Technische Spezifikation

Vergissmeinnicht

Autoren: Yannik
Benedikt
Dmitry
Wajdi

Letzte Änderung: 09.06.2020

Version: 1.0

Inhaltsverzeichnis

Abbildungsverzeichnis

Version Historie

Version:	*Datum:*	*Verantwortlich*	*Änderung*
0.1	06.05.2020	Gruppe	Initiale Dokumenterstellung
0.2	10.05.2020	Yannik	Erstellung Komponentendiagramm
0.3	11.05.2020	Yannik	Erstellung Systeminfrastruktur
0.5	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 Backend Kapitel
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
1.3	05.06.2020	Yannik	Ausgliederung Backend Ausarbeitung API

1 Prozessüberblick

1.1 Technischer Workflow

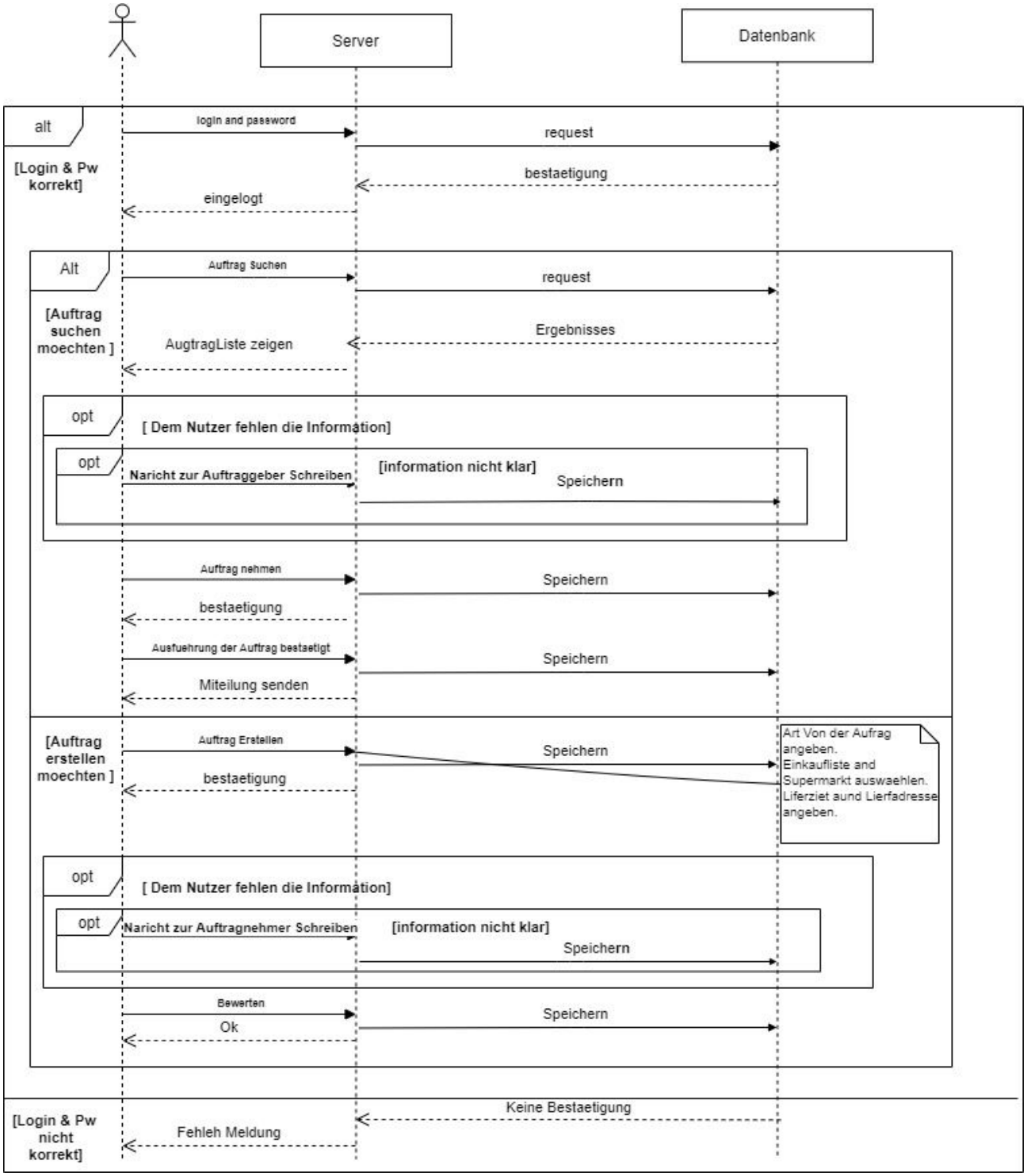


Abbildung 2: Technischer Workflow

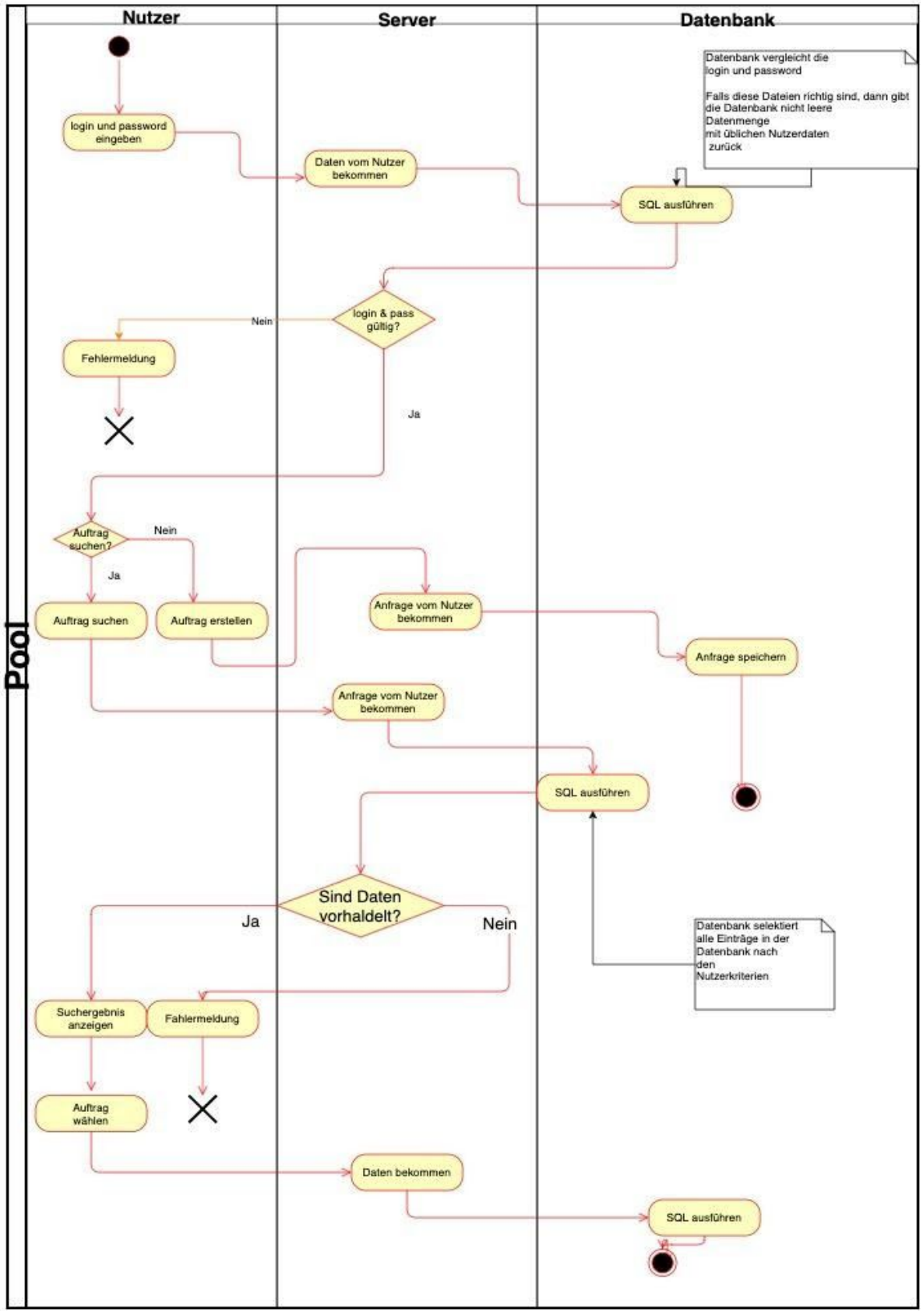


Abbildung 3: Aktivitätsdiagramm

1.2 Nutzer-Workflow

1.2.1 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.

