

Hochschule Aalen
Fakultät Informatik
Sommersemester 2021
Übungsgruppe 4

Software Engineering

Übung 2

Entwurf

Maximilian Borst – 80497@studmail.hs-aalen.de
Jan Kerner – 79671@studmail.hs-aalen.de
Simon Ruttmann – 80751@studmail.hs-aalen.de
Veronika Scheller – 79888@studmail.hs-aalen.de
Michael Ulrich – 77607@studmail.hs-aalen.de

19. Juni 2025

Inhaltsverzeichnis

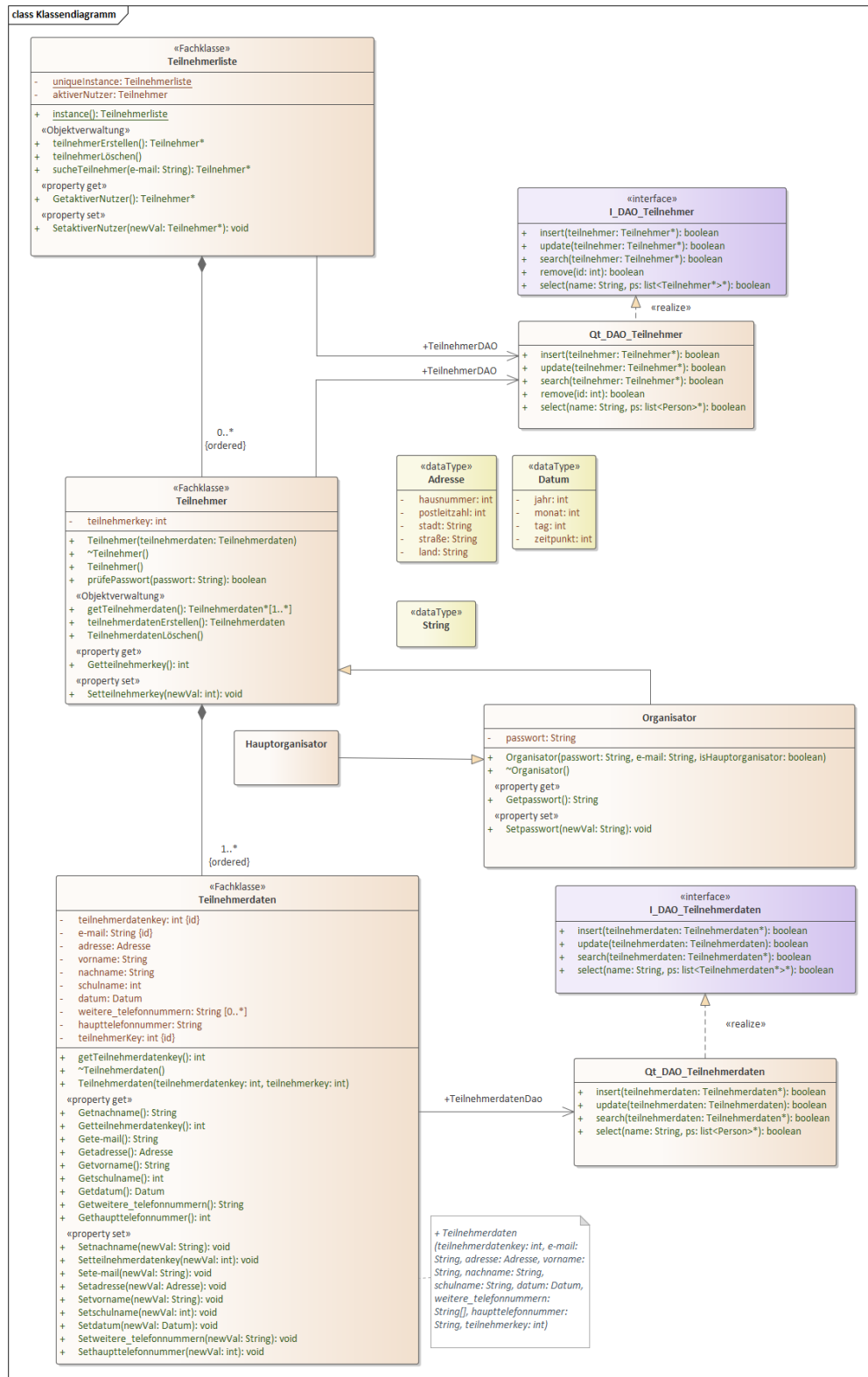
1	Beschreibung der Systemarchitektur (Grobstruktur)	3
2	UML-Klassendiagramm(e) als Entwurfsmodell für die im System zu verwaltenden Daten	4
3	Datenbankschema (create Table Anweisungen)	5
3.1	Teilnehmer	5
3.2	Telefonnummer	5
3.3	Organisator	6
4	Entwurf der graphischen Benutzeroberfläche mit Beschreibung ihrer Funktionen	6
4.1	Startbildschirm - Erstes Starten des Programms	7
4.2	Neues System erstellen	8
4.3	Login	9
4.4	Teilnehmerübersicht	10
4.5	Teilnehmer hinzufügen/bearbeiten	15
4.6	Passwort ändern anderer Organisatoren	16
4.7	Passwort ändern nach erstmaliger Anmeldung	17
4.8	Versionsverlauf	18
4.9	Änderungen zwischen 2 Versionen	19
5	UML-Klassendiagramm(e) für die Komponenten zur grafischen Benutzeroberfläche, Anwendungslogik und Datenhaltung	20

6	Präzise verbale Beschreibungen der einzelnen Klassen und ihrer Attribute und öffentlichen Methoden sowie der Abhängigkeiten und Wechselwirkungen zwischen den Klassen.	21
6.1	Datentyp:QLabel	21
6.2	Datentyp:QLineEdit	21
6.3	Datentyp:QPushButton	21
6.4	Datentyp:QTableWidget	21
6.5	View_Teilnehmerliste	21
6.6	View_Versionsverlauf	22
6.7	View_Teilnehmer/TeilnehmerHinzufügen	22
6.8	View_Einloggen	23
6.9	View_ErsterSystemstart	24
6.10	View_Passwortänderung	24
6.11	View_VersionsverlaufDetailliert	25
6.12	Teilnehmerliste	25
6.13	Teilnehmer	26
6.14	Teilnehmerdaten	27
6.15	I_DAO_Teilnehmer	28
6.16	Qt_DAO_Teilnehmer	28
6.17	Organisator	28
6.18	I_DAO_Teilnehmerdaten	29
6.19	Qt_DAO_Teilnehmerdaten	29
7	Alternativer Entwurf mittels einer zentralen Managerklasse	30

1 Beschreibung der Systemarchitektur (Grobstruktur)

Unsere Systemarchitektur ist am Entwurfsmuster Drei-Schichten-Architektur orientiert. Wir haben uns für dieses Architekturmuster entschieden, da unsere Anwendung ein interaktives Anwendungssystem ist, das eine GUI Oberfläche als Präsentationsschicht, eine Managerklasse mit Interface als Anwendungsschicht und DAO Klassen mit Datenbank als Datenhaltungsschicht besitzt.

