

SEP Projektmappe

PROJEKTMAPPE DES PROJEKTES

SEP-Drive

Dokumentation des Projektes

Gruppe P:

Moyo Fred Aymerick

Mohammed Daoudi

Reine Christiane Demgne

Thomas Fabio Kaiser

Amin Rais

Ian Christopher Schirmer

Mats Paul Tončik

Inhalt

Projektbeschreibung	4
Zyklus I	5
Spezifikationsplanung	5
User-Stories	6
Papierprototypen	7
Strukturdiagramme (Komponenten- und Klassendiagramme)	7
Verhaltensdiagramme (Kommunikationsdiagramme)	7
Funktionalitätsplanung	8
Systemtests	9
Zyklus II	11
Spezifikationsplanung	11
User-Stories	11
Papierprototypen	12
Strukturdiagramme (Komponenten- und Klassendiagramme)	12
Verhaltensdiagramme (Kommunikationsdiagramme)	12
Funktionalitätsplanung	13
Modultests	14
Systemtests	14
Zyklus III	15
Spezifikationsplanung	15
User-Stories	15
Papierprototypen	16
Strukturdiagramme (Komponenten- und Klassendiagramme)	16
Verhaltensdiagramme (Kommunikationsdiagramme)	16
Funktionalitätsplanung	17
Modultests	18
Systemtests	18
Nutzerhandbuch	19
Technische Anforderungen	19
Installationsanleitung	19
Bedienungsanleitung	19

Projektbeschreibung

SEP-Drive ist eine Mitfahr-App, bei der Kunden Fahrtanfragen erstellen und Fahrtangebote von Fahrern erhalten. SEP-Drive übernimmt die Routenplanung und bietet eine kartenbasierte Visualisierung zur Einrichtung von Fahrten, zur Anzeige der geplanten Route und zur Verfolgung der Fahrt, während sie sich vom Startpunkt zum Ziel bewegt.

Zyklus I

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Registrierung eines Benutzers			
1.1	Kunde registrieren	User Story	Mats Paul Tončik	Fertig
1.2	Fahrer registrieren	User Story	Mats Paul Tončik	Fertig
1.3	Registrierung mit Profilbild	User Story	Mats Paul Tončik	Fertig
1.4	E-Mail verifizieren	User Story	Mats Paul Tončik	Fertig
1.5	Verifizierungs-Link erneut senden	User Story	Mats Paul Tončik	Fertig
1.6	Registrierungs-Fenster	Papierprototyp	Mats Paul Tončik	Fertig
1.7	E-Mail-Verifizierungs-Fenster	Papierprototyp	Mats Paul Tončik	Fertig
2.	Login eines Benutzers			
2.1	Kunde/Fahrer anmelden	User Story	Mats Paul Tončik	Fertig
2.2	Sicherheitscode erhalten	User Story	Mats Paul Tončik	Fertig
2.3	Sicherheitscode erneut senden	User Story	Mats Paul Tončik	Fertig
2.4	Login-Fenster	Papierprototyp	Mats Paul Tončik	Fertig
2.5	OTP-Verifizierungs-Fenster	Papierprototyp	Mats Paul Tončik	Fertig
3.	Benutzerprofil (Kunde & Fahrer)			
3.1	Kundenprofil anzeigen	User Story	Moyo Fred Aymerick	Fertig
3.2	Benutzerprofilseite (Kunde)	Papierprototyp	Amin Rais	Fertig
3.3	Benutzerprofilseite (Fahrer)	Papierprototyp	Amin Rais	Fertig

3.4	Fahrerprofil anzeigen	User Story	Moyo Fred Aymerick	Fertig
3.5	Profil anderer Benutzer anzeigen	User Story	Moyo Fred Aymerick	Fertig
3.6	Suchfunktion für Benutzernamen erstellen	User Story	Moyo Fred Aymerick	Fertig
3.7	Klickbares Profil	User Story	Amin Rais	Fertig
3.8	Kundeninformation anzeigen	User Story	Amin Rais	Fertig
3.9	Fahrerinformationen anzeigen	User Story	Amin Rais	Fertig
3.10	Benutzernamen anzeigen	User Story	Amin Rais	Fertig
4.	Fahranfrage erstellen & verwalten			
4.1	Fahranfrage erstellen	User Story	Thomas Fabio Kaiser	Fertig
4.1.1	Startpunkt auswählen	User Story	Thomas Fabio Kaiser	Fertig
4.1.2	Zielpunkt auswählen	User Story	Thomas Fabio Kaiser	Fertig
4.1.3	Autoklasse auswählen	User Story	Thomas Fabio Kaiser	Fertig
4.2	Fahranfrage-Formular erstellen	Papierprototyp	Reine Christiane Demgne	Fertig
4.3	Aktive Fahranfrage anzeigen/löschen	User Story	Thomas Fabio Kaiser	Fertig
4.4	Anzeige aktive Fahranfragen	Papierprototyp	Reine Christiane Demgne	Fertig
4.5	Fehlermeldung / Hinweisdialog	Papierprototyp	Reine Christiane Demgne	Fertig
4.6	Geolocation und Karteneinbindung Anzeige	Papierprototyp	Reine Christiane Demgne	Fertig
5.	Kartenvisualisierung der Route			
5.1	Kartenvisualisierung der Route	User Story	Ian Christopher Schirmer	Fertig
5.2	Kartenansicht der Route	Papierprototyp	Mohammed Daoudi	Fertig
5.3	Darstellung von Pins für Start/Ziel/Stops	User Story	Ian Christopher Schirmer	Fertig
5.4.1	Interaktive Karte (Verschieben)	User Story	Ian Christopher Schirmer	Fertig
5.4.2	Interaktive Karte (Zoom)	User Story	Ian Christopher Schirmer	Fertig
5.5	Darstellung von Strecke, Start-, Ziel- & Zwischenstopp-Markern	Papierprototyp	Mohammed Daoudi	Fertig

5.6	Zoom- & Verschiebefunktion der Karte	Papierprototyp	Mohammed Daoudi	Fertig
6.	Nutzersuche			
6.1	Nutzersuche	User Story	Moyo Fred Aymerick	Fertig
6.2	Suchseite für Nutzer	Papierprototyp	Amin Rais	Fertig
6.3	Suchfeld für Benutzersuche	User Story	Amin Rais	Fertig
6.4	fremde Benutzerinformationen sehen	User Story	Amin Rais	Fertig
6.5	Liste an Nutzern	User Story	Amin Rais	Fertig
7.	Architektur			
7.1	Komponentendiagramm Backend	Komponentendiagramm	Mohammed Daoudi	Fertig
7.2	Klassendiagramm Backend	Klassendiagramm	Moyo Fred Aymerick	Fertig
7.3	Kommunikationsdiagramm	Kommunikationsdiagramm	Thomas Fabio Kaiser	Fertig
7.4	Komponentendiagramm Frontend	Komponentendiagramm	Mohammed Daoudi	Fertig

User-Stories

User Story-ID	1.1
User Story-Beschreibung	Als Kunde möchte ich mich registrieren können, sodass ich Fahranfragen stellen kann.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	1.2
User Story-Beschreibung	Als Fahrer möchte ich mich registrieren können, sodass ich Fahranfragen annehmen kann.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	1.3
User Story-Beschreibung	Als Benutzer möchte ich bei der Registrierung ein Profilbild hochladen können, sodass dieses später in meinem Profil angezeigt wird.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Niedrig
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	1.1, 1.2

User Story-ID	1.4
User Story-Beschreibung	Als Benutzer möchte ich meine Registrierung durch den Verifizierungs-Link abschließen können, um den Dienst zu nutzen.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	1.1, 1.2

User Story-ID	1.5
User Story-Beschreibung	Als Benutzer möchte ich einen neuen Verifizierungslink erhalten können, wenn ich diesen nicht erhalten habe, um die Registrierung abzuschließen.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Mittel
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	1.1, 1.2

User Story-ID	2.1
User Story-Beschreibung	Als Benutzer möchte ich mich mit meinen Zugangsdaten anmelden können, um den Dienst zu nutzen.
Geschätzter Realisierungsaufwand	4 Tage
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.2
User Story-Beschreibung	Als Benutzer möchte ich den per E-Mail erhaltenen Sicherheitscode nutzen können, um die Anmeldung abzuschließen.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	2.1

User Story-ID	2.3
User Story-Beschreibung	Als Benutzer möchte ich einen neuen Sicherheitscode erhalten können, falls ich diesen nicht erhalten habe, um die Anmeldung abzuschließen.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Niedrig
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	2.2

User Story-ID	3.7
User Story-Beschreibung	Als Benutzer möchte ich auf meinen Benutzernamen klicken können damit mir meine Benutzerinformationen angezeigt werden
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	3.10, 3.8, 3.9

User Story-ID	3.8
User Story-Beschreibung	Als Benutzer möchte ich meine Benutzerinformationen sehen können
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	3.10, 3.8

User Story-ID	3.9
User Story-Beschreibung	Als Fahrer möchte ich meine Benutzerinformationen sehen können
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	3.10, 3.8

User Story-ID	3.10
User Story-Beschreibung	Als User möchte dass mein Benutzername angezeigt wird
Geschätzter Realisierungsaufwand	1 Stunde
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	3.7

User Story-ID	4.1
User Story-Beschreibung	Als Kunde möchte ich eine Fahranfrage erstellen, um einen Fahrer zu finden, der mich dahin fährt.
Geschätzter Realisierungsaufwand	3 Tage -> 4.1.1, 4.1.2 eingeschlossen
Priorität	Hoch
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	4.1.1, 4.1.2, 4.1.3

User Story-ID	4.1.1
User Story-Beschreibung	Als Kunde möchte ich meinen Startpunkt eintragen können, in Form von aktueller Position, Points of Interest, bestimmte Adresse oder genaue Koordinaten, um meinen Startort mit dem Fahrer zu teilen.
Geschätzter Realisierungsaufwand	1-2 Tage
Priorität	mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	4.1, 4.1.2, 4.1.3

User Story-ID	4.1.2
User Story-Beschreibung	Als Kunde möchte ich mein Ziel eingeben können, in Form von Koordinaten, Points of Interest oder einer bestimmten Adresse, um meine Zielort mit dem Fahrer zu teilen
Geschätzter Realisierungsaufwand	1-2 Tage
Priorität	mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	4.1, 4.1.1, 4.1.3

User Story-ID	4.1.3
User Story-Beschreibung	Als Kunde möchte ich entscheiden können, mit welcher Autoklasse ich fahren will, sodass ich ein angemessenes Auto für meine Bedürfnisse bekomme.
Geschätzter Realisierungsaufwand	3 Stunden
Priorität	niedrig
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	4.1, 4.1.1, 4.1.2

User Story-ID	4.3
User Story-Beschreibung	Als Kunde möchte ich meine Fahranfrage ansehen und zurückziehen, um meine Entscheidung zurückziehen zu können und den Status anzusehen.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	4.1

User Story-ID	5.1
User Story-Beschreibung	Als Kunde möchte ich die Route angezeigt bekommen, um zu wissen, wie diese geführt ist.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	hoch
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	5.3, 5.4.1, 5.4.2

User Story-ID	5.3
User Story-Beschreibung	Als Kunde möchte ich die Start- und Zielposition sowie Zwischenstoppes sehen können, um zu wissen, ob diese richtig sind.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	hoch
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	5.1, 5.4.1, 5.4.2

User Story-ID	5.4.1
User Story-Beschreibung	Als Kunde möchte ich die Karte verschieben können, um andere Bereiche sehen zu können.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	5.4.2

User Story-ID	5.4.2
User Story-Beschreibung	Als Kunde möchte ich den Zoom der Karte einstellen können.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	5.4.1

User Story-ID	6.3
User Story-Beschreibung	Als Nutzer möchte ich mit dem Benutzernamen nach anderen Nutzern suchen .
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	3.8, 3.9

User Story-ID	6.4
User Story-Beschreibung	Als Nutzer möchte ich auf den Benutzernamen von anderen klicken, um deren Benutzerinformationen zu sehen .
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	6.3, 3.8, 3.9, 3.10

User Story-ID	3.1
User Story-Beschreibung	Als Fahrer möchte ich das Profil eines Kunden aufrufen können, damit ich sowohl persönliche Informationen als auch bisherige Fahrerbewertungen und Fahraktivitäten sehen kann, um fundierte Entscheidungen über eine Fahrtzusage zu treffen.
Geschätzter Realisierungsaufwand	4 Stunden
Priorität	mittel
Autor	Moyo Fred Aymerick
Abhängigkeiten zu anderen User Stories	1.1 , 1.2 , 1.3 , 1.4 , 1.5 , 2.2 , 2.3

User Story-ID	3.5
User Story-Beschreibung	Als Kunde möchte ich das Profil eines Fahrers anzeigen können, sodass ich dessen Fahrzeugtyp, bisherige Fahrten und Bewertungen einsehen kann, um eine fundierte Entscheidung für die Annahme eines Fahrtangebots zu treffen.
Geschätzter Realisierungsaufwand	4 Stunden
Priorität	mittel
Autor	Moyo Fred Aymerick
Abhängigkeiten zu anderen User Stories	1.1 , 1.2 , 1.3 , 1.4 , 1.5 , 2.2 , 2.3

User Story-ID	3.4
User Story-Beschreibung	Als Benutzer möchte ich das Profil anderer Benutzer anzeigen können, sodass ich deren Rolle, Bewertungen und Fahraktivität einsehen kann, um fundierte Entscheidungen bei Fahrtanfragen oder -angeboten zu treffen.
Geschätzter Realisierungsaufwand	4 Stunden
Priorität	mittel
Autor	Moyo Fred Aymerick
Abhängigkeiten zu anderen User Stories	1.1 , 1.2 , 1.3 , 1.4 , 1.5 , 2.2 , 2.3

Papierprototypen

Papierprototyp 1.6

Hast du bereits einen Account? [Jetzt anmelden](#)

Kunde	✓	Fahrer
Vorname*	Nachname*	
Mats Paul	Tončik	
Benutzername*	Geburtsdatum*	
mats	6.10.2004	
E-Mail*	E-Mail bestätigen*	
mats@toncik.com	mats@toncik.com	
Passwort*	Passwort bestätigen*	
.....	foo	

Die Passwörter stimmen nicht überein!



Beim Klicken auf "Jetzt anmelden" wird der User zum Anmeldungs-Komponenten weitergeleitet. Beim Klicken auf "Profilbild auswählen" hat der User die Option ein Profilbild hochzuladen. Wenn man auf das Kalender-Icon klickt, wird ein Datepicker geöffnet. Wenn man auf das durchgestrichene Auge klickt, wird das Passwort angezeigt. Beim Klicken auf "Registrieren" wird die Registrierung verarbeitet.

Papierprototyp 2.4

Hast du keinen Account? [Jetzt registrieren](#)

E-Mail*	mats@toncik.com
Passwort* 
Anmelden	

Wenn man auf das durchgestrichene Auge klickt, wird das Passwort angezeigt. Beim Klicken auf "Jetzt registrieren" wird man zum Registrierungs-Komponenten geleitet. Beim Klicken auf "Anmelden" wird die Anmeldung verarbeitet.

Papierprototyp 2.5

OTP eingeben zum verifizieren: mats@toncik.com

<input type="text"/>					
----------------------	----------------------	----------------------	----------------------	----------------------	----------------------

Überprüfe deinen Posteingang und deinen Spam-Ordner. Nicht erhalten? 

[Bestätigen](#)

In die sechs Felder ist das per E-Mail erhaltene OTP der Reihe nach einzugeben. Beim Klicken auf das Refresh-Icon wird ein neues OTP versendet. Beim Klicken auf "Bestätigen" wird das OTP überprüft und der User weitergeleitet.

Papierprototyp 3.7, 3.10

Dies ist die Menüleiste. Diese ist über alle Seiten vorhanden und ermöglicht das Navigieren innerhalb der Website.

Mit Hilfe des eigenen "Eigenes Profil" kann man eine weitere Seite betreten, in der man seine eigenen Informationen sehen kann. Dabei muss man nur drauf drücken.

Der Text "Eigenes Profil" wird den Namen des Benutzers anzeigen. und das Bild einer Person wird angepasst an das Profilbild des Nutzers insofern vorhanden.

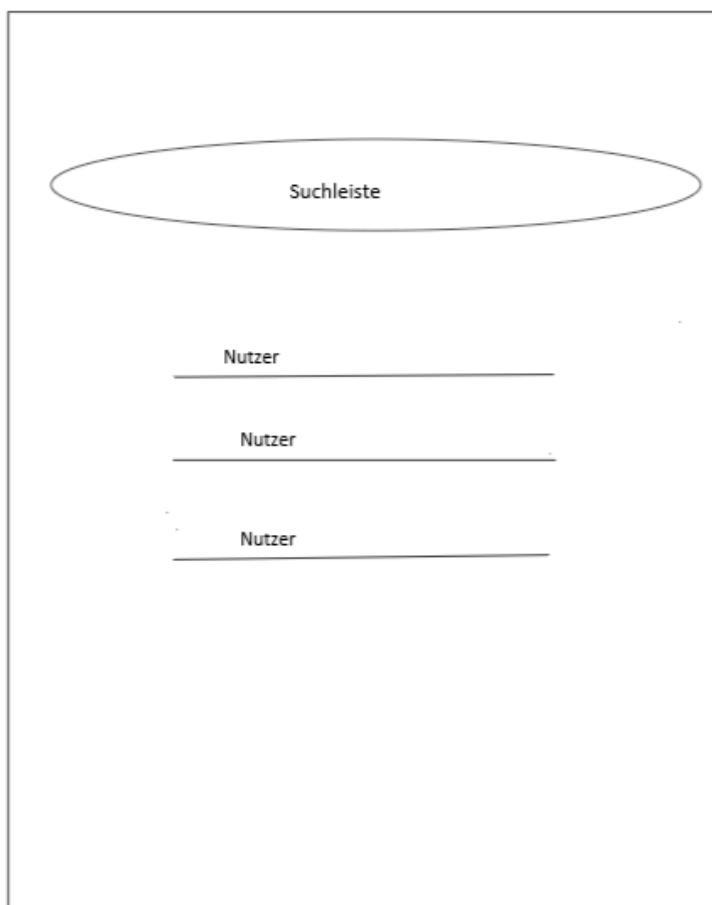


Papierprototyp 6.2, 3.10, 3.7

Drückt man auf der Menüleiste auf "Benutzer suchen" kommt man zu dieser Seite.

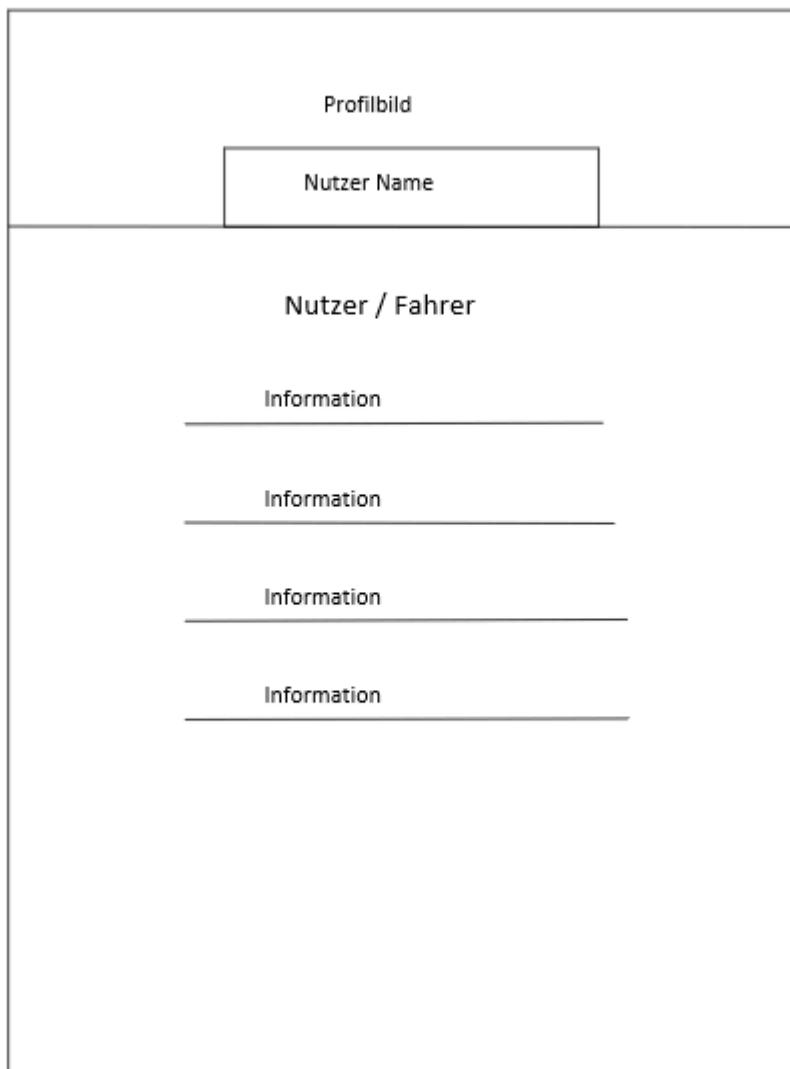
Hier kann man mittels Suchleiste andere Nutzer finden. Diese werden untereinander angezeigt.