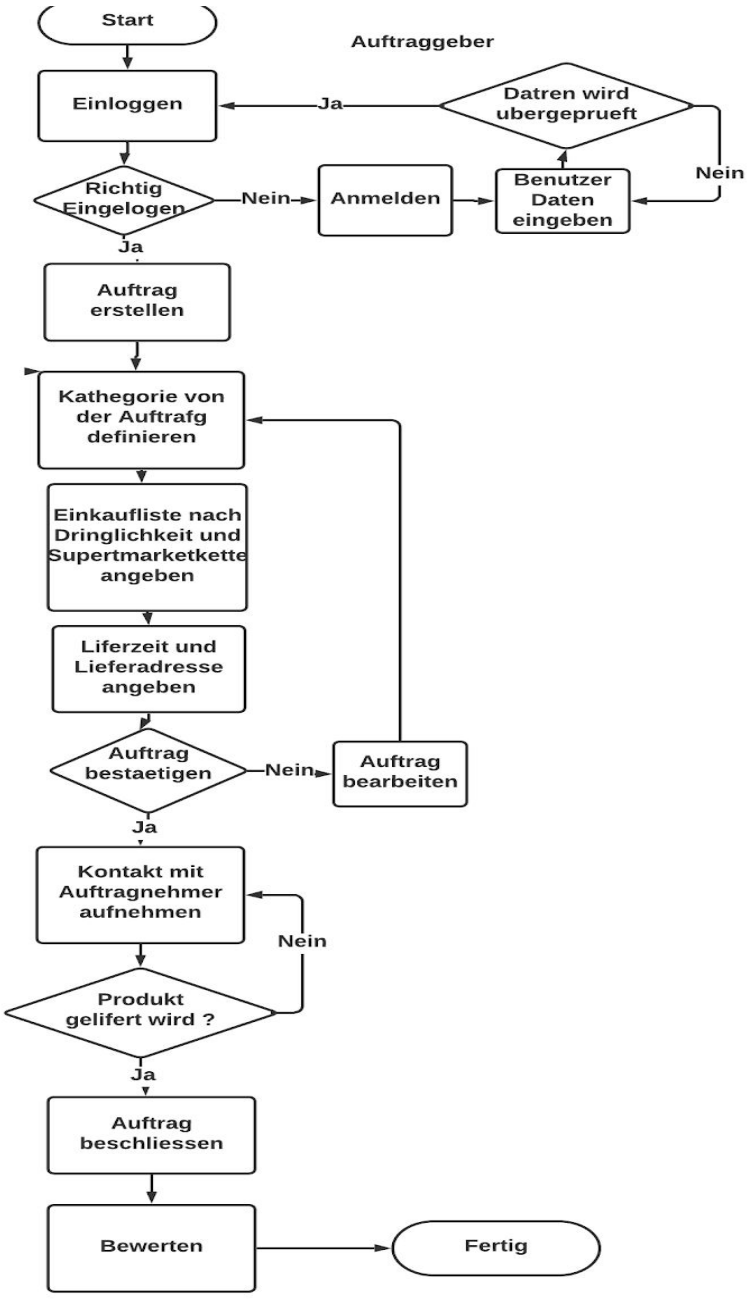


Abbildung 4: Ablaufdiagramm aus Sicht Auftraggeber

1.2.2 Nutzer Workflow : Auftragnehmer

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

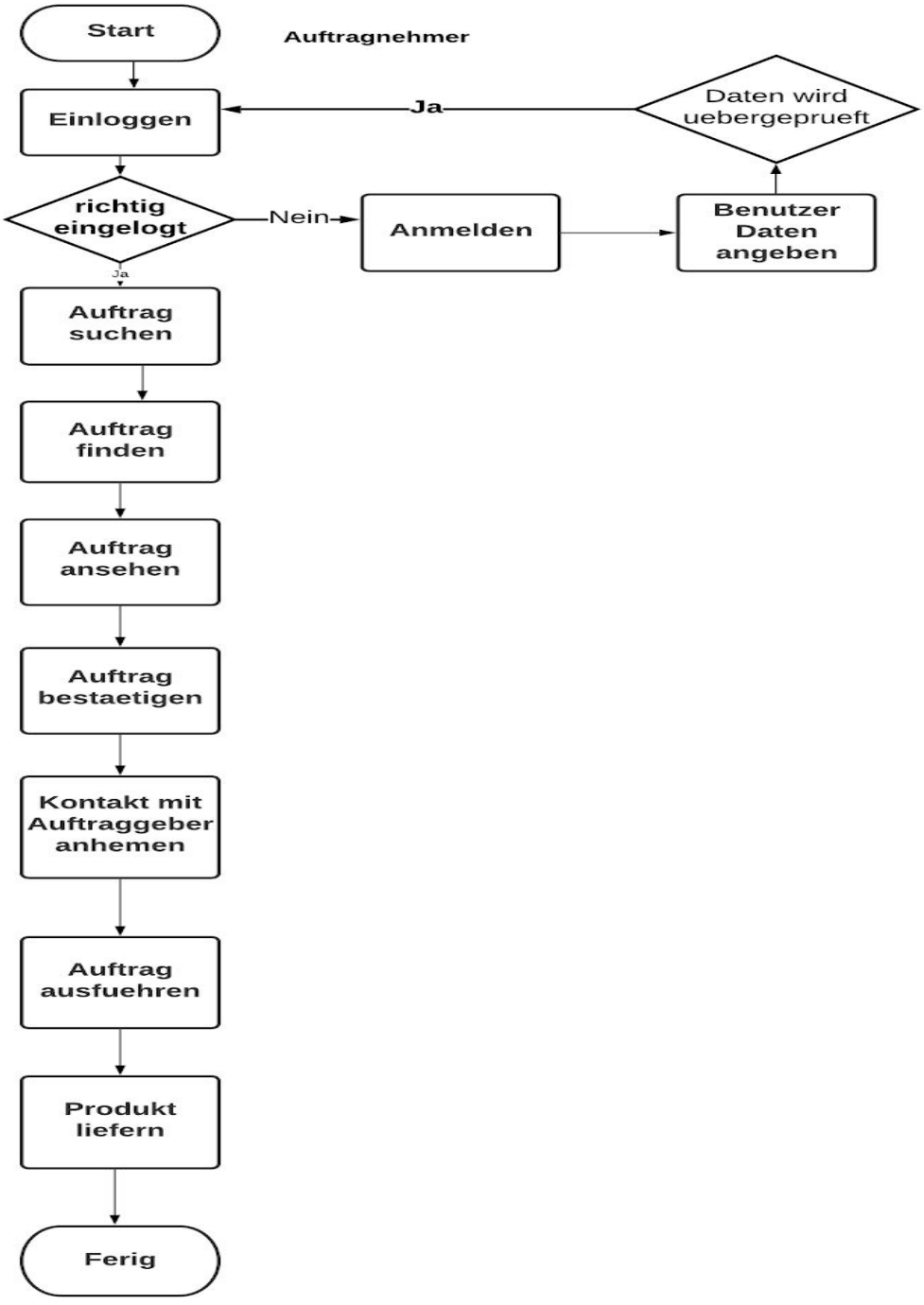


Abbildung 5 : Ablaufdiagramm aus Sicht Auftragnehmer

2 Softwareentwicklung

2.1 Komponentendiagramm

Das Projekt wird in die folgenden Komponenten und Subkomponenten unterteilt:

