

# **Hochschule Osnabrück**

University of Applied Sciences

## **Fakultät**

## **Ingenieurwissenschaften und Informatik**

Schriftliche Ausarbeitung zum Thema:

### **Cold Stone Ice Creator**

im Rahmen des Moduls

Software-Architektur – Konzepte und Anwendungen,  
des Studiengangs Informatik-Technische Informatik

Autor:	Stefan Bierenriede	Marvin Gels
Matr.-Nr.:	852142	868603
E-Mail:	stefan.bierenriede@hs-osnabrueck.de	marvin.gels.1@hs-osnabrueck.de
Themensteller:	Prof. Dr. Rainer Roosmann	

Abgabedatum: 07.08.2022

# Inhaltsverzeichnis

Inhaltsverzeichnis .....	2
Abbildungsverzeichnis .....	3
Tabellenverzeichnis .....	4
Abkürzungsverzeichnis .....	5
1 Einleitung .....	1
1.1 Vorstellung des Themas .....	1
1.2 Ziel der Ausarbeitung .....	1
1.3 Aufbau der Hausarbeit .....	2
2 Darstellung der Grundlagen .....	3
2.1 Maven .....	3
2.2 Quarkus .....	3
2.3 Docker .....	3
2.4 REST-API .....	4
2.5 JSON und JSON-B .....	4
2.6 JAX-RS .....	5
2.7 Panache .....	5
2.8 REST-Assured .....	5
2.9 JTA .....	6
2.10 OpenAPI und Swagger UI .....	6
2.11 Qute Templating Engine .....	6
3 Anwendung .....	7
3.1 Datenbankstruktur .....	7
3.2 Klassenstruktur .....	8
3.3 Use-Case Diagramm .....	9
3.4 Aktivitäts-Diagramm .....	10
3.5 Routen .....	11
3.5.1 REST .....	11
3.6 Webseite .....	16
3.6.1 NavBar .....	16
3.6.2 AllergenBar .....	17
3.6.3 Startseite .....	17
3.6.4 Hauskreationen .....	17
3.6.5 Eigenkreationen .....	20
3.6.6 Bestellungen .....	22
3.6.7 Produkte .....	25
3.6.8 Nutzer .....	26
4 Zusammenfassung und Fazit .....	29
4.1 Bewertung der Ergebnisse .....	29
4.2 Fazit .....	29
4.3 Ausblick .....	29
5 Referenzen .....	31

## Abbildungsverzeichnis

Abbildung 1: Datenbankstruktur als ERD .....	7
Abbildung 2: Klassendiagramm .....	8
Abbildung 3: Use-Case Diagramm.....	9
Abbildung 4: Aktivitäts-Diagramm zum Erstellen einer Eigenkreation .....	10
Abbildung 5: NavBar .....	16
Abbildung 6: AllergenBar .....	17
Abbildung 7: Startseite der Anwendung .....	17
Abbildung 8: Ansicht auf die Hauskreationen Seite als Nutzer oder Nutzerin .....	18
Abbildung 9 Ein Knopf mit der Aufschrift „Neue Hauskreation“ .....	18
Abbildung 10: Die Seite zum Erstellen einer neuen Hauskreation.....	19
Abbildung 11: Das ausgeklappte Menü zum Ändern eines Hausbechers.....	19
Abbildung 12: Liste der Eigenkreationen.....	20
Abbildung 13: Knopf zum Erstellen einer Eigenkreation.....	20
Abbildung 14: Neue Eigenkreation erstellen .....	21
Abbildung 15: Ansicht auf eine einzelne Eigenkreation.....	21
Abbildung 16: Menü zum Ändern einer Eigenkreation.....	22
Abbildung 17: Die Liste aller Bestellungen eines eingeloggten Nutzers .....	22
Abbildung 18: Eine einzelne Bestellung eines eingeloggten Nutzers .....	23
Abbildung 19: Liste aller Bestellungen nach dem abschicken der offenen Bestellung. 23	
Abbildung 20: Detailansicht einer Bestellung, nachdem diese abgeschickt wurde .....	24
Abbildung 21: Liste aller Bestellungen, bei der eine zweite automatisch erstellt wurde24	
Abbildung 22: Ausschnitt aus der Liste aller Zutaten.....	25
Abbildung 23: Kontextmenü zum Ändern einer Zutat .....	26
Abbildung 24: Nutzer Ansicht.....	27
Abbildung 25: Einstellungen Knopf .....	27
Abbildung 26: Zugangsdaten ändern .....	27
Abbildung 27: Nutzer Seite als Administrator .....	28

## Tabellenverzeichnis

Tabelle 1: REST Nutzer erstellen - Unauthentisiert.....	11
Tabelle 2: REST Bestellungen - Nutzer.....	11
Tabelle 3: REST Bestellungen ID - Nutzer .....	12
Tabelle 4: REST Eigenkreation Bestellposten - Nutzer .....	12
Tabelle 5: REST Hauskreation Bestellposten - Nutzer .....	12
Tabelle 6: REST Eigenkreationen – Nutzer.....	12
Tabelle 7: REST Eigenkreationen ID - Nutzer .....	12
Tabelle 8: REST Eigenkreation Zutat - Nutzer .....	13
Tabelle 9: REST Hauskreationen - Nutzer .....	13
Tabelle 10: REST Hauskreationen ID - Nutzer .....	13
Tabelle 11: REST Eis – Nutzer .....	13
Tabelle 12: REST Eis ID – Nutzer .....	13
Tabelle 13: REST Nutzer ID - Nutzer .....	14
Tabelle 14: REST Saucen - Nutzer .....	14
Tabelle 15: REST Saucen ID - Nutzer .....	14
Tabelle 16: REST Zutaten – Nutzer .....	14
Tabelle 17: REST Zutaten ID - Nutzer .....	14
Tabelle 18: REST Hauskreationen - Administrator.....	15
Tabelle 19: REST Hauskreationen ID - Administrator .....	15
Tabelle 20: REST Eis - Administrator.....	15
Tabelle 21: REST Eis ID – Administrator .....	15
Tabelle 22: REST Nutzer - Administrator .....	15
Tabelle 23: REST Saucen - Administrator.....	16
Tabelle 24: REST Zutaten - Administrator.....	16
Tabelle 25: REST Zutaten ID – Administrator .....	16

## Abkürzungsverzeichnis

CDI	Context and Dependency Injection for the Java EE Plattform
ECB	Entity-Controller-Boundary Pattern
SOAP	Simple Object Access Protocol
SWA	Software-Architektur
ERD	Entity Relationship Diagram
DTO	Data Transfer Object
DAO	Data Access Object
NavBar	Navigations Bar auf der Website der Webanwendung
RBAC	Role Based Access Control
Komponenten/ Produkte	Oberbegriff für Eissorten, Zutaten und Saucen

# 1 Einleitung

In der folgenden Einleitung wird zuerst der Problembereich vorgestellt und wie sich daraus das entstandene Projekt ableitet. Am Ende der Einleitung wird ein genereller Ausblick über den Aufbau der Arbeit entwickelt.

## 1.1 Vorstellung des Themas

In Deutschland gibt es seit einigen Jahren sogenanntes Cold Stone Ice. Dies ist ein Trend, der in Amerika entstanden ist. Das Konzept beruht darauf, dass Kunden und Kundinnen sich ihr Eis nach ihren eigenen Ideen zusammenstellen können. Dabei können Kunden und Kundinnen aus Hauskreationen wählen oder sich selbst ihr Eis zusammenstellen. Ein solches Eis, egal ob Haus- oder Eigenkreation, besteht aus ein bis zwei Eissorten, beliebig vielen Zutaten und bei Wunsch einer Sauce. Dabei sind die Eissorten, Zutaten und Saucen alle Bestandteil eines festen Sortiments des Ladens. Dieses Sortiment kann aber auf Grund von geringer Nutzung einzelner Produkte verringert werden. Außerdem kann durch das Einführen neuer Produkte das Sortiment erweitert werden, entweder als feste Zutat oder nur für einen begrenzten Zeitraum wie einem Monat.

In die gewählten Eissorten werden dann die gewünschten Zutaten gemischt, sodass Kunden und Kundinnen ihre Zutaten in ihrem Eis verteilt wieder finden. Zum Schluss wird die optionale Sauce auf das Eis gegeben.

Es gibt eine sehr hohe Anzahl an Eigenkreationen, die sich Kunden und Kundinnen überlegen können, wodurch der Auswahlprozess viel Zeit in Anspruch nehmen kann. Außerdem vergessen Kunden und Kundinnen zwischen den Bestellungen eventuell ihre letzte Kreation, die ihnen zuvor sehr gut geschmeckt hatte, oder die sie vielleicht noch um eine oder mehr Zutaten erweitern wollen.

Des Weiteren kann es bei großen Bestellungen mit mehreren Haus- und/oder Eigenkreationen dazu kommen, dass die Kunden und Kundinnen nicht mehr alles richtig an die Verkäufer und Verkäuferinnen weitergeben können, weil sie eventuell eine Kreation verwechselt haben oder eine Zutat oder eines der anderen Produkte vertauscht haben.

[Gels]

## 1.2 Ziel der Ausarbeitung

Um diesen Problemen und Interessen entgegenzuwirken, wurde eine Web-basierte Anwendung unter Verwendung von Quarkus erstellt. Diese umfasst sowohl eine API-Schnittstelle, sowie eine Benutzeroberfläche.

