

SEP Projektmappe

PROJEKTMAPPE DES PROJEKTES

I <3 Data

Dokumentation des Projektes

Gruppe D:



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 - Super Effizientes Präfekturverwaltungssystem

Einleitung

Das Super Effiziente Präfekturverwaltungsprogramm soll ein System werden, das für den Anwender eine Brücke zwischen Datensätzen und einer Aufbereitung derselben bildet. So soll es Möglichkeiten zum Einlesen von Datensätzen in verschiedenen Formaten geben. Diese Datensätze können ebenfalls von Systemadministratoren überprüft und bearbeitet werden. Weiter können Nutzer in Diskussionsforen über Datensätze diskutieren.

Zyklus 1

(Allgemeines)

Im ersten Zyklus soll zunächst die Grundstruktur des Systems implementiert werden. Dem System soll eine Service-Architektur (bestehend aus mindestens einem Frontend-Service und einem Backend-Service) zu Grunde liegen. Sämtliche Daten des Systems sollen persistent gespeichert werden.

(Registrierung eines Nutzers)

Nutzer sollen die Möglichkeit haben, sich zu registrieren und somit ein Profil zu erstellen. Ein Nutzerprofil soll aus dem Vor- und Nachnamen des Nutzers, einer gültigen E-Mail Adresse, dem Geburtsdatum des Nutzers, einem Passwort und einem optionalen Profilbild bestehen. Ein Nutzer soll die Möglichkeit haben, sich im System mittels seines Profils einzuloggen. Der Login-Vorgang soll durch eine Zwei-Faktor-Authentifizierung realisiert werden, bei der, neben dem Passwort, ein zufällig generierter Sicherheitscode eingegeben werden muss, der per E-Mail an den Nutzer versendet wird. Für die Abnahmen: Es soll einen „super“ Sicherheitscode existieren, der ebenfalls akzeptiert wird.

(Registrierung der Systemadministratoren)

Nutzer sollen die Möglichkeit haben, sich als Systemadministrator zu registrieren und somit ihr Profil zu erstellen. Diese Systemadministrator-Profile sollen aus dem Vor- und Nachnamen, einer gültigen E-Mailadresse und einem Passwort bestehen. Ein Nutzer soll die Möglichkeit haben, sich im System mittels seines Systemadministrator-Profils einzuloggen. Der Login-Vorgang soll ebenfalls durch eine Zwei-Faktor-Authentifizierung realisiert werden, bei der, neben dem Passwort, ein zufällig generierter Sicherheitscode eingegeben werden muss, der per E-Mail an den Nutzer versendet wird. Für die Abnahmen: Es soll ein „super“ Sicherheitscode existieren, der ebenfalls akzeptiert wird.

(Datensatzgrundlage)

Das System soll folgende Datensätze aus dem Datenportal für Deutschland (Open Government¹) persistent speichern (Zur Referenz: <https://www.govdata.de/>):

- Vornamen von Neugeborenen in der Stadt Aachen 2021
- Anzahl der Arbeitssuchenden in der Städteregion Aachen
- Anzahl der Arbeitslosen in der Städteregion Aachen
- Mittlere Jahresbevölkerung nach Geschlecht - kreisfreie Städte und Kreise - Jahr (ab 1975) (XML)
- Privathaushalte- kreisfreie Städte und Kreise – Jahr (XML)
- Straßenliste Stadt Aachen
- Sterbefälle der Stadt Aachen nach Monat, 2015-2021
- Geburten der Stadt Aachen nach Monat, 2015-2021

Die Datensätze sind Online im CSV-Format vorhanden. Somit soll das Programm einen CSV-Reader bereitstellen, der es ermöglicht Datensätze in diesem Format einzulesen und anschließend darzustellen. Für die Datensätze mit (XML)-Bezeichner muss ein Reader für XML-Dateien implementiert werden. Die Datensätze sollen in tabellarischer Form angezeigt und verändert werden können. Weiterhin sollen Nutzer Datensätze anhand ihres Namens und Jahr suchen können. Letztlich sollen Nutzer die Möglichkeit haben ihren Lieblingsdatensatz in ihrem Profil zu verlinken.

Zyklus 2

(Aufbereitung der Datensätze)

Die Nutzer sollen die Möglichkeit haben, sich Datensätze in Form von Graphen oder Diagrammen anzeigen zu lassen. Es sollen mindestens 2 Alternative Anschauungsmöglichkeiten bestehen, davon eine in Form von Graphen (z.B. Treemap, Sunburst, Force-directed Graph) und eine in Form von Diagrammen (z.B. Balkendiagramm, Kuchendiagramm etc.).

(Geodatenauswertung)

Zusätzlich zu den bereits genutzten Datensätzen, soll das System in der Lage sein, Geodaten einzupflegen. Dafür sollen folgende Datensätze zum Aachener Wald in das Programm eingearbeitet werden:

- Schutzhütten im Aachener Wald
- Rettungspunkte im Aachener Wald
- Knotenpunkte im Aachener Wald

Die Datensätze sind im Geojson-Format vorhanden und sollen in Form einer Landkarte in das System eingepflegt werden. In dieser Landkarte sollen Schutzhütten, Rettungspunkte und Knotenpunkte unterschiedlich von einander dargestellt werden.

(Support-Ticket)

Die Nutzer sollen die Möglichkeit haben ein Support-Ticket zu eröffnen, um Systemadministratoren über fehlerhafte Datensätze zu informieren. Die Support-Tickets werden beim Öffnen auf den Status „In Bearbeitung“ gesetzt. Sobald der Systemadministrator das Problem gelöst hat, kann dieser den Bearbeitungsstatus auf „Erlledigt“ setzen. Über die Erledigung des Support-Tickets, soll der Nutzer per E-Mail benachrichtigt werden.

(Freundesliste)

Nutzer sollen die Möglichkeit haben, andere Nutzer als Freunde hinzuzufügen. Eine Freundesliste soll für jeden Nutzer angelegt werden, in der er seine Freunde sehen und verwalten kann. Der Nutzer soll in der Lage sein, Freundschaftsanfragen zu senden und anzunehmen oder abzulehnen. Wenn ein Nutzer eine Freundschaftsanfrage erhält, soll er darüber per E-Mail benachrichtigt werden. Die Freundesliste soll öffentlich oder privat einstellbar sein, so dass andere Nutzer sehen können, wer in der Liste enthalten ist, oder ob sie privat ist.

(Chat)

Nutzer sollen die Möglichkeit haben, mit ihren Freunden über das System zu chatten. Hierfür soll eine Chat-Funktion implementiert werden, der es den Nutzern ermöglicht, private oder Gruppenchats zu erstellen und Nachrichten zu senden. Die Nutzer sollen in der Lage sein, Nachrichten in Echtzeit zu senden und zu empfangen. Außerdem sollen die Nutzer die Möglichkeit haben, Nachrichten zu löschen oder zu bearbeiten, solange der Empfänger sie noch nicht gelesen hat.

Zyklus 3

(Chatbot)

Die Nutzer sollen die Möglichkeit haben, einen Chatbot zu nutzen. Dieser Chatbot soll Fragen beantworten können (z.B.: „Welche Dateiformate für Datensätze werden akzeptiert?“). Die Auswahl der Fragen liegt in der Entscheidung der Entwickler, allerdings muss der Chatbot mindestens 12 Interaktionen vorweisen können.

(Diskussionsforum)

Nutzer sollen die Möglichkeit haben, sich über Datensätze auszutauschen. Um dies zu ermöglichen, soll das System über ein Diskussionsforum verfügen. Jeder Nutzer kann ein Diskussionsthema eröffnen, in dem andere Nutzer kommentieren können. Ein Diskussionsthema soll einen Titel und eine Kategorie enthalten, die der Ersteller festlegt. Andere Nutzer sollen die Möglichkeit haben, dem

Diskussionsthema einen Like zu geben und das Diskussionsthema zu favorisieren. Wenn ein Nutzer ein Diskussionsthema favorisiert, erhält dieser Nutzer E-Mail-Benachrichtigungen über neue Beiträge und Änderungen in diesem Diskussionsthema. Die Diskussionen sollen von Systemadministratoren moderiert werden, um sicherzustellen, dass die Inhalte angemessen und sachlich sind.

(persönliche Profilansicht)

Die Nutzer sollen die Möglichkeit haben eine persönliche Profilansicht zu erstellen. Diese persönliche Profilansicht soll aus bis zu vier von den Benutzer auswählbaren verschiedenen Datensätzen in einer vom Nutzer gewählten Anschauungsmöglichkeit bestehen. Der Nutzer sollen die Möglichkeit haben seine persönliche Profilansicht zu bearbeiten und persistent zu speichern. Die persönliche Profilansicht soll öffentlich oder privat einstellbar sein, so dass andere Nutzer die persönliche Profilansicht sehen können, oder ob es privat ist.

(Export von Datensätzen)

Um Datensätze ebenfalls offline einsehen zu können, soll das System den Export von Datensätzen und deren 2 Alternative Anschauungsmöglichkeiten (siehe Zyklus 2) ermöglichen. Dazu soll ein Nutzer die Möglichkeit haben einen vorhandenen oder eingelesenen Datensatz im .pdf-Format zu exportieren. Die exportierte Datei, soll neben dem Datensatz und deren 2 Alternative Anschauungsmöglichkeiten auch einen geeigneten Titel und das Exportdatum beinhalten.

(Hinzufügen weiterer Datensätze)

Die Nutzer sollen die Möglichkeit haben, das System um weitere Datensätze zu erweitern. Dabei übergeben die Nutzer dem System einen Link unter dem ein Datensatz zu finden ist (z.B.: <https://www.govdata.de/web/guest/suchen/-/details/vornamen-von-neugeborenen-in-der-stadt-aachen-2021>). Das System speichert den Datensatz auf Basis des Link persistent. Vorher implementierte Funktionen, wie das Exportieren oder Aufbereiten von Datensätzen, sollen mit diesen zur Laufzeit hinzugefügten Datensätzen ebenfalls möglich sein. Anmerkungen

Die Datenbank soll als lokale Datenbank angelegt werden (idealerweise als ein eigenständiger Container oder im Backend-Image eingebettet). Bei Datenbanken, die online bereitgestellt werden, kann es passieren, dass diese bei der Abnahme nicht zur Verfügung stehen.

Wir stellen im Laufe der Veranstaltung Technologien vor, mit denen sich diese Aufgabenstellung realisieren lässt. Sollten Sie alternative Technologien verwenden wollen, so können Sie dies gerne tun. Allerdings können wir in diesem Fall eine Hilfestellung bei Problemen nicht garantieren.

Wir veröffentlichen besonders gelungene Software auf unserer SEP-Webseite. Hierzu ist es unbedingt erforderlich, dass das System keine urheberrechtlich geschützten Inhalte (Bilder, Musik, etc.) enthält.

Zyklus I

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
1.	Registrierung der Nutzer			
	User Story 1.1	User Story		fertig
	User Story 1.2	User Story		fertig
	Registrierungsfenster	Papierprototyp		fertig
	Registrierung-Datum-Dropdown	Papierprototyp		fertig
2.	Login der Nutzer			
	User Story 2.1	User Story		fertig
	User Story 2.2	User Story		fertig
	User Story 2.3	User Story		fertig
	Login-Fenster	Papierprototyp		fertig
	2-FaktAuth-Seite	Papierprototyp		fertig
3.	Profil			
	User Story 3.1	User Story		fertig
	User Story 3.2	User Story		fertig
	User Story 3.3	User Story		fertig
	User Story 3.4	User Story		fertig
	User Story 3.5	User Story		fertig
	Profilansicht	Papierprototyp		fertig
	eigenes Profil bearbeiten	Papierprototyp		fertig
4.	Navigation			
	User Story 4.1	User Story		fertig
	User Story 4.2	User Story		fertig
	User Story 4.3	User Story		fertig
	Hamburger Menü	Papierprototyp		fertig

	Dashboard	Papierprototyp	Kanista Karunanathan	fertig
5.	Datensatznutzung			
	User Story 5.1	User Story		fertig
	User Story 5.2	User Story		fertig
	User Story 5.3	User Story		fertig
	Liste der Datensätze	Papierprototyp		fertig
	Datensatztabelle Nutzer	Papierprototyp		fertig
6.	Adminrechte zur Datensatzveränderung			
	User Story 6.1	User Story		fertig
	User Story 6.2	User Story		fertig
	User Story 6.3	User Story		fertig
	Admin-Ansicht-Tabellenbearbeitung	Papierprototyp		fertig
7.	Bezug zu anderen Nutzern			
	User Story 7.	User Story		fertig
	Profilansicht-anderer-Nutzer	Papierprototyp		fertig

User-Stories

User Story-ID	1.1
User Story-Beschreibung	Als Nutzer möchte ich mich mit E-Mail, Namen und Passwort registrieren können, um das System nutzen zu können.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	-

User Story-ID	1.2
User Story-Beschreibung	Als Nutzer möchte ich einen neuen Systemadministrator anlegen können, um mehr Rechte zu haben.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	-

User Story-ID	2.1
User Story-Beschreibung	Als Nutzer möchte ich mich mit Email und Passwort eingeben können, um mich einzuloggen.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	2.3

User Story-ID	2.2
User Story-Beschreibung	Als Systemadministrator möchte ich mich mit E-Mail und Passwort einloggen können, um die Systemfunktionen nutzen zu können.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	2.3

User Story-ID	2.3
User Story-Beschreibung	Als Nutzer möchte ich die Zwei-Faktor-Authentifizierung nutzen können, um mein Konto zu sichern.
Geschätzter Realisierungsaufwand	>= 3 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	2.1, 2.2

User Story-ID	3.1
User Story-Beschreibung	Als Nutzer möchte ich mein eigenes Profil bearbeiten können, damit ich meine Daten aktualisieren kann.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	3.2, 3.3, 3.4, 3.5

User Story-ID	3.2
User Story-Beschreibung	Als Nutzer möchte ich ein Profilbild zu meinem Profil hinzufügen können, damit dieses auf meiner Profilübersicht sichtbar ist.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	3.1

User Story-ID	3.3
User Story-Beschreibung	Als Nutzer möchte ich Profilbilder ändern können, damit das dortige Bild aktuell ist.
Geschätzter Realisierungsaufwand	<= 1 Tage
Priorität	Mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	3.1, 3.2

User Story-ID	3.4
User Story-Beschreibung	Als Nutzer möchte ich meine Lieblings-Datensätze meinem Profil hinzufügen können, damit ich schnell auf sie zugreifen kann.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	3.1, 5.1

User Story-ID	3.5
User Story-Beschreibung	Als Nutzer möchte ich meine Lieblings-Datensätze entfernen können, damit meine Lieblings-Datensätze aktuell sind.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	3.1, 3.4, 5.1

User Story-ID	4.1
User Story-Beschreibung	Als Nutzer möchte ich einen Knopf im Hamburger Menü haben, um mich aus dem System auszuloggen zu können.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	4.2

User Story-ID	4.2
User Story-Beschreibung	Als Nutzer möchte ich mit einem Hamburger Menü durch die Seite navigieren können, um zu anderen Bereichen zu gelangen.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	4.1

User Story-ID	4.3
User Story-Beschreibung	Als Nutzer möchte ich über ein Dashboard verfügen, um eine Übersicht über die Funktionen zu haben und zu den Funktionen zu navigieren.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	4.2

User Story-ID	5.1
User Story-Beschreibung	Als Nutzer möchte ich Datensätze in einer Tabellenansicht einsehen können, weil mir dies die beste Übersicht gibt.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	-

User Story-ID	5.2
User Story-Beschreibung	Als Nutzer möchte ich innerhalb einer Tabelle nach bestimmten Daten suchen können, um diese finden zu können.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	5.1

User Story-ID	5.3
User Story-Beschreibung	Als Nutzer möchte ich einen Datensatz per Name aus der Datensatzliste heraussuchen können, um diesen finden zu können.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	-

User Story-ID	6.1
User Story-Beschreibung	Als Admin möchte ich auf der Adminseite neue Datensätze einspeisen können, damit stets aktuelle Informationen vorhanden sind.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	-

User Story-ID	6.2
User Story-Beschreibung	Als Admin möchte ich Datensätze entfernen, um fehlerhafte/veraltete/doppelte Datensätze aus dem System zu entfernen.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	6.1

User Story-ID	6.3
User Story-Beschreibung	Als Admin möchte ich die Möglichkeit haben, die Datensätze zu aktualisieren, damit die Daten auf aktuellem Stand bleiben.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	Hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	6.1

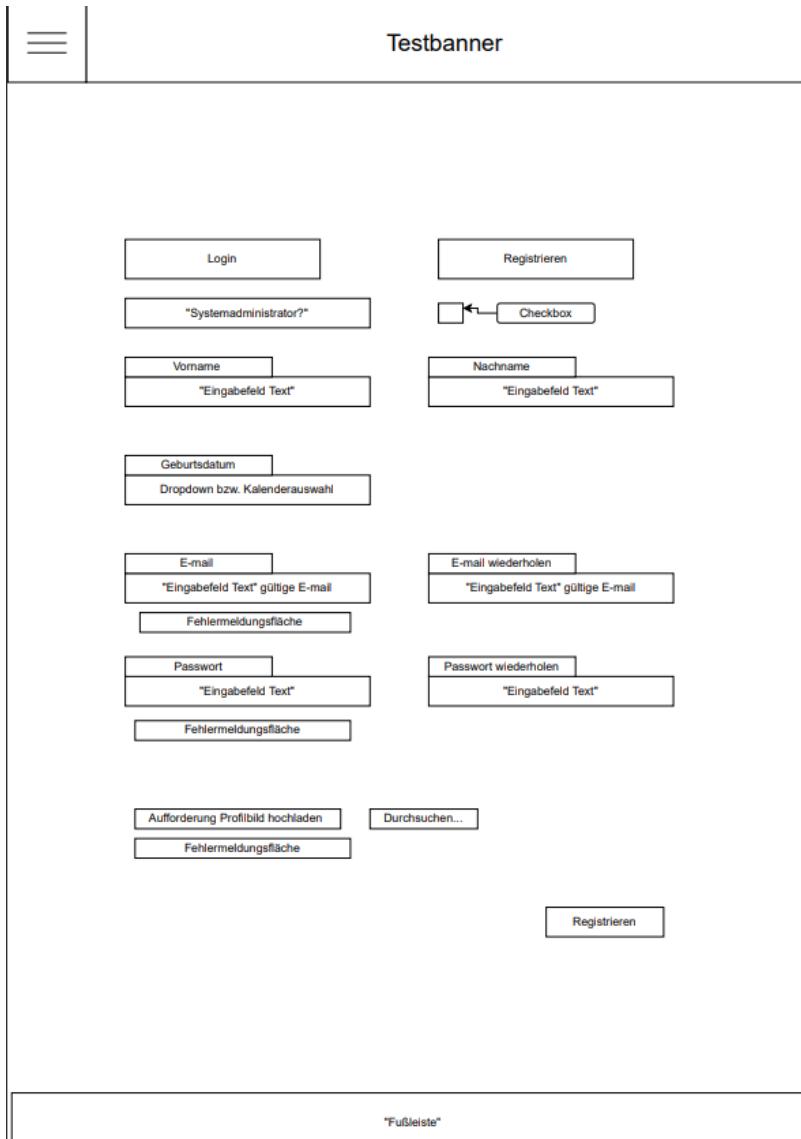
User Story-ID	7.
User Story-Beschreibung	Als Nutzer möchte ich Profile anderer User betrachten können, um zu sehen, wer noch im System eingetragen ist.
Geschätzter Realisierungsaufwand	<= 1 Tag
Priorität	Niedrig
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	3. (Profile müssen vorhanden sein)