2 UML-Klassendiagramm(e) als Entwurfsmodell für die im System zu verwaltenden Daten



3 Datenbankschema (create Table Anweisungen)

3.1 Teilnehmer

```
1 CREATE TABLE teilnehmer(  
2     teilnehmerOID INTEGER AUTO_INCREMENT PRIMARY KEY  
3     Vorname VARCHAR (100) NOT NULL  
4     Nachname VARCHAR (100)  
5     Schulname VARCHAR (100) NOT NULL  
6     Straße VARCHAR (100) NOT NULL  
7     Hausnummer VARCHAR (10) NOT NULL  
8     PLZ VARCHAR (100) NOT NULL  
9     Ort VARCHAR (100) NOT NULL  
10    Land VARCHAR (100) NOT NULL  
11    E-Mail VARCHAR (100) NOT NULL UNIQUE  
12 );
```

Die Tabelle speichert alle Daten der Teilnehmer und gibt jedem Teilnehmer eine eindeutige ID. Alle Werte, außer dem Nachnamen, müssen bei dem Eintragen in die Tabelle bekannt sein, können aber im Nachhinein wieder geändert werden. Diese kann alle Spalten, außer der teilnehmerOID ändern. In dieser Tabelle werden alle Attribute der Klasse Teilnehmerdaten gespeichert, nur die Telefonnummern werden auf eine externe Tabelle ausgelagert.

3.2 Telefonnummer

```
1 CREATE TABLE telefonnummer(  
2     teilnehmerOID REFERENCES teilnehmer(teilnehmerOID)  
3     TelNr VARCHAR (100)  
4     isHaupt BOOLEAN  
5     PRIMARY KEY(teilnehmerOID, TelNr, isHaupt)  
6 );
```

Die Tabelle speichert die Telefonnummern der einzelnen Teilnehmer. Dabei wird unterschieden, ob die Telefonnummer als Haupttelefonnummer angegeben werden soll, oder nicht. Zusätzlich wird noch die zugehörige teilnehmerOID mit der Telefonnummer abgespeichert. Ein Teilnehmer kann mehrere Telefonnummern haben.

3.3 Organisator

```
1 CREATE TABLE Organisator(  
2     teilnehmerOID REFERENCES teilnehmer(teilnehmerOID)  
3     E-Mail VARCHAR (100) REFERENCES teilnehmer(E-Mail)  
4     Passwort VARCHAR (100) NOT NULL  
5     IsHauptOrg BOOLEAN  
6 );
```

Die Tabelle Organisator speichert die Objekte der Klassen Organisator und Hauptorganisator. Die teilnehmerOID ist dabei ein Fremdschlüssel aus der Tabelle teilnehmer, das Gleiche gilt für die E-Mail. Die Spalte isHauptOrg sagt über einen boolean-Wert aus, ob der Organisator ein Hauptorganisator ist.

4 Entwurf der graphischen Benutzeroberfläche mit Beschreibung ihrer Funktionen

Interaktiver Prototyp:

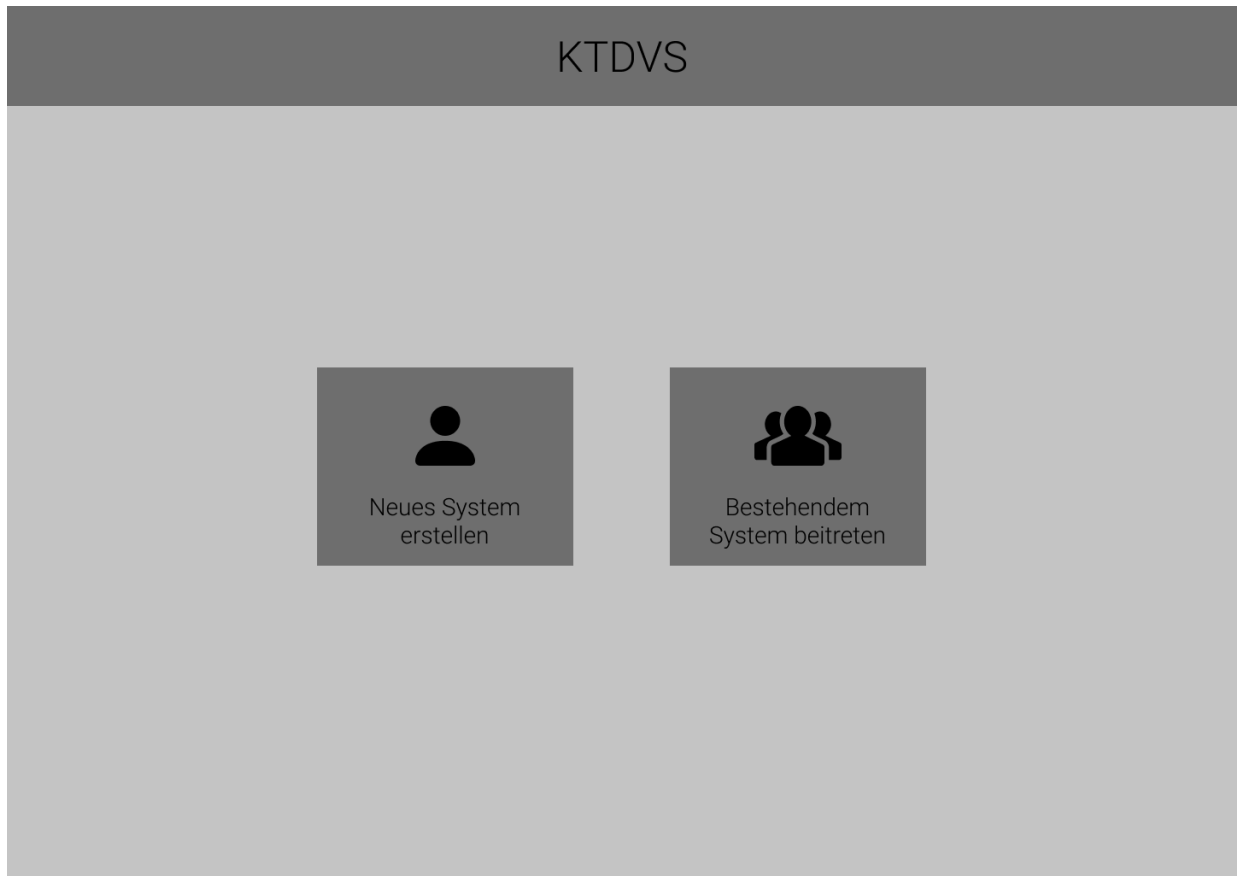
<https://www.figma.com/proto/tc3G3hYe6vGsbr1jqEykhL/KTDVS?node-id=4%3A53&scaling=min-zoom&page-id=0%3A1>