Klickt man auf deren Benutzernamen, kann man deren Benutzerinformationen sehen.



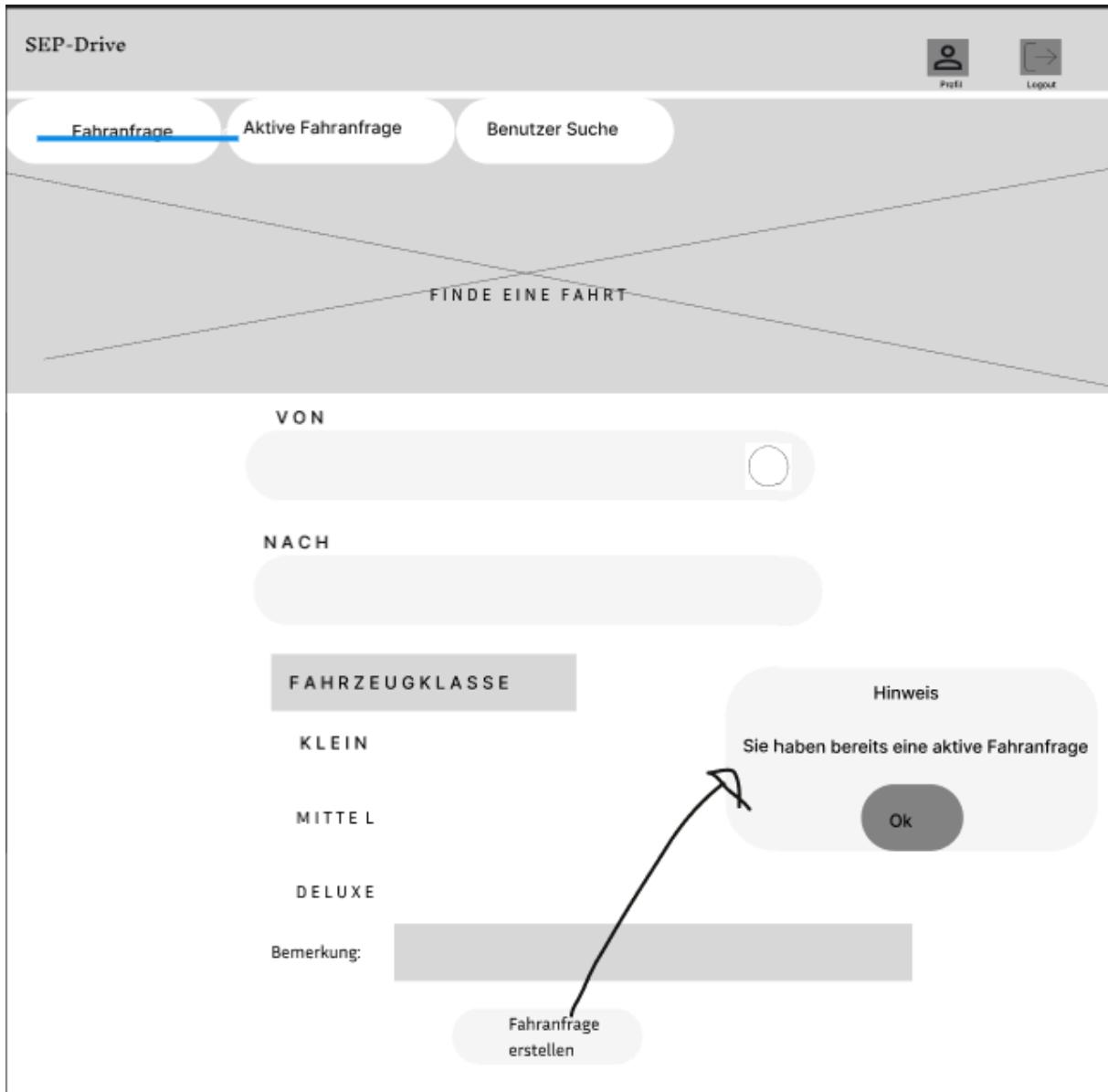
Papierprototyp 3.8, 3.9, 6.4

Drückt man auf einen Benutzernamen (eigenen oder fremden) kann man die Benutzerinformationen sehen. Diese werden untereinander angezeigt. Möchte man die Fahrerinformationen eines Benutzers sehen, kann man auf den Knopf "Fahrer" drücken. Dieser zeigt ausschließlich die Fahrerinformationen an, sofern vorhanden. Sind keine Fahrerinformationen vorhanden, weil der Nutzer kein Fahrer ist, so erscheint dieser Knopf erst nicht.



Papierprototyp 4.2, 4.6

Nach einer erfolgreichen Anmeldung/Registrierung kommt man zum Startbildschirm des Programms.



Hier kann man seine Fahranfrage erstellen. Zudem gibt es oben eine Menüleiste.

Wenn man auf die Menüleiste drückt, kommt man zu den anderen Seiten. Drückt man auf Fahranfrage wird diese Seite angezeigt.

Bei "von" und "nach" kann jeweils ein Standort ausgewählt werden, der später auf der Karte ersichtlich wird.

Papierprototyp 4.4

Wenn man auf "Aktive Fahranfrage" klickt, wird diese Seite angezeigt.

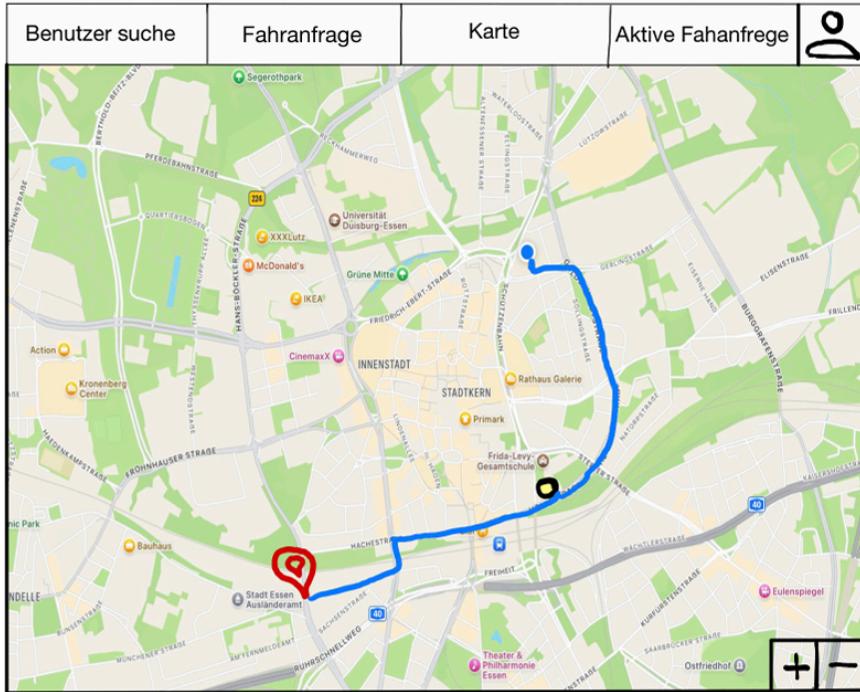


Hier kann man seine eigene Fahranfrage sehen und bei Bedarf löschen.

Papierprototyp 5.2

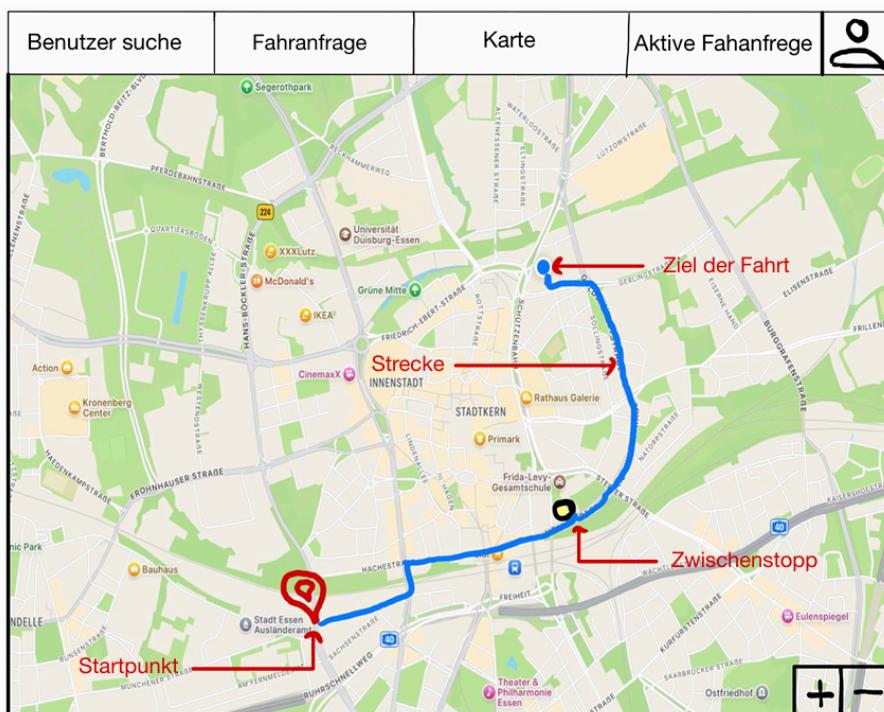
Nachdem man eine Anfrage geschickt hat, wird man zur Seite der Karte geschickt. Dort sieht man nun die Strecke der Fahrt inklusive Zwischenstopps.

Die Seite verfügt wieder über eine Menüleiste, die dir erlaubt, zwischen den Seiten zu wechseln.



Papierprototyp 5.1

Die Karte zeigt auch mithilfe von verschiedenen Markern an wo sich welche Art von Punkt befindet (Start, Zwischenstopp, Ziel)

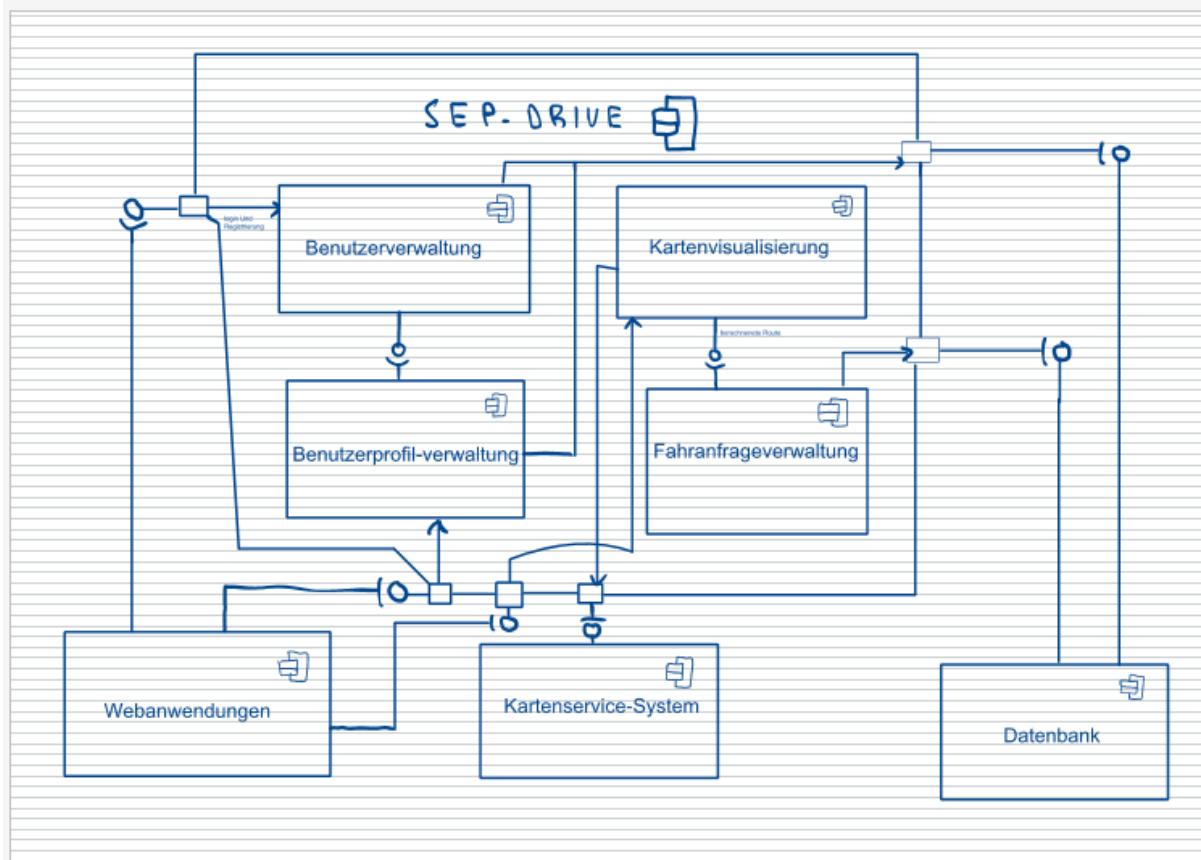
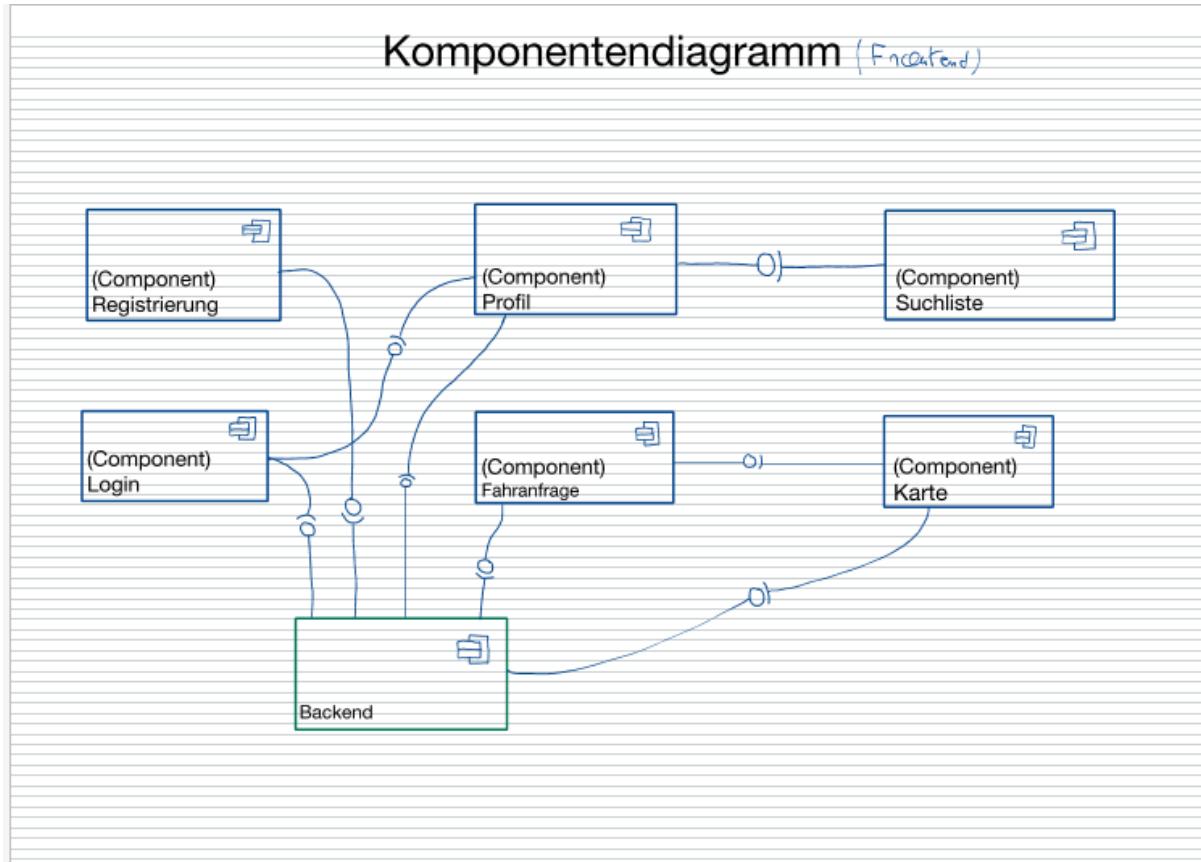


Papierprototyp 5.6

Die Karte ist durch zusammenführen der Finger auf dem Mauspad und den Knöpfen unten rechts zoombar. Zudem kann man mit dem Mauspad die Karte auch verschieben.

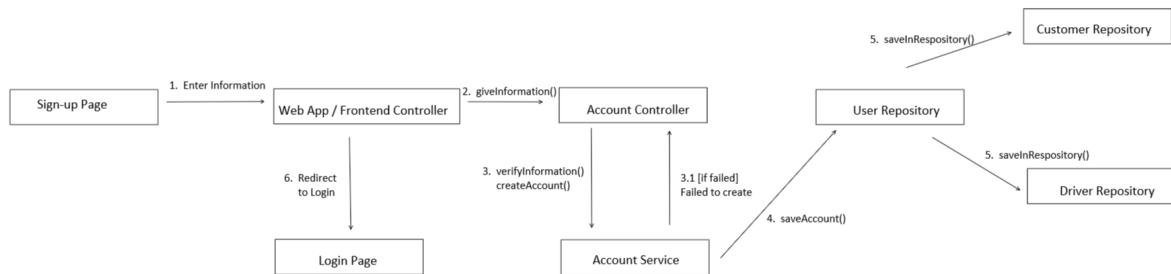


Strukturdiagramme (Komponenten- und Klassendiagramme)

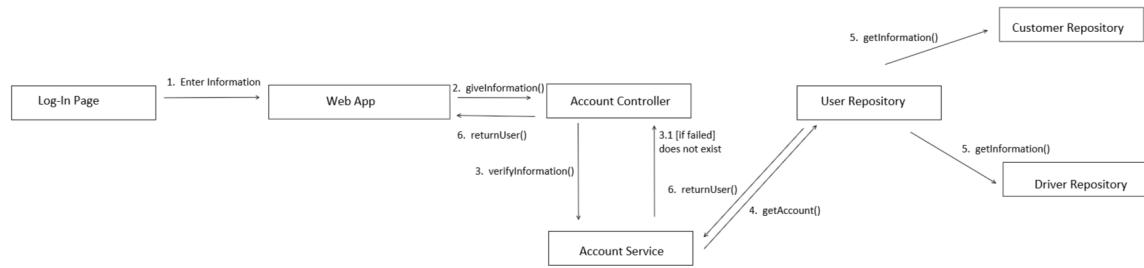


Verhaltensdiagramme (Kommunikationsdiagramme)