Die Anwendung ermöglicht es den Kunden und Kundinnen sich auf der Seite alle Eissorten, Zutaten und Saucen anzusehen und aus diesen eine Eigenkreation zu erstellen. Anschließend können diese Eigenkreationen, sowie auch Hauskreationen, in die aktuelle Bestellung des Nutzens aufgenommen werden. Aus dieser Bestellung können bestimmte Posten, also eine Menge an Kreationen, wieder entfernt werden. Außerdem kann eine Bestellung abgeschickt werden und somit auch eine neue Bestellung zu beginnen. Eine solche Bestellung kann somit aus mehreren Eigen- und/oder Hauskreationen bestehen. Kunden und Kundinnen können außerdem auf ihr Profil zugreifen und dort ihre Login Daten abändern. Vorherige selbst erstellte Eigenkreationen und Hauskreationen allgemein, kann sich jeder Kunde oder jede Kundin anschauen.

Ein Administrator kann außerdem noch neue Eissorten, Zutaten und Saucen einfügen und bestehende Komponenten verändern, sowie Hauskreationen erstellen oder abändern. Der Administrator kann die Eigenkreationen und Bestellungen aller Kunden und Kundinnen einsehen. All diese Funktionen werden im Folgenden weitergehend beschrieben und erklärt. Außerdem wird auf den Aufbau, die genutzten Pattern und Techniken eingegangen.

[Gels]

### **1.3 Aufbau der Hausarbeit**

Der Aufbau der Hausarbeit gliedert sich in verschiedene Kapitel, die hier im Folgenden kurz erklärt werden sollen.

Nach der Einleitung in Kapitel 1, in der das Thema erklärt und ein Ziel gesetzt wurde, folgt in Kapitel 2 eine Darstellung der Grundlagen. Hier werden die genutzten Techniken erklärt und unter anderem Abwägungen erläutert, warum spezifische Techniken anderen vorgezogen wurden.

In Kapitel 3 wird zuerst mit Hilfe von Diagrammen die Struktur und die Funktionsweise des Projekts erläutert. Es werden die Datenbank- und die Klassenstruktur beschrieben, außerdem werden die hier genutzten Pattern erläutert. Des Weiteren werden mit Hilfe eines Use-Case Diagramms die Zugriffsrechte auf die Schnittstelle dargestellt. Das letzte Diagramm ist ein Aktivitätsdiagramm, das ausschnitthaft einen Prozess der Anwendung darstellt. Außerdem behandelt das dritte Kapitel die Routen, mit denen auf die Schnittstellen der erzeugten REST-API zugegriffen werden kann. Im vierten und letzten Teil der Hausarbeit wird die Struktur und die Funktionalität der Website mit Hilfe von Bildschirmausschnitten gezeigt. Hier werden die Seiten und ihre Subseiten detailliert erklärt, ohne auf die technischen Hintergründe tiefergehend einzugehen.

Im letzten Kapitel werden zuerst die Ergebnisse daraufhin bewertet, wie gut sie den gegebenen Projektvorschlag bearbeiten. Mögliche Probleme und Veränderungen, die sich während der Projektbearbeitungsphase ergeben haben, werden behandelt. Anschließend wird ein Fazit aus dem Projekt gezogen. Am Ende der Hausarbeit wird ein Ausblick gestellt, welche möglichen Erweiterungen erstellt werden können, sowohl technisch als auch inhaltlich.

[Gels]

## 2 Darstellung der Grundlagen

In diesem Kapitel werden die technischen Grundlagen erklärt, auf denen das Projekt aufbaut. Dazu werden alle verwendeten Technologien auf verständliche Art und Weise dargestellt.

### 2.1 Maven

Maven ist das Build- und Anwendungsmanagementsystem, das für dieses Projekt genutzt wird. Es übernimmt mit Hilfe des Project Object Model, auch pom.xml, die Abhängigkeitsverwaltung, so dass Dependencies automatisch im Build-Prozess geladen werden. Außerdem updatet, installiert und validiert Maven die Kompatibilität. So müssen Entwickler und Entwicklerinnen sich nicht mehr mit den Abhängigkeiten beschäftigen, was wiederum zu weniger Fehlern führt. Maven durchläuft einen sogenannten Lifecycle, der vom Entwickler oder von der Entwicklerin bestimmt werden kann. Dieser Lifecycle besteht in der Regel unter anderem aus einer Überprüfung des Programms auf Korrektheit, dem Kompilieren und den definierten Tests. Danach wird das Projekt verpackt, weitere Tests werden ausgeführt und es wird überprüft, ob das Paket richtig gepackt wurde. Daraufgehend wird das Paket in dem Maven Repository installiert und von dort aus ausgeführt. Dieser Prozess nimmt dem Entwickler viel Arbeit ab und senkt somit auch die Wahrscheinlichkeit auf Fehler.

(vgl. [ @MVN ]) [Gels]

### 2.2 Quarkus

Als Framework für dieses Projekt wurde Quarkus gewählt. Quarkus ist ein Kubernetes-natives Java Framework, das aus verschiedenen Java Libraries und Standards besteht. (vgl. [ @Qrk ])

Quarkus liefert eine effektive Möglichkeit für verschiedene webbasierte Umgebungen. Das Framework wurde explizit für verschiedene Standards und auch andere Frameworks entwickelt. Durch die Dependency-Injection-Lösung von Quarkus, welche auf Contexts und Dependency Injection (auch CDI) beruhen, können Erweiterungen einfach in das Projekt aufgenommen werden. Durch die Container-first Herangehensweise von Quarkus werden unter anderem der Speicherverbrauch und die Zeit zum Hochfahren des Systems minimiert. (vgl. [ RED ]) [Gels]

### 2.3 Docker

Für das Projekt wurde Docker genutzt, mit dem ein einfaches und schnelles builden der Software möglich ist. Docker stellt sogenannte Container zur Verfügung, auf denen Software isoliert zur lokalen Infrastruktur ausgeführt werden kann. Mit diesem System ist es möglich die Software auf verschiedenen Betriebssystemen lauffähig zu starten. Daher bietet Docker einen gravierenden Vorteil gegenüber Virtuellen Maschinen, da es viel ressourcenschonender und kompakter arbeitet.

Docker erstellt dazu in den Container eine eigene Umgebung, mit der sichergestellt wird, dass sich die Anwendungen nicht gegenseitig stören. In diesen Containern speichert Docker den Code der Anwendung und alle seine Abhängigkeiten. Docker erstellt sogenannte Images, die einen Schnappschuss eines Containers darstellt. Wird dieses Image auf einem neuen System ausgeführt, erstellt Docker automatisch einen lauffähigen Container. So gesehen sind Image Klassen und die erstellten Container sind die Instanzen der Klasse.



In dieser Hausarbeit wurde Docker zum einen für die Datenbank genutzt. Dafür wird beim Ausführen der Anwendung parallel eine PostgreSQL Datenbank gestartet, auf der die Daten der Anwendung persistiert werden können. Zum anderen wird zum Testen der Anwendung ein Testcontainer gestartet, auf dem alle erstellten JUnit Tests abgesichert zum System durchlaufen werden, ohne die Anwendung zur Laufzeit zu stören. So können die Tests laufend angepasst und ausgeführt werden.

(vgl. [DKR]) [Bierenriede]

## 2.4 REST-API

Eine API ist ein Application Programming Interface und besteht aus verschiedenen Protokollen zur Entwicklung und Integration von Anwendungssoftware, sowie mehreren Werkzeugen und Definitionen. Dabei definiert dieses Interface, wie andere auch, klare Eingabeparameter und liefert bei richtiger Eingabe eine Antwort oder führt entsprechend der Eingabe eine Aktion auf dem System aus. Dabei kann die API auch klar definieren, welche Nutzer auf welche Dienste Zugriff haben und auf welche nicht.

Genauer eingegrenzt, was die API zulassen soll, wird durch die REST. Dies ist eine Sammlung von Beschränkungen der Architektur und kein Protokoll oder ein Standard, wodurch es Entwicklern ermöglicht wird, die REST auf verschiedenste Arten und Weisen zu implementieren. So wird meist der Ressourcenstatus an Clients in Form eines HTTP-Formats ausgegeben. Dabei wird in diesem Projekt mit JSON gearbeitet.

REST-APIs sind im Vergleich zu anderen Standards schneller, weil sie nicht vorgeschriebenen Protokollen wie SOAP entsprechen müssen, bei denen Sicherheitsaspekte und andere Dinge vorgeschrieben sind.

(vgl. [RRA]) [Gels]

## 2.5 JSON und JSON-B

JavaScript Object Notation oder kurz JSON ist ein Format für den Datenaustausch, welches in diesem Projekt im Rahmen der REST-API genutzt wird. Es ist für Menschen gut zu lesen und kann von Maschinen gut geparkt und erzeugt werden. JSON ist ein von anderen Programmiersprachen losgelöstes Textformat, es befolgt jedoch ähnliche Konventionen wie zum Beispiel C, Java und JavaScript, wodurch es Entwicklern, die diese Sprachen bereits kennen, einfach fällt sich in JSON einzufinden.

(vgl. [JSN])

JSON-B wird in diesem Projekt benutzt, um Java Objekte auf und von JSON zu mappen. Vor JSON-B mussten Entwickler dieses Mapping meist selbst übernehmen oder auf nicht standardisierte Lösungen zurückgreifen. Damit JSON-B ein Objekt serialisieren oder deserialisieren kann, müssen die Variablen sichtbar sein, entweder dadurch, dass sie public gesetzt werden oder mit Hilfe von Getter- und Settermethoden.

Dafür wurden in diesem Projekt DTOs eingeführt, um eine sichere Schnittstelle für Objekte zu gewährleisten.