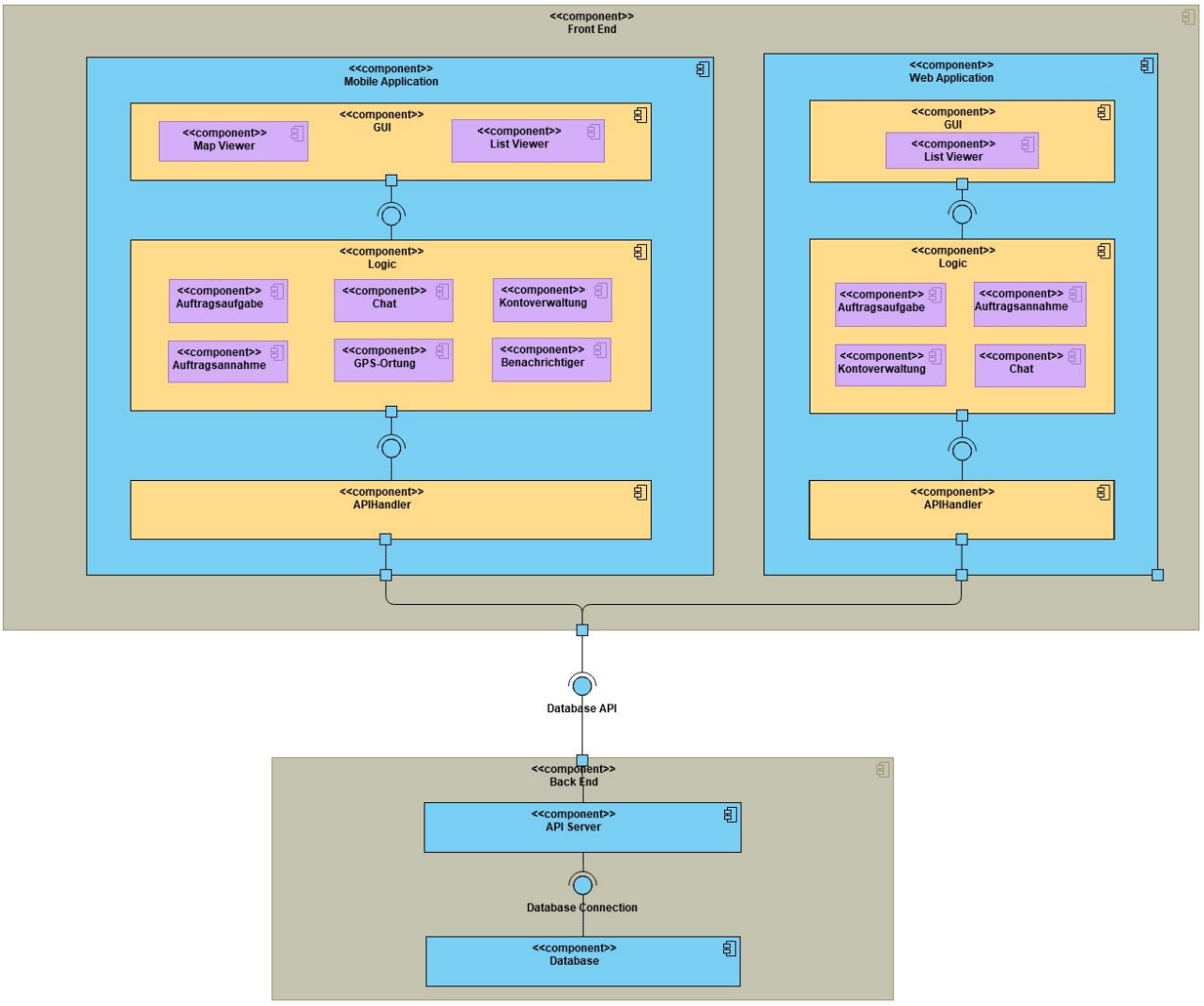


Abbildung 6: Komponentendiagramm

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

2.2 Klassendiagramm

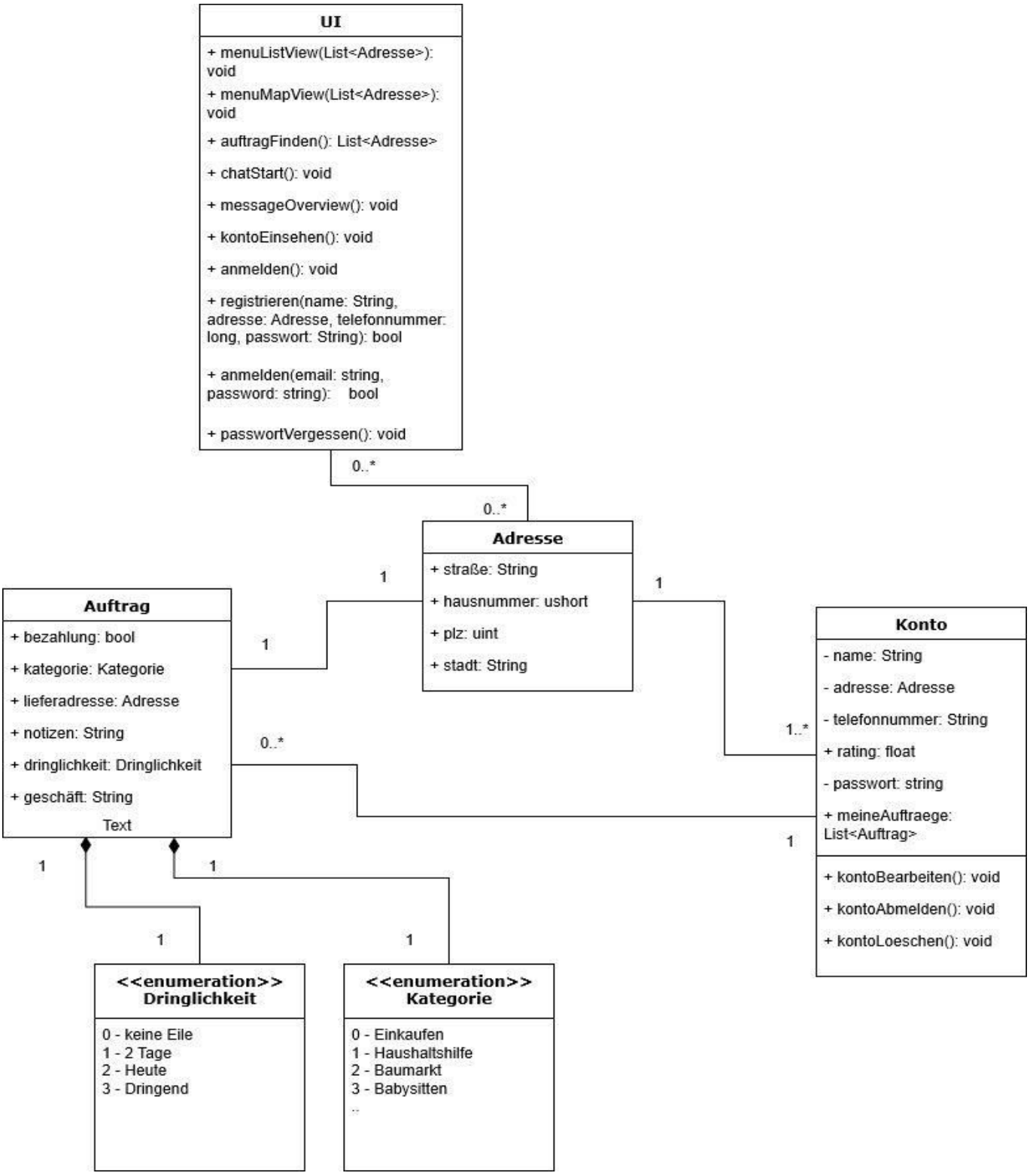


Abbildung 7: Klassendiagramm

2.3 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.

2.3.1 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 Hilfegesuchs (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 zeig die Webseite das Ergebnis. sonst die Fehlermeldung

2.3.2 Mockups

Visualisierungen der späteren Benutzeroberfläche für Website und Mobile Applikation.