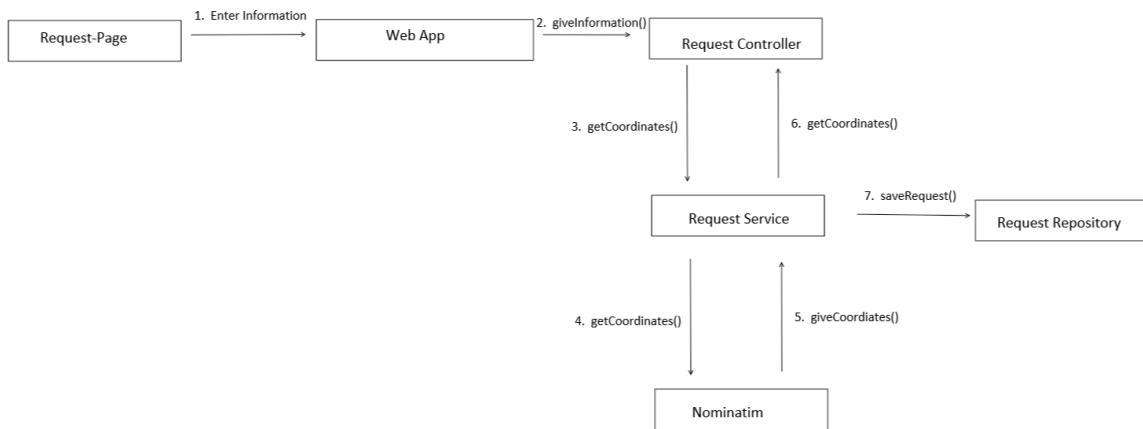
7.3 Kommunikationsdiagramm Sign-up



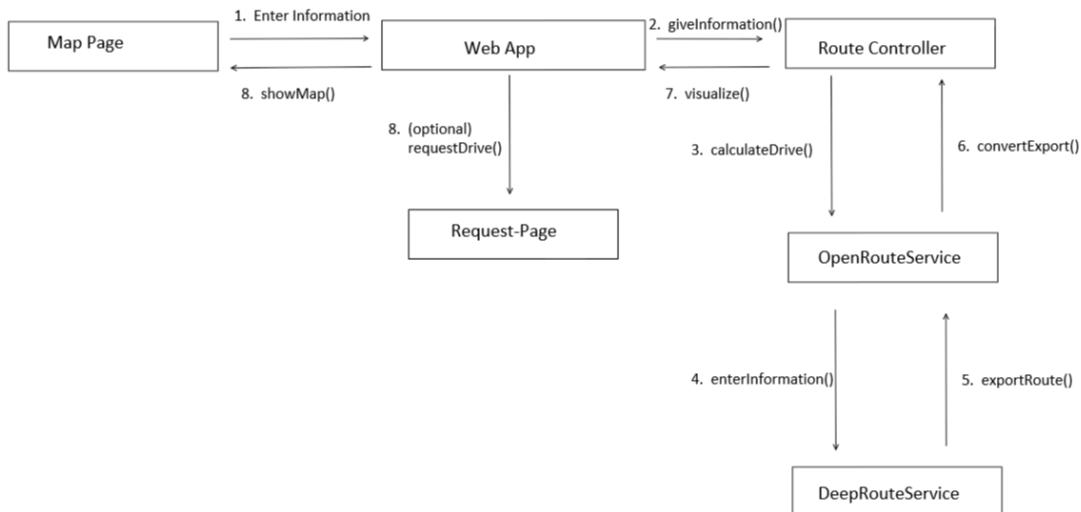
7.3 Kommunikationsdiagramm Log-In



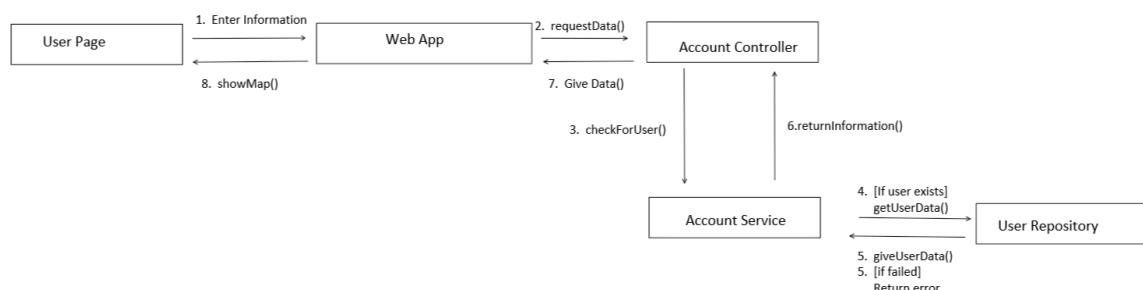
7.3 Kommunikationsdiagramm Request-Page



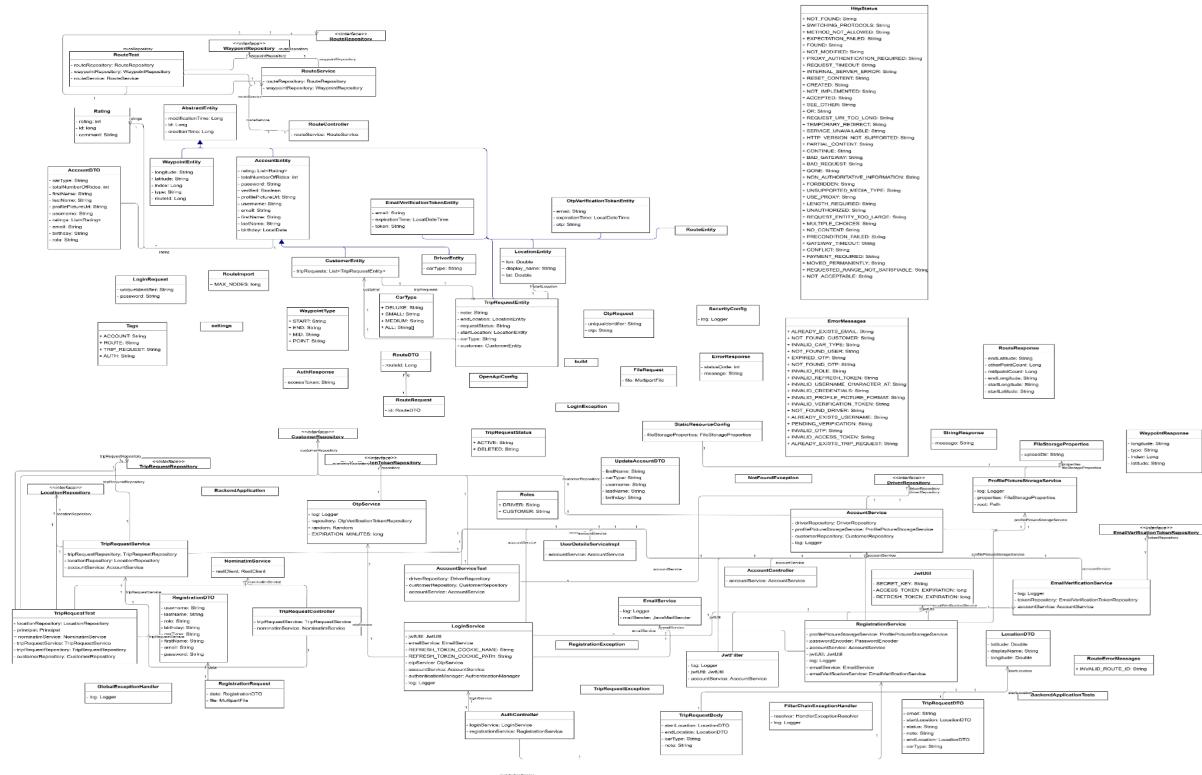
7.3 Kommunikationsdiagramm Map-Page



7.3 Kommunikationsdiagramm User-Page



7.3 Klassendiagramm



Funktionalitätsplanung

ID	Funktionalität	Verantwortlicher	Abhängige Funktionalitäten	Verknüpft e User-Stories	Quellcode-referenz	Status
1.	Registrierung von Benutzern					
1.1.1	Datenbank-Modell der abstrakten Entität	Mats Paul Tončik			entity.AbstractEntity	Fertig
1.1.2	Datenbank-Modell des abstrakten Account	Mats Paul Tončik	1.1.1		entity.AccountEntity	Fertig
1.1.3	Datenbank-Modell des Kunden	Mats Paul Tončik	1.1.2		entity.CustomerEntity	Fertig
1.1.4	Datenbank-Modell des Fahrers	Mats Paul Tončik	1.1.2		entity.DriverEntity	Fertig
1.1.5	Datenbank-Modell des Email Verifikations Tokens	Mats Paul Tončik	1.1.1		entity.EmailVerificationTokenEntity	Fertig
1.2.1	Registrierungs-Prozess starten	Mats Paul Tončik	1.1.3, 1.1.4, 1.1.5	1.1, 1.2	auth.registration.RegistrationService.register	Fertig
1.2.2	Verifizierungs-Email an User senden	Mats Paul Tončik	1.1.5	1.1, 1.2	auth.registration.RegistrationService.sendVerificationEmail, auth.email.EmailService.sendVerificationLink, email.EmailVerificationService.save	Fertig
1.2.3	Verifizierungs-Email erneut senden	Mats Paul Tončik	1.1.1, 1.2.1, 1.2.2	1.5	auth.registration.RegistrationService.resendVerificationEmail	Fertig
1.2.4	Registrierungs-Prozess abschließen	Mats Paul Tončik	1.2.1, 1.2.2, 1.2.3	1.4	auth.registration.RegistrationService.verifyEmail, email.EmailVerificationService.verify	Fertig
1.3.1	Registrierungspage	Mats Paul Tončik	1.2.1	1.3	components/register	Fertig
2.	Login von Benutzern					
2.1.1	Datenbank-Modell des OTP Verifikations Tokens	Mats Paul Tončik	1.1.1		entity.OTPVerificationTokenEntity	Fertig
2.2.1	Login-Prozess starten	Mats Paul Tončik	1.2.4	2.1	auth.login.LoginService.login	Fertig

2.2.2	OTP per E-Mail senden	Mats Paul Tončik	2.2.1	2.3	auth.OtpService.generateOtp, email.EmailService.sendOtp	Fertig
2.2.3	OTP erneut senden	Mats Paul Tončik	2.2.2	2.3		Fertig
2.2.4	Login-Prozess abschließen	Mats Paul Tončik	2.2.1	2.2, 2.3	auth.login.LoginService.verifyOtp, auth.login.OtpService.validateOtp	Fertig
2.2.5	Access-Token refreshen	Mats Paul Tončik			auth.login.LoginService.refresh	Fertig
2.2.6	JWT Hilfsklasse	Mats Paul Tončik	2.2.7		auth.JwtUtil	Fertig
2.2.7	JWT Filter	Mats Paul Tončik	2.2.1		auth.JwtFilter	Fertig
2.3.1	Loginpage	Mats Paul Tončik	2.2.1, 2.2.4		components/login	Fertig
2.3.2	Authentication Interceptor	Mats Paul Tončik	2.2.5		interceptors/auth.interceptor.ts	Fertig
2.3.3	Angular Auth Service	Mats Paul Tončik	2.2.4, 2.2.5	2.2	services/angular-auth.service	Fertig
2.3.4	Authentication Guard	Mats Paul Tončik			guards/auth.guard.ts	Fertig
3.	Kartenvisualisierung der Route					
3.1	MapComponent	Mohammed Daoudi	4.2,4.3	5.1,5.3	map.component.ts map.component.html map.component.css map.component.spec.ts	Fertig
3.2	Datenbankmodell von Routen und Waypoints	Ian Christopher Schirmer			entity.RouteEntity entity.WaypointEntity route.RouteRepository route.WaypointRepository route.WaypointType	Fertig
3.3	Import von GeoJSON	Ian Christopher Schirmer	3.2		route.RouteService route.FileRequest route.jsonimporter.RouteImport	Fertig
3.4	Routen/Waypoints an das Frontend bereitstellen	Ian Christopher Schirmer	3.2		route.RouteDTO route.RouteController route.RouteService route.RouteRequest route.response.RouteResponse route.response.WaypointResponse route.RouteErrorMessages	Fertig

4.	Fahr Anfrage erstellen					
4.1.1	Fahr Anfrage Modell in Datenbank	Thomas Fabio Kaiser	4.2	4.1	entity.TripRequestEntity entity.LocationEntity TripRequestRepository	Fertig
4.1.2	Location Modell in Datenbank	Thomas Fabio Kaiser	4.2	4.1	entity.LocationEntity LocationRepository	Fertig
4.2	Fahr Anfrage erstellen	Thomas Fabio Kaiser	4.2.1, 4.3, 4.1.1, 4.1.2, 4.5	4.1, 4.1.1, 4.1.2, 4.1.3	TripRequestController(create), TripRequestService, TripRequestRepository , TripRequestEntity, LocationEntity, TripRequestStatus, TripRequestBody	Fertig
4.2.1	Informationen aus Frontend erhalten	Thomas Fabio Kaiser	4.2	4.1.1, 4.1.2, 4.1.3	TripRequestController, TripRequestService, TripRequestBody, LocationDTO	Fertig
4.3	Koordinaten erhalten	Thomas Fabio Kaiser	4.2.1	4.1.1, 4.1.2	TripRequestController(searchLocations), TripRequestService, Package: Nominatim	Fertig
4.4	Fahr Anfrage löschen	Thomas Fabio Kaiser	4.1.1	4.3	TripRequestController (deleteCurrentActiveTripRequest), TripRequestService, TripRequestRepository	Fertig
4.4.1	Fahr Anfrage an Frontend schicken	Thomas Fabio Kaiser	4.1.1, 4.4	4.3	TripRequestController(getCurrentActiveTripRequest), TripRequestService, TripRequestDTO	Fertig
4.5	Fahranfrage Formular erstellen	Reine Christiane Demgne	4.2,4.6,4.1	4.2	fahranfrage-erstellen.component.html fahranfrage-erstellen.component.ts fahranfrage-erstellen.component.css	Fertig
4.5	Fahranfrage als aktiv markieren	Reine Christiane Demgne	4.1	4.4	aktive-fahranfrage.component.html aktive-fahranfrage.component.ts aktive-fahranfrage.component.css	Fertig
4.6	Startpunkt und Ziel angeben	Reine Christiane Demgne	4.3, 4.1	4.2,4.6	fahranfrage-erstellen.component.html	Fertig

					fahranfrage-erstellen.component.ts fahranfrage-erstellen.component.css	
4.7	Aktive Fahranfrage löschen	Reine Christiane Demgne	4.1,4.4	4.4	aktive-fahranfrage.component.html aktive-fahranfrage.component.ts aktive-fahranfrage.component.css	Fertig
4.8	Hinweisdialog	Reine Christiane Demgne	4.5,4.1	4.5	active-ride-dialog.com ponent.html active-ride-dialog.com ponent.ts active-ride-dialog.com ponent.css	Fertig
4.9	Löschdialog	Reine Christiane Demgne	4.1,4.4	4.4	delete-ride-dialog.com ponent.css delete-ride-dialog.com ponent.html delete-ride-dialog.com ponent.ts	Fertig
5.	Benutzerseiten					
5.1	Benutzerseite	Amin Rais	5.3, 5.5,	3.7,3.8, 3.9,3.10	profile profile.html profile.css profile.ts	fertig
5.2	Benutzerliste	Amin Rais	5.3, 5.5	6.3, 6.4, 6.5, 3.7,	profile-search profile-search.html profile-search.css profile-search.ts	fertig
5.3	Benutzerprofil suchen	Moyo Fred Aymerick	1.1.2, 1.1.3, 1.1.4	3.1, 3.5, 3.4	AccountService.java AccountController.java AccountDTO.java	
5.4	Benutzerprofil ändern (Vorname, Nachname, Profilbild...)	Moyo Fred Aymerick	1.1.2, 1.1.3, 1.1.4	3.1, 3.5, 3.4	AccountService.java AccountController.java AccountDTO.java UpdateAccountDto.java	
5.5	Benutzerprofil anzeigen	Moyo Fred Aymerick	1.1.2, 1.1.3, 1.1.4	3.1, 3.5, 3.4	AccountService.java AccountController.java AccountDTO.java	

Systemtests

Datum	20.05.2025		
Tester	Markus Tohn		
SW-Version	V 1.0.0		
Vorbedingung(en)	Nutzer "Markus Tohn" mit username "Markus0123" existiert nicht im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer gibt bei der Registrierung seine persönlichen Daten.	Das System zeigt die eingegebenen Daten im Formular an.	✓
2	Der Benutzer gibt das Passwort "Passwort" auf der Tastatur ein.	Das System zeigt das Passwort durch „*"-Symbole zensiert an.	✓
3	Der Benutzer klickt auf "Registrieren".	Das System zeigt die Meldung "Verifizierungs-Email erfolgreich versandt!" auf dem Display an. Die Email wurde an die angegebene Email-Adresse geschickt.	✓
4	Der Benutzer klickt auf den Verifizierungs-Link in der Email.	Die Email-Adresse wird verifiziert und der Benutzer ist im System angemeldet.	✓
Nachbedingung(en)	Der Benutzer "Markus Tohn" ist im System registriert und angemeldet.		
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Markus Tohn		
SW-Version	V 1.0.0		
Vorbedingung(en)	Der Benutzer "Markus Tohn" ist im System angemeldet.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X

1	Der Benutzer klickt oben rechts auf sein Profil.	Das System zeigt ein Dropdown-Menü an.	✓
2	Der Benutzer klickt auf "Logout".	Das System meldet den Benutzer ab.	✓
Nachbe-dingung(en)	Der Benutzer "Markus Tohn" wurde abgemeldet.		✓
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Markus Tohn		
SW-Version	V 1.0.0		
Vorbedin-gung(en)	Der Benutzer "Markus Tohn" ist im System registriert.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer gibt im Login-Formular seine Benutzerdaten an.	Das System zeigt die Daten an, das Passwort als *	✓
2	Der Benutzer klickt auf "Anmelden".	Das System startet den Anmeldeprozess und verschickt ein OTP per Email.	✓
3	Der Benutzer gibt das OTP an.	Das System zeigt das OTP an.	✓
4	Der Benutzer klickt auf "Bestätigen".	Das System leitet auf die Startseite weiter.	✓
Nachbe-dingung(en)	Der Benutzer "Markus Tohn" ist angemeldet.		
Testurteil	Test bestanden.		

Datum	20.05.2025
Tester	Hakan Oruc

SW-Version	V 1.0.0		
Vorbedin-gung(en)	Der Benutzer "Hakan Oruc" muss angemeldet sein.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Benutzer klickt auf Fahranfrage.	Das System zeigt ein Formular für die Erstellung einer Fahranfrage	<input checked="" type="checkbox"/>
2	Der Benutzer füllt das Formular aus und klickt auf Fahranfrage erstellen.	Das System meldet bei gleichen Adressen einen Fehler: "Start- und Zieladresse dürfen nicht gleich sein." Wenn sie nicht gleich sind, erstellt das System erfolgreich eine Fahranfrage. Das jedoch nur, wenn der User keine aktive Fahranfrage hat.	<input checked="" type="checkbox"/>
3	Nachdem der Benutzer auf Fahranfrage erstellen geklickt hat kommt man zu aktive Fahranfrage. Der Benutzer klickt dann auf den Button Löschen.	Das System löscht die Fahranfrage und es kann eine neue Fahranfrage gestellt werden.	
Nachbe-dingun g(en)	Der Benutzer "Markus Tohn" wurde abgemeldet.		
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Delgesch Ali Daud und Lukas Polfsuß		
SW-Version	V 1.0.0		
Vorbedin-gung(en)	Der Nutzer ist mit seinem Account angemeldet		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Benutzer klickt auf das Profilbild mit den Initialien	Die Buttons "Profile" und "Log out" werden angezeigt	<input checked="" type="checkbox"/>
2	Der Nutzer klickt auf Profile	Das System öffnet die eigene Benutzeroberseite mit den korrekten Daten des Accounts	<input checked="" type="checkbox"/>

Nachbe-dingung(en)	Der Nutzer sieht die Daten von sich	√
Testurteil	Test bestanden.	

Datum	20.05.2025		
Tester	Delgesch Ali Daud und Lukas Polfsuß		
SW-Version	V 1.0.0		
Vorbedingung(en)	In dem System sind Accounts registriert und der Nutzer ist angemeldet.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	√ / X
1	Der Benutzer geht auf Social	Alle registrierten Accounts werden angezeigt	√
2	Der Nutzer sucht in der Searchbar nach einem Benutzernamen	Das System filtert nach alle Accounts die der Suche entsprechen	√
3	Der Nutzer klickt auf einen Account	Das System zeigt die Accountdaten an	√
Nachbe-dingung(en)	Der Nutzer hat einsicht über die Accountdaten eines anderen Nutzers		
Testurteil	Test bestanden.		

Datum	20.05.2025		
Tester	Kaviksana Ramkumar		
SW-Version	V 1.0.0		
Vorbedingung(en)	GeoJSON Karte von OpenRouteService und Benutzer angemeldet.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	√ / X
1	Der Benutzer drückt auf "Karte".	Die Karte sollte angezeigt werden.	√

2	Der Benutzer sollte auf “GeoJSON auswählen” gehen.	Der Benutzer kann eine GeoJSON-Datei auswählen.	✓
3	Der Benutzer wählt eine GeoJSON-Datei aus	Unter dem “GeoJSON auswählen”-Fläche wird der Dateiname der ausgewählten Datei angezeigt.	✓
4	Der Benutzer klickt auf den Pfeil	Die Route aus der Datei wird angezeigt.	✓
Nachbedingung(en)	Der Benutzer sieht die Karte		
Testurteil	Test bestanden.		