(vgl. [JSB]) [Gels]

## 2.6 JAX-RS

Für das Mapping zwischen HTTP Verben und Java Methoden wird JAX-RS genutzt. JAX-RS ist ein Werkzeug, dass Entwicklern beim Erstellen einer Web-Applikation nach dem REST Pattern hilft. Dabei können Annotationen benutzt werden um damit Java Objekte auf die Web-Ressourcen zu mappen. Dies sind unter anderem Annotationen wie POST, GET, PUT, PATCH und DELETE, wodurch der Entwickler festlegen kann, auf welche Art von HTTP Verb die Java Methode gemappt werden soll. Des Weiteren gibt es Annotationen wie @Path mit der der relative URI Pfad einer Klasse oder Methode deklariert werden kann. Annotationen wie @Produces und @Consumes legen fest welchen Typ von Datenobjekt eine Klasse oder eine Methode akzeptiert oder ausgibt. Andere Annotationen bieten Entwicklern noch weitere Möglichkeiten ihre Methoden und Klassen anzupassen.

(vgl. [@JXB], [@JXR], [@JXO]) [Gels]

## 2.7 Panache

Zur Persistierung der Daten wurde Hibernate Panache eingesetzt, dass das Active Record Pattern umsetzt. Mit dem Active Record Pattern werden die Klassen als Datenbanktabellen abgebildet, um so einen einfachen Zugriff auf sie zu gewährleisten. Dazu werden Annotationen verwendet, die auf Klassen und Variablen verwendet werden können. Konkret werden die Klassen, die persistiert werden sollen mit der Annotation @Entity annotiert. Panache legt dann automatisch für die Objektvariablen die passenden Spalten in der Tabelle der Datenbank an. Mit Hilfe von @Password kann eine Objektvariable annotiert werden, um so die eingegeben Daten verschlüsselt in der Datenbank abzulegen. Panache bietet bereits viele verschiedene Zugriffsmöglichkeiten, um auf die Daten in der Datenbank zu zugreifen.

Als Datenbanktyp wurde eine PostgreSQL Datenbank verwendet auf die mithilfe von Hibernate ORM zugegriffen wird. Die Verwendung von ORM Hibernate ermöglicht einen Zugriff auf viele verschiedene Datenbanktypen, die einfach ausgetauscht werden können.

(vgl. [@QHP]) [Bierenriede]

## 2.8 REST-Assured

Um eine korrekte Funktionsweise des Systems zu gewährleisten, wurde die REST-API mittels JUnit Tests getestet. Dazu werden Test-Klassen angelegt die mit @QuarkusTest annotiert sind, in den Klassen wird mittels der Java Bibliothek REST-Assured getestet. Die Test-Methoden innerhalb der Klasse, werden mit @Test annotiert. REST-Assured bietet viele Funktionen, die zum Testen von REST-APIs verwendet werden können. Dazu werden http-Request gesendet die dann ausgewertet werden können. Dafür unterstützt REST-Assured HTTP-Verben wie GET, POST, PUT und DELETE. Zusätzlich wurden Hamcrest Matcher verwendet, um die empfangenen JSON Objekte auszuwerten.

Die Test-Cases bestehen grundlegend aus demselben Aufbau der wie folgt aussieht, given(), when() und then(). Dabei stellt given() die Ausgangssituation bereit, when() die zu testende Route und die resultierende Änderung. Anschließend wird mit then() das erwartete Ergebnis überprüft, wozu die Methode statusCode() verwendet wird.

(vgl. [@BQT]) [Bierenriede]

## 2.9 JTA

Zur Sicherung von Transaktionen wird in diesem Projekt mit JTA gearbeitet. JTA ist kurz für Java Transaction API und bildet ein Interface zwischen dem Transaktions Manager und den Systemen die Transaktionen durchführen wollen. So kann durch eine `@Transactional` Annotation eine Methode für eine Transaktion gekennzeichnet werden, wodurch beim Aufruf dieser Methode ein gesicherter Zugriff auf die Datenbank möglich ist.

(vgl. [`@JTA`]) [Gels]

## 2.10 OpenAPI und Swagger UI

OpenAPI wird verwendet, um die REST-API zu beschreiben, dazu werden die REST Klassen annotiert. Durch die Annotationen werden die Endpunkte der REST beschrieben, um die Benutzung für andere Entwickler zu erleichtern. Die Beschreibung der REST-API kann mittels `/q/openapi` generiert werden, wodurch eine Übersicht über alle REST Schnittstellen dargestellt wird.

Zum manuellen Testen der REST-API wird Swagger UI verwendet, darüber wird eine Web UI bereitgestellt mit der die einzelnen REST Endpunkte getestet werden. Das UI wird mithilfe der OpenAPI Spezifikationen generiert. Swagger UI ist unter `/q/swagger-ui` erreichbar, dafür muss die Anwendung im Entwicklermodus gestartet werden. Dort können die verschiedenen Funktionen der API getestet werden. Es wird dargestellt welche HTTP-Verben für eine Route bereitstehen und was die Form des erwarteten Schemas ist.

(vgl. [`@QOS`]) [Bierenriede]

## 2.11 Qute Templating Engine

Mit Qute soll das Erstellen eines Web Frontends mit Quarkus erleichtert werden, denn Qute ist speziell auf Quarkus angepasst worden. Qute soll den Programmierfluss fördern, in dem es Änderungen während der Laufzeit übernimmt und anzeigt. Dazu muss Quarkus im Entwicklermodus ausgeführt werden, wird nun eine Änderung an einer Datei im Pfad `src/main/resources/templates` vorgenommen, übernimmt Qute diese dynamisch und bildet sie direkt im Frontend ab. Außerdem werden viele Fehler zur compilezeit erkannt und angezeigt.

Für Qute wurde in den `boundries` ein extra Ordner `web` angelegt, in dem die vorhandenen REST Klassen übernommen und angepasst werden. Da HTML nur die HTTP-Verben GET und POST akzeptiert müssen die REST-API Methoden und Routen angepasst werden. Im `templates`-Pfad werden die HTML Seiten abgelegt auf die Qute zugreifen soll. In den HTML Seiten werden sogenannte „Template extension methods“ benutzt, um Zugriff auf Objekte aus der REST-API zu bekommen. So können in den HTML Seiten eigene Methoden erstellt werden, um verschiedene Daten einfach und flexibel anzeigen zu lassen.

(vgl. [`@QTE`]) [Bierenriede]

### 3 Anwendung

In diesem Kapitel wird die Struktur des Projekts anhand von verschiedenen Darstellungen gezeigt und erklärt. Angefangen mit einem ERD über ein vereinfachtes Klassendiagramm, einem Use-Case Diagramm und zuletzt einem Aktivitätsdiagramm.

#### 3.1 Datenbankstruktur

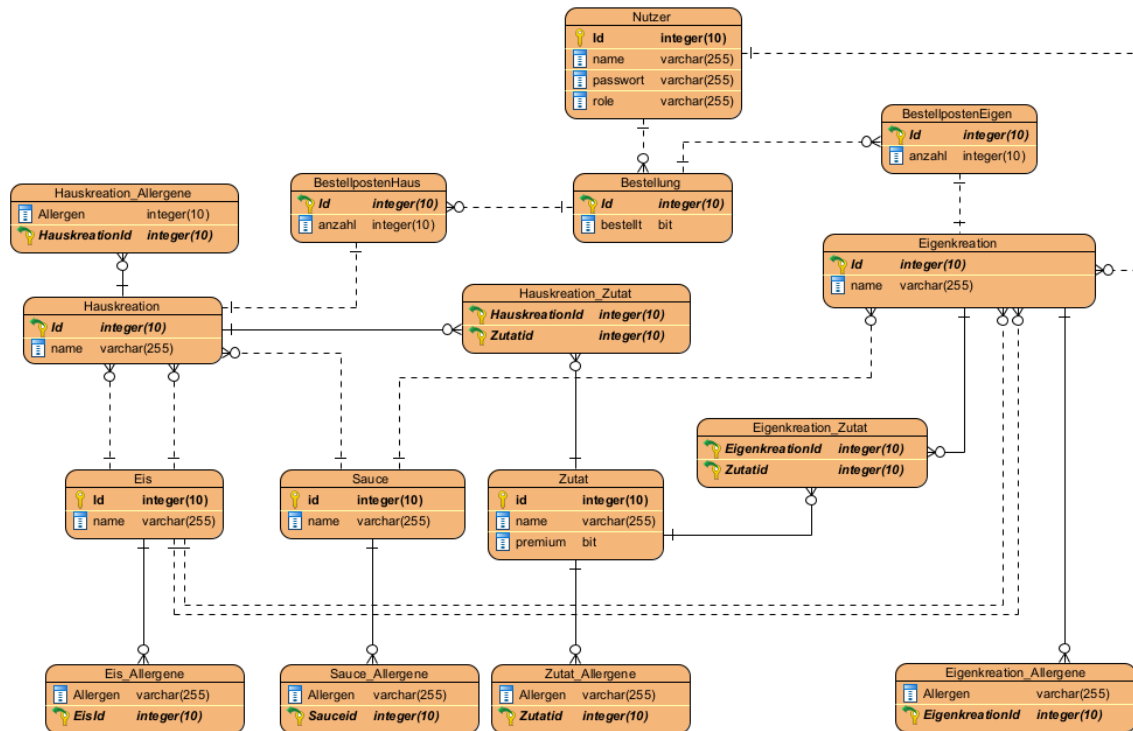


Abbildung 1: Datenbankstruktur als ERD

In Abbildung 1 ist die in diesem Projekt verwendete Datenbankstruktur abgebildet. Die Grundstruktur sind die verschiedenen Zutaten, Saucen und Eissorten, die verwendet werden können. Aus der Kombination dieser Komponenten lassen sich dann die Hauskreationen und Eigenkreationen erstellen. All diese Entitäten haben zusätzlich eine schwache Entität Allergene. In dieser werden die verschiedenen Allergene gespeichert, die die jeweiligen Komponenten haben. In den Hauskreationen, bzw. den Eigenkreationen, werden die Allergene aller Komponenten dann gebündelt gespeichert, damit auf einem Blick ersichtlich wird, welche Allergene in den jeweiligen Kreationen enthalten sind. Eine Bestellung besteht aus einer Menge von Hauskreationen und Eigenkreationen, diese werden mit der Anzahl in den jeweiligen Bestellposten in der Bestellung gespeichert. Jede erstellte Bestellung wird von genau einem Nutzer gehalten. Neben den Bestellungen hat der Nutzer noch eine Liste seiner selbst erstellten Eigenkreationen. So kann jeder Nutzer auf alle Hauskreationen und seine Eigenkreationen zugreifen.

