

# Technische Spezifikation

Vergissmeinnicht

Autoren: Yannik  
Benedikt  
Dmitry  
Wajdi

Letzte Änderung: 19.05.2020

Version: 1.0

---

## **Inhaltsverzeichnis**

Anwendungsüberblick	4
<b>Prozessüberblick</b>	<b>6</b>
Technischer Workflow	6
Nutzer-Workflow	8
Nutzer Workflow : Auftragnehmer	8
Nutzer Workflow : Auftraggeber	9
Technische Spezifikation SW	10
Komponentendiagramm	10
Klassendiagramm	11
Systeminfrastruktur	12
Datenbank	13
Beschreibung der Implementierung	15
Entwicklung	15
Webseitenentwicklung	16
Anmeldung, Suchverfahren	16
Mockups	17
Backend: Datenbank, Web-Server und GraphQL	21
MariaDB	21
Apollo Server und GraphQL API	21
NodeJS und express	22
ASP.NET Core	22
Arbeitsaufteilung	23

---

## **Abbildungsverzeichnis**

Abbildung 1: Use-Case Diagramm	5
Abbildung 2: grober technischer Workflow	6
Abbildung 3: feiner technischer Workflow	7
Abbildung 4: Ablaufdiagramm	8
Abbildung 5: Ablaufdiagramm	9
Abbildung 6: Komponentendiagramm	10
Abbildung 7: Klassendiagramm	11
Abbildung 8: Systeminfrastruktur	12
Abbildung 9: ERM-Modell	13
Abbildung 10: Logisches Datenmodell	14
Abbildung 11: Loginform. Hauptseite	17
Abbildung 12: Hauptseite der Suchergebnisse	18
Abbildung 13: Hauptseite	19
Abbildung 14: Mobile Applikation	20
Abbildung 15: MariaDB	21
Abbildung 16: GraphQL + Apollo	21
Abbildung 17: NodeJS + Express	22
Abbildung 18: ASP.NET Core	22

### **Copyright**

© Mohammad Abuosba

Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

### **Version Historie**

<i>Version:</i>	<i>Datum:</i>	<i>Verantwortlich</i>	<i>Änderung</i>
0.1	06.05.2020	Gruppe	Initiale Dokumenterstellung
0.2	10.05.2020	Yannik	Erstellung Komponentendiagramm
0.3	11.05.2020	Yannik	Erstellung Systeminfrastruktur
0.4	12.05.2020	Benedikt	Einfuegen Use Case
0.5	12.05.2020	Benedikt	Einfuegen Klassendiagramm
0.6	14.05.2020	Yannik	Überarbeitung Systeminfrastruktur Erstellung Tabelle Komponentendiagramm
0.7	16.05.2020	Wajdi , Dmitry	Erstellung Sequenzdiagramm
0.8	16.05.2020	Wajdi	Einfuegen Sequenzdiagramm
0.9	18.05.2020	Yannik	Erstellung 3.4.3 Backend
1.0	18.05.2020	Benedikt	App-Entwicklung
1.1	19.05.2020	Dmitry	Datenbank Activity Diagramm Mockup Web
1.2	19.05.2020	wajdi	Einfuegen Ablaufdiagramm

## 1 Anwendungsüberblick

Der Anwendungsüberblick wird durch ein Use-Case-Diagramm verdeutlicht:

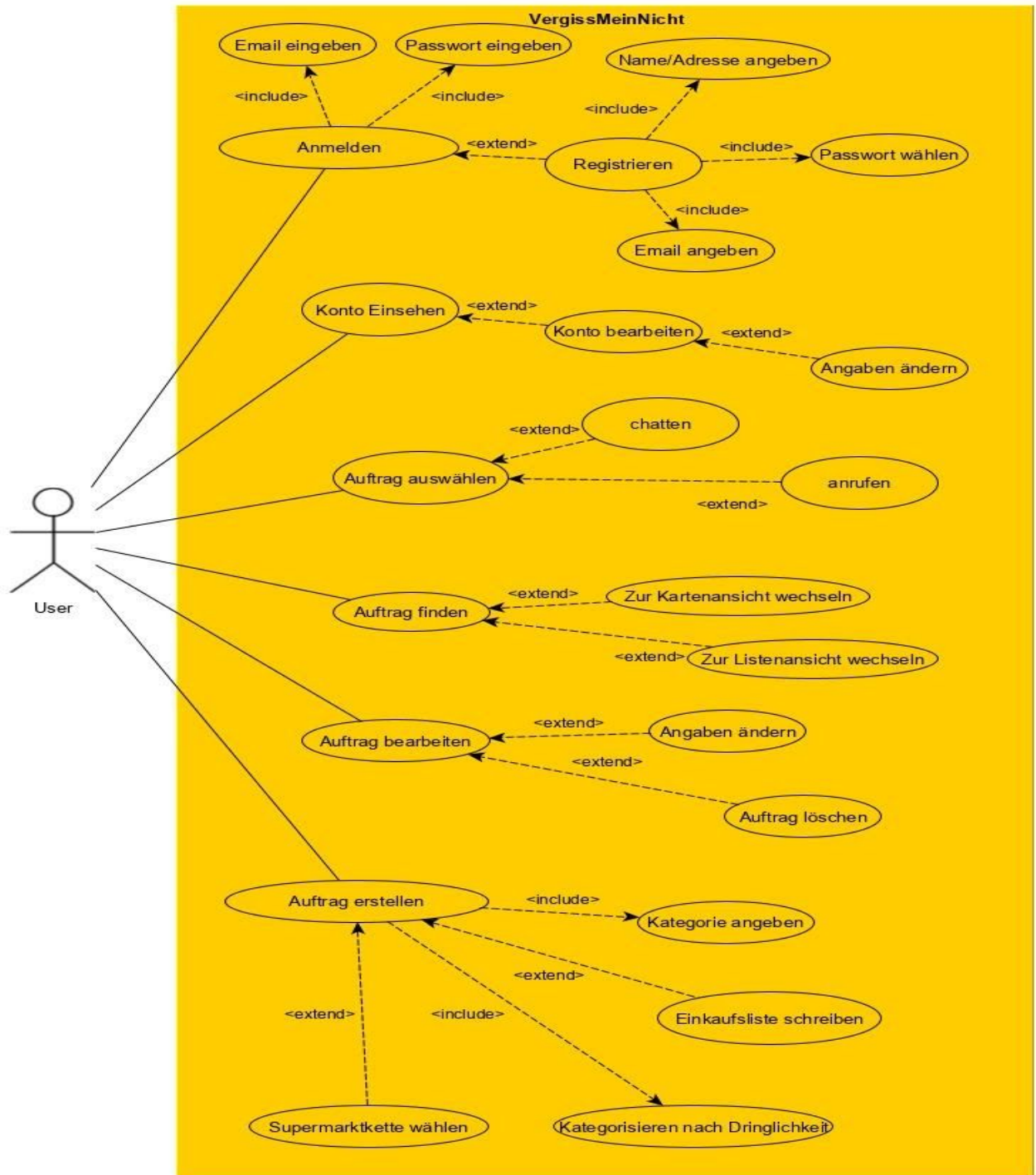


Abbildung 1: Use-Case Diagramm

## 2 Prozessüberblick

### 2.1 Technischer Workflow

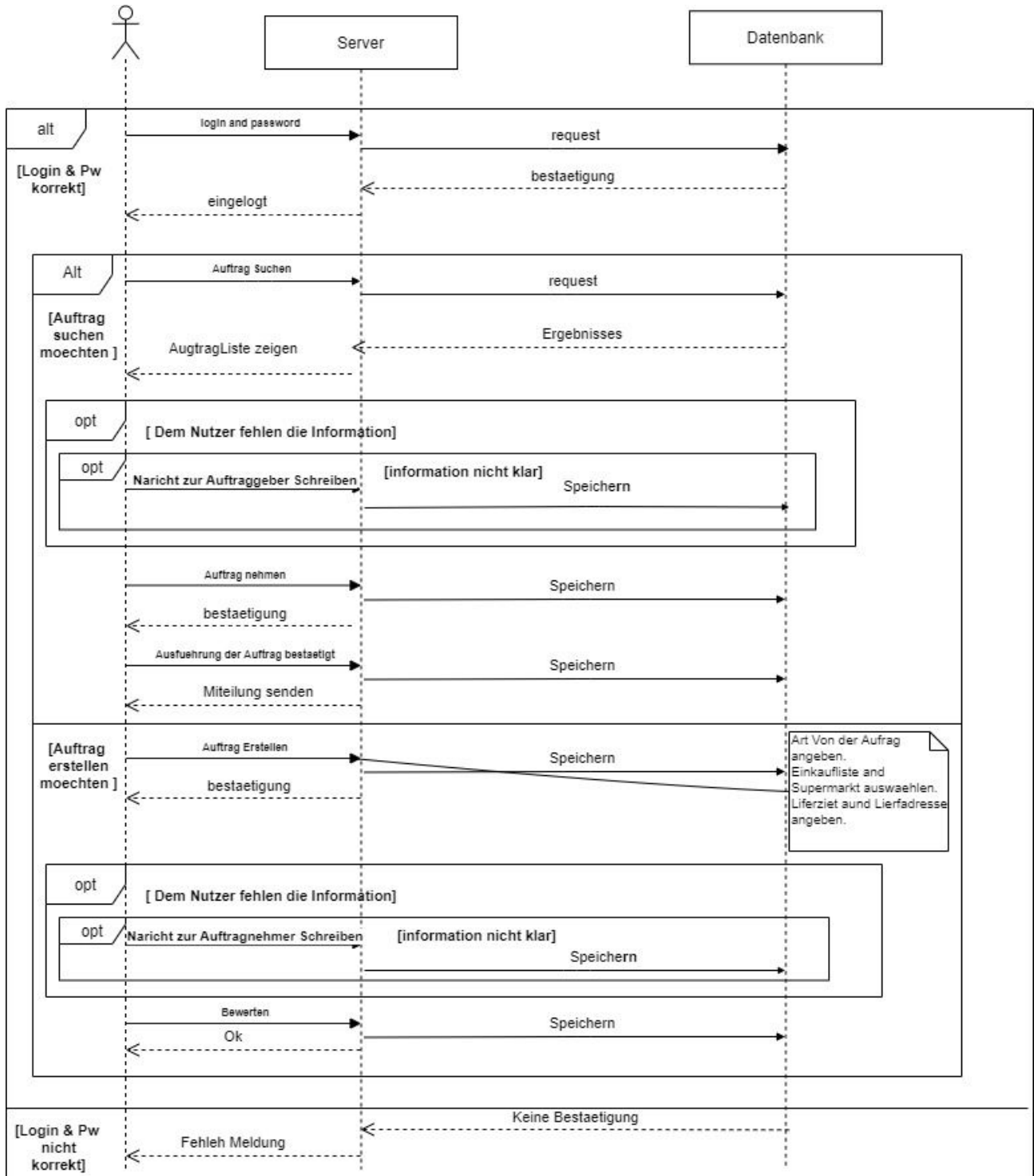


Abbildung 2: Technischer Workflow

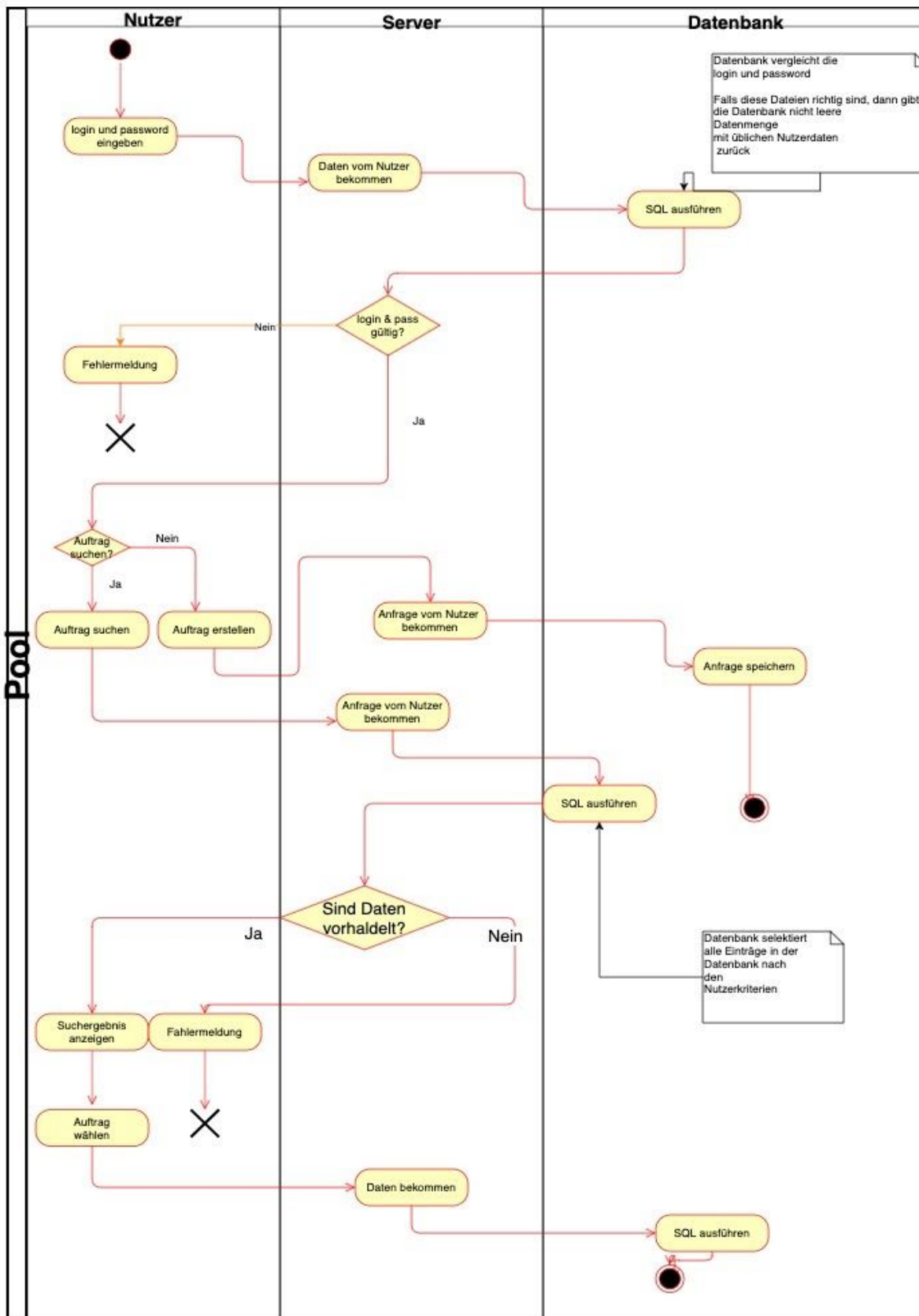


Abbildung 3: Aktivitätsdiagramm

## 2.2 Nutzer-Workflow

### 2.2.1 Nutzer-Workflow : Auftraggeber

Dieses Ablaufdiagramm beschreibt den Fall , wenn der Benutzer der Auftragnehmer ist. Um den Auftrag auszuführen , sollte der Benutzer die Schritte des Prozesses durchführen.