Zyklus II

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Geldkonten			
1.1	Aufladen des Geldkontos	User-Story	Amin Rais	Fertig
1.2	Überweisen des Fahrpreises	User-Story	Amin Rais	Fertig
1.3	Anzeige für Kontoaufladung	Papierprototyp	Amin Rais	Fertig
2.	Liste der verfügbaren Fahranfragen			
2.1	Tabelle für die Liste der verfügbare Fahranfrage	Papierprototyp	Reine Christiane Demgne	fertig
2.2	Eingabefeld für die Aktuelle Adresse der Fahrer	Papierprototyp	Reine Christiane Demgne	fertig
2.4	Entfernung zwischen der aktuellen Position des Fahrers und dem Startpunkt der Fahrt	User-Story	Moyo Fred Aymerick	Fertig
2.5	Liste der verfügbaren Fahranfragen	User Story	Moyo Fred Aymerick	Fertig
3.	Fahrtplanung			
3.1	Automatische Erstellung der Route	User-Story	Thomas Fabio Kaiser	Fertig
3.1.1	Route bei Einsicht der aktiven Fahrt Anfrage ansehen	User-Story	Thomas Fabio Kaiser	Fertig
3.2	Berechnung der Gesamtdistanz, Gesamtdauer und Fahrpreis	User-Story	Thomas Fabio Kaiser	Fertig
3.3	Anzeige der Route auf Karte	Papier Prototyp	Thomas Fabio Kaiser	Fertig
3.4	Anzeige der Route bei Einsicht der aktiven Fahr Anfrage	Papier Prototyp	Thomas Fabio Kaiser	Fertig
4.	Fahrtangebote			
4.1	Live-Benachrichtigung über neue Fahrtangebote	Papier Prototyp	Daoudi Mohammed	fertig
4.2	Angebotsübersicht für Kunden	Papier Prototyp	Daoudi Mohammed	fertig
4.3	Fahrangebot abgeben	User-Story	Moyo Fred Aymerick	fertig

4.4	Fahrangebot zurückziehen	User-Story	Moyo Fred Aymerick	fertig
4.5	Fahrangebot annehmen oder ablehnen	User-Story	Moyo Fred Aymerick	fertig
4.6	Fahrer Fahranfrage übernehmen	Papier Prototyp	Daoudi Mohammed	In Bearbeitung
5.	Durchführung einer Fahrt			
5.1	Start der Simulation	User-Story	Mats Paul Tončik	Fertig
5.2	Stop der Simulation	User-Story	Mats Paul Tončik	Fertig
5.3	Anpassung der Geschwindigkeit	User-Story	Mats Paul Tončik	Fertig
5.4	Bewertung der Fahrt	User-Story	Mats Paul Tončik	Fertig
5.5	Verlauf der Simulation	User-Story	Mats Paul Tončik	Fertig
5.6	Route der Simulation	User-Story	Mats Paul Tončik	Fertig
5.7	Wichtige Punkte der Route	User-Story	Mats Paul Tončik	Fertig
5.8	Simulation	Papierprototyp	Mats Paul Tončik	Fertig
6.	Fahrthistorie			
6.1	Anzeige der Fahrthistorie	User-Story	Ian Christopher Schirmer	Fertig
6.2	Sortierung der Einträge	User-Story	Ian Christopher Schirmer	Fertig
6.3	Suche nach Personen	User-Story	Ian Christopher Schirmer	Fertig
6.4	Tabelle für Fahrt Historie	Papierprototyp	Reine Christiane Demgne	fertig
6.5	Fahrer/Kunde über Suchleiste suchen	Papierprototyp	Reine Christiane Demgne	fertig
7.	Architektur			
7.1	Komponentendiagramm		Daoudi Mohammed	Fertig
7.2	Klassendiagramm		Fred Aymerick Moyo	Fertig
7.3	Kommunikationsdiagramm		Thomas Fabio Kaiser	Fertig

User-Stories

User Story-ID	4.5
User Story-Beschreibung	Als Kunde möchte ich ein eingegangenes Fahrangebot akzeptieren oder ablehnen können, sodass ich selbst entscheide, mit welchem Fahrer ich fahren möchte.

Geschätzter Realisierungsaufwand	1 tag
Priorität	Mittel
Autor	fred
Abhängigkeiten zu anderen User Stories	2.1;2.2;2.3;2.4

User Story-ID	4.4
User Story-Beschreibung	Als Fahrer möchte ich mein abgegebenes Angebot zurückziehen können, sodass ich meine Entscheidung ändern kann, solange der Kunde noch nicht reagiert hat.
Geschätzter Realisierungsaufwand	1 tag
Priorität	Mittel
Autor	fred
Abhängigkeiten zu anderen User Stories	2.1;2.2;2.3;2.4

User Story-ID	4.5
User Story-Beschreibung	Titel: Fahrangebot für eine Anfrage abgeben Als Fahrer möchte ich auf eine offene Fahranfrage ein Angebot abgeben können, sodass der Kunde sehen kann, dass ich bereit bin, ihn zu fahren.
Geschätzter Realisierungsaufwand	1 tag
Priorität	Mittel
Autor	fred
Abhängigkeiten zu anderen User Stories	2.1;2.2;2.3;2.4

User Story-ID	1.1
User Story-Beschreibung	Als Nutzer ich mein Geldkonto aufladen können, damit ich mit dem Geld nachher Fahrten bezahlen kann
Geschätzter Realisierungsaufwand	5 Tage
Priorität	Hoch
Autor	Amin Rais

Abhängigkeiten zu anderen User Stories	1.2, 1.3
---	----------

User Story-ID	1.2
User Story-Beschreibung	Als Nutzer möchte ich, das zum Ende einer Fahrt, die Fahrtkosten vom Geldkonto des Kunden angezogen werden und dem Fahrer auf sein Konto überwiesen werden
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	1.2, 1.3

User Story-ID	3.1
User Story-Beschreibung	Als Kunde möchte ich bei Erstellung der Fahrt Anfrage live die erstellte Route ansehen können, sodass ich die erstellte Route nachvollziehen kann.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	3.1.1, 3.2

User Story-ID	3.1.1
User Story-Beschreibung	Als Kunde möchte ich bei der Einsicht meiner aktiven Fahrt Anfrage meine Route anschauen können, sodass ich den Status meiner Route anschauen kann.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	3.1, 3.2

User Story-ID	3.2
User Story-Beschreibung	Als Kunde möchte ich meine Gesamtdistanz, Gesamtdauer und Preis der Fahrt ansehen können, sodass ich mehr Informationen über meine Fahrt habe.
Geschätzter Realisierungsaufwand	4 Stunden
Priorität	Mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	3.1, 3.1.1

User Story-ID	5.1
User Story-Beschreibung	Als Teilnehmer einer Fahrt möchte ich die Simulation starten können, sodass sich der Marker bewegt.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	5.2, 5.3

User Story-ID	5.2
User Story-Beschreibung	Als Teilnehmer einer Fahrt möchte ich die Simulation stoppen können, sodass sich der Marker nicht mehr bewegt.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	5.1, 5.3

User Story-ID	5.3
User Story-Beschreibung	Als Teilnehmer einer Fahrt möchte ich die Simulationsgeschwindigkeit anpassen können, um die Geschwindigkeit des Markers zu erhöhen.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	5.1, 5.2

User Story-ID	5.4
User Story-Beschreibung	Als Teilnehmer der Fahrt möchte ich die Fahrt bewerten können, sodass der Partner sehen kann, wie ich die Fahrt fand.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	mittel
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	5.5
User Story-Beschreibung	Als Teilnehmer der Fahrt möchte ich den Marker auf der Karte sehen, sodass ich weiß wie weit die Simulation vorangeschritten ist.
Geschätzter Realisierungsaufwand	6 Stunden
Priorität	hoch
Autor	Mats Paul Tončik

Abhängigkeiten zu anderen User Stories	
---	--

User Story-ID	5.6
User Story-Beschreibung	Als Teilnehmer der Fahrt möchte ich die Route auf der Karte sehen, sodass ich weiß entlang welcher Route die Fahrt geht.
Geschätzter Realisierungsaufwand	30 Minuten
Priorität	hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	5.7
User Story-Beschreibung	Als Teilnehmer der Fahrt möchte ich wichtige Punkte auf der Karte sehen, sodass ich weiß wo die Fahrt startet und endet.
Geschätzter Realisierungsaufwand	30 Minuten
Priorität	hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	6.1
User Story-Beschreibung	Als Benutzer möchte ich die eigene Fahrthistorie als Tabelle einsehen können.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	hoch
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	

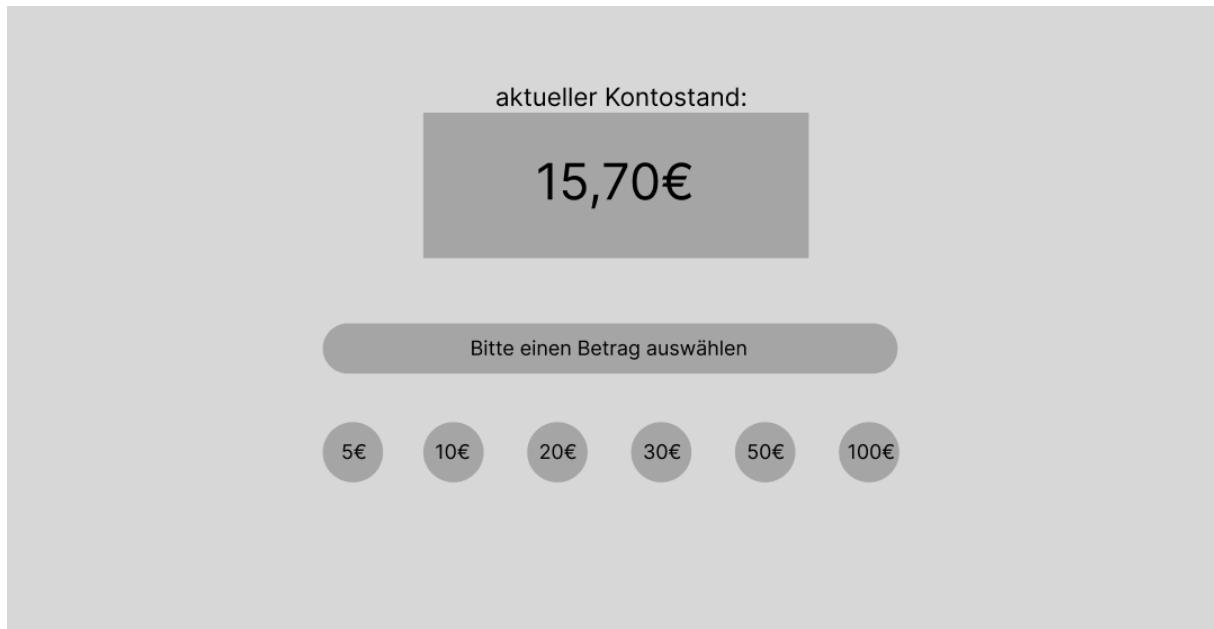
User Story-ID	6.2
User Story-Beschreibung	Als Benutzer möchte ich die Tabelle der Fahrthistorie nach jeder Spalte sortieren.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	6.1

User Story-ID	6.3
User Story-Beschreibung	Als Kunde möchte ich gezielt nach Fahrten eines Fahrers suchen, als Fahrer umgekehrt.

Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	6.1

Papierprototypen

ID: 1.4



ID: 2.1, 2.2, 4.3

Wenn man auf "Verfügbare Fahranfragen" klickt, kommt man zu dieser Seite.

The image shows a hand-drawn wireframe of a mobile application interface. At the top left is a logo consisting of three horizontal lines inside a circle, with a curved arrow pointing from the right side of the logo towards the text "5EP-Drive". To the right of the logo is a user icon. Below the header, there is a navigation bar with four items: "Fahranfrage erstellen" (red background), "Aktive Fahranfrage" (green background), "Verfügbare Fahranfragen" (highlighted in blue and red, indicating the current view), and "Fahrhistorie" (red background). To the right of the navigation bar is a search bar with a magnifying glass icon and a "Speichern" button. The main content area is a table with the following columns: Fahrt-ID, Datum, Uhrzeit, Entfernung, Kundenname, Kundenbewertung, Gewünschte Fahrzeugklasse, and Option. The rows contain the following data:

Fahrt-ID	Datum, Uhrzeit	Entfernung	Kundenname	Kundenbewertung	Gewünschte Fahrzeugklasse	Option
123	5/10/2026 15:30	10 Km	Ali Lao	★★★★★	Mittel	Annehmen
124	-	-	-	-	-	Annehmen
...	-	-	-	-	-	Annehmen
...	-	-	-	-	-	Annehmen
130	-	-	-	-	-	Annehmen

ID: 3.3

Suchleiste <input type="text"/> →	
☰ Startadresse	
☰ Zwischenstop	
☰ Endadresse	
Autoklasse: <input type="radio"/> Klein <input checked="" type="radio"/> Mittel <input type="radio"/> Deluxe	
Distanz : _____	
Preis : _____	
Dauer : _____	<input type="button" value="Fahrt erstellen"/>

Vorschlag 1
Vorschlag 2
Vorschlag 3
Vorschlag 4

Karte Karte Karte Karte Karte Karte

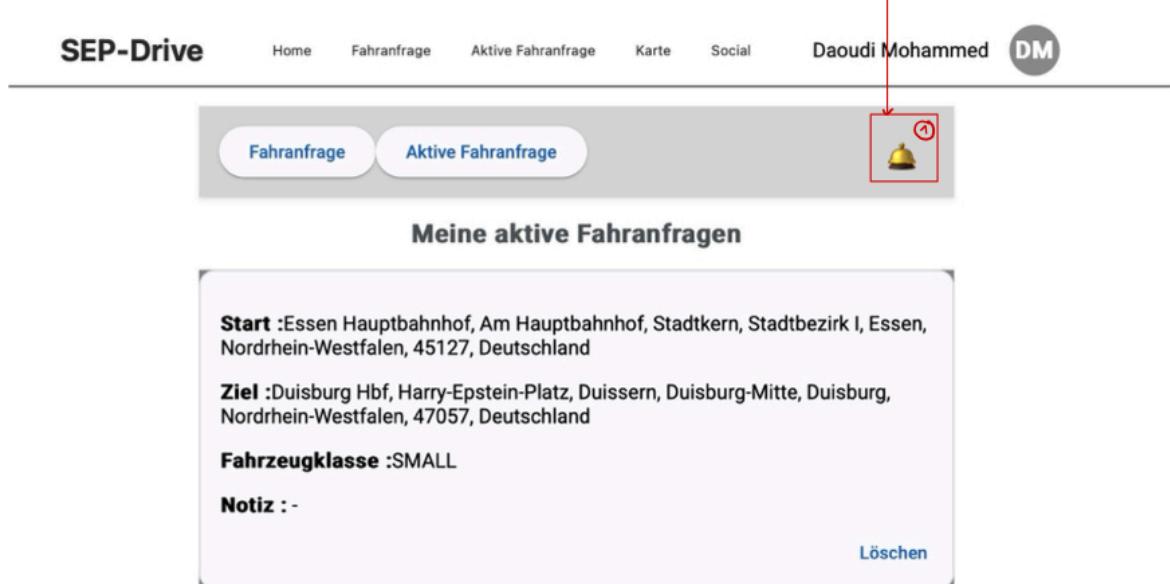
ID: 3.4

Aktuelle Fahrt
☰ Startadresse
☰ Zwischenstop
☰ Zwischenstop
☰ Endadresse
Autoklasse : <u>Mittel</u>
Distanz: _____
Preis : _____
Dauer: _____ <input type="button" value="Löschen"/>

Karte Karte Karte Karte Karte Karte

ID: 4.1

Wenn der Kunde auf  klickt dann sieht er die Angebote



The screenshot shows the SEP-Drive application interface. At the top, there is a navigation bar with the logo "SEP-Drive" and links for "Home", "Fahranfrage", "Aktive Fahranfrage", "Karte", "Social", and a user profile "Daoudi Mohammed DM". A red arrow points from the text above to a red-bordered square containing a yellow bell icon with a red circle and a number "1" on it. Below the navigation bar, there are two tabs: "Fahranfrage" and "Aktive Fahranfrage". The "Aktive Fahranfrage" tab is selected. Underneath, a section titled "Meine aktive Fahranfragen" displays a single active trip entry. The trip details are as follows:

Start : Essen Hauptbahnhof, Am Hauptbahnhof, Stadtteil, Stadtbezirk I, Essen, Nordrhein-Westfalen, 45127, Deutschland

Ziel : Duisburg Hbf, Harry-Epstein-Platz, Duissern, Duisburg-Mitte, Duisburg, Nordrhein-Westfalen, 47057, Deutschland

Fahrzeugklasse : SMALL

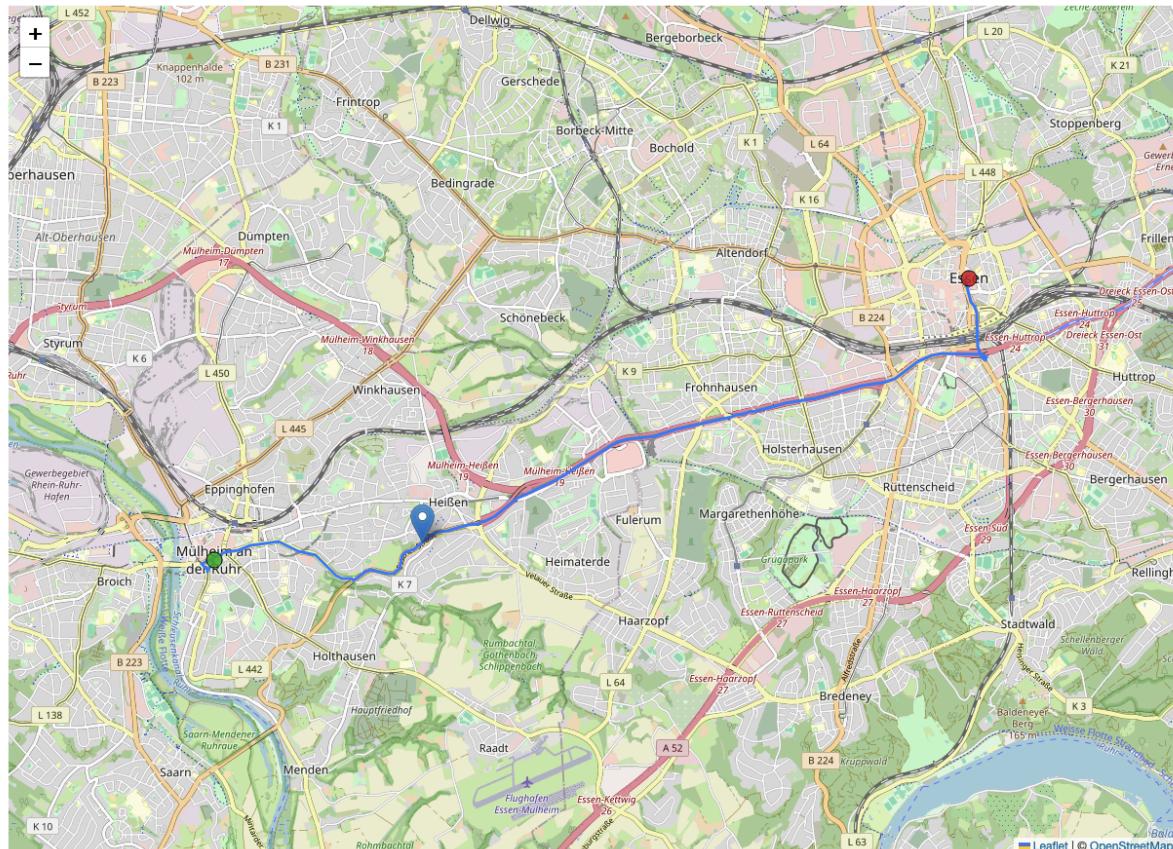
Notiz : -

At the bottom right of this card is a blue "Löschen" (Delete) button.

ID: 4.2

🚗 Angebote für deine aktive Fahranfrage				
Fahrername	Bewertung	Fahrten	Distanz	
Anna F.		18	7,2 km	Annehmen Ablehnen
Jonas M.		3	7,9 km	Annehmen Ablehnen

ID: 5.8



Der Benutzer kann die Fahrt simulieren. Dafür kann er über den Knopf unten links die Simulation starten bzw. pausieren. Mit dem Slider lässt sich die Simulationsgeschwindigkeit zwischen mindestens 3 Sekunden und maximal 30 Sekunden einstellen. Der blaue Marker auf der Karte zeigt die aktuelle Position der Simulation an.