In diesem Prototypen kann sich durch die gesamte Anwendung geklickt werden. Alle Daten die enthalten sind, sind Beispiele wie das Programm während einer Nutzung aussehen könnte. Auch sind nicht alle Variationen der einzelnen Schaltflächen zu sehen. Aus diesem Grund sind entweder Felder leer auch wenn an dieser Stelle Daten eingetragen wären, oder es stehen immer die selben Daten/Texte an diesen Stellen.

Sollte man nicht wissen welche Schaltflächen nutzbar sind, kann man innerhalb des Prototypen auf eine beliebige Stelle klicken und alle interaktiven Teile werden blau hervorgehoben.

Alle einzelnen Fenster können auch unterhalb des Prototypen wie Bilder angeschaut werden. Hier besteht jedoch keine Reihenfolge sondern die Bilder werden beliebig dargestellt.

4.1 Startbildschirm - Erstes Starten des Programms



Beim ersten Starten des Systems wird abgefragt, ob ein neues System erstellt oder einem bestehenden System beigetreten werden soll. Wählt man die linke Schaltfläche "Neues System erstellen" aus, wird man zu "Neues System erstellen" (4.2) weitergeleitet. Nutzt man die rechte Schaltfläche "Bestehendem System beitreten" wird man zu "Login" (4.3) weitergeleitet.

4.2 Neues System erstellen

The screenshot shows a web interface with a dark grey header bar. On the left of the header is a button labeled "Zurück". On the right is the text "KTDVS". Below the header is a light grey main area. In the center of this area are two input fields. The first is labeled "E-Mail:" and the second is labeled "Passwort:". Below these fields is a dark grey button labeled "Erstelle Hauptorganisator".

Man hat hier die Möglichkeit ein neues System zu erstellen. Hierfür trägt man die E-Mail und das Passwort in die dafür ausgewiesenen Felder ein und wählt dann "Erstelle Hauptorganisator" aus. Daraufhin wird man auf "Teilnehmerübersicht" (4.4) weitergeleitet. Betätigt man die Schaltfläche links oben "Zurück" wird man auf "Startbildschirm - Erstes Starten des Programms" (4.1) weitergeleitet.

4.3 Login



The screenshot shows a web interface for KTDVS. At the top, a dark gray header bar contains a 'Zurück' button on the left and the text 'KTDVS' in the center. Below the header, the main content area is light gray. In the center of this area, there are two input fields. The first is labeled 'E-Mail:' and the second is labeled 'Passwort:'. Below these two fields is a dark gray button labeled 'Login'.

Auf diesem Bildschirm meldet man sich in einem bestehenden System an. Dafür muss die E-Mail und das Passwort in den dafür ausgewiesenen Feldern eingetragen werden und mit Login bestätigt werden. Bei erfolgreichem Login wird man, beim ersten Zugriff der eingegebenen E-Mail auf das System, zu "Passwort ändern" (4.6) weitergeleitet. Sollte bereits ein Login mit der eingegebenen E-Mail erfolgt sein, findet man sich daraufhin in "Teilnehmerübersicht" (4.4) wieder.

4.4 Teilnehmerübersicht



Dieser Screen präsentiert sich, wenn nur der Hauptorganisator im System eingetragen ist. Durch Betätigen der Schaltfläche links oben "Teilnehmer hinzufügen", kann man dem System einen neuen Teilnehmer über "Teilnehmer hinzufügen/bearbeiten" (4.5) hinzufügen. Der Text "Programm mit Daten füllen" ist nur vorhanden, da es sich hier um einen Prototypen handelt. Über diesen Text kann das Programm im Prototyp mit weiteren Daten gefüllt werden.

<div>Teilnehmer hinzufügen</div> <div>KTDVS</div> <div></div>		
▼	Max Mustermann	E-Mail: max.mustermann@gmail.com
▼	Simon Ruttmann	E-Mail: simon.ruttmann@studmail.htw-aalen.de
▼	Jan Kermer	E-Mail: jan.kermer@studmail.htw-aalen.de
▼	Maximilian Borst	E-Mail: maximilian.borst@studmail.htw-aalen.de
▼	Michael Ulrich	E-Mail: michael.ulrich@studmail.htw-aalen.de
▼	Veronika Scheller	E-Mail: veronika.scheller@studmail.htw-aalen.de
▼	Daniela Hirsch	E-Mail: daniela.hirsch@gmail.com
▼	Maxi Musterfrau	E-Mail: maxi.musterfrau@gmail.com
▼	Mini Mustertochter	E-Mail: mini.mustertochter@gmail.com
▼	David Pfeffer	E-Mail: david.pfeffer@gmail.com
▼	Janina Schmid	E-Mail: janina.schmid@gmail.com

Sobald Teilnehmer im System vorhanden sind, werden diese alle hier angezeigt. Um die Details einzelner Teilnehmer zu sehen muss auf diese gedrückt werden. Daraufhin wird eine detailliertere Ansicht ausgeklappt. In unserem Prototypen sind nur die Einträge "Max Mustermann" und "Veronika Scheller" implementiert.