2.3.2.1 Website

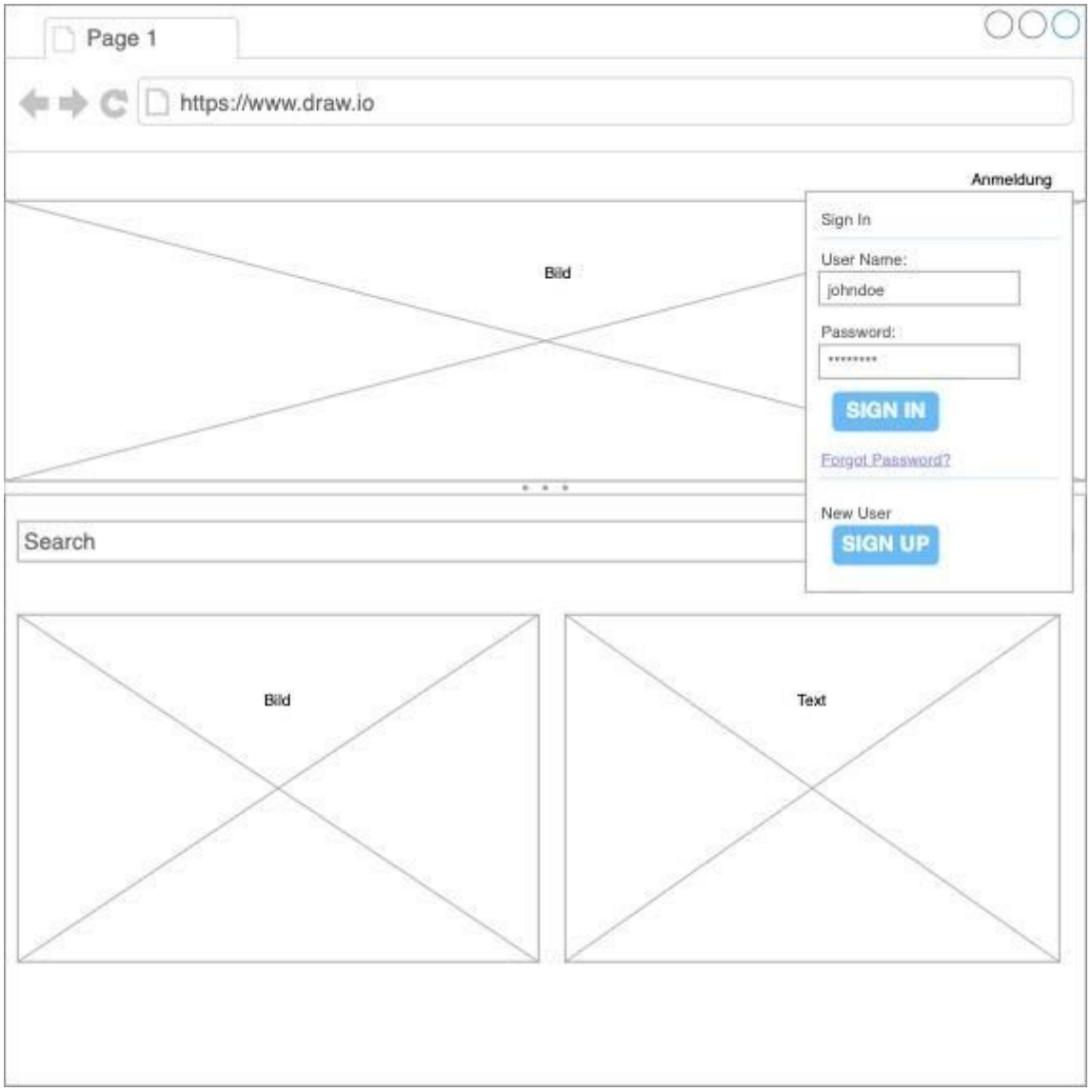


Abbildung 11: Loginform. Hauptseite

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.