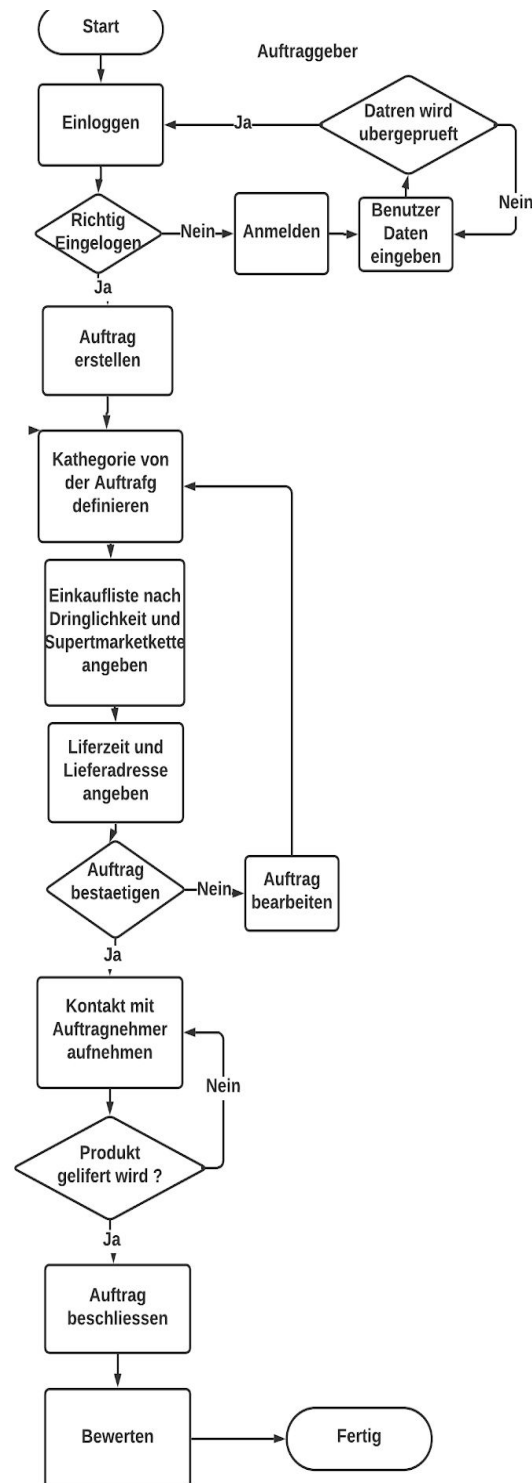


Abbildung 4: Ablaufdiagramm



## 2.2.2 Nutzer Workflow : Auftragnehmer

Wäre den Benutzer der Auftraggeber , dann gelten in diesem Fall die Schritte des folgenden Ablaufdiagramms:

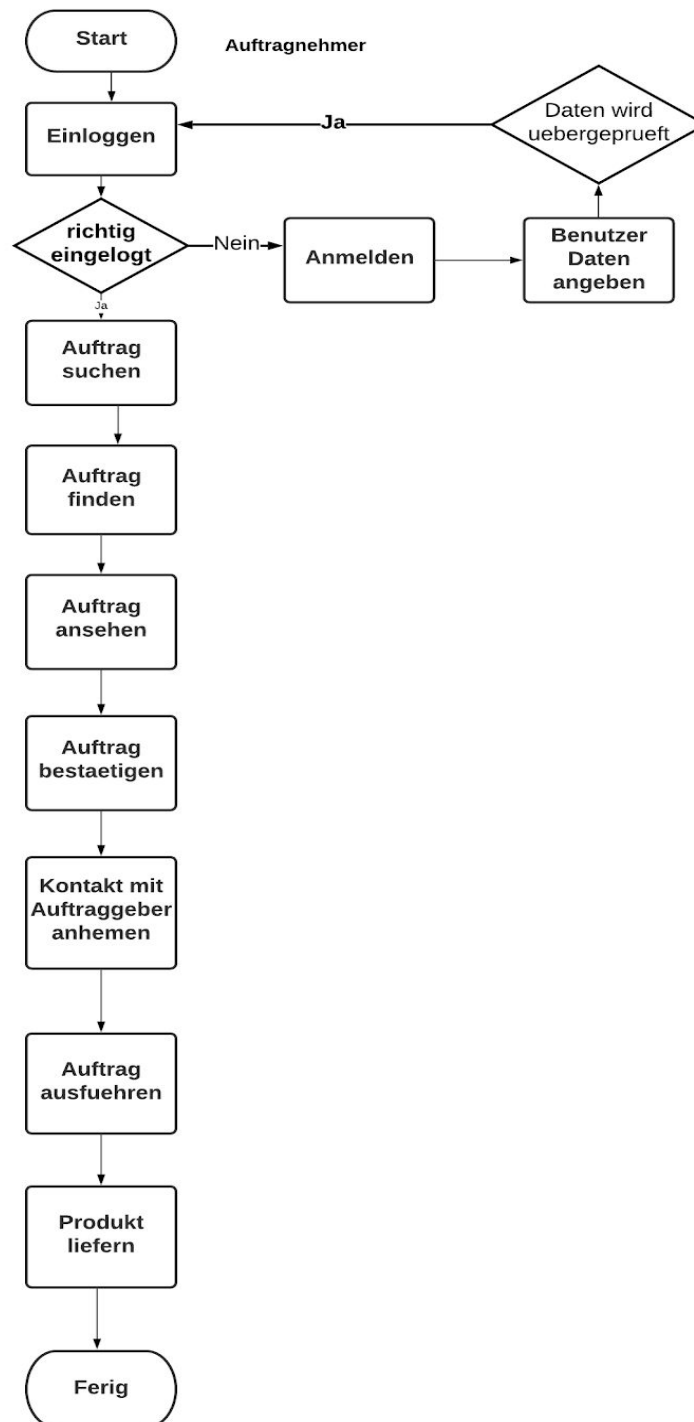


Abbildung 5 : Ablaufdiagramm

### 3 Technische Spezifikation SW

#### 3.1 Komponentendiagramm

Das Projekt wird in die folgenden Komponenten und Subkomponenten unterteilt:

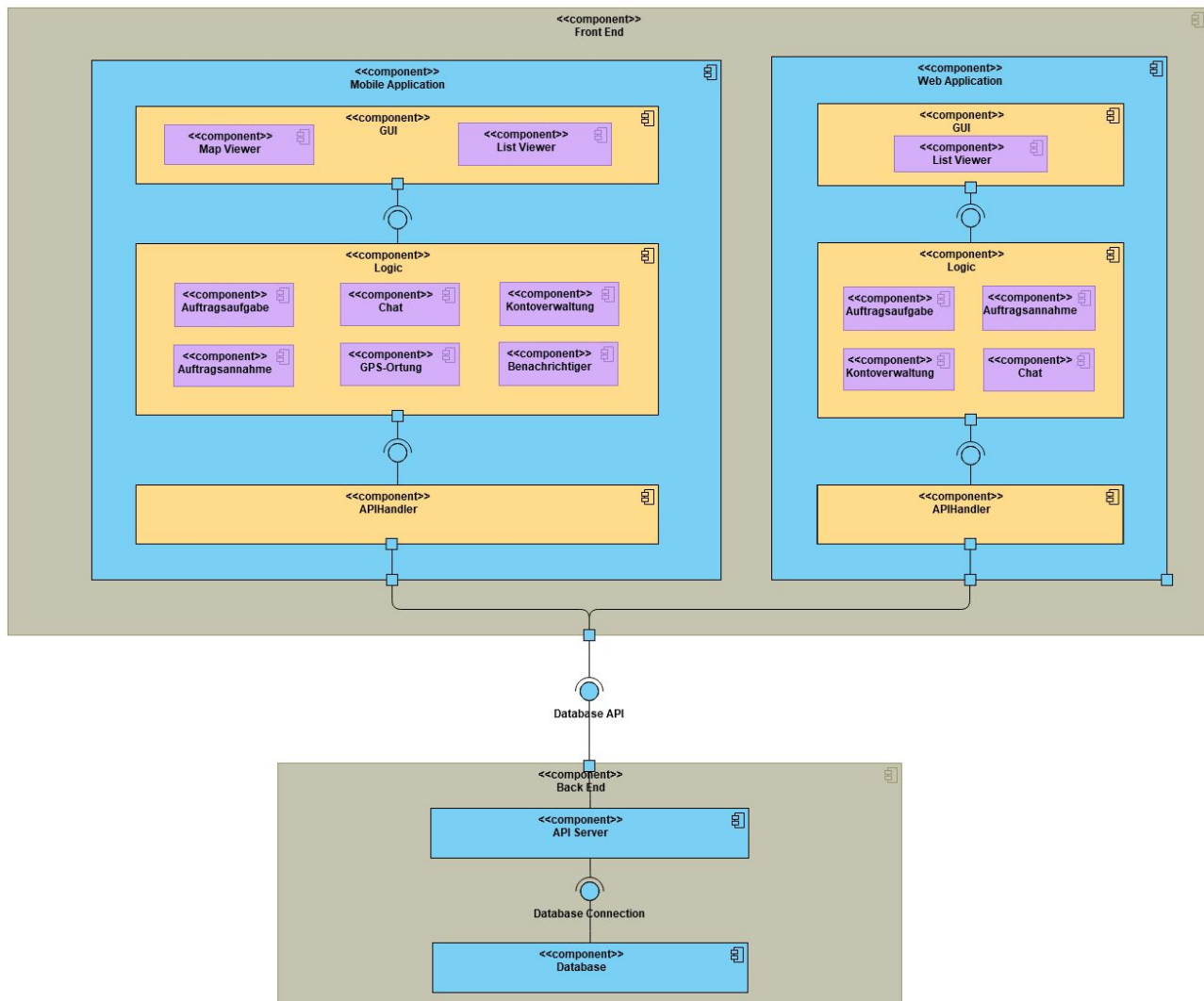


Abbildung 6: Komponentendiagramm

Komponente	Sub-Komponente	Erfasste Anforderung
GUI	List Viewer	FA - 2.1 Darstellung der Aufträge in Listenform
	Map Viewer	FA - 2.2 Darstellung der Aufträge in Kartenform
Logic	Kontoverwaltung	FA - 0 Konto
	Chat	FA - 4 Chat
	Auftragsaufgabe	FA - 1.1 Erstellung eines Auftrages
	Auftragsannahme	FA - 1.6 Bearbeitung eines Auftrages
	GPS-Ortung	FA - 1.4 Erfassung des Standorts der Benutzer
	Benachrichtiger	keine