Teilnehmer hinzufügen		KTDVS		
▲	Max Mustermann		Versionsverlauf	
Schulname: Muster	Adresse: Musterstr. 420, Musterstadt	E-Mail: max.mustermann@gmail.com	Daten ändern	
Land: Deutschland	Telefonnummer: xxxxx-xxxxx		als Organisator hinzufügen	
Kommentar:				
▼	Simon Ruttmann		E-Mail: simon.ruttmann@studmail.htw-aalen.de	
▼	Jan Kermer		E-Mail: jan.kermer@studmail.htw-aalen.de	
▼	Maximilian Borst		E-Mail: maximilian.borst@studmail.htw-aalen.de	
▼	Michael Ulrich		E-Mail: michael.ulrich@studmail.htw-aalen.de	
▲	Veronika Scheller		Versionsverlauf	
Schulname: Scheller	Adresse: Musterstr. 69, Aalen	E-Mail: veronika.scheller@studmail.htw-aalen.de	Daten ändern	
Land: Deutschland	Telefonnummer: xxxxx-xxxxx, xxxxx-xxxxx		als Organisator hinzufügen	
Kommentar:				
▼	Daniela Hirsch		E-Mail: daniela.hirsch@gmail.com	

In dieser Ansicht sind die Teilnehmer "Max Mustermann" und "Veronika Scheller" ausgeklappt worden. Nun besteht die Möglichkeit den "Versionsverlauf" (4.7) dieses Teilnehmers zu betrachten. Über "Daten bearbeiten" (4.5) können die Daten geändert werden und über "als Organisator hinzufügen" können dem Teilnehmer zusätzliche Rechte zugewiesen werden.

Auf den drei folgenden Bildern ist diese Darstellung der Details in verschiedenen Konfigurationen zu sehen.

Teilnehmer hinzufügen

KTDVS

▲

Max Mustermann

Schulname: Muster

Land: Deutschland

Kommentar:

Adresse: Musterstr. 420, Musterstadt

Telefonnummer: xxxxx-xxxxx

E-Mail: max.mustermann@gmail.com

Versionsverlauf

Daten ändern

als Organisator hinzufügen

Programm mit Daten füllen

Teilnehmer hinzufügen

KTDVS

▲

Max Mustermann

Schulname: Muster

Land: Deutschland

Kommentar:

Adresse: Musterstr. 420, Musterstadt

Telefonnummer: xxxxx-xxxxx

E-Mail: max.mustermann@gmail.com

Versionsverlauf

Daten ändern

als Organisator hinzufügen

▼

Simon Ruttmann

E-Mail: simon.ruttmann@studmail.htw-aalen.de

▼

Jan Kermer

E-Mail: jan.kermer@studmail.htw-aalen.de

▼

Maximilian Borst

E-Mail: maximilian.borst@studmail.htw-aalen.de

▼

Michael Ulrich

E-Mail: michael.ulrich@studmail.htw-aalen.de

▼

Veronika Scheller

E-Mail: veronika.scheller@studmail.htw-aalen.de

▼

Daniela Hirsch

E-Mail: daniela.hirsch@gmail.com

▼

Maxi Musterfrau

E-Mail: maxi.musterfrau@gmail.com

▼

Mini Mustertochter

E-Mail: mini.mustertochter@gmail.com

Teilnehmer hinzufügen		KTDVS	
▼	Max Mustermann	E-Mail: max.mustermann@gmail.com	
▼	Simon Ruttmann	E-Mail: simon.ruttmann@studmail.htw-aalen.de	
▼	Jan Kermer	E-Mail: jan.kermer@studmail.htw-aalen.de	
▼	Maximilian Borst	E-Mail: maximilian.borst@studmail.htw-aalen.de	
▼	Michael Ulrich	E-Mail: michael.ulrich@studmail.htw-aalen.de	
▲	Veronika Scheller	<div>Versionsverlauf</div> <div>Daten ändern</div> <div>als Organisator hinzufügen</div>	
	Schulname: Scheller Land: Deutschland Kommentar:	Adresse: Musterstr. 69, Aalen Telefonnummer: xxxxx-xxxxx, xxxxx-xxxxx	E-Mail: veronika.scheller@studmail.htw-aalen.de
▼	Daniela Hirsch	E-Mail: daniela.hirsch@gmail.com	
▼	Mini Mustertochter	E-Mail: mini.mustertochter@gmail.com	
▼	David Pfeffer	E-Mail: david.pfeffer@gmail.com	

4.5 Teilnehmer hinzufügen/bearbeiten

The screenshot shows the KTDVS web interface for editing a participant. At the top, there is a dark header bar with a button labeled "Abbrechen" on the left, the title "KTDVS" in the center, and a document icon on the right. Below the header, the user's name "Max Mustermann" is displayed next to a small icon. The main content area is divided into three sections. On the left, there is a vertical list of input fields for personal data: "Vorname:", "Nachname:", "Schulname:", "Straße:", "Hausnummer:", "Plz:", "Ort:", "Land:", "E-Mail:", "Telefonnummer:", and "weitere Telefonnummern:". Fields for "Vorname:", "Schulname:", "Straße:", "Hausnummer:", "Plz:", "Ort:", "Land:", "E-Mail:", and "Telefonnummer:" are marked with a red asterisk, indicating they are mandatory. Below the "weitere Telefonnummern:" field is a small square button with a plus sign. In the center, there is a large text area labeled "Kommentar...". On the right, there is a vertical stack of four buttons: "Versionsverlauf", "Organisatorrechte entfernen", "Passwort ändern", and "Speichern".

In diesem Bildschirm kann man Teilnehmer ändern oder hinzufügen. Hierzu fügt man in die ausgewiesenen Felder die zugehörigen Daten ein. Die mit einem Stern markierten Felder sind hierbei verpflichtend. Sollen mehr als zwei Telefonnummern angegeben werden, kann die Schaltfläche links unten, die durch ein Quadrat umrandetes Plus gekennzeichnet wird, auswählen. Daraufhin erscheint ein weiteres Eingabefeld. Sollten alle Pflichtfelder ausgefüllt sein, kann man die "Speichern" Schaltfläche (rechts mittig) betätigen und somit die Änderungen speichern.


Durch Betätigen der Schaltfläche "Abbrechen" (links oben) kehrt man zu "Teilnehmerübersicht" (4.4) zurück.

Wählt man die Schaltfläche "Versionsverlauf" (rechts oben) wird man zu "Versionsverlauf" (4.7) gebracht.

