

Technische Spezifikation

Vergissmeinnicht

Autoren: Yannik
Benedikt
Dmitry
Wajdi

Letzte Änderung: 18.05.2020

Version: 0.4

Inhaltsverzeichnis

Anwendungsüberblick	4
Prozessüberblick	6
Technischer Workflow	6
Nutzer-Workflow	7
Technische Spezifikation SW	8
Überblick Komponenten	8
Systeminfrastruktur	10
Datenbank	11
Beschreibung der Implementierung	12
Webseiten-Entwicklung	12
App-Entwicklung	14
Backend: Datenbank, Web-Server und GraphQL	16
MariaDB	16
Apollo Server und GraphQL API	16
NodeJS und Express	16
Arbeitsaufteilung	17
Offene Fragen	18

Abbildungsverzeichnis

Abbildung 1: Fachlicher Workflow	5
Abbildung 2: grober technischer Workflow	6
Abbildung 3: feiner technischer Workflow	7
Abbildung 4: Komponentendiagramm	8
Abbildung 5: Klassendiagramm	9
Abbildung 6: Datenmodell	10
Abbildung 7: Sequenzdiagramm	11

Copyright

© Mohammad Abuosba

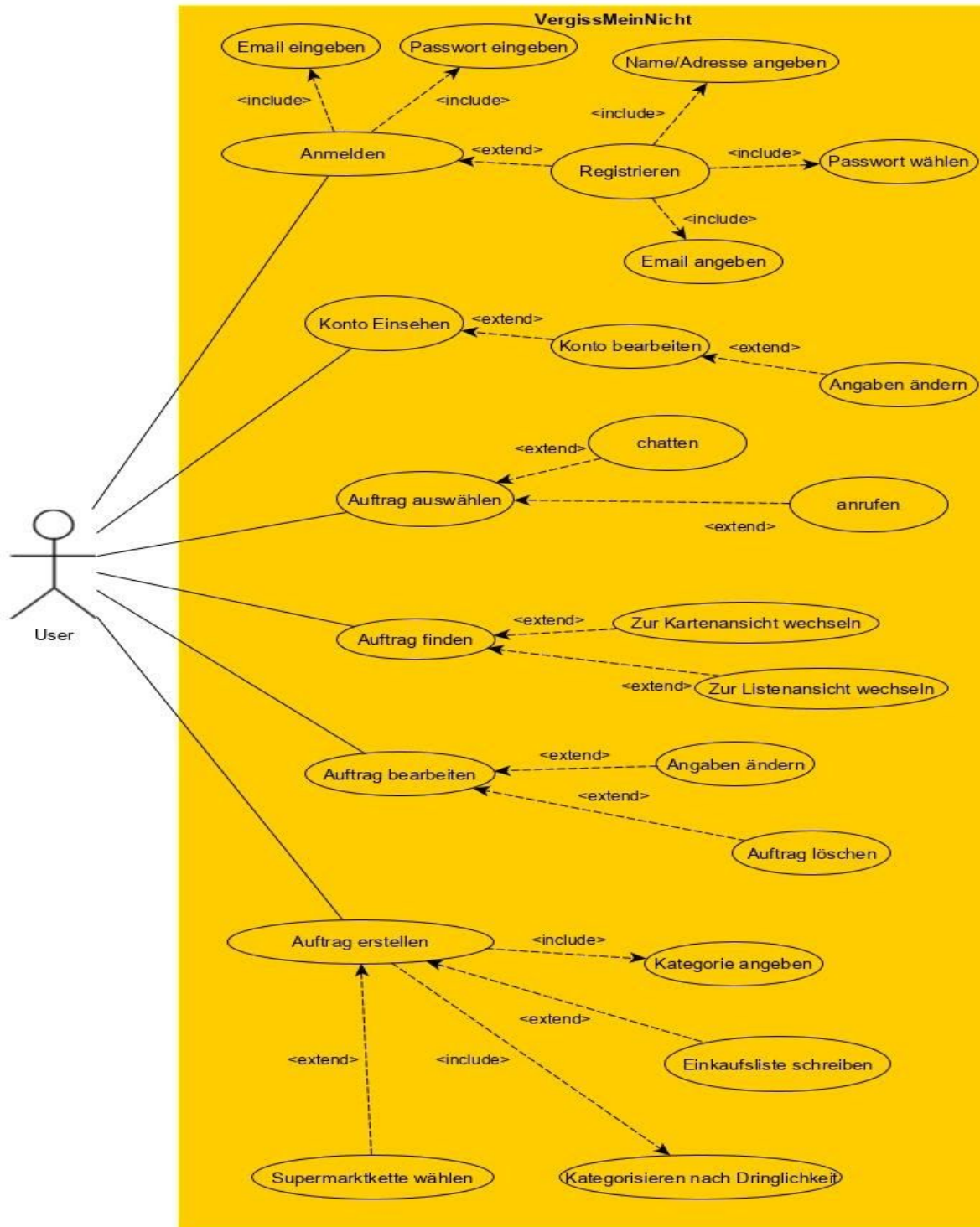
Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

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	Einfuegen Activity Diagramm
0.8	18.05.2020	Yannik	Erstellung 3.4.3 Backend
0.9	18.05.2020	Benedikt	App-Entwicklung
1.0			
1.1			

1 Anwendungsüberblick

Use-Case-Diagramm:



2 Prozessüberblick

Hier soll der übergeordnete Überblick des gesamten Prozesses graphisch dargestellt werden.
Dabei können zwei Ebenen unterschieden werden