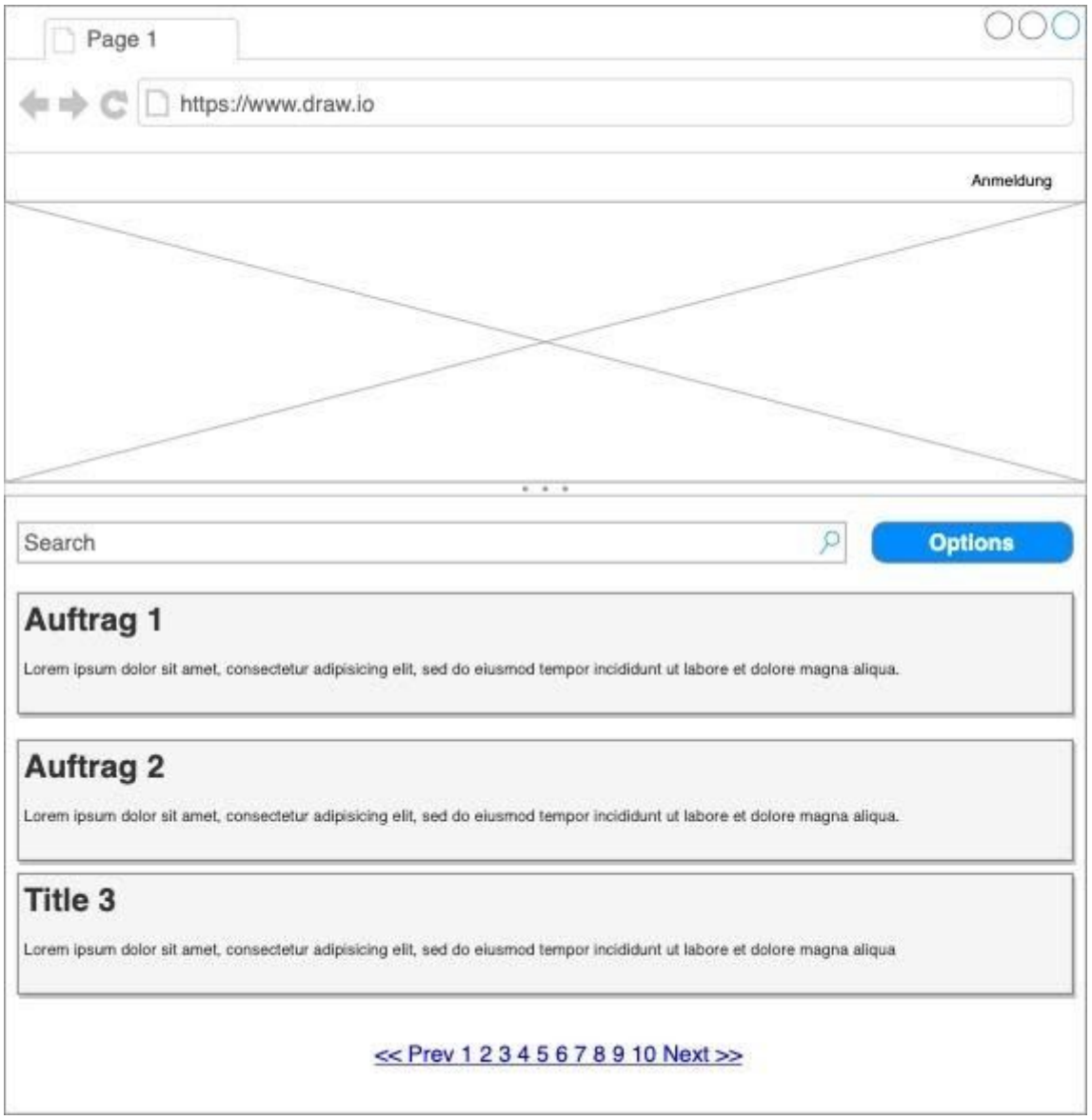


Abbildung 12: Hauptseite mit Suchergebnisse

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

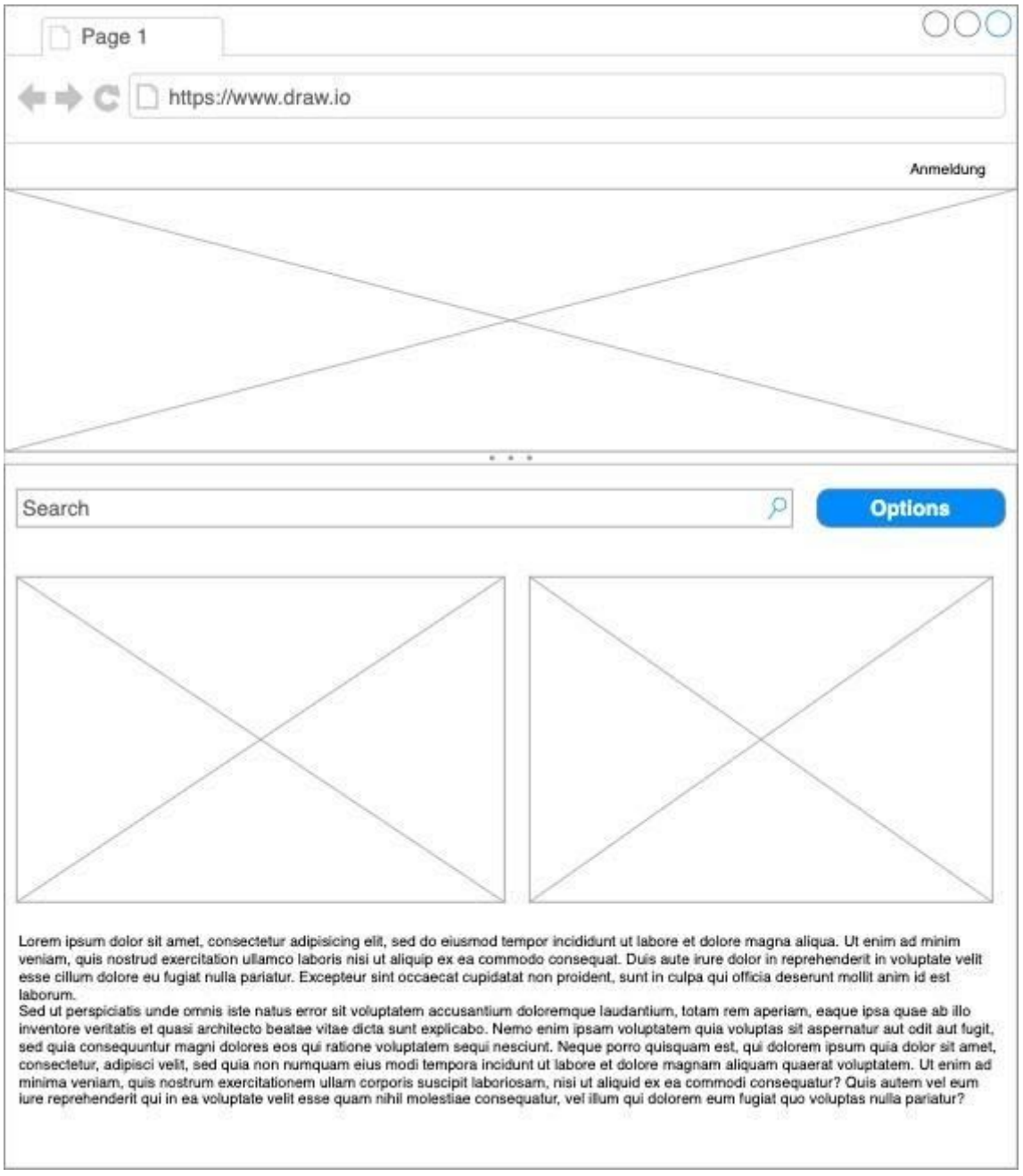


Abbildung 13: Hauptseite

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

2.3.2.2 Mobile Applikation

GUI für die Auftragssuche:

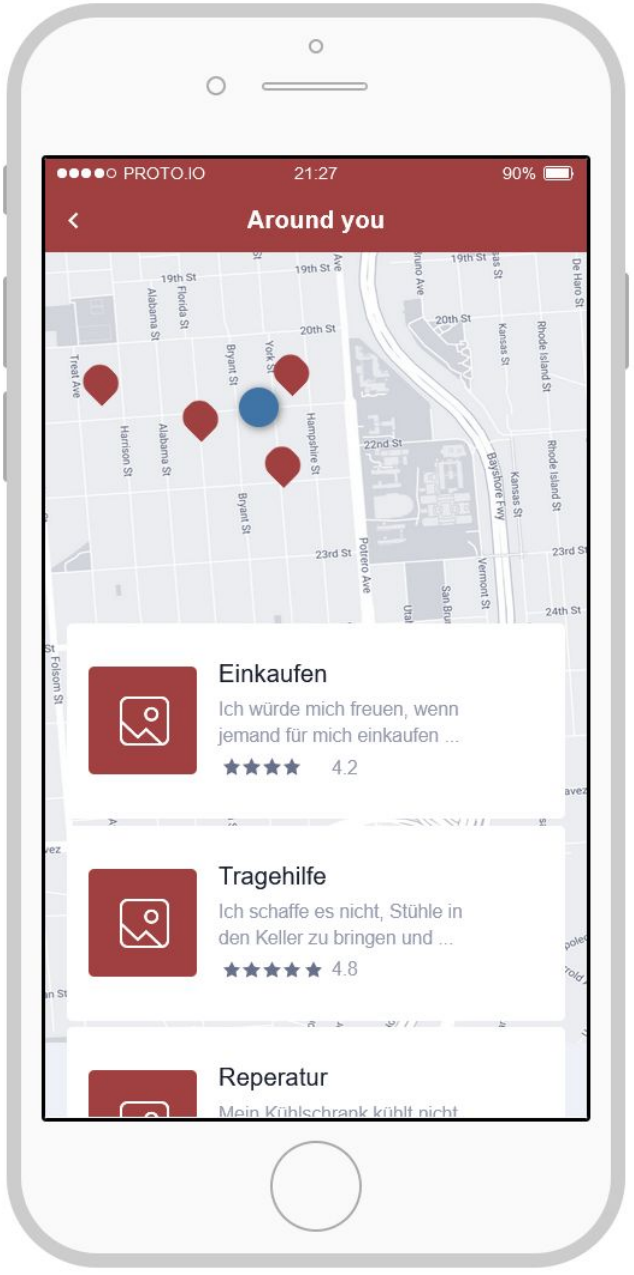


Abbildung 14: Mobile Applikation

3 Backend: Datenbank, Web-Server und GraphQL

3.1 Systeminfrastruktur

Wir verwenden als Backend einen Ubuntu-Server, welcher eine MySQL-Datenbank hosten wird. Ein API-Server wird Teile der Datenbank für die Außenwelt, im Speziellen für die Mobile App und den Webserver, 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.