Nutzt man "Organisatorrechte entfernen" (rechts oben) kann man einem Organisator seine Rechte entziehen, auf das System zuzugreifen.

Mit der Schaltfläche "Passwort ändern" (rechts mittig) kann man, sollte man als Hauptorganisator angemeldet sein, das Passwort der Organisatoren ändern. Man wird zu "Passwort ändern" (4.7) weitergeleitet.

4.6 Passwort ändern anderer Organisatoren

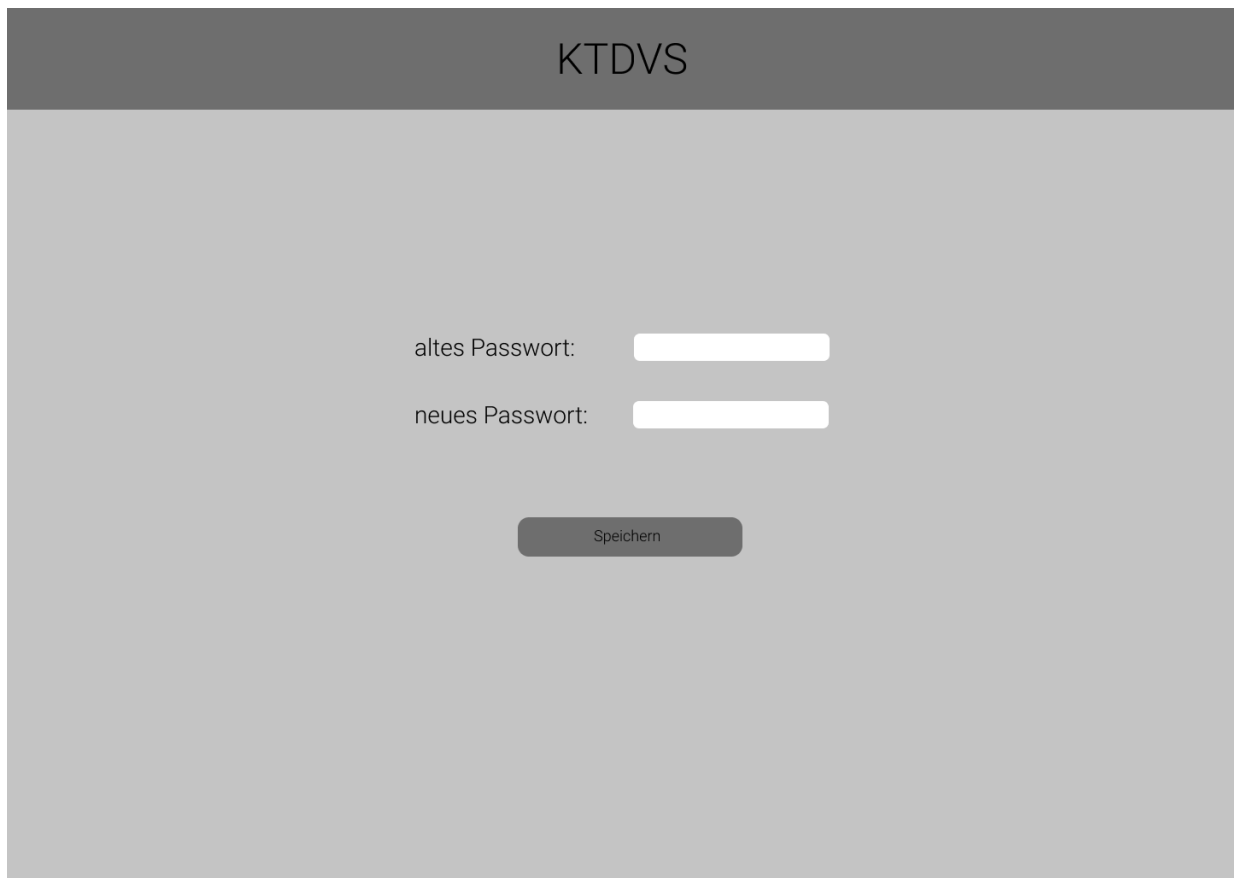


The screenshot shows a mobile application interface for KTDVS. At the top, there is a dark grey header bar. On the left side of this bar is a light grey button labeled "Zurück". On the right side is the text "KTDVS" in white. Below the header, the background is a light grey gradient. In the center, the text "neues Passwort:" is displayed next to a white rectangular input field. Below the input field is a dark grey button labeled "Passwort ändern".

Hier kann man das Passwort des vom Hauptorganisator ausgewählten Organisators ändern. Hierfür wird das neue Passwort in das zugehörige Feld eingetragen. Durch den "Speichern" Button wird das Passwort geändert.

Durch Betätigen der "Abbrechen"-Schaltfläche (links oben) kehrt man zu dem Screen zurück von dem man kam.

4.7 Passwort ändern nach erstmaliger Anmeldung



The screenshot shows a web interface for KTDVS. At the top, there is a dark gray header bar with the text "KTDVS" in white. Below the header, the background is a light gray. In the center, there are two input fields. The first field is preceded by the text "altes Passwort:" and the second by "neues Passwort:". Both fields are white with rounded corners. Below these fields is a dark gray button with the text "Speichern" in white.

Hier muss ein Organisator beim erstmaligen Anmelden sein eigenes Passwort ändern. Hierfür wird das alte und neue Passwort in die zugehörigen Felder eingetragen. Durch den "Speichern" Button wird bei korrekter Eingabe des alten Passworts, das Passwort geändert.

Durch Betätigen der "Abbrechen"-Schaltfläche (links oben) kehrt man zu dem Screen zurück von dem man kam.

4.8 Versionsverlauf

KTDVS			
Zurück			
▼ Änderungsdatum	▼ geänderte Person	▼ Organisator	
14.04.2021 12:21:10	Veronika Ulrich	Michael Ulrich	Details
13.04.2021 10:32:16	Veronika Scheller	Michael Ulrich	Details
11.04.2021 20:59:55	Veronika Scheller	Maximilian Borst	Details
06.04.2021 14:20:00	Veronika Scheller	Simon Ruttmann	Details
06.04.2021 08:40:34	Veronika Scheller	Simon Ruttmann	Details

Auf diesem Bildschirm werden alle Organisatoren, welche Änderungen an einem ausgewählten Teilnehmer vorgenommen haben, angezeigt. Durch Betätigen der "Details" Schaltfläche wird man auf die Änderung zwischen der vorherigen Version und der darauffolgenden (4.10) weitergeleitet.

