

SEP Projektmappe

PROJEKTMAPPE DES PROJEKTES

Gruppe N

Dokumentation des Projektes

Gruppe N:

Abdullah Reslan

Kerem Oener

Nevzat Yaprak

Chantal Jedrschyk

Inhalt

Projektbeschreibung	4
Zyklus I	6
Spezifikationsplanung	6
User-Stories	8
Papierprototypen	10
Strukturdiagramme (Komponenten- und Klassendiagramme)	15
Verhaltensdiagramme (Kommunikationsdiagramme)	17
Funktionalitätsplanung	18
Systemtests	22
Zyklus II	27
Spezifikationsplanung	27
User-Stories	28
Papierprototypen	31
Strukturdiagramme (Komponenten- und Klassendiagramme)	36
Verhaltensdiagramme (Kommunikationsdiagramme)	39
Funktionalitätsplanung	41
Modultests	45
Systemtests	45
Zyklus III	50
Spezifikationsplanung	50
User-Stories	52
Papierprototypen	54
Strukturdiagramme (Komponenten- und Klassendiagramme)	58
Verhaltensdiagramme (Kommunikationsdiagramme)	61
Funktionalitätsplanung	63
Modultests	66
Systemtests	67
Nutzerhandbuch	73
Technische Anforderungen	73
Installationsanleitung	73
Bedienungsanleitung	74

Projektbeschreibung

SEP-Drive ist eine Mitfahr-App, bei der sich Nutzer als Fahrer oder Kunde registrieren können. Kunden erstellen Fahrtanfragen mit Start, Ziel und optionalen Stopps. Die App berechnet automatisch die Route, die Dauer, die Distanz und den Preis.

Fahrer sehen verfügbare Anfragen und können diese anbieten. Kunden können die verfügbaren Angebote einsehen und ein Angebot annehmen oder ablehnen. Wenn ein Fahrer eine Anfrage angeboten hat, können Fahrer und potenzieller Kunde sich über einen Chat austauschen.

Sobald ein Kunde ein Angebot annimmt, wird die geplante Route auf einer interaktiven Karte visualisiert und eine Echtzeitsimulation der Fahrt kann gestartet werden. Während der Simulation können Kunden das Ziel oder Zwischenstopps ändern und die Route wird dann angepasst.

Zahlungen erfolgen über persönliche Geldkonten, die mit virtuellem Guthaben aufgeladen werden können. Nach Abschluss der Fahrt wird der Betrag automatisch vom Kundenkonto abgebucht und dem Fahrer gutgeschrieben.

In den Benutzerprofilen lassen sich persönliche Daten sowie die Fahrhistorie mit Details zu vergangenen Fahrten einsehen. Fahrer können zusätzlich statistische Grafiken zu Einnahmen, Fahrtzeit und gefahrenen Distanzen erstellen.

Zyklus I

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Registrierung der Nutzer			
1.1	User Story 1	User Story	Nevzat	fertig
1.2	Registrierungs-Fenster	Papierprototyp	Nevzat	fertig
2.	Backend Architektur			
2.1	Komponentendiagramm Backend	Diagramm	Abdullah	fertig
2.2	Klassendiagramm Backend	Diagramm	Kerem	fertig
2.3	Kommunikationsdiagramme Backend	Diagramm	Nevzat	fertig
3.	Frontend			
3.1	Komponentendiagramm Frontend	Diagramm	Chantal	fertig
4.	Login eines Benutzers			
4.1	User Story 2	User Story	Kerem	fertig
4.2	Login-Fenster	Papierprototyp	Kerem	fertig
5.	Benutzerprofile			
5.1	User Story 3	User Story	Abdullah	fertig
5.2	Kundenprofil	Papierprototyp	Abdullah	fertig
5.3	Fahrerprofil	Papierprototyp	Abdullah	fertig
6.	Kartenvisualisierung von Routen			
6.1	User Story 4	User Story	Chantal	fertig
6.2	Interaktive Karte	Papierprototyp	Chantal	fertig
7.	Fahranfragen erstellen			
7.1	User Story 5	User Story	Chantal	fertig
7.2	User Story 6	User Story	Kerem	fertig

7.3	Fahranfragen-Fenster	Papierprototyp	Kerem	fertig
7.4	Aktive-Fahrt Fenster	Papierprototyp	Nevzat	fertig

User-Stories

User Story-ID	1
User Story-Beschreibung	Als Benutzer möchte ich ein Registrierungsformular ausfüllen können, damit ich mich auf der Plattform anmelden kann.
Geschätzter Realisierungsaufwand	3d
Priorität	mittel
Autor	Nevzat Yaprak
Abhängigkeiten zu anderen User Stories	

User Story-ID	2
User Story-Beschreibung	Als Benutzer möchte ich ein Login-Formular nutzen können, damit ich mich mit meinen Zugangsdaten anmelden kann.
Geschätzter Realisierungsaufwand	3d
Priorität	hoch
Autor	Kerem
Abhängigkeiten zu anderen User Stories	

User Story-ID	3
User Story-Beschreibung	Als Nutzer möchte ich auf mein Benutzerprofil zugreifen können, um meine Profildaten einsehen zu können.
Geschätzter Realisierungsaufwand	1d
Priorität	niedrig
Autor	Abdullah
Abhängigkeiten zu anderen User Stories	

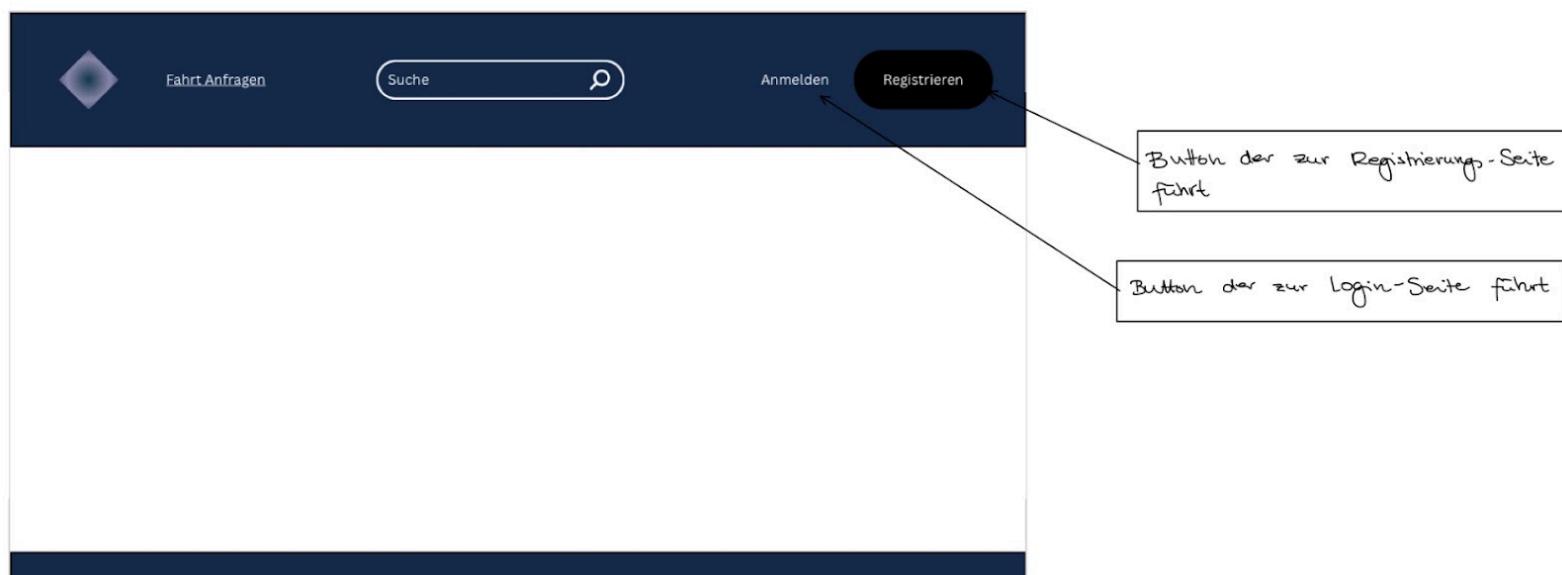
User Story-ID	4
User Story-Beschreibung	Als Benutzer möchte ich eine Route auf einer interaktiven Karte sehen, damit ich meine geplante Strecke besser nachvollziehen kann.
Geschätzter Realisierungsaufwand	5d
Priorität	hoch
Autor	Chantal
Abhängigkeiten zu anderen User Stories	

User Story-ID	5
User Story-Beschreibung	Als Kunde möchte ich eine Fahrt Anfrage über ein Formular erstellen können, damit ich eine Mitfahrmöglichkeit organisieren kann.
Geschätzter Realisierungsaufwand	3d
Priorität	mittel
Autor	Chantal
Abhängigkeiten zu anderen User Stories	

User Story-ID	6
User Story-Beschreibung	Als Kunde möchte ich eine aktive Fahrt Anfrage auf einer Seite sehen können, und sie gegebenenfalls löschen können.
Geschätzter Realisierungsaufwand	3d
Priorität	mittel
Autor	Kerem
Abhängigkeiten zu anderen User Stories	4,5

Papierprototypen

Registrierung eines Benutzers



The registration form is divided into two sections. On the left, there is an optional profile picture input field with a placeholder "Optionales Profilbild hinzufügen" and a "Profilbild" button with a plus sign. On the right, there is a group of required registration fields enclosed in a bracket, labeled "Erforderliche Registrierungsdaten". These fields include: Nutzernname, Vorname, Nachname, E-Mail, Geburtsdatum, Passwort, and a dropdown menu for "Rolle: (Fahrer/Kunde)". Below these fields is a "Registrieren" button. Callout boxes point to the "Rolle" dropdown with "Auswahl : Fahrer / Kunde" and the "Registrieren" button with "Button zur Registrierung".

Login eines Benutzers

Fahrt Anfragen

Suche



Anmelden

Registrieren

Nutzername:
Passwort:



Anmelden

Eingabe der Anmeldedaten

Button zur Anmeldung

Benutzerprofile

Fahrt Anfragen

Suche



Username



Durch klicken auf Username / Profil
gelangt der Benutzer auf sein Benutzerprofil

Durch Suche eines Usernamen gelangt ein
Benutzer auf das Profil

Fahrt Anfragen Suche Anmelden Registrieren

Benutzerprofil

Profilbild → Benutzerprofil mit Rolle Kunde

Username: Bewertung des Bookers

Rolle: (Kunde) ★★★★☆ (5.0)

Nutzerdaten → Benutzerprofil mit Rolle Kunde

Vorname: Auto-Klasse: ▼

Nachname: Auswahl der Auto-Klasse: (Klein, Medium, Deluxe)

E-Mail:

Geburtsdatum:

Gesamtanzahl der Fahrten:

Fahrt Anfragen Suche Anmelden Registrieren

Benutzerprofil

Profilbild → Benutzerprofil mit Rolle Fahrer

Username: Bewertung des Bookers

Rolle: (Fahrer) ★★★★☆ (5.0)

Nutzerdaten → Benutzerprofil mit Rolle Fahrer

Vorname: Auto-Klasse: ▼

Nachname: Auswahl der Auto-Klasse: (Klein, Medium, Deluxe)

E-Mail:

Geburtsdatum:

Gesamtanzahl der Fahrten:

Kartenvisualisierung von Routen



Fahranfragen erstellen

Fahrt Anfragen

(Button führt zu dieser Seite)

Ziel-Eingabe - Feld:
Eingabe kann als Adresse, Point-of-Interest oder als Koordinate erfolgen.

Startpunkt - Eingabe - Feld:
Eingabe kann als Adresse, Point-of-Interest oder als Koordinate erfolgen.

Suche

Username

Fahranfrage

Startpunkt

Ziel

Fahrzeugklasse

Stop hinzufügen +

Fahrt Anfragen

Button zur Suche der aktuellen Position

Auswahl der Fahrzeugklasse : (Klein, Medium, Deluxe)

Button zum Hinzufügen von Zwischenstopps (Auswahl zwischen Adressen, POIs oder Koordinaten)

Fahrt Anfragen

Während aktiver Fahrt kann keine weitere Fahrt aufgetragen werden.

Kartenvisualisierung der Route

Bestätigung, dass die Fahrtanfrage aktiv ist

Aktive Fahrtanfrage

Start

Ziel

Fahrtzeit:

fahrt Informationen

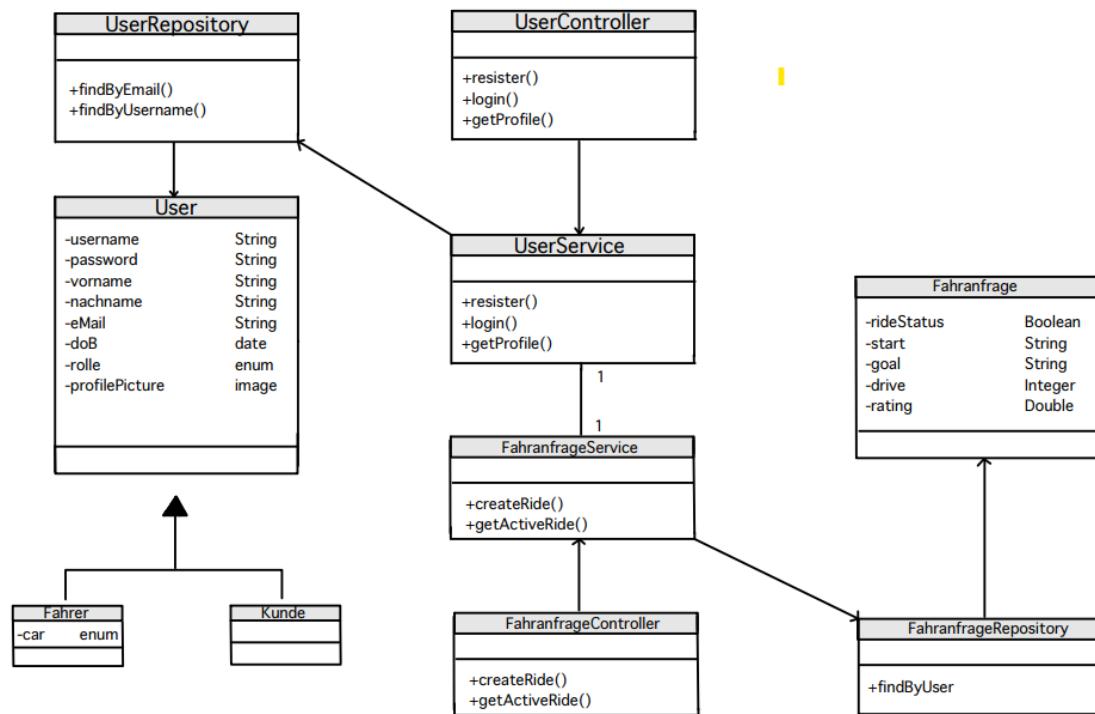
Fahrtanfrage aktiv ✓

Fahrtanfrage löschen

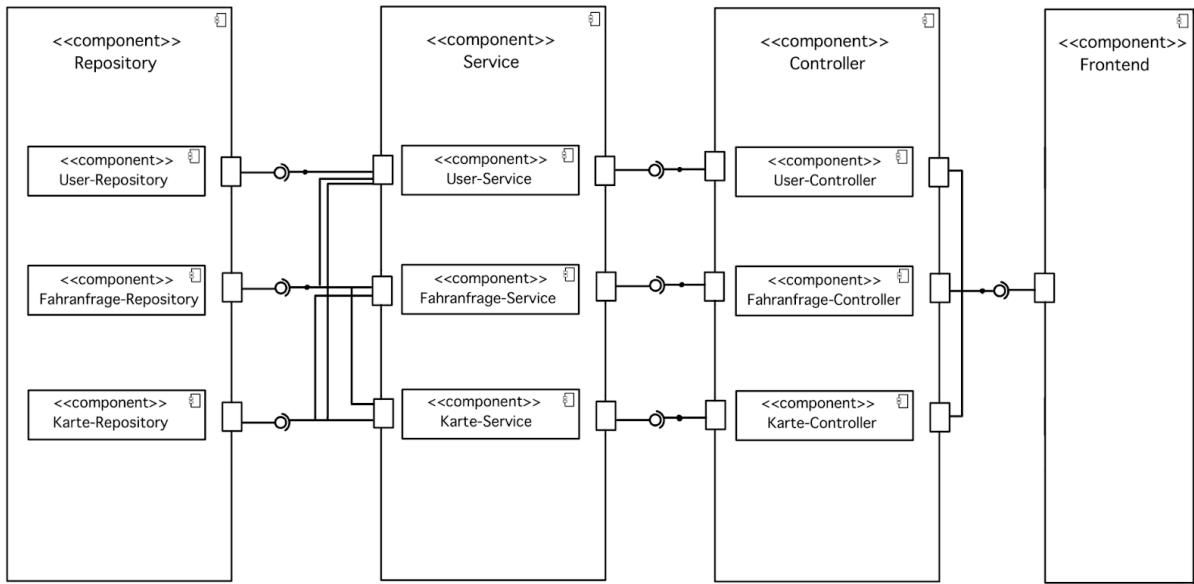
Button zur Löschung der Fahrtanfrage

Strukturdiagramme (Komponenten- und Klassendiagramme)

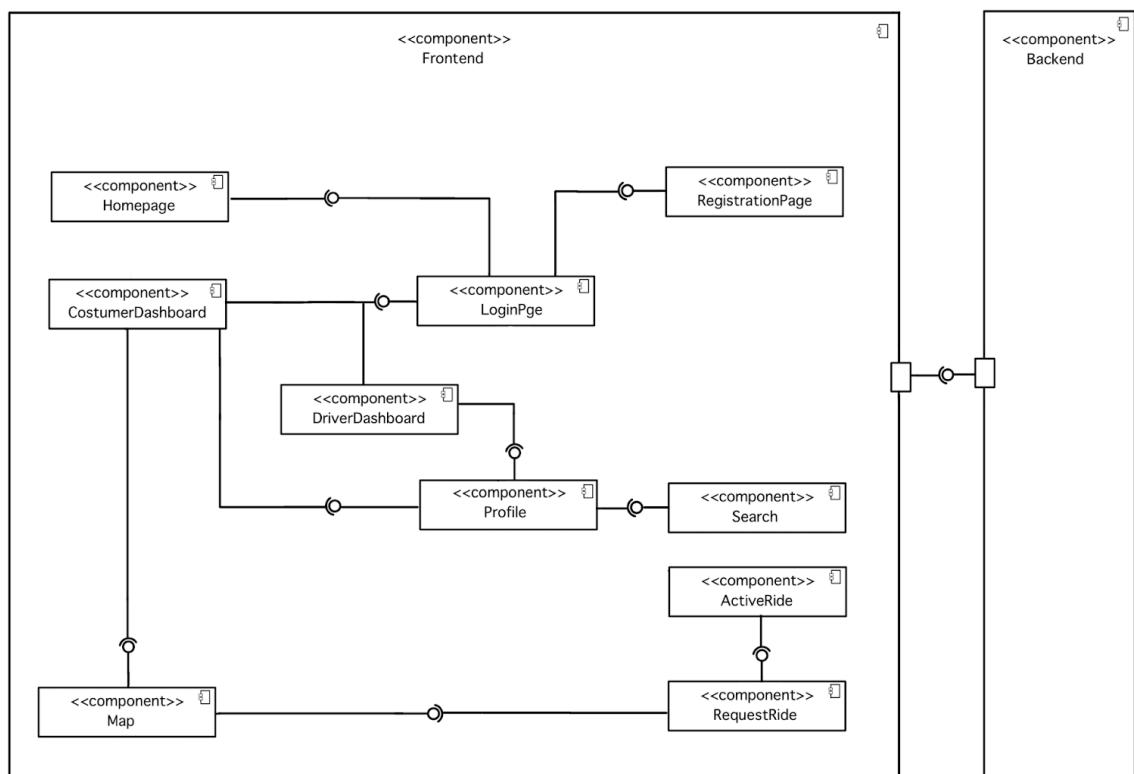
Klassendiagramm:



Komponentendiagramm (Backend):

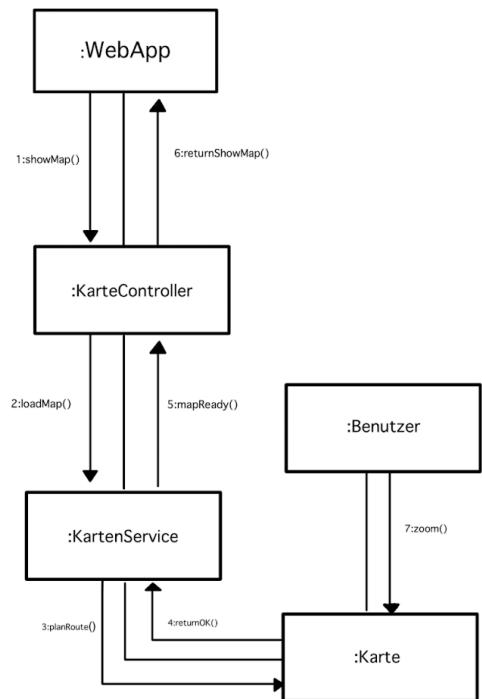


Komponentendiagramm Frontend:

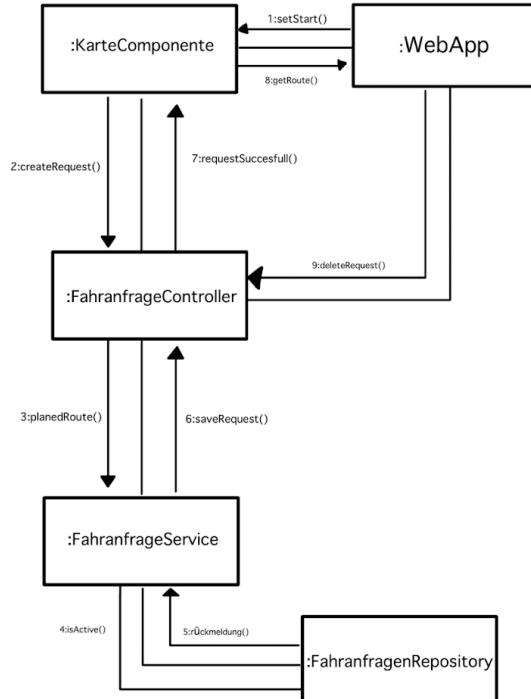


Verhaltensdiagramme (Kommunikationsdiagramme)

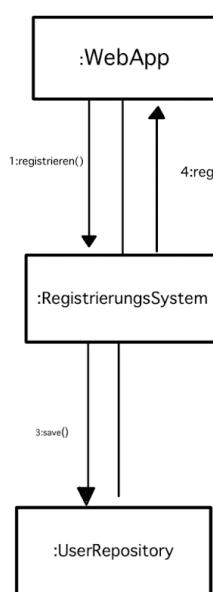
Kartenvisualisierung:



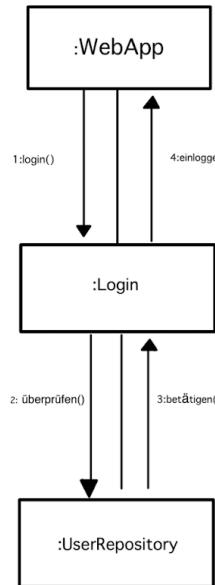
Fahranfrage:



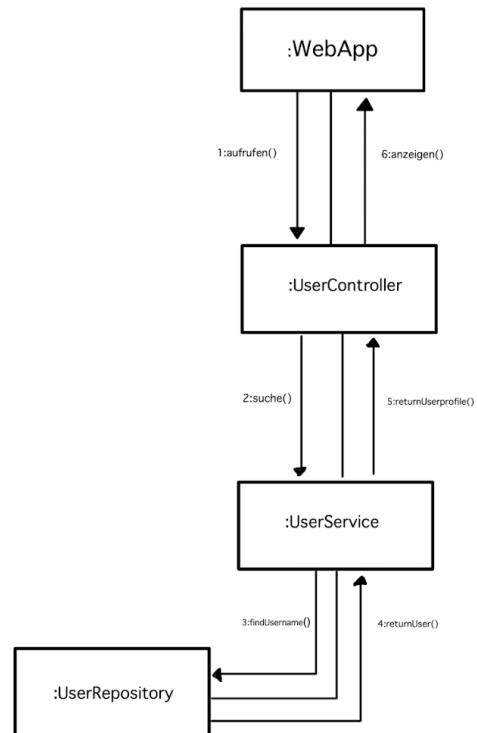
Registrierungssystem:



Anmelde Sytsem:



Benutzerprofil aufrufen:



Funktionalitätsplanung

ID	Funktionalität	Verantwortliche r	Abhängige Funktionalitä ten	Ver kn üpf te Us er-Sto rie s	Quellcode-referenz	Status
1.	Registrierung eines Benutzers					
1.1	Registrierung im Frontend	Kerem	1.2	1	frontend/src/component s/Register.js	Fertig
1.2	AuthService im Frontend	Kerem	1.3, 2.2		frontend/src/services/Au thService.js	Fertig
1.3	Registrierung im Backend	Nevzat	1.2	1	backend/controller/User Controller.java, backend/service/UserSer vice.java backend/repository/User Repository.java, backend/entity/User.java , backend/entity/Role.java, backend/dto/UserRegistr ationRequest.java backend/exception/Cred entialAllreadyInUseExcep tion.java	Fertig
2.	Login eines Benutzers					
2.1	Login im Frontend	Kerem	1.2	2	frontend/src/component s/Login.js	Fertig
2.2	Login im Backend	Nevzat	1.2	2	backend/controller/User Controller.java, backend/service/UserSer vice.java	Fertig

					backend/repository/UserRepository.java, backend/entity/User.java, backend/dto/UserLoginRequest.java backend/exception/InvalidCredentialException.java	
3.	Benutzerprofil e					
3.1	Benutzerprofil e im Frontend	Chantal	3.3	3	frontend/src/components/Profile.js	Fertig
3.2	Suche im Frontend	Chantal			frontend/src/components/Search.js	Fertig
3.3	User Service im Frontend	Kerem	3.4	3	frontend/src/services/UserService.js	Fertig
3.4	Benutzerprofil e im Backend	Abdullah	3.3	3	backend/controller/UserController.java (Zeilen: 77-80), backend/service/UserService.java (Zeilen: 51-54,81,97-121), backend/dto/UserProfileRequest.java, backend/controller/DriveController.java (Zeilen:85-88), backend/service/DriveService.java (Zeilen: 239-315), backend/repository/DriveRepository.java (Zeilen: 22-23), backend/entity/ Drive.js (Zeilen:65-57), backend/entity/User.java (Zeilen:57-59)	Fertig

4.	Kartenvisualisierung von Routen					
4.1	Kartenvisualisierung im Frontend	Kerem		4	frontend/src/components/MapComponent.js	Fertig
5.	Fahranfragen erstellen					
5.1	Routen Eingabe im Frontend	Kerem		5	frontend/src/components/ AddressInput.js , frontend/src/services/GeocodingService.js	Fertig
5.2	Routenplanung im Frontend	Kerem	4.1, 5.1, 5.3	4, 5	frontend/src/components/ RoutePlanner.js , frontend/src/services/RouteService.js	Fertig
5.3	Fahranfragen Service im Frontend	Chantal	5.5	5	frontend/src/services/RideRequestService.js	Fertig
5.4	Aktive Fahranfragen im Frontend	Chantal	1.2, 5.5	6	frontend/src/components/ ActiveRide.js , frontend/src/services/ActiveRideService.js	Fertig
5.5	Fahranfragen im Backend	Nevzat	5.3, 5.4	5, 6	backend/controller/DriveController.java, backend/dto/DriveRequest.java, backend/entity/Drive.java, backend/entity/DriveStatus.java, backend/repository/DriveRepository.java, backend/service/DriveService.java (Zeilen: 1-238)	Fertig
6.	Service-Architektur					

6.1	Frontend-Architektur realisiert	Kerem			frontend/src	Fertig
6.2	Backend-Architektur realisiert	Nevzat			backend/main/java/seprive.gruppen.backend	Fertig
7.	Allgemein					
7.1	Navigationsleiste	Kerem	1.2		frontend/src/components/NavBar.js	Fertig
7.2	User Dashboard im Frontend	Kerem	1.2, 3.3		frontend/src/components/ DriverDashboard.js , frontend/src/components/CustomerDashboard.js	Fertig
7.3	PrivateRoute	Kerem	1.2		frontend/src/components/PrivateRoute.js	Fertig
7.4	Exception Handler	Nevzat			backend/controller/Glob alExceptionHandler.java backend/controller/Valid ationExceptionHandler.java	Fertig

Systemtests

Datum	20.05.2025		
Tester	Simon Lobo		
SW-Version	V 1.0.0		
Vorbedingung(en)			
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / x
1	Der Benutzer gibt die erforderlichen Daten in die Felder ein.	Das System zeigt die Daten in den jeweiligen Feldern an.	v
2	Der Benutzer lädt ein optionales Profilbild hoch.	Das System zeigt den Dateinamen des Bildes an.	v
3	Der Benutzer klickt auf „Registrieren“.	Das System leitet den User zum Login.	v
Nachbedingung(en)	Nutzer im System registriert.		
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Simon Lobo		
SW-Version	V 1.0.0		
Vorbedingung(en)	Nutzer „Max Mustermann“ ist im System angemeldet.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / x
1	Der Benutzer klickt auf „Profil“.	Das System zeigt das Benutzerprofil und alle Details entsprechend der Rolle des Nutzers an.	v

2	Der Benutzer klickt auf „Suche“.	Das System zeigt die Suchfunktion an.	✓
3	Der Benutzer gibt einen Nutzernamen in die Suche ein, der existiert.	Das System zeigt das Profil des gesuchten Benutzers an.	✓
4	Der Benutzer gibt einen Nutzernamen in die Suche ein, der nicht existiert.	Das System zeigt „No user found“.	✓
Nachbedingung(en)	Nutzer kann sein Benutzerprofil und Profile von anderen Nutzern einsehen.		✓
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Simon Lobo		
SW-Version	V 1.0.0		
Vorbedingung(en)	Nutzer „Max Mustermann“ ist im System mit Passwort „geheim“ registriert		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer gibt den Benutzernamen „Max Mustermann“ auf der Tastatur ein.	Das System zeigt „Max Mustermann“ auf dem Display an.	✓
2	Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein.	Das System zeigt das Passwort durch „*“-Symbole zensiert an.	✓
3	Der Benutzer klickt auf „Anmelden“.	Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an.	✓
Nachbedingung(en)	Nutzer ist im System angemeldet.		
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Simon Lobo		
SW-Version	V 1.0.0		
Vorbedingung(en)	Nutzer „Max Mustermann“ ist am System angemeldet und hat keine aktive Fahrt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer klickt auf „Route planen“.	Das System zeigt ein Formular für Fahranfragen und eine Karte an.	✓
2	Der Benutzer wählt die aktuelle Position als Startpunkt.	Das System zeigt die Koordinaten im Startpunkt an und die Karte setzt einen Pin auf die aktuelle Position.	✓
3	Der Benutzer klickt auf „Stop hinzufügen“.	Das System zeigt ein weiteres Feld für einen Stopp an.	✓
4	Der Benutzer gibt in einem beliebigen Feld eine Adresse ein.	Das System zeigt eine Leiste mit Auswahloptionen an.	✓
5	Der Benutzer wählt eine Adresse aus den Auswahloptionen aus.	Das System zeigt die Adresse im jeweiligen Feld an und die Karte setzt einen Pin auf die ausgewählte Position.	✓
6	Der Benutzer gibt in einem beliebigen Feld einen POI ein.	Das System zeigt eine Leiste mit Auswahloptionen an.	✓
7	Der Benutzer wählt einen POI aus den Auswahloptionen aus.	Das System zeigt den POI im jeweiligen Feld an und die Karte setzt einen Pin auf die ausgewählte Position.	✓
8	Der Benutzer klickt in einem beliebigen Feld die Checkbox „Koordinaten“.	Das System zeigt ein Eingabefeld für Koordinaten in der jeweiligen Zeile.	✓
9	Der Benutzer gibt Koordinaten ein und klickt	Das System zeigt die Koordinaten im jeweiligen Feld an und setzt einen Pin auf die jeweilige Position.	✓

	auf "Koordinaten" bestätigen.		
10	Der Benutzer wählt eine Fahrzeugklasse aus.	Das System zeigt im Feld die Fahrzeugklasse an.	✓
11	Der Nutzer klickt auf "Route berechnen".	Das System zeigt in der Karte die Route zwischen Start, optionalen Zwischenstopps und dem Ziel an.	✓
12	Der Nutzer klickt auf "Fahrt anfragen".	Das System zeigt an, dass die Fahrt erfolgreich erstellt wurde.	✓
Nachbedingung(en)	Der Nutzer hat eine Fahrtanfrage erstellt.		
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Simon Lobo		
SW-Version	V 1.0.0		
Vorbedingung(en)	Nutzer „Max Mustermann“ ist im System als Kunde angemeldet und hat eine Fahrt angefragt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer klickt auf "Aktive Fahrt".	Das System zeigt die Seite "Aktive Fahrt" an und der Nutzer sieht Details zu seiner aktiven Fahrt.	✓
2	Der Benutzer klickt auf "Fahrt stornieren".	Das System zeigt an, dass die Fahrt storniert wurde und leitet den Kunden zu seinem Dashboard weiter.	✓
Nachbedingung(en)	Der Nutzer hat seine aktive Fahrt storniert.		
Testurteil	Test bestanden.		

Zyklus II

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Geldkonten			
1.1	User Story 1	User Story	Kerem	fertig
1.2	User Story 2	User Story	Kerem	fertig
1.3	Geldkonten-Prototyp	Papierprototyp	Kerem	fertig
2.	Backend Architektur			
2.1	Komponentendiagramm Backend	Diagramm	Abdullah	fertig
2.2	Klassendiagramm Backend	Diagramm	Kerem	fertig
2.3	Kommunikationsdiagramme Backend	Diagramm	Nevzat	fertig
3.	Frontend			
3.1	Komponentendiagramm Frontend	Diagramm	Chantal	fertig
4.	Liste der Fahrtenfragen			
4.1	User Story 3	User Story	Nevzat	fertig
4.2	User Story 4	User Story	Nevzat	fertig
4.3	Fahrtangebote und Fahrtenfragen-Prototyp	Papierprototyp	Nevzat	fertig
5.	Fahrtplanung			
5.1	User Story 5	User Story	Abdullah	fertig
5.2	Fahrtplanung und Fahrtangebote-Prototyp	Papierprototyp	Abdullah	fertig
6.	Fahrtangebote			
6.1	User Story 6	User Story	Abdullah	fertig
6.2	User Story 7	User Story	Kerem	fertig
7.	Durchführung einer Fahrt			
7.1	User Story 8	User Story	Chantal	fertig

7.2	Durchführung einer Fahrt Prototyp	Papierprototyp	Chantal	fertig
8.	Fahrthistorie			
8.1	User Story 9	User Story	Chantal	fertig
8.2	Fahrthistorie-Prototyp	Papierprototyp	Chantal	fertig

User-Stories

User Story-ID	1
User Story-Beschreibung	Als Kunde möchte ich Geld auf mein SEP-Konto laden, damit ich Fahrten bezahlen kann.
Geschätzter Realisierungsaufwand	3d
Priorität	Hoch
Autor	Kerem
Abhängigkeiten zu anderen User Stories	

User Story-ID	2
User Story-Beschreibung	Als Fahrer möchte ich Geld auf mein SEP-Konto gutgeschrieben bekommen, wenn ich Fahrten erfolgreich abschließe.
Geschätzter Realisierungsaufwand	2d
Priorität	Hoch
Autor	Kerem
Abhängigkeiten zu anderen User Stories	US-1

User Story-ID	3
User Story-Beschreibung	Als Fahrer möchte ich eine Liste mit offenen Fahranfragen sehen, um passende Fahrten auswählen zu können.
Geschätzter Realisierungsaufwand	3d
Priorität	Hoch
Autor	Nevzat
Abhängigkeiten zu anderen User Stories	

User Story-ID	4
User Story-Beschreibung	Als Fahrer möchte ich die Fahranfragen nach Distanz, Datum, Kunde oder Fahrzeugklasse sortieren können, um schnell relevante Fahrten zu finden.
Geschätzter Realisierungsaufwand	1d
Priorität	Niedrig
Autor	Nevzat
Abhängigkeiten zu anderen User Stories	US-3

User Story-ID	5
User Story-Beschreibung	Als Kunde möchte ich die geplante Route, Distanz, Dauer und Fahrtpreis meiner Anfrage sehen, damit ich informiert bin, bevor ich ein Angebot akzeptiere.
Geschätzter Realisierungsaufwand	3d
Priorität	Mittel
Autor	Abdullah
Abhängigkeiten zu anderen User Stories	

User Story-ID	6
User Story-Beschreibung	Als Fahrer möchte ich eine Fahranfrage übernehmen und dem Kunden ein Angebot machen, damit ich die Fahrt durchführen kann.
Geschätzter Realisierungsaufwand	4d
Priorität	Hoch
Autor	Abdullah
Abhängigkeiten zu anderen User Stories	US-3