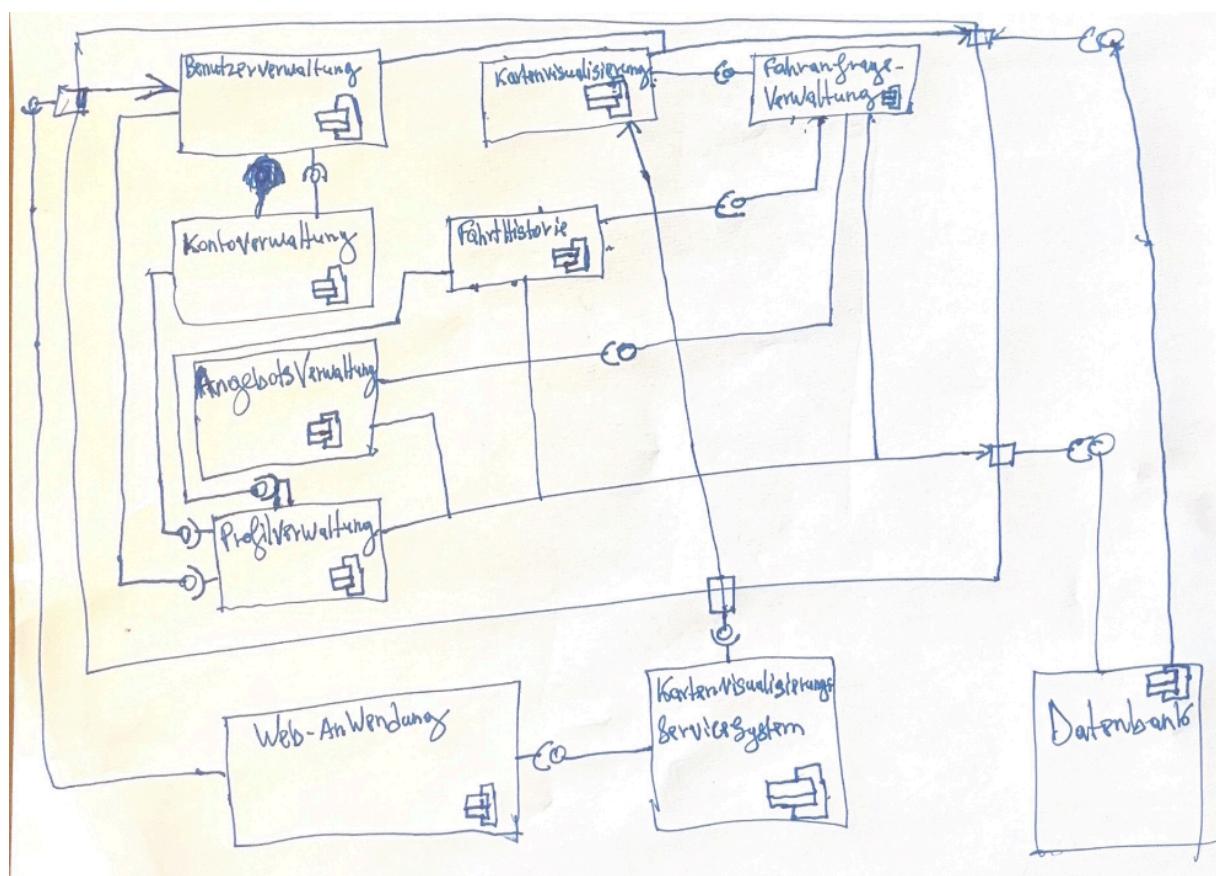
ID: 6.4, 6.5

Wenn man auf Fahrthistorie klickt, kommt man zu diese Seite

The image shows a hand-drawn wireframe of a mobile application interface. At the top, there is a header with three horizontal lines on the left, the text "SEP-Drive" in the center, and a user icon on the right. Below the header, there is a navigation bar with four items: "Fahranfrage erstellen" (highlighted with a red box), "Aktive Fahranfrage" (highlighted with a red box), "Verfügbare Fahranfragen" (highlighted with a green box), and "Fahrthistorie" (highlighted with a green box). To the right of the navigation bar is a search bar containing the placeholder "Fahrer / Kundenname" with a downward arrow icon, and a "suchen" button below it. The main content area is a table with 12 columns and several rows of data. The columns are labeled: Fahrt-ID, Datum/Uhrzeit, Distanz, Fahrt-Dauer, Betrag, Kunden-bew., Fahrer-bew., Kunden-name, Kunden-benutzer-name, Fahrer-name, and Fahrer-benutzer-name. The first row contains data: 123, 4.05.2025 / 14:00, 10 Km, 20 min, 10 €, ★★★★, ★★★★, Ali Leo, ali20, Fred Moyo, Fred10. The second row has three dots in each column. The third row has dashes in each column.

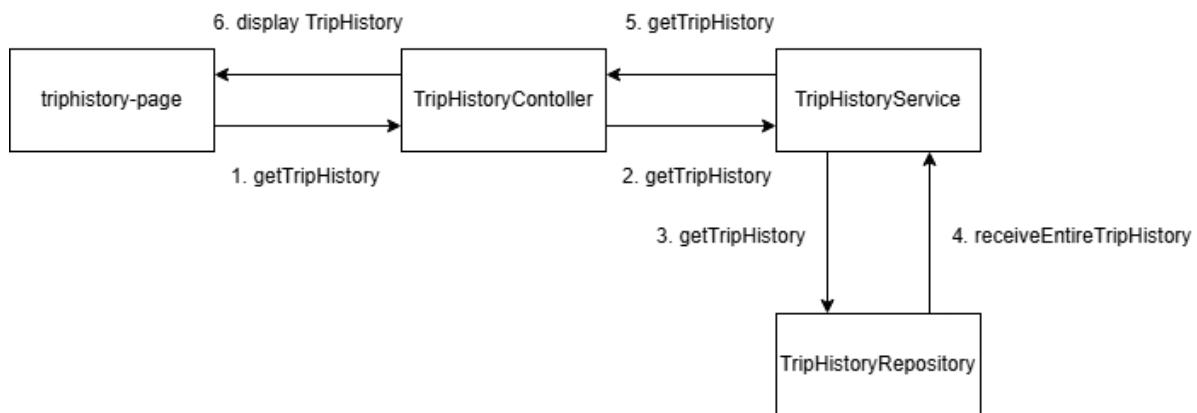
Fahrt-ID	Datum/Uhrzeit	Distanz	Fahrt-Dauer	Betrag	Kunden-bew.	Fahrer-bew.	Kunden-name	Kunden-benutzer-name	Fahrer-name	Fahrer-benutzer-name
123	4.05.2025 / 14:00	10 Km	20 min	10 €	★★★★	★★★★	Ali Leo	ali20	Fred Moyo	Fred10
...							
-	-	-	-	-	-	-	-	-	-	-

Strukturdiagramme (Komponenten- und Klassendiagramme)

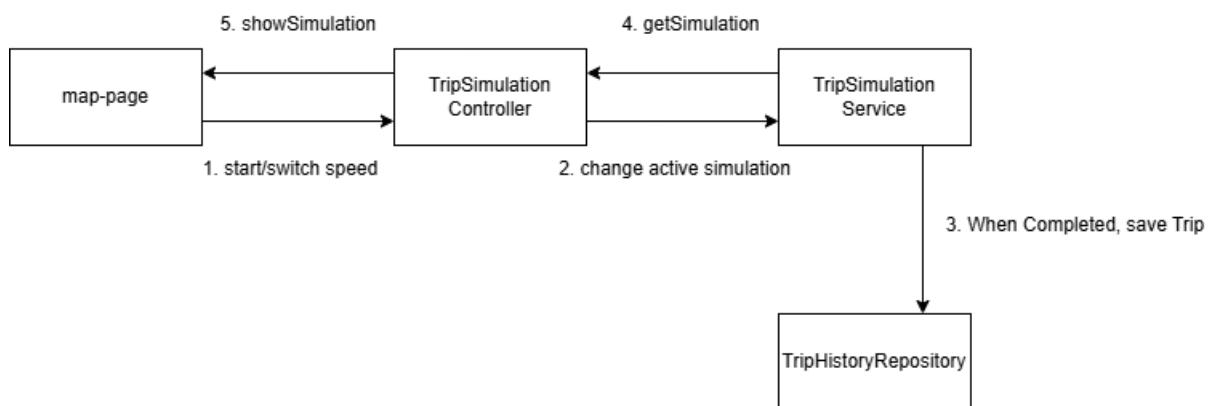


Verhaltensdiagramme (Kommunikationsdiagramme)

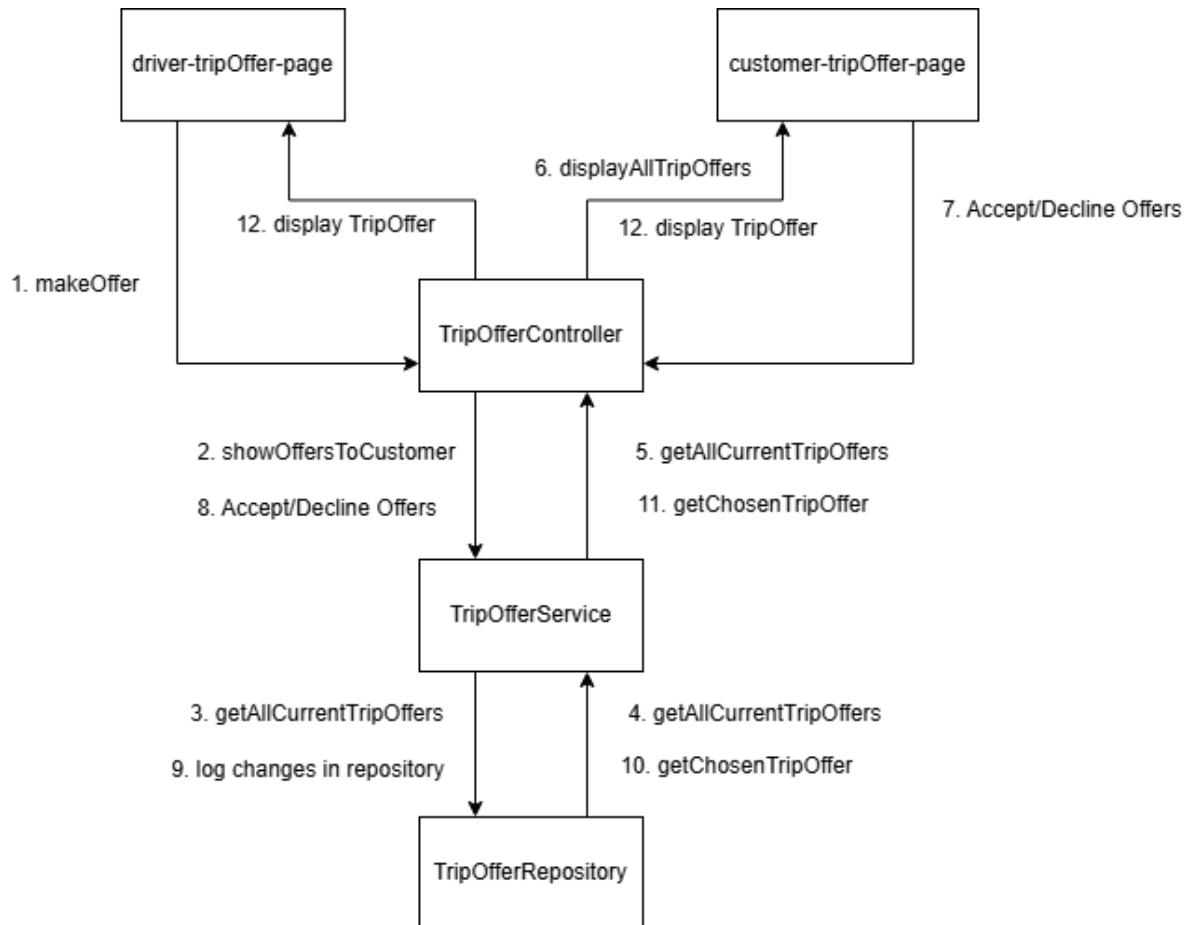
7.3 Kommunikationsdiagramm Fahrt Historie:



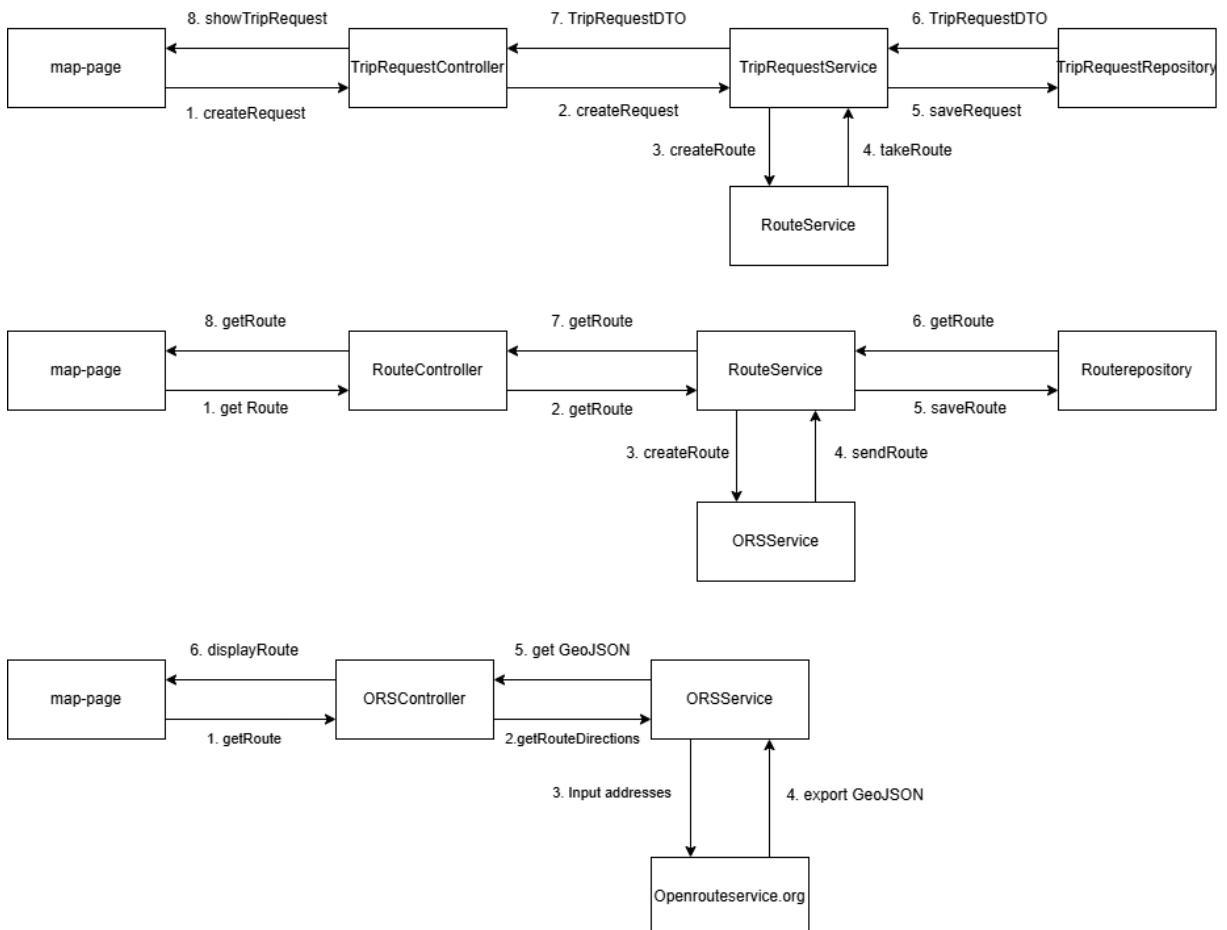
7.3 Kommunikationsdiagramm Fahrsimulation:



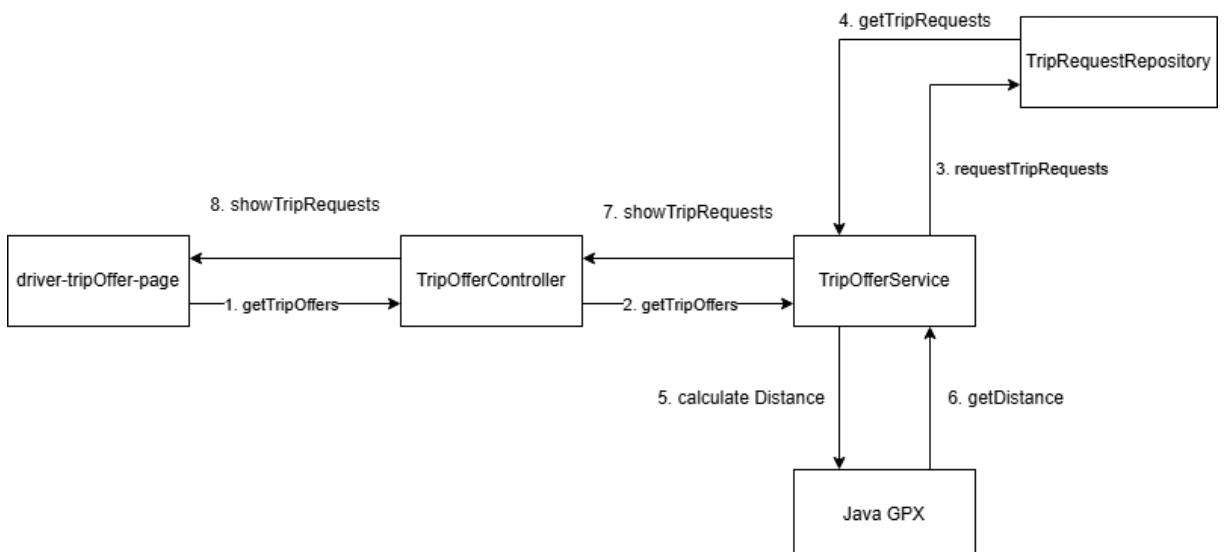
7.3 Kommunikationsdiagramm Fahrangebot:



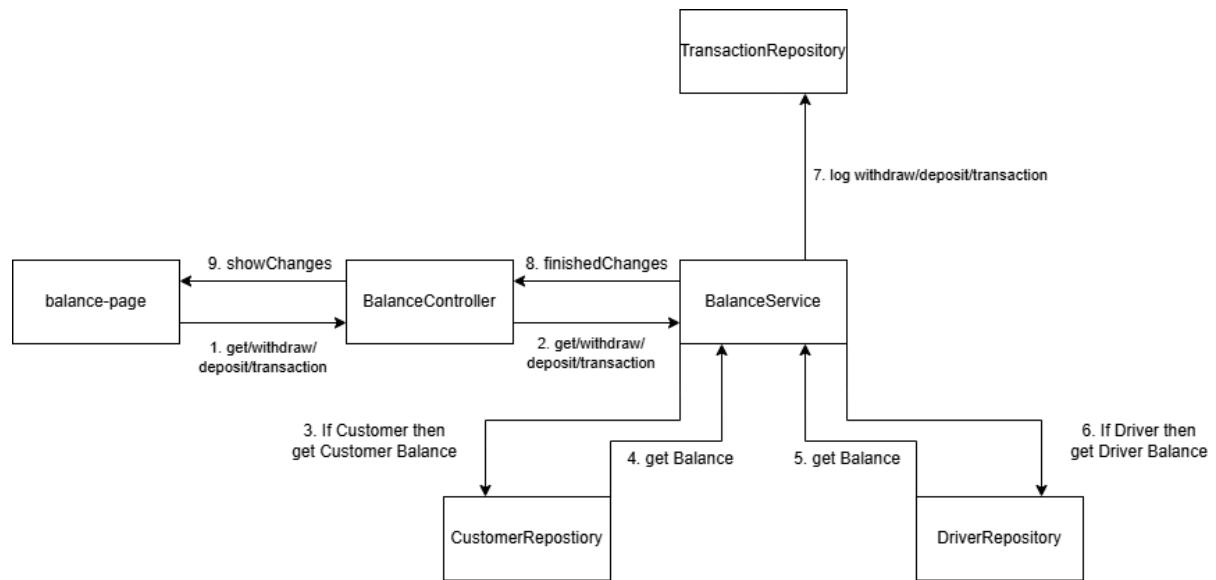
7.3 Kommunikationsdiagramm Fahrplanung:



7.3 Kommunikationsdiagramm Liste der verfügbaren Fahranfragen:



7.3 Kommunikationsdiagramm Geldkonten:



Funktionalitätsplanung

ID	Funktionalität	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcodereferenz	Status
1.	Geldkonten					
1.1	Aufladen des Geldkontos	Amin Rais	1.4	1.1	backend\src\main\java\com\sep\backend\account\balance TransactionService.java TransactionController	Fertig
1.2	Berechnen des Geldkontos	Amin Rais	1.1,1.3	1.2	backend\src\main\java\com\sep\backend\account\balance TransactionService.java TransactionController	Fertig
1.3	Abheben des Geldkontos	Amin Rais	1.4	1.3	backend\src\main\java\com\sep\backend\account\balance TransactionService.java TransactionController	Fertig
1.4	Darstellung des Geldkontos	Amin Rais	1.1, 1.3	1.1	frontend\src\app\components\balance Balance.html Balance.ts balance.css	Fertig
2.	Liste der verfügbaren Fahranfragen					
2.1	Tabelle der verfügbaren Fahranfragen	Reine Christiane Demgne	2.2		frontend/src/app/components/available-triprequest/available-triprequest.component.ts frontend/src/app/components/available-triprequest/available-triprequest.component.html (außer ngOnDestroy(),getCurrentActiveTripOffer(),revokeTripOffer(),createTripOffer(),und notification service)	Fertig
2.2	Input Feld für der Fahrers Startadresse	Reine Christiane Demgne	2.2		frontend/src/app/components/available-triprequest/available-triprequest.component.ts available-triprequest.component.html	Fertig