Papierprototypen

Legende:

- "Text" → Textflächen/Indikatoren; keine klickbaren Knöpfe
- Text → klickbare Knöpfe
- Runde Kästen: Erläuterungen

Registrierungsfenster:



anklickbare Knöpfe:

- Login → **Login-Fenster**
- Durchsuchen → **Dateiauswahlfenster**
- Registrieren
 - Eingaben korrekt → **2-FaktAuth-Seite**
 - Eingabe fehlerhaft → **Registrierungsfenster** mit entspr. Fehlermeldung
- Dropdown bzw. Kalenderauswahl → **Registrierung-Datum-Dropdown**
- ☰ → **Hamburger-Menü**

Registrierung-Datum-Dropdown:

	Testbanner
<div style="display: flex; justify-content: space-around;"> Login Registrieren </div>	
<div style="display: flex; justify-content: space-around;"> "Systemadministrator?" <input type="checkbox"/> ← Checkbox </div>	
<div style="display: flex; justify-content: space-around;"> Vorname Nachname </div>	
<div style="display: flex; justify-content: space-around;"> "Eingabefeld Text" "Eingabefeld Text" </div>	
<div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Geburtsdatum</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Dropdown bzw. Kalenderauswahl</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">01 Januar 2023</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">02 Februar 2022</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">03 März 2021</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">"Eingabefeld Text" gültige E-mail</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Fehlermeldungsfläche</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Passwort</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Passwort wiederholen</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Aufforderung Profilbild hochladen</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Durchsuchen...</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Fehlermeldungsfläche</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Registrieren</div> </div>	
<div style="display: flex; justify-content: space-around;"> "Eingabefeld Text" "Eingabefeld Text" </div>	
<div style="display: flex; justify-content: space-around;"> "Eingabefeld Text" "Eingabefeld Text" </div>	
<div style="display: flex; justify-content: space-around;"> "Fußleiste" </div>	

- **☰ → Hamburger-Menü**
- neben das Kalender-Auswahlfenster klicken → **Registrierungsfenster**

Login-Fenster:

The wireframe shows a login interface. At the top left is a hamburger menu icon. To its right is the title "Testbanner". Below the title are two rectangular buttons: "Login" on the left and "Registrieren" on the right. The next section contains two input fields: "Username" and "Passwort", each with a descriptive label above it. Below these fields are two more rectangular buttons: "Passwort vergessen" on the left and "Login" on the right. At the bottom of the window is a horizontal bar labeled "Fußleiste".

anklickbare Knöpfe:

- **☰ → Hamburger-Menü**
- Registrieren → **Registrierungsfenster**
- Login → **2-FaktAuth-Seite**
- ggf. Passwort vergessen-Schaltfläche

2-FaktAuth-Seite:

The screenshot shows a user interface for a two-factor authentication page. At the top left is a hamburger menu icon. To its right is a banner labeled "Testbanner". Below the banner is a section titled "2-Faktor Authentifizierung". This section contains three buttons: "Logout", "Code erneut senden", and "Abschicken". Below these buttons is a text input field with the placeholder "Bitte geben Sie den Code, der Ihnen per E-mail gesendet wurde hier ein". At the bottom of this section is a button labeled "Fehlermeldung".

"Fußleiste"

anklickbare Knöpfe:

- **☰ → Hamburger-Menü**
- **Logout → Login**
- **abschicken**
 - korrekt → **Dashboard**
 - falsch → **2-FaktAuth-Seite** mit Fehlermeldung
- **Code erneut senden → 2-FaktAuth-Seite**

Eigene Profilansicht:

The mockup shows a user profile page. At the top right is a 'Testbanner' (test banner). Below it is a circular placeholder for a profile picture with the button 'Profilbild ändern...' (Change profile picture). To the left of the profile picture is a box containing the text 'Profil von Max Mustermann'. Below this are two buttons: 'Persönliche Daten' (Personal data) and 'bearbeiten' (edit). The main area contains five input fields for personal data: Vorname (first name), Nachname (last name), E-Mail (email), Geburtsdatum (date of birth), and another Vorname field. Below this is a section titled 'Lieblingdatensätze' (Favourite data sets) containing four items: 'Datensatz 1' (Data set 1) with an 'X' mark, 'Datensatz 2' (Data set 2) with an 'X' mark, 'Datensatz 3' (Data set 3) with an 'X' mark, and 'Datensatz 4' (Data set 4) with an 'X' mark. A button 'Datensatz hinzufügen' (Add data set) is located below the list.

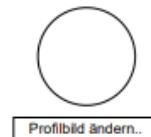
anklickbare Knöpfe:

- \equiv → **Hamburger-Menü**
- bearbeiten → **eigenes-Profil-bearbeiten**
- Datensatz hinzufügen → **Liste der Datensätze**
- x bei best. Lieblingsdatensatz → **Profilansicht** (entspr. Datensatz entfernt)

eigenes-Profil-bearbeiten:

≡	Testbanner
---	------------

"Profil von Max Mustermann"	
"Persönliche Daten" bestätigen	
"Vorname"	
"Nachname"	
"E-Mail"	
"Geburtsdatum"	



"Lieblingdatensätze"	
----------------------	--

Datensatz 1	X
Datensatz 3	X

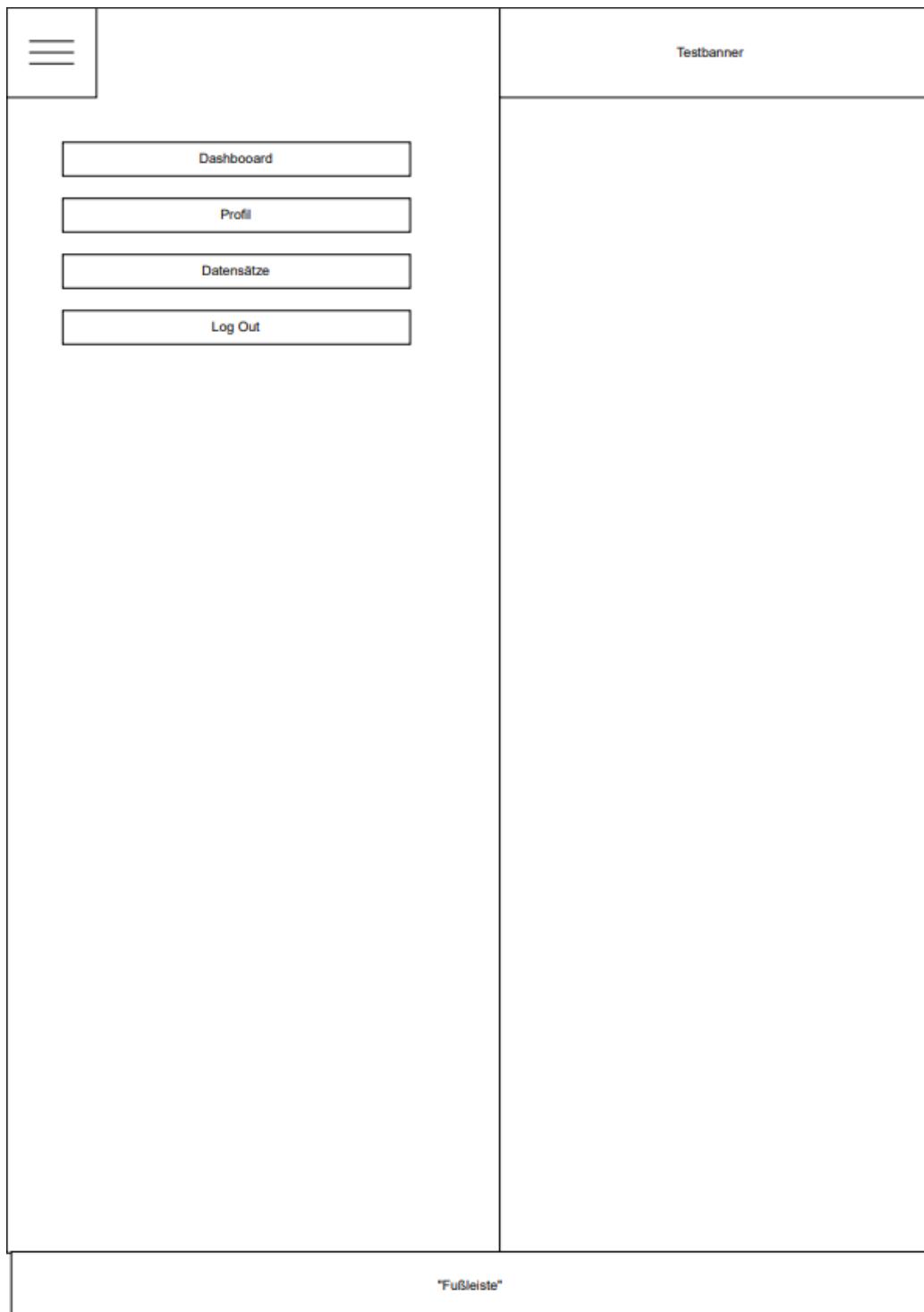
Datensatz hinzufügen

	"Fußleiste"
--	-------------

anklickbare Knöpfe:

- ≡ → **Hamburger-Menü**
- bestätigen → **Profilansicht**
- Profilbild ändern → **Dateiauswahlfenster**
- Datensatz hinzufügen → **Liste der Datensätze**
- Ändern des Geburtsdatums = analoges Fenster zu **Registrierung-Datum-Dropdown** öffnet sich

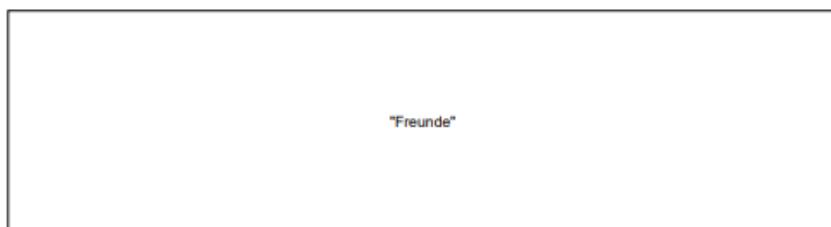
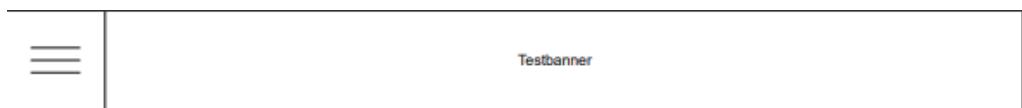
Hamburger-Menü:



anklickbare Knöpfe:

- $\equiv \rightarrow$ **Hamburger-Menü**
- Dashboard \rightarrow **Dashboard**
- Profil \rightarrow **Profilansicht**
- Datensätze \rightarrow **Liste der Datensätze**
- Logout bzw. wenn nicht angemeldet \rightarrow **Login**

Dashboard:



anklickbare Knöpfe:

- **≡ → Hamburger-Menü**
- Datensatzauswahl unter “beliebte Datensätze” → **Datensatztabelle Nutzer** bzw. **Admin-Ansicht-Tabellenbearbeitung**
- Personenauswahl unter “Freunde” → **Profilansicht-anderer-Nutzer**

Liste der Datensätze:



anklickbare Knöpfe:

- $\equiv \rightarrow$ **Hamburger-Menü**
- Auswahl einer der Datensätze \rightarrow **Datensatztabelle Nutzer bzw. Admin-Ansicht-Tabellenbearbeitung**
- $*$ \rightarrow **Liste der Datensätze** (Stern nun andere Farbe = Datensatz zu Lieblingsdatensätzen hinzugefügt)

Datensatztabelle Nutzer:

The mockup shows a header section with a hamburger menu icon and a banner labeled "Testbanner". Below this is a toolbar with buttons for "Sichtbar nur für Admins" (visible only for admins), "Datensatz 1*", "Suche:", and "Favoriten" (with a star icon). A search bar and a magnifying glass icon are also present. The main area is a data grid with 10 rows and 6 columns, each containing a "Suchfeld" (search field) and a downward arrow icon.

"Fußleiste"

anklickbare Knöpfe:

- \equiv → **Hamburger-Menü**
- * → **Datensatztabelle Nutzer** (Stern nun andere Farbe = Datensatz zu Lieblingsdatensätzen hinzugefügt/davon entfernt)
- Lupe → **Datensatztabelle Nutzer**

Admin-Ansicht-Tabellenbearbeitung:

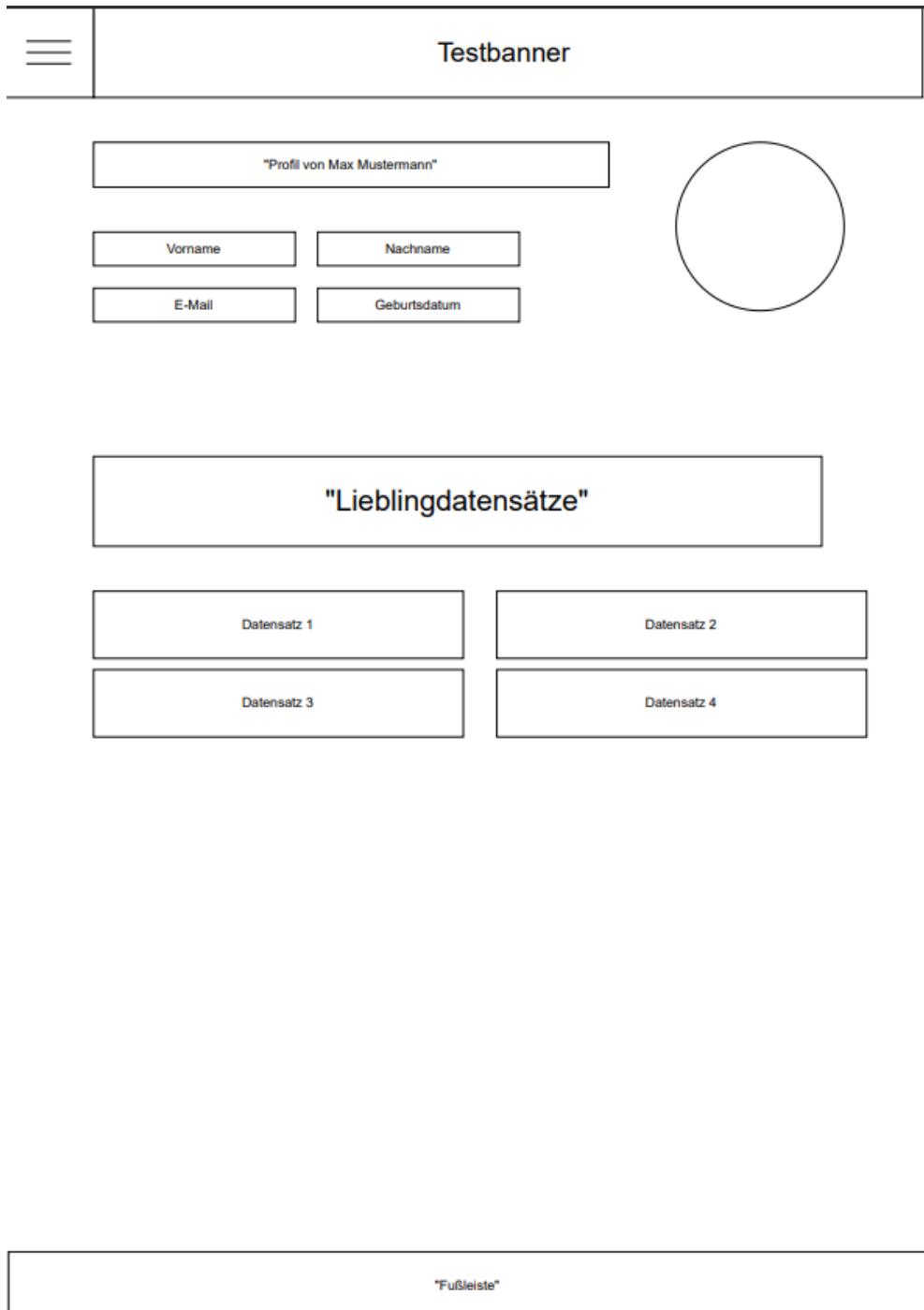
The screenshot shows the Admin-View Table Editing interface. At the top left is a hamburger menu icon. To its right is a banner with the text "Testbanner". Below the banner are several buttons: a toggle button for "Sichtbar nur für Admins" (Visible only for Admins), a button for "Datensatz 1" (Data Record 1), a "Favoriten" (Favorites) button with a star icon, and buttons for "Suche" (Search), "Daten hinzufügen" (Add Data), "Daten entfernen" (Delete Data), and "Daten updaten" (Update Data). Below these buttons is a row of five columns labeled "Column 1" through "Column 5", each containing an "Eingabefeld" (Input Field). A green checkmark icon is positioned next to the fifth column. Below this row is a section labeled "Fehlerausgabe" (Error Output). The main area consists of a data grid with 10 rows and 6 columns, each containing a downward arrow icon.



anklickbare Knöpfe:

- **Ξ → Hamburger-Menü**
- *** → Admin-Ansicht-Tabellenbearbeitung** (Stern nun andere Farbe = Datensatz zu Lieblingsdatensätzen hinzugefügt)
- Suche / Daten hinzufügen / Daten entfernen / Daten updaten → **Admin-Ansicht-Tabellenbearbeitung**, Columns und Eingabefelder beziehen sich nun auf entspr. Bearbeitungsart
- Bestätigung → **Admin-Ansicht-Tabellenbearbeitung** (nach suchen, hinzufügen, entfernen oder updaten oder mit Fehlermeldung)

Profilansicht-anderer-Nutzer:

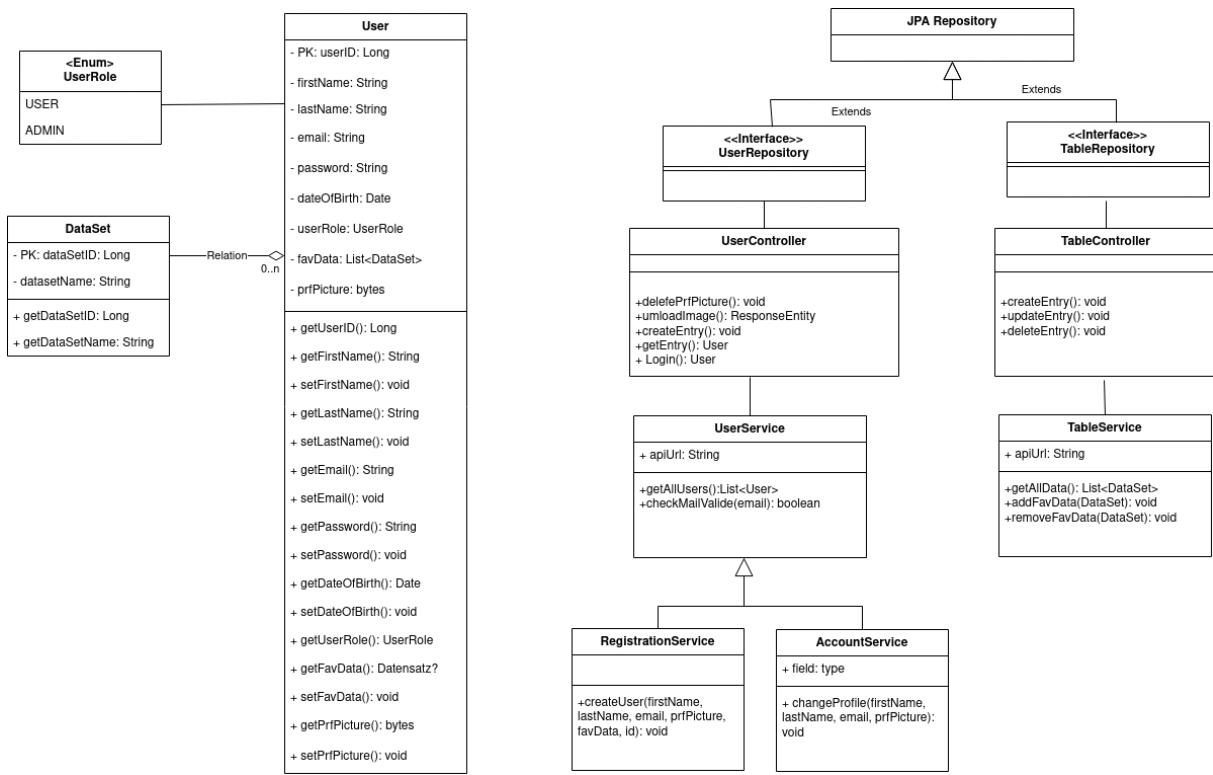


anklickbare Knöpfe:

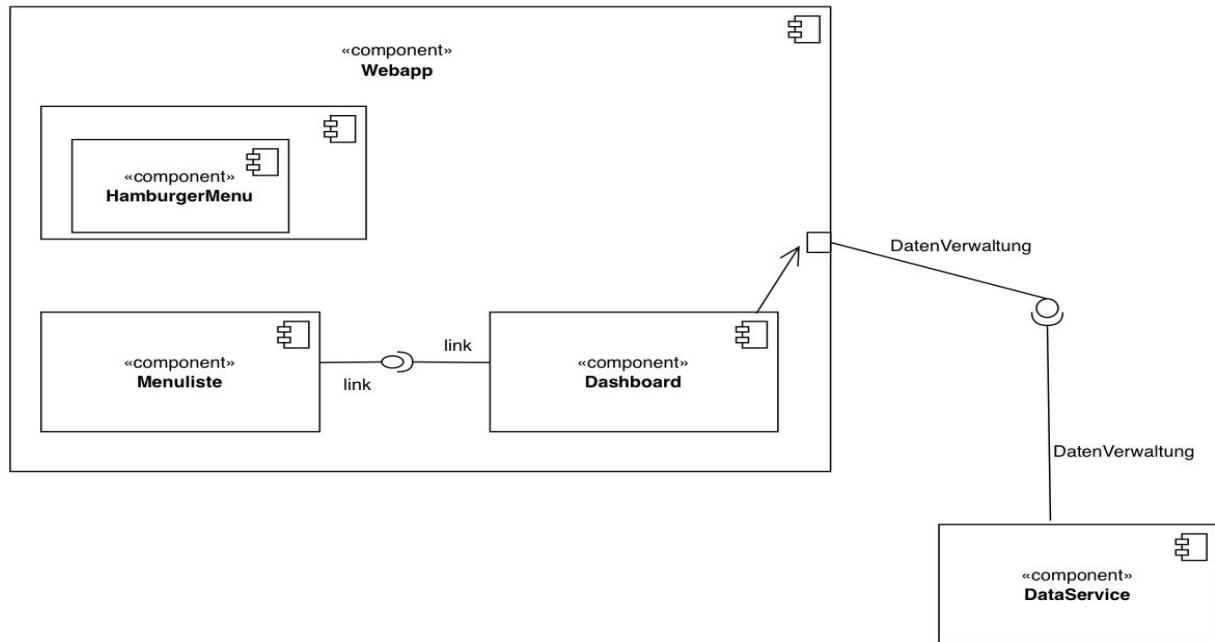
- $\equiv \rightarrow$ Hamburger-Menü
- Datensätze \rightarrow Datensatztabelle Nutzer bzw. Admin-Ansicht-Tabellenbearbeitung

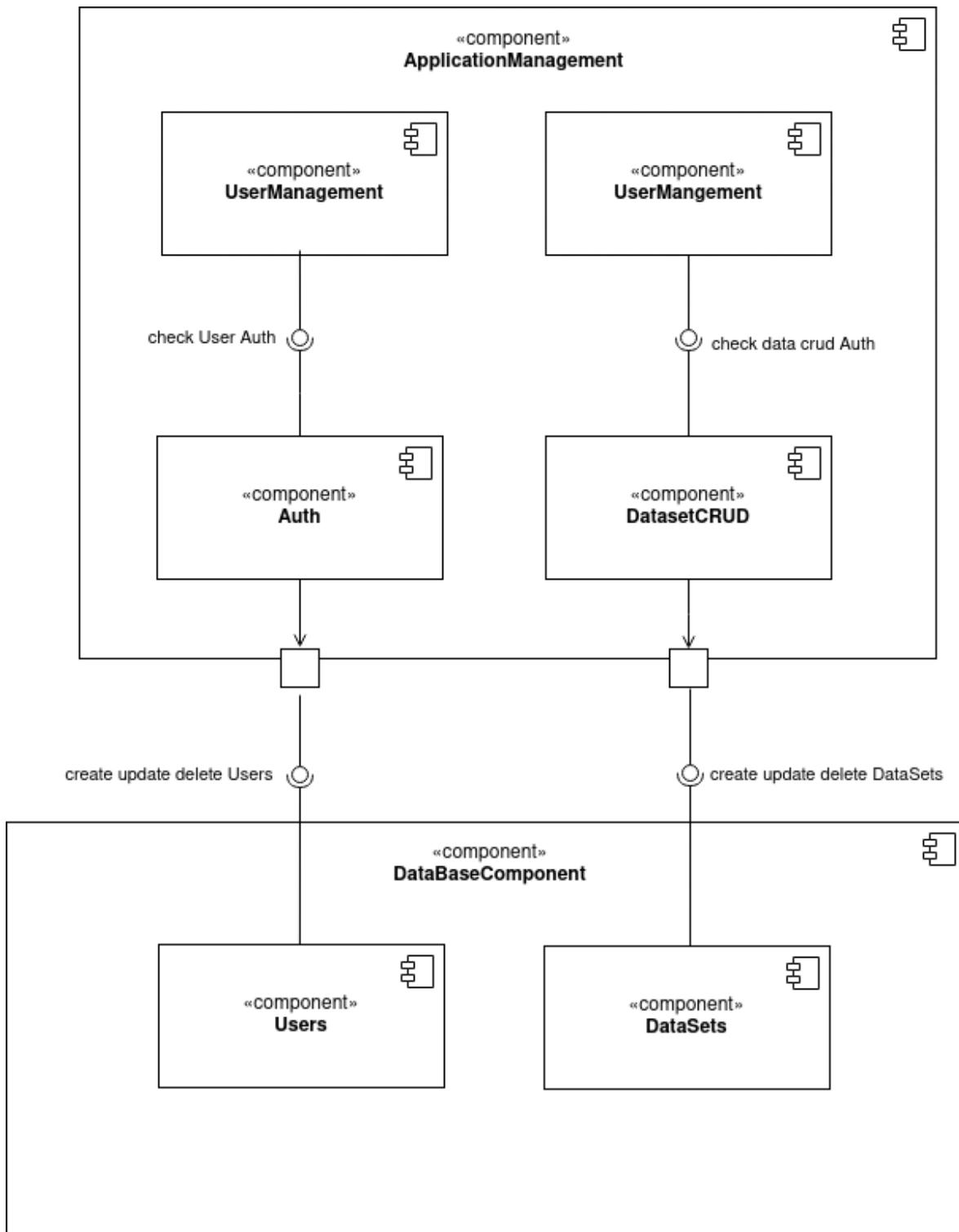
Strukturdiagramme (Komponenten- und Klassendiagramme)

Klassendiagramme



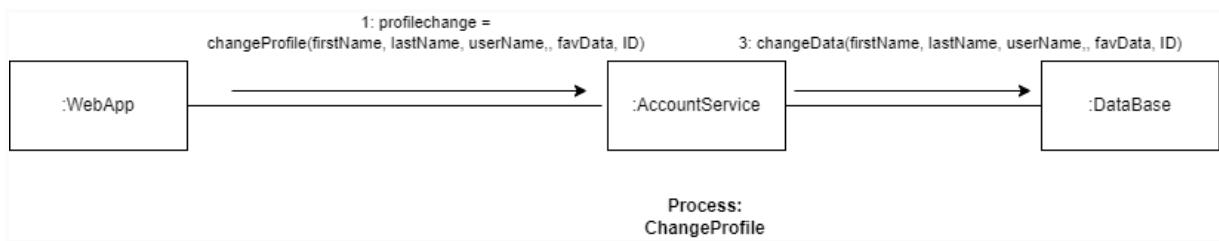
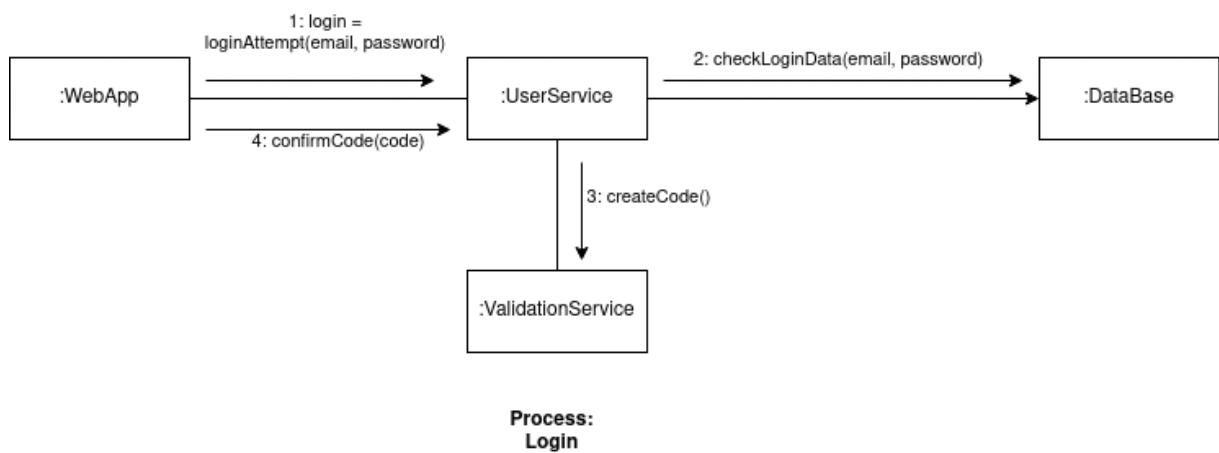
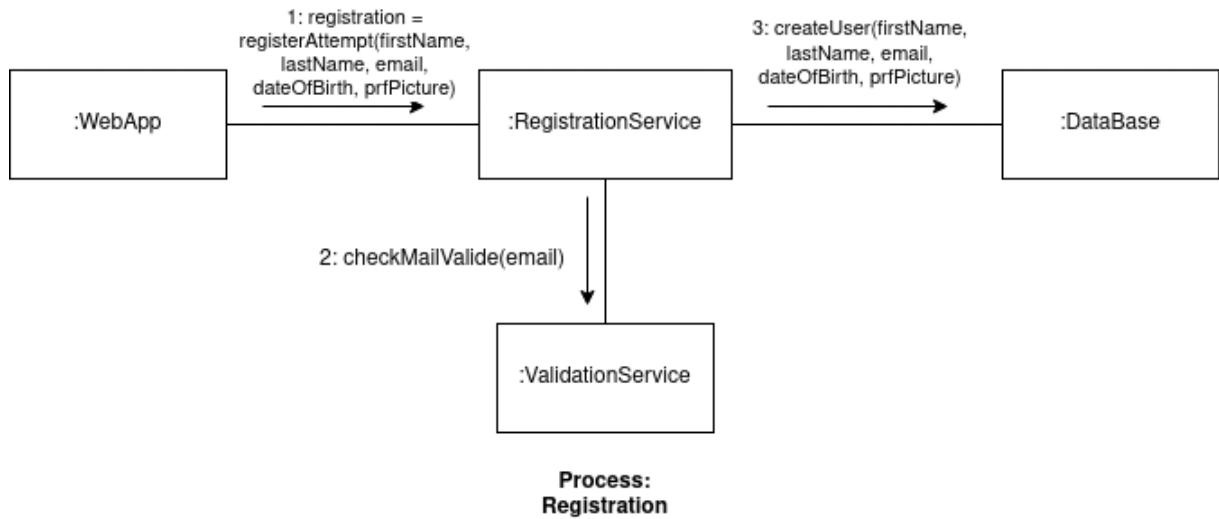
Komponentendiagramme

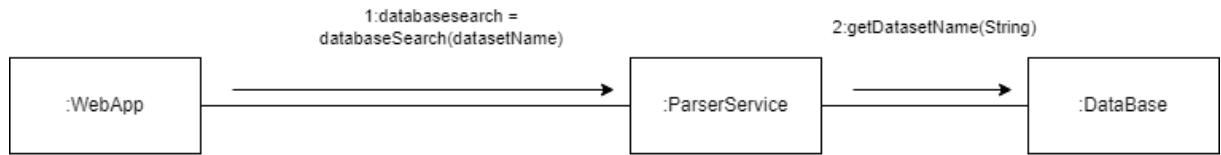




Verhaltensdiagramme (Kommunikationsdiagramme)

Aufgeführt sind die geplanten Services, d.h. Servicenamen und Zuständigkeiten haben sich im Laufe der Entwicklung zum Teil verändert:





Process:
Database Search



Process:
Data Upload



Process:
Data Changes



Process:
Data deletion

Funktionalitätsplanung

ID	Funktionalität	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
0.1	html-Grundstruktur	[REDACTED]			bottom (.html, .scss, .spec.ts, .ts) app-component.html styles: _cards.scss, _header.scss	fertig
0.2	Anwendungsgrundstruktur anlegen	[REDACTED]			/src/app; /src/main	fertig
0.3	Routing	[REDACTED]			app.routing.module.ts	fertig
1.	Registrierung					
1.1	Registrierungsview (frontend)	[REDACTED]	0.1	1.1, 1.2	registration: registration.html, registration.scss	fertig
1.2	Registrations-eingaben-Prüfung	[REDACTED]	1.1	1.1, 1.2	registration: registration.component.ts	fertig
1.3	Nutzer speichern (→ backend)	[REDACTED]	6.1	1.1, 1.2	registration: registration.component.ts → userService (Z. 88-91, 39-43)	fertig
1.4	Verknüpfung Email-Senden & 2-Fakt-Auth nach Registration	[REDACTED]	1.2, 2.2, 2.5	1.1, 1.2	registration: registration.component.ts → this.EmailService (Z. 102ff)	fertig
2.	Login					
2.1	Login-View (frontend)	[REDACTED]	0.1, 1.1, 0.3	2.1, 2.2	login: login.html, login.scss	fertig
2.2	2-Faktor-Authentifizierung: Email-Service & -Controller	[REDACTED]	0.1, 1.2, 0.3	2.3	EmailService.ts, EmailController.java, application.properties (ab Z.11)	fertig
2.3	Auth-View (frontend)	[REDACTED]	0.1, 1.2	1.1	two-fa: two-fa.html, two-fa.scss	fertig

2.4	Supersicherheitscode einbinden	Edward	2.2		EmailSenderService.java: createVerificationCode(), getVerificationCode()	fertig
2.5	Email-Versendung (backend)			2.3	EmailSenderService.java: sendMail(), sendRegisterConfirm(), sendAuthCode()	fertig
2.6	Login-Sicherung für interne Views		0.1, 1.1, 1.2	4.1, 2.1, 2.3	AuthenticationController.java, AuthenticationService.java, AuthenticationResponse.java, AuthenticationRequest.java, RegisterRequest.java	fertig
3.	Profil					
3.1	View eigenes Profil (frontend)		0.1, 1.3, 0.3	3.1, 3.2, 3.4	editprofile: editprofile.html, editprofile.css,	fertig
3.2	view eigenes Profil ändern (frontend)		0.1, 1.3, 3.1, 0.3, 2.6	3.2, 3.3, 3.4, 3.5	editprofile: editprofile.spec.ts, editprofile.ts, dataview.ts, dataview.html	fertig
3.3	view Profil anderer (frontend)		0.1, 1.3, 0.3, 2.6	7	viewprofile .html, .ts, .spec.ts, .css)	fertig
3.4	Profil-Veränderung (→ backend)		3.1, 3.2	3.1	UserController.java, UserRepository.java, Image.java, ImageRepository.java, ImageUploadResponse.java, userservice.ts user.ts	fertig
3.5	Nutzerklasse - Controller (backend)			1.1, 1.2, 3.1, 3.2, 3.3, 3.4, 3.5	User.java	fertig
3.6	view Nutzerliste		0.3, 0.1, 3.5		userview (.html, .ts, .spec.ts, .css),	fertig

	(1x für Entwicklung & 1x für andere Nutzer)				userlist (.html, .ts, .spec.ts, .css)	
4.	Datenbank					
4.1	view Datenbank ändern (Admin) (→ backend)	Human-Basic	0.1, 6.2, 0.3, 2.6	6.2, 6.3	tableview<Nr. 1-8> (.html, .ts, .spec.ts, .css)	fertig
4.2	view liste von Datenbanken (frontend)		0.1, 4.3, 6.2, 0.3, 2.6	5.3	dataview (.html, .ts, .spec.ts, .css)	fertig
4.3	Datenliste nicht-Admin (frontend)		0.1, 0.3, 2.6	5.1, 5.2	tableview<Nr. 1-8> (.html, .ts, .spec.ts, .css)	fertig
4.4	Datensätze einspeisen (→ backend)		5.1	6.1	Table<Nr. 1-8>Controller.java, Table<Nr. 1-8>Repository.java, Table<Nr. 1-8>Service.java; VNeugeboreneAC.java, Sterbefaelle.java, Arbeitssuchende.java, Arbeitslose.java, MittlereJahresBevoelkerung.java, Privathaushalte.java, Strassenliste.java, Geburten.java	fertig
4.5	CSV-Reader		5.1, 4.4	6.1	PHPParserDOM.java	fertig
4.6	XML-Reader		5.1, 4.4	6.1	MJBPParserDOM.java	fertig
5.	Navigation					
5.1	view Dashboard (frontend)		0.1, 0.3, 2.6	4.3	dashboard(.html, .css, .spec.ts, .ts)	fertig
5.2	Hamburger Menü (frontend)		0.1, 0.3	4.1, 4.2	hamburgermenu(.html, .scss, .spec.ts, .ts)	fertig
6.	Backend					

6.1	Datenbank anlegen		6.2, 0.2		application.properties (Z. 6-10)	fertig
6.2	REST API erstellen (springboot)		6.1, 0.2		RestApplication.java, pom.xml	fertig
6.3	Web-Browser Anbindung		0.2, 6.1			fertig
7.	Docker					
7.1	Docker images		6.2, 6.3		Dockerfile_backend, Dockerfile_frontend	fertig
7.2	tar Dateien		6.2, 7.1		Dockerfile_backend.tar, Dockerfile_frontend.tar	fertig
7.3	Verknüpfung von Docker-Images in 1 Netzwerk und Speichern der Datenbank (Persistenz)		7.1, 7.2, 6.1		docker-compose.yml, application.properties (Z. 1-5)	fertig

Systemtests

Parallelgruppe = Gruppe C

Datum	11.5.2023		
Tester			
SW-Version	1		
Vorbedingung(en)	Registrationsseite geöffnet, kein angemeldeter User		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer gibt seine Registrationsdaten ein: Vorname, Nachname, Geburtsdatum, gewünschter Nutzernname, Passwort und E-Mail-Adresse. Der Nutzer wiederholt E-Mail-Adresse und Passwort.	Die eingegebenen Daten werden in den Eingabefeldern angezeigt. Das Passwort als *.	✓
2	Der Nutzer klickt auf den Registrieren-Button.	Bei Falschangaben wird der Nutzer aufgefordert, diese zu korrigieren.	✓
3	Ggf. korrigiert der Nutzer die Eingaben und klickt erneut auf Registrieren.	Der Nutzer erhält eine E-Mail, dass die Registration erfolgreich war, und wird auf die 2-Faktor-Authentifizierungsseite weitergeleitet.	✓
Nachbedingung(en)	2-Faktor-Eingabe-Seite		
Testurteil	Test bestanden		

Datum	11.5.2023		
Tester			
SW-Version	1		
Vorbedingung(en)	Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ und E-Mail-Adresse „max@gmail.com“ 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“.	Es erfolgt eine Weiterleitung auf die Bestätigungsseite der 2-Faktor-Authentifizierung und eine E-Mail wird gesendet an „max@gmail.com“ mit einem 6-stelligen Code.	✓
4	Der Benutzer gibt den erhaltenen Code ein und bestätigt das ganze	Der Benutzer wird auf das Dashboard weitergeleitet.	X
Nachbedingung(en)	Der Nutzer ist im System angemeldet.		
Testurteil	Test nicht bestanden		

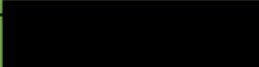
Datum	12.5.2023		
Tester	[REDACTED]		
SW-Version	1		
Vorbedin-gung(en)	Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ und E-Mail-Adresse „max@gmail.com“ registriert.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer gibt den Benutzernamen „Max Mustermann“ auf der Tastatur 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“.	Es erfolgt eine Weiterleitung auf die Bestätigungsseite der 2-Faktor-Authentifizierung und eine E-Mail wird gesendet an „max@gmail.com“ mit einem 6-stelligen Code.	✓
4	Der Benutzer gibt den erhaltenen Code ein und bestätigt das ganze	Der Benutzer wird auf das Dashboard weitergeleitet.	✓
Nachbedingun g(en)	Der Nutzer ist im System angemeldet.		
Testurteil	Test bestanden.		