## 3.2 Klassendiagramm

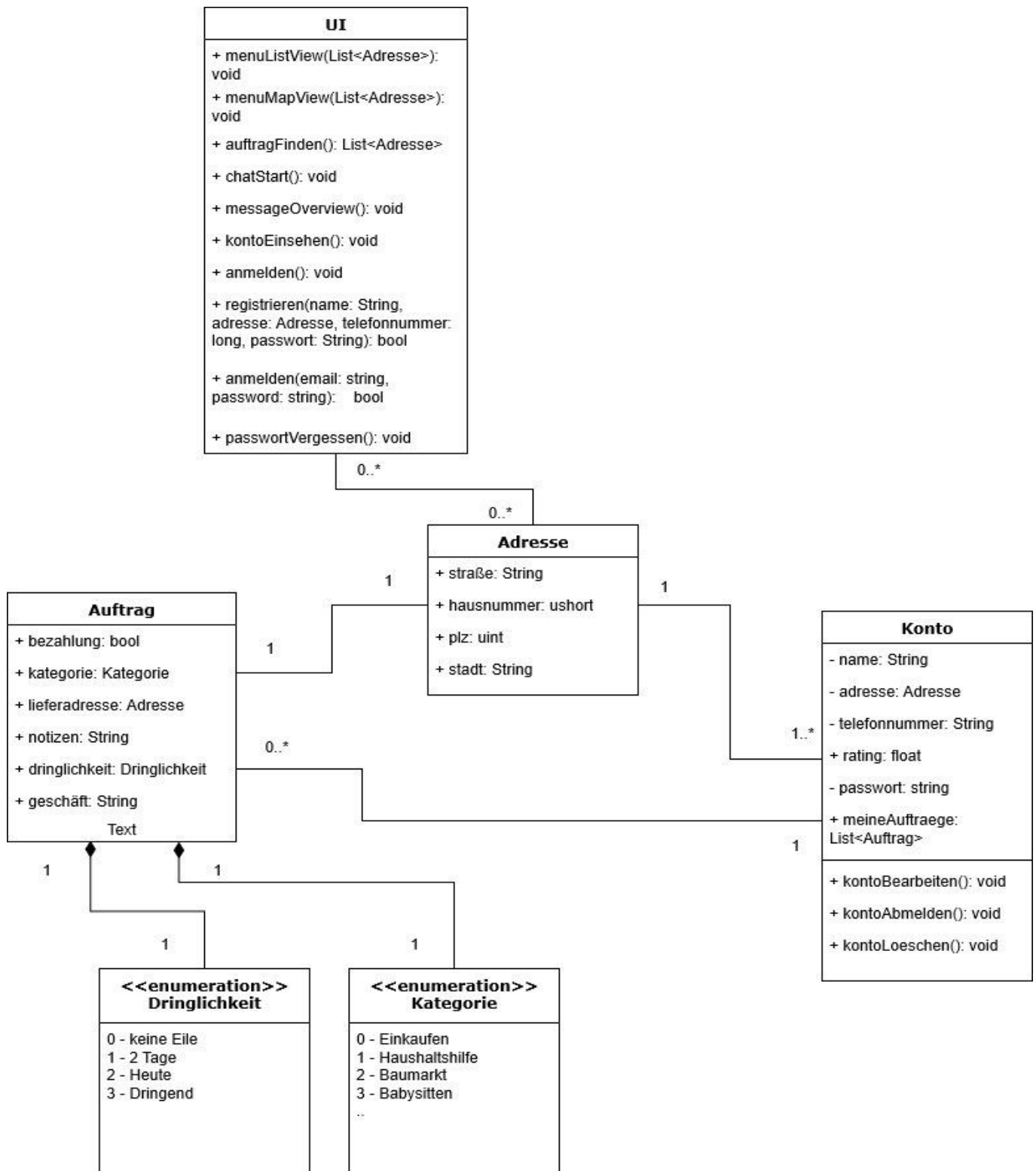


Abbildung 7: Klassendiagramm

### 3.3 Systeminfrastruktur

Wir verwenden als Backend einen Ubuntu-Server, welcher eine MySQL-Datenbank hosten wird. Ein API-Server wird Teile der Datenbank zur Verfügung stellen. Die App wird direkt mit der API kommunizieren, wohingegen die Website von einem Web-Server zur Verfügung gestellt wird, der mit dem API-Server kommuniziert.

