

# REST-Food

## Dokumentation



# Inhaltsverzeichnis

<b>Dokumentation</b>	<b>1</b>
<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>Vorgehen</b>	<b>3</b>
Frontend	3
Backend	4
<b>Anleitung</b>	<b>5</b>
Aufsetzen des Projekts	5
Tests	5
<b>Grafiken</b>	<b>6</b>
Klassendiagramm	6
Mockup	6
<b>Persönliche Reflexionen</b>	<b>7</b>
Valentin	7
Ruben	7
Yan	7
Loris	8

# Vorgehen

Kategorie	Task	Zuständiger	Status	Priorität	Aufwand geschätzt (min)	Komplexität
Frontend	Mockup	Loris	Erledigt	1	60	Einfach
Frontend	Startseite	Ruben	Erledigt	1	250	Mittel
Frontend	About us	Ruben	Erledigt	2	250	Mittel
Frontend	Menuseite	Loris	Erledigt	3	500	Schwierig
Frontend	Reservationseite	Loris	Erledigt	4	600	Schwierig
Frontend	Menuleiste	Loris	Erledigt	1	200	Mittel
Backend	Klassendiagramm	Valentin	Erledigt	1	60	Mittel
Backend	Menu API	Valentin	Erledigt	3	800	Schwierig
Backend	Reservations API	Yan	Erledigt	4	800	Schwierig
Dokumentation	Grundstruktur	Valentin	Erledigt	1	45	Einfach
Dokumentation	Vorgehen Backend	Yan	Erledigt	1	90	Einfach
Dokumentation	Vorgehen Frontend	Ruben	Erledigt	1	90	Einfach
Dokumentation	Anleitung	Alle	Erledigt	1	100	Einfach
Dokumentation	Swagger für Backend	Valentin	Zu erledigen	4	70	Einfach
Dokumentation	Javadoc Backend	Yan	Erledigt	4	80	Einfach
Dokumentation	Individuelle Reflexion	Alle	In Bearbeitung	1	4x 45	Einfach
Präsentation	Powerpoint	Ruben	Erledigt	1	250	Einfach
Präsentation	Präsentation üben	Alle	Zu erledigen	1	45	Einfach
Scrum	Product Backlog	Valentin	Erledigt	1	45	Einfach
Scrum	Sprint Planning	Valentin	Erledigt	1	30	Einfach
Scrum	Daily Scrum	Alle	Erledigt	1	6x 30	Einfach
Scrum	Daily Sprint	Alle	Erledigt	1	6x 25	Einfach

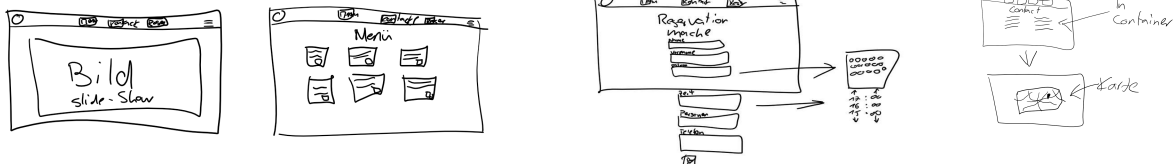
Aktueller Stand: + Scrum

## Frontend

Um das Frontend für unser Projekt zu erstellen, haben Loris und Ruben als erstes ein kleines Mockup der Seiten erstellt. Wir haben uns dazu entschieden, nur eine ungefähre Skizze zu machen und den Rest während dem Programmieren zu entscheiden.

Mockup

Freitag, 3. Februar 2023 10:26



Wir haben zuerst gemeinsam die Startseite gemacht und danach die anderen Seiten zwischen Loris und Ruben eingeteilt. Nachdem wir das Design hatten, konnten wir anfangen, Daten aus dem Backend zu nehmen und einzufügen, was zu einigen kleinen Problemen führte.

Die grösste Herausforderung für uns war, eine Reservation zu machen und diese Informationen an das Backend zu geben. Wir haben zuerst versucht, durch ein normales Textfeld einfache Strings weiterzugeben, was aber nicht durchdacht war, da das Datum ein bestimmtes Format haben sollte und man in einem String irgendwelche Buchstaben und Zahlen in beliebigen Reihenfolgen eingeben kann.

Schlussendlich hat Loris es geschafft, diese Textfelder durch Datum und Zeit Eingabefelder zu ersetzen.