[Bierenriede]

### 3.2 Klassenstruktur



Abbildung 2: Klassendiagramm

In Abbildung 2 ist ein vereinfachtes Klassendiagramm des Projekts dargestellt. Für die Übersichtlichkeit wurden auf die Attribute, Funktionen und Verbindungen verzichtet. Vielmehr soll mit diesem Diagramm der generelle Aufbau beschrieben werden.

Das Projekt wurde in vier grundlegende Pakete aufgeteilt: bestellung, kreation, nutzer und produkt. Dies folgt der Fachlichen Schichtung, in welcher das Projekt nach den inhaltlichen Gesichtspunkten aufgeteilt wird. Dabei kümmern sich die Pakete um ihre eigenen Bereiche. In bestellung werden Bestellungen und Bestellposten erzeugt, verwaltet und gelöscht. In kreation findet die Eigen- und Hauskreationsverwaltung statt. Die Nutzer und deren Bearbeitung passiert in dem Paket nutzer und in dem Paket produkt findet die Verwaltung der Komponenten Eis, Saucen und Zutaten statt.

Weitergehend werden diese vier Pakete dann nach dem ECB-Modell in jeweils weitere vier Pakete aufgeteilt: boundary, control, entity und gateway. Hierbei handelt es sich um die Technische Schichtung. Die boundary Pakete kümmern sich um die direkte Schnittstelle mit Systemen und Kommunikation außerhalb des Projekts, durch das Paket rest (und durch das web Paket, welches hier der Übersicht halber ausgelassen wurde). Außerdem enthält dieses Paket noch ein weiteres Paket dto, welches Entitäten nach außen hin kapselt. In dem Paket kreation gibt es zudem noch das dao Paket, welches benutzt wird, um zwischen den vier inhaltlichen Paketen zu kommunizieren.

Die control Pakete bieten Interfaces, die die Kommunikation zwischen den boundary und gateway Paketen klar definieren und somit sichere Zugriffe auf die Datenbank, oder andere Infrastruktur die sich in den gateway Paketen befinden, vereinfachen.

In den entity Paketen finden sich die Entitäten, also die Objekte, die die Daten halten und die auch mit Hilfe von Panache in der Datenbank gespeichert werden.

Das Paket gateway enthält alle Funktionen und Zugriffe, die mit der allgemeinen Infrastruktur des Projekts zu tun haben. In diesem Fall sind es Zugriffe auf eine Datenbank mit Hilfe von ORM und Panache.

Eine Verbindung der Technischen und der Fachlichen Schichtung sorgt für eine übersichtlichere Trennung und für eine Trennung, die auch für eine losere Kupplung sorgen kann, wenn diese richtig umgesetzt wird.

[Gels]

### 3.3 Use-Case Diagramm

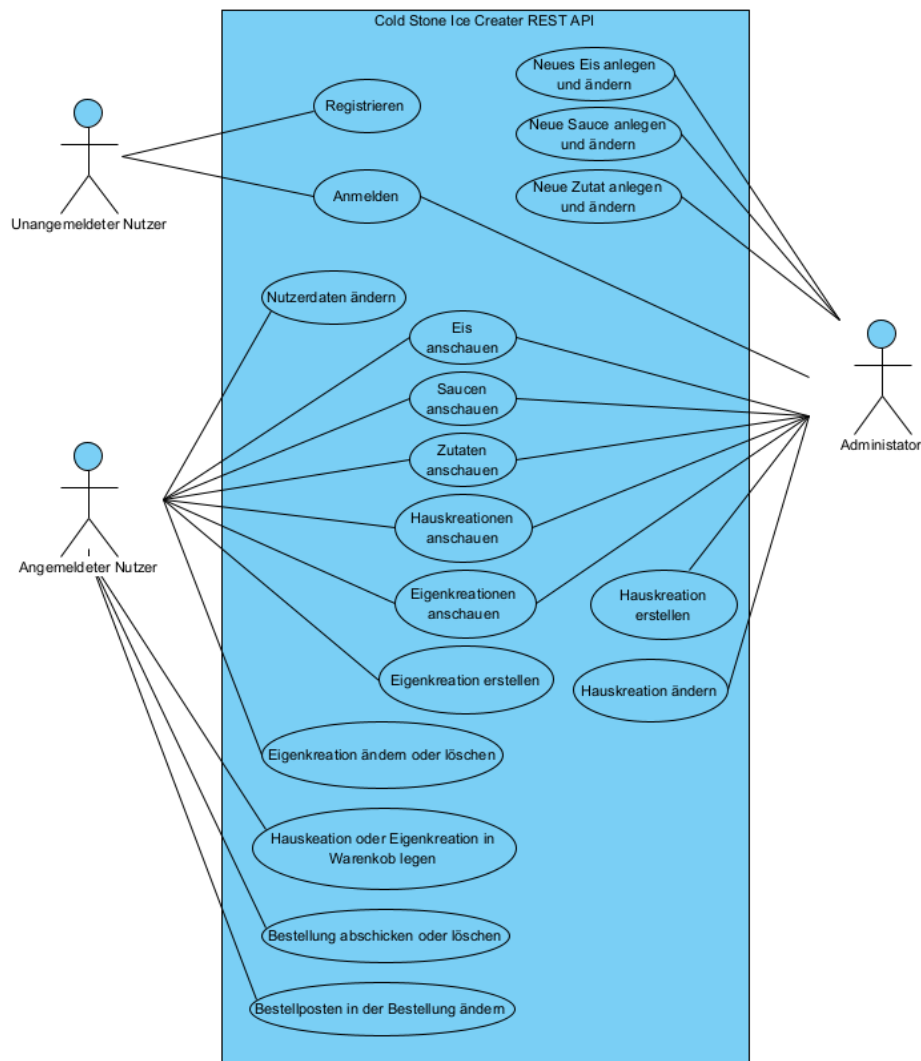


Abbildung 3: Use-Case Diagramm

Abbildung 3 zeigt das Use-Case Diagramm des Systems. Für das System gibt es im Wesentlichen drei verschiedene Akteure: den Unangemeldeten Nutzer, den Angemeldeten Nutzer und den Administrator.

Als unangemeldeter Nutzer oder Nutzerin gibt es nicht viele Anwendungsfälle, dieser kann sich lediglich auf der Seite registrieren und so einen neuen Nutzer anlegen oder, falls schon ein Nutzer hinterlegt wurde, sich mit diesem anmelden.

Nach dem Anmelden stehen dem angemeldeten Nutzer oder Nutzerin viele verschiedene Funktionen zur Verfügung. Unter anderem können sich angemeldete Nutzer und Nutzerinnen alle Eissorten, Saucen und Zutaten anschauen. Außerdem kann sich der angemeldete Nutzer oder die Nutzerin alle vorhandenen Hauskreationen sowie seine selbst erstellten Eigenkreationen anschauen. Erstellte Eigenkreationen können nur von den jeweiligen Nutzern und Nutzerinnen selbst angeschaut und verwaltet werden. Wenn ein Nutzer oder Nutzerin eine neue Eigenkreation erstellt, wird diese auch automatisch der aktuellen Bestellung des Nutzers oder Nutzerin hinzugefügt. Wenn ein Nutzer oder Nutzerin fertig mit seiner/ihrer Bestellung ist, kann

diese abgeschickt werden, nach dem Abschicken können keine Änderungen mehr an der Bestellung vorgenommen werden.

Ein Administrator hat im Grunde dieselben Zugriffsrechte wie der angemeldete Nutzer, nur kann dieser zusätzlich noch Komponenten oder Hauskreationen erstellen oder verändern. Außerdem kann er die erstellten Eigenkreationen zwar einsehen, diese aber nicht verändern. Des Weiteren kann er sich alle Bestellungen anschauen sowie alle registrierten Nutzer einsehen.

[Bierenriede]

### 3.4 Aktivitäts-Diagramm

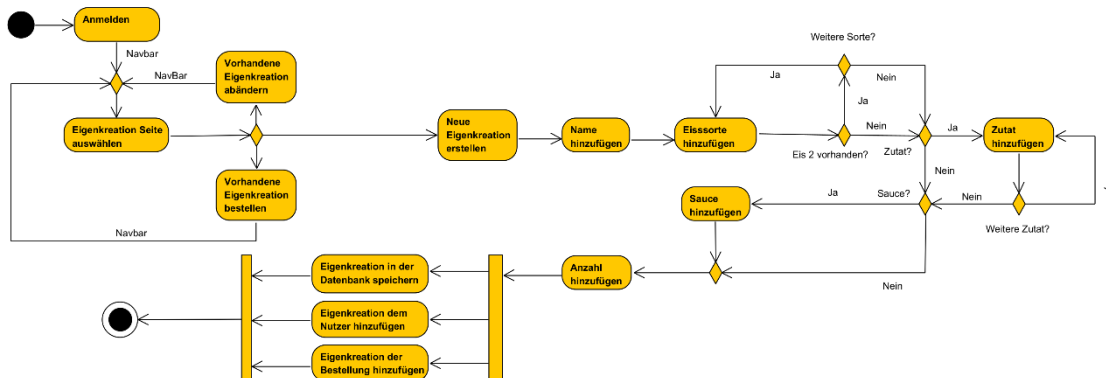


Abbildung 4: Aktivitäts-Diagramm zum Erstellen einer Eigenkreation

In Abbildung 4 ist das Aktivitäts-Diagramm für einen Ausschnitt des Projekts zu sehen. Beispielsweise am Erstellen einer Eigenkreation wird hier die Nutzung des Programms dargestellt.