2.1 Technischer Workflow

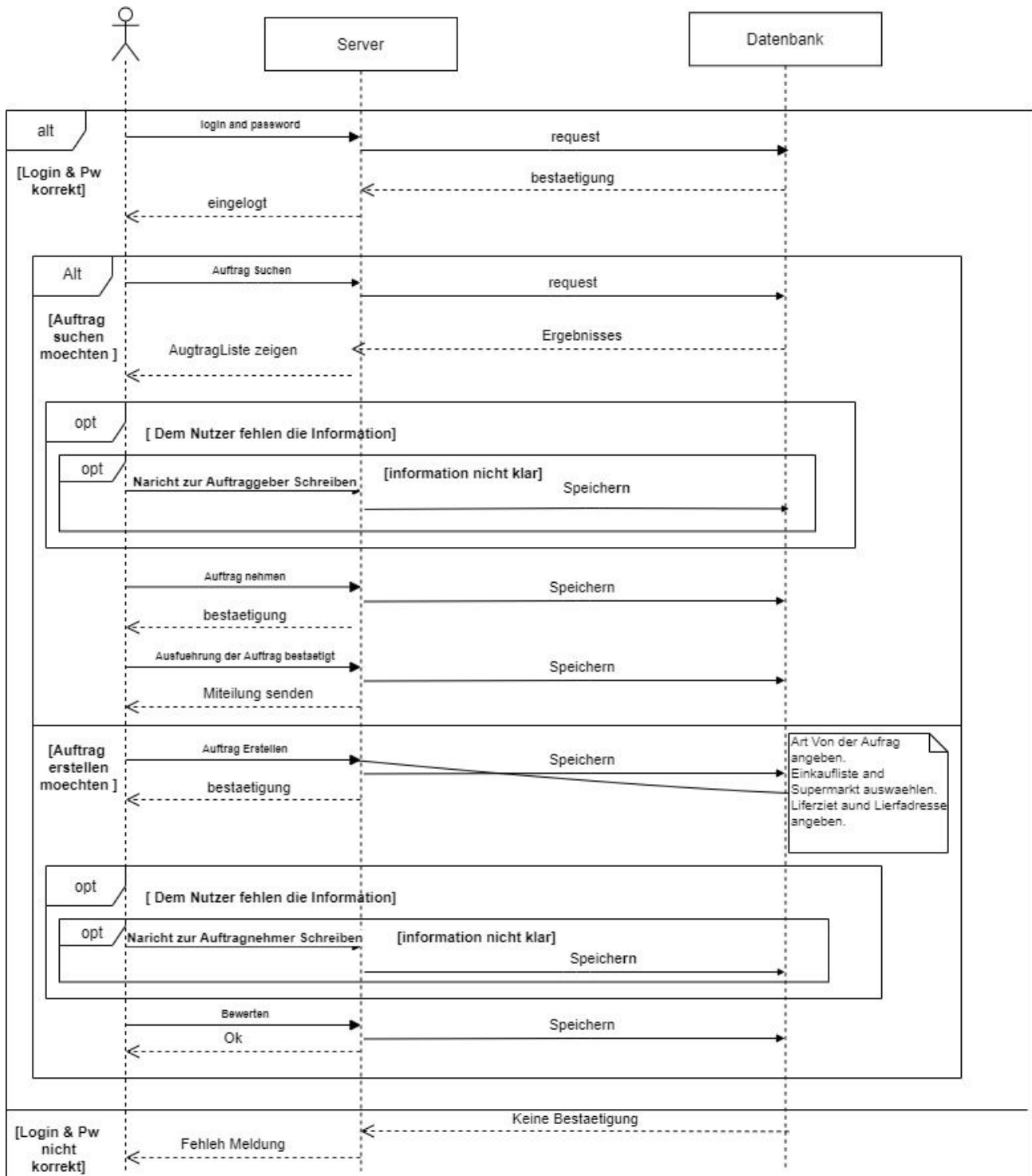
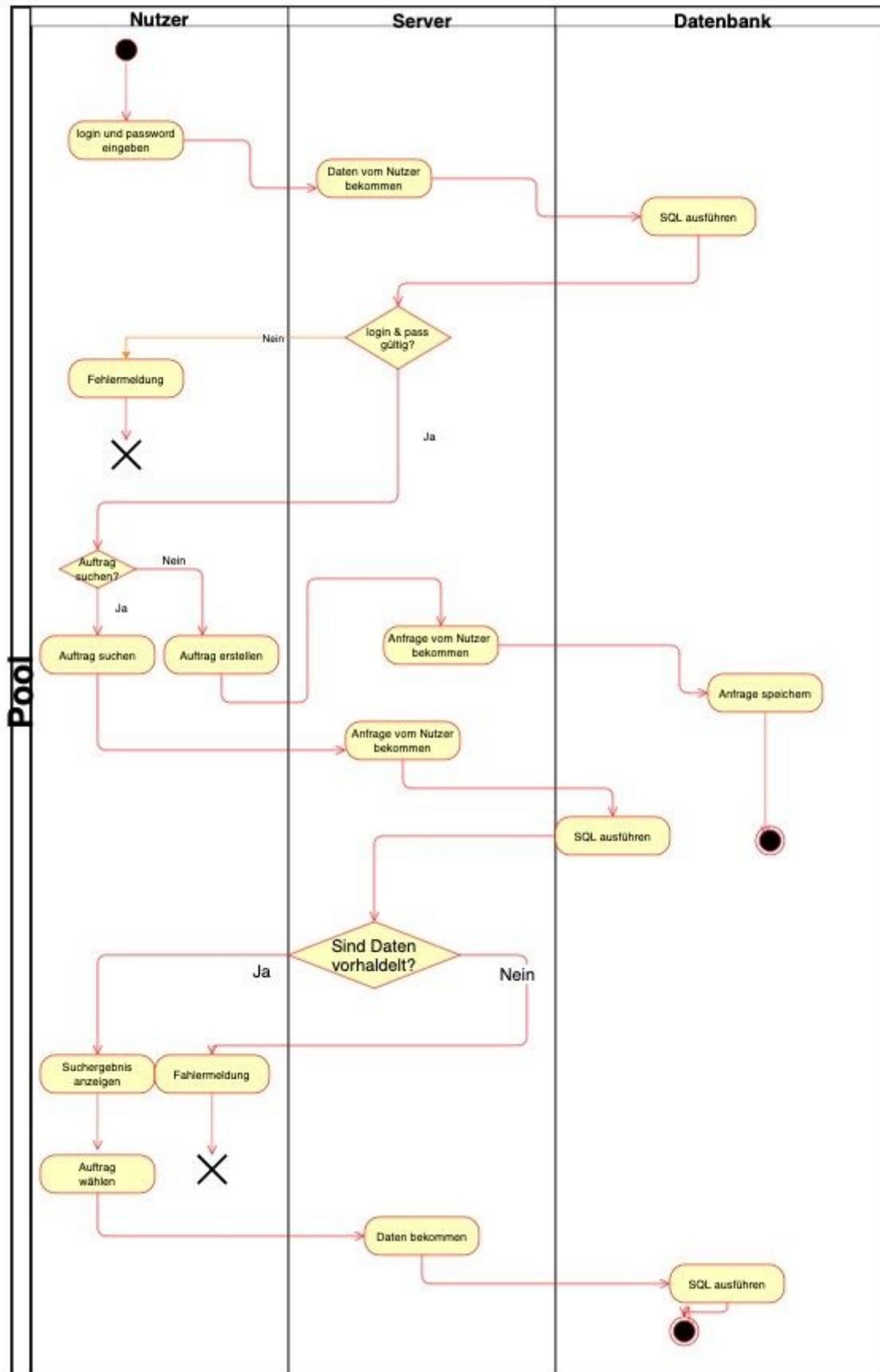