## Backend

Um das Backend unseres Projektes zu erstellen, haben Valentin und Yan zuerst zusammen einen Plan der Klassen erstellt, die wir brauchen werden. Die Klassen, die wir entschieden haben zu machen waren, die folgenden Klassen und Felder:

1. Menu: Ein Menu, das unseren Kunden angeboten wird
  - 1.1. [int] menuID
  - 1.2. [int] relevance
  - 1.3. [String] name
  - 1.4. [int] price
  - 1.5. [bool] vegetarian
  - 1.6. [String] type
  - 1.7. [String] description
  - 1.8. [String] img\_url
2. Reservation: Eine Buchung eines Tisches
  - 2.1. [int] reservationID
  - 2.2. [int] reserverID
  - 2.3. [String] visitorCount
  - 2.4. [String] date
  - 2.5. [String] time
3. Client: Daten unseren Kunden die eine Reservation machen
  - 3.1. [int] clientID
  - 3.2. [String] firstName
  - 3.3. [String] lastName
  - 3.4. [String] phoneNumber

Nach einiger Zeit haben wir entschieden, die "Client" Klasse herauszunehmen und stattdessen mehr Informationen in die "Reservation" Klasse zu speichern. Wir haben es gemacht, weil es die Verwaltung der Daten viel vereinfacht hat. Also hatten wir am Ende die folgenden Klassen:

1. Menu:
  - 1.1. [int] menuID
  - 1.2. [int] relevance
  - 1.3. [String] name
  - 1.4. [int] price
  - 1.5. [bool] vegetarian
  - 1.6. [String] type
  - 1.7. [String] description
  - 1.8. [String] img\_url
2. Reservation:
  - 2.1. [int] reservationID
  - 2.2. [int] reserverID
  - 2.3. [String] visitorCount
  - 2.4. [String] firstName
  - 2.5. [String] lastName
  - 2.6. [String] date
  - 2.7. [String] time

2.8. [String] phoneNumber

2.9.

Wir sind nicht auf viele Probleme beim Entwickeln des Backends gestossen. Wir hatten einige Probleme mit Gradle und der Verbindung mit der Datenbank. Bei wir haben die Probleme mit Gradle dadurch gelöst, dass wir die Version auf eine ältere gewechselt haben. Das Problem mit der Verbindung zu der Datenbank war einiges schwerer zu lösen. Bei der Verbindung wählten am Anfang einen falschen Ansatz. Wir wollten die gesamte Datenbank manuell mit SQL aussetzen. Uns wurde aber bewusst, dass es ein sehr ineffizienter Lösungsansatz war und er gar nicht funktionierte. Deshalb überlesen wir das ganze Spring Boot, was unser Problem löst. Als Letztes haben wir noch das Backend mit dem Frontend verbunden. Hierbei mussten wir praktisch nichts machen, da wir den Code bereits beim Programmieren darauf ausgerichtet haben. Natürlich testeten wir unsere API davor noch gründlich mit Postmann.

## Anleitung

### Aufsetzen des Projekts

Um das Projekt überhaupt zum Laufen, zu bringen braucht es die Funktionen von 3 verschiedenen Programmen. Als Erstes muss man durch die Ausführung von diesem Befehl im CMD Prompt den Docker installieren.

```
docker run -d -p 5432:5432 -e POSTGRES_PASSWORD=123123 --name rest-food-db postgres
```

Danach kann man den Docker Container ganz einfach einstellen.



Danach das Backend Projekt in IntelliJ öffnen und den Gradle BootRun starten. Somit wäre das Backend bereit und man muss noch das Frontend starten.

Um das Frontend zu starten, das Projekt in VS Code öffnen, das Terminal öffnen und den Befehl "npm start" eingeben.

Somit kann man nun das Projekt im Browser anschauen und benutzen.

