

# Dokumentation des Projektes



## TEAM

4. Semester MM / MI

### MM7:

Patrick Düll  
Jolanda Jerg  
Antonia Opitz  
Ismael Abdelatif

### MI7:

Oliver Maicher  
Yannick Möller  
Patrick Thomasius

## TECH STACK

### frontend:

Kunde: Android App  
Business: React Native,  
JavaScript

### backend:

NodeJS, GraphQL, RethinkDB

## MOTIVATION

übersichtliches Menü

schnelle Bestellung  
(Concurrency,  
Beschleunigung)

digitale Bezahlung  
PayPal

Bestellübersicht für das  
Restaurant

## Team



Yannick Möller, Antonia Opitz, Oliver Maicher, Patrick Thomasius, Ismael Abdelatif, Patrick Düll, Jolanda Jerg (von l. o. nach r. u.)

## Fernzugriff per SSH

Innerhalb des MI-VPNs ist per Terminal ein Remotezugriff durch SSH-Protokoll auf die VM möglich. Dort findet das Hosting des gesamten Projekts statt:

```
ssh root@webspec62.mi.hdm-stuttgart.de
```

mit dem Passwort: WebDev2019++

Starten des Servers, im Ordner “first-order/server”: *npm run stable*

## Idee “bestell.schnell”

Basierend auf der Entwicklung von Smartphones und deren Gegenwärtigkeit im Alltag und Präsenz in mittlerweile fast allen Gesellschaftsschichten entstand in freiem Diskurs die Idee eines zentralen Services zur Aufgabe von Bestellungen in Gastlokalen.

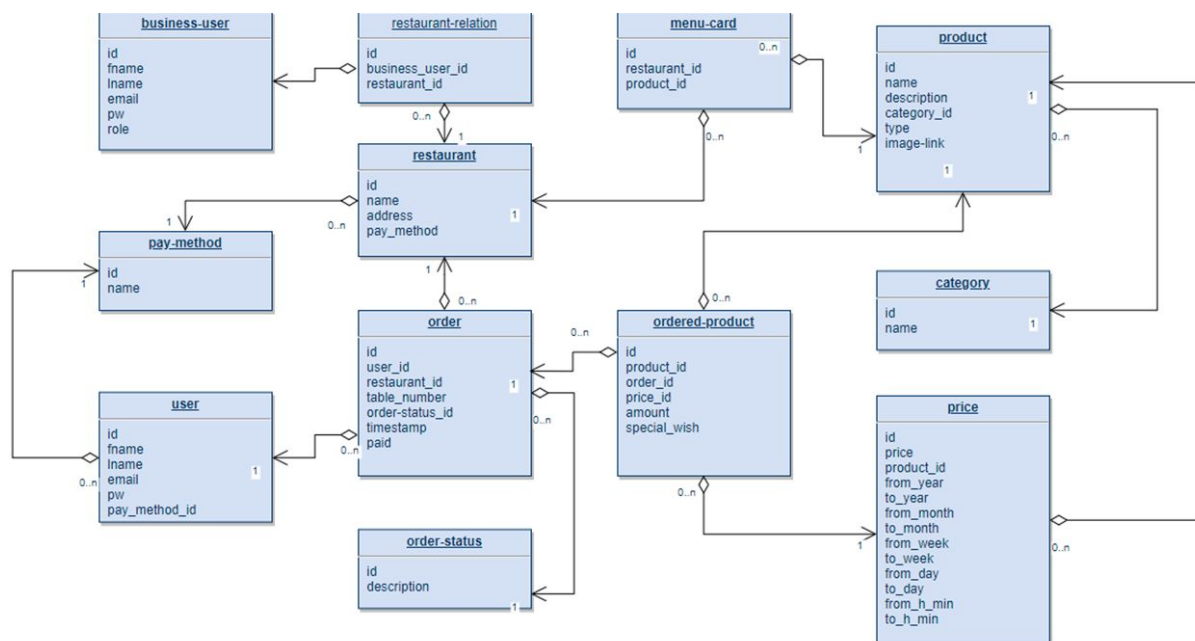
Unser Service bietet dem Nutzer die Möglichkeit in einem teilnehmenden Lokal mit dem eigenen Smartphone Bestellungen aufzugeben und zu bezahlen.