Abbildung 2: Technischer Workflow



2.2 Nutzer-Workflow

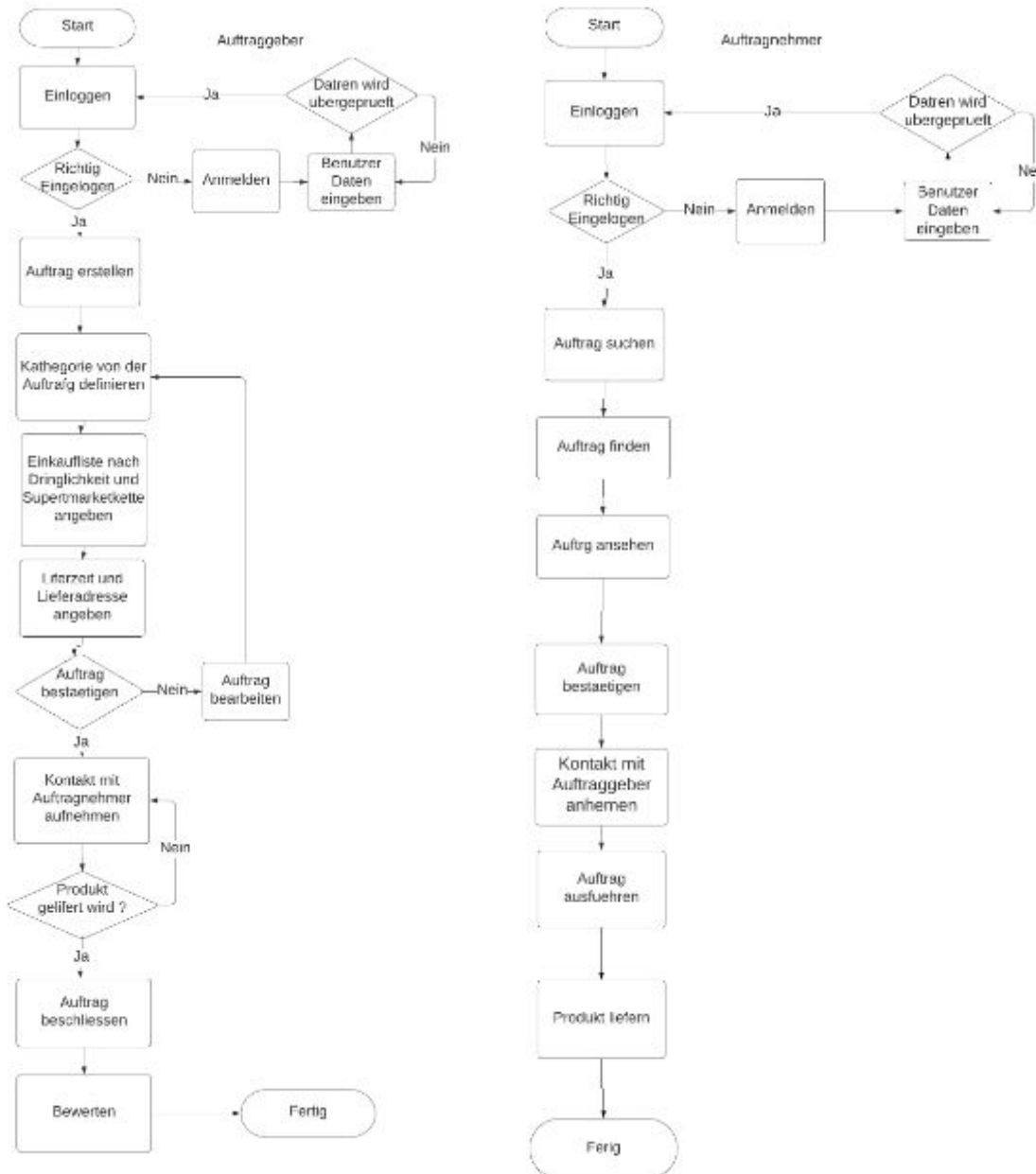


Abbildung 2: grober technischer Workflow

Abbildung 3: finer technischer Workflow

3 Technische Spezifikation SW

3.1 Überblick Komponenten

Das Projekt wird in die folgenden Komponenten und Subkomponenten unterteilt:

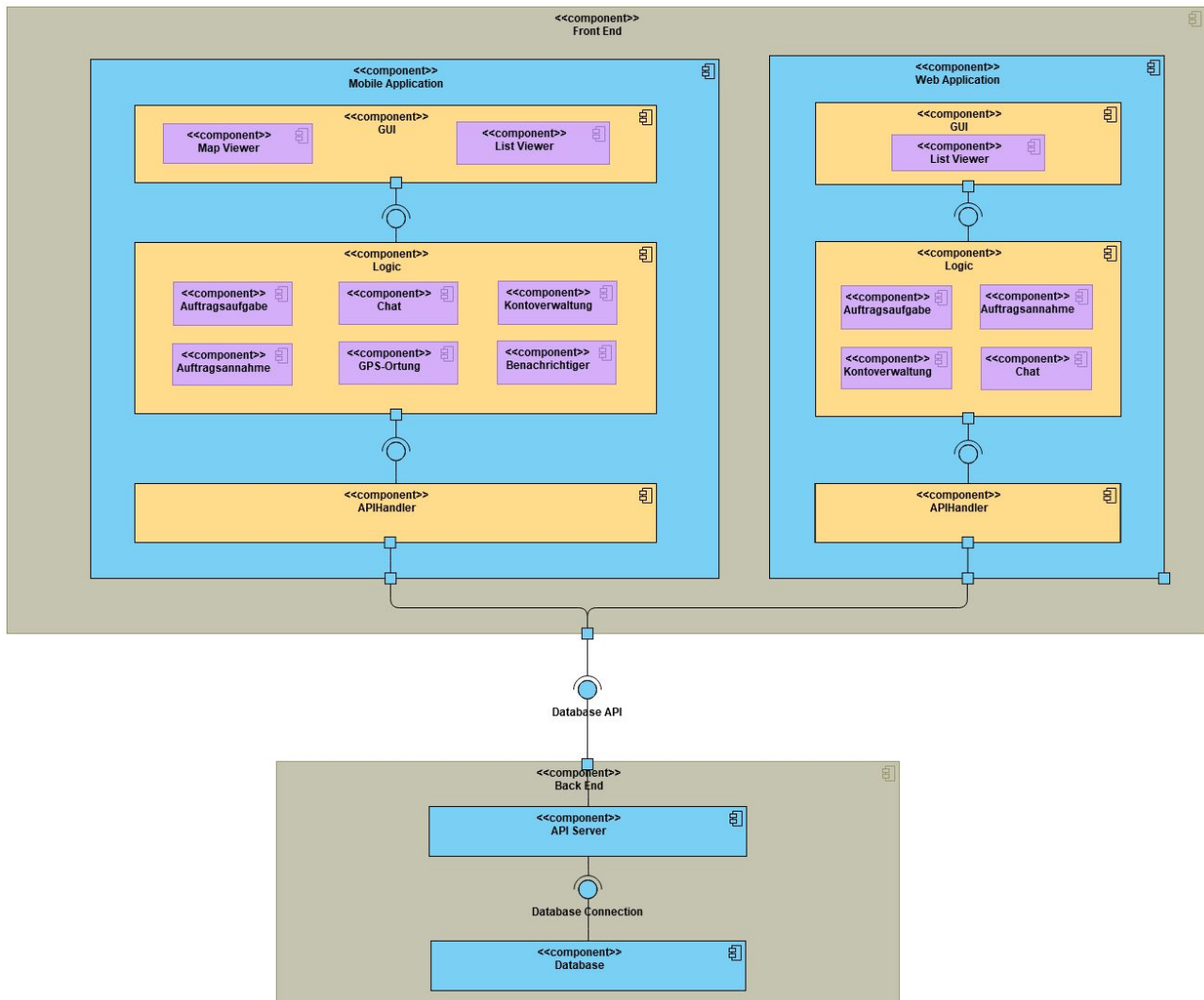


Abbildung 4: Komponentendiagramm

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

3.1.1. Überblick der Klassen:

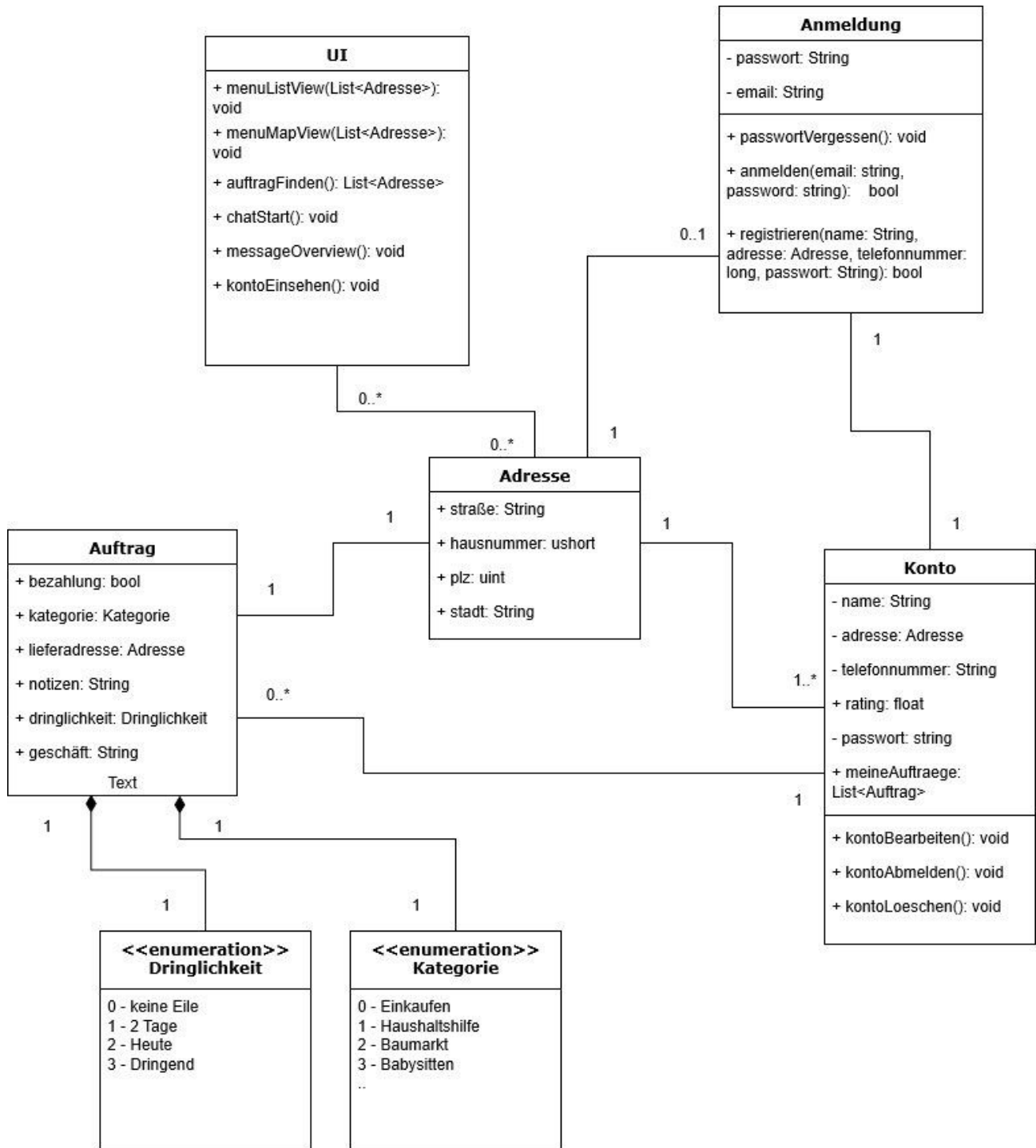


Abbildung 5: Klassendiagramm

3.2 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.