Datum	11.5.2023		
Tester			
SW-Version	1		
Vorbedingung(en)	Nutzer ist am System angemeldet und registriert		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Benutzer klickt im Hamburgermenü auf "Profil".	Der Nutzer ist jetzt auf einer Seite, wo er sein Profil bearbeiten kann.	✓
2	Der Benutzer klickt auf Upload Image, wählt ein Bild aus und klickt auf "Upload Image"	Das System zeigt das ausgewählte Bild auf dem Profil an.	✓
3	Der Benutzer klickt auf "Delete Image".	Das System zeigt das Bild nicht mehr an.	✓
4	Der Benutzer gibt neue Werte in den Eingabeflächen zu Nutzernname, Vorname, Nachname und Email ein und klickt auf "Speichern".	Das System zeigt auf der Seite die aktualisierten Daten des Nutzers an.	✓
Nachbedingung(en)	Nutzer hat sein Profil aktualisiert.		
Testurteil	Test bestanden		

Datum	11.5.2023		
Tester			
SW-Version	1		
Vorbedingung(en)	Der Benutzer ist ein Systemadministrator und im System angemeldet. Der Benutzer befindet sich auf einer Tabellenansicht.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Benutzer kann neue Datensätze einspeisen: Er klickt auf Durchsuchen, wählt eine passende Daten-Datei aus und klickt auf Upload .	Der Nutzer erhält eine aktuelle Ansicht der Daten: Die eingegebenen Daten werden in einer Tabelle angezeigt.	<input checked="" type="checkbox"/>
2	Der Benutzer wählt per ID-Eingabe bestimmte Daten aus und klickt auf Delete Entry.	Der Nutzer erhält eine aktuelle Ansicht der Daten: Die gelöschten Daten werden nicht mehr in der Tabelle angezeigt.	<input checked="" type="checkbox"/>
Nachbedingung(en)	Die von Nutzer geänderten Daten wurden im System gespeichert. Der Benutzer ist weiterhin angemeldet und befindet sich auf der Tabellenansichtsseite.		
Testurteil	Test bestanden		

Datum	12.5.2023		
Tester			
SW-Version	1		
Vorbedingung(en)	Der Benutzer ist auf der Seite DataView.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Benutzer sucht nach einem Datensatz.	Das System filtert die Datensätze nach der Suche.	<input checked="" type="checkbox"/>
2	Der Benutzer wählt einen Datensatz aus.	Das System verlinkt den Nutzer auf den Datensatz.	<input checked="" type="checkbox"/>
Nachbedingung(en)	Der Benutzer ist auf der Seite des Datensatzes		

Testurteil	Test bestanden
-------------------	-----------------------

Datum	11.5.2023		
Tester			
SW-Version	1		
Vorbedin-gung(en)	Der Benutzer ist auf der Seite eines Datensatzes.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Nutzer sucht nach einem Eintrag.	Das System filtert die Einträge nach der Suche.	<input checked="" type="checkbox"/>
Nachbedingung(en)	Der Benutzer ist auf der Seite des Datensatzes.		
Testurteil	Test bestanden		

Datum	12.5.2023		
Tester			
SW-Version	1		
Vorbedin-gung(en)	Der Benutzer muss sein Profil angelegt haben und befindet sich auf der Seite /userlist.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Benutzer sucht nach anderen Nutzern.	Es werden alle angelegten Nutzer angezeigt.	<input checked="" type="checkbox"/>
Nachbedingung(en)	Systemdaten unverändert, Benutzer befindet sich auf Seite /userlist.		
Testurteil	Test bestanden		

Datum	11.5.2023		
Tester			
SW-Version	1		
Vorbedingung(en)	Der Benutzer muss eingeloggt sein und befindet sich auf der Seite /dataview.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer klickt neben einem Datensatz auf den Herzbutton.	Der Datensatz ist zu den Lieblingsdatensätzen des Nutzers hinzugefügt und wird angezeigt bei der Profilansicht.	✓
2	Der Nutzer klickt auf der Editprofileseite seines Profils bei der Tabelle seiner Lieblingsdatensätze auf den entfernen Knopf.	Der Button ändert seine Zustand	✓
Nachbedingung(en)	Der Datensatz wurde aus den Lieblingsdatensätzen des Nutzers entfernt.		
Testurteil	Test bestanden		

Zyklus II

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
8.	Datenvisualisierung			
	User Story 8.1	User Story		fertig
	User Story 8.2	User Story		fertig
	Datensatz mit Diagrammansicht	Papierprototyp		fertig
9.	Karte Aachener Wald			
	User Story 9.1	User Story		fertig
	User Story 9.2	User Story		fertig
	User Story 9.3	User Story		fertig
	KarteGeoDaten	Papierprototyp		fertig
10.	Support			
	User Story 10.1	User Story		fertig
	User Story 10.2	User Story		fertig
	User Story 10.3	User Story		fertig
	User Story 10.4	User Story		fertig
	User Story 10.5	User Story		fertig
	UserTicketAnsicht	Papierprototyp		fertig
	AdminTicketAnsicht	Papierprototyp		fertig
	TicketAnsicht	Papierprototyp		fertig
	TicketErstellung	Papierprototyp		fertig
11.	Freunde			
	User Story 11.1	User Story		fertig
	User Story 11.2	User Story		fertig
	User Story 11.3	User Story		fertig
	User Story 11.4	User Story		fertig
	User Story 11.5	User Story		fertig
	User Story 11.6	User Story		fertig
	ProfilansichtFreundeEnable d	Papierprototyp		fertig
	ProfilansichtFreundeDisable d	Papierprototyp		fertig
	eigeneProfilansichtFreund eslisteBearbeiten	Papierprototyp		fertig
12.	Chat			
	User Story 12.1	User Story		fertig
	User Story 12.2	User Story		fertig

	User Story 12.3	User Story		fertig
	ChatUndAnfragen	Papierprototyp		fertig
13.	Sonstiges			
	Hamburgermenu2.0	Papierprototyp		fertig

User-Stories

User Story-ID	8.1
User Story-Beschreibung	Ich als Nutzer möchte mir die Datensätze visuell als Diagramme aufbereiten lassen können, um eine bessere Übersicht zu haben.
Geschätzter Realisierungsaufwand	> 2 Tage
Priorität	niedrig
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	-

User Story-ID	8.2
User Story-Beschreibung	Ich als Nutzer möchte mir die Datensätze visuell als Graph aufbereiten lassen können, um eine bessere Übersicht zu haben.
Geschätzter Realisierungsaufwand	> 2 Tage
Priorität	[REDACTED]
Autor	mittel
Abhängigkeiten zu anderen User Stories	

User Story-ID	9.1
User Story-Beschreibung	Ich als User möchte Zugriff auf eine Karte vom Aachener Wald haben, um eine geographische Einordnung der Region zu haben.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	

User Story-ID	9.2
User Story-Beschreibung	Als Admin möchte ich Geodatensätze zu Schutzhütten, Rettungspunkte und Knotenpunkte im Aachener Wald einspeisen können, damit die Karte auf dem aktuellen Stand ist.
Geschätzter Realisierungsaufwand	> 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	9.1

User Story-ID	9.3
User Story-Beschreibung	Als Nutzer möchte ich, dass verschiedene Daten auf der Karte mit anderen Symbolen dargestellt werden, damit ich die Geodaten unterscheiden kann.

Geschätzter Realisierungsaufwand	>= 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	9.1

User Story-ID	10.1
User Story-Beschreibung	Als Nutzer möchte ich eine Inbox in meinem Hamburgermenu haben, um zu neuen empfangenen Nachrichten zu navigieren.
Geschätzter Realisierungsaufwand	<= 2 Tage
Priorität	niedrig
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	

User Story-ID	10.2
User Story-Beschreibung	Als Admin möchte ich eine Inbox in meinem Hamburgermenu haben, um neue Supporttickets einsehen zu können.
Geschätzter Realisierungsaufwand	<= 2 Tage
Priorität	niedrig
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	10.3

User Story-ID	10.3
User Story-Beschreibung	Als Nutzer möchte ich Supporttickets senden können, um Systemadministrator über Fehler auf den Datensätzen zu informieren.
Geschätzter Realisierungsaufwand	> 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	

User Story-ID	10.4
User Story-Beschreibung	Als Nutzer möchte ich eine Nachricht per E-Mail erhalten, wenn mein Supportticket bearbeitet wurde, damit ich darüber informiert bin.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	10.3, 10.4

User Story-ID	10.5
User Story-Beschreibung	Als Admin möchte ich per Klick in einem Supportticket eine Nachricht an den Nutzer senden können, dass dieses Supportticket bearbeitet wurde, um den Nutzer darüber in Kenntnis zu setzen.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	10.3

User Story-ID	11.1
User Story-Beschreibung	Als Nutzer möchte ich Freundschaftsanfragen versenden können, um einen anderen Nutzer meiner Freundesliste hinzuzufügen.
Geschätzter Realisierungsaufwand	> 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	

User Story-ID	11.2
User Story-Beschreibung	Als Nutzer möchte ich erhaltene Freundschaftsanfragen ablehnen können, um den jeweiligen Nutzer nicht zu meiner Freundesliste hinzuzufügen.
Geschätzter Realisierungsaufwand	<= 2 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	11.1, 11.3

User Story-ID	11.3
User Story-Beschreibung	Als Nutzer möchte ich erhaltene Freundschaftsanfragen annehmen können, um den jeweiligen Nutzer zu meiner Freundesliste hinzuzufügen.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	11.1, 11.2

User Story-ID	11.4
User Story-Beschreibung	Als Nutzer möchte ich die Möglichkeit haben, meine Freundesliste auf öffentlich einzustellen, um diese auf meinem Profil anzuzeigen.
Geschätzter Realisierungsaufwand	ca. 1 Tag
Priorität	mittel
Autor	[REDACTED]

Abhängigkeiten zu anderen User Stories	11.5
---	------

User Story-ID	11.5
User Story-Beschreibung	Als Nutzer möchte ich die Möglichkeit haben, meine Freundesliste auf privat einzustellen, um diese auf meinem Profil zu verbergen.
Geschätzter Realisierungsaufwand	ca. 1 Tag
Priorität	mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	11.4

User Story-ID	11.6
User Story-Beschreibung	Als Nutzer möchte ich eine Nachricht per E-Mail erhalten, wenn ich eine Freundschaftsanfrage empfange, um zeitnah darüber informiert zu sein.
Geschätzter Realisierungsaufwand	ca. 2 Tage
Priorität	mittel
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	

User Story-ID	12.1
User Story-Beschreibung	Als Nutzer möchte ich einen Chat mit einem oder mehreren anderen Nutzern erstellen können, um Nachrichten auszutauschen.
Geschätzter Realisierungsaufwand	> 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	12.2

User Story-ID	12.2
User Story-Beschreibung	Als Nutzer möchte ich Nachrichten per Chat versenden können, um anderen Nutzern meine Nachricht mitzuteilen.
Geschätzter Realisierungsaufwand	> 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	12.1

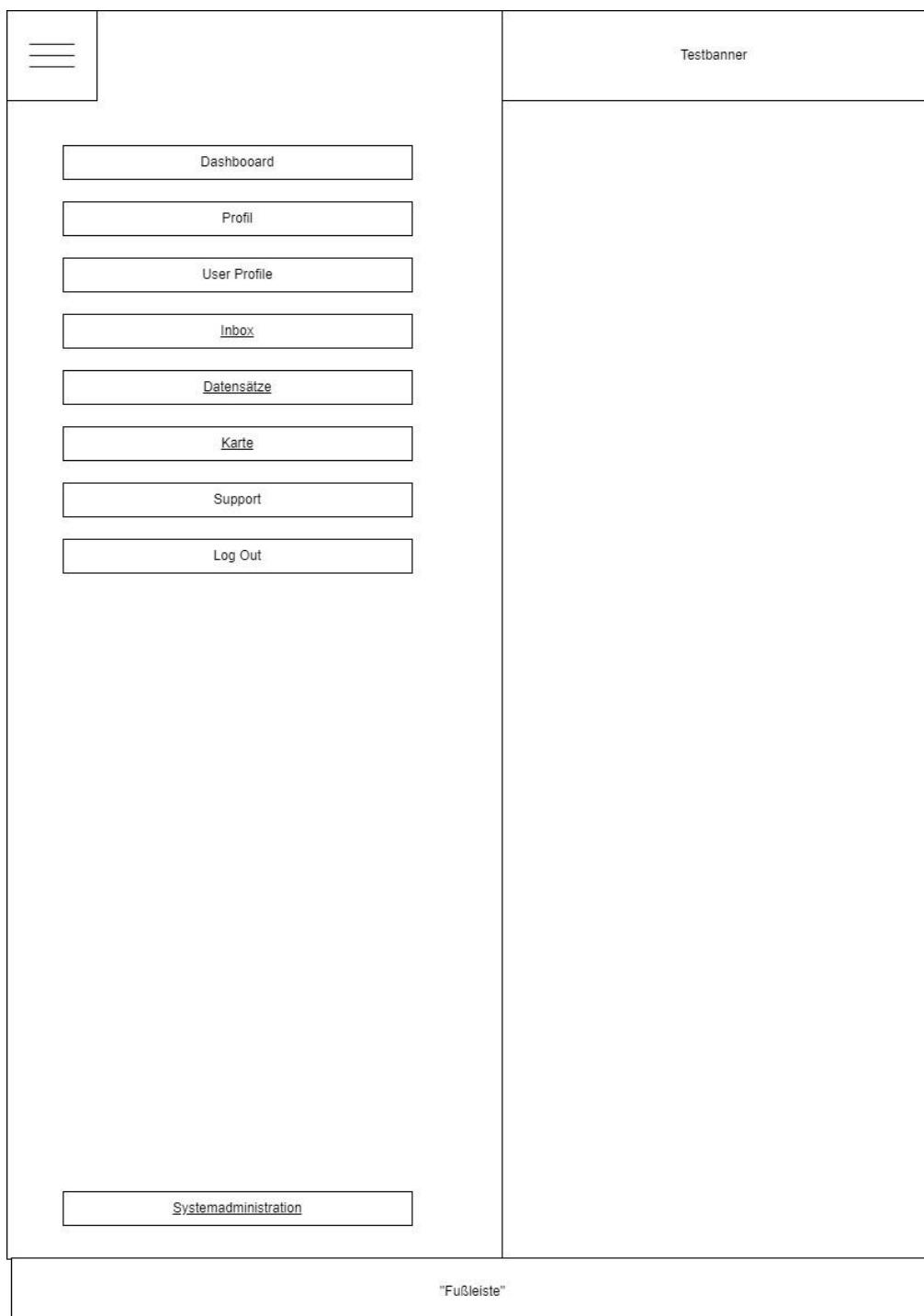
User Story-ID	12.3
User Story-Beschreibung	Als Nutzer möchte ich Chatnachrichten bearbeiten/löschen können, um eventuelle Fehler zu korrigieren, bevor die Nachricht gelesen wurde.

Geschätzter Realisierungsaufwand	> 3 Tage
Priorität	hoch
Autor	[REDACTED]
Abhängigkeiten zu anderen User Stories	12.1, 12.2

Papierprototypen

(weiterhin gilt: Anführungszeichen = nicht-klickbarer Text, runde Kästen mit Pfeil = nur für Papierprototyp eingefügte Notiz/Info zu dieser Stelle, eckige Kästen = Buttons, eckige Kästen mit Pfeil = aufgehende Fenster beim Anklicken dieser Stelle)

Hamburgmenü2.0:



Profilansicht (andere Nutzer) erweitern zu:

- Profilansicht Freunde Enabled

The screenshot shows the 'I ❤ DATA' application interface. On the left, there is a user profile card for 'admin'. The profile picture is a blue placeholder icon. The profile information includes: Vorname: Admin, Nachname: Admin, Geburtsdatum: (empty), and Email: admin@mail.com. Below the profile information is a button labeled 'Freundschaftsanfrage senden'. To the right of the profile card is a sidebar titled 'I ❤ DATA' at the top. The sidebar contains a section titled 'Freundesliste' which lists six entries, each consisting of a small blue user icon and the text 'Vorname Nachname'. To the right of the sidebar is another section titled 'Lieblingsdatensätze von admin' which lists several data series: 'Vornamen Aachen', 'Sterbefälle Aachen', 'Anzahl der Arbeitssuchenden in der Städteregion Aachen', 'Anzahl der Arbeitslosen in der Städteregion Aachen', 'Mittlere Jahresbevölkerung nach Geschlecht - kreisfreieStädte und Kreise - Jahr (ab 1975)', 'Privathaushalte- kreisfreie Städte und Kreise - Jahr', 'Straßenliste Aachen', and 'Geburten der Stadt Aachen nach Monat, 2015-2022'.