Nachdem sich der Nutzer oder die Nutzerin angemeldet hat, kann diese/r über die Navbar auf die Seite wechseln, auf der alle aktuellen Eigenkreationen des eingeloggten Nutzers angezeigt werden. Hier kann gewählt werden, ob eine bereits vorhandene Eigenkreation abgeändert werden soll, ob eine bereits vorhandene Eigenkreation bestellt werden soll oder ob eine neue Eigenkreation erstellt werden soll. Bei dem Abändern und dem Bestellen einer bereits vorhandenen Eigenkreation wird der Nutzer oder die Nutzerin auf jeweils andere Seiten weitergeleitet, die in diesem Aktivitäts-Diagramm nicht weiter behandelt werden sollen. Beim Erstellen einer neuen Eigenkreation kommt der Nutzer oder die Nutzerin auf eine Subseite, auf der als erstes ein Name eingetragen werden muss. Danach wird der Nutzer oder die Nutzerin aufgefordert aus der Liste aller aktuell in der Datenbank vorhandenen Eissorten eine zu wählen und es besteht die Möglichkeit eine zweite zu wählen. Sollte bei der zweiten Eissorte nichts gewählt werden, so wird die Wahl der ersten Sorte auf die zweite übertragen. Nach der Wahl der Eissorte kann der Nutzer oder die Nutzerin auswählen, ob er/sie Zutaten in der Eigenkreation haben möchte. Dabei kann der Nutzer oder die Nutzerin aus einer Liste der aktuell in der Datenbank gespeicherten Zutaten wählen und die entweder einfügen oder nicht. Bei den Zutaten gibt es keine Mindest- oder Maximalmenge wie bei den Eissorten. Als Letztes für die Erstellung der Eigenkreation besteht noch die Möglichkeit eine Sauce zu wählen. Dabei kann der Nutzer oder die Nutzerin aus der Liste aller Saucen, die aktuell in der Datenbank vorhanden sind, wählen. Die Sauce kann entweder weggelassen werden oder es wird eine gewählt, mehrere zu wählen ist nicht möglich. Bevor der Nutzer oder die Nutzerin die Eigenkreation erstellt, muss noch eine Anzahl angegeben werden. Diese Anzahl bestimmt die Menge der Eigenkreationen, die in die aktuell offene Bestellung des Nutzers oder der Nutzerin hinzugefügt werden soll. Sobald die Eigenkreation erstellt werden soll, wird diese als Eigenkreationsentität

in der Datenbank gespeichert. Außerdem wird diese Eigenkreation der Liste der Eigenkreationen des Nutzers oder der Nutzerin hinzugefügt, sodass dieser/diese in Zukunft wieder auf diese Eigenkreation zugreifen kann, um sie zu bestellen oder zu verändern. Gleichzeitig wird eine Menge entsprechend der Anzahl an neu erstellen Eigenkreationen in die aktuell offene Bestellung des Nutzers eingefügt. Damit ist die Erstellungsaktivität einer Eigenkreation beendet.

[Gels]

### 3.5 Routen

Die im Projektvorschlag angegebenen Routen wurden im Laufe des Entwicklungsprozesses angepasst, weshalb sich die im nachfolgenden Kapitel vorgestellten Routen zu den vorher definierten, teilweise unterscheiden werden. Zusätzlich wurden noch Routen für das Quarkus Qute Web-Frontend eingefügt, die im Nachfolgenden aber nicht extra beschrieben werden.

[Bierenriede]

#### 3.5.1 REST

##### Ohne Authentisierung

Folgende Route kann von allen Personen aufgerufen werden, da sie keine Authentisierung benötigt:

<b>/api/nutzer</b>	
GET	--
PUT	--
POST	Erstellt einen neuen Nutzer
DELETE	--

Tabelle 1: REST Nutzer erstellen - Unauthentisiert

##### Angemeldete Nutzerinnen und Nutzer sowie Administratoren

Diese Routen können von allen angemeldeten Nutzerinnen und Nutzern sowie der Administratoren benutzt werden:

<b>/api/bestellungen</b>	
GET	Gibt alle Bestellungen des Nutzers zurück
PUT	--
POST	Schickt die aktuelle Bestellung ab
DELETE	Löscht die aktuelle Bestellung

Tabelle 2: REST Bestellungen – Nutzer



<b>/api/bestellungen/{id}</b>	
GET	Gibt eine bestimmte Bestellung zurück
PUT	--
POST	--
DELETE	--

Tabelle 3: REST Bestellungen ID - Nutzer

<b>/api/bestellungen/{id}/eigenposten/{eigenpostenId}</b>	
GET	--
PUT	Ändert die Anzahl einer bestimmten Eigenkreation in der Bestellung
POST	--
DELETE	Löschen eines bestimmten Eigenbestellpostens aus der Bestellung

Tabelle 4: REST Eigenkreation Bestellposten - Nutzer

<b>/api/bestellungen/{id}/hausposten/{hauspostenId}</b>	
GET	--
PUT	Ändert die Anzahl einer bestimmten Hauskreation in der Bestellung
POST	--
DELETE	Löschen eines bestimmten Hausbestellpostens aus der Bestellung

Tabelle 5: REST Hauskreation Bestellposten - Nutzer

<b>/api/eigenkreationen</b>	
GET	Gibt alle Eigenkreationen des Nutzers zurück
PUT	--
POST	Erstellt eine neue Eigenkreation
DELETE	--

Tabelle 6: REST Eigenkreationen – Nutzer

<b>/api/eigenkreationen/{id}</b>	
GET	Gibt eine bestimmte Eigenkreation zurück
PUT	Ändern einer bestimmten Eigenkreation
POST	Fügt eine bestimmte Eigenkreation der Bestellung hinzu
DELETE	Löschen einer bestimmten Eigenkreation

Tabelle 7: REST Eigenkreationen ID – Nutzer

<b>/api/eigenkreationen/{id}/zutaten/{zutatnummer}</b>	
GET	--
PUT	Ändern einer Zutat in einer bestimmten Eigenkreation
POST	--
DELETE	--

Tabelle 8: REST Eigenkreation Zutat - Nutzer

<b>/api/hauskreationen</b>	
GET	Gibt alle Hauskreationen zurück
PUT	--
POST	--
DELETE	--

Tabelle 9: REST Hauskreationen - Nutzer

<b>/api/hauskreationen/{id}</b>	
GET	Gibt eine bestimmte Hauskreation zurück
PUT	--
POST	Fügt eine bestimmte Hauskreation der Bestellung hinzu
DELETE	--

Tabelle 10: REST Hauskreationen ID - Nutzer

<b>/api/eis</b>	
GET	Gibt alle Eissorten zurück
PUT	--
POST	--
DELETE	--

Tabelle 11: REST Eis – Nutzer

<b>/api/eis/{id}</b>	
GET	Gibt eine bestimmte Eissorte zurueck
PUT	--
POST	--
DELETE	--

Tabelle 12: REST Eis ID – Nutzer

<b>/api/nutzer/{id}</b>	
GET	Gibt einen bestimmten Nutzer zurück
PUT	Ändern eines bestimmten Nutzers
POST	--
DELETE	Löschen eines bestimmten Nutzers

Tabelle 13: REST Nutzer ID - Nutzer

<b>/api/saucen</b>	
GET	Gibt alle Saucen zurück
PUT	--
POST	--
DELETE	--

Tabelle 14: REST Saucen - Nutzer

<b>/api/saucen/{id}</b>	
GET	Gibt eine bestimmte Sauce zurück
PUT	--
POST	--
DELETE	--

Tabelle 15: REST Saucen ID - Nutzer

<b>/api/zutaten</b>	
GET	Gibt alle Zutaten zurück
PUT	--
POST	--
DELETE	--

Tabelle 16: REST Zutaten – Nutzer

<b>/api/zutaten/{id}</b>	
GET	Gibt eine bestimmte Zutat zurück
PUT	--
POST	--
DELETE	--

Tabelle 17: REST Zutaten ID – Nutzer

## Administratoren

Diese Routen können ausschließlich von Administratoren benutzt werden:

<b>/api/hauskreationen</b>	
GET	--
PUT	--
POST	Erstellen einer neuen Hauskreation
DELETE	--

Tabelle 18: REST Hauskreationen - Administrator

<b>/api/hauskreationen/{id}</b>	
GET	--
PUT	Ändern einer bestimmten Hauskreation
POST	--
DELETE	Löschen einer bestimmten Hauskreation

Tabelle 19: REST Hauskreationen ID - Administrator

<b>/api/eis</b>	
GET	--
PUT	--
POST	Erstellen einer neuen Eissorte
DELETE	--

Tabelle 20: REST Eis - Administrator

<b>/api/eis/{id}</b>	
GET	--
PUT	Ändern einer bestimmten Eissorte
POST	--
DELETE	Löschen einer bestimmten Eissorte

Tabelle 21: REST Eis ID – Administrator

<b>/api/nutzer</b>	
GET	Gibt alle Nutzer zurück
PUT	--
POST	--
DELETE	--

Tabelle 22: REST Nutzer – Administrator

<b>/api/saucen</b>	
GET	--
PUT	--
POST	Erstellen einer neuen Sauce
DELETE	--

Tabelle 23: REST Saucen - Administrator

<b>/api/zutaten</b>	
GET	--
PUT	--
POST	Erstellen einer neuen Zutat
DELETE	--

Tabelle 24: REST Zutaten - Administrator

<b>/api/zutaten/{id}</b>	
GET	--
PUT	Ändern einer bestimmten Zutat
POST	--
DELETE	Löschen einer bestimmten Zutat

Tabelle 25: REST Zutaten ID – Administrator

[Bierenriede]

### 3.6 Webseite

Im Folgenden wird die mit Hilfe von Quarkus Qute erstellte Webseite dargestellt.