Nach Benutzen der "Zurück"-Schaltfläche kehrt man zur detaillierten Personenansicht (4.5) zurück.

4.9 Änderungen zwischen 2 Versionen

Zurück

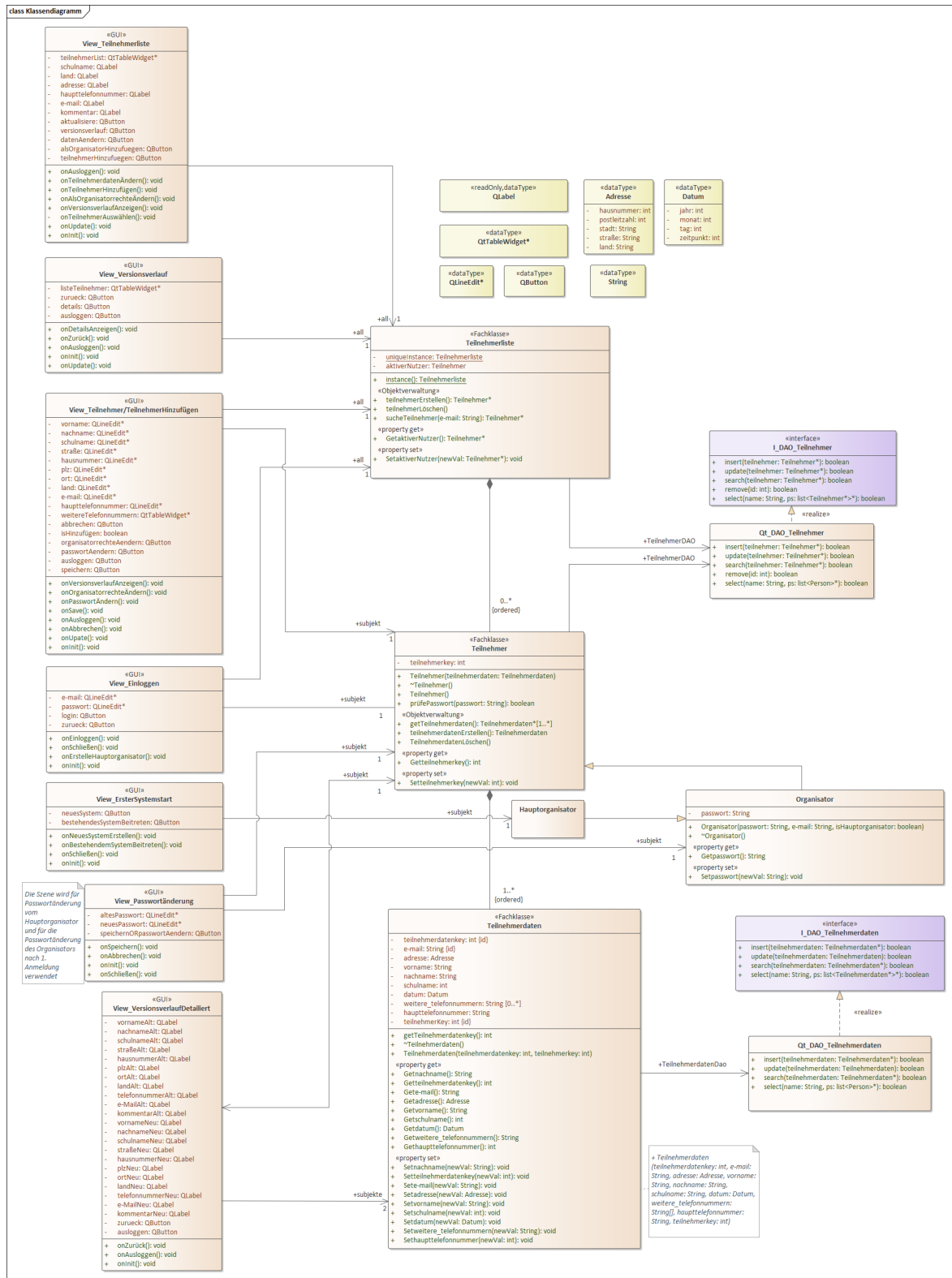
KTDVS

Max Mustermann	🔄 alte Version	Max Mustermann	🔄 neue Version	
Vorname:	<input type="text"/>	Vorname:	<input type="text"/>	Änderungskommentar...
Nachname:	<input type="text"/>	Nachname:	<input type="text"/>	
Schulname:	<input type="text"/>	Schulname:	<input type="text"/>	
Straße:	<input type="text"/>	Straße:	<input type="text"/>	
Hausnummer:	<input type="text"/>	Hausnummer:	<input type="text"/>	
Plz:	<input type="text"/>	Plz:	<input type="text"/>	
Ort:	<input type="text"/>	Ort:	<input type="text"/>	
Land:	<input type="text"/>	Land:	<input type="text"/>	
Telefonnummer:	<input type="text"/>	Telefonnummer:	<input type="text"/>	
E-Mail:	<input type="text"/>	E-Mail:	<input type="text"/>	
Kommentar:	<input type="text"/>	Kommentar:	<input type="text"/>	

Hier sieht man die genaue Änderung die an einem Teilnehmer vorgenommen wurde.

Durch Betätigen der "Zurück"-Schaltfläche kehrt man zum vorherigen Screen zurück.

5 UML-Klassendiagramm(e) für die Komponenten zur grafischen Benutzungsoberfläche, Anwendungslogik und Datenhaltung



6 Präzise verbale Beschreibungen der einzelnen Klassen und ihrer Attribute und öffentlichen Methoden sowie der Abhängigkeiten und Wechselwirkungen zwischen den Klassen.

6.1 Datentyp:QLabel

- Wird in den **View-Klassen** verwendet um Text darzustellen.
- Der Datentyp wird ausschließlich zum lesen verwendet und ist deshalb als «readOnly» deklariert.

6.2 Datentyp:QLineEdit

- Wird in den **View-Klassen** verwendet um Eingabefelder darzustellen. Diese können unter anderem auch bereits Text beinhalten.

6.3 Datentyp:QPushButton

- Wird in den **View-Klassen** verwendet um Buttons darzustellen.