2.3	Sortierung nach jeder Spalte	Reine Christiane Demgne	2.3		frontend/src/app/components/available-triprequest/available-triprequest.component.ts available-triprequest.component.html	Fertig
2.4	Liste der verfügbaren Fahranfragen und deren Schnittstelle erstellen	Moyo Fred Aymerick			backend/src/main/java/com/sep/backend/trip/request/TripRequestService.java backend/src/main/java/com/sep/backend/trip/request/TripRequestController.java	Fertig
3.	Fahrtplanung					
3.1	ORS GeoJSON	Thomas Fabio Kaiser	3.2	3.1, 3.2	backend/src/main/java/com/sep/backend/ors/data.*	Fertig
3.2	Route bestimmen und an Frontend schicken	Thomas Fabio Kaiser	3.1	3.1	backend/src/main/java/com/sep/backend/ors/ORSController backend/src/main/java/com/sep/backend/ors/ORSService backend/src/main/java/com/sep/backend/ors/RouteRequestBody backend/src/main/java/com/sep/backend/ors/data/ORSFeatureCollection	Fertig
3.3	Route Entity speichern	Thomas Fabio Kaiser	3.1, 3.2	3.1	backend/src/main/java/com/sep/backend/entity/RouteEntity backend/src/main/java/com/sep/backend/route/RouteRepository backend/src/main/java/com/sep/backend/route/RouteService backend/src/main/java/com/sep/backend/trip/request/TripRequestService	Fertig
3.4	Fahrfrage Erstellen im Frontend	Thomas Fabio Kaiser	3.2	3.1, 3.2	frontend/src/app/components/trip-request/trip-request.component.ts frontend/src/app/components/trip-request/trip-request.component.html frontend/src/app/components/trip-request/trip-request.component.css backend/src/.../nominatim/NominatimController backend/src/.../trip/request/TripRequestDTO	Fertig

3.5	Aktive Fahranfrage im Frontend	Thomas Fabio Kaiser	3.2	3.1.1, 3.2	<p>frontend/src/app/components/trip-request/trip-request.component</p> <p>frontend/src/app/components/trip-request/trip-request.component.html</p> <p>frontend/src/app/components/trip-request/trip-request.component.css</p> <p>backend/src/main/java/com/sep/backend/trip/request/TripRequest DTO</p>	Fertig
4.1	Fahrtangebote (Backend)				<p>Pfad:</p> <p>backend/src/main/java/com/sep/backend/trip/offer</p> <p>Für alle:</p> <p>TripOfferController</p> <p>TripOfferService</p> <p>TripOfferRepository</p> <p>TripOfferStatus</p>	
4.1.1	Angebote an Kunden bereitstellen	Ian Christopher Schirmer			TripOffer TripOfferService.getTripOfferList	Fertig
4.1.2	Anfrage hinzufügen	Ian Christopher Schirmer		4.3	TripOfferService.createNewTripOffer	Fertig
4.1.3	Anfrage akzeptieren/ ablehnen	Ian Christopher Schirmer		4.5	TripOfferService.acceptTripOffer TripOfferService.rejectTripOffer	Fertig
4.1.4	Anfrage zurückziehen	Ian Christopher Schirmer		4.4	TripOfferService.revokeTripOffer	Fertig
4.2	Fahrtangebote (Frontend)					
4.2.1	button Fahrt anbieten oder zurückziehen Fahrer	Daoudi Mohammed		4.3, 4.4	<p>frontend/src/app/components/available-triprequest/available-triprequest.component.ts</p> <p>frontend/src/app/components/available-triprequest/available-triprequest.component.html</p> <p>(außer onSave(),currentLocation(),ngAfterViewInit(),triprequestService)</p>	Fertig
4.2.2	Button deaktivieren, wenn Fahrer bereits angeboten hat	Daoudi Mohammed			<p>frontend/src/app/components/available-triprequest/available-triprequest.component.ts</p> <p>frontend/src/app/components/available-triprequest/available-triprequest.component.html</p>	Fertig

					(außer onSave(),currentLocation(),ngAfte rViewInit(),triprequestService)	
4.2.3	Angebotsübersicht Kunde	Daoudi Mohammed			frontend/src/app/components/trip-offers/trip-offers.component.ts frontend/src/app/components/trip-offers/trip-offers.component.html frontend/src/app/components/trip-offers/trip-offers.component.css	Fertig
4.2.4	Kunde Fahrangebot annehmen oder ablehnen	Daoudi Mohammed	4.5		frontend/src/app/components/trip-offers/trip-offers.component.ts frontend/src/app/components/trip-offers/trip-offers.component.html frontend/src/app/components/trip-offers/trip-offers.component.css	Fertig
4.2.5	Live benachrichtigung bei neuem Fahrangebot	Daoudi Mohammed			frontend/src/app/components/notification/notification.component.ts frontend/src/app/components/notification/notification.component.html	Fertig
5.	Durchführung einer Fahrt					
5.1	Benachrichtigungen senden	Mats Paul Tončik	5.2, 5.4	4.2.5	backend/src/main/java/com/sep/backend/notification/Notification.java backend/src/main/java/com/sep/backend/notification/NotificationController.java backend/src/main/java/com/sep/backend/notification/NotificationRepository.java backend/src/main/java/com/sep/backend/notification/NotificationService.java	Fertig

					backend/src/main/java/com/sep/backend/notification/NotificationType.java backend/src/main/java/com/sep/backend/notification/NotificationTypes.java backend/src/main/java/com/sep/backend/validators/NotificationTypeValidator.java	
5.2	Benachrichtigungen erhalten	Mats Paul Tončik	4.2.5, 5.1, 5.4		frontend/src/app/services/angular-notification.service.ts frontend/src/app/services/stomp.service.ts	Fertig
5.3	Datenbankmodelle	Mats Paul Tončik		5.4, 5.6, 5.7	backend/src/main/java/com/sep/backend/entity/LocationEntity.java backend/src/main/java/com/sep/backend/entity/NotificationEntity.java backend/src/main/java/com/sep/backend/entity/RouteEntity.java backend/src/main/java/com/sep/backend/entity/TransactionEntity.java backend/src/main/java/com/sep/backend/entity/TripHistoryEntity.java backend/src/main/java/com/sep/backend/entity/TripOfferEntity.java backend/src/main/java/com/sep/backend/entity/TripRequestEntity.java	Fertig
5.4	WebSocket Config	Mats Paul Tončik	5.1, 5.5, 5.6		backend/src/main/java/com/sep/backend/AuthChannelInterceptor.java backend/src/main/java/com/sep/backend/WebSocketAuthService.java backend/src/main/java/com/sep/backend/WebSocketConfig.java	Fertig
5.5	Serverseitige Simulation einer Fahrt	Mats Paul Tončik	5.6	5.*	backend/src/main/java/com/sep/backend/trip/simulation/data/SimulationAction.java backend/src/main/java/com/sep/backend/trip/simulation/data/SimulationActionParameters.java backend/src/main/java/com/sep/backend/trip/simulation/data/SimulationActionTypes.java	Fertig

					backend/src/main/java/com/sep/backend/trip/simulation/ActionStore.java backend/src/main/java/com/sep/backend/trip/simulation/TripSimulationController.java backend/src/main/java/com/sep/backend/trip/simulation/TripSimulationService.java backend/src/main/java/com/sep/backend/trip/simulation/TripSimulationStompController.java	
5.6	Clientseitige Simulation einer Fahrt	Mats Paul Tončik	5.5	5.*	frontend/src/app/components/trip-visualizer frontend/src/app/components/trip-simulation frontend/src/app/resolvers/trip-offer.resolver.ts frontend/src/app/components/trip-offer	Fertig
5.7	Pipes	Mats Paul Tončik			frontend/src/app/pipes/* frontend/src/app/pipes/car-type.pipe.ts frontend/src/app/pipes/euro.pipe.ts frontend/src/app/pipes/meter-to-km.pipe.ts frontend/src/app/pipes/seconds-to-time.pipe.ts frontend/src/app/pipes/trip-offer-status.pipe.ts	Fertig
6.	Fahrthistorie					
6.1	Input Feld für Suche nach Fahrername oder Kundenname	Reine Christiane Demgne	6.5	6.3	frontend/src/app/components/trip-history/trip-history.component.ts frontend/src/app/components/trip-history/trip-history.component.html	Fertig
6.2	Tabelle der abgeschlossenen Fahrten	Reine Christiane Demgne	6.4	6.1	frontend/src/app/components/trip-history/trip-history.component.ts frontend/src/app/components/trip-history/trip-history.component.html	Fertig
6.3	Sortierung nach jede Spalte	Reine Christiane Demgne	6.4	6.2	frontend/src/app/components/trip-history/trip-history.component.ts frontend/src/app/components/trip-history/trip-history.component.html	Fertig
6.4	Fahrthistorie erstellen und	Moyo Fred Aymerick			backend\src\main\java\com\sep\backend\trip\history\TripHistoryC	Fertig

	anzeigen lassen				ontroller.java , backend\src\main\java\com\sep\ backend\trip\history\TripHistoryD TO.java backend\src\main\java\com\sep\ backend\trip\history\TripHistoryR epository.java backend\src\main\java\com\sep\ backend\trip\history\TripHistoryS ervice.java	
--	--------------------	--	--	--	---	--

Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status
1	Abheben vom Geldkonto	backend\src\test\java\com\sep\backend\BalanceServiceTest.java	Fertig
2	Überweisen des Fahrpreises	backend\src\test\java\com\sep\backend\BalanceServiceTest.java	Fertig
3	Kleines/Mittlere und Deluxe Auto valide Distanz	backend/src/test/java/com/sep/backend/TripRequestTest.java -> testSmallCarValidDistance(), testMediumCarValidDistance(), testDeluxeCarValidDistance()	Fertig
4	Invalide Autoklasse	backend/src/test/java/com/sep/backend/TripRequestTest.java -> testInvalidCarType()	Fertig
5	Negative Distanz + 0 Distanz	backend/src/test/java/com/sep/backend/TripRequestTest.java -> testNegativeDistance(), testZeroDistance()	Fertig
6	Valide Fahrpreisberechnung mit kleinen, mittleren und deluxe Auto	backend/src/test/java/com/sep/backend/TripRequestTest.java -> testSmallCarRoundUp(), testMediumCarEdgeCase(), testDeluxeCarLongDistance()	Fertig
7	Liste der verfügbaren Fahrfrage anzeigen	backend/src/test/java/com/sep/backend/tripServiceIntegrationTest	Fertig
8	Trip Historie anzeigen	backend/src/test/java/com/sep/backend/tripServiceIntegrationTest	Fertig

Systemtests

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Kunde gibt Adressen in die Suchleiste ein. Vertauscht die Reihenfolge der Adressen.	Route wird angezeigt auf der Karte mit Start, Ende und Zwischenstopps. Distanz, Dauer und Preis werden angezeigt bei richtiger Route. Route wird live geupdated bei neuen Adressen oder Reihenfolge.	✓
2	Kunde klickt auf Fahranfrage erstellen	Fahranfrage wird erfolgreich erstellt und wird zur aktiven Fahranfrage weitergeleitet. Bei aktiver Fahranfrage ist die Route immer noch sichtbar mit den Stopps.	✓
3	Kunde klickt auf Fahranfrage löschen	Fahranfrage wird gelöscht und Kunde wird zurück zur Fahranfragen Erstellung weitergeleitet.	✓
Nachbedingung(en)	Fahranfrage existiert nicht.		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Kunde gibt Adressen in die Suchleiste ein	Route wird angezeigt auf der Karte mit Start, Ende und Zwischenstopps. Distanz, Dauer und Preis wird angezeigt bei richtiger Route	✓
2	Kunde klickt auf Fahranfrage erstellen	Fahranfrage wird erfolgreich erstellt und wird zur aktiven Fahranfrage weitergeleitet. Bei aktiver Fahranfrage ist die Route immer noch sichtbar mit den Stopps.	✓
Nachbedingung(en)	Fahranfrage existiert.		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Aleyna Aydin		
SW-Version	v2-RC4		
Vorbedingung(en)	Ein Kunde hat eine Fahranfrage erstellt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Fahrer wählt “Anfragen” in der Tabelle der Fahranfragen aus.	Ein Angebot wird abgegeben	✓
Nachbedingung(en)	Der Kunde kann das Angebot einsehen. Der Fahrer kann nur noch sein Angebot sehen.		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Aleyna Aydin		
SW-Version	v2-RC4		
Vorbedingung(en)	Ein Fahrer hat ein Angebot erstellt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde klickt auf “Angebote anzeigen”.	Der Kunde sieht die Angebote	✓
2	Der Kunde wählt X aus	Der Fahrer erhält eine Ablehnung.	✓
Nachbedingung(en)	Das Angebot ist abgelehnt.		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Aleyna Aydin		
SW-Version	v2-RC4		
Vorbedingung(en)	Ein Fahrer hat ein Angebot erstellt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde klickt auf "Angebote anzeigen".	Der Kunde sieht die Angebote	✓
2	Der Kunde wählt ✓ aus	Der Fahrer erhält eine Information, dass das Angebot akzeptiert wurde, und kann mit der Simulation starten.	✓
Nachbedingung(en)	Das Angebot ist akzeptiert.		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Aleyna Aydin		
SW-Version	V2-RC4		
Vorbedingung(en)	Ein Fahrer hat ein Angebot erstellt.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Fahrer klickt bei seinem Angebot auf zurückziehen	Das Angebot wird zurückgezogen	✓
Nachbedingung(en)	Das Angebot ist zurückgezogen.		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System. Ein Fahrer existiert im System		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Kunde gibt Adressen in die Suchleiste ein. Vertauscht die Reihenfolge der Adressen.	Route wird angezeigt auf der Karte mit Start, Ende und Zwischenstopps. Distanz, Dauer und Preis werden angezeigt bei richtiger Route. Route wird live geupdated bei neuen Adressen oder Reihenfolge.	✓
2	Kunde klickt auf Fahranfrage erstellen	Fahranfrage wird erfolgreich erstellt und wird zur aktiven Fahranfrage weitergeleitet. Bei aktiver Fahranfrage ist die Route immer noch sichtbar mit den Stopps.	✓
3	Fahranfrage wird durchgeführt	Animation wird durchlaufen, bis zum Zielort	✓
4	Fahranfrage wurde abgeschlossen	Geld vom Konto des Kunden wird abgehoben und zum Konto des Fahrers hinzugefügt	✓
Nachbedingung(en)	Fahranfrage existiert nicht, Konten besitzen aktualisierte Beträge		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde öffnet über das Profil der Navigationsbar sein Konto	Die Kunde wird zu einer neuen Seite geführt in welcher er sein Konto sehen kann	✓
2	der Kunde gibt einen Betrag ein und drückt abheben	der Betrag wird abgehoben wenn genug geld vorhanden ist, sonst gibt es eine fehlermeldung. Die Anzeige des Kontostandes ändert sich auf den aktuellen Wert Die Transaktionshistorie aktualisiert sich um die Transaktion	✓
Nachbedingung(en)	zahl verschwindet nach betätigen des Knopfes aus der Bar, Kontostand aktualisiert, transaktionshistory aktualisiert sich		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde öffnet über das Profil der Navigationsbar sein Konto	Die Kunde wird zu einer neuen Seite geführt in welcher er sein Konto sehen kann	✓
2	der Kunde gibt einen Betrag ein und drückt einzahlen	der Betrag wird eingezahlt und die Anzeige des Kontostandes wird aktualisiert Die Transaktionshistory aktualisiert sich um die Transaktion	✓
Nachbedingung(en)	Zahl verschwindet nach betätigen des Knopfes aus der Bar,Kontostand aktualisiert, transaktionshistory aktualisiert sich		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System und ist schon angemeldet . Ein Fahrer existiert im System und Fahranfrage wurde schon durchgeführt .		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	der kunde klickt auf die Fahrthistorie-Button	er wird zu der Seite der Liste der Verfügbaren Fahrtanfrage geführt .	<input checked="" type="checkbox"/>
2	der kunde klickt auf der SpaltenName.	die Tabelle musst nach der angeklickten SpaltenName sortiert wird .	<input checked="" type="checkbox"/>
Nachbedingung(en)	Die abgeschlossenen Fahrten des eingeloggten Benutzers (Kunde oder Fahrer) werden vollständig angezeigt.		
Testurteil	Bestanden		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System und ist schon angemeldet . Ein Fahrer existiert im System und Fahranfrage wurde schon durchgeführt .		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	der driver klickt auf die Fahrtanfrage-Button	driver wird in einer neuen Seite nach seiner Adresse gefragt	<input checked="" type="checkbox"/>
2	der driver gibt seine Adresse ein	driver bekommt dann die Tabelle der Liste der Verfügbaren Fahrt Anfrage	<input checked="" type="checkbox"/>
Nachbedingung(en)	Die derzeitigen Fahrt Anfrage werden vollständig angezeigt.		
Testurteil	Bestanden		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde öffnet über das Profil der Navigationsbar sein Konto	Die Kunde wird zu einer neuen Seite geführt in welcher er sein Konto sehen kann	✓
2	der Kunde gibt einen Betrag ein und gibt eine ungültige Zahl mit zu vielen Nachkommastellen an (Bsp. 12.123456) und versucht es einzuzahlen	Der Betrag wird auf die ersten beiden Nachkommastellen begrenzt. Der eingezahlte Betrag wäre 12.12. Die Transaktionshistory aktualisiert sich um die Transaktion	✓
Nachbedingung(en)	zahl verschwindet nach betätigen des Knopfes aus der Bar, Kontostand aktualisiert, transaktionshistory aktualisiert sich		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde öffnet über das Profil der Navigationsbar sein Konto	Die Kunde wird zu einer neuen Seite geführt in welcher er sein Konto sehen kann	✓
2	der Kunde gibt einen Betrag ein und benutzt ein punkt statt einem komma und drückt auf einzahlen	System akzeptiert beide Eingaben, mit punkt oder komma und aktualisiert den betrag dementsprechend Die Transaktionshistory aktualisiert sich um die Transaktion	✓
Nachbedingung(en)	zahl verschwindet nach bestätigen des Knopfes aus der Bar, Kontostand aktualisiert, transaktionshistory aktualisiert sich		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Markus Tohn		
SW-Version	v2-RC4		
Vorbedingung(en)	Der Kunde existiert im System.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde öffnet über das Profil der Navigationsbar sein Konto	Die Kunde wird zu einer neuen Seite geführt in welcher er sein Konto sehen kann	✓
2	der Kunde gibt einen Betrag ein und drückt abheben	der Betrag wird abgehoben wenn genug geld vorhanden ist, sonst gibt es eine fehlermeldung. Die Anzeige des Kontostandes ändert sich auf den aktuellen Wert. Die Transaktionshistory aktualisiert sich um die Transaktion	✓
Nachbedingung(en)	zahl verschwindet nach betätigen des Knopfes aus der Bar, Kontostand aktualisiert, transaktionshistory aktualisiert sich		
Testurteil	Test bestanden.		

Datum	17.06.2025		
Tester	Daniel Golly		
SW-Version	v2-RC4		
Vorbedingung(en)	Es existiert ein angenommenes Fahrangebot zwischen Kunde und Fahrer. Beide Simulations-Teilnehmer haben zur Simulation navigiert.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Kunde klickt auf das Play-Icon.	Die Simulation startet.	✓
2	Der Fahrer klickt auf das Pause-Icon.	Die Simulation stoppt.	✓
3	Der Kunde verschiebt den Slider nach rechts.	Die Geschwindigkeit wird verringert.	✓
4	Der Kunde klickt auf das Play-Icon.	Die Simulation geht weiter.	✓
5		Die Simulation endet und ein Bewertungs-Dialog öffnet sich.	✓
6	Der Kunde klickt auf den vierten Stern von links und anschließend auf Bewerten.	Die Bewertung des Kunden wird für die Fahrhistorie gespeichert.	✓
7	Der Fahrer klickt auf den dritten Stern von links und anschließend auf Bewerten.	Die Bewertung des Fahrers wird für die Fahrhistorie gespeichert.	✓
8		Kunde und Fahrer werden zurück auf die Startseite navigiert.	✓
Nachbedingung(en)	Die Simulation gilt als abgeschlossen und wird in der Fahrhistorie angezeigt.		
Testurteil	Test bestanden.		