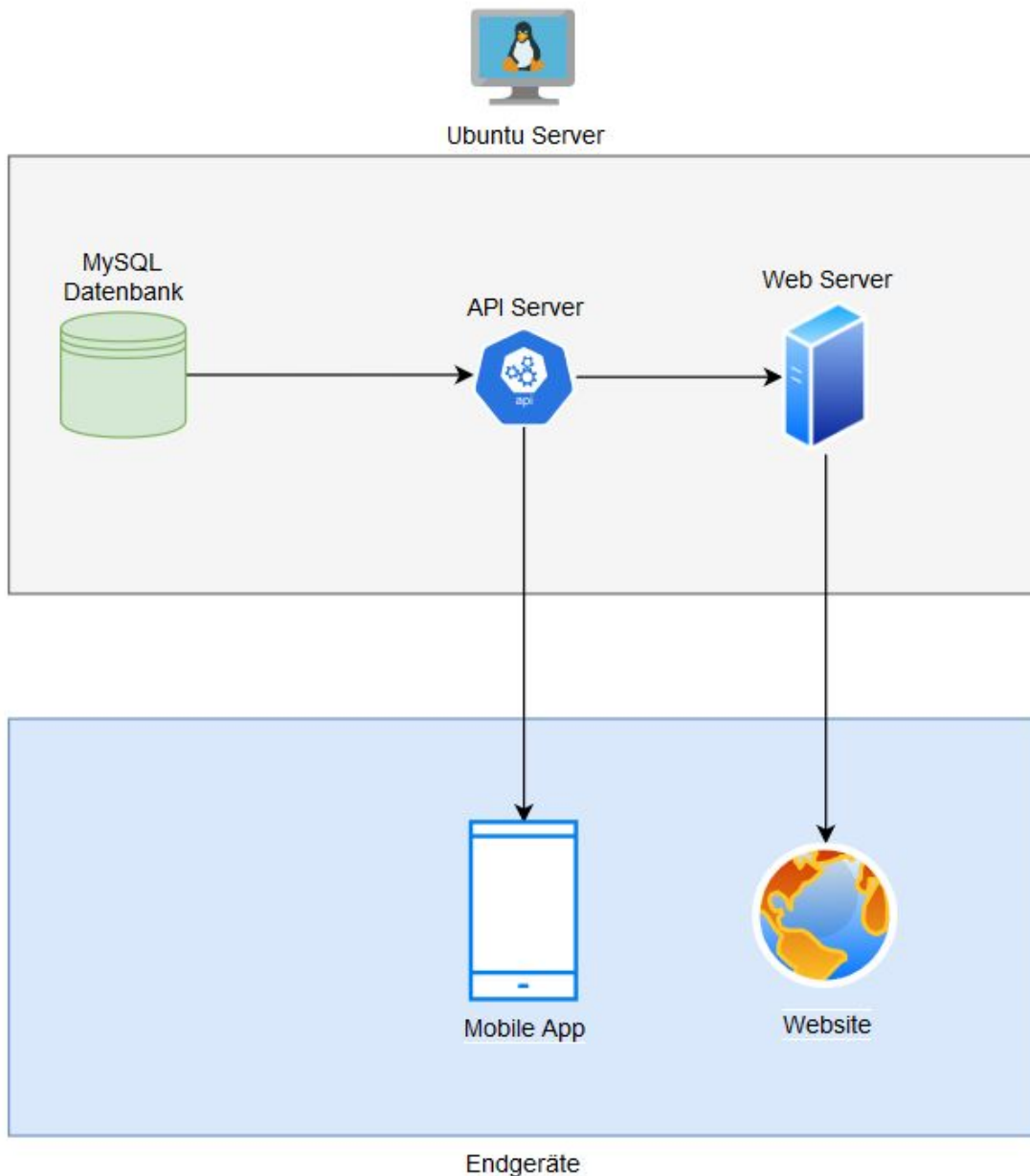
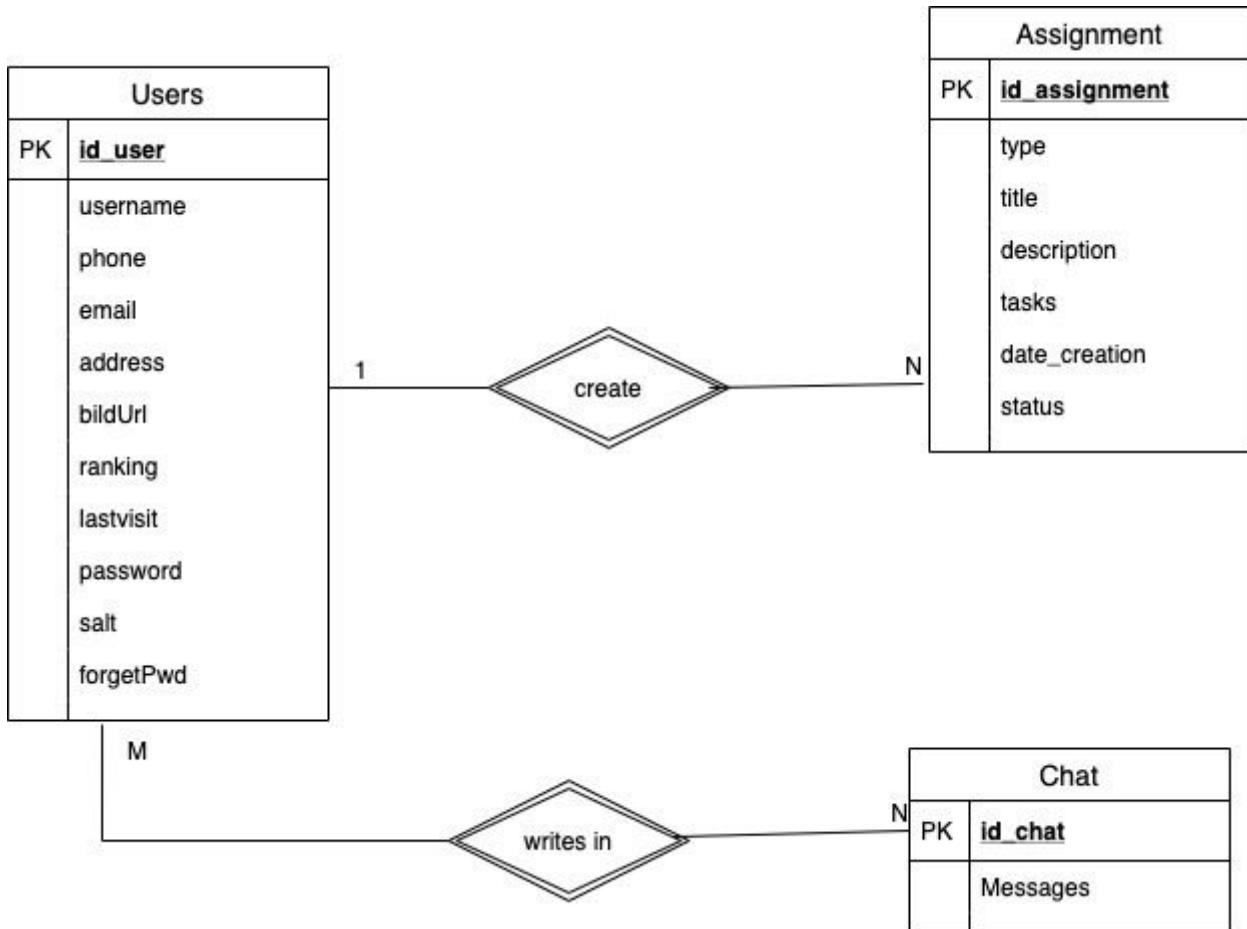
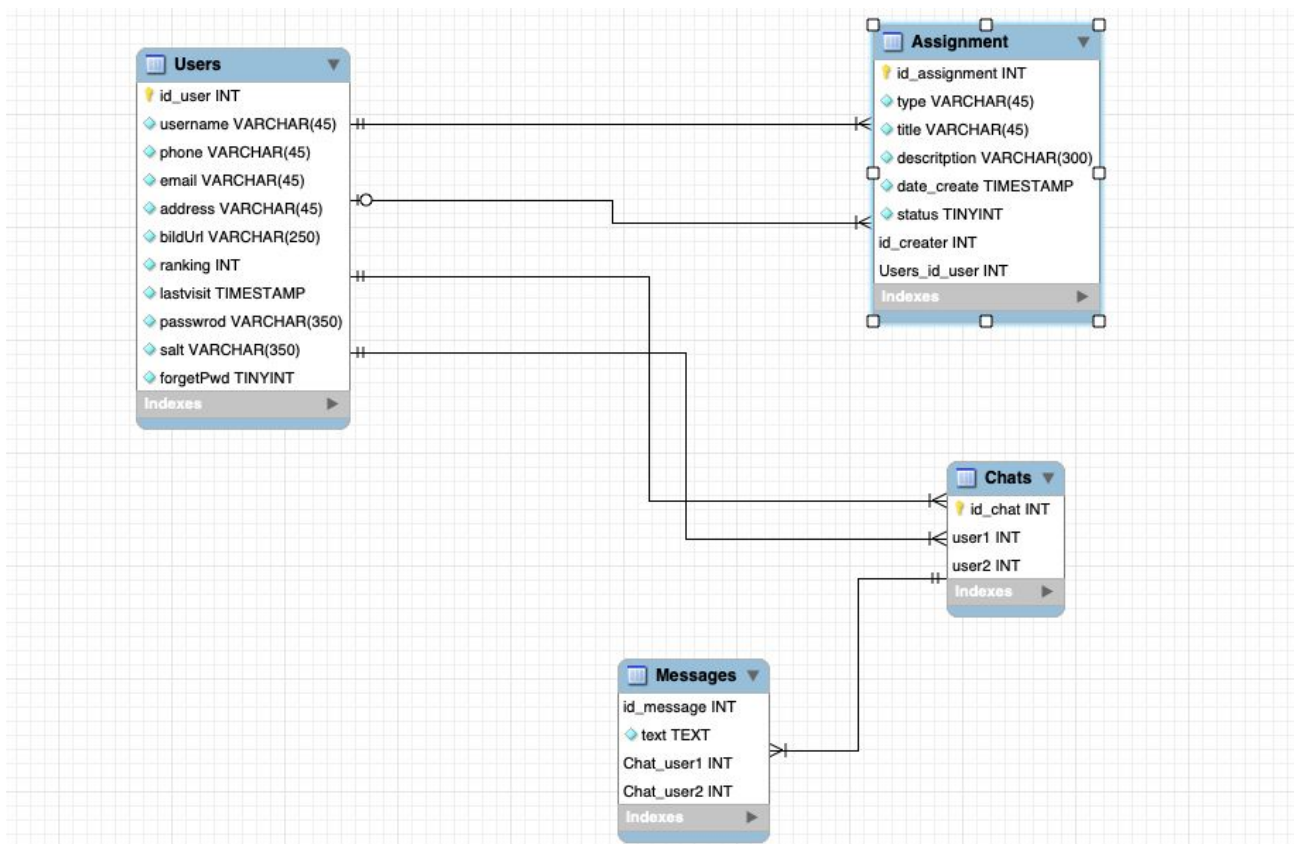


Abbildung 6: Systeminfrastruktur

3.3 Datenbank





3.4 Beschreibung der Implementierung

Detaillierte Beschreibung der notwendigen Entwicklungen/Änderungen pro Komponente zur Durchführung der gewünschten neuen Funktionen

3.4.1 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.

