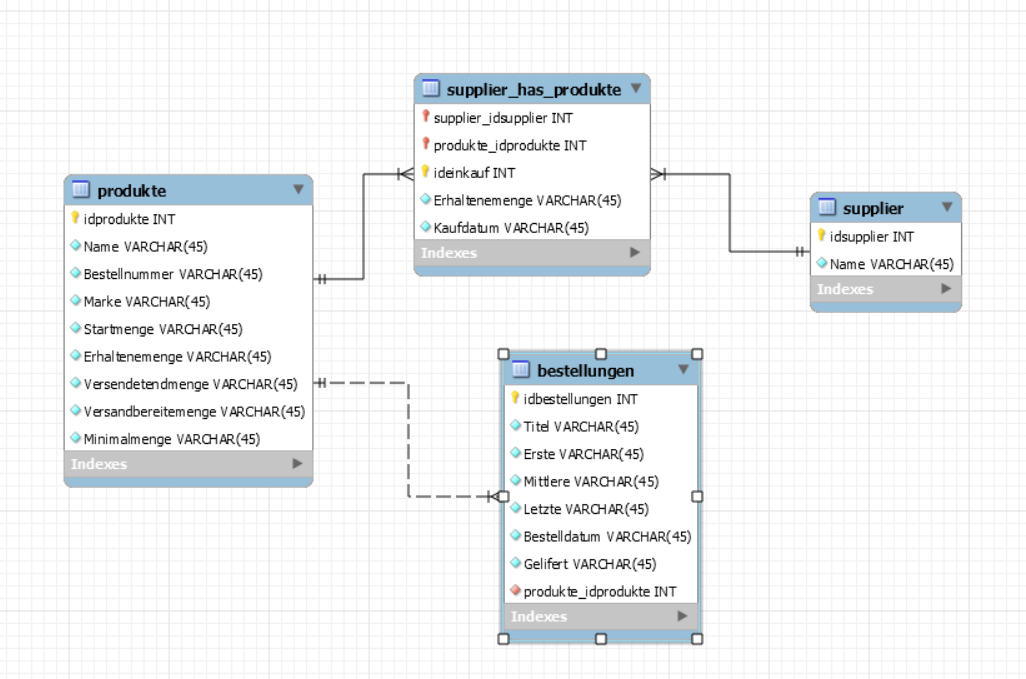


Abbildung : Inventar

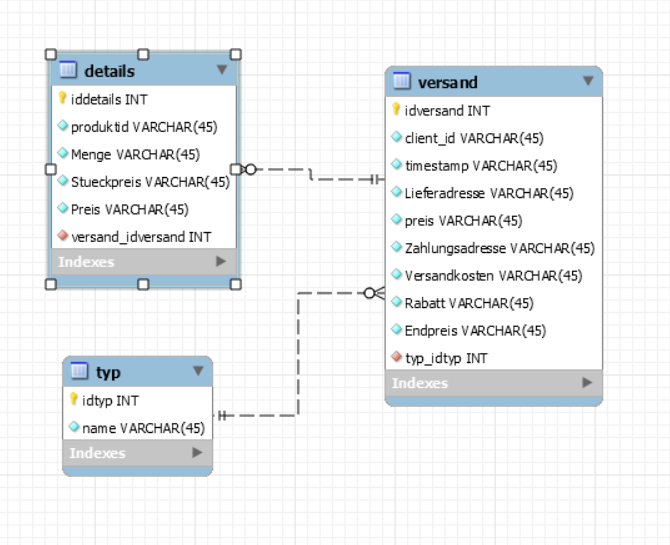


Abbildung : Versand

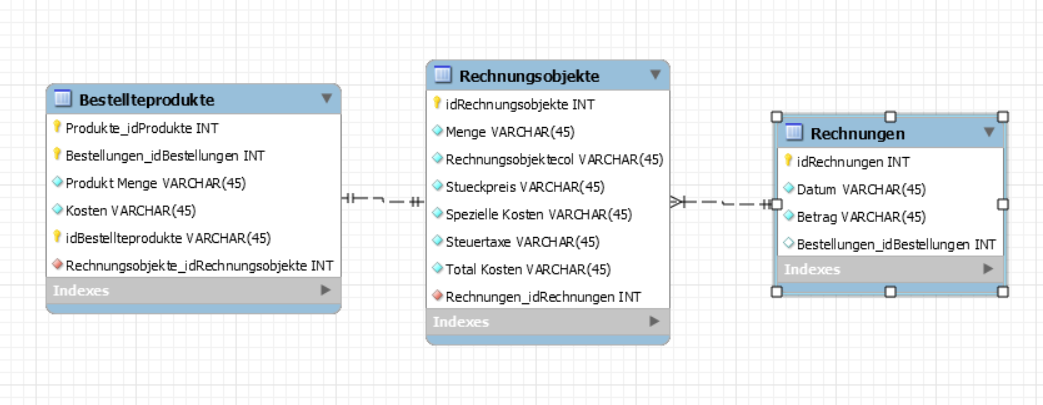


Abbildung : Zahlung

# Fazit

Das Projekt ist fehlgeschlagen. Es fehlte uns definitiv an Gruppenmitglieder mit Programmierfähigkeiten oder Erfahrungen in der IT. Die Anzahl an vielen verschiedenen Systemen wie auch Frameworks hat uns überfördert.