Anhand eines QR-Codes am Sitzplatz erfolgt die Zuordnung zum aktuellen Restaurant mit Tisch- und Sitzplatznummer. Bei erstmaliger Nutzung muss sich der Kunde registrieren und ein Konto zur Bezahlung angeben. Danach können die Angebote des Restaurants in der App eingesehen und Gerichte zur Bestellung ausgewählt werden. Zur Beendigung des Bestellvorgangs bezahlt der Kunde direkt online oder wie gewohnt nach dem Essen. Auf der Gegenseite wird für den Anbieter eine zentrale Plattform für die Bearbeitung der Bestellungen geboten. In der Küche können vorliegende Bestellungen eingesehen und bearbeitet werden und das Dienstpersonal erhält Informationen zur Auslieferung an den Gast und aktualisiert den Status der Bestellung.

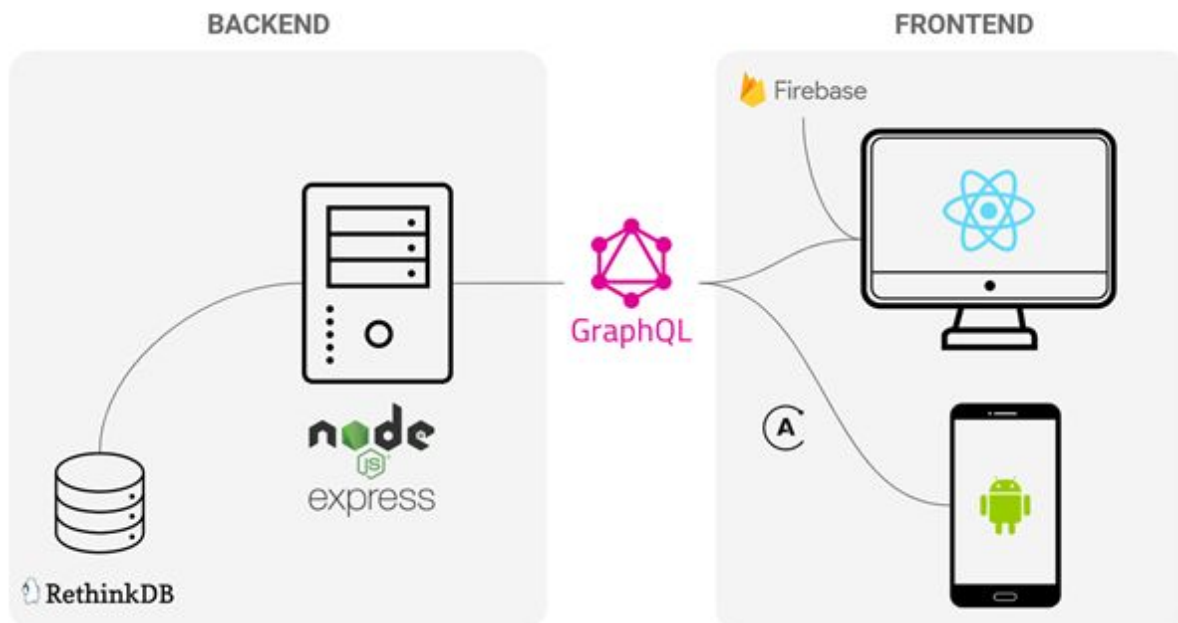
Es unterscheidet sich insofern von bestehenden Systemen, dass die Gäste ihr eigenen Endgeräte mitbringen und zur Bestellung verwenden. Damit erübrigen sich für den Anbieter die Anschaffung von zusätzlicher Hardware und die damit verbundenen Kosten. Desweiteren besteht der Vorteil des Systems in Zeitersparnis für beide Seiten. Einem Gast ist es jederzeit möglich eine Bestellung aufzugeben, unabhängig vom Personal des Lokals, auch Nachbestellungen sind damit ohne Wartezeit möglich. Ebenfalls trägt die digitale Bestellung zur Fehlerminimierung bei, d.h. Übertragungsfehler durch das Personal werden ausgeschlossen, da der Gast selbst die Verantwortung für die Auswahl trägt und somit bspw. auch individuell Informationen zu Allergenen und Zusatzstoffen erhält. Außerdem ist damit die korrekte Summe bei der Zahlung garantiert, die bereits vor Ausführung der Bestellung erfolgen kann. Leider geht hiermit der Arbeitsverlust von Personal einher, dieses muss einer automatisierten Lösung weichen und sollte dennoch ein Gast auf persönlicher Basis bestellen wollen, muss das natürlich weiterhin möglich sein. Die Kostenersparnis des Anbieters steht hier im Gegensatz zum Einkommensquelle des einfachen Bedienungspersonals. Weiterhin gibt es zwei Möglichkeiten: Aufbauend auf der digitalen Bestellung bringt das Personal