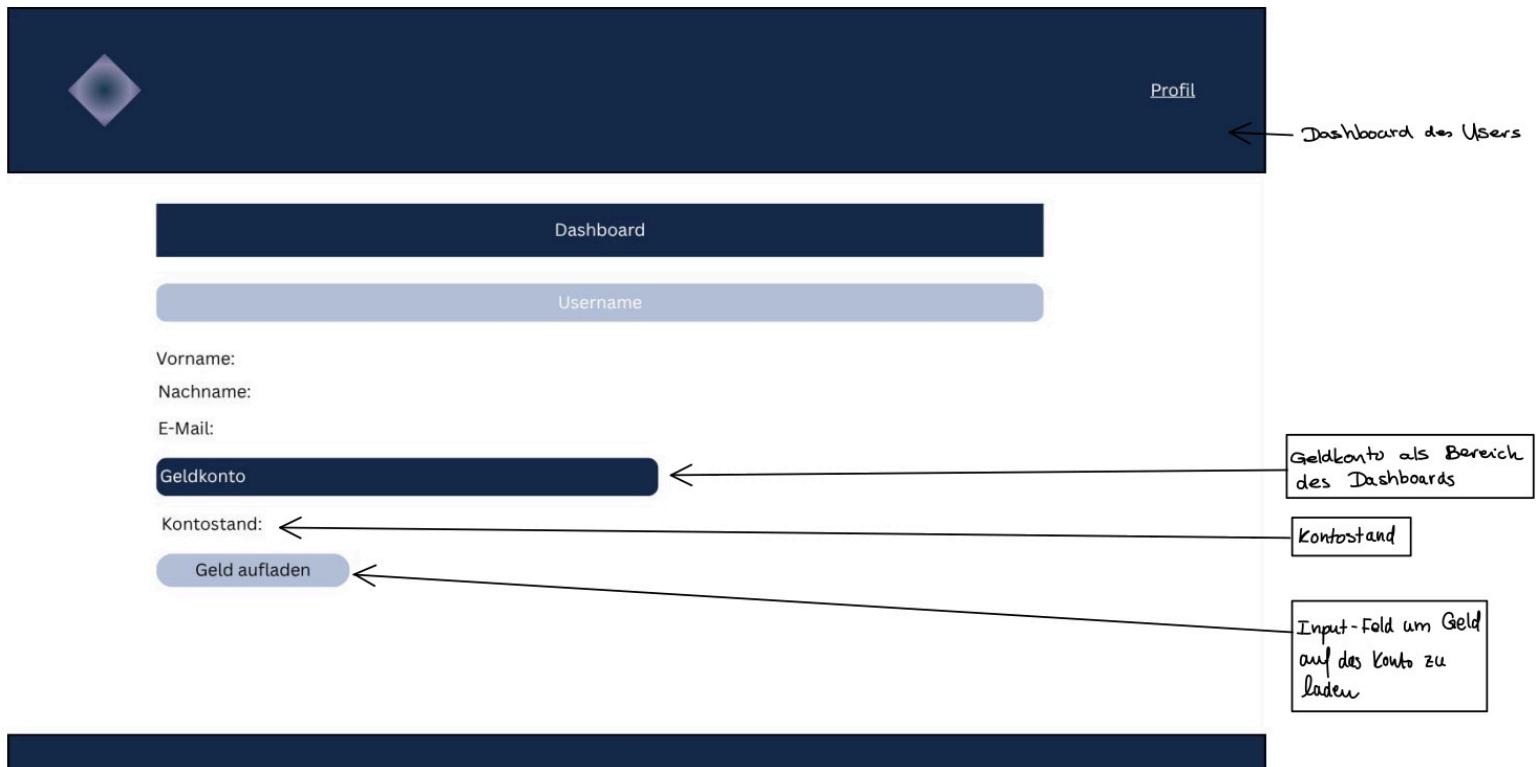
User Story-ID	7
User Story-Beschreibung	Als Kunde möchte ich Angebote von Fahrern sehen und eins davon annehmen oder ablehnen, um eine passende Fahrt zu finden.
Geschätzter Realisierungsaufwand	3d
Priorität	Hoch
Autor	Kerem
Abhängigkeiten zu anderen User Stories	US-6

User Story-ID	8
User Story-Beschreibung	Als Benutzer möchte ich die Fahrt auf einer Karte verfolgen und steuern können (Start, Pause, Geschwindigkeit), um den Fahrtverlauf visuell nachvollziehen zu können.
Geschätzter Realisierungsaufwand	4d
Priorität	Hoch
Autor	Chantal
Abhängigkeiten zu anderen User Stories	

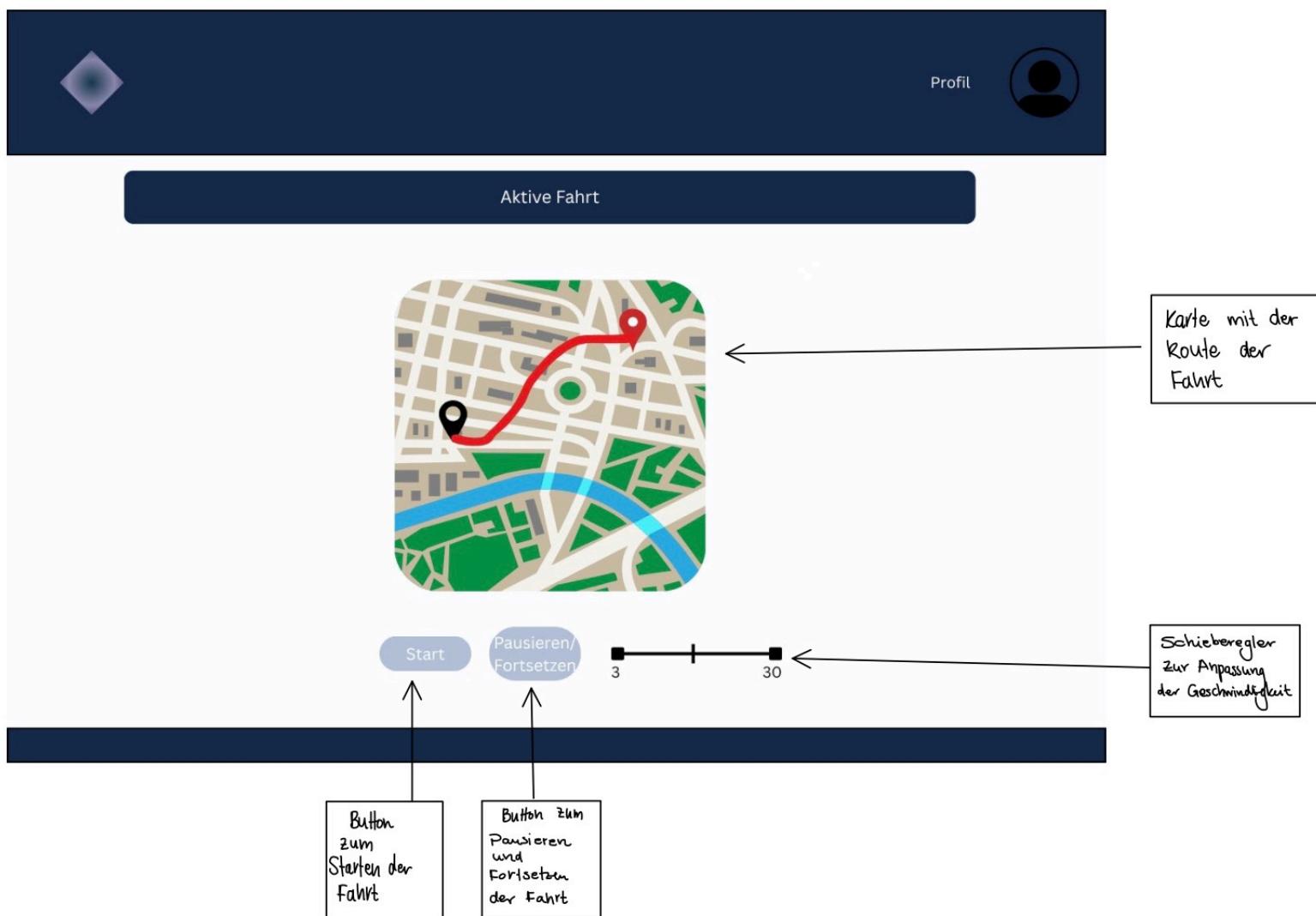
User Story-ID	9
User Story-Beschreibung	Als Benutzer möchte ich meine abgeschlossenen Fahrten mit Details wie Distanz, Dauer, Kosten und Bewertung sehen, damit ich meine Aktivität nachvollziehen kann.
Geschätzter Realisierungsaufwand	2d
Priorität	Mittel
Autor	Chantal
Abhängigkeiten zu anderen User Stories	

Papierprototypen

Geldkonten



Durchführung einer Fahrt



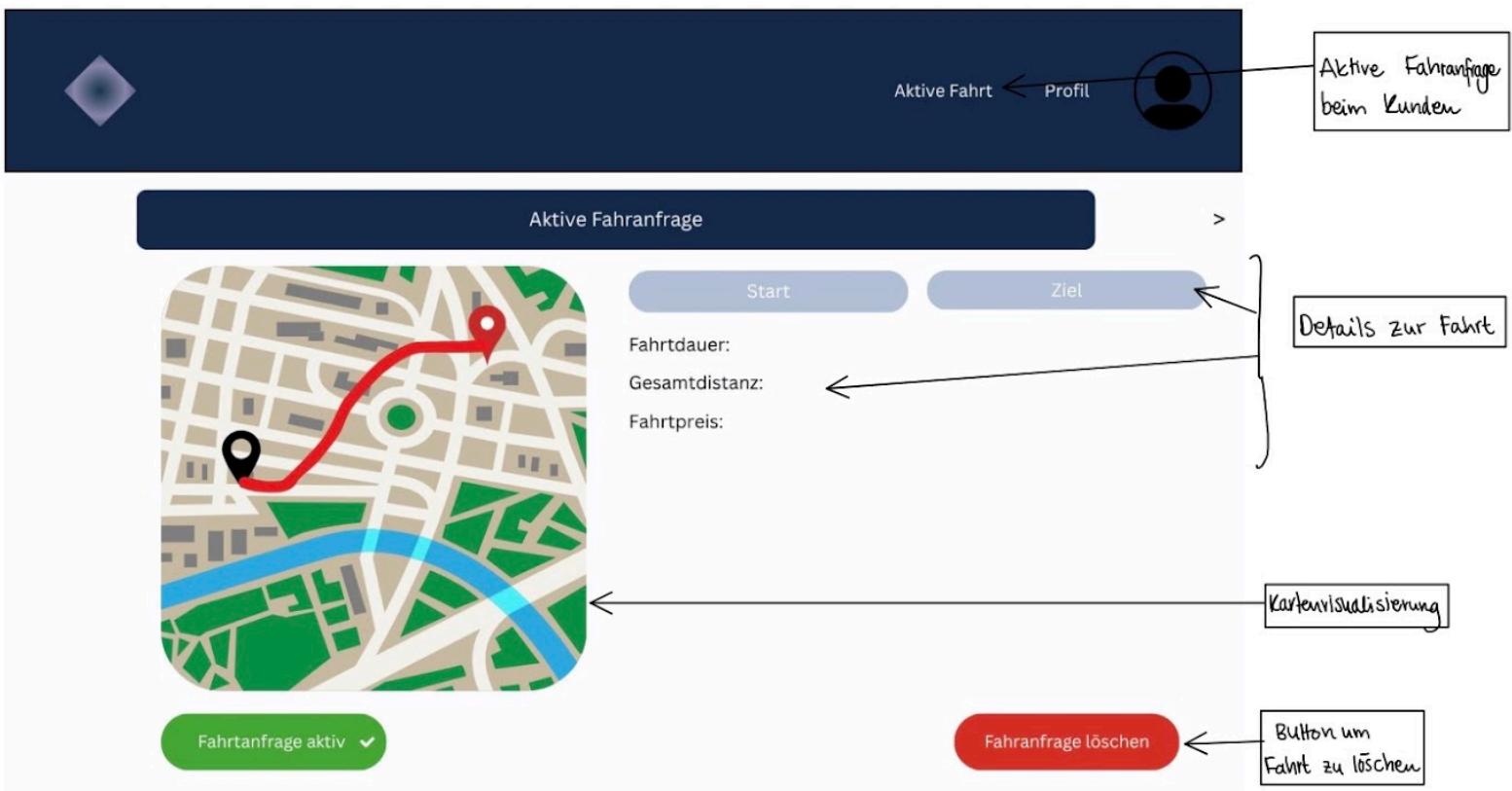
Fahrtangebote und Fahrtanfragen

The screenshot shows a mobile application interface. At the top, there is a dark blue header with a profile icon and the text "Fahranfragen" and "Profil". Below the header is a teal-colored section titled "Fahranfragen". This section contains a table with columns: Fahrt-ID, Zeitpunkt der Erstellung, Entfernung, Kundenname, Kundenbewertung, Fahrzeugklasse, Distanz, Dauer, and Preis. To the right of the table are three green circular buttons with checkmarks, each labeled "Buttons zum Annehmen einer Fahrt". Above the table, a grey rounded rectangle labeled "Startpunkt" has an arrow pointing to it from a text box labeled "Eingabe des Standorts". A text box labeled "Liste der aktiven Fahranfragen" also points to the table. On the far left, a small purple diamond icon is visible.

This screenshot shows a mobile application interface. At the top, there is a dark blue header with a profile icon and the text "Angenommene Fahranfrage" and "Profil". Below the header is a dark blue section titled "Angenommene Fahranfrage". A text box labeled "Button, falls Fahrer eine Anfrage annimmt" points to the profile icon. At the bottom, there is a red rounded rectangle containing the text "Angebot zurückziehen".

This screenshot shows a mobile application interface. At the top, there is a dark blue header with a profile icon and the text "Angenommene Fahranfrage" and "Profil". Below the header is a dark blue section titled "Angenommene Fahranfrage". A text box labeled "Informationen über die angenommene Fahrt" points to the profile icon. At the bottom, there is a red rounded rectangle containing the text "Angebot zurückziehen".

Fahrtplanung und Fahrtangebote



The screenshot shows a mobile application interface for viewing driver offers. At the top, there is a header bar with the text "Fahrer Angebote". Below the header is a table with columns: "Fahrer", "Bewertung", "Gesamtdistanz", and "Gesamtanzahl der Fahrten". A bracket on the right side groups these column headers under the heading "Sortier - Optionen". In the body of the table, there are three rows, each containing two green "Annehmen" (Accept) buttons and two red "Ablehnen" (Reject) buttons. A bracket on the right side groups these buttons under the heading "Button, um einen Fahrer abzulehnen". Below the table, there is a large green button with an upward arrow, which is grouped under the heading "Button, um einen Fahrer anzunehmen".

Fahrthistorie

The screenshot shows a dark blue header bar with a diamond-shaped logo on the left and a user profile icon with the word "Profil" on the right. Below the header is a search bar containing the text "Suche" and a magnifying glass icon. The main content area has a teal header bar with the title "Fahrthistorie". Below the header is a table with the following columns:

Fahrt-ID	Abschlusszeitpunkt	Distanz	Fahrtzeit	Betrag	Bewertung (Kunde)	Bewertung (Fahrer)	Name (Kunde)

Annotations on the right side of the screen:

- A callout box points to the user profile icon with the text "Fahrthistorie im Profil abrufbar".
- A callout box points to the search bar with the text "Such-Option (für Namen)".
- A callout box points to the table with the text "Tabelle mit vergangenen Fahrten und den jeweiligen Informationen".
- A callout box points to the bottom right corner of the table with the text "Sortier - Optionen".

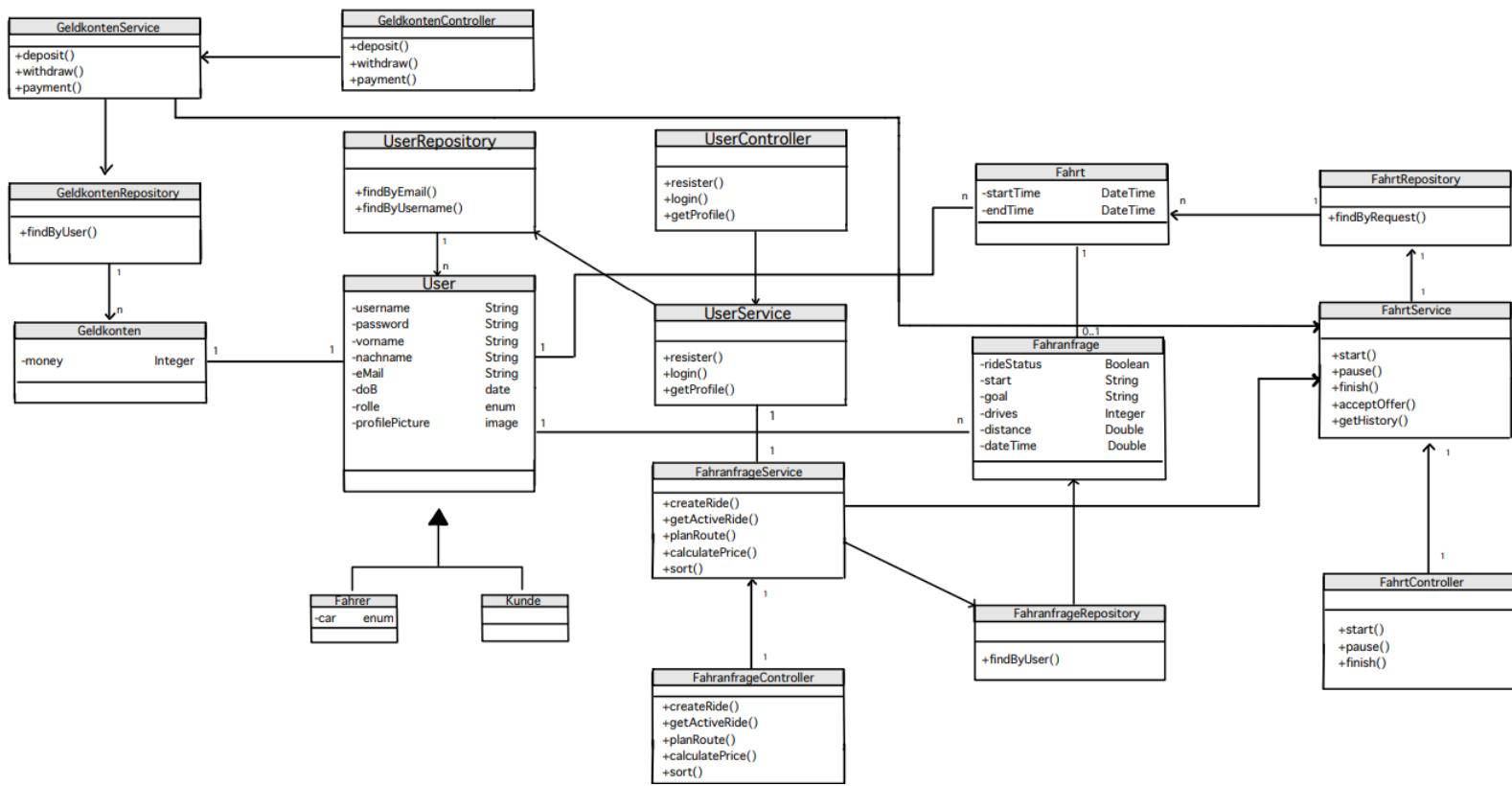
The screenshot shows a continuation of the table from the previous screen. The header bar is dark blue with a diamond-shaped logo and a user profile icon labeled "Profil". The main content area has a teal header bar with the title "Fahrthistorie". Below the header is a table with the following columns:

Benutzername (Kunde)	Benutzername (Fahrer)	Name (Fahrer)

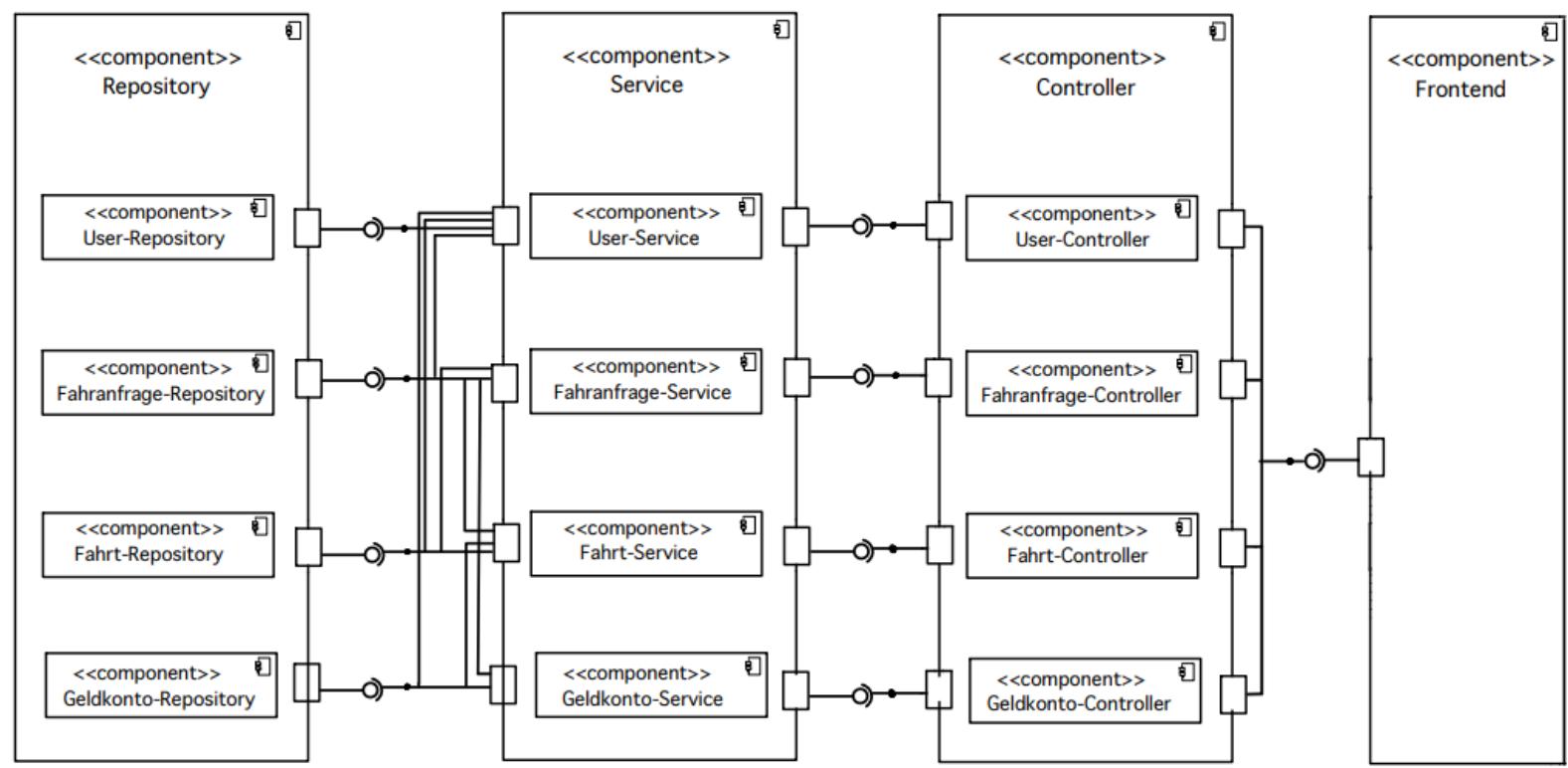
A large arrow points from the right side of the previous table towards the start of this table, indicating a continuation. A callout box on the right side of the screen points to this arrow with the text "Weiter-führung der vorherigen Tabelle (nach rechts)".

Strukturierte Diagramme (Komponenten- und Klassendiagramme)

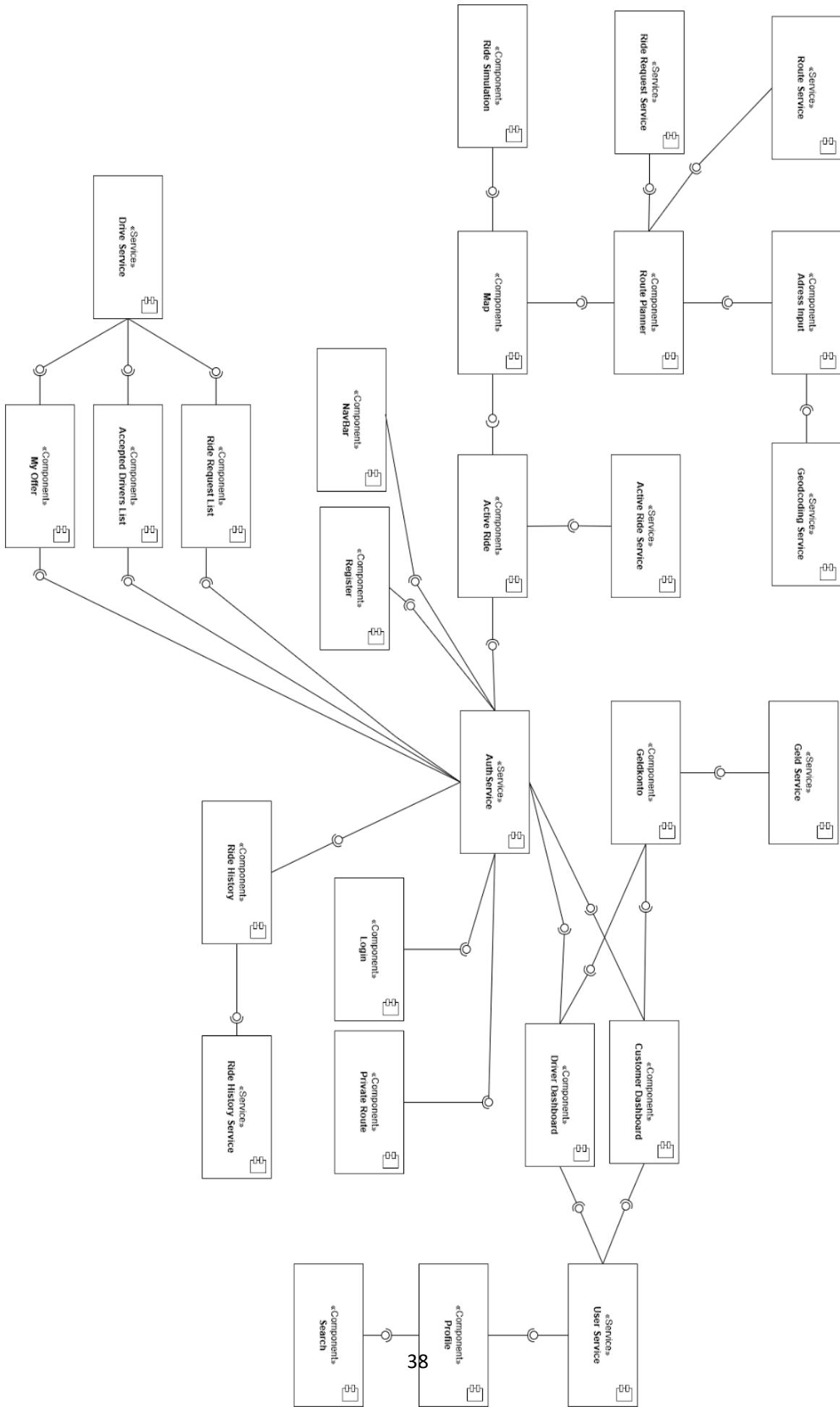
Klassendiagramm:



Komponentendiagramm (Backend):

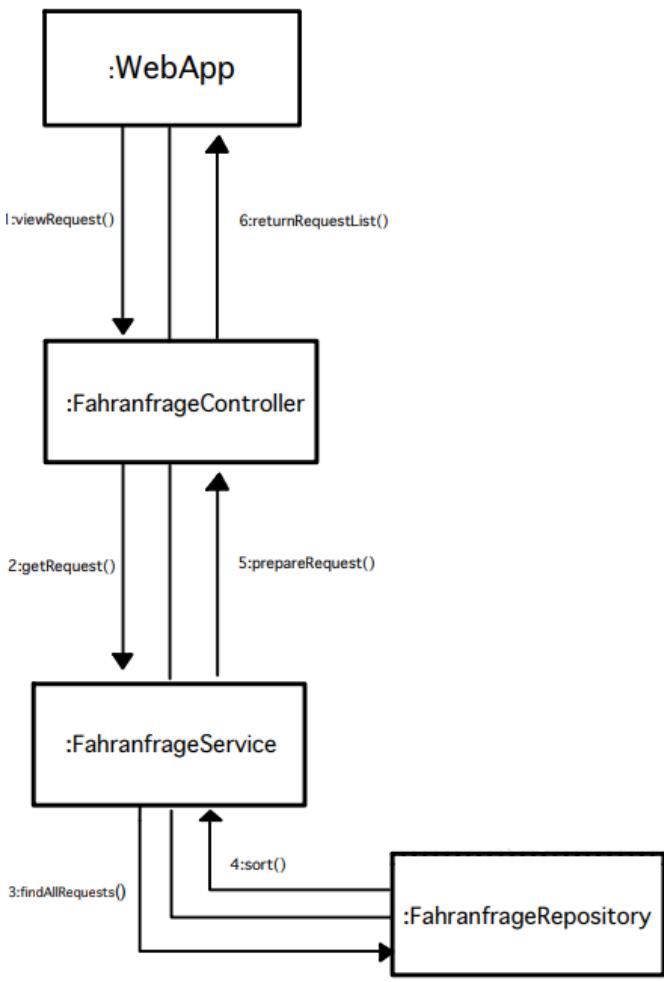


Komponentendiagramm (Frontend)

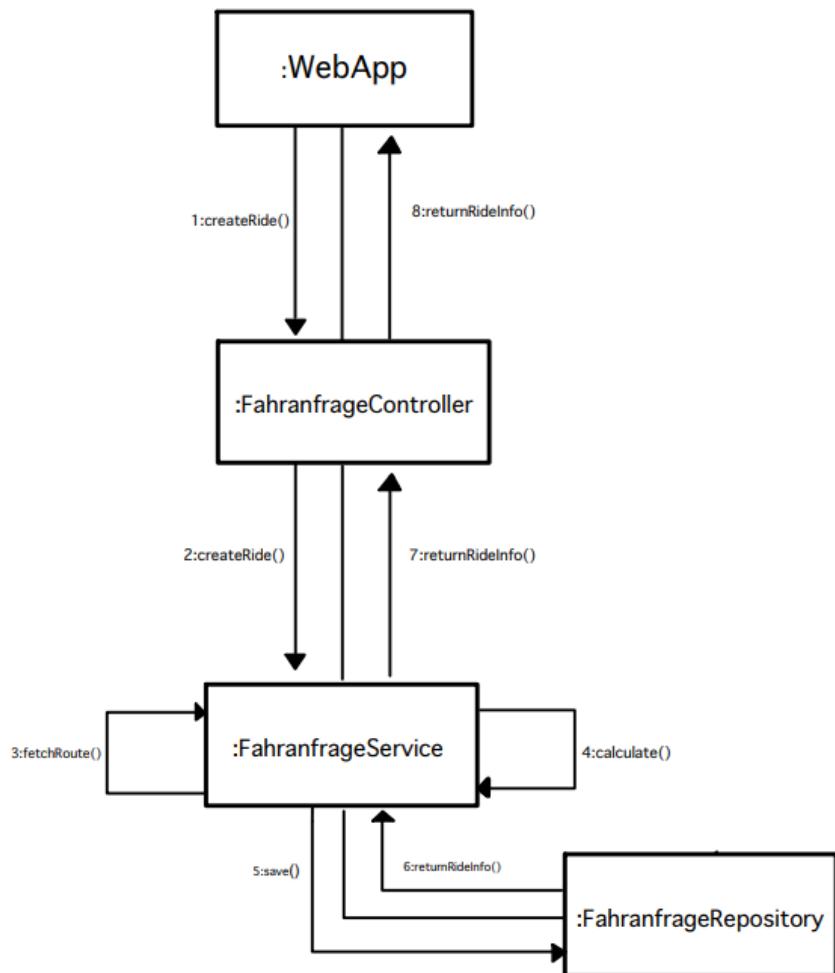


Verhaltensdiagramme (Kommunikationsdiagramme)

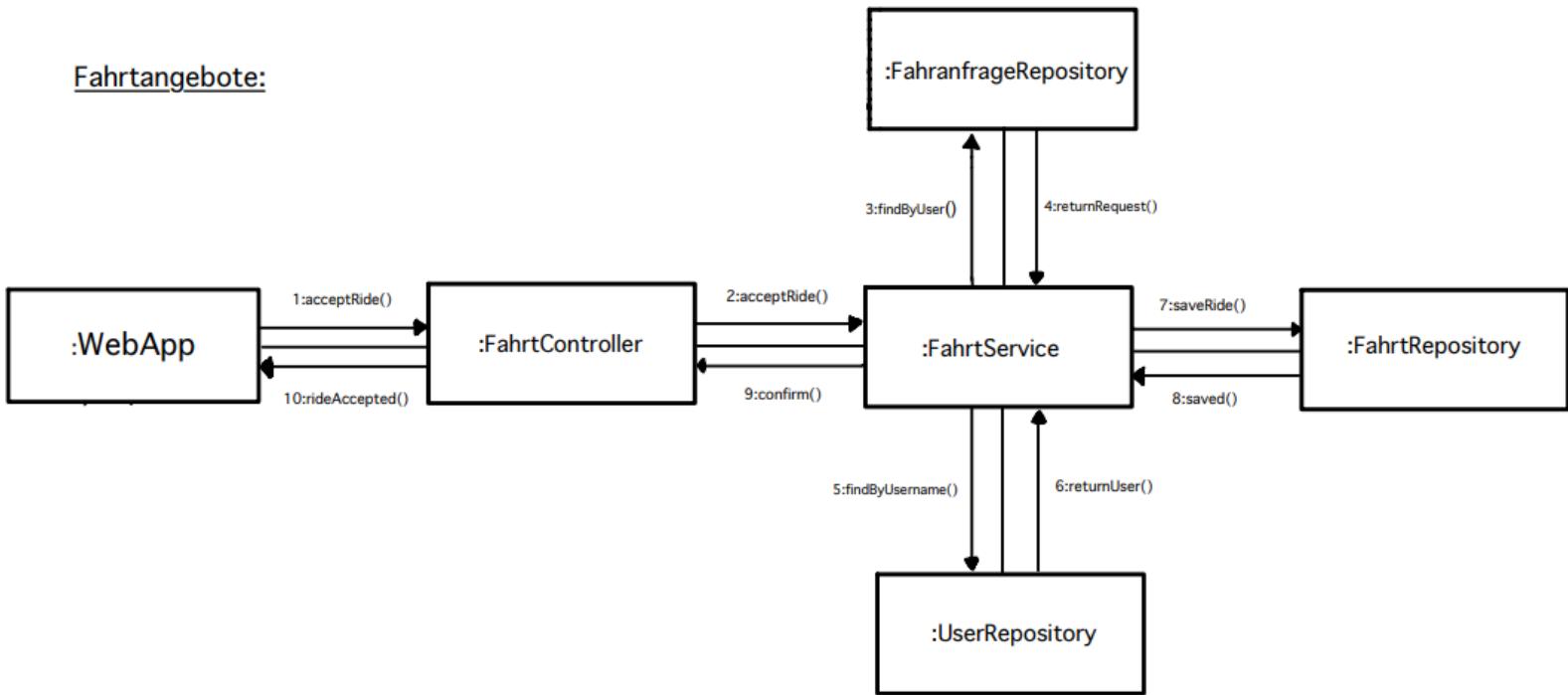
Liste der Fahrten:



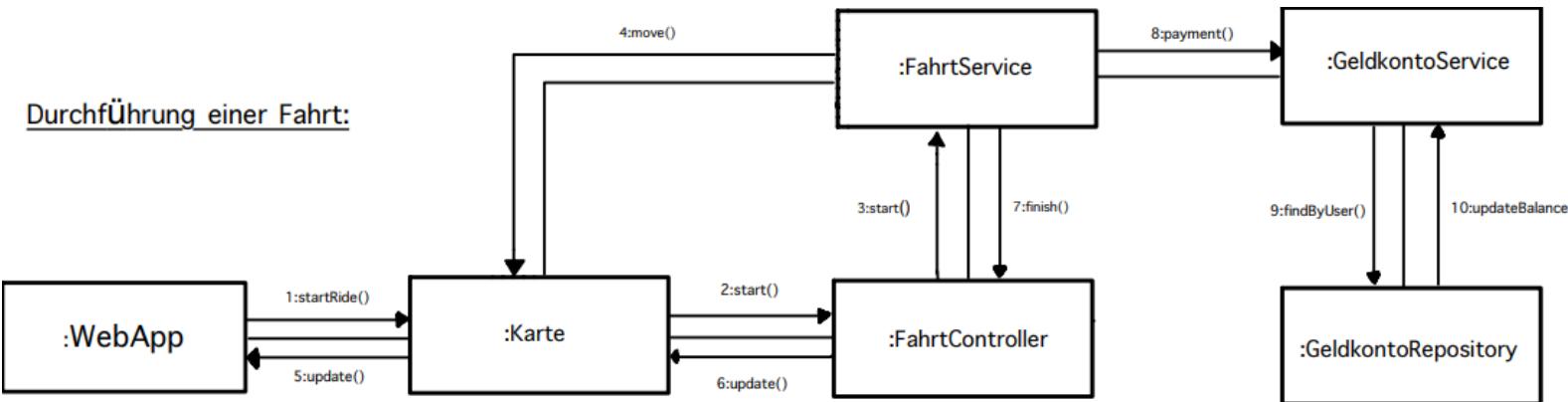
Fahrtplanung:



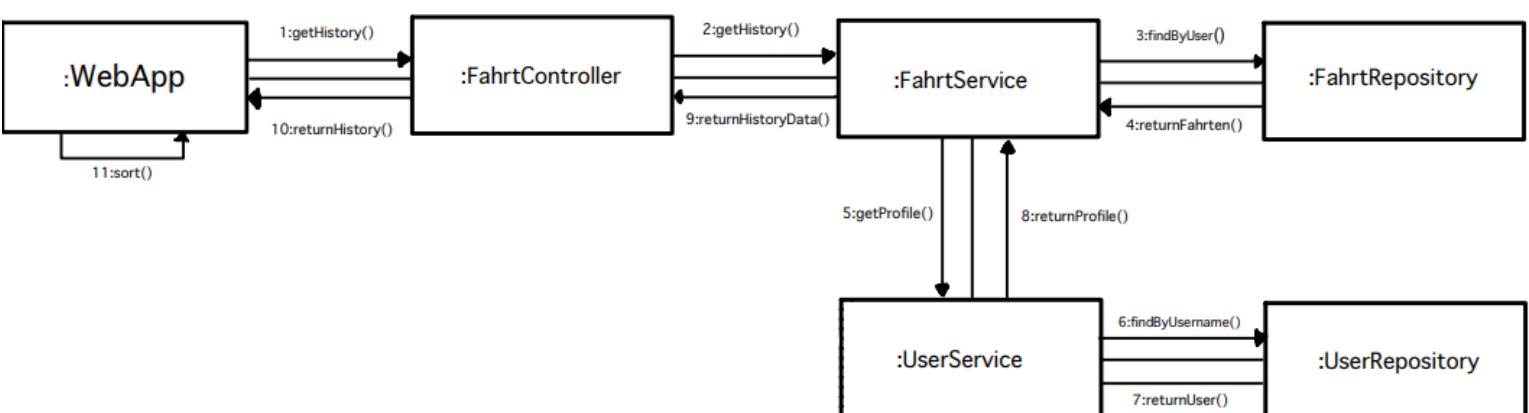
Fahrtangebote:



Durchführung einer Fahrt:



Fahrtangebote:



Funktionalitätsplanung

ID	Funktionalität	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
1.	Geldkonten					
1.1	Geldkonten im Frontend	Chantal	1.2	1,2	frontend/src/components/ Wallet.js , frontend/src/services/ WalletService.js	Fertig
1.2	Geldkonten im Backend	Abdullah	1.1	1,2	backend/entity/Wallet.java, backend/entity/WalletTransaction.java, backend/repository/WalletRepository.java, , backend/repository/WalletTransactionRepository.java, backend/service/WalletService.java, backend/controller/WalletController.java, backend/dto/DepositRequest.java backend/dto/TransferRequest.java	Fertig
2.	Liste der verfügbaren Fahranfragen					
2.1	Liste der Fahranfragen im Frontend	Kerem	2.2	3,4	frontend/src/components/ RideRequestList.js frontend/src/services/ RouteService.js	Fertig

					frontend/src/services/ /DriveService.js frontend/src/utils/sortUtils.js	
2.2	Fahranfragen im Backend	Nevzat	2.1	3,4	backend/config/WebSocketConfig.java backend/controller/DriveSimulationController.java	Fertig
3.	Fahrtplanung					
3.1	Fahrtplanung im Frontend	Chantal	3.2	5	frontend/src/components/ /ActiveRide.js frontend/src/services/RideRequestService.js frontend/src/components/RoutePlanner.js (Zeilen:19,20,96,119,126,127,128,150-152, 251) frontend/src/services/ /RouteService.js (Zeile 26)	Fertig
3.2	Fahrtplanung im Backend	Nevzat	3.1	5	backend/controller/DriveController.java, backend/service/DriveService.java (Zeilen: 1-238), backend/repository/DriveRepository.java, backend/entity/Drive.java, backend/entity/Point.java	Fertig
4.	Fahrtangebote					
4.1	Fahrtangebote im Frontend	Kerem	4.2	6,7	frontend/src/components/ /AcceptedDriversList.js frontend/src/components/RideRequestList.js	Fertig

					frontend/src/services/ DriveService.js frontend/src/utils/ sortUtils.js	
4.2	Fahrtangebote im Backend	Nevzat	4.1	6,7	backend/controller/DriveOfferController.java, backend/entity/DriveOffer.java, backend/entity/DriveOfferStatus.java, backend/repository/DriveOfferRepository.java, backend/dto/DriveOfferNotificationResponse.java, backend/dto/DriveOfferRequest.java, backend/dto/DriveOfferResponse.java, backend/dto/DriveOfferWithDriveResponse.java, backend/service/DriveOfferService.java backend/service/NotificationSocketService.java	Fertig
5.	Durchführung einer Fahrt					
5.1	Durchführung einer Fahrt im Frontend	Kerem	5.2	8	frontend/src/components/ DriverSimulation.js frontend/src/components/ MapComponent.js	Fertig

					frontend/src/components/ CustomerSimulation.js frontend/src/websocket/ WebSocketService.js frontend/src/components/ StatusMonitor.js frontend/src/services/ DriveService.js	
5.2	Durchführung einer Fahrt im Backend	Nevzat	5.1	8	backend/config/WebSocketConfig.java, backend/service/DriveSimulationService.java	Fertig
6.	Fahrthistorie					
6.1	Fahrthistorie im Frontend	Chantal	6.2	9	frontend/src/components/ RideHistory.js , frontend/src/services/ RideHistoryService.js	Fertig
6.2	Fahrthistorie im Backend	Abdullah	6.1	9	backend/service/DriveService.java (Zeilen: 239-315)	Fertig

Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status
T1	Gültige Einzahlung	WalletServiceTest.doDeposit_validAmount_updatesWalletAndCreatesTransaction()	bestanden
T2	Geldüberweisung gültig	WalletServiceTest.transfer_validTransfer_worksCorrectly()	bestanden
T3	Unzureichendes Guthaben	WalletServiceTest.transfer_InsufficientFunds_throwsException()	bestanden
T4	Wallet nicht gefunden	WalletServiceTest.doDeposit_walletNotFound_throwsException()	bestanden

Systemtests

Datum	17.06.2025		
Tester	Chantal Jedrschyk		
SW-Version	V 2.0.0		
Vorbedingungen	Der Nutzer ist als Kunde im System angemeldet		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / x
1	Der Nutzer klickt auf sein Dashboard.	Das System zeigt das Dashboard mit dem Kontostand und einer Option zum Aufladen des Geldkontos an.	v
2	Der Nutzer gibt einen Geldbetrag ein und lädt diesen hoch.	Das System bestätigt den Betrag und der Kontostand aktualisiert sich um den Betrag.	v
Nachbedingungen	Der Kunde hat Geld auf seinem Konto.		v
Testurteil	Bestanden		

Datum	17.06.2025		
Tester	Chantal Jedrschyk		
SW-Version	V 2.0.0		
Vorbedingung(en)	Der Nutzer ist als Fahrer im System angemeldet und es wurden mindestens zwei Fahranfragen von Kunden erstellt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer klickt auf "Fahranfragen" in der Navigationsleiste.	Das System zeigt eine Tabelle mit verfügbaren Fahranfragen an.	✓
2	Der Nutzer klickt auf das Sortiersymbol einer beliebigen Spalte.	Das System sortiert die jeweilige Spalte aufsteigend/absteigend.	✓
3	Der Nutzer wählt die aktuelle Position als Startpunkt.	Das System zeigt in der Spalte "Entfernung" die Distanz zwischen dem Fahrer und dem Startpunkt der Fahranfrage.	✓
4	Der Nutzer gibt in dem Suchfeld eine Adresse ein.	Das System zeigt eine Leiste mit Auswahloptionen an.	✓
5	Der Nutzer wählt eine Adresse aus den Auswahloptionen aus.	Das System zeigt in der Spalte "Entfernung" die Distanz zwischen der ausgewählten Adresse und dem Startpunkt der Fahranfrage.	✓
6	Der Nutzer gibt in dem Suchfeld einen POI ein.	Das System zeigt eine Leiste mit Auswahloptionen an.	✓
7	Der Nutzer wählt einen POI aus den Auswahloptionen aus.	Das System zeigt in der Spalte "Entfernung" die Distanz zwischen dem ausgewählten POI und dem Startpunkt der Fahranfrage.	✓
8	Der Nutzer klickt auf die Checkbox "Koordinaten".	Das System zeigt ein Eingabefeld für Koordinaten an.	✓
9	Der Nutzer gibt Koordinaten ein und klickt auf "Koordinaten bestätigen".	Das System zeigt in der Spalte "Entfernung" die Distanz zwischen den ausgewählten Koordinaten und dem Startpunkt der Fahranfrage.	✓
10.	Der Nutzer klickt auf "Anbieten".	Das System benachrichtigt den Nutzer, dass das Fahrangebot erfolgreich gesendet wurde.	✓
Nachbedingung(en)	Der Fahrer hat ein Fahrangebot angenommen.		

Testurteil	Bestanden
-------------------	-----------

Datum	17.06.2025		
Tester	Chantal Jedrschyk		
SW-Version	V 2.0.0		
Vorbedingung(en)	Der Nutzer ist als Kunde im System angemeldet.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / x
1	Der Nutzer klickt auf "Route planen".	Das System zeigt ein Formular für Fahranfragen und eine Karte an.	v
2	Der Nutzer gibt seinen gewünschten Start- und Zielpunkt an und klickt auf "Route berechnen".	Das System stellt die Route auf der Karte dar und zeigt die Gesamtdistanz, die Dauer und den Preis der Fahrt an.	v
3	Der Nutzer klickt auf "Fahrt anfragen".	Das System zeigt an, dass die Fahrt erfolgreich erstellt wurde.	v
4	Der Nutzer klickt auf "Aktive Fahrt"	Das System zeigt die Seite "Aktive Fahrt" an und der Nutzer sieht die Karte mit seiner Route, die Gesamtdistanz, die Dauer und den Preis der Fahrt.	v
Nachbedingung(en)	Der Nutzer hat eine Aktive Fahranfrage.		
Testurteil	Bestanden		

Datum	17.06.2025		
Tester	Chantal Jedrschyk		
SW-Version	V 2.0.0		
Vorbedingung(en)	Die Nutzer sind als Kunde und als Fahrer im System angemeldet und der Kunde hat eine Fahranfrage erstellt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / x

1	Der Fahrer klickt auf "Fahranfragen".	Das System zeigt eine Liste der aktiven Fahranfragen an, unter denen auch die des angemeldeten Kunden ist.	✓
2	Der Fahrer wählt die Fahranfrage des Kunden aus und klickt auf "Anbieten".	Das System benachrichtigt den Fahrer, dass die Fahrt erfolgreich angeboten wurde und es benachrichtigt den Kunden, dass ein neues Fahrerangebot vorliegt.	✓
3	Der Kunde klickt auf "Fahrerangebote".	Das System zeigt dem Kunden die Details zu den Fahrern, die seine Fahrt angenommen haben.	✓
4.1	Der Kunde klickt auf "Annehmen".	Das System ändert den Status des Angebots auf "Accepted".	✓
4.2	Der Kunde klickt auf "Ablehnen".	Das System ändert den Status des Angebots auf "Rejected".	✓
Nachbedingung(en)			
Testurteil	Bestanden		

Datum	17.06.2025		
Tester	Chantal Jedrschyk		
SW-Version	V 2.0.0		
Vorbedingung(en)	Der Nutzer ist als Fahrer angemeldet und verfügt über ein angenommenes Fahrangebot.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer klickt auf "Fahrtansicht"	Das System zeigt die Seite der Fahrt-Simulation an.	✓
2	Der Nutzer setzt den Schieberegler auf 30 Sekunden und klickt auf "Start".	Das System startet die Simulation der Fahrt auf der Karte.	✓
3	Der Nutzer klickt auf "Pause".	Das System pausiert die Fahrt.	✓
4	Der Nutzer klickt auf "Fortsetzen".	Das System führt die Fahrtsimulation fort.	✓

5	Der Nutzer setzt den Schieberegler auf 5 Sekunden.	Die Fahrtsimulation wird schneller.	✓
6	Der Nutzer klickt auf "Fahrt abschließen".	Das System benachrichtigt den Nutzer, dass die Fahrt abgeschlossen ist.	✓
7	Der Nutzer klickt auf sein Dashboard.	Das System zeigt den Kontostand des Fahrers an, der jetzt den Betrag der Fahrt beinhaltet.	✓
Nachbedingung(en)	Im Profil und in der Fahrthistorie des Fahrers ist eine weitere Fahrt verzeichnet und der Fahrpreis wurde dem Geldkonto gutgeschrieben.		
Testurteil	Bestanden		

Datum	17.06.2025		
Tester	Chantal Jedrschyk		
SW-Version	V 2.0.0		
Vorbedingung(en)	Der Nutzer ist im System angemeldet und hat eine Fahrtsimulation abgeschlossen		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer klickt in seinem Profil auf Fahrthistorie.	Das System zeigt eine Tabelle mit den abgeschlossenen Fahrten des Nutzers an.	✓
2	Der Nutzer klickt auf das Sortiersymbol einer beliebigen Spalte.	Das System sortiert die ausgewählte Spalte aufsteigend/absteigend.	✓
3	Der Nutzer gibt in der Suchleiste einen Namen ein.	Das System filtert die Tabelle nach dem eingegebenen Namen.	✓
Nachbedingung(en)			
Testurteil	Bestanden		

Zyklus III

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1	Backend			
1.1	Komponentendiagramm Backend	Diagramm	Nevzat	Fertig
1.2	Klassendiagramm Backend	Diagramm	Abdullah	Fertig
1.3	Kommunikationsdiagramme Backend	Diagramm	Abdullah	Fertig
2	Frontend			
2.1	Komponentendiagramm Frontend	Diagramm	Kerem	Fertig
3	Chat			
3.1	User Story 1	User Story	Nevzat	Fertig
3.2	User Story 5	User Story	Nevzat	Fertig
3.2	Chat Fenster (Papierprototyp 1)	Papierprototyp	Kerem	Fertig
4	Fahrten mit mehreren Zwischenstopps			
4.1	User Story 2	User Story	Abdullah	Fertig
4.2	Papierprototyp 2	Papierprototyp	Chantal	Fertig
5	Grafische Statistiken			
5.1	User Story 3	User Story	Nevzat	Fertig
5.2	Statistiken Fenster (Papierprototyp 3)	Papierprototyp	Chantal	Fertig

6	Live Änderungen während der Fahrt			
6.1	User Story 4	User Story	Nevzat	Fertig
6.2	Papierprototyp 4	Papierprototyp	Chantal	Fertig

User-Stories

User Story-ID	1
User Story-Beschreibung	Als Fahrer möchte ich nach Abgabe eines Fahrangebots mit meinem potentiellen Kunden in einem Chat kommunizieren können, damit ich Details zur Fahrt besprechen oder Rückfragen stellen kann.
Geschätzter Realisierungsaufwand	5d
Priorität	hoch
Autor	Nevzat
Abhängigkeiten zu anderen User Stories	

User Story-ID	2
User Story-Beschreibung	Als Kunde möchte ich beim Erstellen einer Fahranfrage mehrere Zwischenstopps angeben können, damit alle meine gewünschten Orte abgefahren werden können.
Geschätzter Realisierungsaufwand	2d
Priorität	mittel
Autor	Abdullah
Abhängigkeiten zu anderen User Stories	

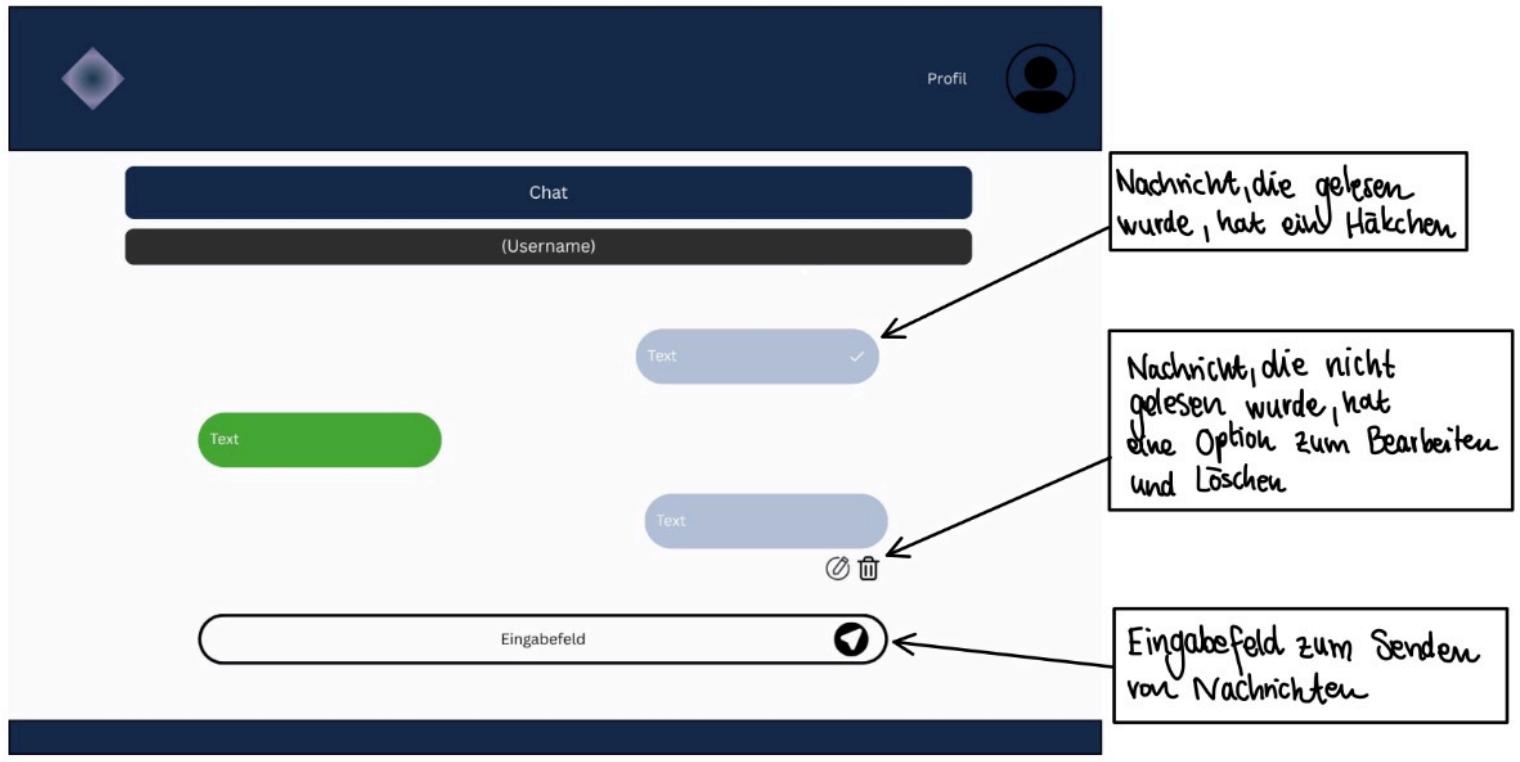
User Story-ID	3
User Story-Beschreibung	Als Fahrer möchte ich Diagramme zu meinen bisherigen Fahrten und Leistungen einsehen können, damit ich meine Einnahmen, gefahrene Distanz, Fahrzeit und meine durchschnittliche Bewertung analysieren kann.
Geschätzter Realisierungsaufwand	2d
Priorität	mittel
Autor	Nevzat
Abhängigkeiten zu anderen User Stories	