Allgemein

Die Webseite basiert auf das Technogiestack wie HTML, CSS, Angular, React, JavaScript. Das ist die bekannteste Technologie, die man heutzutage benutzt, um die Webseite zu erstellen.

Die Webseite soll Loginform, Suchformular, Bewertungsform enthalten.

Alle Nutzer dürfen mit einer Suchformular suchen und die Suchausgabe durch die HTML-Elemente selektieren

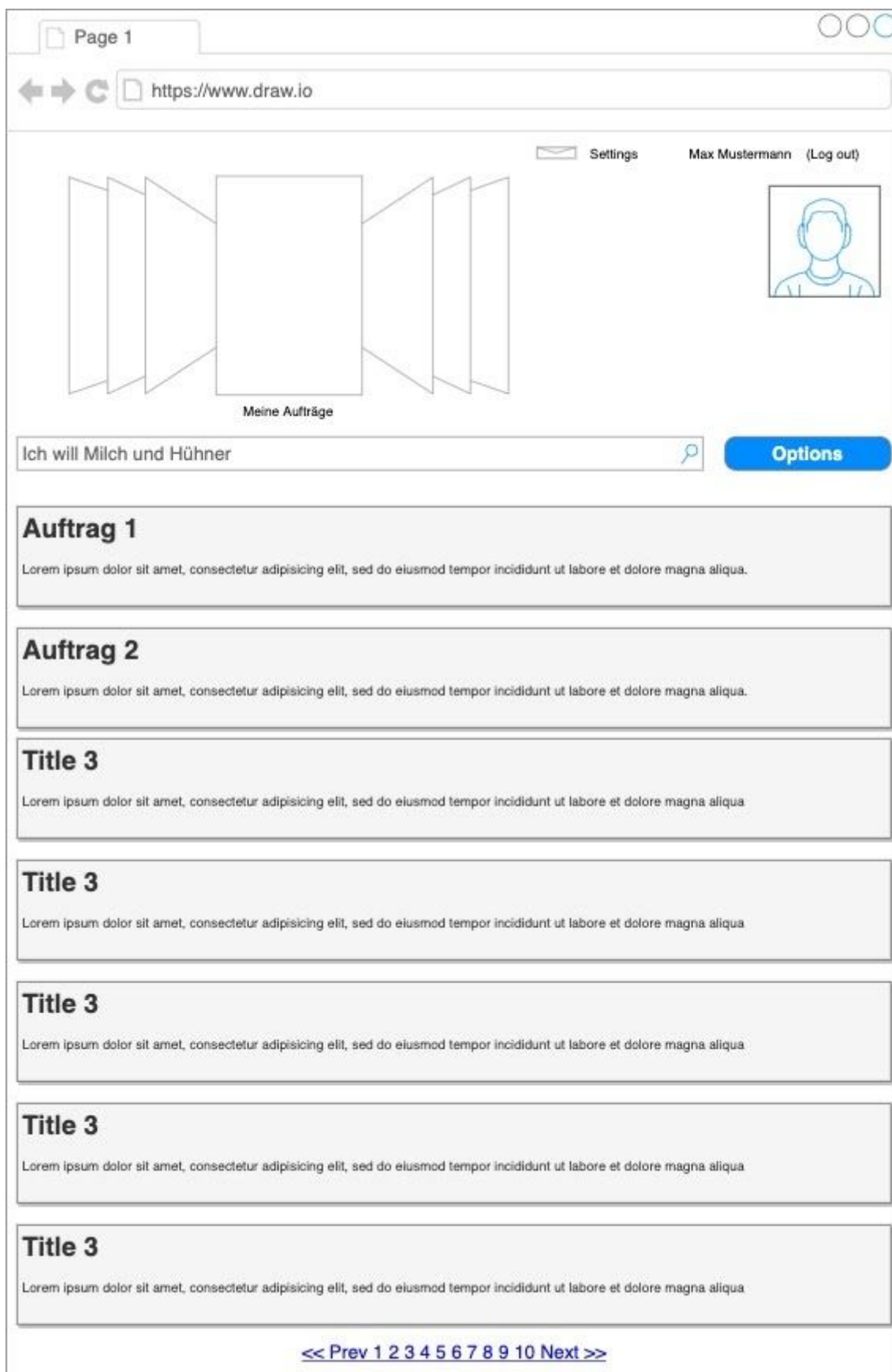
Die Webseite hat eine Möglichkeit den Nutzerkonto zu steuern und die Aufträge zu bearbeiten.

Anmeldung, Suchverfahren

Die Webseite sendet die Anfrage zu Server und bekommt die Antwort. Wenn die Antwort vom Server eine nicht leere Menge mit Daten hat, dann zeigt die Webseite das Ergebnis. sonst die Fehlermeldung

3.4.2 Mockups

The mockup depicts a web browser window with a single tab labeled 'Page 1'. The address bar shows 'https://www.draw.io'. A navigation bar at the top right contains the text 'Anmeldung'. The main content area is initially obscured by a large 'X'. Below this, a search bar with the placeholder 'Search' and a magnifying glass icon is positioned next to a blue 'Options' button. The content area features three stacked cards: 'Auftrag 1', 'Auftrag 2', and 'Title 3', each with a title and a paragraph of Lorem Ipsum text. At the bottom, a pagination control shows '<< Prev 1 2 3 4 5 6 7 8 9 10 Next >>'. The browser window includes standard OS window controls (minimize, maximize, close) in the top right corner.