Zyklus III

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Fahrer-Leaderboard			
1.1	Fahrer-Leaderboard Anzeigen	User Story	Moyo Fred Aymerick	Fertig
1.2	Tabelle der Fahrer-Leaderboard	Papierprototyp	Reine Christiane Demgne	Fertig
2.	Grafische Statistiken			
2.1	Grafische Statistiken anzeigen	User Story	Ian Christopher Schirmer	Fertig
2.2	Grafische Statistiken	Papierprototyp	Amin Rais	Fertig
2.3	Darstellung der Statistiken	User Story	Amin Rais	Fertig
3.	Live-Änderung			
3.1	Neue Ziele hinzufügen	User-Story	Thomas Fabio Kaiser	Fertig
3.2	Ziele löschen	User-Story	Thomas Fabio Kaiser	Fertig
3.3	Ziele bewegen	User-Story	Thomas Fabio Kaiser	Fertig
3.4	Live-Änderungen bei Simulation anbieten	Papierprototyp	Thomas Fabio Kaiser	Fertig
4.	Chat			
4.1	Kunde und Fahrer können chatten, wenn ein Angebot existiert	Papierprototyp	Mohammed Daoudi	fertig
4.2	Die Nachrichten werden in Echtzeit gesendet und empfangen	Papierprototyp	Mohammed Daoudi	fertig
4.3	Nachrichten können gelöscht oder bearbeitet werden, solange der Empfänger sie noch nicht gelesen hat	Papierprototyp	Mohammed Daoudi	fertig
4.4	Nachricht senden	User Story	Mats Paul Tončik	Fertig
4.5	Nachricht löschen	User Story	Mats Paul Tončik	Fertig
4.6	Nachricht bearbeiten	User Story	Mats Paul Tončik	Fertig
4.7	Nachricht empfangen	User Story	Mats Paul Tončik	Fertig
5.	Architektur			
5.1	Klassendiagramm		Moyo Fred Aymerick	
5.2	Komponentendiagramm		Moyo Fred Aymerick	
5.3	Kommunikationsdiagramm		Thomas Fabio Kaiser	In Bearbeitung

User-Stories

User Story-ID	1.1
User Story-Beschreibung	Als Benutzer möchte ich ein Leaderboard sehen, das alle registrierten Fahrer zeigt, damit ich vergleichen kann, welche Fahrer besonders aktiv oder gut bewertet sind.
Geschätzter Realisierungsaufwand	1 Tag
Priorität	Niedrig
Autor	Moyo Fred Aymerick
Abhängigkeiten zu anderen User Stories	1.1,1.2,5.1,5.2

User Story-ID	2.1
User Story-Beschreibung	Als Fahrer möchte ich, dass meine Einnahmen, gefahrene Distanz, Fahrzeit und durchschnittliche Bewertung als Diagramme angezeigt werden.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Mittel
Autor	Ian Christopher Schirmer
Abhängigkeiten zu anderen User Stories	

User Story-ID	2.3
User Story-Beschreibung	Als Nutzer möchte ich die Informationen Einnahmen, gefahrene Distanz, Fahrzeit und durchschnittliche Bewertung in Form eines Diagrammes angezeigt bekommen
Geschätzter Realisierungsaufwand	2 Tage
Priorität	mittel
Autor	Amin Rais
Abhängigkeiten zu anderen User Stories	2.1

User Story-ID	3.1
User Story-Beschreibung	Als Kunde möchte ich, dass ich während der Fahrt neue Ziele hinzufügen kann, um spontan meine Fahrt zu ändern.
Geschätzter Realisierungsaufwand	2 Tage
Priorität	Hoch
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	5.1, 5.2, 5.3, 5.4, 5.5

User Story-ID	3.2
User Story-Beschreibung	Als Kunde möchte ich, dass ich während der Fahrt Ziele löschen kann, die ich nicht mehr befahren will.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	5.1, 5.2, 5.3, 5.4, 5.5

User Story-ID	3.3
User Story-Beschreibung	Als Kunde möchte ich, dass ich während der Fahrt die Reihenfolge meiner Ziele ändern kann, um spontan andere Ziele früher zu erreichen
Geschätzter Realisierungsaufwand	2 Tage
Priorität	Mittel
Autor	Thomas Fabio Kaiser
Abhängigkeiten zu anderen User Stories	5.1, 5.2, 5.3, 5.4, 5.5

User Story-ID	4.3
User Story-Beschreibung	Als Benutzer möchte ich meinem Fahrt-Partner eine Nachricht senden, damit wir uns über die angebotene Fahrt austauschen können.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	

User Story-ID	4.4
User Story-Beschreibung	Als Benutzer möchte ich meine Nachricht bearbeiten, wenn ich mich verschrieben haben, damit mein Gegenüber die richtigen Informationen erhält
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	4.3

User Story-ID	4.5
User Story-Beschreibung	Als Benutzer möchte ich meine Nachricht löschen, so dass mein Gegenüber nicht sieht, was ich eigentlich geschrieben habe.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch

Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	4.3

User Story-ID	4.6
User Story-Beschreibung	Als Benutzer möchte ich eine Nachricht von meine Gegenüber empfangen können, dass ich weiß was er mir mitteilen möchte.
Geschätzter Realisierungsaufwand	3 Tage
Priorität	Hoch
Autor	Mats Paul Tončik
Abhängigkeiten zu anderen User Stories	4.3

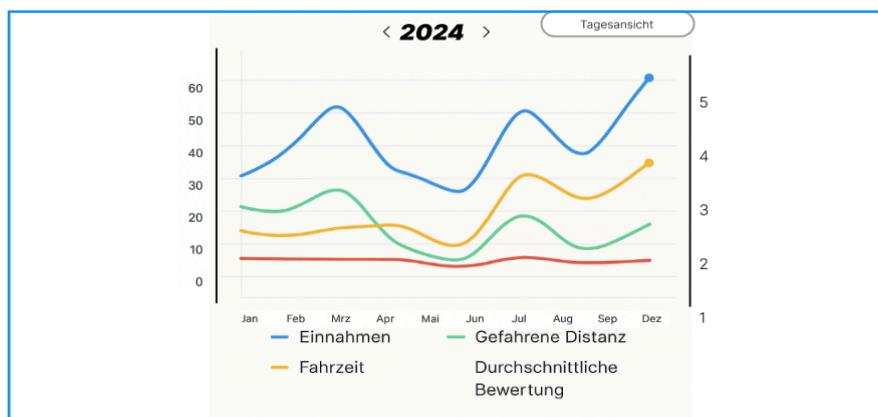
Papierprototypen.

1.2

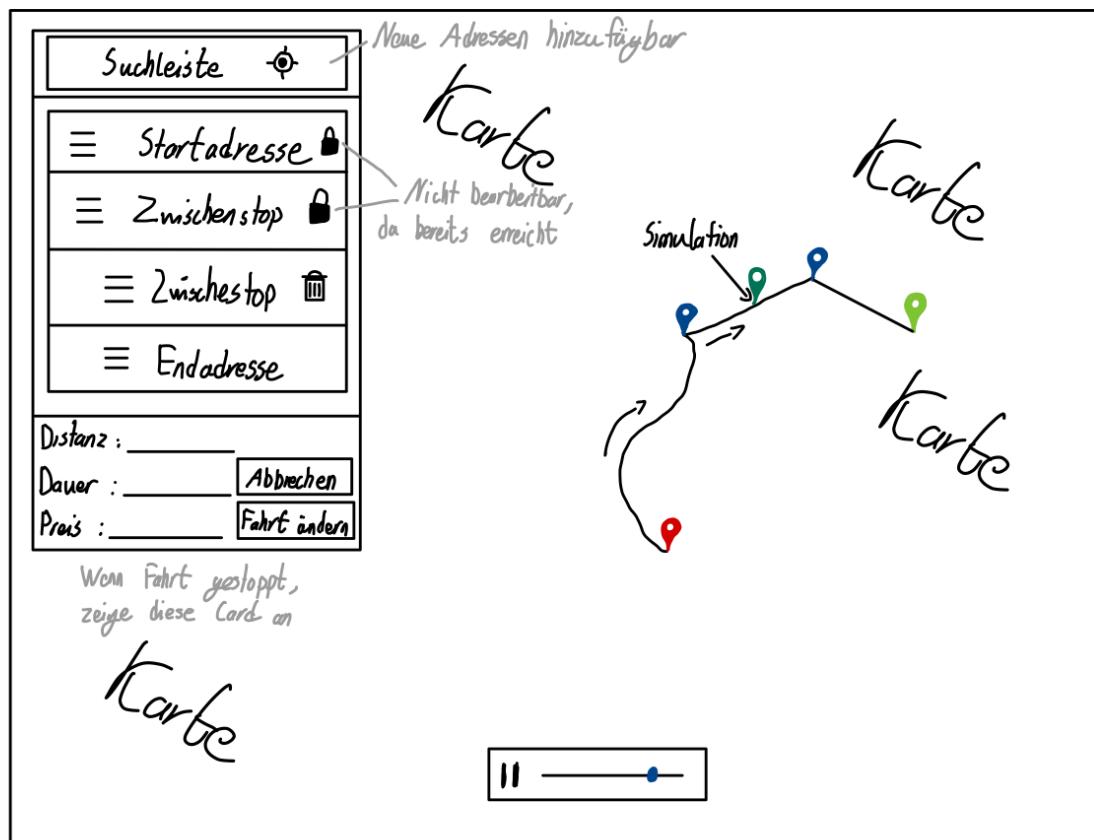
Fahrergr. Leaderboard						
Fahrer-Benutzername ↑	Fahrername ↑	Gesamtge-fahrene Distanz ↑	Durchschnitt-lische Bewertung ↑	Gesamte Fahrzeit ↑	durchgeführ-t ten Fahrten ↑	Verdientes Geld ↑
Ali20	Ali Leo	20 Km	★★★★★	100 min	3	150 €
...
..

2.2

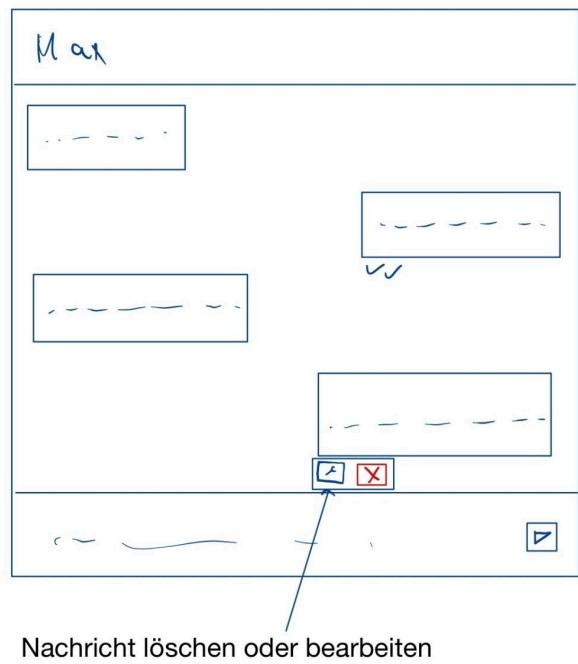
Konto:	134.00 €
Gesamtanzahl der Fahrten:	0



Papierprototyp 3.2

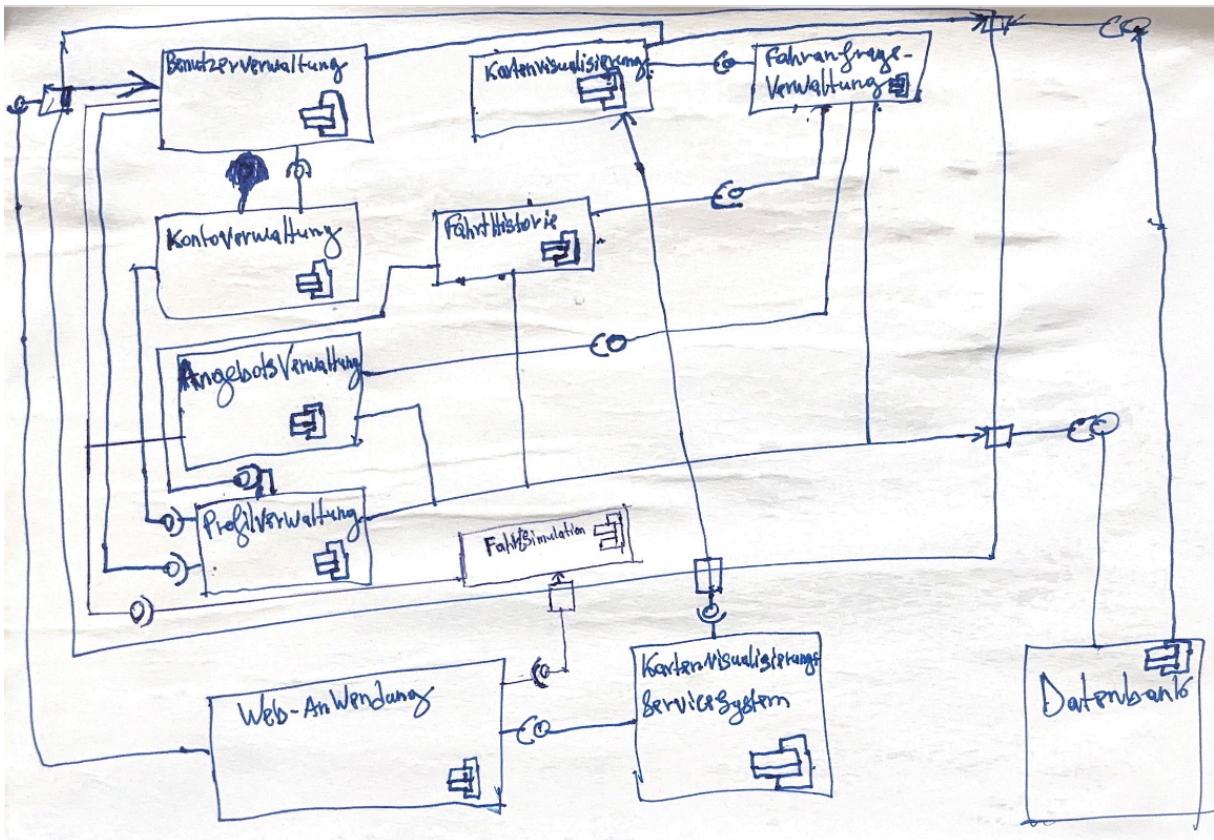


Papierprototyp 4.1,4.2 und 4.3



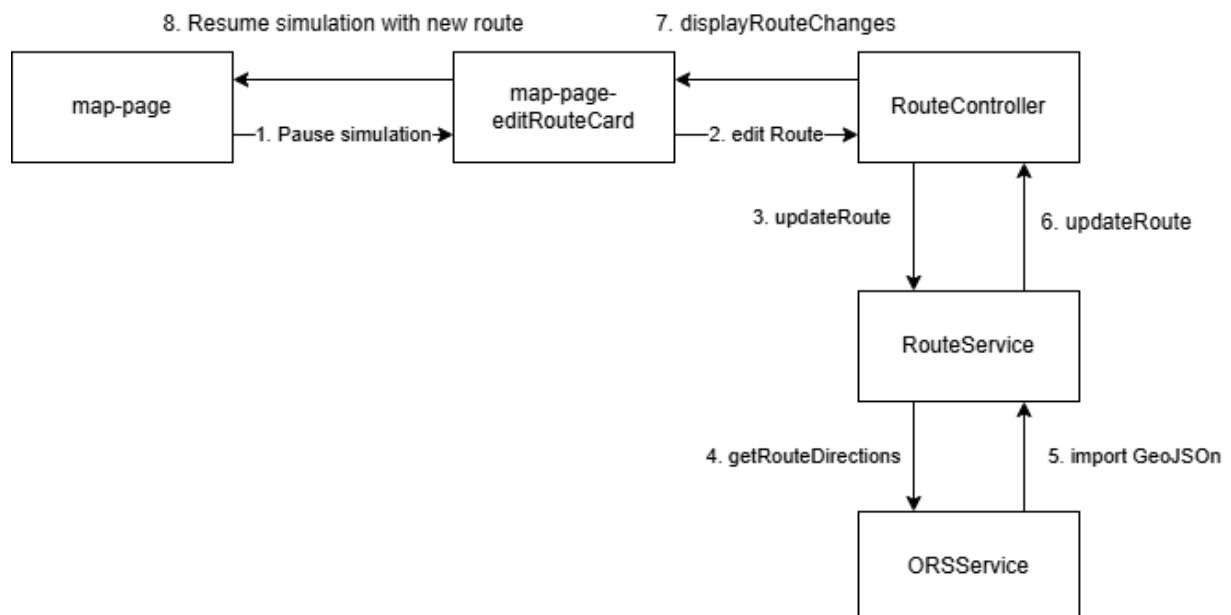
Strukturdiagramme (Komponenten- und Klassendiagramme)

Komponentendiagramm

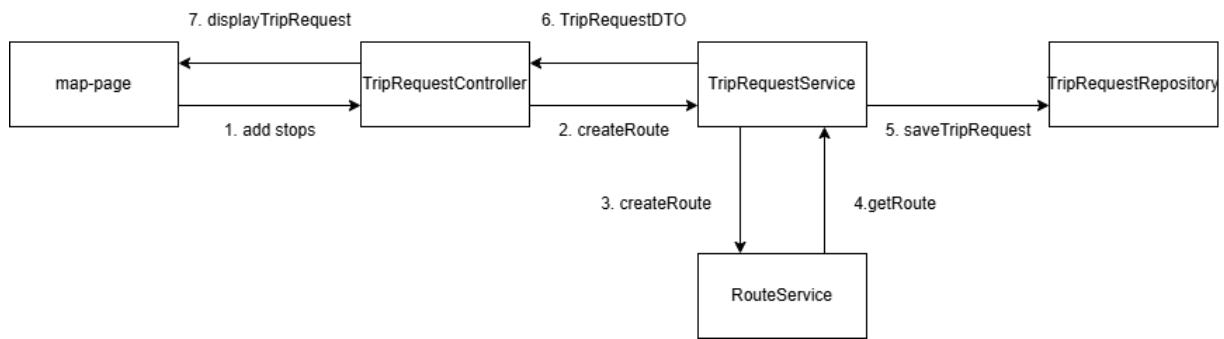


Verhaltensdiagramme (Kommunikationsdiagramme)

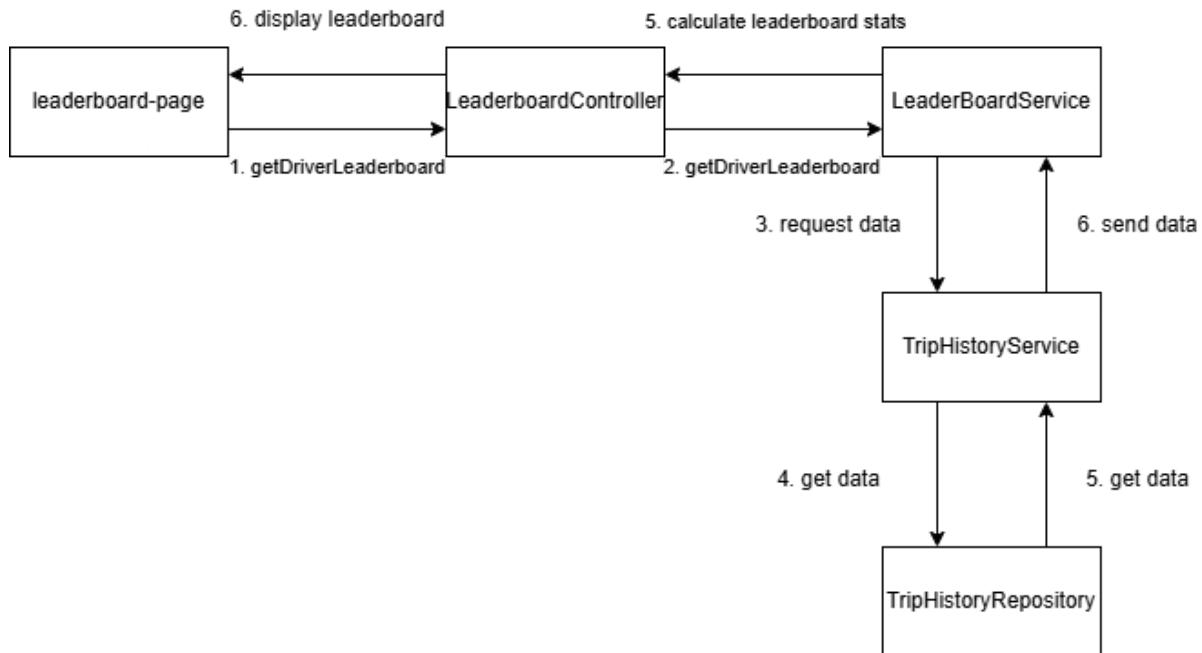
5.3 Kommunikationsdiagramm Live Änderung während der Fahrt