User Story-ID	4
User Story-Beschreibung	Als Kunde möchte ich während einer laufenden Fahrsimulation die Simulation pausieren können, um das Ziel oder Wegpunkte zu ändern, damit ich meine Fahrt flexibel anpassen kann und die Simulation anschließend mit der neuen Route fortsetzen kann.
Geschätzter Realisierungsaufwand	3d
Priorität	hoch
Autor	Nevzat
Abhängigkeiten zu anderen User Stories	

User Story-ID	5
User Story-Beschreibung	Als Benutzer möchte ich meine Nachrichten im Chat bearbeiten oder löschen können, solange mein Chatpartner diese nicht gelesen hat.
Geschätzter Realisierungsaufwand	2d
Priorität	hoch
Autor	Nevzat
Abhängigkeiten zu anderen User Stories	1

Papierprototypen

Chat

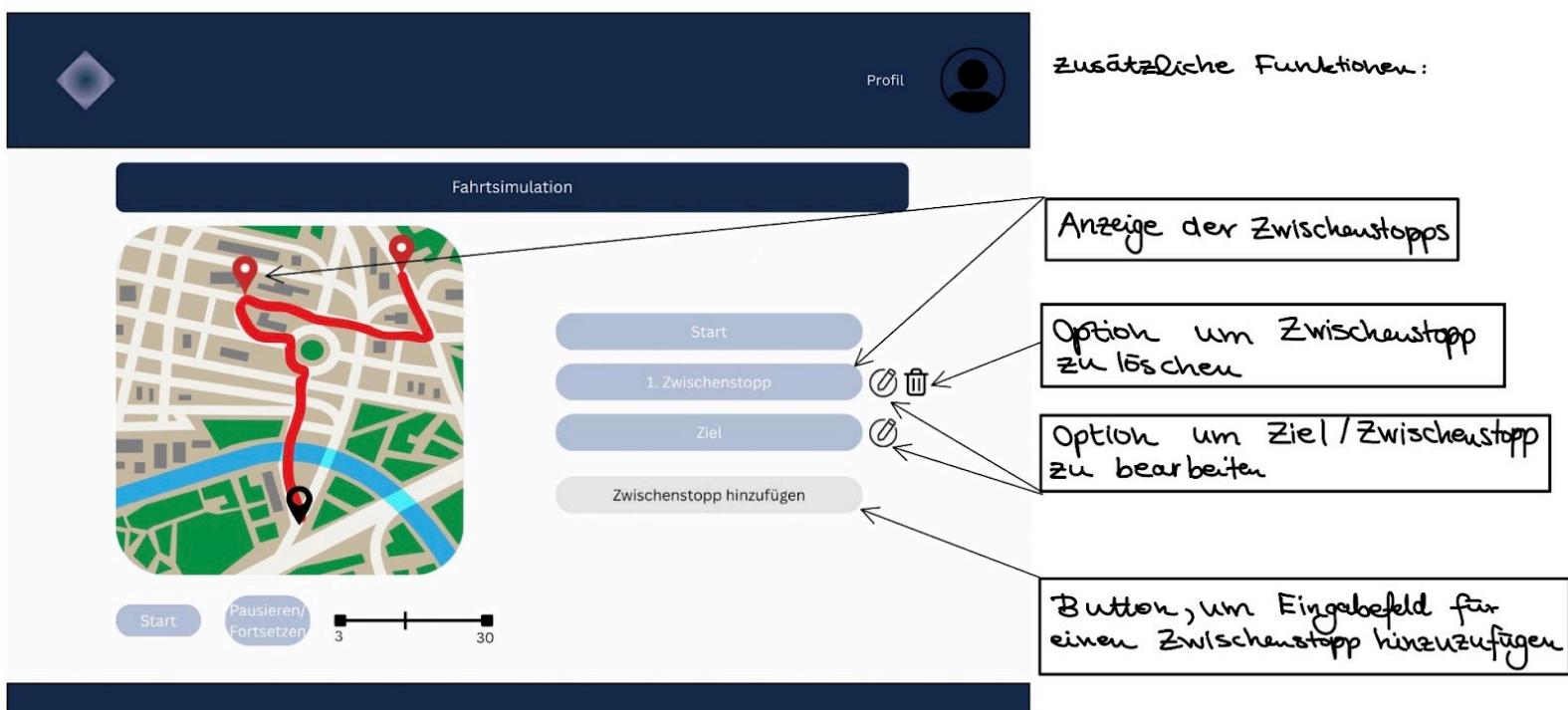


Fahrer	Gesamtdistanz	Gesamtanzahl der Fahrten

Annehmen Ablehnen Chat
Annehmen Ablehnen Chat
Annehmen Ablehnen Chat

Button zur Chat-Funktion neben dem Angebot des Fahrers

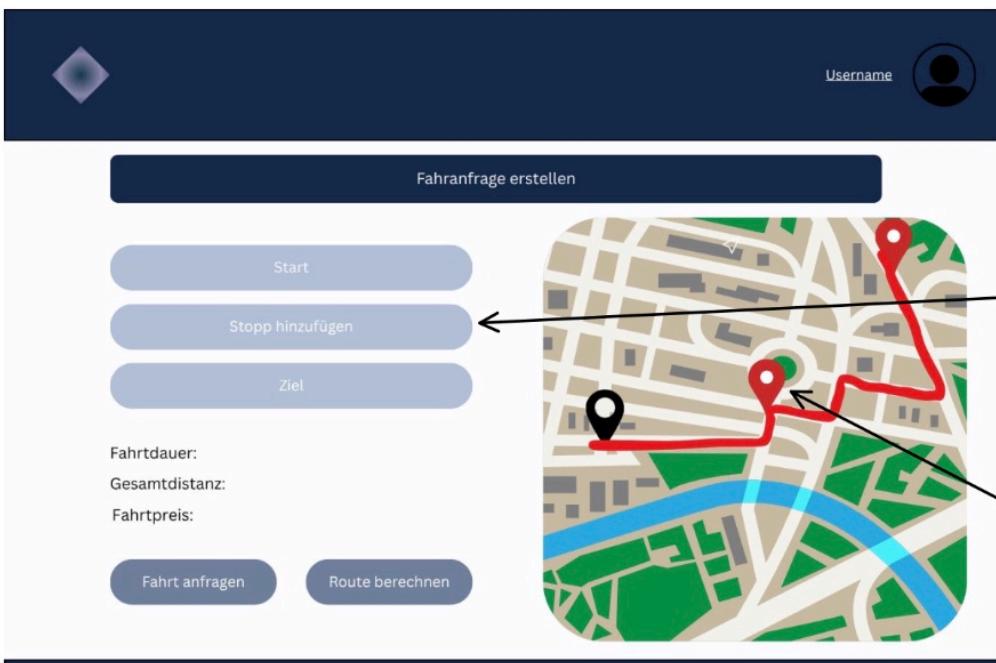
Live-Änderungen während der Fahrt



Grafische Statistiken



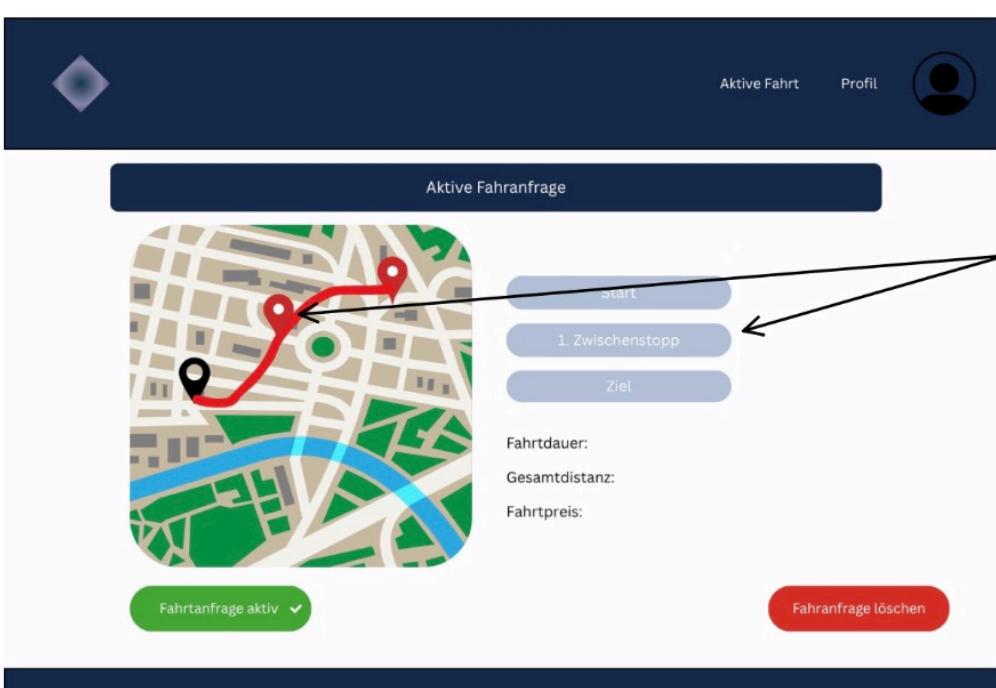
Fahrten mit mehreren Zwischenstoppes



Zusätzliche Funktion:

Button um ein Eingabefeld für einen Zwischenstopps hinzuzufügen

Pin für Zwischenstopps auf der Karte

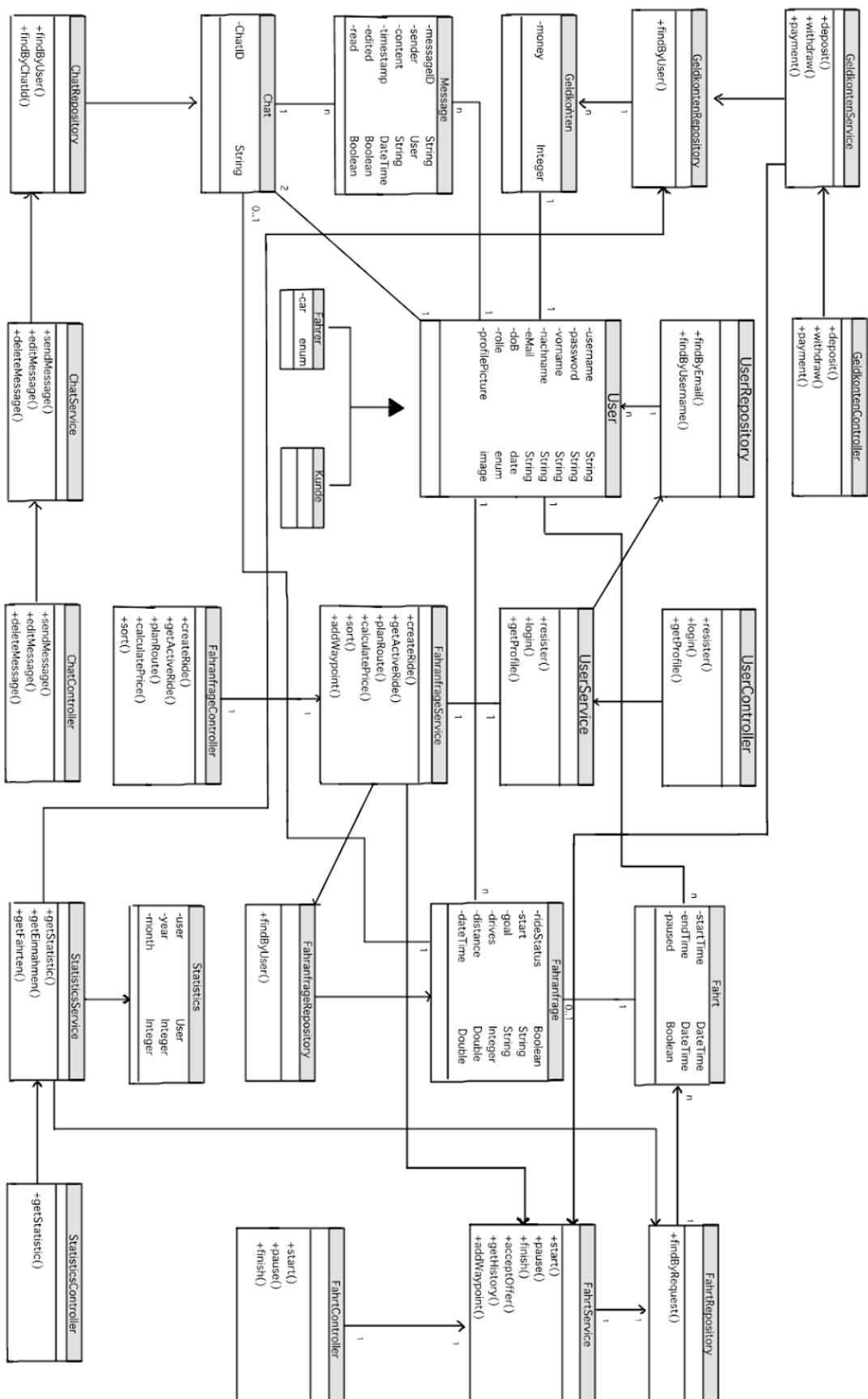


Zusätzliche Funktion:

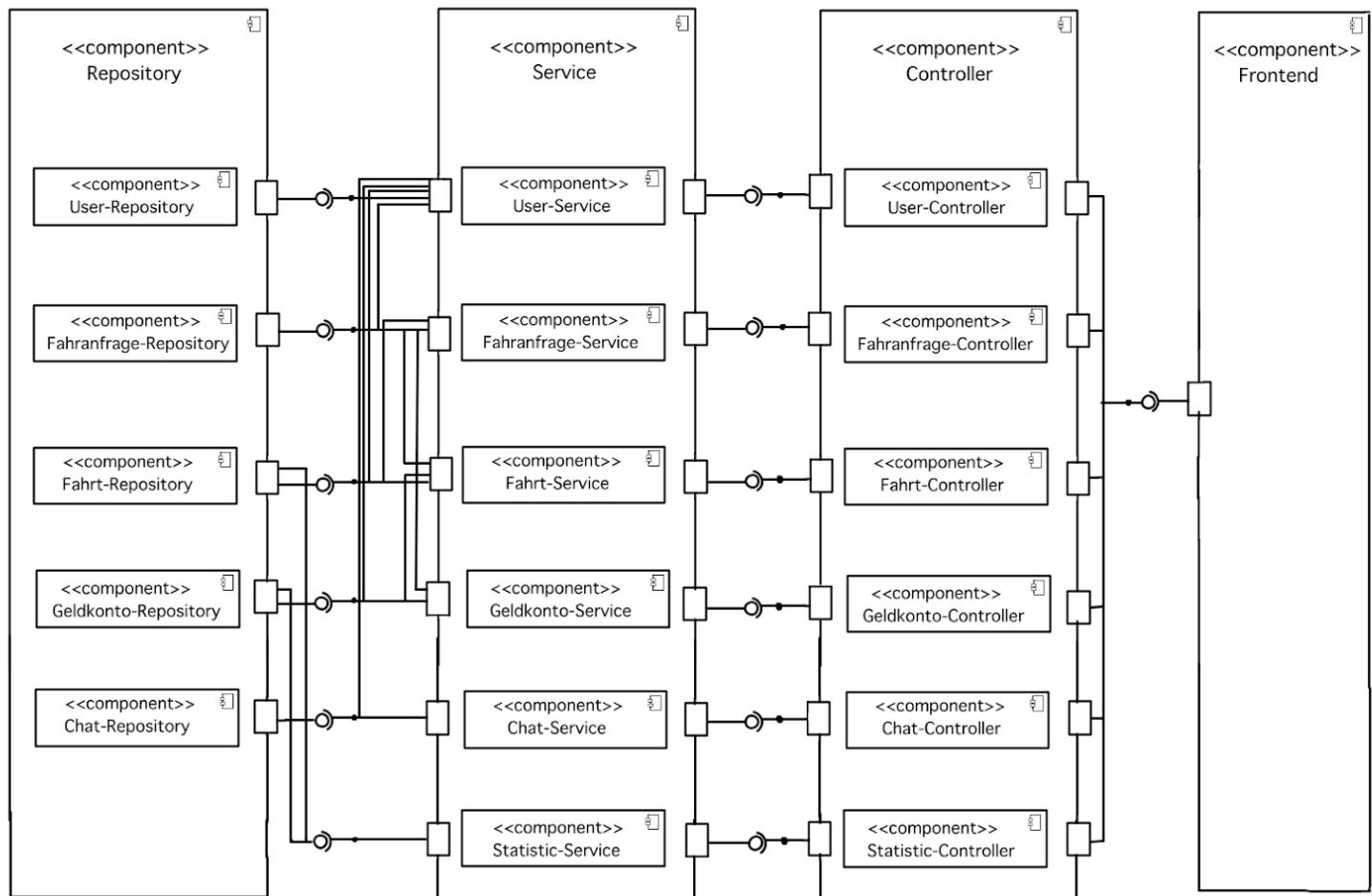
Anzeige der Zwischenstopps im Aktiven Fahrt Fenster

Strukturdiagramme (Komponenten- und Klassendiagramme)