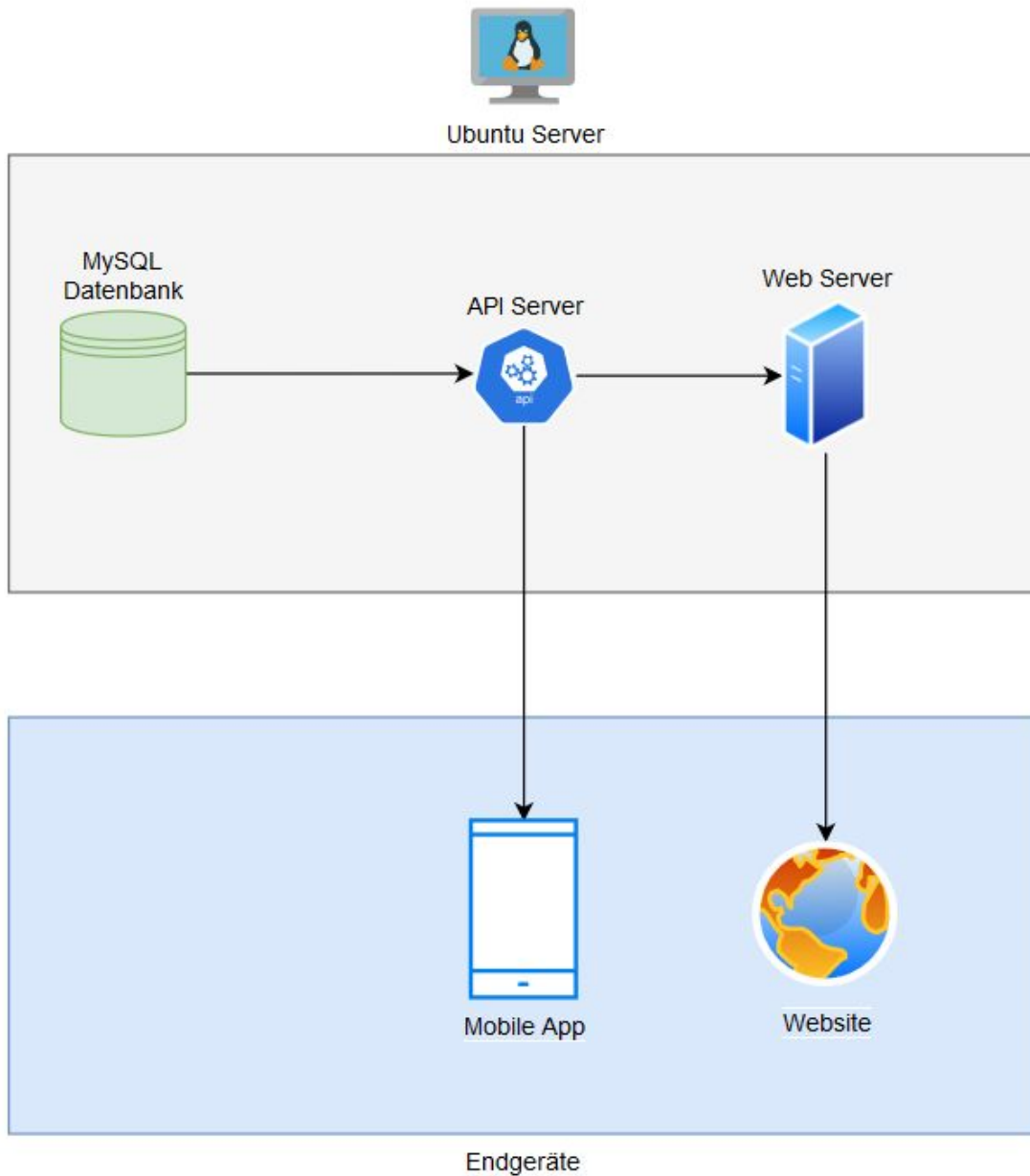


Abbildung 8: Systeminfrastruktur

### 3.4 Datenbank

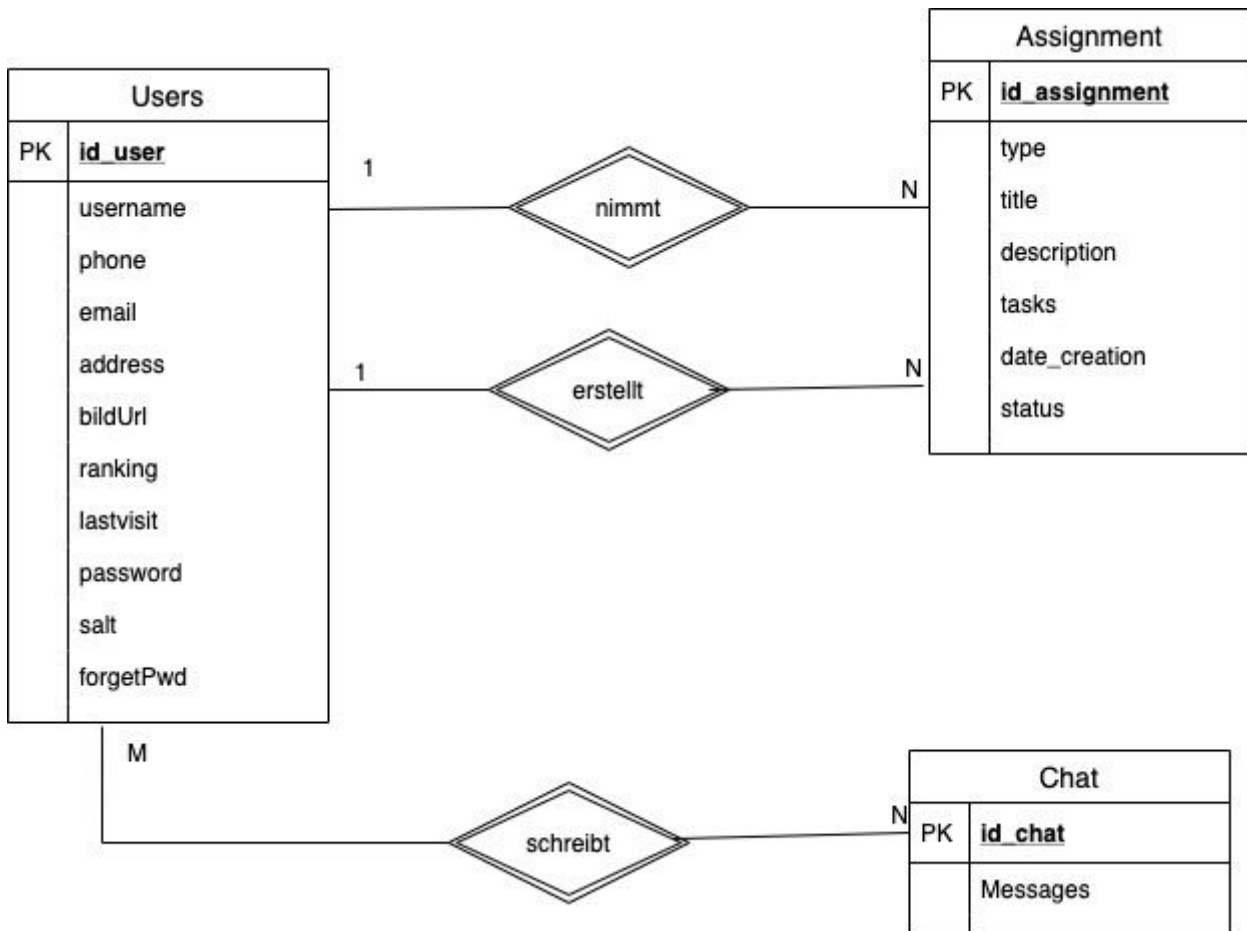


Abbildung 9: ERM-Modell

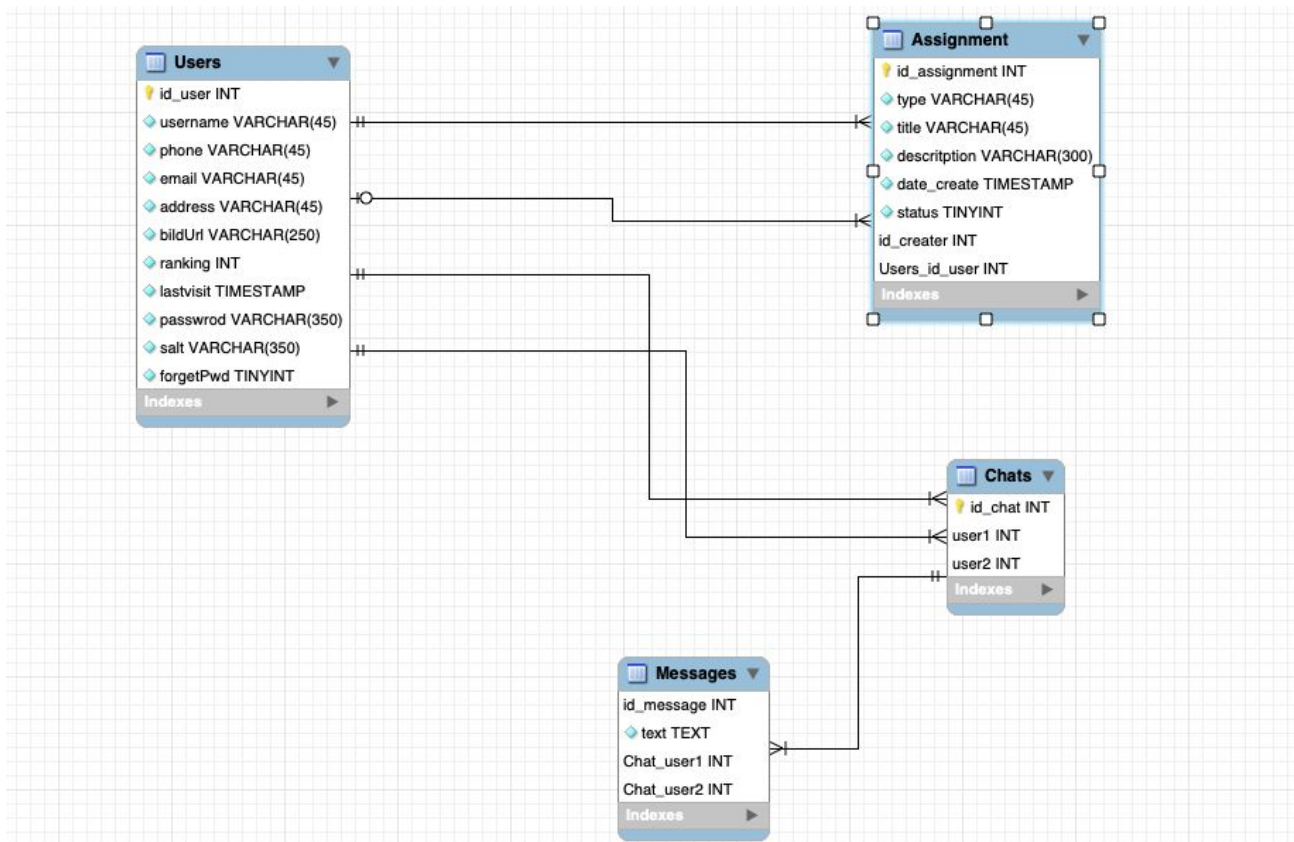


Abbildung 10: Logisches Datenmodell

## 3.5 Beschreibung der Implementierung

Detaillierte Beschreibung der notwendigen Entwicklungen/Änderungen pro Komponente zur Durchführung der gewünschten Funktionen für Website und Mobile Applikation, sowie Visualisierungen der Benutzeroberflächen für die verschiedenen Plattformen.