#### 3.6.1 NavBar

Hauskreationen	Eigenkreationen	Bestellungen	Eissorten	Zutaten	Saucen	Nutzer
----------------	-----------------	--------------	-----------	---------	--------	--------

Abbildung 5: NavBar

In Abbildung 5 ist die sogenannte NavBar zu sehen. Diese ermöglicht es Nutzern und Nutzerinnen zwischen den verschiedenen Seiten zu wechseln. So kommt ein eingeloggter Nutzer oder eine eingeloggte Nutzerin mit einem Klick auf die "Bestellungen" Schaltfläche auf die Subseite, auf der alle eigenen Bestellungen aufgeführt werden. Die NavBar ist auf jeder Seite zu sehen und kann somit jederzeit von Nutzern oder Nutzerinnen benutzt werden, um auf die gewünschte Seite zu wechseln.

[Gels]

### 3.6.2 AllergenBar



Abbildung 6: AllergenBar

In Abbildung 6 ist die sogenannte AllergenBar zu sehen. Diese ermöglicht es Nutzern und Nutzerinnen auf Seiten der Produkte, sowie auf denen der Kreationen, nach Allergenen zu filtern. Klickt ein Nutzer oder eine Nutzerin so zum Beispiel auf der Seite “Hauskreationen” auf die “Gluten” Schaltfläche, so werden alle Hauskreationen auf der Seite angezeigt, für die das Allergen Gluten nicht eingetragen ist. Das Allergen “Vegan” umfasst alle tierischen Allergene: Ei, Gelatine, Honig und Laktose. Sollte “Vegan” ausgewählt werden, so werden automatische diese vier Allergene herausgefiltert.

Durch einen erneuten Klick auf das gleiche Allergen in der AllergenBar wird dieses wieder entfernt und die Lebensmittel mit diesem Allergen werden wieder angezeigt. Außerdem entfernt ein neu laden der Seite über die zuvor beschriebene NavBar alle eingetragenen Allergene.

[Gels]

### 3.6.3 Startseite

## Willkommen beim Ice Cream Creator

Abbildung 7: Startseite der Anwendung

In Abbildung 7 ist die Startseite zu sehen. Auf dieser hat der Nutzer oder die Nutzerin die Möglichkeit sich zu registrieren. Sollte bereits ein Account bestehen, kann über die zuvor beschriebene NavBar auf die gewünschte Seite gewechselt werden.

[Gels]

### 3.6.4 Hauskreationen

In der Hauskreationen Seite gibt es zwei Ansichten: die des Nutzers oder der Nutzerin und die des Administrators.

### Cookie Lover Deluxe

**Eissorte**

- Vanille

**Toppings**

- Cookie Dough
- Cookies
- Oreo

**Soße**

- Schokolade

**Allergene**

- Laktose
- Gluten

Preis 5.90 €

Ansehen

1

In den Warenkorb

### Schoko Schock

**Eissorte**

- Schokolade

**Toppings**

- Brownie
- Schokoflakes
- Schokosahne

**Soße**

- Schokolade

**Allergene**

- Laktose
- Gluten

Preis 5.90 €

Ansehen

1

In den Warenkorb

Abbildung 8: Ansicht auf die Hauskreationen Seite als Nutzer oder Nutzerin

In Abbildung 8 ist eine beispielhafte Ansicht der Hauskreationen Seite aus Sicht eines Nutzers oder einer Nutzerin zu sehen. Auf dieser können alle aktuell in der Datenbank gespeicherten Hauskreationen eingesehen werden. So wird zuerst der Name angezeigt und danach die Produkte, aus denen die Kreation besteht. Darunter werden die Allergene angezeigt, die sich aus den Komponenten der Kreation ableiten. Anschließend ist der Preis angezeigt, der sich aus einem Grundpreis, der Menge und Art der Zutaten und ob eine Sauce in der Kreation vorhanden ist, berechnet.

Mit dem “Ansehen” Knopf wird der Nutzer oder die Nutzerin auf die Subseite für die gewählte Hauskreation geleitet. Diese ist rein zur Information gedacht und beinhaltet keine weitere Funktionalität für den Nutzer oder die Nutzerin.

In dem Eingabefeld kann eine Menge eingetragen werden, entsprechend welcher dann die Anzahl der gewählten Hauskreation in die offene Bestellung des Nutzers oder der Nutzerin eingefügt wird. Sollte es keine offene Bestellung geben, so wird eine neue erstellt.

Neue Hauskreation

Abbildung 9 Ein Knopf mit der Aufschrift „Neue Hauskreation“

Als Administrator wird die Hauskreationen Seite noch um den Knopf aus Abbildung 9 erweitert. Mit diesem gelangt der Administrator auf eine Subseite auf der eine neue Hauskreation erstellt werden kann.

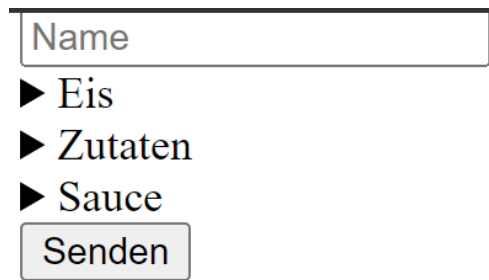


Abbildung 10: Die Seite zum Erstellen einer neuen Hauskreation

In Abbildung 10 ist diese Subseite zu sehen. In dem Eingabefeld kann der Name eingegeben werden. Dieser sollte nur Buchstaben beinhalten. Mit einem Klick auf die Dreiecke neben den Produktkategorien, werden die entsprechenden Felder ausgeklappt und es werden so alle in der Datenbank gespeicherten Komponenten der jeweiligen Kategorie aufgelistet. Bei der Eissorte muss eine gewählt werden, die zweite ist optional. Die Zutaten sind optional und lassen sich mit den Checkboxes hinzufügen. Am Ende kann noch eine Sauce gewählt werden. Sobald der Administrator auf "Senden" drückt, wird die Hauskreation angelegt und allen Nutzern und Nutzerinnen auch ab sofort angezeigt. Dabei werden der Preis und die Allergene aus der Liste der Produkte berechnet.

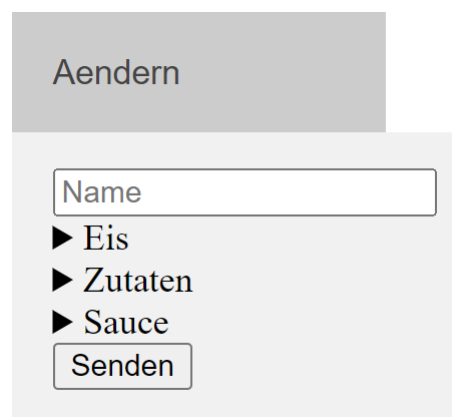


Abbildung 11: Das ausgeklappte Menü zum Ändern eines Hausbechers

In Abbildung 11 ist der nur für Administratoren sichtbare "Aendern" Knopf zu sehen. Dieser befindet sich auf der über den "Ansehen" Knopf erreichbaren Subseite. Ähnlich wie auf der Subseite zum Erstellen einer neuen Hauskreation kann ein Administrator hier eine bereits bestehende Hauskreation abändern. Über die Dreiecke neben den Produktkategorien kann das jeweilige Menü ausgeklappt werden. Bei Eis und Sauce werden die zuvor gewählten Produkte eingetragen und können durch einen Klick geändert werden. Zutaten müssen hinzugefügt werden.

[Gels]



### 3.6.5 Eigenkreationen

Jeder Nutzer oder Nutzerin hat die Möglichkeit Eigenkreationen zu erstellen und diese zu bestellen. Ebenso können sie bereits erstellte Eigenkreationen bearbeiten und abändern. Über den Reiter Eigenkreationen in der NavBar gelangen die Nutzer und Nutzerinnen auf die Seite der Eigenkreationen. Auf der Seite werden der Nutzerin oder dem Nutzer dann ihre/seine bereits erstellten Eigenkreationen angezeigt.

#### Eigenkreationen

Erstelle deine eigene Kreation

##### Test

###### Eissorte

- Schokolade
- Erdbeere

###### Toppings

- Daimtorte
- Banane
- Soße
- Mangopuere

###### Allergene

- GLUTEN
- EI
- LAKTOSE

Preis: 5.70 €

Ansehen

1

In den Warenkorb

##### Test Kreation

###### Eissorte

- Salted Caramell
- Vanille Vegan

###### Toppings

- Cashews
- Blaubeere
- Brauseufos
- Haselnusskrokant
- Soße

- Erdbeere hausgemacht

###### Allergene

- NUSS
- LAKTOSE

Preis: 6.90 €

Ansehen

1

In den Warenkorb

Abbildung 12: Liste der Eigenkreationen

Abbildung 12 zeigt die Liste der Eigenkreationen von einem Nutzer oder einer Nutzerin, aus dieser können die bereits erstellten Kreationen erneut bestellt werden, dazu muss eine Anzahl in das Feld eingeben und auf "In den Warenkorb" geklickt werden.