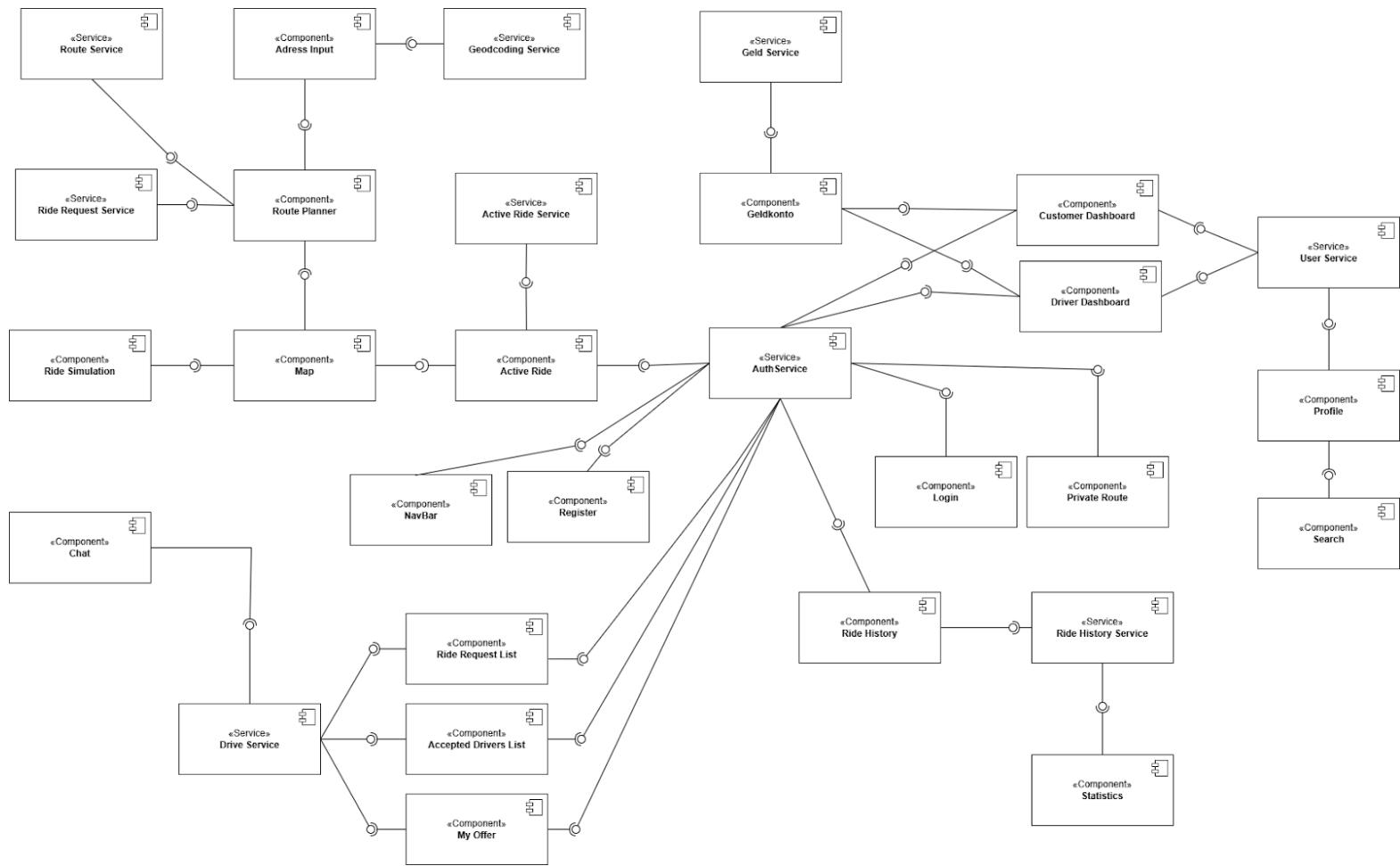
Klassendiagramm



Komponentendiagramm (Backend):

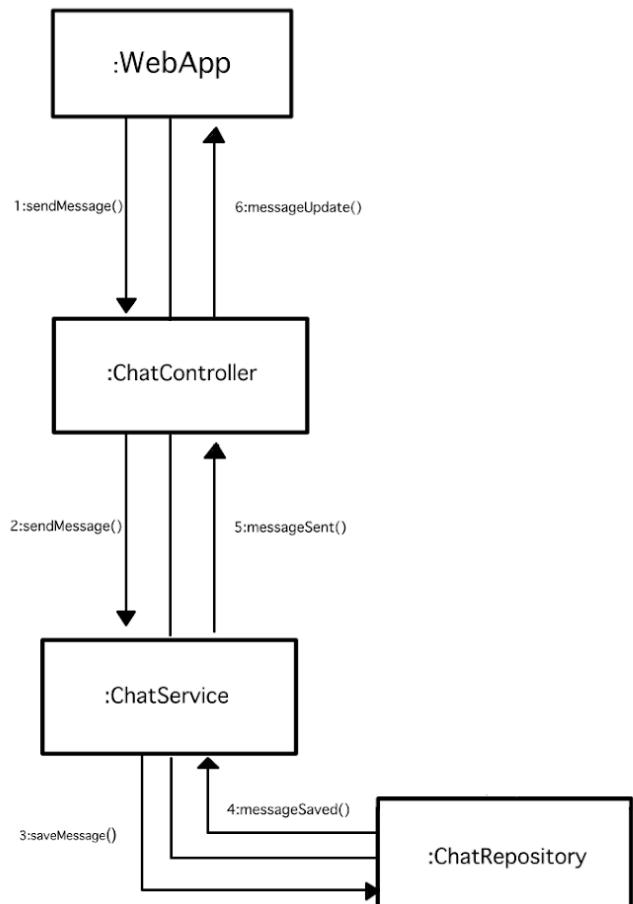


Komponentendiagramm (Frontend):

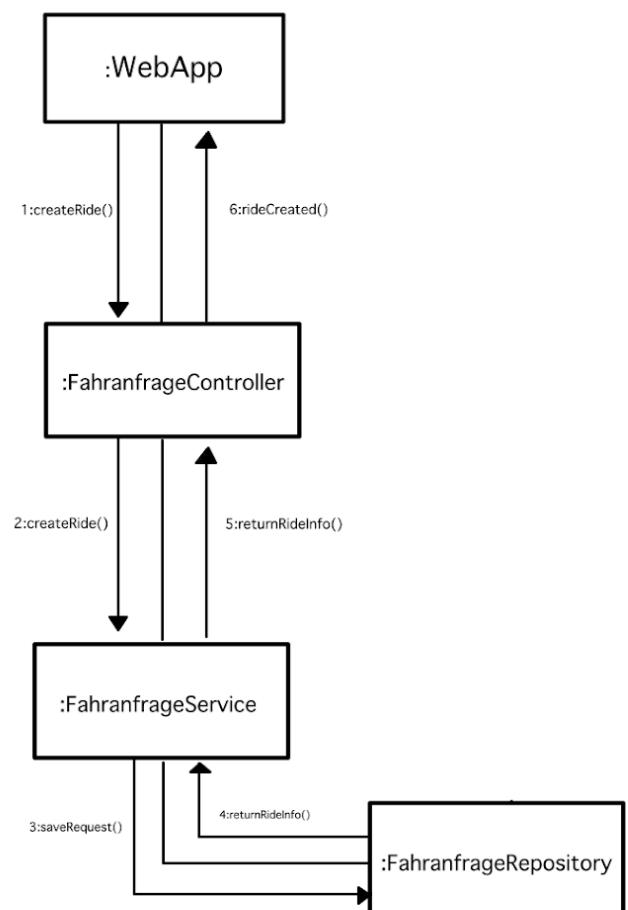


Verhaltensdiagramme (Kommunikationsdiagramme)

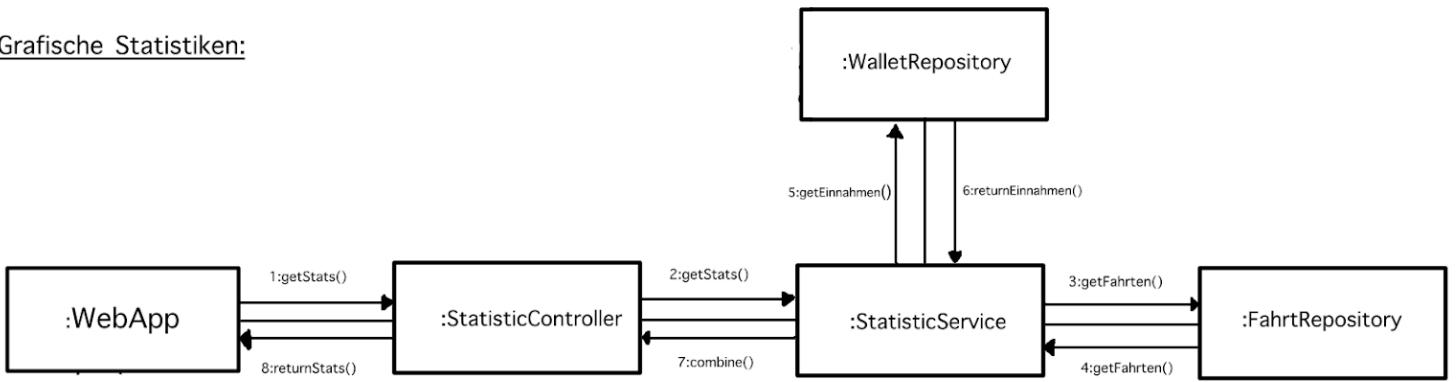
Chat:



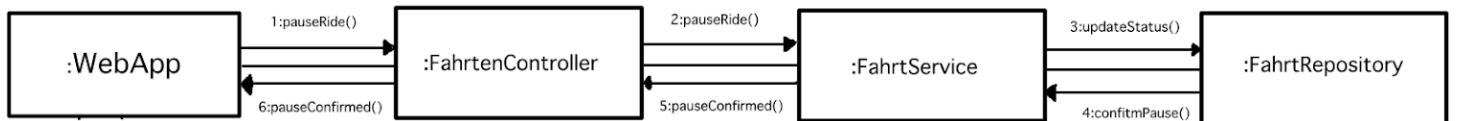
Fahranfrage mit mehreren Zwischenstopps:



Grafische Statistiken:



Live-Änderungen bei der Fahrt:



Funktionalitätsplanung

ID	Funktionalität	Verantwortliche r	Abhängige Funktionalitä ten	Verknü pfte User-St ories	Quellcode-referenz	Status
1.	Chat					
1.1	Chat im Frontend	Chantal	1.2	US-1, US-5	frontend/src/components/Chat.js frontend/src/components/ AcceptedDrive rsList.js (Zeilen 98, 128-133) frontend/src/components/ RideRequestLi st.js (Zeilen 256,298-304) frontend/src/services/ChatService.js	Fertig
1.2	Chat im Backend	Nevzat	1.1	US-1, US-5	backend/controller/ChatController.java backend/controller/ChatWebSocketController.java backend/dto/ChatMessageRequest.java backend/entity/Chat Message.java backend/repository/ ChatMessageRepository.java backend/service/Cha tService.java backend/dto/ApiRes pone.java backend/dto/ChatM essageResponse.java	Fertig
2.	Fahrten mit mehreren Zwischenstop ps					

2.1	Zwischenstopps im Frontend	Kerem	2.2	US-2	<p>frontend/src/components/RoutePlanner.js</p> <p>frontend/src/components/AddressInput.js</p> <p>frontend/src/components/ActiveRide.js (Zeilen 61-66)</p> <p>frontend/src/services/RouteService.js</p> <p>frontend/src/services/RideRequestService.js (Zeilen 43-74 Funktion: updateRequest)</p> <p>frontend/src/components/MapComponent.js</p>	Fertig
2.2	Zwischenstopps im Backend	Abdullah	2.1	US-2	<p>backend/entity/Stop.java</p> <p>backend/service/DriveService.java (Zeilen 239-315)</p>	Fertig
3.	Grafische Statistiken					
3.1	Statistiken im Frontend	Chantal	3.2	US-3	frontend/src/components/Statistics.js	Fertig
3.2	Statistiken im Backend	Abdullah	3.1	US-3	backend/service/DriveService.java (Zeilen 239-315)	Fertig
4.	Live Änderungen während der Fahrt					
4.1	Live Änderungen im Frontend	Kerem	4.2	US-4	<p>frontend/src/components/MapComponent.js</p> <p>frontend/src/components/CustomerSimulation.js</p> <p>frontend/src/components/DriverSimulation.js</p>	Fertig

					frontend/src/components/ RoutePlanner.js frontend/src/services/ DriveService.js frontend/src/websocket/WebSocketService.js	
4.2	Live Änderungen im Backend	Nevzat	4.1	US-4	backend/controller/DriveController.java backned/service/DriveService.java (Zeilen 1-238)	Fertig
5.	Generell					
5.1	AuthContext	Kerem			frontend/src/context/AuthContext.js	Fertig
5.2	Demo Route	Kerem			frontend/src/DemoRoute/duisburgEssenRoute.json	Fertig

Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status
T1	Gültige Einzahlung	WalletServiceTest.doDeposit_validAmount_updatesWalletAndCreatesTransaction()	bestanden
T2	Geldüberweisung gültig	WalletServiceTest.transfer_validTransfer_worksCorrectly()	bestanden
T3	Unzureichendes Guthaben	WalletServiceTest.transfer_InsufficientFunds_throwsException()	bestanden
T4	Wallet nicht gefunden	WalletServiceTest.doDeposit_walletNotFound_throwsException()	bestanden
T5	Fahrpreisermittlung(Kosten angegeben)	FahrpreisEinheitstests.mapDriveRequestToDrive_driveRequestCostExistsCase_test()	bestanden
T6	Fahrpreisermittlung(Kosten nicht angegeben)	FahrpreisEinheitstests.mapDriveRequestToDrive_driveRequestCostDontExistsCase_test()	bestanden

Systemtests

Datum	15.07.2025		
Tester	Simon Lobo		
SW-Version	V 3.0.0		
Vorbedingung(en)	Der Benutzer ist im System als Kunde angemeldet und hat keine aktive Fahrt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	v / X
1	Der Benutzer klickt auf "Route planen".	Das System zeigt ein Formular für Fahranfragen und eine Karte an.	v
2	Der Benutzer wählt eine beliebige Startadresse aus.	Das System zeigt die ausgewählte Adresse im Startpunkt an und die Karte setzt einen Pin auf die Position.	v
3	Der Benutzer klickt auf "Stop hinzufügen".	Das System zeigt ein weiteres Eingabefeld für einen Stopp an.	v
4	Der Benutzer gibt in dem Feld "Stop 1" eine Adresse ein.	Das System zeigt eine Leiste mit Auswahloptionen an.	v
5	Der Benutzer wählt eine Adresse aus den Auswahloptionen aus.	Das System zeigt die Adresse im jeweiligen Feld an und die Karte setzt einen Pin auf die ausgewählte Position.	v
6	Der Benutzer gibt in dem Feld "Stop 1" einen POI ein.	Das System zeigt eine Leiste mit Auswahloptionen an.	v
7	Der Benutzer wählt einen POI aus den Auswahloptionen aus.	Das System zeigt den POI im jeweiligen Feld an und die Karte setzt einen Pin auf die ausgewählte Position.	v
8	Der Benutzer klickt in dem Feld "Stop 1" die Checkbox "Koordinaten".	Das System zeigt ein Eingabefeld für Koordinaten in der jeweiligen Zeile an.	v
9	Der Benutzer gibt Koordinaten ein und klickt auf "Koordinaten" bestätigen.	Das System zeigt die Koordinaten im jeweiligen Feld an und setzt einen Pin auf die jeweilige Position.	v

10	Der Benutzer wählt ein beliebiges Ziel aus.	Das System zeigt das Ziel im Feld an und setzt einen Pin auf die ausgewählte Position.	
11	Der Nutzer klickt auf "Route berechnen".	Das System zeigt in der Karte die Route zwischen Start, dem Zwischenstopp und dem Ziel an. Außerdem werden die Strecke, der Preis und die Dauer angezeigt.	✓
12	Der Nutzer klickt auf "Fahrt anfragen".	Das System zeigt an, dass die Fahrt erfolgreich erstellt wurde.	✓
Nachbedingung(en)	Der Nutzer hat eine aktive Fahrtanfrage.		
Testurteil	Test bestanden.		