6.4 Datentyp:QTableWidget

- Wird in den **View-Klassen** verwendet Daten in einer Tabelle darzustellen.

6.5 View__Teilnehmerliste

- Diese Klasse dient dazu die Teilnehmer eines Systems in einer Tabelle darzustellen und beim Auswählen eines Teilnehmer seine Daten anzuzeigen.
- Die View__Teilnehmerliste besitzt eine Assoziation zur Klasse Teilnehmerliste, da diese die Teilnehmerliste anzeigt und somit Zugriff auf das Objekt der Klasse Teilnehmerliste benötigt.
- Die Methode **onAusloggen()** loggt den Benutzer aus und führt ihn zurück zur Startszene.
- Die interne Methode **onTeilnehmerAuswählen()** wird durch das Klicken auf den Pfeil neben den Namen ausgelöst. Dadurch werden die Daten des ausgewählten Teilnehmers ausgeklappt und dem Benutzer angezeigt.
- Die Methode **onTeilnehmerdatenÄndern()**, so wie die Methode **onTeilnehmerHinzufügen()**, dient der Weiterleitung auf die Szene der **View__Teilnehmer/TeilnehmerHinzufügen**, wobei die Textfelder vorausgefüllt und veränderbar sind.

- Die Methode **onAlsOrganisatorrechteÄndern()** ist identisch zur Methode **onOrganisatorrechteÄndern()** aus der Klasse **View__Teilnehmer/TeilnehmerHinzufügen**.
- Bei der Methode **onVersionsverlaufAnzeigen()** wird der Benutzer auf die Szene der Klasse **View__Versionsverlauf** weitergeleitet.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.
- Mit der Methode **onUpdate()** aktualisiert die Klasse ihre Daten.

6.6 View__Versionsverlauf

- Die Methode **onDetailsAnzeigen()** leitet den Benutzer auf die Szene der Klasse **View__VersionsverlaufDetailliert** weiter.
- Die **View__Versionsverlauf** benötigt Zugriff auf die Objektverwaltung der Teilnehmer, um die Änderungen als Items über die **QTableWidet** anzuzeigen.
- Die Methode **onZurück()** bringt den Benutzer auf die letzte Szene zurück. In diesem Fall ist es immer die Szene der **View__Teilnehmerliste**.
- Die Methode **onAusloggen()** wird beim Schließen des Fensters ausgeführt, führt eine Abmeldung des Organisators vom System durch und schließt das Fenster.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.
- Mit der Methode **onUpdate()** aktualisiert die Klasse ihre Daten.

6.7 View__Teilnehmer/TeilnehmerHinzufügen

- Diese Klasse ermöglicht das Hinzufügen und Bearbeiten der Daten eines Teilnehmers. Wenn der Teilnehmer erst hinzugefügt wird, ist nur der Button "Speichern" zu sehen.
- Die **View__Teilnehmer** beinhaltet eine Assoziation zur Teilnehmerliste, sowie zu Teilnehmer. Dies ist notwendig, um einen Teilnehmer auszuwählen und dessen Daten in der View anzuzeigen.
- Die Methode **onVersionsVerlaufAnzeigen()** leitet den Benutzer zur Szene der Klasse **View__Versionsverlauf**. Dort wird der Versionsverlauf des aktuell ausgewählten Teilnehmers angezeigt.

- Die Methode **onOrganisatorRechteÄndern()** ermöglicht dem Hauptorganisator die Organisatorrechte eines Teilnehmers zu ändern. Der Button, der diese Methode auslöst, wird also nur dem Hauptorganisator angezeigt.
Wenn der ausgewählte Teilnehmer noch kein Organisator ist, wird der Button "als Organisator hinzufügen" angezeigt. Die Methode wird äquivalent zur Methode **onAlsOrganisatorrechteÄndern()** der Klasse **Teilnehmerliste** behandelt. Der Benutzer wird dann auf die Szene der Klasse **View_Passwortänderung** weitergeführt und kann dort ein Passwort für den neuen Organisator eingeben.
Ist der Teilnehmer bereits ein Organisator, erscheint stattdessen der Button "Organisatorrechte entfernen". Wird dieser betätigt, wird das Fenster neu geladen, nachdem der Teilnehmer aus der Klasse **Organisator** entfernt wurde.
- Die Methode **onPasswortÄndern()** wird ebenfalls durch einen Button ausgeführt, welche nur dem Hauptorganisator zur Verfügung gestellt wird. Dieser ist zudem nur sichtbar, wenn es sich bei dem ausgewählten Teilnehmer um einen Organisator handelt. Durch das Betätigen des Buttons, wird der Hauptorganisator auf die Szene der Klasse **View_Passwortänderung** weitergeleitet.
- Die Methode **onSave()** bewirkt die Speicherung der Daten, indem die eingegebenen Daten an die Fachklasse **Teilnehmer** weitergeleitet werden.
- die Methode **onAusloggen()** wird beim Schließen des Fensters ausgeführt, führt eine Abmeldung des Organisators vom System durch und schließt das Fenster.
- Die Methode **onAbbrechen()** verwirft alle gemachten Änderungen und leitet den Benutzer auf die vorherigen Szene. **View_Teilnehmerliste** zurück.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.
- Mit der Methode **onUpdate** aktualisiert die Klasse ihre Daten.

6.8 View__Einloggen

- Diese Klasse stellt das Anmeldefenster dar. Wird ein neues System erstellt, wird dieses Fenster mit dem Button "Erstelle Hauptorganisator" angezeigt. Will man sich standardmäßig bei einem bestehendem System anmelden, so wird dieses Fenster mit dem Button "Einloggen" angezeigt.
- Die Assoziationen zur Teilnehmerliste und Teilnehmer wird dazu verwendet um anhand der eingegebenen E-Mail den entsprechenden Teilnehmer auszuwählen und über ihn das eingegebene Passwort zu prüfen.

- Die Methode **onEinloggen()** überprüft ob die E-Mail und das Passwort im System vorkommen und zueinander gehören. Ist das der Fall wird der Benutzer eingeloggt und auf das nächste Fenster geleitet. Ansonsten wird ihm eine Fehlermeldung angezeigt, die den Benutzer darum bittet sich erneut Anzumelden, da die E-Mail oder das Passwort falsch sind.
- Bei der Methode **onErstelleHauptorganisator** wird ein neues System erstellt. Die eingegebene E-Mail wird dem Hauptorganisator zugeteilt.
- Die Methode **OnSchließen()** wird die Schließung des Fensters durchführen.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.