Erstelle deine eigene Kreation

Abbildung 13: Knopf zum Erstellen einer Eigenkreation

Am Anfang der Seite ist der "Erstelle deine eigene Kreation" Knopf, wie in Abbildung 13 zu sehen, mit welchem es möglich ist eine neue Eigenkreation zu erstellen.

**Erstelle deinen eigenen Becher!**

Name

► Eis

► Zutaten

► Sauce

Anzahl

Daten absenden

Abbildung 14: Neue Eigenkreation erstellen

Abbildung 14 zeigt die Subseite, in der das Menü zum Erstellen der Eigenkreation zu sehen ist. Hier kann ein eigener Name für die Kreation festgelegt werden, neben den verschiedenen Eissorten, Zutaten und Saucen. Um die Erstellung der Eigenkreation abzuschließen, muss auf den “Daten absenden” Knopf geklickt werden, nachdem die Anzahl angegeben wurde. Daraufhin wird die Eigenkreation automatisch der aktuellen Bestellung hinzugefügt.

War das Erstellen erfolgreich, wird der Nutzer oder die Nutzerin zurück auf die Seite der Eigenkreationen weitergeleitet. Hier kann der Nutzer oder die Nutzerin, wie in Abbildung 12 zu sehen, auf den “Ansehen” Knopf klicken, um die Eigenkreation einzeln anzuschauen.

**Test**

**Eissorte**

- Schokolade
- Erdbeere

**Toppings**

- Daimtorte
- Banane

**Soße**

- Mangopuree

**Allergene**

- GLUTEN
- EI
- LAKTOSE

**Preis:** 5.70 €

Ändere jetzt deinen Becher fuer eine noch bessere Kreation!

Test

► Eis

► Zutaten

► Sauce

Daten absenden

Abbildung 15: Ansicht auf eine einzelne Eigenkreation

In der Abbildung 15 ist nun die Eigenkreation einzeln aufgelistet, dort ist es auch möglich die Eigenkreation zu verändern. Verändert werden kann die Eigenkreation in dem unten neue Eissorten, Zutaten oder eine neue Sauce eingegeben werden. Ebenso ist es möglich den Namen der Eigenkreation zu ändern.

Ändere jetzt deinen Becher fuer eine noch bessere Kreation!

Test

▼ Eis

☒ Schokolade  
☐ Vanille  
☐ Joghurt  
☒ Erdbeere  
☐ Mango  
☐ Schokolade Vegan  
☐ Vanille Vegan  
☐ Nuss-Nougat  
☐ Salted Caramell

► Zutaten

► Sauce

Daten absenden

Abbildung 16: Menü zum Ändern einer Eigenkreation

Klickt man nun auf eine der Punkte öffnet sich ein Dropdown Menü, in dem man die neuen Komponenten ganz einfach auswählen kann, ähnlich zu dem beim Erstellen einer neuen Eigenkreation. Vorausgewählt sind schon die Komponenten, die bereits in der Eigenkreation enthalten sind, was in Abbildung 16 zu sehen ist. Wie schon in 3.6.2 beschrieben kann durch Auswahl in der AllergenBar nach Allergenen gefiltert werden. Durch Auswahl des Filters werden automatisch Komponenten, die dem Filter entsprechen, ausgeblendet. Ist man zufrieden mit den Änderungen an seiner Eigenkreation, können diese mit Klick auf den “Daten absenden” Knopf gespeichert werden.

[Bierenriede]

### 3.6.6 Bestellungen

## Liste aller Bestellungen

### Bestellung

**4 x Mein eigener Becher**

**1 x Cookie Lover Deluxe**

**3 x Glatteis**

**Gesamtpreis: 47.20 €**

Ansehen

Bestellung Loeschen

Bestellung abschicken

Abbildung 17: Die Liste aller Bestellungen eines eingeloggtten Nutzers

In Abbildung 17 ist die Bestellungen Seite aus der Sicht eines eingeloggtten Nutzers oder einer eingeloggtten Nutzerin zu sehen. Ein Nutzer oder eine Nutzerin bekommt nur seine/ihre eigene/n Bestellungen angezeigt. Dabei befindet sich bereits eine beispielhafte Eingabe für eine Bestellung im System. Diese umfasst aktuell 3 unterschiedliche Kreationen, dabei wurde die Kreation “Mein eigener Becher” vier Mal, die Kreation “Cookie Lover Deluxe” einmal und die

Kreation "Glatteis" drei Mal hinzugefügt. Unter den Kreationen steht ein Gesamtpreis, der sich aus den Einzelpreisen berechnet.

### Die gesuchte Bestellung:

#### Eigenkreationen:

4 x Mein eigener Becher, 23.60 €

Ansehen

Loeschen

#### Hauskreationen:

1 x Cookie Lover Deluxe, 5.90 €

Ansehen

Loeschen

3 x Glatteis, 17.70 €

Ansehen

Loeschen

Bestellung abschicken

Abbildung 18: Eine einzelne Bestellung eines eingeloggten Nutzers

Mit dem Knopf "Ansehen" kommt der Nutzer oder die Nutzerin auf die Subseite die auf Abbildung 18 zu sehen ist. Auf dieser wird die gewählte Bestellung detaillierter beschrieben. Die Kreationen werden nach Eigen- und Hauskreationen aufgeteilt. Ebenso wird der Gesamtpreis auf die einzelnen Posten aufgeteilt.

Auf den einzelnen Posten kann nun auch jeweils ein "Ansehen" Knopf gefunden werden. Mit diesem kommt der Nutzer oder die Nutzerin auf die Subseite der Eigenkreation, die in 3.6.5 beschrieben ist oder auf die Subseite der Hauskreation, die in 3.6.4 beschrieben ist, je nachdem um welche Art von Kreation es sich handelt.

Des Weiteren gibt es pro Posten einen Knopf "Loeschen" mit dem sich dieser Posten aus der aktuellen Bestellung löschen lässt. Dieser Knopf ist nur sichtbar, solange eine Bestellung noch nicht abgeschickt wurde.

Wenn es eine noch nicht abgeschickte Bestellung des eingeloggten Nutzers oder der eingeloggten Nutzerin gibt, dann wird auf der über die NavBar erreichbaren Bestellungen Seite und auf der entsprechenden Bestellungen Subseite ein "Bestellung abschicken" Knopf angezeigt.

### Liste aller Bestellungen

#### Bestellung

4 x Mein eigener Becher

1 x Cookie Lover Deluxe

3 x Glatteis

Gesamtpreis: 47.20 €

Ansehen

Abbildung 19: Liste aller Bestellungen nach dem abschicken der offenen Bestellung

In Abbildung 19 ist die Seite nach dem Abschicken der Bestellung zu sehen. Die Bestellung kann nur noch angesehen werden, jedoch können keine schreibenden Prozesse mehr ausgeführt werden.

### Die gesuchte Bestellung:

#### Eigenkreationen:

4 x Mein eigener Becher, 23.60 €

Ansehen

#### Hauskreationen:

1 x Cookie Lover Deluxe, 5.90 €

Ansehen

3 x Glatteis, 17.70 €

Ansehen

Abbildung 20: Detailansicht einer Bestellung, nachdem diese abgeschickt wurde

In Abbildung 20 ist die Subseite nach dem Abschicken der Bestellung zu sehen. Auch hier können nur noch lesende, aber keine schreibenden Prozesse mehr ausgeführt werden.

## Bestellung

4 x Mein eigener Becher

1 x Cookie Lover Deluxe

3 x Glatteis

Gesamtpreis: 47.20 €

Ansehen

## Bestellung

1 x Kinder Wahnsinn

Gesamtpreis: 6.30 €

Ansehen

Bestellung Loeschen

Bestellung abschicken

Abbildung 21: Liste aller Bestellungen, bei der eine zweite automatisch erstellt wurde

In Abbildung 21 ist die Bestellungen Seite zu sehen, nachdem die erste Bestellung abgeschickt und eine weitere Kreation hinzugefügt wurde. Diese neue Kreation "Kinder Wahnsinn" wird in

einer neuen Bestellung aufgeführt. Für diese neue Bestellung gilt wieder der lesende und schreibende Zugriff.

[Gels]

### 3.6.7 Produkte

Im Folgenden werden beispielhaft an den Zutaten alle Produkte und ihre Seiten erklärt. Die Zutaten unterscheiden sich von den anderen Produkten, indem sie noch eine Premiumkomponente haben. Diese existiert bei den Eissorten und Saucen nicht und kann daher bei der Betrachtung dieser vernachlässigt werden.

#### **Brownie (hausgemacht)**

Dies ist eine Premiumzutat

**Allergene**

- Gluten

#### **Browniedough (hausgemacht)**

Dies ist eine Premiumzutat

**Allergene**

- Gluten

#### **Cashews**

Dies ist eine Premiumzutat

**Allergene**

- Nuss

Abbildung 22: Ausschnitt aus der Liste aller Zutaten

In Abbildung 22 sind beispielhafte Zutaten auf der über die NavBar erreichbaren Zutaten Seite aus Sicht eines eingeloggten Nutzers oder einer eingeloggten Nutzerin zu sehen. Diese sind wie folgt aufgebaut: Als Überschrift einer jeden Zutat steht der Name. Darunter ist aufgeführt, ob es sich bei dieser Zutat um eine Premiumzutat handelt oder nicht. Dies ist für die Berechnung des Preises relevant, da Premiumzutaten mehr kosten als nicht Premiumzutaten. Als nächstes werden alle Allergene angezeigt, die für diese Zutat vorhanden sind. Mit der AllergenBar aus 3.6.2 können diese Allergene herausgefiltert werden.