### 3.5.1 Entwicklung

#	Komponentendetails	Erforderliche Arbeiten
A1	GUI	Anmelden/Registrieren
A2	Anmeldung	Einloggen mit Email und persönlichem Passwort
A3	Registrierung	Erstanlegen eines Kontos in der Datenbank
A4	Auftrag finden	Suchen eines Auftrages über eine Listenansicht oder einer Karte
A5	Auftrag erstellen	Erstellen eines Hilfesuchts (Auftrag)
A6	Kontaktaufnahme	Auftragnehmer und -geber können in Kontakt miteinander treten

#### A1: GUI

- Mit Xamarin.Forms wird eine simple UI implementiert
- Am Anfang steht dabei eine Anmeldemaske, die ggf. weiter zu einem Registrierungsformular führt
- Oben rechts kann jederzeit auf die Kontodetails und Nachrichten zugegriffen werden
- Die Aufträge können entweder über eine Karte oder in Form einer Liste angesehen werden

#### A2: Anmeldung

- Wird immer am Anfang der Benutzung der App aufgerufen.
- Mit Eingabe der Email und des persönlichen Passwortes wird auf das Konto in der Datenbank zugegriffen.
- Die beiden Parameter Email & Passwort werden hier als Strings übergeben.
- Bei korrekten Eingaben wird der Zugang zum Konto und den weiteren Funktionen der App bereitgestellt.
- Bei falschen Eingaben gibt es eine Rückmeldung über die falsche Eingabe und der Zugang wird nicht gewährt.

#### A3: Registrierung

- Wird nur bei Neuanlegung eines Kontos aufgerufen.
- Erstanlegung eines Kontos in der Datenbank.
- Dabei gibt User Name, Adresse, Telefonnummer, Passwort und Email als Parameter der Funktion an.
- Bei gültigen Eingaben die noch nicht in der Datenbank vorhanden sind, wird ein neues Konto in der DB angelegt.
- Bei ungültigen Angaben gibt es eine Rückmeldung über den Fehler.

#### **A4: Auftrag finden**

- Aufträge können entweder in Form einer Liste oder in Form einer kleiner Pins auf einer Karte dargestellt und gefunden werden.
- Übergeben wird dabei nur die Position des User, anhand dessen die Aufträge aus der Datenbank nach Entfernung sortiert und ggf. auf der Karte dargestellt werden.
- Nähere Auftragsdetails werden durch Anklicken angezeigt.
- Dort können die Aufträge angenommen werden, wodurch sie dem Konto in der Datenbank zugeordnet werden.

#### **A5: Auftrag erstellen**

- Durch Anklicken eines "Plus"-Icons oben links kann ein neuer Auftrag erstellt werden.
- Dazu muss ein Formular ausgefüllt werden in denen die Übergabeparameter festgelegt werden.
- Dabei muss Dringlichkeit, Bezahlung (Ja/Nein), Lieferadresse, Uhrzeit, Kategorie angegeben werden.
- Optional können Geschäft und Notizen, wie z. B. die Einkaufsliste hinzugefügt werden
- Dieser wird in der Datenbank gespeichert, wo er für Suchen ab dann zur Verfügung steht.
- Bei falschen Eingaben gibt es eine Rückmeldung, welche Eingabe falsch war.
- Aufträge können zu jederzeit im Konto des Users unter "Meine Aufträge" bearbeitet werden.

#### **A6: Kontaktaufnahme**

- Sobald ein Auftrag einem Konto zugeordnet wird, können beide Parteien mit einem Chat in Kontakt treten, um weitere Details zu besprechen.
- Dieser Chat ist dann bei beiden Konten in den Nachrichten zu finden.
- Bei jeder neu erhaltenen Nachricht kriegt der User ein Push-Notification auf sein Smartphone.
- Durch die dem Konto zugeordnete Telefonnummer kann auch direkt über einen Sprachanruf miteinander kommuniziert werden.

#### **Webseitenentwicklung**

Die Webseite basiert auf das Technogiestack wie HTML, CSS, Angular, React, JavaScript. Das sind die gängigsten Technologien, die man heutzutage benutzt, um eine Website zu erstellen.

Die Website soll Login Form, Suchformular, Bewertungsform enthalten.

Alle Nutzer dürfen mit einer Suchformular suchen und die Suchausgabe durch die HTML-Elemente selektieren. Die Webseite hat eine Möglichkeit den Nutzerkonto zu steuern und die Aufträge zu bearbeiten.

#### **Anmeldung, Suchverfahren**

Die Webseite sendet die Anfrage zum Server, welcher darauf antwortet. Wenn die Antwort vom Server eine nicht leere Menge mit Daten hat, dann zeigt die Webseite das Ergebnis. sonst die Fehlermeldung



## 3.5.2 Mockups

Visualisierungen der späteren Benutzeroberfläche für Website und Mobile Applikation.