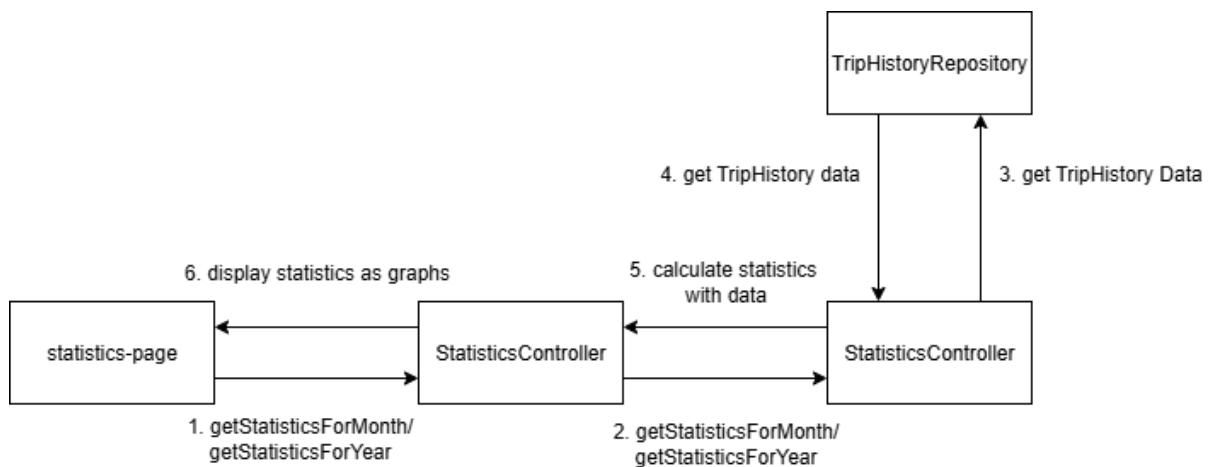
5.3 Kommunikationsdiagramm Zwischenstopps



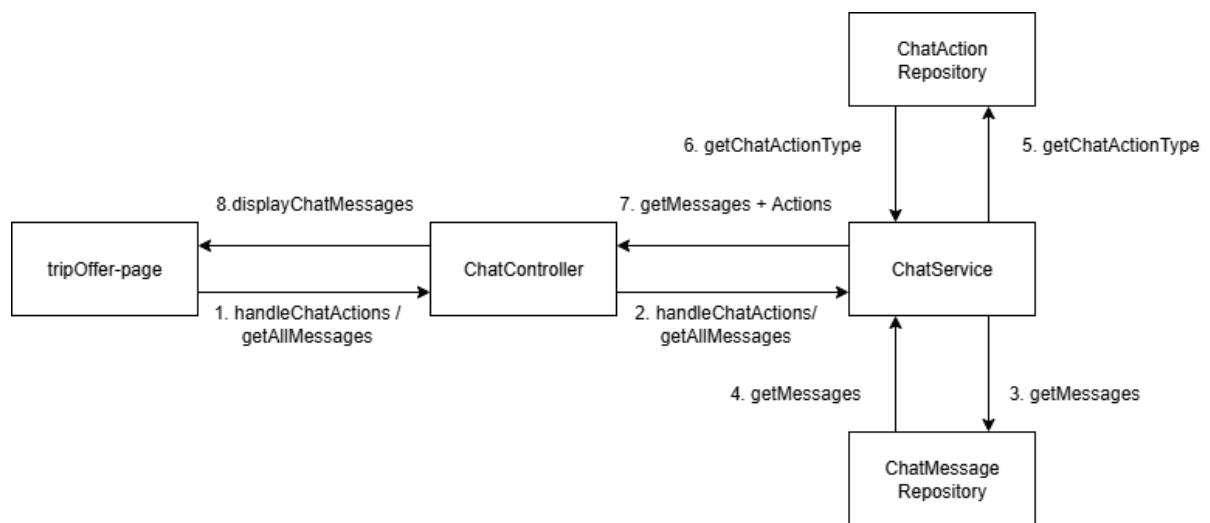
5.3 Kommunikationsdiagramm Leaderboard



5.3 Kommunikationsdiagramm Statistiken



5.3 Kommunikationsdiagramm Chat



Funktionalitätsplanung

ID	Funktionalität	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
1	Fahrer-Leaderboard					
1.1	FahrerLeadboard erstellen	Moyo Fred Aymerick	3,5,6	1.1	backend/src/main/java/com/sep/backend/account/Leaderboard/Leaderboard.java backend/src/main/java/com/sep/backend/account/Leaderboard/LeaderboardController.java backend/src/main/java/com/sep/backend/account/Leaderboard/LeaderboardService.java	Fertig
1.2	Leaderboard mit enthaltenen Daten anzeigen	Reine Christiane Demgne	1.2.2	1.1	frontend/src/app/components/leaderboard/leaderboard.component.ts frontend/src/app/components/leaderboard/leaderboard.component.html frontend/src/app/components/leaderboard/leaderboard.component.css	
1.3	Eingabefeld fürs Filtern nach Fahrername	Reine Christiane Demgne	1.2.1	1.1	frontend/src/app/components/leaderboard/leaderboard.component.ts frontend/src/app/components/leaderboard/leaderboard.component.html frontend/src/app/components/leaderboard/leaderboard.component.css	
1.4	Sortierung nach jeder Spalte	Reine Christiane Demgne	1.2.1, 1.2.2	1.1	frontend/src/app/components/leaderboard/leaderboard.component.ts frontend/src/app/components/leaderboard/leaderboard.component.html frontend/src/app/components/leaderboard/leaderboard.component.css	

2	Grafische Statistiken					
2.1	Jahresstatistiken an Frontend übergeben	Ian Christopher Schirmer		2.1	<p>backend/src/main/java/com/sep/backend/statistics/StatisticsController</p> <p>backend/src/main/java/com/sep/backend/statistics/StatisticsService</p> <p>backend/src/main/java/com/sep/backend/statistics/StatisticsType</p> <p>backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getAvgRatingStatisticsByDriver</p> <p>backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getSumDistanceStatisticsByDriver</p> <p>backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getSumTimeStatisticsByDriver</p> <p>backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getSumRevenueStatisticsByDriver</p>	Fertig
2.2	Monatsstatistiken an Frontend übergeben	Ian Christopher Schirmer		2.1	<p>backend/src/main/java/com/sep/backend/statistics/StatisticsController</p> <p>backend/src/main/java/com/sep/backend/statistics/StatisticsService</p> <p>backend/src/main/java/com/sep/backend/statistics/StatisticsType</p> <p>backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getAvgRatingStatisticsByDriver</p> <p>backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getSumDistanceStatisticsByDriver</p>	Fertig

					repository.getSumDistanceStatisticsByDriver backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getSumTimeStatisticsByDriver backend/src/main/java/com/sep/backend/trip/history/TripHistoryRepository.getSumRevenueStatisticsByDriver	
2.3	Frontend Darstellung	Amin Rais	2.1 , 2.2	2.1, 2.3	frontend/src/app/components/stats/ stats.component.ts frontend/src/app/components/stats/stats.component.html	fertig
3	Live-Änderungen					
3.1	Neue Route auswählen	Thomas Fabio Kaiser		3.1, 3.2, 3.3	frontend/src/app/components/trip-simulation/ trip-simulation.component.ts -> (updateLocalRoute, onCancel, reroute_unlock/reroute_lock, + weitere Methoden importiert von trip-request-component) frontend/src/app/components/trip-simulation/trip-simulation.component.html frontend/src/app/components/trip-simulation/trip-simulation.component.css	Fertig
3.2	Neue Route anzeigen mit aktueller Position	Thomas Fabio Kaiser		3.1, 3.2, 3.3	frontend/src/app/components/trip-simulation/ trip-simulation.component.ts -> (updateLocalRoute, onCancel, reroute_unlock/reroute_lock, + weitere Methoden importiert von trip-request-component) frontend/src/app/components/trip-simulation/trip-simulation.component.html frontend/src/app/components/trip-simulation/trip-simulation.component.css	Fertig
3.3	Neue Route erstellen	Thomas Fabio Kaiser		3.1, 3.2, 3.3	frontend/src/app/components/trip-simulation/ trip-simulation.component.ts	Fertig

					<p>onent.ts -> (updateRoute, findClosestAnimationIndex)</p> <p>frontend/src/app/components/trip-simulation/trip-simulation.component.html</p> <p>frontend/src/app/components/trip-simulation/trip-simulation.component.css</p> <p>backend/src/main/java/com/sep/backend/trip/simulation/data/*</p> <p>backend/src/main/java/com/sep/backend/route/RouteController.java</p> <p>backend/src/main/java/com/sep/backend/route/RouteService.java</p>	
3.4	Neue Route anzeigen beim Fahrer	Thomas Fabio Kaiser		3.1, 3.2, 3.3	<p>frontend/src/app/components/trip-simulation/trip-simulation.component.ts -> (updateRoute, handleSimulationAction)</p> <p>frontend/src/app/components/trip-simulation/trip-simulation.component.html</p> <p>frontend/src/app/components/trip-simulation/trip-simulation.component.css</p> <p>frontend/src/app/components/trip-visualizer/trip-visualizer.component.ts</p>	In Bearbeitung
3.5	Simulation fortfahren von aktueller Position mit neuer Route	Thomas Fabio Kaiser		3.1, 3.2, 3.3	<p>frontend/src/app/components/trip-simulation/trip-simulation.component.ts -> (updateRoute, handleSimulationAction)</p> <p>frontend/src/app/components/trip-simulation/trip-simulation.component.html</p> <p>frontend/src/app/components/trip-simulation/trip-simulation.component.css</p> <p>frontend/src/app/components/trip-visualizer/trip-visualizer.component.ts</p>	In Bearbeitung

4	Chat					
4.1	Nachrichten senden, empfangen, bearbeiten und löschen – solange die Nachricht noch nicht vom Empfänger gesehen wurde.	Mohammed Daoudi	4.2	4.4, 4.5, 4.6 und 4.7	frontend/src/app/components/chat/chat.component.ts frontend/src/app/components/chat/chat.component.html frontend/src/app/components/chat/chat.component.css	Fertig
4.2	Nachrichten senden, empfangen, bearbeiten und löschen	Mats Paul Tončik	4.1	4.3, 4.4, 4.5 & 4.6	backend/src/main/java/com/sep/backend/trip/offer/chat/action/ChatActionRepository.java backend/src/main/java/com/sep/backend/trip/offer/chat/action/ChatActionType.java backend/src/main/java/com/sep/backend/trip/offer/chat/action/ChatActionTypes.java backend/src/main/java/com/sep/backend/trip/offer/chat/action/ChatTransactionService.java backend/src/main/java/com/sep/backend/trip/offer/chat/action/InboundChatAction.java backend/src/main/java/com/sep/backend/trip/offer/chat/message/ChatMessageDirections.java backend/src/main/java/com/sep/backend/trip/offer/chat/message/ChatMessageRepository.java backend/src/main/java/com/sep/backend/trip/offer/chat/message/OutboundChatMessage.java backend/src/main/java/com/sep/backend/trip/offer/chat/ChatController.java backend/src/main/java/com/sep/backend/trip/offer/chat/ChatException.java backend/src/main/java/com/sep/backend/trip/offer/chat/ChatService.java backend/src/main/java/com/sep/backend/trip/offer/chat/ChatSocketController.java	Fertig

Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status
1	Abrufen der Fahrer-Leaderboard	backend/src/test/java/com/sep/backend/TripRequestServiceIntegrationTest.java	Fertig

Systemtests

Datum			
Tester			
SW-Version			
Vorbedingung(en)			
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1			
2			
3			
Nachbedingung(en)			
Testurteil			

Nutzerhandbuch

Technische Anforderungen

Für die gewünschte Nutzung des Systems ist ein handelsüblicher Computer mit x86_64-Architektur, mindestens acht GB RAM, mindestens Docker Version 27, Windows 10 (oder höher) sowie Google Chrome als Webbrowser notwendig. Als andere Browser bieten sich außerdem Mozilla Firefox und Microsoft Edge an.

Installationsanleitung

Die .tar-Archive, die docker-compose.yaml, die von uns zur Verfügung gestellte Environment-Datei und der Ordner *nginx* mit der Konfiguration für den Webserver müssen sich zur Installation im selben Verzeichnis befinden. Zuerst müssen die als .tar vorhandenen Images durch folgende Eingaben in der Konsole, im selben Verzeichnis, geladen werden:

- docker load -i backend_image_pi_beta.tar
- docker load -i frontend_image_pi_beta.tar
- docker load -i postgres_14-alpine.tar

Anschließend lässt sich das System über die docker-compose.yaml durch folgende Eingabe im selben Verzeichnis starten:

- docker-compose up -d

Das System läuft nun vollständig und lässt sich wie gewünscht bedienen.

Bedienungsanleitung

Für alle Funktionen, die man beim entwickelten System verwenden will, muss man einen Account, entweder als Kunde oder Fahrer, besitzen. Der Aufruf der Seite erfolgt auf <http://localhost:80> bzw. <http://localhost>. In der Navigationsleiste hat man die Möglichkeit, sich zu registrieren oder sich anzumelden, erst dann ist das System nutzbar. Bei der Anmeldung bzw. Registrierung ist den auf dem Bildschirm sichtbaren Anweisungen Folge zu leisten. Der OTP-Code lässt sich mit dem Supercode der ersten sechs Fibonacci-Zahlen umgehen: **112358**. Manche Funktionen sind nur für die jeweilige Rolle verfügbar, andere Funktionen hingegen für jeden Benutzer.

Nach einer erfolgreichen Anmeldung sind in der Navigationsleiste, der Rolle entsprechend, Knöpfe vorhanden durch welche man mit einem Klick zur gewünschten Funktion kommt:

- Social: Hier hat man die Option, durch die auf der oberen Seite angeordnete Suchleiste nach einem Benutzernamen zu suchen. Bei erfolgreicher Suche werden unterhalb der Suchleiste Profilkarten der gefundenen Benutzer angezeigt. Mit einem Klick auf diese gelangt man auf die Profilseite des Nutzers. Sein eigenes Profil findet man auch über die Suche, ebenso ist dieses durch einen Klick auf seinen Namen in der oberen rechten Ecke und einem anschließenden Klick auf Profil zu erreichen.
- Konto: Sowohl Kunden als auch Fahrer können ihre Geldkonten einsehen, indem sie in der Navbar auf ihren Namen/ihr Profilbild klicken und anschließend auf "Konto" klicken. Auf der Kontoseite ist der momentane Kontostand zu sehen. Im Input kann der Kunde einen Betrag eingeben, welchen er einzahlen bzw. abheben will. Durch das Klicken auf die entsprechenden

Button wird die Aktion ausgeführt. Des Weiteren sieht der Benutzer seine vorherigen Transaktionen, dazu zählen Einzahlungen, Abhebungen und Überweisungen.

- **Fahranfrage:** Als Kunde kann man Fahranfragen stellen. Dafür muss der Kunde in der Navbar auf "Fahranfrage" klicken. Der Kunde hat die Möglichkeit, die Stops seiner Fahrt durch den Input in der linken oberen Ecke zu suchen. Anschließend kann er rechts daneben den gewünschten Stop auswählen. Ein Stop kann durch ein Klicken auf das Delete-Icon rechts daneben wieder entfernt werden. Die Reihenfolge der Stopps kann per Drag-and-Drop geändert werden. Durch die Radio-Buttons kann der Fahrer die gewünschte Fahrzeugklasse auswählen. Darunter werden ihm die Informationen zur Fahrt angezeigt. Durch ein Klick auf "Fahranfrage erstellen" wird die Fahranfrage erstellt.

Anschließend sieht der Kunde sein aktive Fahranfrage. Durch einen Klick auf "Löschen" kann er diese löschen. Durch einen Klick auf "Angebote anzeigen" kann man alle Fahrtangebote für seine aktive Fahranfrage sehen. Die Fahrtangebote werden in tabellarischer Form dargestellt und der Kunde kann diese nach Bewertung, gesamte Anzahl der Fahrten bzw. gesamte gefahrene Distanz der Fahrer sortieren. Durch ein Klick auf das entsprechende Icon wird das Fahrtangebot angenommen bzw. abgelehnt.

Nimmt der Kunde das Angebot an, wird die Möglichkeit des Löschen entfernt und der Kunde kann durch einen Klick auf "Simulieren" zur Simulation gelangen.

- **Fahranfragen:** Als Fahrer kann man sich die verfügbaren Fahranfragen anschauen. Dafür muss man in der Navbar auf "Fahrtanfragen" klicken. Der Fahrer muss zuerst seine momentane Position angeben. Durch einen Klick auf das Fadenkreuz werden die aktuellen Koordinaten ausgewählt. Beim Anklicken der unteren Hälfte des Eingabefelds kann man nun seine finale Startadresse auswählen. Nach dem Klicken auf "Suchen" werden die verfügbaren Fahranfragen angezeigt. Dem Fahrer werden alle verfügbaren Fahranfragen gezeigt und er bekommt die Möglichkeit durch das Klicken auf "Anfragen" dem Kunden für die entsprechende Anfrage ein Angebot zu stellen. Anschließend erhält der Kunde eine Benachrichtigung über ein neues Angebot. Der Fahrer sieht nun auf der selben Seite sein aktives Fahrtangebot. Solange das Angebot noch nicht vom Kunden akzeptiert wurde, kann er dieses durch ein Klicken auf "Zurückziehen" noch zurückziehen. Auch darüber würde der Kunde eine entsprechende Benachrichtigung erhalten.

Für eine Fahranfrage hat der Fahrer die Chance auf maximal ein Fahrtangebot. Sollte der Fahrer das Angebot zurückziehen bzw. der Kunde das Fahrtangebot ablehnen, hat der Fahrer keine Chance ein neues Angebot zu stellen. Sollte der Fahrer es trotzdem versuchen, wird er darüber durch die Snackbar informiert.

Wenn der Kunde das Fahrtangebot akzeptiert, erhält der Fahrer eine Benachrichtigung. In der Snackbar erscheint diese Benachrichtigung und der Fahrer hat die Möglichkeit, die Simulation durch einen Klick auf "Simulieren" zu starten. Sollte der Fahrer, nachdem das Fahrtangebot akzeptiert wurde, die Seite neu laden bzw. neu zu dieser hin navigieren ist anstatt des "Zurückziehen"-Buttons, ein "Simulieren"-Button vorhanden, über welchen zur Simulation navigiert werden kann.

- **Simulation:** Nachdem ein Kunde das Fahrtangebot eines Fahrers akzeptiert hat, können sowohl Fahrer als auch Kunde über das Klicken auf die entsprechenden Knöpfe (s.o.) zur Simulation navigieren. Solange nicht beide zur Simulation navigiert haben, ist die Navigation blockiert. Sobald beide Teilnehmer der Fahrt anwesend sind, wird die "Simulations-Steuerung" aktiviert. Durch ein Klicken auf das Play- bzw. Pause-Icon kann die Simulation gestartet bzw. pausiert werden. Durch den Slider rechts daneben lässt sich die Simulationsdauer zwischen drei und 30 Sekunden einstellen.

Nach Abschluss der Simulation erscheint ein Dialog, durch welchen der Kunde bzw. der Fahrer die Fahrt bewerten kann. Dafür muss der Benutzer auf die jeweilige Sternenanzahl die er

geben will, klicken und abschließend auf “Bewerten” klicken. Danach wird er zurück auf die Startseite geleitet.

- Fahrhistorie: Sowohl Kunden als auch Fahrer können ihr Fahrhistorie ansehen. Zu dieser gelangt man durch ein Klicken auf “Fahrhistorie” in der Navbar. Dabei werden die vergangenen Fahrten in tabellarischer Form angezeigt. Der Benutzer kann die Tabelle entsprechend jeder Spalte sortieren. Über den Input in der linken oberen Ecke kann der Nutzer vergangene Fahrten über den Namen des Kunden bzw. Fahrers suchen.

Wichtiger Hinweis: In der Version v2-RC2 war noch ein fataler Fehler vorhanden, bei welchem Seiten nicht richtig refresht wurden. Dieser ist bisher in der finalen Version v2-RC4 nicht aufgetreten. Sollte bei der Ausführung einer Seite diese trotzdem nicht wie in der Bedienungsanleitung beschrieben aussehen bzw. funktionieren, so ist diese bitte nach einer kurzen Wartezeit neu zu laden.