**Brownie (hausgemacht)**

Dies ist eine Premiumzutat

**Allergene**

- Gluten

Aendern

Name

Premium: ☒ Ja ☐ Nein

▼ Allergene:

- ☐ ALKOHOL
- ☐ EI
- ☐ GELATINE
- ☐ GLUTEN
- ☐ HONIG
- ☐ KOFFEIN
- ☐ LAKTOSE
- ☐ NUSS
- ☐ SUESSSTOFF

Senden

**Browniedough (hausgemacht)**

Dies ist eine Premiumzutat

**Allergene**

- Gluten

Aendern

**Cashews**

Dies ist eine Premiumzutat

**Allergene**

- Nuss

Aendern

Abbildung 23: Kontextmenü zum Ändern einer Zutat

In Abbildung 23 sieht man die gleiche Seite aus Sicht eines Administrators. Das “Aendern” Menü ist bereits für die “Brownie (hausgemacht)” Zutat ausgeklappt. In diesem Menü kann ein Administrator die entsprechende Zutat abändern. Dabei wird zuerst ein Name gewählt, wobei dieser gleichbleiben oder sich ändern kann. Mit den Radio-Feldern “Ja” und “Nein” neben dem “Premium:” Text kann der Administrator entscheiden, ob sich der Premiumzustand der Zutat ändert. In der Liste des Allergene Menüs kann der Administrator auswählen, welche der Allergene auf die Zutat zutreffen.

Nachdem der “Senden” Knopf gedrückt wurde, wird die Zutat in der Datenbank neu abgespeichert und alle Kreationen, die diese Zutat beinhalten, werden entsprechend aktualisiert. Des Weiteren werden die Allergenlisten aller Kreationen angepasst.

[Gels]

**3.6.8 Nutzer**

Auch auf der Seite für die Nutzerinnen und Nutzer gibt es wie bei den Hauskreationen unterschiedliche Ansichten für die Nutzer und den Administratoren.

# Hallo Stefan!

## Deine Accountinformationen:

**Stefan**

Kunde

Einstellungen

Abbildung 24: Nutzer Ansicht

Auch die Nutzer Seite, welche in Abbildung 24 zu sehen ist, ist über die NavBar erreichbar, auf dieser werden der Nutzerin oder dem Nutzer die persönlichen Accountinformationen angezeigt. Diese Informationen bestehen aus dem Nutzernamen und der Rolle, die dem Nutzer oder der Nutzerin zugewiesen wurde.

Einstellungen

Abbildung 25: Einstellungen Knopf

Die persönlichen Zugangsdaten können auch geändert werden, dazu einfach auf den in Abbildung 25 zu sehenden Knopf "Einstellungen" klicken.

Name
Passwort
Daten absenden

Abbildung 26: Zugangsdaten ändern

Daraufhin öffnen sich zwei Eingabefelder, wie in Abbildung 26 zusehen, in denen ein neuer Nutzername und ein neues Passwort festgelegt werden kann. Bestätigt wird die Änderung mit dem Klick auf den "Daten absenden" Knopf.

Der Administrator hingegen erhält beim Öffnen der Nutzer Seite eine andere Ansicht als der Nutzer.



## **Alle Registrierten Nutzer**

**0, admin, Admin**

**1, user, Kunde**

**2, Stefan, Kunde**

Abbildung 27: Nutzer Seite als Administrator

In Abbildung 27 ist die Nutzer Seiten Ansicht für den Administrator zu sehen, dieser sieht nicht seine Accountinformationen, sondern eine Übersicht über alle registrierten Nutzerinnen und Nutzer in der Datenbank. Die Einträge bestehen aus der Nutzer ID, dem Nutzernamen und der dem Account zugewiesenen Rolle.

[Bierenriede]

## 4 Zusammenfassung und Fazit

In diesem Kapitel werden die vorgestellten Ergebnisse kurz Bewertung, außerdem wird ein kurzes Fazit für die Anwendung gezogen. Zum Schluss kommt noch ein Ausblick für die Anwendung, wie diese noch weiterentwickelt werden kann und welche Teile vielleicht noch verbessert oder eingefügt werden könnten.

### 4.1 Bewertung der Ergebnisse

Die entwickelte Anwendung erfüllt alle im Projektvorschlag beschriebenen Musskriterien, leider konnte aus Zeitmangel keines der Wunschkriterien implementiert werden. Zur Entwicklung der Anwendung wurden die in der Vorlesung und Praktikum vorgestellten Techniken und Methoden angewendet.

Bei der Implementierung der Funktionen wurde stets darauf geachtet die Funktionalitäten simple und einfach zu gestalten, um eine möglichst hohe Nutzungsvielfalt zu gewährleisten. Während der Entwicklung mussten immer wieder Entscheidungen getroffen werden, in welche Richtung die Anwendung ihre Schwerpunkte legt. Anfangs war der Plan diese auf die Security, speziell dem Identitäts- und Zugriffsmanagement, inklusive sicherem Ein- und Ausloggen, zu legen. Schlussendlich wurde eine Zugriffskontrolle per RBAC realisiert und bietet einen guten Mittelweg, um eine Zugriffskontrolle zu implementieren, welche nicht zu komplex wird. Dafür wurde während der Implementierung der einzelnen Funktionen sehr auf eine lose Kopplung zwischen den einzelnen Modulen geachtet. Konkret wurde dies mit verschiedenen Injects und Events umgesetzt.

[Bierenriede]

### 4.2 Fazit

Die Anwendung hat einen ersten soliden Entwicklungsstand erreicht, der immer noch weiter ausbaufähig ist, trotzdem können mit der entwickelten REST-API bereits funktionale Anwendungen umgesetzt werden.

[Bierenriede]

### 4.3 Ausblick

Auf Basis dieser Anwendung können noch weitere Features implementiert werden, vorangehend die im Projektvorschlag beschriebenen Wunschkriterien. Genauso ist auch das Frontend sehr einfach implementiert worden. Dort könnte noch die Oberfläche angepasst werden, um die dargestellten Daten etwas besser zu strukturieren. Des Weiteren könnte der Login noch verbessert werden, dafür kann zum Beispiel Keycloak oder Java Web Tokens verwendet werden.

[Bierenriede]

Inhaltlich kann das Projekt noch um einige andere Punkte erweitert werden. So wäre zum Beispiel eine dedizierte Seite für den Administrator sinnvoll, auf der dieser eine statische Übersicht, über die am meisten genutzten Hauskreationen und Produkte erhält. Mit dieser Statistik könnte so agil reagiert werden, sodass die entsprechenden Produkte nicht in zu geringen oder hohen Mengen gekauft werden. Außerdem könnten so Produkte, die nicht mehr genutzt werden, frühzeitig aus dem Sortiment genommen und eventuell durch neue ersetzt werden. Eine

andere mögliche Erweiterung, wäre das Einfügen einer Gutscheineseite, mit der Kunden Gutscheine kaufen und verschenken und vorzeigen können. Außerdem könnte so ein Treuepunktsystem digitalisiert werden.

[Gels]

## 5 Referenzen

Web-Seiten zuletzt am 01.08.2022 abgerufen.

- [@MVN] Apache Maven Tutorial | Baeldung, <https://www.baeldung.com/maven>
- [@QRK] What is Quarkus? - Quarkus, <https://quarkus.io/about/>
- [@RED] Was ist Quarkus? <https://www.redhat.com/de/topics/cloud-native-apps/what-is-quarkus>
- [@DKR] Was ist Docker? Was sind die Vorteile? Einfach erklärt (opc-router.de) <https://www.opc-router.de/was-ist-docker/#:~:text=Docker%20ist%20ein%20beliebtes%20Open,sowie%20die%20Systemressourcen%20gemeinsam%20nutzen.>
- [@RRA] Was ist eine REST-API? <https://www.redhat.com/de/topics/api/what-is-a-rest-api>
- [@JSN] JSON <https://www.json.org/json-de.html>
- [@JSB] JSON-B <https://rieckpil.de/whatis-json-binding-json-b/>
- [@JXB] JAX-RS is just an API! Baeldung <https://www.baeldung.com/jax-rs-spec-and-implementations>
- [@JXR] Chapter 2. Developing JAX-RS Web Services Red Hat JBoss Enterprise Application Platform [https://access.redhat.com/documentation/en-us/red\\_hat\\_jboss\\_enterprise\\_application\\_platform/7.0/html/developing\\_web\\_services\\_applications/developing\\_jax\\_rs\\_web\\_services](https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.0/html/developing_web_services_applications/developing_jax_rs_web_services)
- [@JXO] Chapter 13 Building RESTful Web Services with JAX-RS <https://docs.oracle.com/cd/E19798-01/821-1841/6nmq2cp1v/index.html#:~:text=JAX%2DRS%20is%20a%20Java,development%20of%20RESTful%20web%20services.>
- [@QHP] <https://quarkus.io/guides/hibernate-orm-panache>
- [@BQT] Testing Quarkus Applications  
<https://www.baeldung.com/java-quarkus-testing>
- [@JTA] Understanding Java Transaction API (JTA) Management Tutorial <https://www.progress.com/tutorials/jdbc/understanding-jta>
- [@QOS] Open API und Swagger UI <https://quarkus.io/guides/openapi-swaggerui>
- [@QTE] Quarkus Templating Engine <https://quarkus.io/guides/qute>