### 3.5.2.1 Website-Mockups

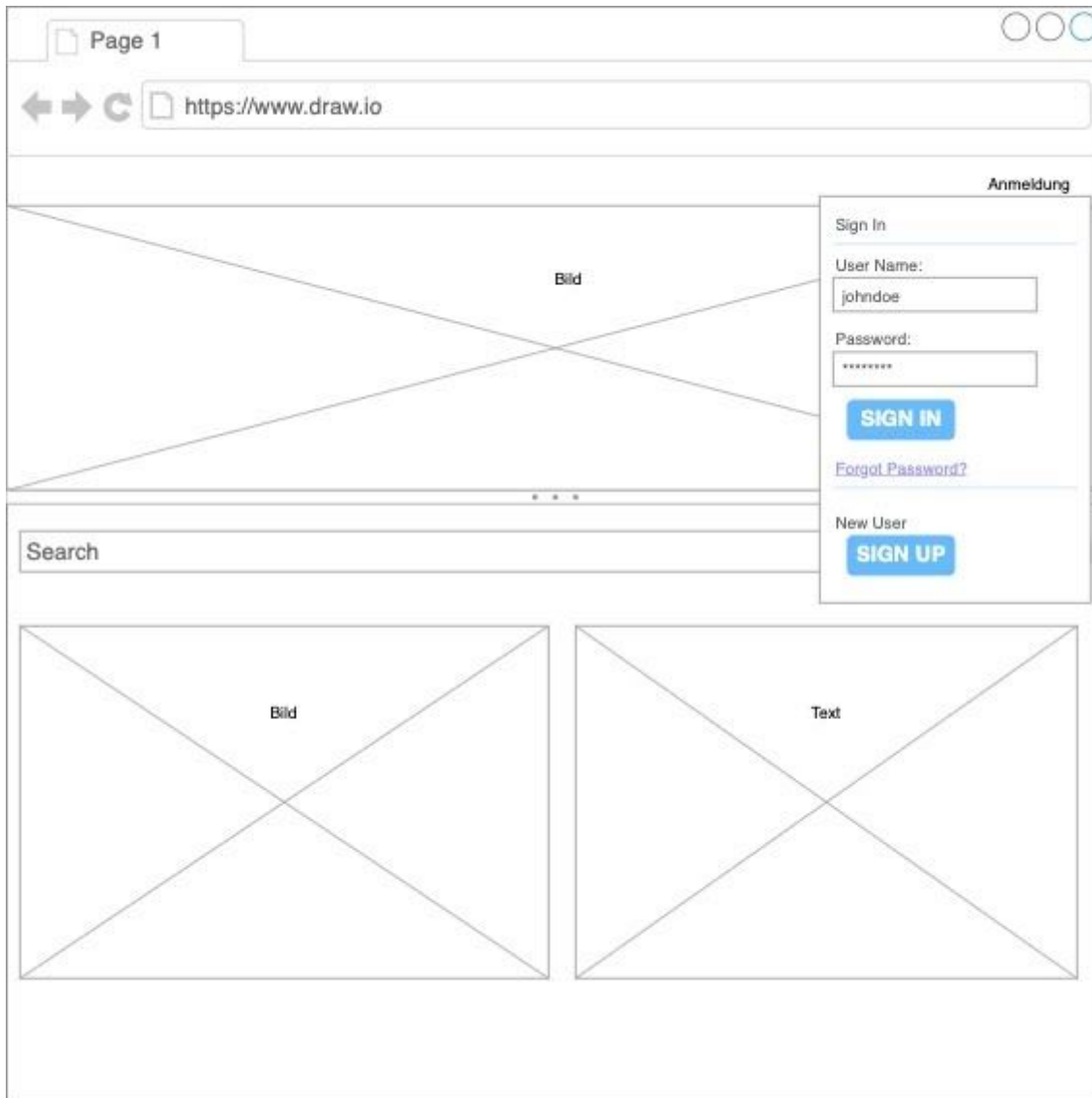


Abbildung 11: Loginform. Hauptseite

Der Nutzer sieht als die erste Seite eine Seite, wo man die Werbung mit Angebote anzeigen kann. Alle Nutzer können schon etwas suchen und die Angebote anschauen, sogar wenn man sich nicht eingeloggt hat, Man kann sich jederzeit registrieren und anmelden. Das letzte findet nur dann, wenn man gültige Benutzername und Kennwort eingegeben hat.

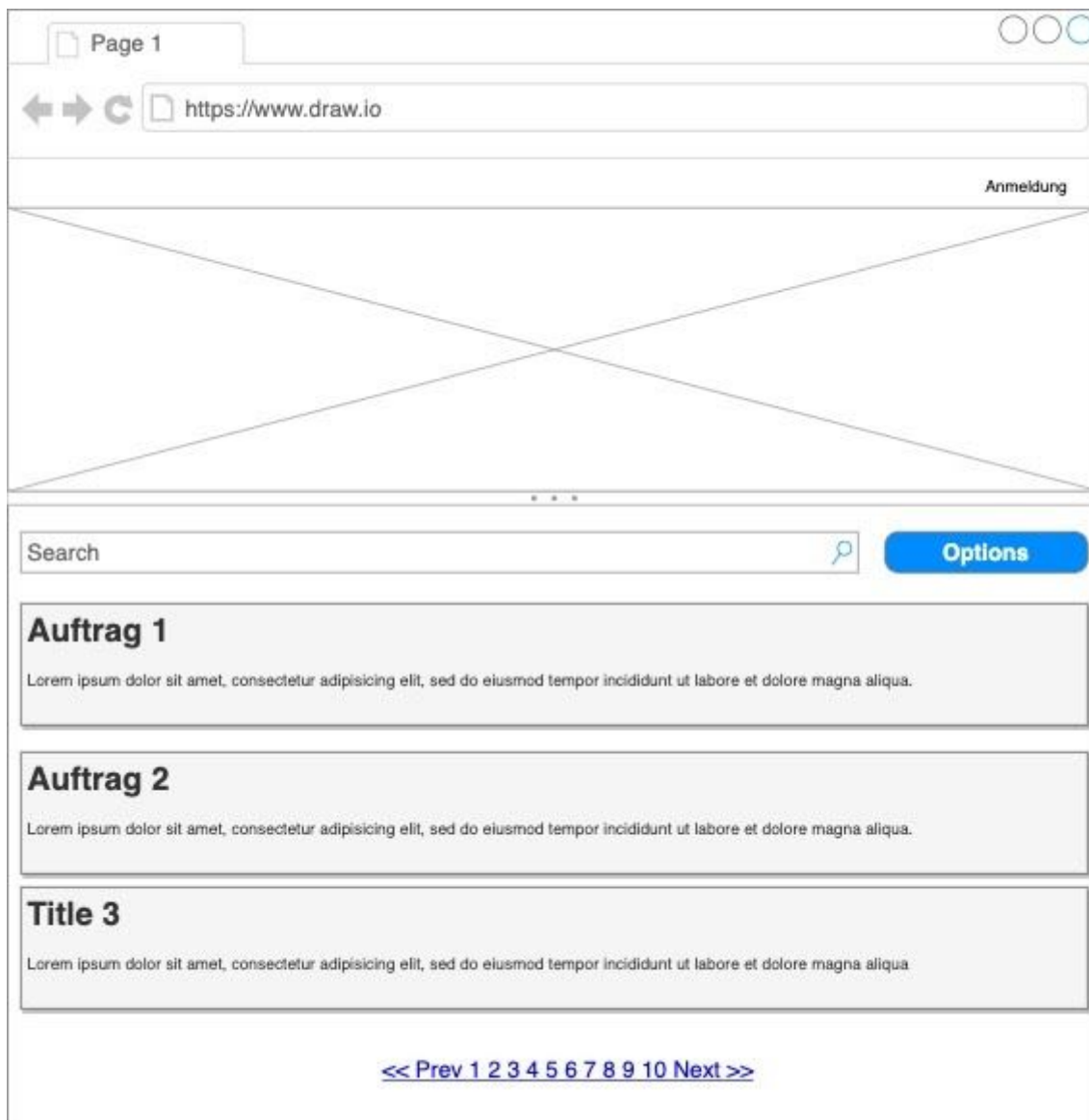


Abbildung 12: Hauptseite mit Suchergebnisse

Beim Suchen sieht man viele Aufträge, die man auflisten kann. Jeder Auftrag kann geklickt werden, um die ausführliche Information zu sehen.