Universität Duisburg-Essen, SEP SS2023 | Gruppe D | I ❤ DATA

- Profilansicht Freunde Disabled

The screenshot shows the 'I ❤ DATA' application interface. The layout is identical to the previous one, featuring a user profile card for 'admin' on the left and a sidebar on the right. However, the 'Freundesliste' section in the sidebar is now empty, replaced by a large black lock icon. The rest of the sidebar content remains the same, listing the same data series as before.

Universität Duisburg-Essen, SEP SS2023 | Gruppe D | I ❤ DATA

eigene Profilansicht ändern zu: eigene Profilansicht Freundesliste Bearbeiten

I ❤ DATA



Profil von admin

Vorname: Admin
Nachname: Admin
Geburtsdatum:
Email: admin@mail.com

[Edit Profile](#)

Freundesliste

	Vorname Nachname	

Lieblingsdatensätze von admin

Vornamen Aachen
Sterbefälle Aachen
Anzahl der Arbeitssuchenden in der Städteregion Aachen
Anzahl der Arbeitslosen in der Städteregion Aachen
Mittlere Jahresbevölkerung nach Geschlecht - kreisfreie Städte und Kreise - Jahr (ab 1975)
Private Haushalte- kreisfreie Städte und Kreise - Jahr
Straßenliste Aachen
Geburten der Stadt Aachen nach Monat, 2015-2022

Universität Duisburg-Essen, SEP SS2023 | Gruppe D | I ❤ DATA

Karte Geo-Daten:

I ❤ DATA

1  "Schutzhütten" einblenden/
ausblenden

2  "Knotenpunkte" einblenden/
ausblenden

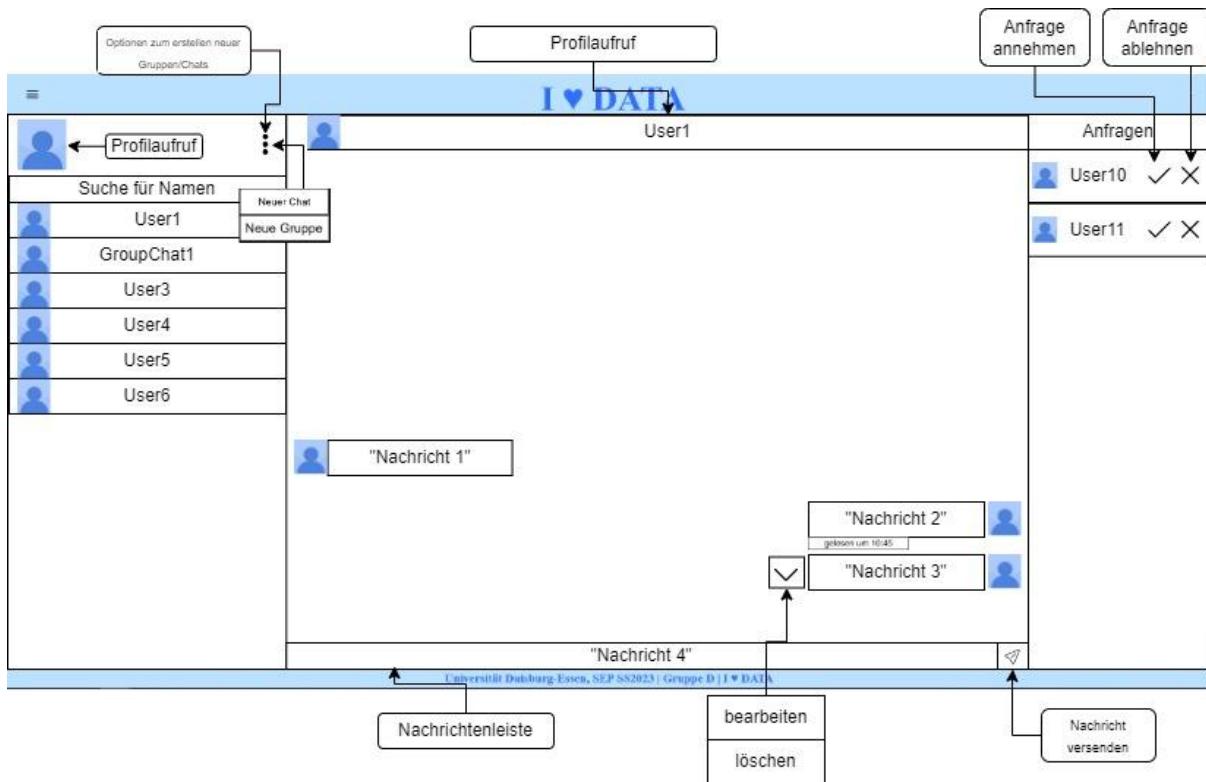
3  "Rettungspunkte" einblenden/
ausblenden

Karte Aachener Wald

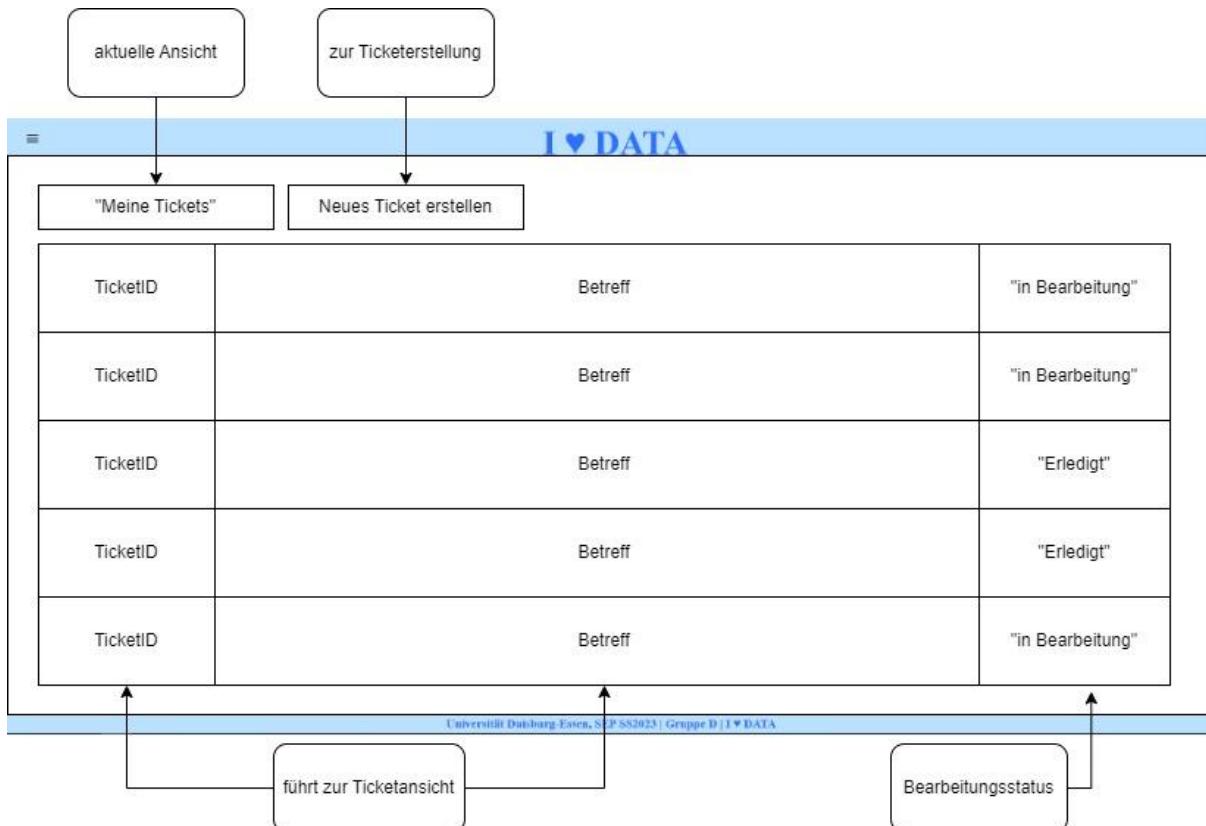


Universität Duisburg-Essen, SEP SS2023 | Gruppe D | I ❤ DATA

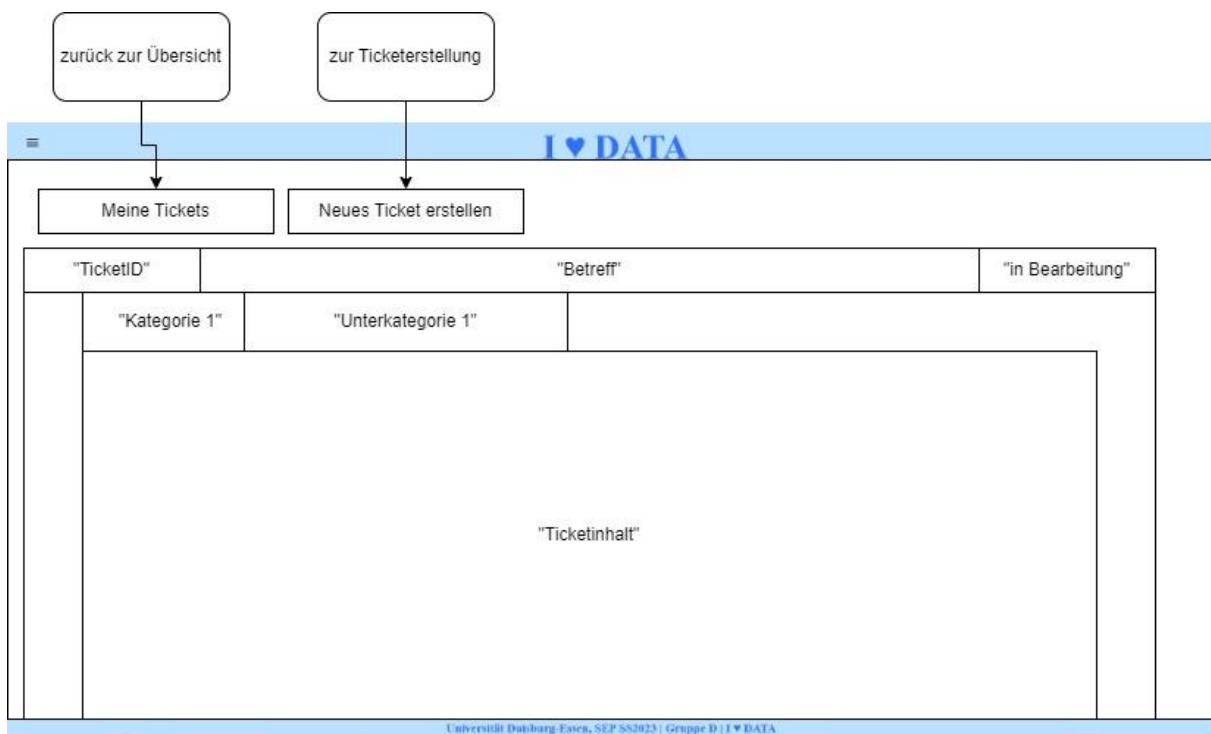
Chats und Anfragen:



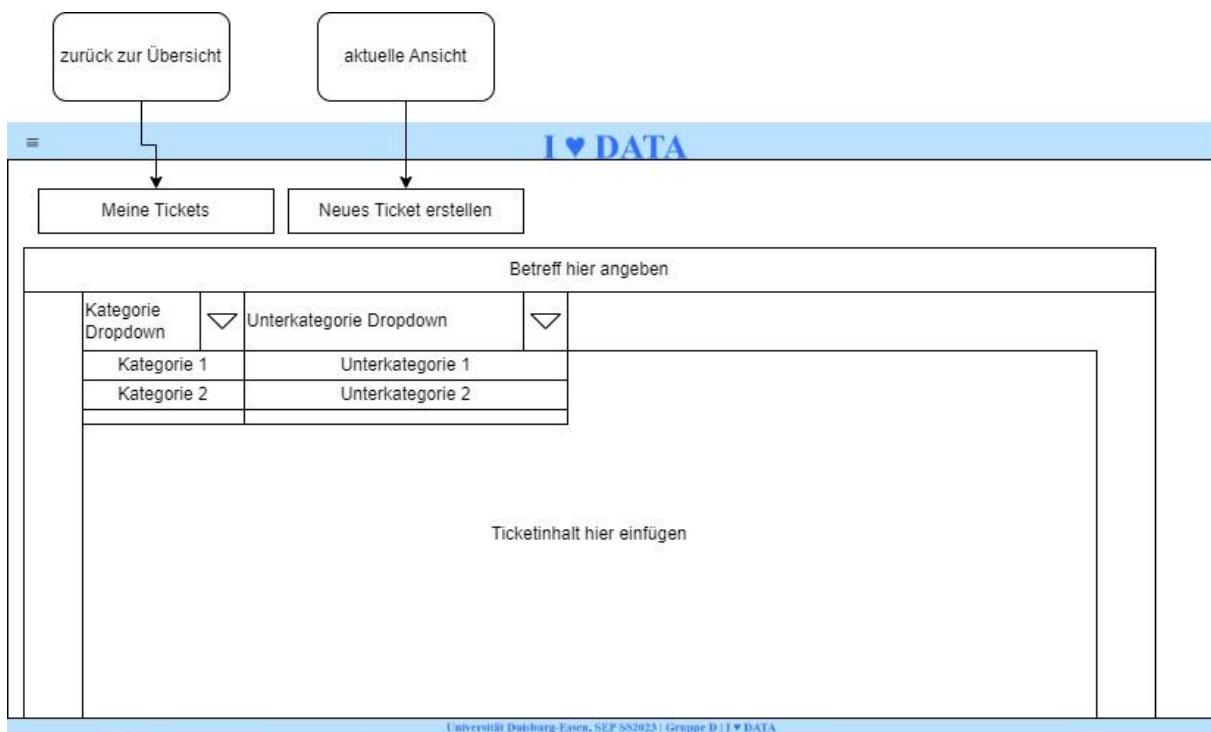
User Ticket Übersicht:



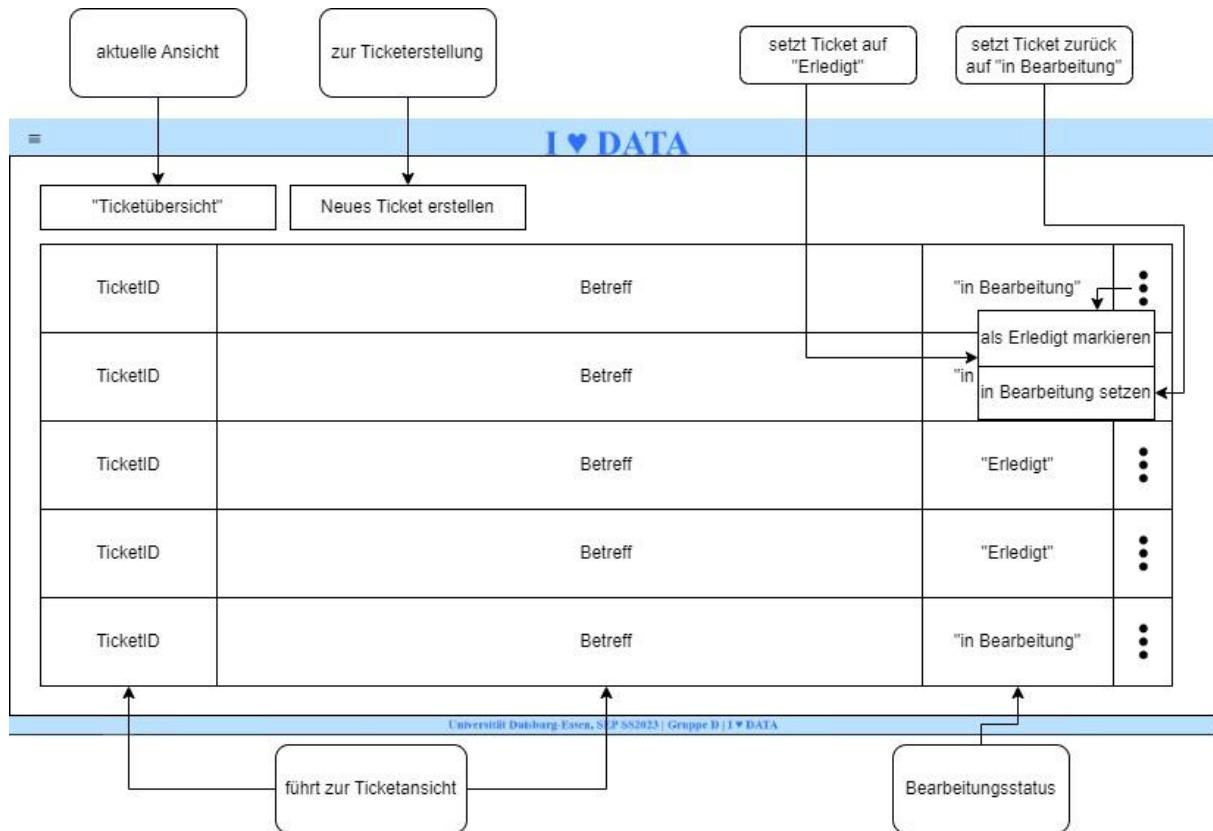
TicketAnsicht:



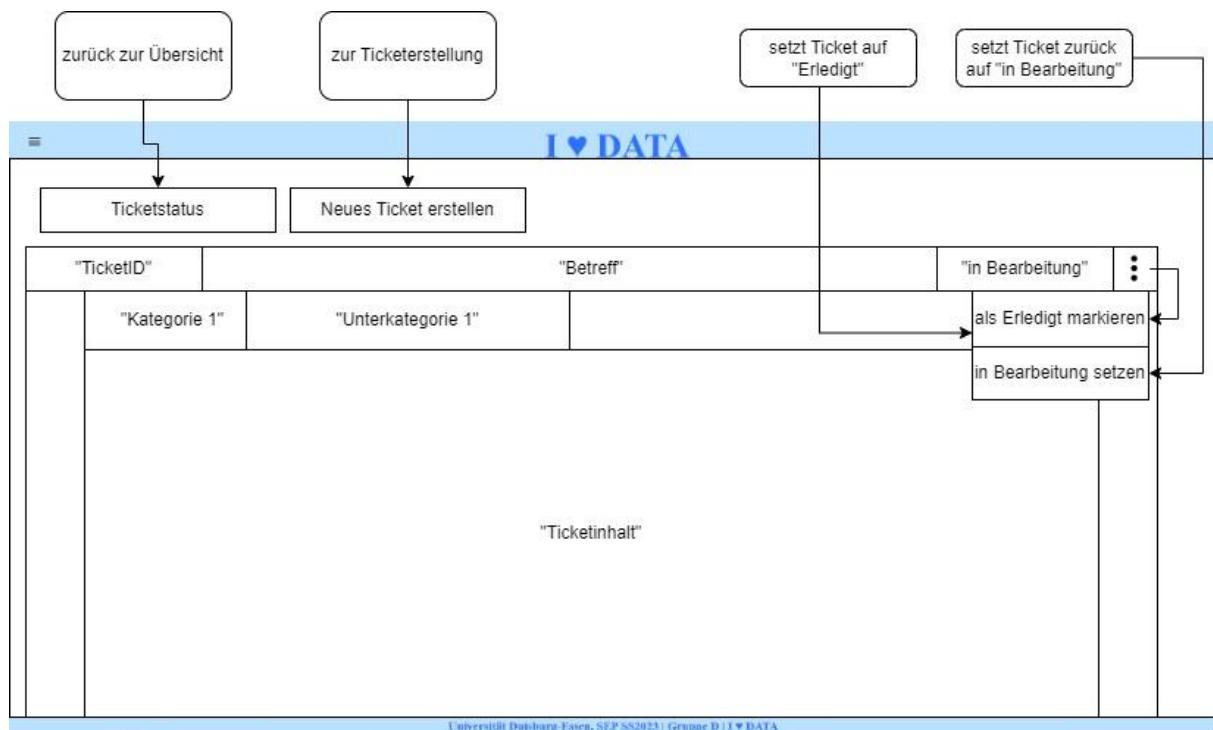
TicketErstellung:



Admin Ticket Übersicht:



Admin Ticket Ansicht:



Zugeklappt: Diagramm/Graph Ansicht:

Diagramm/Graph Ansicht (Zugeklappt):

Vornamen Aachen

Search

ID	Anzahl	Vorname	Geschlecht	Position
				Items per page: 25 0 of 0 < > >>

Universität Duisburg-Essen, SEP SS2023 | Gruppe D | I ❤ DATA

Aufgeklappt: Diagramm in Datensatz-Ansicht:

Diagramm in Datensatz-Ansicht (Aufgeklappt):

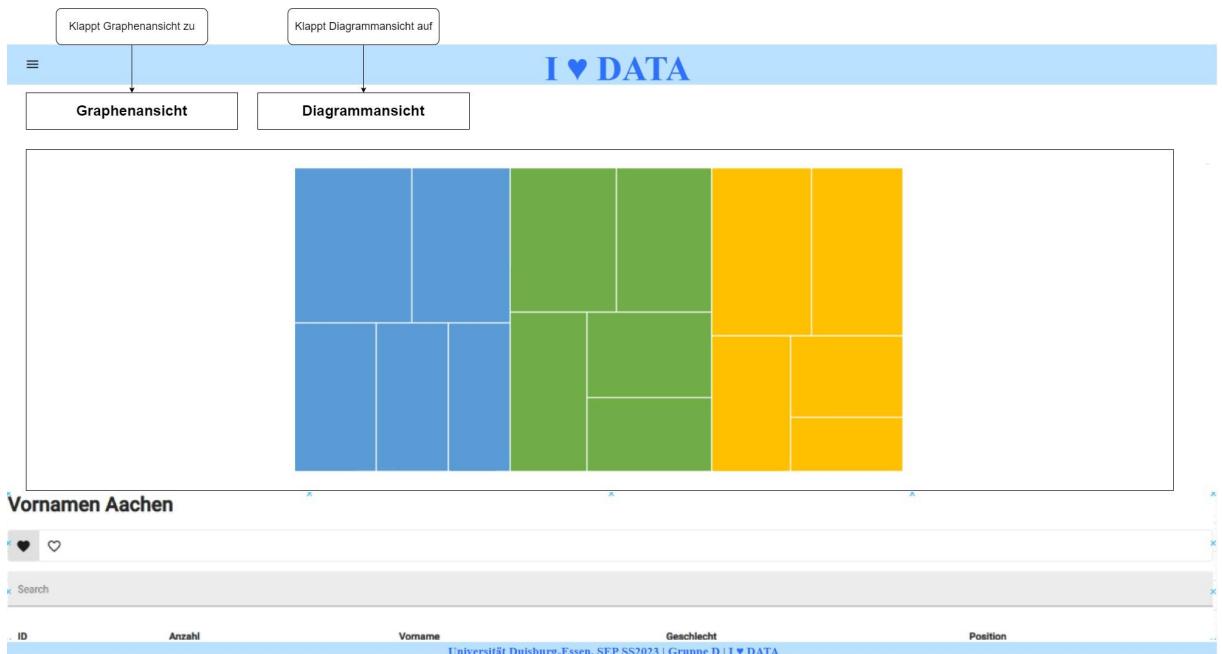
ID	Anzahl	Vorname	Geschlecht	Position
Produkt 4	133			
Produkt 3	133			
Produkt 7	261			
Produkt 8	312			
Produkt 5	318			
Produkt 6	345			
Produkt 1	351			
Produkt 9	367			
Produkt 10	379			
Produkt 2	389			

Vornamen Aachen

Search

Universität Duisburg-Essen, SEP SS2023 | Gruppe D | I ❤ DATA

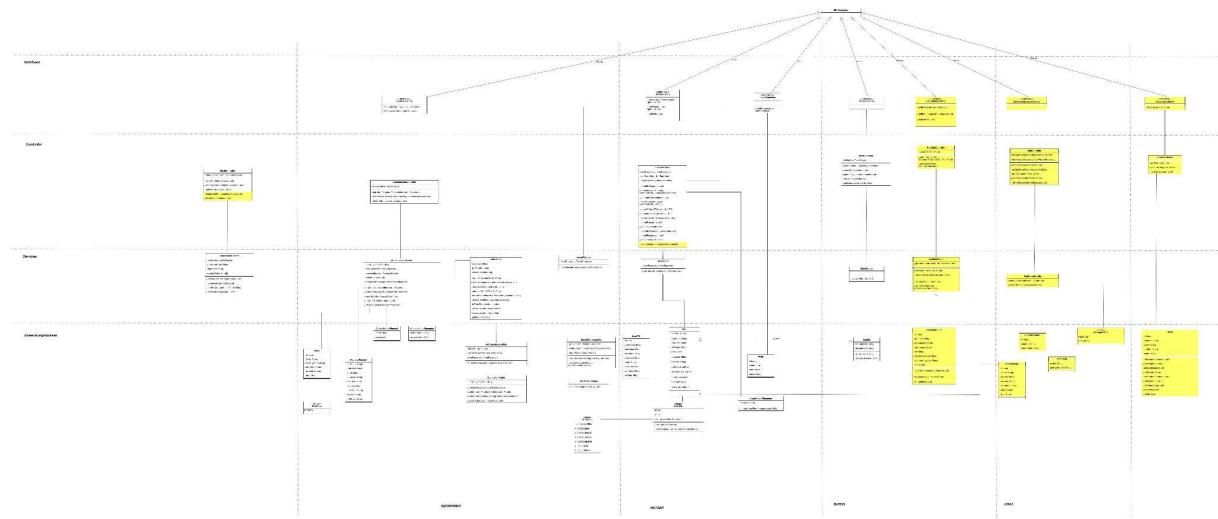
Aufgeklappt: Graphenansicht in Datensatz-Ansicht:



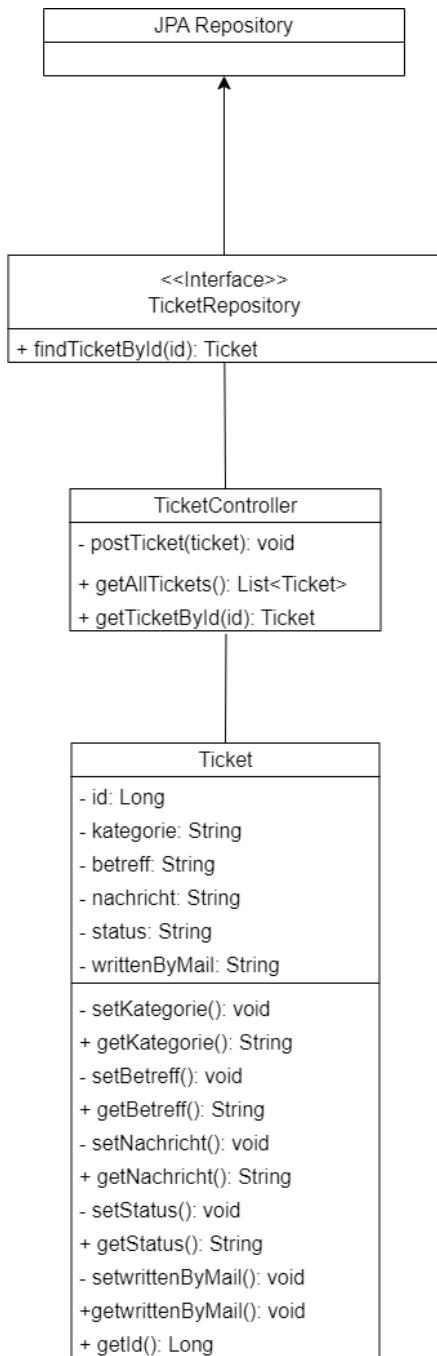
Strukturdiagramme (Komponenten- und Klassendiagramme)

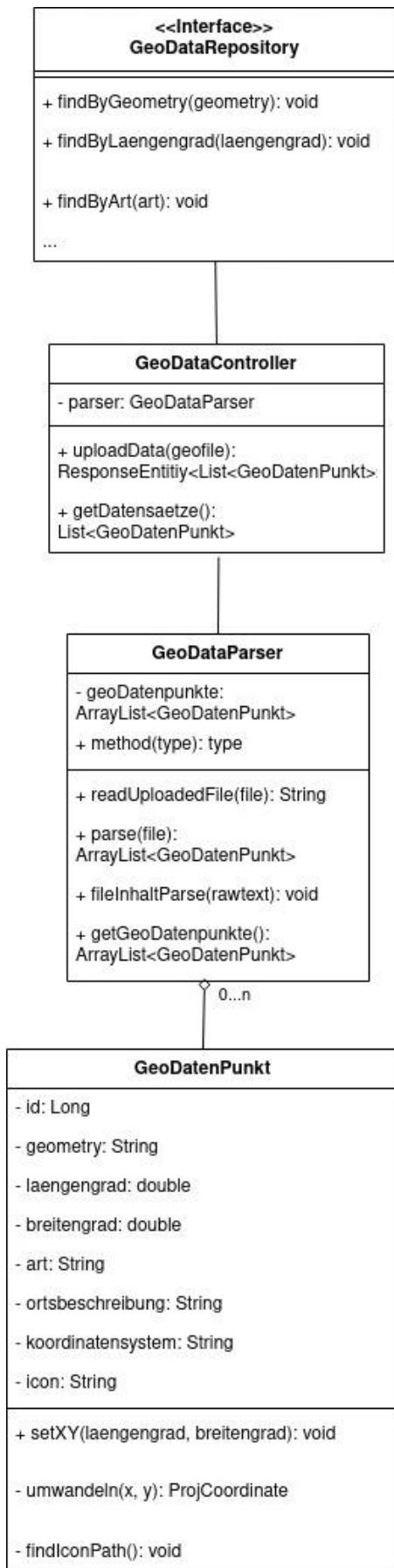
Klassendiagramm

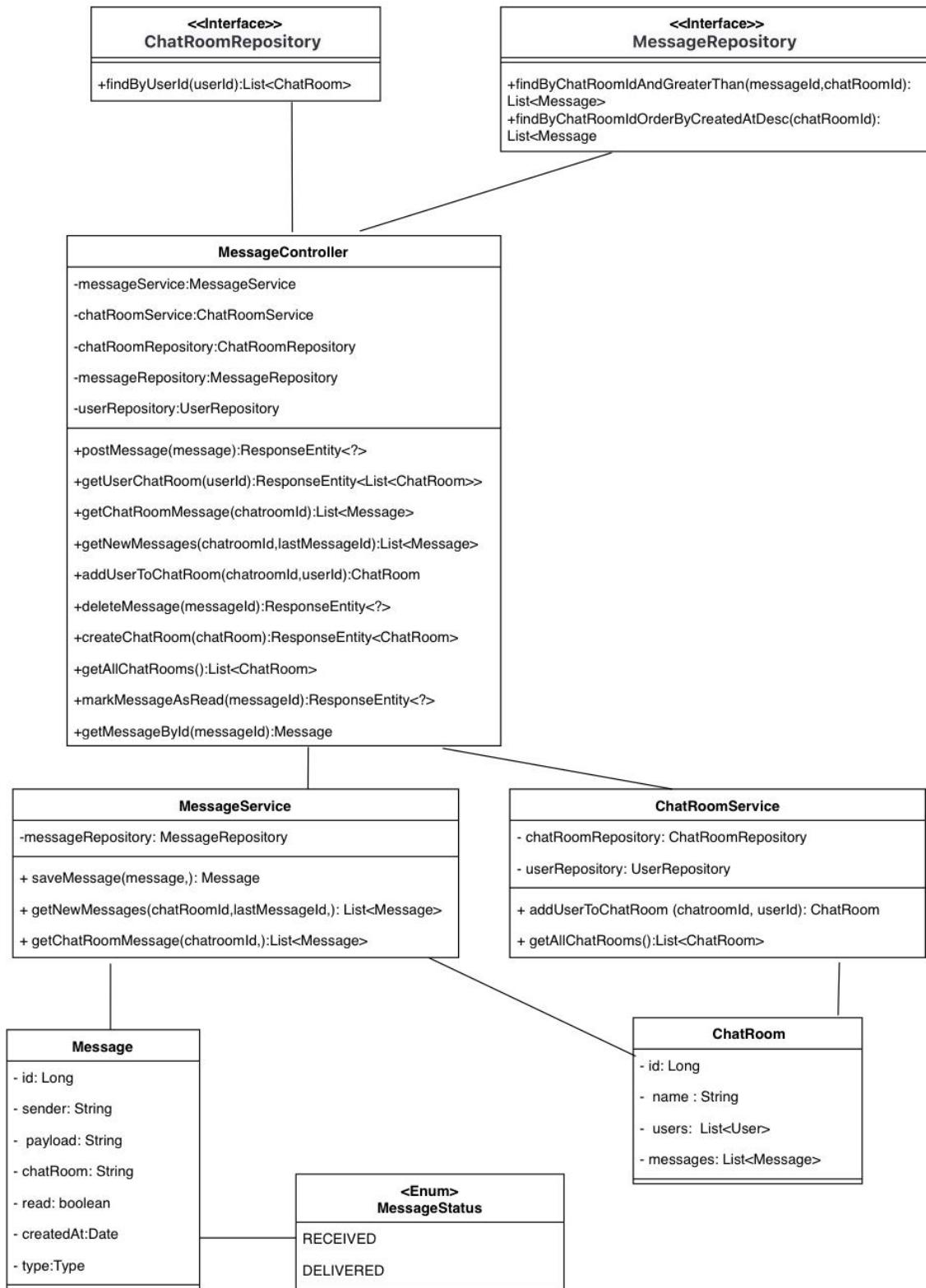
(gelb markiert sind die Änderungen des 2. Zyklus)



→ Ausschnitte der neuen Bereiche:



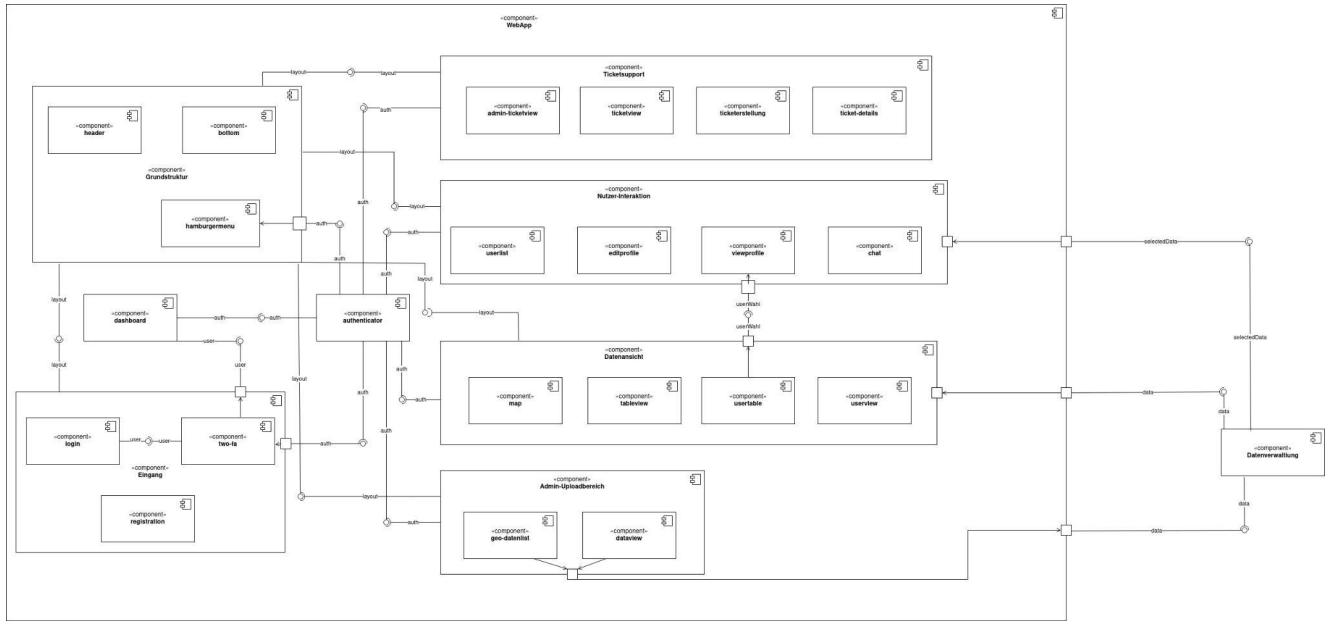




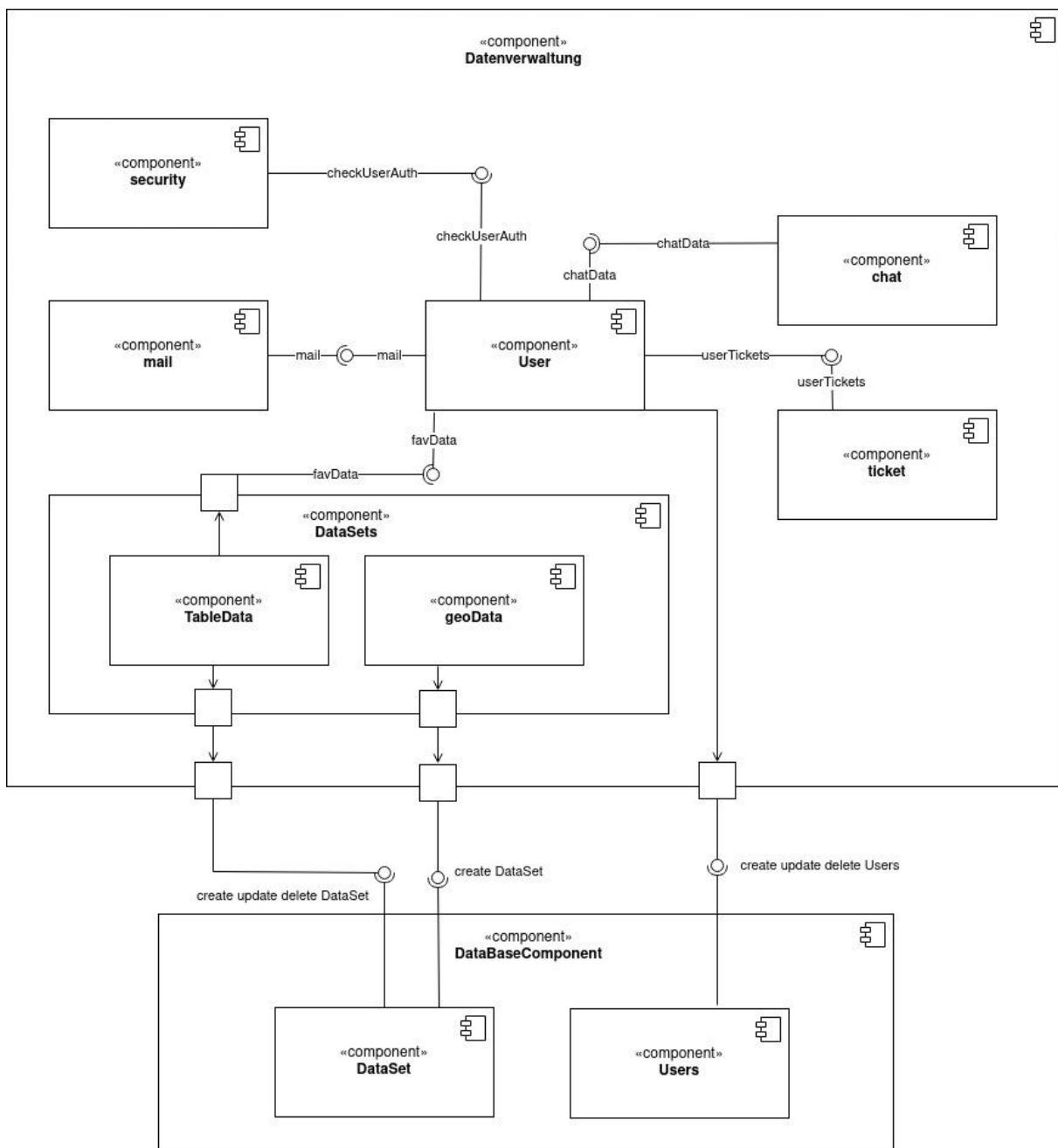
Komponentendiagramme

frontend (mit Link zum Backend):

(Der Übersicht halber wurden Abhängigkeiten von Komponenten nicht gezeichnet, wenn diese für alle Komponenten einer Überkomponente benötigt werden. Daher wird in diesen Fällen auch keine Schnittstelle per an der entsprechenden Überkomponentengrenze verwendet.)