Datum	15.07.2025		
Tester	Simon Lobo		
SW-Version	V 3.0.0		
Vorbedingung(en)	Der Benutzer ist im System als Kunde angemeldet und als Fahrer angemeldet. Der Kunde hat eine aktive Fahranfrage.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Fahrer klickt in der Navigationsleiste auf "Fahranfragen".	Das System zeigt die offenen Fahranfragen, darunter auch die Fahranfrage des angemeldeten Kunden.	✓
2	Der Fahrer klickt auf "Anbieten" für die Fahranfrage des angemeldeten Kunden.	Das System erweitert die Tabelle um eine Spalte "Chat".	✓
3	Der Fahrer klickt auf den Button "Chat".	Das System öffnet die Chat-Seite zwischen dem Kunden und dem Fahrer.	✓
4	Der Fahrer gibt eine Nachricht in die Eingabezeile ein und drückt auf "Senden"	Das System zeigt die Nachricht im Chat an.	✓

5	Der Fahrer klickt auf "Bearbeiten"	Das System zeigt eine Eingabezeile zum Bearbeiten der Nachricht an.	✓
5.1	Der Fahrer klickt auf "Abbrechen"	Die Eingabezeile zum Bearbeiten verschwindet.	✓
5.2	Der Fahrer bearbeitet die Nachricht und klickt auf "Ändern"	Die Nachricht ändert sich und die Eingabezeile verschwindet.	✓
6	Der Fahrer klickt auf den Button "Löschen"	Die Nachricht verschwindet.	✓
7	Der Kunde klickt auf "Fahrerangebote"	Das System zeigt die Liste der Fahrerangebote an, wobei in der Spalte "Chat" jeweils ein Button für den Chat mit jedem Fahrer ist.	✓
8	Der Kunde klickt in der Zeile für den angemeldeten Fahrer auf "Chat"	Das System zeigt den Chat mit dem Fahrer an.	✓
9	Der Kunde sendet eine Nachricht an den Fahrer	Der Fahrer sieht die Nachricht direkt auf seiner Chat Oberfläche.	✓
10	Der Kunde klickt auf "Bearbeiten", ändert seine Nachricht und klickt auf "Ändern".	Das System informiert den Kunden, dass das Bearbeiten nicht mehr möglich ist.	✓
11	Der Kunde klickt auf "Löschen"	Das System informiert den Kunden, dass das Löschen nicht mehr möglich ist	✓
Nachbedingung(en)			
Testurteil	Test bestanden		

Datum	15.07.2025		
Tester	Simon Lobo		
SW-Version	V 3.0.0		
Vorbedingung(en)	Der Benutzer ist im System als Kunde und als Fahrer angemeldet. Der Kunde hat eine aktive Fahranfrage und der Fahrer hat diese angenommen. Der Kunde hat das Angebot des Fahrers angenommen.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Fahrer klickt auf "Fahrtansicht" in der Navigationsleiste.	Das System zeigt die Ansicht für die Fahrt-Simulation an.	✓
2	Der Fahrer klickt auf "Start"	Die Fahrt-Simulation fängt an.	✓
3	Der Fahrer klickt auf "Pause"	Die Fahrt-Simulation hält an.	✓
4	Der Kunde klickt auf "Fahrtansicht" in der Navigationsleiste	Das System zeigt die Ansicht für die Fahrt-Simulation an.	✓
5	Der Kunde klickt auf "Route bearbeiten"	Das System zeigt dem Kunden eine Übersicht an, um seine Route zu bearbeiten.	✓
6	Der Kunde klickt auf "Stop hinzufügen"	Das System zeigt dem Kunden eine Übersicht, um einen neuen Stopp hinzuzufügen.	✓
7	Der Kunde wählt eine Adresse für den neuen Stopp.	Das System zeigt den neuen Stopp als Pin auf der Karte an.	✓
8	Der Kunde wählt eine neue Adresse für das Ziel aus.	Das System zeigt das neue Ziel auf der Karte an	✓
9	Der Kunde klickt auf "Route berechnen".	Das System zeigt die neue Route auf der Karte, die neu berechnete Strecke, den Preis und die Dauer an.	✓
10	Der Kunde klickt auf "Stopp entfernen"	Das System entfernt den Pin für diesen Stopp	✓

11	Der Kunde klickt auf "Route berechnen"	Das System zeigt die neue Route auf der Karte, die neu berechnete Strecke, den Preis und die Dauer an.	✓
12	Der Kunde klickt auf "Fahrt aktualisieren"	Das System benachrichtigt den Kunden, dass die Fahrt aktualisiert wurde	✓
13	Der Kunde klickt auf "Zurück zur Fahrtansicht"	Das System zeigt die Fahrtsimualtion an.	✓
14	Der Fahrer klickt auf "Fortsetzen"	Das System setzt die Fahrt mit der neuen Route fort.	✓
15	Der Fahrer klickt auf "Fahrt abschließen"	Das System informiert den Fahrer, dass die Fahrt abgeschlossen ist.	✓
Nachbedingung(en)	Der Kunde und der Fahrer haben eine abgeschlossene Fahrt		✓
Testurteil	Test bestanden		

Datum	15.07.2025		
Tester	Simon Lobo		
SW-Version	V 3.0.0		
Vorbedingung(en)	Der Benutzer ist als Fahrer im System angemeldet und hat mindestens eine abgeschlossene Fahrt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Fahrer klickt in der Navigationsleiste auf "Profil"	Das System zeigt das Profil des Fahrers an.	✓
2	Der Fahrer klickt auf "Statistiken"	Das System zeigt eine Seite für Statistiken an.	✓

3	Der Fahrer wählt eine Art der Statistiken.	Das System zeigt in der Leiste die Art der Statistik an.	✓
4	Der Fahrer wählt unter Ansicht "monatlich" aus	Das System zeigt eine weitere Auswahloption an, um das Jahr auszuwählen.	✓
5	Der Fahrer wählt ein Jahr aus.	Das System zeigt ein Balkendiagramm mit der ausgewählten Art und den Monaten an.	✓
6	Der Benutzer zeigt mit der Maus auf den Balken eines Monats.	Das System zeigt den Wert des gewählten Monats an.	✓
7	Der Benutzer wählt unter "Ansicht" nun "täglich" aus	Das System zeigt eine weitere Auswahloption an um den Monat auszuwählen	✓
8	Der Benutzer wählt einen Monat aus.	Das System zeigt ein Balkendiagramm mit der ausgewählten Art und den Tagen des Monats an.	✓
9	Der Benutzer zeigt mit der Maus auf den Balken eines Tages.	Das System zeigt den Wert des gewählten Tags an.	✓
Nachbe-dingung(en)			
Testurteil	Test bestanden		

Nutzerhandbuch

Technische Anforderungen

Die Anwendung wurde auf folgenden Systemen getestet:

Windows 11 mit Google Chrome Browser und 16GB Arbeitsspeicher. Virtualisierung muss aktiv sein.

MacOS 15.4.1 mit Safari, M1 und 8GB Arbeitsspeicher.

Installationsanleitung

Einrichtung der SEP-Webanwendung

1. Docker Desktop installieren, einrichten und starten

Zum Ausführen der Anwendung wird Docker Desktop benötigt. Docker Desktop kann unter <https://www.docker.com/products/docker-desktop> heruntergeladen werden.

2. Projektdateien herunterladen und entpacken.

3. Zum Projektverzeichnis navigieren

Im geöffneten Terminal ist in den Ordner zu wechseln. Dies geschieht mit dem Befehl cd, gefolgt vom Pfad zum Ordner. Zum Beispiel:

```
cd C:\Users\Name\Downloads\Abgabe-SEP\tar-dateien\
```

Terminal oder Eingabeaufforderung öffnen und das Verzeichnis mit den .tar Dateien auswählen. Anschließend folgende Befehle eingeben: “docker load -i sepdrive-backend.tar” und “docker load -i sepdrive-frontend.tar”.

```
docker load -i sepdrive-backend.tar
```

```
docker load -i sepdrive-frontend.tar
```

Nachdem diese Prozesse fertig sind, werden folgende Befehle eingeben: “docker run -d -p 3000:3000 sepdrive-frontend” und “docker run -d -p 8080:8080 sepdrive-backend”.

docker run -d -p 3000:3000 sepdrive-frontend

docker run -d -p 8080:8080 sepdrive-backend

4. Webanwendung aufrufen

Nach erfolgreichem Start kann die Webanwendung im Browser unter folgenden Adressen abgerufen werden.

Frontend: <http://localhost:3000>

Backend(Kein-UI): <http://localhost:8080>

H2-Console: <http://localhost:8080/h2-console/>

aufgerufen werden.

Bedienungsanleitung

1. Registrierung und Login

Registrierungsseite aufrufen

Das Registrierungsformular kann über die Navigationsleiste (Menüpunkt „Registrieren“) oder über den „Registrieren“-Button auf der Login-Seite geöffnet werden.

Angaben zur Registrierung

Für die Registrierung sind folgende Informationen erforderlich:

Nutzername

Nachname

Vorname

E-Mail-Adresse

Geburtsdatum

Rolle (Fahrer oder Kunde)

Hinweis: Für Namen und Passwort bestehen keine spezifischen Anforderungen.

Profilbild (optional)

Ein Profilbild kann optional hinzugefügt werden.

Registrierung abschließen

Nach dem Ausfüllen aller Pflichtfelder kann der Button „Registrieren“ betätigt werden.

Login

Nach erfolgreicher Registrierung erfolgt die Anmeldung über die Eingabe von Nutzernamen und Passwort.

Nach dem Login wird automatisch auf das Dashboard weitergeleitet.

Hinweis:

Für die **gleichzeitige Nutzung** der Anwendung mit mehreren Nutzer oder Rollen (z. B. Fahrer und Kunde) gilt:

Pro Browser-Fenster ist nur eine Sitzung zulässig.

Um z. B. Fahrer und Kunde gleichzeitig zu testen, muss:

eine Rolle im **Inkognito-/Privatmodus** ausgeführt werden,

oder ein anderer Browser (z. B. Edge, Brave oder ein anderer idealerweise Chromium-basierter Browser) verwendet werden.

Nur so ist eine parallele und korrekte Nutzung beider Rollen möglich.

2. Routenplanung

Hinweis zum Routing- und Geocoding-Dienst:

Die OpenRouteService-Server können zeitweise nicht erreichbar sein.

Der aktuelle Verfügbarkeitsstatus lässt sich in der Statusleiste am unteren Seitenrand einsehen und bei Bedarf erneut prüfen. Dort befindet sich auch ein Link zur offiziellen ORS-Statusseite mit weiteren Informationen. (<https://status.openrouteservice.org/>)

Falls der Dienst offline ist, kann alternativ die **lokal gespeicherte Demo-Route** für Test- oder Demonstrationszwecke verwendet werden.

Routenplaner öffnen

Der Routenplaner kann über die Navigationsleiste unter “Route Planen” geöffnet werden.

Eingabe der Route

Die Route kann auf folgende Weise geplant werden:

Adresse oder POI (Point of Interest)

Ein Ort (z. B. Adresse oder POI) wird in das Eingabefeld eingegeben.

Anschließend kann ein Dropdown-Menü geöffnet werden, aus dem der gewünschte Ort ausgewählt werden kann.

Koordinaten (optional)

Wird das Kontrollkästchen „Koordinaten auswählen“ aktiviert, können Breiten- und Längengrad eingegeben werden.

Die Eingaben müssen anschließend über die Bestätigungs-Schaltfläche übernommen werden.

Hinweis: Für Koordinaten, die in nicht erreichbaren Gebieten liegen (z. B. Gewässer oder Orte ohne Straßen im Umkreis von 350 m), können keine Routen geplant werden und müssen angepasst werden.

Aktueller Standort (optional)

Der aktuelle Standort kann – sofern Standortfreigabe aktiviert ist – als Startpunkt genutzt werden.

Zwischenstopps (optional)

Bis zu drei Zwischenstopps können zur Route hinzugefügt oder entfernt werden.

Hinweis: Nach jedem Bearbeiten der Route die Route möglichst neu berechnen, um die Daten zu aktualisieren.

Kartenservice auswählen

Im Routenplaner kann zwischen zwei Kartenservices gewählt werden.

Dies ermöglicht die Nutzung eines alternativen Dienstes, falls ein Anbieter technische Probleme hat oder der andere Anbieter bevorzugt wird.

Route berechnen

Nach vollständiger Eingabe der Daten kann die Route über den Button „Route planen“ berechnet werden. Dabei werden auch die Distanz, der Preis und die Dauer berechnet.

Routenanzeige

Die berechnete Strecke wird auf der Karte angezeigt – inklusive Pins und Linienführung.

Die Pins sind klickbar und geben Auskunft darüber, ob es sich um Start, Zwischenstopp oder Ziel handelt.

Route anfragen

Entspricht die berechnete Route den gewünschten Angaben, kann sie über den entsprechenden Button als Fahrtanfrage übermittelt werden.

Hinweis: Der Kunde muss ausreichend Geld auf dem Konto haben, um die Fahrt anfragen zu können.

Anfrage einsehen oder Stornieren

Die Seite der Aktive Anfrage kann über die Navigationsleiste unter „Aktive Fahrt“ aufgerufen werden. Dort kann die Anfrage auch storniert werden.

Demo-Route

Für Test- oder Demonstrationszwecke steht eine vordefinierte Demo-Route zur Verfügung, die ohne eigene Eingaben angezeigt werden kann.

Hinweis: Die Demo-Route ist offline und eventuell nicht aktuell.

3. Profile

Das Profil kann durch die Navigationsleiste unter “Profil” aufgerufen werden.

Für die Suche eines Nutzers, kann in der Navigationsleiste “Suche” ausgewählt werden. Anschließend kann man im Suchfeld den gewünschten Username eingeben und suchen.

4. Fahrthistorie

Über die Navigationsleiste kann ein Nutzer sein Profil unter “Profil” aufrufen. Dort kann er über den Button “Fahrthistorie” zu der Tabelle mit seinen abgeschlossenen Fahrten gelangen.

Um eine beliebige Spalte zu sortieren, muss der Nutzer auf das Sortiersymbol der jeweiligen Spalte klicken. Durch das Suchfeld kann der Nutzer die Tabelle nach Namen filtern.

5. Geldkonten

Der Nutzer kann sein Geldkonto in seinem Dashboard finden. Nutzer können dort ihren Kontostand einsehen.

Kunden können außerdem einen Betrag eingeben, den sie hochladen möchten und ihn dann durch den Button “hochladen” auf ihr Konto übertragen.

Vor dem Erstellen einer Fahranfrage sollte der Kunde über ausreichend Geld auf seinem Konto verfügen, da die Anfrage sonst nicht abgesendet werden kann.

6. Fahranfragen einsehen (für Fahrer)

Fahrer können aktuelle Fahranfragen von Kunden in einer Tabelle einsehen. Die Seite wird über den Button „Fahranfragen“ in der Navigationsleiste geöffnet.

Die Einträge lassen sich nach verschiedenen Kriterien wie Preis oder Entfernung sortieren.

Durch Eingabe des eigenen aktuellen oder gewünschten Standortes kann die Entfernung zum Startpunkt der jeweiligen Fahrt berechnet und angezeigt werden. Zusätzlich wird die Entfernung auch in der Luftlinie angezeigt.

Über den Button in der jeweiligen Zeile kann ein Fahrtangebot an den Kunden gesendet werden. Durch einen Klick auf den Benutzernamen eines Kunden kann das jeweilige Kundenprofil geöffnet und eingesehen werden.

Abgegebene Angebote können vom Fahrer im "Dein Fahrangebot" Bereich über der Tabelle mit dem "Zurückziehen" Button wieder zurückgezogen werden.

Wichtiger Hinweis:

Wurde eine Anfrage vom Kunden bereits abgelehnt oder das Angebot zuvor vom Fahrer zurückgezogen, kann kein neues Angebot für diese Fahrt erstellt werden. In diesem Fall sollte eine andere Anfrage ausgewählt oder auf neue Fahrten gewartet werden.

7. Fahrer Angebote einsehen (für Kunden)

Kunden können die erhaltenen Fahrtangebote der Fahrer über die Navigationsleiste aufrufen.

Die Tabelle zeigt alle eingegangenen Angebote und bietet Sortierungsmöglichkeiten, z. B. nach Bewertung des Fahrers oder Preis.

Durch einen Klick auf den Benutzernamen eines Fahrers kann das jeweilige Fahrerprofil geöffnet und eingesehen werden.

Für jedes Angebot stehen zwei Optionen zur Verfügung:

Ablehnen: Das Angebot wird verworfen.

Annehmen: Die Fahrt wird bestätigt und kann anschließend vom Fahrer begonnen werden.

8. Durchführung einer Fahrt

Sobald der Kunde ein Fahrangebot angenommen hat, wird dies in dem Bereich „Dein Fahrangebot“ beim Fahrer als ACCEPTED markiert. Der Fahrer kann anschließend über die Navigationsleiste die Fahrtansicht öffnen und die Simulation mit dem Button „Starten“ beginnen. Während der Simulation sollte die Fahreransicht **geöffnet bleiben** und möglichst **im Vordergrund angezeigt** werden.

Nach dem Start der Fahrt kann der aktuelle Standort bzw. der Fortschritt entlang der Route in Echtzeit auf der Karte verfolgt werden. Diese Ansicht steht auch dem Kunden zur Verfügung und kann ebenfalls über den Button „Fahrtansicht“ in der Navigationsleiste geöffnet werden.

Während der Fahrt gibt es folgende Steuerungsmöglichkeiten:

Pause: Die Fahrt kann jederzeit mit dem Button „Pause“ unterbrochen werden.

Fortsetzen: Mit dem Button „Fortsetzen“ kann die Simulation fortgeführt werden.

Geschwindigkeit anpassen: Über einen Schieberegler lässt sich die Geschwindigkeit der Simulation flexibel einstellen.

Die aktuelle Geschwindigkeit sowie die Position des Reglers werden in Echtzeit zwischen Fahrer und Kunde synchronisiert, sodass beide stets denselben Stand sehen.

Am Ende der Route kann der Fahrer die Fahrt über den Button „Beenden“ abschließen.

9. Live-Änderung während einer Fahrt

Wenn die Fahrt pausiert ist, kann der Kunde über den Button „Route bearbeiten“ den Routenplaner öffnen und Änderungen an der aktuellen Strecke vornehmen. Dabei gilt:

Es können bis zu drei Zwischenstopps hinzugefügt oder entfernt werden.

Das Ziel kann geändert werden.

Der Startpunkt ist fest und entspricht automatisch dem aktuellen Standort zum Zeitpunkt der Pause, er kann also nicht verändert werden.

Nach Auswahl der gewünschten Adressen oder Koordinaten kann die Route neu berechnet werden. Sobald die angezeigte Route den Erwartungen entspricht, kann sie über den Button „Fahrt aktualisieren“ übernommen und die bisherige Route überschrieben werden.

Mit dem Button „Zurück zur Fahrtansicht“ kann zur Fahrtsimulation zurückgekehrt und die Fahrt fortgesetzt werden.

Hinweis zur Demo-Route:

Die Demo-Route steht optional für Testzwecke zur Verfügung, ist jedoch offline und enthält keine aktuellen Geodaten. Auch nach einer Fahrtaktualisierung berücksichtigt die Demo-Route nicht den aktuellen Fahrfortschritt oder das Entfernen von Zwischenstopps.

Wenn eine Internetverbindung besteht und die OpenRouteService-Server erreichbar sind, sollte die Route möglichst online und in Echtzeit neu berechnet werden. Die Demo-Route ist primär für Tests und Vorführungen ohne Internetzugang oder bei Serverproblemen gedacht.

10. Chat

Wenn ein Fahrer eine Fahranfrage eines Kunden annimmt, erscheint neben der ausgewählten Fahrt in der Fahranfragen-Tabelle ein Button, der zum Chat mit dem Kunden führt. Kunden können über die Tabelle der Fahrerangebote einen ausgewählten Chat aufrufen.

Um eine Nachricht zu senden, gibt der Benutzer den Text in das Eingabefeld ein und klickt auf „Senden“.

Zum Löschen einer Nachricht klickt der Benutzer unter der jeweiligen Nachricht auf „Löschen“.

Zum Bearbeiten klickt der Benutzer unter der Nachricht auf „Bearbeiten“. Daraufhin öffnet sich ein Eingabefeld, in dem der Text angepasst werden kann. Mit einem Klick auf „Abbrechen“ wird das Eingabefeld geschlossen und die Nachricht bleibt unverändert. Wird hingegen auf „Ändern“ geklickt, wird die bearbeitete Nachricht übernommen.

11. Statistiken

Fahrer können über ihr Profil mit dem Button „Statistiken“ zur Übersicht zum Erstellen von Statistiken gelangen. Dort wählen sie zunächst die Art der Statistik sowie die gewünschte Ansicht aus.

Wird als Ansicht „monatlich“ gewählt, erscheint die Auswahl für das Jahr. Bei Auswahl von „täglich“ wird zusätzlich zur Jahresauswahl auch eine Monatsauswahl angezeigt.

Sobald alle Optionen ausgewählt wurden, wird die grafische Statistik dargestellt.

Fährt man mit der Maus über einen bestimmten Tag oder Monat, wird der genaue Wert für den jeweiligen Zeitraum angezeigt.