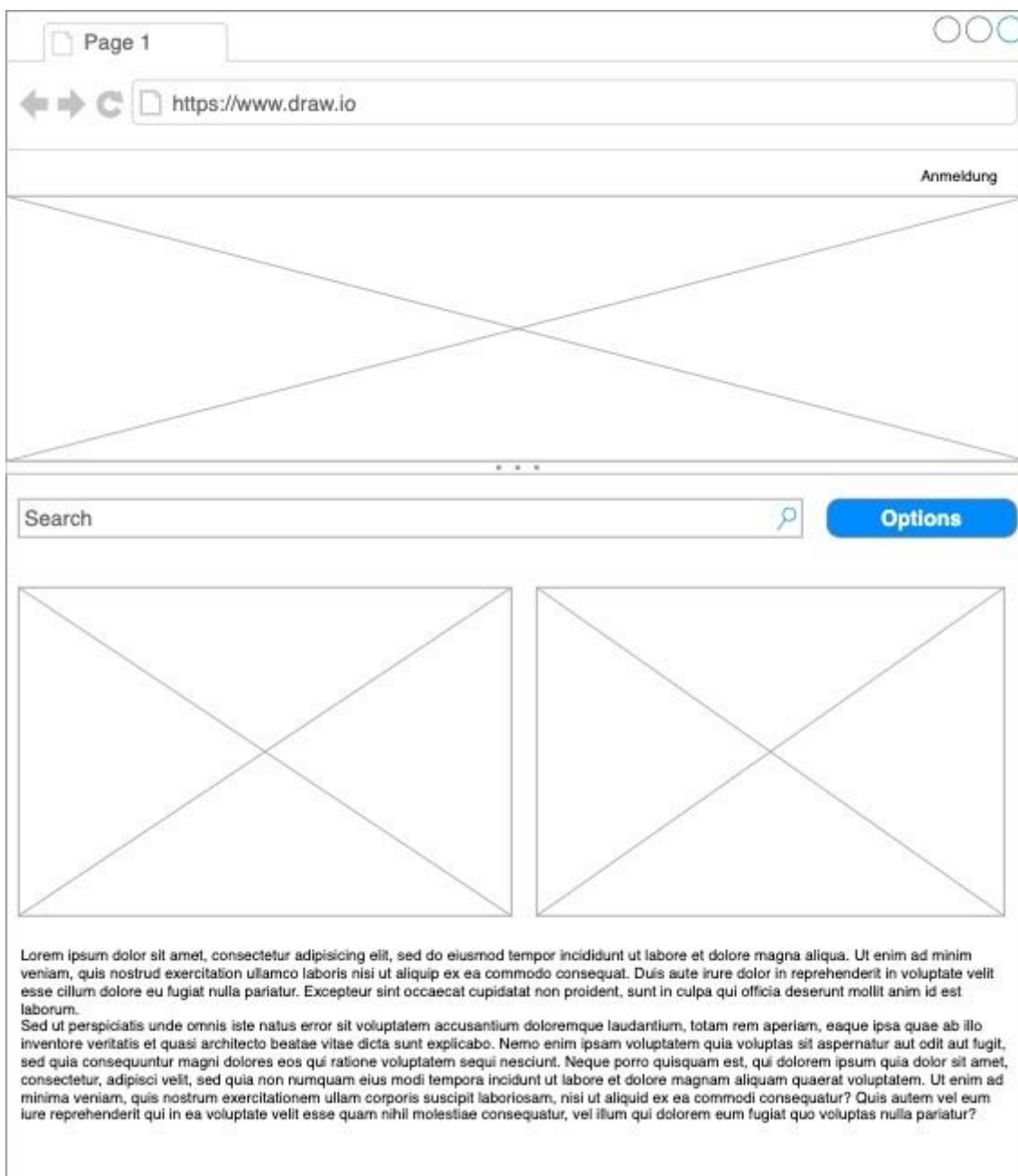


Abbildung 13: Hauptseite

Hier wird die Mockup für die Hauptseite dargestellt, worauf der Nutzer landet.

### 3.5.2.2 Mobile Applikation

GUI für die Auftragssuche:

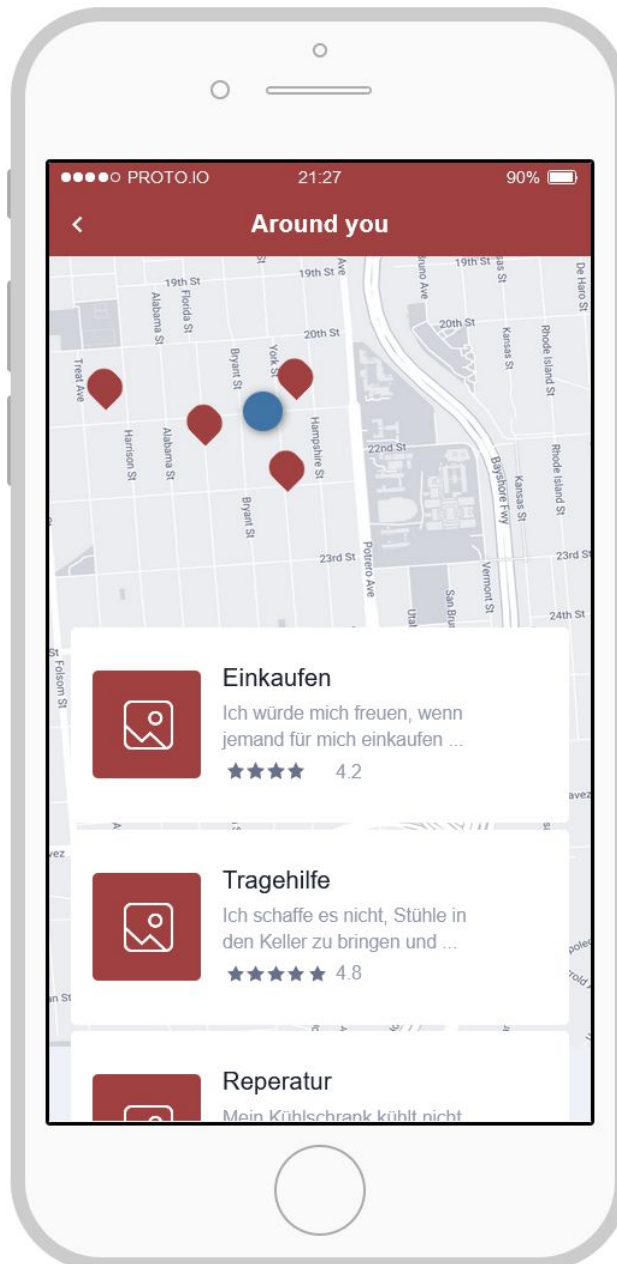


Abbildung 14: Mobile Applikation

### 3.5.3 Backend: Datenbank, Web-Server und GraphQL

Die Struktur des Backends kann Abbildung 6 entnommen werden.

#### 3.5.3.1 MariaDB

Zur Speicherung der betriebsnotwendigen Daten wird eine relationale Datenbank, die MariaDB, verwendet.



Abbildung 15: MariaDB

Bildquelle: <https://downloads.mariadb.org/devmedia/images/v2/MariaDB-Foundation-500x500.png?version=782492dca4047c851a115ac32c9ea427f9c79fce>

#### 3.5.3.2 Apollo Server und GraphQL API

Damit die Webseite und die App Zugriff auf die gespeicherten Daten der Datenbank erhalten, findet eine Kommunikation über *GraphQL* statt.

Die Verwendung der *GraphQL* Sprache hat mehrere Vorteile gegenüber der Verwendung der REST-Technologie. Sie erlaubt uns, einen einzigen Endpunkt anzusprechen und genau zu spezifizieren, welche Teile des angeforderten Objektes wir benötigen. Darüber hinaus ist *GraphQL* eine "streng typisierte" Sprache, wodurch Fehler verhindert werden.

Um die API logisch vom Webserver zu trennen, setzen wir hierfür einen *Apollo*-Server auf (theoretisch wäre auch eine Integration von *GrahQL* in den *express* Webserver denkbar).

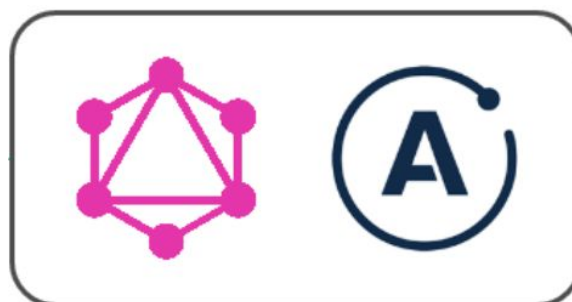


Abbildung 16: GraphQL + Apollo

Bildquelle: <https://romankudryashov.com/blog/2020/02/how-to-graphql/images/architecture.png>

### 3.5.3.3 NodeJS und express

NodeJS ist eine Laufzeitumgebung für JavaScript. Diese wird verwendet, damit der *express*-Server (aber auch der *Apollo*-Server) funktionieren können.

Wir verwenden express als Webserver, um die anzufertigende Webseite zu hosten. Dafür bietet express als JavaScript-Framework weitreichende Funktionalitäten.



Abbildung 17: NodeJS + express

Bildquelle: [https://miro.medium.com/max/365/1\\*Jr3NFSKTfQWRUyjbIBSKeg.png](https://miro.medium.com/max/365/1*Jr3NFSKTfQWRUyjbIBSKeg.png)

### 3.5.3.4 ASP.NET Core

Alternativ zu den oben genannten Technologien wäre auch die Verwendung des ASP.NET Web Frameworks denkbar, damit wir Front- und Backend übergreifend C# zur Programmierung verwenden können.

Mit diesem ist es möglich, sowohl die API als auch die Webseite zu programmieren.

Da uns hierzu die Erfahrung fehlt, werden beide Varianten getestet und sich im Laufe des Projekts für eine der beiden Varianten entschieden.



Abbildung 18: ASP.NET Core

Bildquelle: <https://static.gunnarpeipman.com/wp-content/uploads/2019/10/aspnet-core-featured.png>

## 4 Arbeitsaufteilung

Aufgabe/Kapitel	Autor	Status
Anwendungsüberblick	Benedikt	✓
Prozessüberblick	-	-
Nutzer-Workflow	Wajdi	✓
Technischer Workflow	Dmitry, Wajdi	✓
Technische Spezifikation SW	-	-
Komponentendiagramm	Yannik	✓
Klassendiagramm	Benedikt	✓
Systeminfrastruktur	Yannik	✓
Datenbank	Dmitry	✓
Beschreibung der Implementierung	-	-
Entwicklung	Dmitry, Benedikt	✓
Mockups	Dmitry, Benedikt	✓
Backend	Yannik	✓