entweder weiterhin das Bestellte an den Tisch oder in Anlehnung an Schnellrestaurants gibt es eine Selbstabholtheke. Beide Varianten könnten ihre Anwendung finden, wobei es sogar eine dritte Option gibt. Bereits am Abend der Media Night gab es einen Interessenten. Eine Führungskraft aus dem operativen Geschäft der Bäckerei-Kette war interessiert uns als Anbieter eines solchen Systems zu gewinnen. Der kleine Unterschied in ihrem Beispiel besteht in der Tatsache, dass die Bäckerei-Kunden Waren vorbestellen und diese dann vor Ort abholen, wobei das mit unserem Service ebenfalls leicht umzusetzen ist.

## Datenmodell - erster Entwurf



## Tech-Stack



## Beschreibung des Tech-Stacks

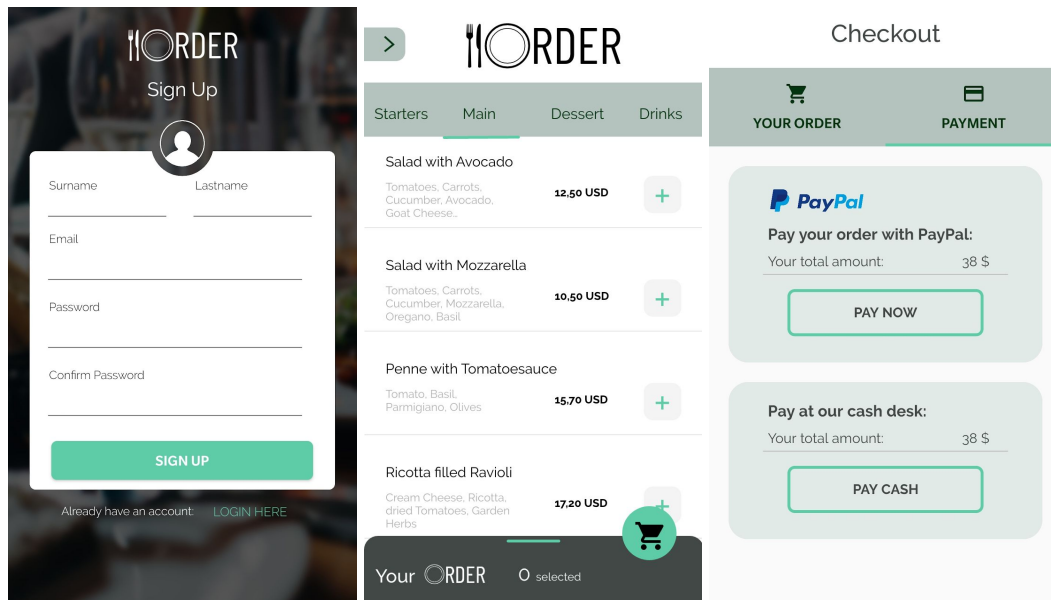
Ein *Node.js*-Server bildet die Basis unseres gesamten Backends, wobei der Einfachheit halber das *Express*-Framework zur Verwendung kam.

Hauptdatenbank unserer Anwendung ist *RethinkDB* mit Schema durch *Thinky*, da *MongoDB* und *Mongoose* bereits für uns bekannte Technologien darstellen. Gleichermassen gilt der Ansatz für die Verwendung von *GraphQL* anstelle einer standardisierten REST-API, ein vielversprechender Ansatz für die DB-Arbeit.

Das Frontend für den Restaurantbetreiber ist realisiert mit *React.js* und um die Oberflächengestaltung effizienter zu machen, kam *Material UI* zur Verwendung. Dies ermöglicht eine schnelle und einfache, aber optisch ansprechende Oberfläche.

Für das Speichern und Abrufen der Produktbilder wurde *Firebase DB* genutzt. Allen voran die schnelle und problemlose Implementation waren der dafür ausschlaggebende Punkt. *Firebase* selbst ist nur mit dem *React* Frontend verbunden, um unnötige Datenwege zu vermeiden.

Als Endnutzer-Anwendung haben wir eine *Android*-App in *Java* programmiert.



Frontend des Restaurantbesitzers als auch die Endnutzer-Anwendung kommunizieren via *GraphQL* mit dem Server und somit der Datenbank.