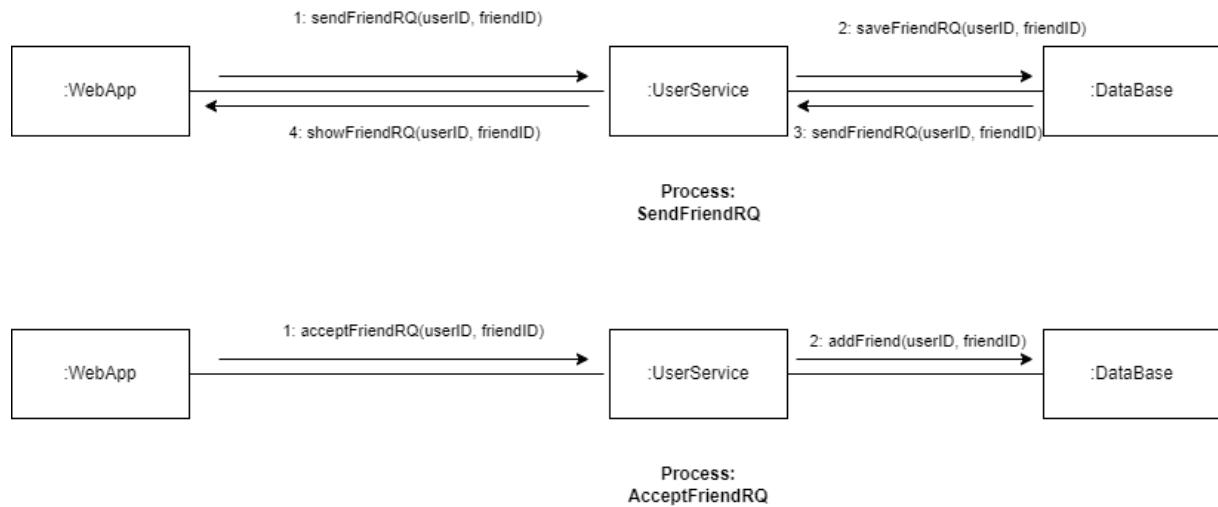


backend:

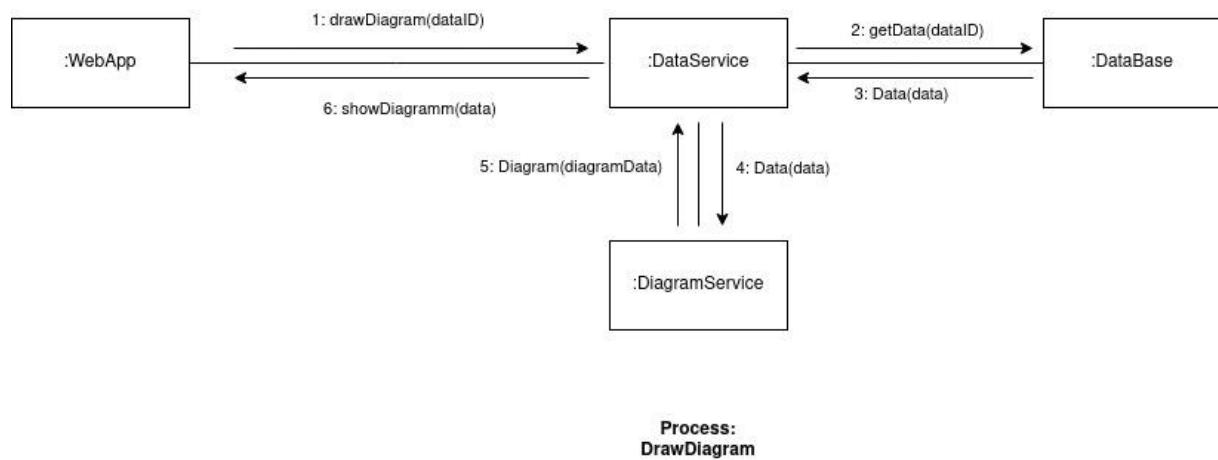


Verhaltensdiagramme (Kommunikationsdiagramme)

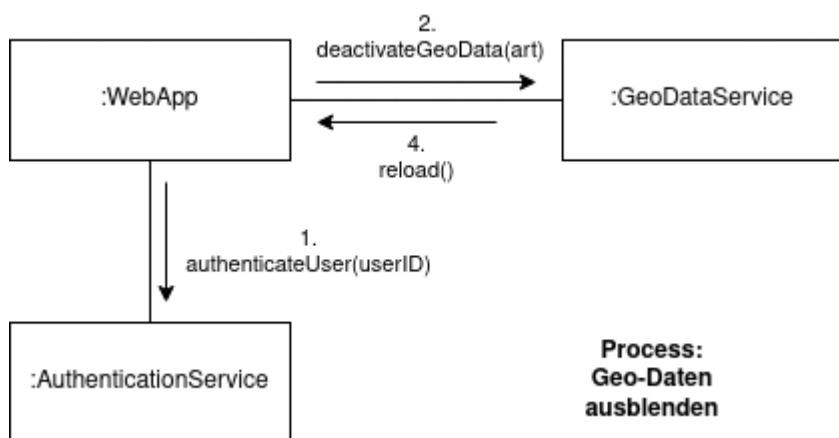
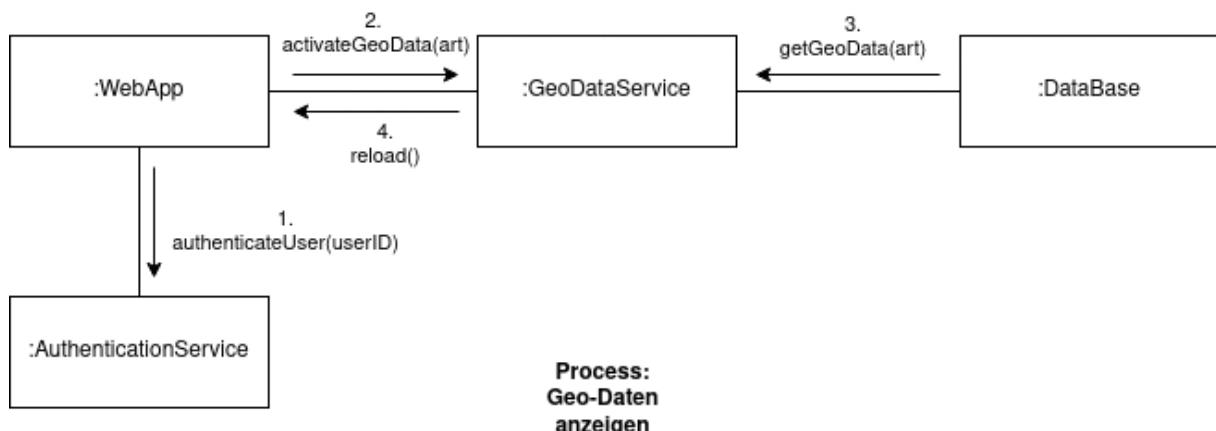
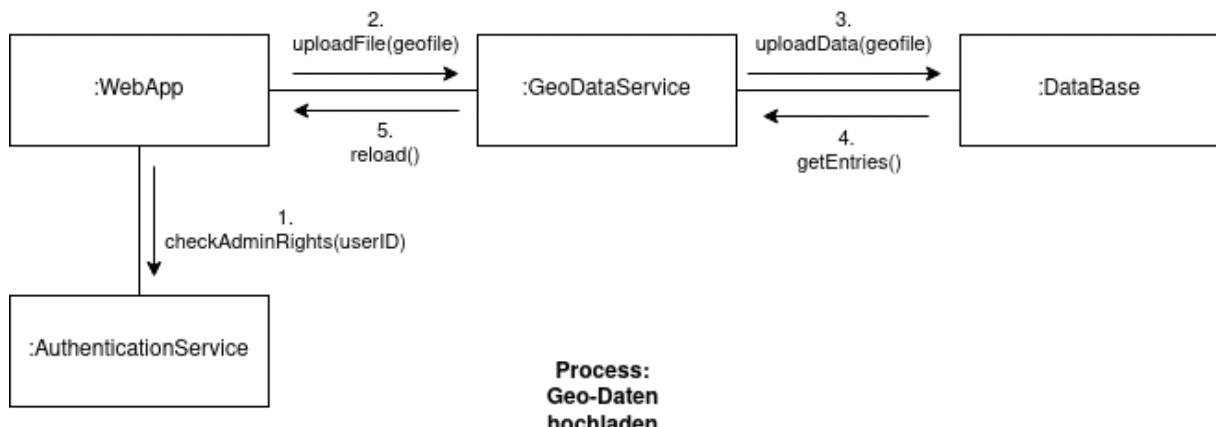
2. Freundesliste:



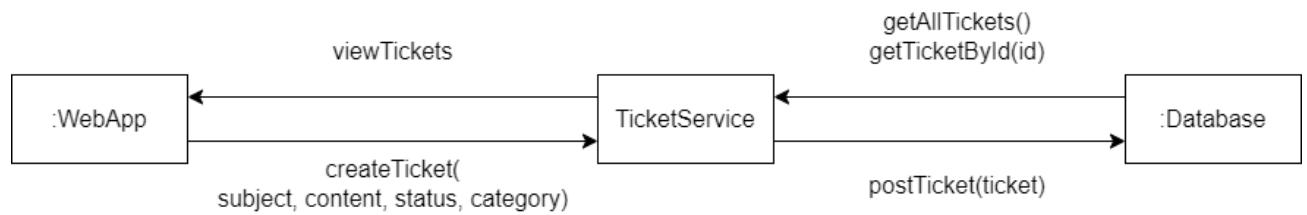
3. Aufbereitung der Datensätze:



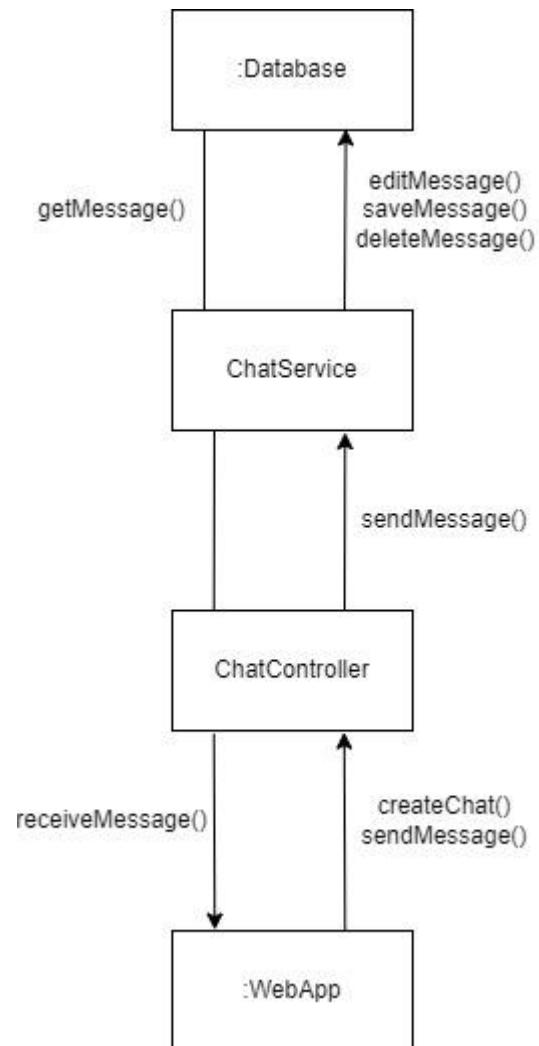
4. Geodaten:



5. Supportticket:



6. Chat:



Funktionalitätsplanung

(Da die Nummerierung der Funktionalitäten je Zyklus neu startet (im Gegensatz zu den User Stories!) werden Bezüge zum Zyklus I per I.x.x dargestellt.)

ID	Funktionalität	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
1.	Nacharbeitung					
1.1	Registrations-Checks aktualisieren (Auth)		I.1.1	1.1	registration-component.ts	fertig
1.2.1	Header anklickbar: Link zur Startseite		I.0.1		hamburgermenu.html, hamburgermenu.css, hamburgermenu.ts	fertig
2.	Freundesliste					
2.1	Sicherung: Nutzer-Änderungszugriff nur bei eigenem Profil			11.1 - 11.6	viewprofile.component(.ts,.css,.html,.specs.ts) editprofile.component(.ts,.css,.html,.specs.ts)	fertig
2.2	Freundesliste (backend)		2.3	11.1-11.3, 11.6	User.java UserController.java	fertig
2.3	Freundesliste in Userprofil einbinden (frontend)		2.4, I.3.4		viewprofile.component(.ts,.css,.html,.specs.ts)	fertig
2.4	Freundesanfrage		2.3, 2.2		viewprofile.component(.ts,.css,.html,.specs.ts) inbox.component(.ts,.css,.html,.specs.ts)	fertig
3.	Aufbereitung der Datensätze					
3.1	usertable: Einbau Diagrammoption		I.4.1	8.1, 8.2	usertable(1,2,3,4,8).ts, usertable(1,2,3,4,8).html, diagrammodel.ts,	fertig

					barchart.ts, piechart.ts, treemap.ts,	
4.	Geodaten					
4.1	Geodatenverknüpfung (& weitere Adminzugänge) im hamburgermenü	5.4		9.1, 9.2, 9.3	hamburger-menue.html: mat-expansion-panel	fertig
4.2	geojson einlesen = Parser				GeoDataService.java, GeoDatenPunkt.java, GeoDataRepository.java, GeoDataController.java	fertig
4.3	json-Daten in Tabelle (nicht sichtbar) speichern = Adminview	4.2	9.2		iconArt.ts, geoDatenPkt.ts; geo-datenlist.component (.ts, .css, .html, .spec.ts), geo-data.service(.spec).ts	fertig
4.4	Daten in Landkarte (versch. Symbole) einbauen (= frontend)	4.2	9.1, 9.3		map.component (.ts, .css, .html, .spec.ts)	fertig
5.	Supportticket					
5.1	Liste aller offenen Tickets für Admins	5.4	10.1, 10.2, 10.3, 10.4, 10.5		Ticketview.com ponent(.ts, .html, .css)	fertig
5.2	E-Mail Bestätigung (gespeicherter Nutzer)	1.2.2	10.4		EmailService.ts, EmailSenderService.java, EmailController.java	fertig
5.3	view: Anlegen eines Tickets	5.4	10.1, 10.2, 10.3, 10.4, 10.5		ticket-erstellung.component(.ts, .html, .css)	fertig
5.4	view Admin: bearbeiten eines Tickets	5.1, 5.3			ticket-details.component(.ts, .html, .css)	fertig
6.	Chat					

6.1	einbinden in Systemstruktur: view Chats erstellen (frontend)			12.1,12.2	chat.ts, chat-list.ts, chatbox.ts, chat-headerbox.ts	fertig
6.2	view Chat		6.1	12.1,12.2	MessageController.java, chat.ts, chat-list.ts, chatbox.ts, chat-headerbox.ts	fertig
6.3	Speichern der Nachrichten		6.2	12.2	MessageController.java	fertig
6.4	Nachrichten löschen & bearbeiten		6.2, 6.3	12.2,12.3	MessageController.java, chat.ts, chat-list.ts, chatbox.ts, chat-headerbox.ts, message.ts	fertig
6.5	Kontrolle, ob Nachricht gelesen wurde		6.2, 4.3, 6.4	12.2,12.3	MessageController.java,message.ts, chatbox.ts	fertig
6.7	Mitglieder hinzufügen (Gruppenchats)		6.1	12.1,12.2	MessageController.java, chat-headerbox.ts, chatlist.ts	fertig
7.	Modultests					
7.1	Modultest Geodata		4.	-	GeoDataTest.java	fertig
7.2	grober Modultest User		I.3.4	-	UserControllerTest.java	fertig
7.3	Modultest Supporttickets		5.	-	ticket-erstellung.component.spec.ts	fertig
7.4	Modultest Registration		I.1.1	1.1	registration.component.spec.ts	fertig
8.	Sonstiges					
8.1	ggf. Dockerfiles überarbeiten (Speicherlimit)		-	-	.gitlab-ci.yml, karma.conf.js	fertig

Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status
1	GeoDataController, GeoDataParser: upload = einlesen der Geodaten, Fehlerabfangen bei fehlender Koordinatensysteminfo	GeoDataTest.java	fertig
2	UserController/Repository: Nutzer anlegen, Nutzer löschen, Nutzerdaten ändern (! nicht mehr Funktionalitäten)	UserControllerTest.java	fertig
3	ticket-erstellung.component: Ticket anlegen, Warnung bei Eingabe leeren Tickets, Navigation zu Ticketview	ticket-erstellung.component.spec.ts	fertig
4	Registration.component.ts: Registrationseingaben-Prüfung = entsprechende Fehlermeldungen bei falschen Eingaben	registration.component.spec.ts	fertig

Systemtests

Parallelgruppe = Gruppe C

2. Freundesliste:

Datum	15.06.2023		
Tester	[REDACTED]		
SW-Version	2		
Vorbedingung(en)	Mindestens zwei User existieren im System, der User ist eingeloggt		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der User ruft das Profil eines anderen Users auf	Das System zeigt die Profilseite des anderen Users an	✓
2	Der User klickt auf die Schaltfläche "Freundschaftsanfrage senden"	Das System schickt eine Freundschaftsanfrage	✓
3			

Nachbedingung(en)	In der Inbox des anderen Users liegt eine Freundschaftsanfrage	
Testurteil	Test bestanden	

Datum	15.06.2023		
Tester			
SW-Version	2		
Vorbedingung(en)	Der User hat eine Freundschaftsanfrage offen		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der User ruft seine Inbox auf	Die offenen Freundschaftsanfragen werden angezeigt	<input checked="" type="checkbox"/>
2	Der User akzeptiert die Freundschaftsanfrage	Das System löscht die Anfrage und fügt beide User gegenseitig in die Freundeslisten ein	<input checked="" type="checkbox"/>
Nachbedingung(en)	Die User sind in der Freundesliste gespeichert		
Testurteil	Test bestanden		

Datum	15.06.2023		
Tester			
SW-Version	2		
Vorbedingung(en)	Der User hat eine Freundschaftsanfrage offen		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der User ruft seine Inbox auf	Die offenen Freundschaftsanfragen werden angezeigt.	<input checked="" type="checkbox"/>

2	Der User lehnt die Freundschaftsanfrage ab	Das System löscht die Freundschaftsanfrage	✓
Nachbedingung(en)	Die Freundschaftsanfrage ist gelöscht und die Inbox ist leer		
Testurteil	Test bestanden		

Datum	15.06.2023		
Tester	[REDACTED]		
SW-Version	2		
Vorbedingung(en)	Der User hat einen anderen User in seiner Freundesliste		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der User öffnet seine Profileditier Seite auf	Das System zeigt die Profileditier Seite an.	✓
2	Der User klickt auf das Löschesymbol neben dem Freund.	Das System löscht den Freund aus der Freundesliste.	✓
Nachbedingung(en)	Der Freund ist nicht mehr in der Freundesliste.		
Testurteil	Test bestanden		

3. Aufbereitung der Datensätze:

Datum	15.06.2023		
Tester	[REDACTED]		
SW-Version	2		
Vorbedingung(en)	Der User ist eingeloggt, Datensatz "Geburten Aachen ist eingespeist."		

Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der User klickt im Hamburgermenü auf "Datensätze".	Das System zeigt die Liste der Datensätze.	✓
2	Der User klickt auf den Datensatz "Geburten der Stadt Aachen nach Monate, 2015-2022".	Das System zeigt die Seite des Datensatz "Geburten Aachen".	✓
3	Der User klickt auf das Dropdownmenü "Wähle ein Jahr" und wählt das Jahr 2017 aus.	Das Jahr 2017 ist ausgewählt.	✓
4	Der User klickt auf den Knopf "Kuchendiagramm zeichnen".	Das System zeigt in einem Pop-Up-Fenster ein Kuchendiagramm zum Datensatz Geburten Aachen 2017.	✓
Nachbedingung(en)	Das Kuchendiagramm wird angezeigt.		
Testurteil	Test bestanden		

4. Geodaten:

Datum	15.06.2023		
Tester	[REDACTED]		
SW-Version	2		
Vorbedingung(en)	Der Nutzer ist eingeloggt, besitzt Admin-Rechte und befindet sich auf der Seite /geo (im Hamburgermenü unter "Adminbereich" unter "Geodaten einlesen"). Der Nutzer hat einen der 3 Geodaten-Sätze als Json-Datei zur Verfügung.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer wählt über "Datei auswählen" die Json-Geo-Datei aus und klickt auf Upload.	Ein Popup-Fenster öffnet sich mit der Meldung "File uploaded successfully."	✓

2	Der Nutzer klickt auf "ok".	Die Seite lädt neu und zeigt die Geodaten nun in der unten stehenden Tabelle an. Im Infobereich wird die Liste um die in der Json-Datei vorhandenen Typen von Geodaten erweitert.	√
Nachbedingung(en)	Die Geodaten sind im System gespeichert. Der Nutzer befindet sich auf der Seite /geo.		
Testurteil	Test bestanden		

Datum	15.06.2023		
Tester			
SW-Version	2		
Vorbedingung(en)	Der Nutzer ist angemeldet und es wurden von einem Admin zuvor Geodaten mindestens eines Typen eingelesen.		
Schritt	Aktion (User)	Erwartete Reaktion (System)	√ / X
1	Der Nutzer klickt auf das Hamburgermenüzeichen (≡) und in der aufgehenden Auswahlleiste auf "Landkarte".	Der Nutzer wird zur Seite /map weitergeleitet, auf der eine Landkarte im Bereich von Aachen zu sehen ist. Es gibt Buttons für die versch. Geodatentypen, von denen diejenigen aktiviert (nicht ausgegraut) sind, zu denen Daten im System vorliegen.	
2	Der Nutzer klickt auf "alle Geodaten anzeigen".	Es erscheinen Icons auf der Karte und deren Bereich wird angepasst, sodass alle Icons sichtbar sind. Die Buttons der einzelnen Geodatentypen springen um von "... anzeigen" auf "...ausblenden".	
3	Der Nutzer klickt auf "alle Geodaten ausblenden".	Die Icons in der Karte verschwinden. Die Buttons der einzelnen Geodatentypen springen zurück auf "... anzeigen".	
Nachbedingung(en)	Der Nutzer ist weiterhin eingeloggt im System und befindet sich auf der Seite /map.		
Testurteil	Test bestanden		

6. Chat:

Datum	15.06.2023		
Tester			
SW-Version	2		
Vorbedingung(en)	Der Nutzer erstellt einen neuen User über einen anderen Browser		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Der Nutzer klickt auf den linken button	Der Nutzer kann den gewünschten User für seinen privaten Chat auswählen	✓
2	Der Nutzer kann seine Nachricht eingeben	Die Nachricht wird im Chat angezeigt	✓
Nachbedingung(en)	Der Nutzer hat einen privaten Chat		
Testurteil	Test bestanden		

Datum	15.06.2023		
Tester			
SW-Version	2		
Vorbedingung(en)	Wenn die Nachricht nicht gelesen wurden		
Schritt	Aktion (User)	Erwartete Reaktion (System)	✓ / X
1	Die Nachricht wird mit dem Buttons delete angezeigt	Der Nutzer kann seine Nachricht löschen	✓
2	Die Nachricht wird mit dem Button update angezeigt	Der Nutzer kann seine Nachricht updaten	✓

Nachbedingung(en)	Der Nutzer kann seine Nachrichten löschen und updaten	
Testurteil	Test bestanden	

Datum	15.06.2023		
Tester			
SW-Version	2		
Vorbedingung(en)	Der Nutzer drückt auf den rechten Button		
Schritt	Aktion (User)	Erwartete Reaktion (System)	<input checked="" type="checkbox"/> / <input type="checkbox"/>
1	Der Nutzer gibt seinen gewünschten Gruppennamen ein und fügt die User ein	Der Gruppenchat wird angezeigt	
Nachbedingung(en)	Der Gruppenchat wird angezeigt		
Testurteil	Test bestanden		

Zyklus III

Verbesserungsideen:

- Registrierung = eingeloggt, nicht zu Login
 - einheitliche Sprache (dt/engl)
 - Docker in pipeline
 - ungenutzte Imports?
-
- Komponenten in Ordnern zusammenfassen? z.B. login + registrierung + two-fa
 - branches durchgehen und veraltete löschen

Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
...				
...				
...				

User-Stories

Template:

User Story-ID	
User Story-Beschreibung	
Geschätzter Realisierungsaufwand	
Priorität	
Autor	
Abhängigkeiten zu anderen User Stories	
Zugehörige Szenarien	

Papierprototypen

Strukturdiagramme (Komponenten- und Klassendiagramme)

Verhaltensdiagramme (Kommunikationsdiagramme)

Funktionalitätsplanung

ID	Funktionalität	Verantwortlich er	Abhängige Funktionalitä ten	Verknüpfte User-Stories	Quellcode-referenz	Status
1.						
1.1						
1.2						
1.2. 1					
2.						
2.1						
8.2	Controller zusammenfüg en/umstruktur ieren	[REDACTED]		6.1, 6.2, 6.3		

Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status

Systemtests

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

Nutzerhandbuch

Technische Anforderungen

Es muss ein funktionsfähiger Webbrowser vorhanden sein (keine Internetverbindung).

Es muss docker installiert sein. Die nötigen Berechtigungen zum Starten von docker Images und docker volumes müssen vorhanden sein.

Installationsanleitung

In einem Ordner müssen die Dateien docker_image_frontend.tar, docker_image_backend.tar und die docker-compose.yml vorliegen. Dieser Ordner wird im Folgenden als Hauptordner bezeichnet.

Die Docker-Images müssen geladen werden. Dies geschieht per Kommandozeile im Hauptordner und Eingabe von:

```
docker load -i docker_image_frontend.tar  
docker load -i docker_image_backend.tar
```

Danach kann per Kommandozeilenbefehl

```
docker compose up
```

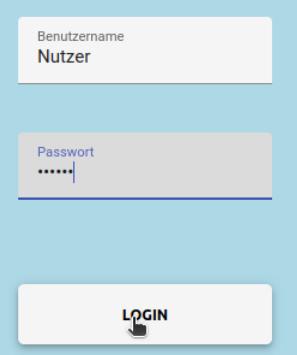
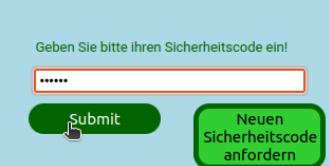
im Hauptordner die Anwendung gestartet werden.

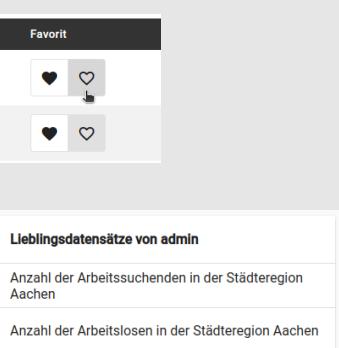
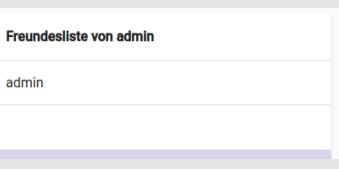
Die Seite kann daraufhin in einem beliebigen Internetbrowser unter localhost:4200/ aufgerufen werden.

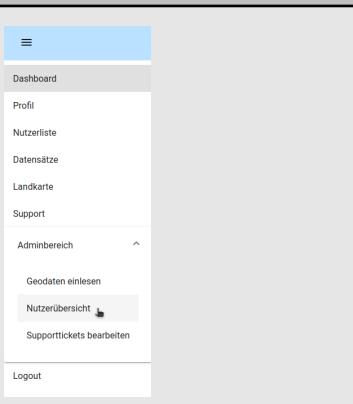
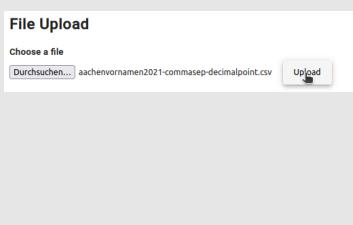
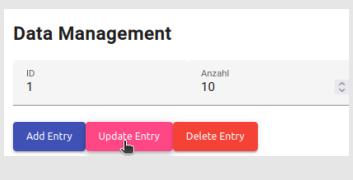
Bedienungsanleitung

(Die Farbgestaltung oder das Design kann je nach Softwareversion variieren.)

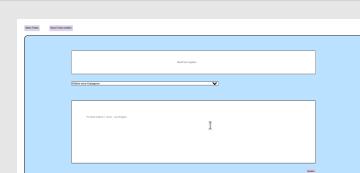
Start:

	<p>Die Startseite von I-love-Data stellt den Login-Bereich dar. Bei bereits im System angemeldeten Nutzer kann sich über den Nutzernamen und das Passwort angemeldet werden.</p>
	<p>Über die Startseite von I-Love-Data kann per Klick auf den Registrieren-Button zu einer Registrationsseite gewechselt werden. Es kann sich registriert werden, wozu die persönlichen Daten eingetragen werden müssen und ein Nutzernname gewählt wird. Wenn später Rechte zum Ändern der Daten und Datensätze erwünscht sind, muss ein Haken bei Systemadministrator gesetzt werden.</p>
	<p>Nach der Registration wird zur Bestätigung eine E-Mail erhalten, in der auch ein Anmeldecode für das Einloggen im System enthalten ist. Es kann sich allerdings auch erst später im System angemeldet werden, indem er die Login-/Startseite öffnet und seinen Nutzernamen und sein Passwort einträgt und dies per Klick auf den Login-Button bestätigt.</p> <p>Für die endgültige Anmeldung im System muss der Code aus der erhaltenen E-Mail eingegeben werden. Wenn dieser Code verloren ist, kann per Klick auf "neuen Code anfordern" ein neuer Validierungscode generiert werden.</p>
interne Struktur:	
	<p>Im System steht ein Dashboard mit aktuellen Informationen zur Verfügung.</p> <p>Um zu anderen Seiten zu gelangen, kann oben links per Hamburger-Menü die gewünschte Seite ausgewählt werden. Der Logout erfolgt über das Hamburger-Menü.</p>
	<p>Es gibt eine Liste aller Nutzer und eine Liste der Datensätze.</p> <p>In der Nutzerliste kann per Klick auf "Profil anzeigen" das persönliche Profil anderer Nutzer angezeigt werden.</p>

 <p>Suche nach Datensätzen... Suchen</p> <p>Datensatz Favorit</p> <p>Vornamen Aachen</p> <p>Sterbefälle Aachen</p>	<p>In der Datensatzliste kann über eine Suchfeldeingabe gesucht werden. Per Klick auf entsprechende Listeneinträge kann auf Seiten von Datensätzen gewechselt werden. Die Datensätze können per Klick auf den Herzbutton zu den Lieblingsdatensätzen hinzugefügt oder daraus entfernt werden.</p>
 <p>alle Geo-Daten ausblenden</p> <p>Schutzhütten anzeigen Rettungspunkte anzeigen Knotenpunkte anzeigen</p>	<p>Es steht eine Karte des Aachener Waldes zur Verfügung. Für vorhandene Daten gibt es in 3 Kategorien Aktivierungs- und Deaktivierungs-Buttons der Icons.</p>
<p>Nutzerbereich:</p>	
 <p>Profil bearbeiten</p> <p>Username Nutzer</p> <p>Vorname Max</p> <p>Nachname Mustermann</p> <p>E-Mail max@muster.de</p> <p>Speichern Abbrechen</p>	<p>Die eigenen Profildaten können verändert werden. Nach Eingabe der korrigierten Daten muss auf Speichern geklickt werden.</p> <p>Bei falschen Eingaben muss der Abbrechen-Button geklickt werden.</p>
 <p>Durchsuchen... Keine Datei ausgewählt.</p> <p>Upload Image</p>	<p>Im Profilbereich kann ein Profilbild hinzugefügt werden.</p>
 <p>Favorit</p> <p>Lieblingsdatensätze von admin</p> <p>Anzahl der Arbeitssuchenden in der Städteregion Aachen</p> <p>Anzahl der Arbeitslosen in der Städteregion Aachen</p>	<p>Es können durch Anklicken des Herz-Buttons Datensätze im eigenen Profil hinterlegt werden.</p> <p>Diese erscheinen dann im eigenen Profil in der Lieblingsdatensatzliste.</p>
 <p>Freundesliste von admin</p> <p>admin</p>	<p>Im Profilbereich befindet sich die Freundesliste.</p>

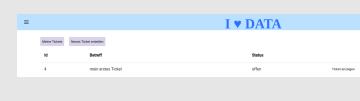
	Die eigene Freundesliste kann im Profilbearbeitungsmodus auf öffentlich oder privat eingestellt werden.
	Anderen Nutzern kann über deren Profilansicht eine Freundschaftsanfrage gesendet werden. Diese erscheint daraufhin in der Inbox des Nutzers.
	Die Freundschaftsanfragen an einen selbst finden sich unter /inbox und können per einfachem Klick angenommen oder abgelehnt werden.
Datenbereich (für Admins):	
	Für Admins gibt es einen eigenen Bereich, der auch im Hamburgermenü als solches benannt ist. Hierüber können Steuerelemente wie der Datenupload und die Benutzerverwaltung erreicht werden.
	Auf der Seite eines Datensatzes (dataview) befindet sich für Administratoren die Option, eine Datei hochzuladen. Hier sind entsprechende csv- bzw. xml-Dateien möglich. Wenn das Format der gewählten Datei zur Art der angegebenen Tabelle (s. Kopfzeile) passt, werden die Werte in einer Tabelle angezeigt.
	Über eine Suchleiste können Werte in der Tabelle gesucht werden. Über die Eingabe einer ID können Administratoren Datensatztabellen per Add-, Update-, Delete von Einträgen verändern.
	Auf der Adminseite der Geodaten (/geo) befindet sich die Option, Geodaten im Geo-Json-Format hochzuladen. Bereits vorhandene Geodaten können in der Liste "Info" entnommen werden.

Supporttickets:

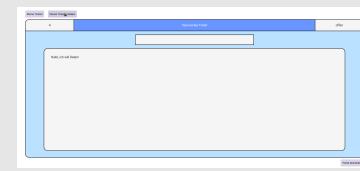


Bei Problemen und Anfragen können - unter Angabe von Kategorie, Betreff und Inhalt - Supporttickets erstellt werden. Informationen über den Status des Tickets werden über die Ticketansicht oder über die Ticketliste erhalten.

Für Nutzer ohne Adminrechte erhält ein angelegtes Ticket zuerst immer den Status "offen". Bei Admins hingegen wird der Status auf "in Bearbeitung" gesetzt.

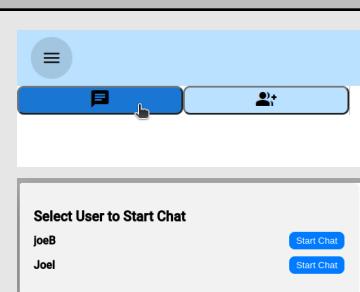


Über einen Klick auf "meine Tickets" werden die eigenen Tickets aufgelistet.



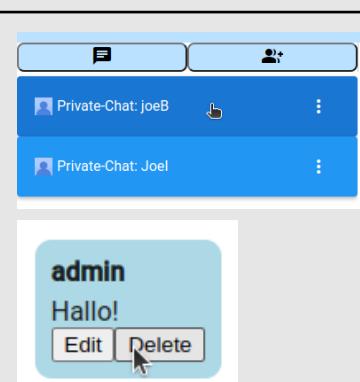
Die Ticketansicht ist für Admins zugreifbar per "Ticket anzeigen", dadurch wird der Status des Tickets auf "in Bearbeitung" gesetzt. Zurück kann per "meine Tickets" oder "Neues Ticket erstellen" navigiert werden. In der Ticketansicht kann der Status des Tickets von "in Bearbeitung" auf "Erledigt" gesetzt.

Chat:



Im Chatbereich (/chat) können persönliche Chats über den linken Button oben gestartet werden.

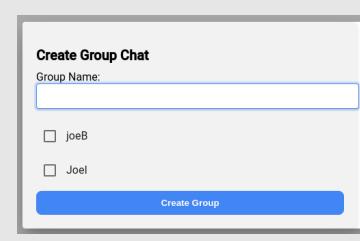
Im sich öffnenden Popup-Fenster muss der gewünschte Gesprächspartner angeklickt werden. Es erscheint ein neuer Chat in der Liste links und es muss die Popupbox durch einen Klick daneben geschlossen werden.



Chats müssen per Klick auf sie geöffnet werden.

Dann können Nachrichten gesendet werden. Eigene Nachrichten werden am rechten Rand der Seite angezeigt, Nachrichten des Gesprächspartners erscheinen am linken Rand des Chatbereiches.

Nachrichten lassen sich löschen oder bearbeiten, solange der andere Gesprächspartner sie noch nicht gesehen hat.



Über den oberen rechten Button lassen sich Gruppenchats anlegen. Auch hier muss gewählt werden, welchen anderen Nutzer dies betreffen soll.