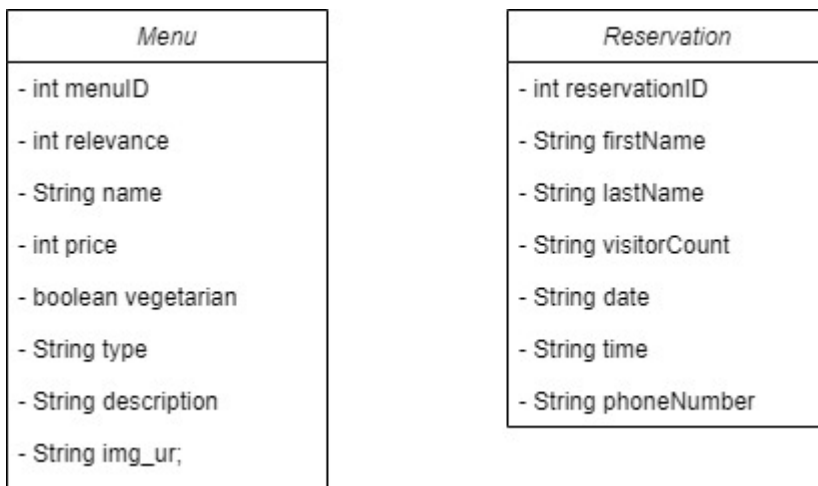
## Tests

Wir führten verschiedene Test sowohl im Frontend, als auch im Backend durch. Im Frontend testeten wir den Code durch Blackbox und Whitebox Tests. Diese waren bis auf ein paar kleine Ausnahmen erfolgreich. Das Backend testeten wir ebenfalls durch Blackbox und Whitebox Test. Wir führten aber zusätzlich noch Postmann Test durch, welche, wie man sieht erfolgreich waren.

PASS	Endpoint Exists
PASS	Data Exists
PASS	Status test
PASS	Valid Response body

## Grafiken

### Klassendiagramm

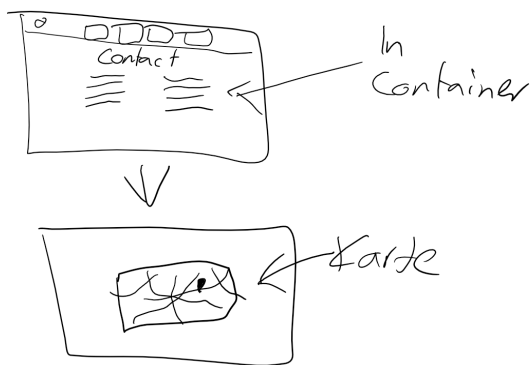


[https://drive.google.com/file/d/1gWxVumOtMboGRAXgCgM8\\_pNtIINNFyO9Z/view?usp=sharing](https://drive.google.com/file/d/1gWxVumOtMboGRAXgCgM8_pNtIINNFyO9Z/view?usp=sharing)

### Mockup

Mockup  
Freitag, 3. Februar 2023 10:26





## Persönliche Reflexionen

### Valentin

Ich fand, dass wir in diesem Projekt sehr gute Ergebnisse erzielen konnten. Die Teamarbeit funktionierte sehr gut. Alle konnten ihren Beitrag leisten und wir haben uns alle immer gut verstanden. Wir konnten zudem sehr effizient arbeiten und hatten keine grossen Probleme. Auch die Kommunikation lief überraschend gut. Wir hatten nicht zu wenige oder viele Calls und sprachen alles sonstiges über den Teams-Chat. Das hat mir sehr gut gefallen und bin ich von anderen Projekten nicht so gewohnt. Mit der Scrum Arbeitsweise hatte ich am Anfang Probleme, konnte mich aber mit der Zeit sehr vertraut machen und habe es lieben gelernt. Ich hoffe, dass wir in nächster Zeit noch mehr solcher Projekte haben, da es einfach sehr viel Spass bereitet hat.

### Ruben

Ich bin zufrieden mit unserem Projekt. Die Kommunikation lief sehr gut durch den Teams-Chat und durch die Calls von zuhause. Wir haben direkt mit dem Frontend gestartet, was auch dazu führte, dass wir sehr schnell schon einen grossen Fortschritt hatten. Wir hatten eigentlich keine grossen Probleme, ausser dass wir die `node_modules` nicht im Gitignore-File hatten und dadurch immer riesige Commits hatten.

### Yan

Ich denke, dass wir das Projekt gut gemacht haben. Meiner Meinung nach waren unsere Kommunikation und Planung sehr erfolgreich gewesen. Es sind uns nicht so viele Probleme begegnet. Ich persönlich hatte nur zweimal ein Problem mit Gradle, aber am Ende konnte ich sie lösen.

## Loris

Die Gruppenarbeit, bei der ich mich auf das Frontend konzentriert habe, war eine bereichernde Erfahrung für mich. Im Büro war die Kommunikation viel besser und hat sich ständig ausgetauscht. Während des ersten Homeoffice-Tag war ich nicht besonders produktiv, aber beim zweiten Mal habe ich meine Arbeitsweise verbessert und konnte effektiver arbeiten. Die Möglichkeit, am Frontend zu arbeiten, hat mir viel Freude bereitet, da ich meine kreativen Fähigkeiten einsetzen und visuell ansprechende Lösungen entwickeln konnte. Es war inspirierend, gemeinsam mit anderen Teammitgliedern an einem Projekt zu arbeiten und Ideen auszutauschen. Trotz der anfänglichen Herausforderungen bin ich stolz darauf, wie ich mich weiterentwickelt und letztendlich durchgearbeitet habe, um positive Ergebnisse zu erzielen.