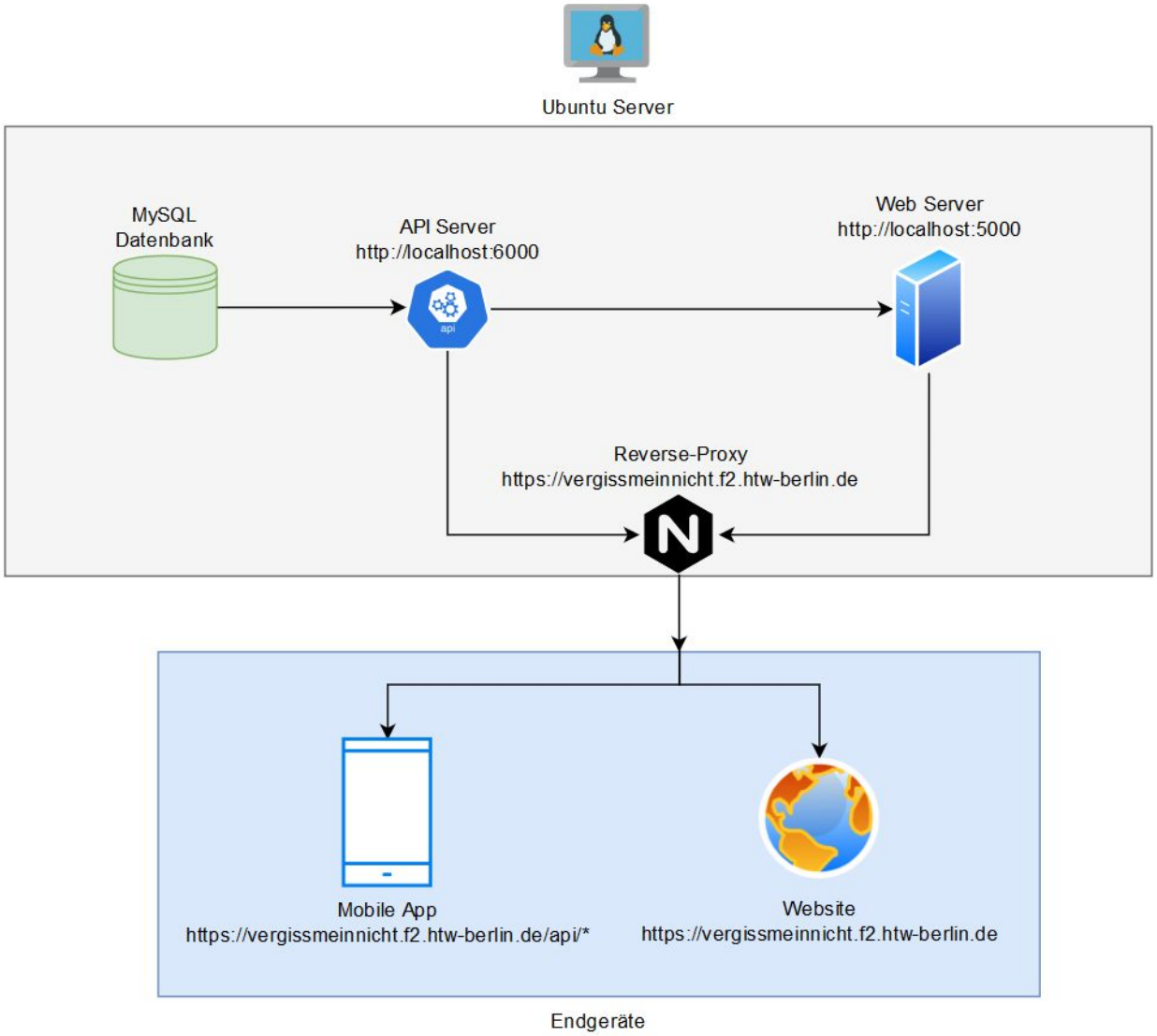


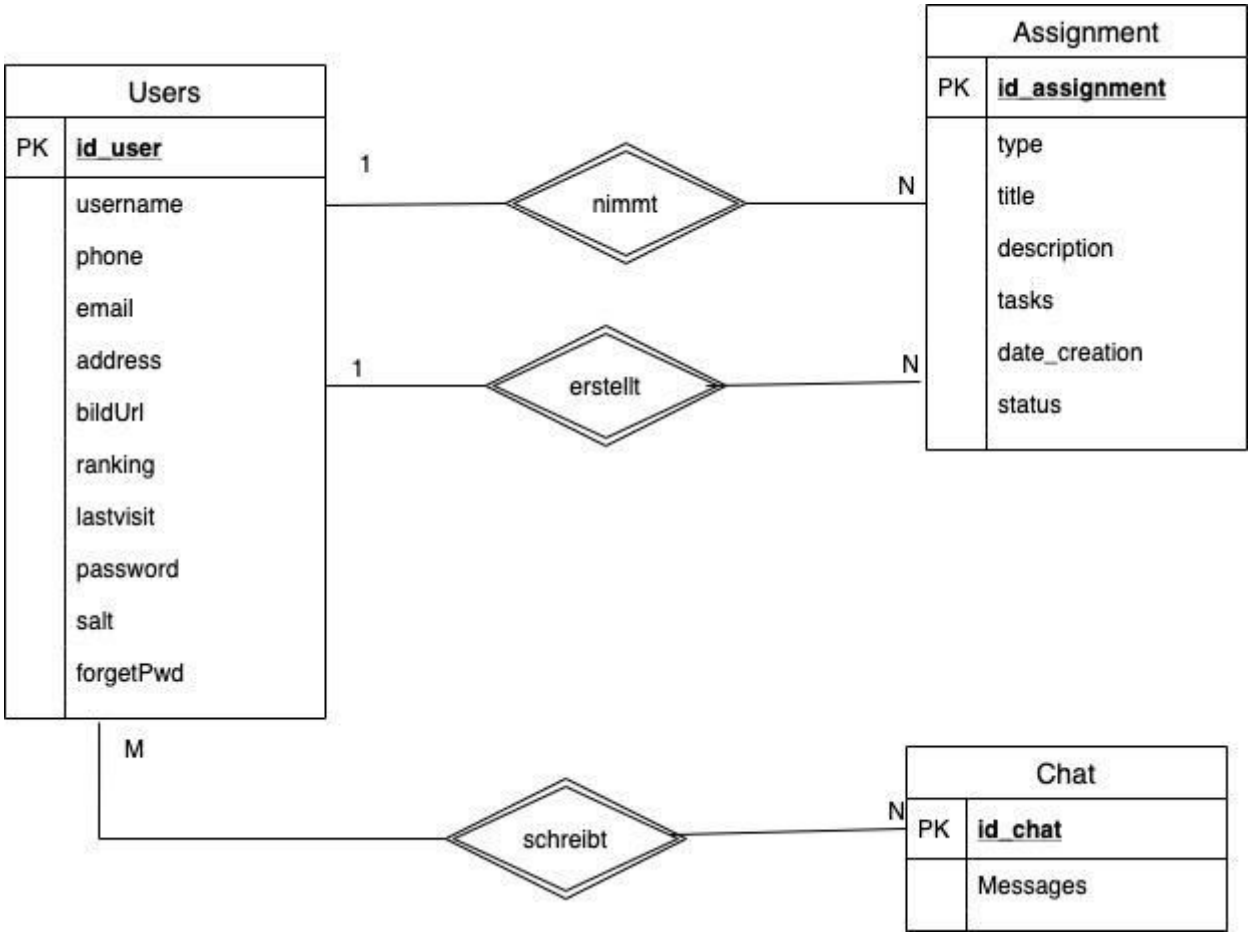
Abbildung 8: Systeminfrastruktur

3.2 Datenbank

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

3.2.1 Datenbankschema



3.3 API: ASP.NET Core

Zur Entwicklung der API haben wir uns für das ASP.NET Web Frameworks entschieden, damit wir Front- und Backend übergreifend C# zur Programmierung verwenden können.
Mit diesem wäre es möglich, sowohl die API als auch die Webseite zu programmieren.

Abbildung 18: ASP.NET Core
Bildquelle: https://static.gunnarpeipman.com/wp-content/uploads/2019/10/aspnet-core-featured.png

Einer der Endpoints unserer API ist für Informationen über Nutzer zuständig und erreichbar unter der Adresse https://vergissmeinnicht.f2.htw-berlin.de/api/users.

Folgende Operationen auf den Endpoint sind möglich:

Informationen...	**Einzelner user**	**Mehrere user**
… erfragen	HTTP Get api/users/3	HTTP Get api/users
… eintragen	-	HTTP Post api/users
… ändern	HTTP Put api/users/3	-
… löschen	HTTP Delete api/users/3	-

Die HTTP Post und Put Anfragen müssen im Body der Anfrage die jeweils neuen Werten enthalten, damit diese in die Datenbank aufgenommen werden können. HTTP Get und Delete werden ohne Parameter aufgerufen.

Zusätzlich zu den Nutzern, dem /api/users Endpoints, sind folgende weitere Endpoints erreichbar:

- Für Aufträge: https://vergissmeinnicht.f2.htw-berlin.de/api/jobs
- Für Nachrichten: https://vergissmeinnicht.f2.htw-berlin.de/api/messages
- Für Chats: https://vergissmeinnicht.f2.htw-berlin.de/api/chats
- Für Bewertungen: https://vergissmeinnicht.f2.htw-berlin.de/api/ratings

Für diese gelten die selben Operationen, wie sie oben für den Nutzer-Endpoint beschrieben wurden.

3.4 Reverse Proxy und Systemd

Abbildung 18: NGINX
Bildquelle:https://www.nginx.com/wp-content/uploads/2018/08/NGINX-logo-rgb-large.png

Wie in Abbildung 8 zu sehen ist, verwenden wir einen Reverse Proxy, wodurch jeglicher Traffic, der unseren Server erreicht, zuerst durch den Reverse Proxy gehen muss, bevor API oder Webserver erreicht werden können. Dies hat mehrere Vorteile:

1. **Sicherheit**: Alle Komponenten befinden sich hinter dem Reverse Proxy, wodurch jeglicher unerwünschter Traffic unterbunden werden kann und somit die dahinterstehenden Komponenten nie erreicht wird.
2. **Load Balancing**: Sollten wir unerwartet viel Traffic erfahren, kann der Reverse Proxy mehrere zukünftige Server so konfigurieren, dass alle verschiedenen Server ungefähr gleich viel “Load” erfahren. Somit sind wir für den Fall der Fälle gewappnet und können einfach zusätzliche Server aufsetzen.
3. **Vereinfachte HTTPS Zertifizierung**: Da nur ein Endpoint nach außen hin sichtbar und erreichbar ist, reicht es, diesen mit HTTPS Zertifizierung auszustatten und SSL-tauglich zu machen, wodurch nun jeder Datenverkehr zu dem Reverse Proxy verschlüsselt wird. Haben die Daten den Reverse Proxy erreicht, werden sie dort entschlüsselt und werden auch in der Form an die jeweiligen Komponenten weitergeleitet.

Unsere Webseite und API laufen als Dotnet-Projekte auf dem Ubuntu-Server und werden über den Befehl “dotnet insertnamehere.dll --urls=http://localhost:port” ausgeführt.
Die Prozesse der Webseite und API wurden als Services über den Linux Systemdaemon systemd konfiguriert, sodass dieser diese administriert. So wird dafür gesorgt, dass bei unerwartetem Abbruch der Prozesses oder bei Absturz der ganzen Maschine die Prozesse neu gestartet und entsprechende Logs aufgezeichnet werden.

4 Arbeitsaufteilung

Aufgabe/Kapitel	Autor	Status
Prozessüberblick	-	X
Technischer Workflow	Dmitry, Wajdi	X
Nutzer-Workflow	Wajdi	X
Nutzer-Workflow : Auftraggeber	Wajdi	X
Nutzer Workflow : Auftragnehmer	Wajdi	X
Softwareentwicklung	-	✓
Komponentendiagramm	Yannik	✓
Klassendiagramm	Benedikt	✓
Implementierung	Dmitry, Benedikt	✓
Entwicklung	Dmitry, Benedikt	✓
Mockups	Dmitry, Benedikt	✓
Website	Dmitry, Benedikt	✓
Mobile Applikation	Yannik	✓
Backend: Datenbank, Web-Server und GraphQL	-	-
Systeminfrastruktur	Yannik	✓
Datenbank	Yannik	✓
Datenbankschema	Dmitry	(✓)
API: ASP.NET Core	Yannik	✓
Reverse Proxy und Systemd	Yannik	(✓)