Page 1

https://www.draw.io

Sign In

User Name:

john doe

Password:

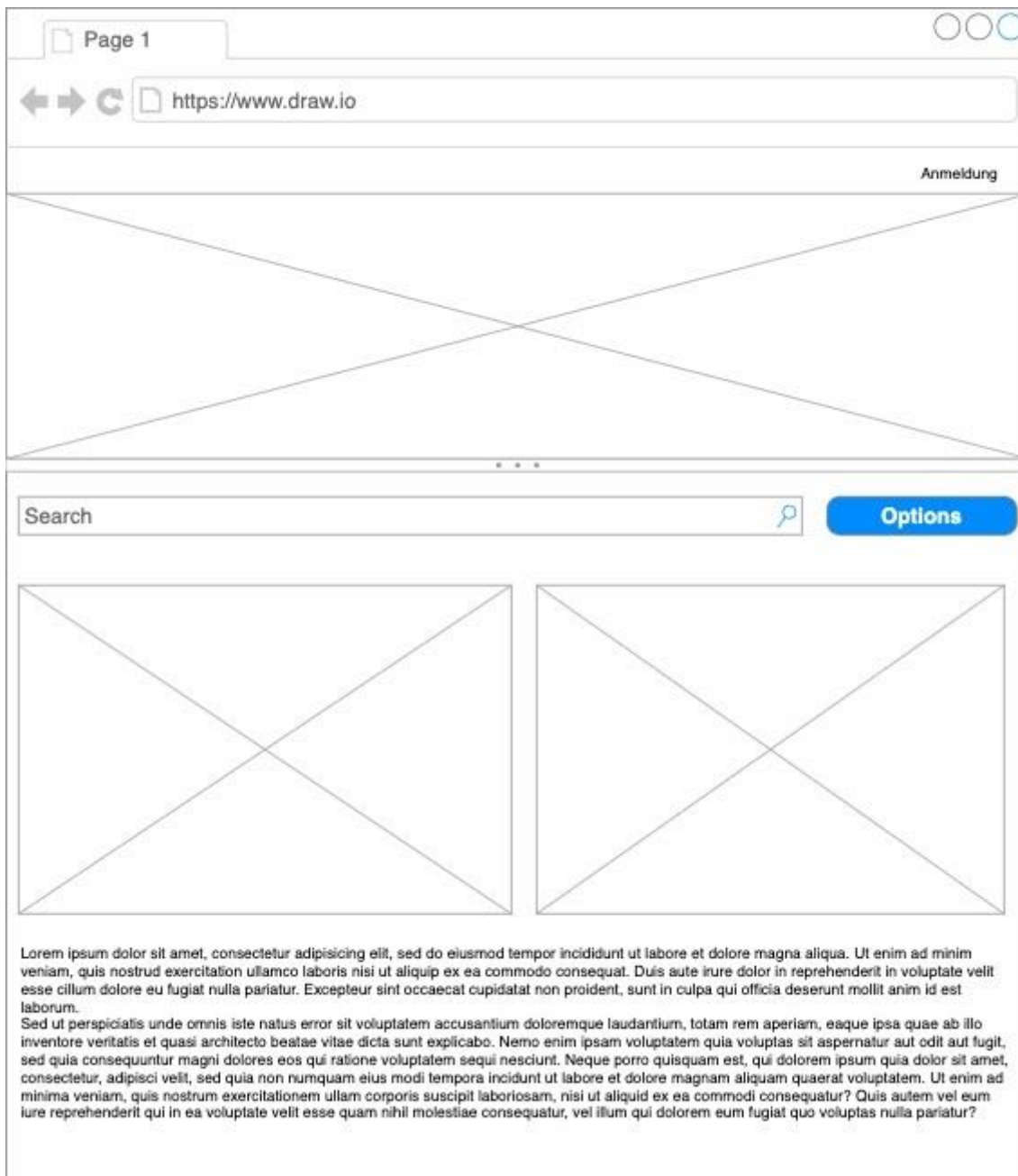
SIGN IN

[Forgot Password?](#)

New User

SIGN UP

Search



3.4.3 Backend: Datenbank, Web-Server und GraphQL

Die Struktur des Backends kann Abbildung 6 entnommen werden.

3.4.3.1 MariaDB

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



Abbildung 8: MariaDB

3.4.3.2 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 *GraphQL* in den *express* Webserver denkbar).

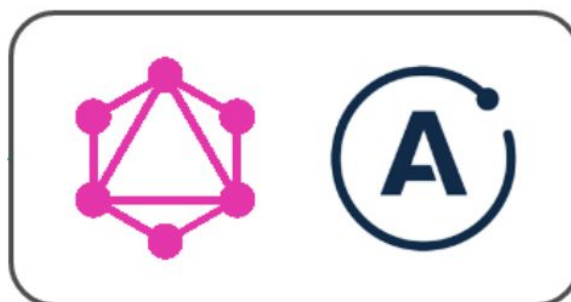


Abbildung 9: GraphQL + Apollo

3.4.3.3 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 10: NodeJS + express

4 Arbeitsaufteilung

Aufgabe/Kapitel	Autor	Noch zu machen	Fertig
Anwendungsüberblick	Benedikt		✓
Prozessüberblick	-	-	-
Nutzer-Workflow	Wajdi		
Technischer Workflow	Dmitry, Wajdi		
Technische Spezifikation SW	-	-	-
Überblick Komponenten	Benedikt, Yannik		✓
Systeminfrastruktur	Yannik		✓
Datenbank	Dmitry		
Beschreibung der Implementierung	-	-	-
Webseiten-Entwicklung	Dmitry		
App-Entwicklung	Benedikt		(✓)
Backend - REST API	Yannik		✓

5 Offene Fragen

#	Issue	Status	Owner	Deadline
	Listen/Kartenansicht nicht dasselbe?		Benji	
	Mit Bezahlung als Möglichkeit oder ohne?			
	Live Verfolgung/Status Updates? Lieferzeit			
	Xamarin			
	Werden Konten als Objekt übergeben?			
	Registrierung extra Klasse?			

Authenticate: google?