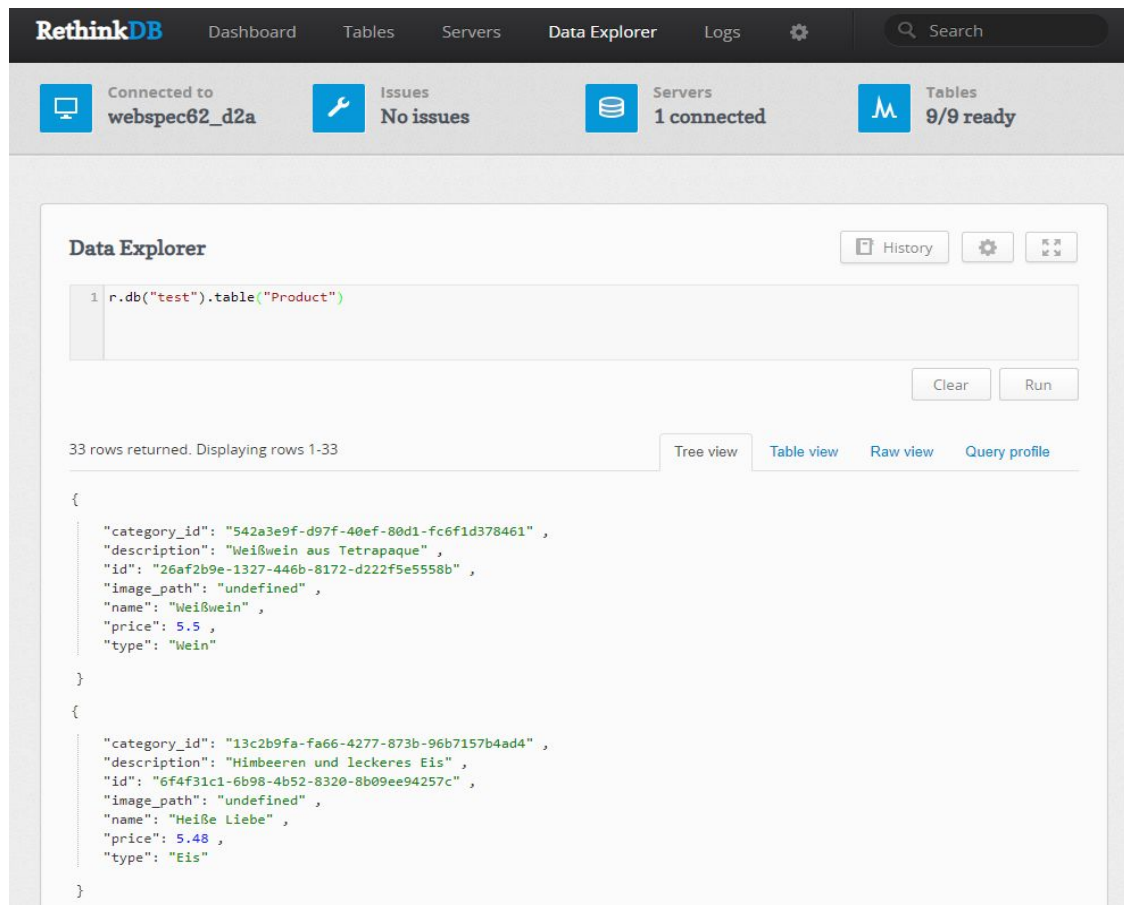
## Vorgehen

Zu Beginn des Sommersemesters 2019 lag der Fokus auf der Themenfindung für das erste große Projekt für die Medianight der HdM. In einer Gruppe aus ca. 15 Studenten wurden in der Lernwelt Ideen gebrainstormt bis sich zwei Anwendungen und deren Teams herauskristallisierten. Zunächst war die Ausarbeitung der Idee von entscheidender Rolle, um das Projekt bestmöglich umsetzen zu können. Außerdem erfolgte dabei die weitere Einteilung in die einzelnen Teams und deren Bereiche:

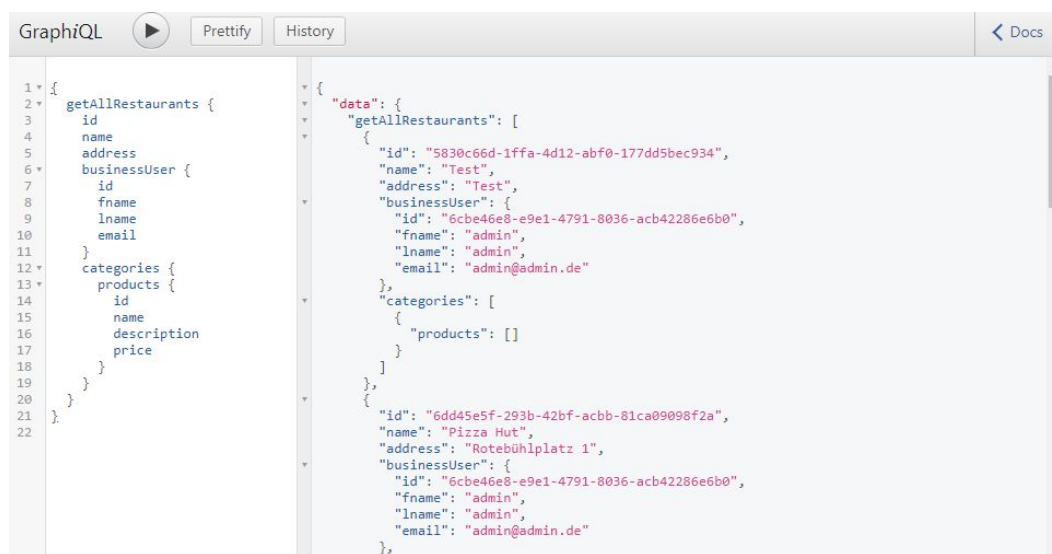
- Backend
- Frontend
- Design
- Projektmanagement

Bald darauf stand der Entschluss fest eine native *Android*-Applikation für den Kunden und für das Restaurant eine Webapplikation mit *JavaScript* und *React* zu realisieren. Im Backend sollte ein *Node.js*-Server mit *RethinkDB* aufgesetzt werden.

Um die direkte Datenbankabfragen zu Testzwecken machen zu können, wurde die RethinkDB Administrationsseite freigeschaltet:



Um die Implementierung sinnvoll testen zu können, verwendeten wir *GraphiQL*, ein Interface zum Testen der GraphQL-Abfragen durch direkte Eingabe.





## Fazit

Ein Semester-Projekt ist die Simulation des Berufslebens wie in einem IT-Sandkastens mit gleichzeitigem Management des Teams, des Projekts und der Ressourcen. Die Probleme gerade bei der Kommunikation zwischen den einzelnen Teammitgliedern und der Terminfindung mit absolut unterschiedlichen Zeitplänen machten sich relativ schnell bemerkbar. Alles in allem kann sich das Ergebnis sehen lassen, immerhin hat uns im Rahmen der MediaNight ein Familienmitglied der Bäckereikette “Der Beck” mit Sitz in Erlangen sein Interesse zur Zusammenarbeit bekundet. Leider verschwinden wir jetzt erst einmal alle ins Auslands- oder Praxissemester. Dennoch zeigt uns diese Anfrage, dass unser Konzept auf Nachfrage am Markt stoßen sollte.

- im nächsten Projekt ein kleineres Team
- früher anfangen/weniger prokrastinieren
- GraphQL lohnt sich

## Probleme

- Apollo-Client (zu spät implementiert) sowohl innerhalb der Android-App, als auch die Verbindung zum Server (mit der richtigen Adresse)
- Datenbank und Thinky (unausgereifte Dokumentation des Frameworks)
- DB Modell nochmal ändern, da auf rel. DB. angepasst
- Zeitmanagement
- fehlende Deadlines
- strikte Aufgabenverteilung
- Projektmanagement
- Konzept bereits auf dem Markt, Ausweitung hält sich im Rahmen
- aber: tatsächliche Interessenten bei der Media Night
- Usability noch nicht ausgereift
- Zu komplexes Datenmodell (und in SQL gedacht) am Anfang bereitet Schwierigkeiten

## Erfahrungen

- Funktionaler Prototyp mit Design
- Backend - Datenbankmodell & GraphQL Schema
- Frontend: native App mit Android Studio / Java
- Frontend: Web-Applikation mit JavaScript und React mit Material UI
- Einbau von Apollo in Android - connection zu GraphQL
- Planung Medianight: Plakat, Präsentation, Stand...
- Server aufsetzen

### Verbesserungsvorschläge bei der MediaNight

---

Usability

Platzzuweisung

Kunden kauften auch

Was du gekauft hast, gibt es hier auch

Menge eines Produkts

Haptisches Feedback bei Listenauswahl

Smoothes Scrollen in der Liste

Zurück-Taste muss noch richtig implementiert werden

Bestellstatus für Nutzer

User-Frontend als PWA