6.9 View__ErsterSystemstart

- Diese Klasse ist für die Darstellung des folgenden Fensters verantwortlich: Das Auswahlfenster, beim ersten Starten des Systems, welches dem Nutzer die Wahl zwischen dem Erstellen eines Systems und dem Beitreten eines bestehenden Systems bietet.
- Die **View__ErsterSystemstart** besitzt eine Verbindung zur Klasse Hauptorganisator, da im Rahmen eines neuen erstellen eines Systems ein Objekt vom Typ Hauptorganisator angelegt werden muss.
- Die beiden öffentlichen Methoden **onNeuesSystemErstellen()** und **onBestehendemSystemBeitreten()** realisieren die Weiterleitung auf den entsprechenden nächsten Bildschirm.
- Die Methode **OnSchließen()** wird die Schließung des Fensters durchführen.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.

6.10 View__Passwortänderung

- (1) Die Klasse ist für die Passwortänderungen beim ersten Anmelden mit einem zugewiesenen Passwort eines Organisators zuständig und (2) die Änderung des Passworts eines Organisators durch den Hauptorganisator.
- Die **View__Passwortänderung** beinhaltet eine Assoziation zu einem Objekt des Typs Organisator. Auf dieses Objekt wird die Passwortänderung vom Hauptorganisator (2. Assoziation) ausgeführt.
- Die Methode **onSpeichern()**, wird durch einen Klick auf den Button "Speichern" ausgelöst. Sie prüft in (1) das alte Passwort und überschreibt bei positiver Prüfung das alte Passwort mit dem neuen, in (2) erfolgt keine Prüfung.

- Die Methode **onAbbrechen()** ist nur in (2) verfügbar und wird ausgelöst durch das Klicken auf den Zurück-Button, sie wechselt in die vorherige Szene.
- Die Methode **onSchließen()** (1) wird beim Schließen des Fensters ausgeführt und führt eine Abmeldung des Organisators vom System durch und schließt das Fenster. Für den Fall (2) verhält sich die Methode wie **onAusloggen()**.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.

6.11 View_VersionsverlaufDetailliert

- Diese Klasse veranschaulicht eine bewirkte Änderung in den Daten eines Teilnehmers. Angezeigt wird das mit zwei nebeneinander stehenden Anmeldeschablonen, die nun nicht bearbeitet werden können und damit nicht anklickbar sind. Links soll dann die vorherige Version zu sehen und rechts die Schablone mit den Änderungen.
- Beim View_VersionsverlaufDetailliert wird ein Teilnehmer ausgewählt, und 2 ausgewählte Teilnehmerdaten des Teilnehmers gegenübergestellt.
- Die Methode **onZurück()** führt den Benutzer in das vorherige Fenster zurück. Dies ist das Fenster **View_Versionsverlauf**.
- Die Methode **onAusloggen()** sorgt dafür das der Benutzer ausgeloggt wird. Danach wird der Bildschirm **View_Einloggen** angezeigt.
- Die Methode **onInit()** wird beim Öffnen der Seite ausgeführt und baut die GUI der Seite auf.

6.12 Teilnehmerliste

- Diese Klasse bildet die Fachklasse der Teilnehmerliste.
- Es besteht eine geordnete Komposition zwischen den Klassen Teilnehmerliste und Teilnehmer. Die Teilnehmerliste dient somit als Objektverwaltung der Objekte Teilnehmer.
- Die Methode **instance()** zeigt auf das Attribut uniqueinstance. Wenn das Attribut noch NULL ist, wird ein neues Objekt der Instanz Teilnehmerliste erstellt und in das Attribut gespeichert. Hier wird das Entwurfsmuster des Singeltons angewandt.
- Die Methode **teilnehmerErstellen()** ruft den Konstruktor der Klasse **Teilnehmer** auf und übergibt die neuen/geänderten Daten des Teilnehmers.
- Die Methode **teilnehmerLöschen()** löscht einen bestehenden Teilnehmer. Diese Methode wird benötigt um bestehende Teilnehmer aus dem System zu löschen, um sie als Organisator hinzuzufügen.

- Die Methode **sucheTeilnehmer(e-mail:String): Teilnehmer**, prüft ob es die angegebene E-mail in der Datenbank gibt und liefert das damit verknüpfte Teilnehmerobjekt zurück, falls es gefunden wird. Alternativ wird NULL zurückgeliefert.
- Die Methode **GetAktiverNutzer() : Teilnehmer** liefert bei Aufruf das Teilnehmerobjekt des gerade operierenden Operators zurück.
- Die Methode **SetAktiverNutzer(newVal: Teilnehmer)** wird dafür verwendet, den angemeldeten Benutzer zu setzen. Das Attribut wird benötigt, um unter anderem bei Datenänderungen die Person einzutragen, welche die Änderung vornahm.

6.13 Teilnehmer

- Diese Fachklasse stellt Teilnehmerobjekte der Datenbank dar.
- Die Klasse Teilnehmer besitzt eine geordnete Komposition zwischen Teilnehmerdaten und Teilnehmer und dient somit als Objektverwaltung der Teilnehmerdaten. Die Teilnehmerobjekte selbst, werden von der Teilnehmerliste verwaltet.
- Die Methode **Teilnehmer(teilnehmerdaten: Teilnehmerdaten)** ist ein Konstruktor der Teilnehmerklassen, sie initialisiert das erzeugte Objekt mit teilnehmerdaten.
- Die Methode **~ Teilnehmer()** ist der Destruktor des Objekts Teilnehmer
- Die Methode **Teilnehmer()** ist der Standardkonstruktor.
- Die Methode **getTeilnehmerDaten():Teilnehmerdaten** gibt die Teilnehmerdaten eines Teilnehmers zurück.
- Die Methode **prüfePasswort(password:String):boolean** überprüft ob der übergebene Parameter mit dem Passwort des Teilnehmers übereinstimmt und gibt darauf ein boolean-Wert zurück.
- Die Methode **teilnehmerDatenErstellen():Teilnehmerdaten** ruft den Konstruktor der Klasse **Teilnehmerdaten** auf.
- Die Methode **teilnehmerDatenLöschen()** ruft den Destruktor der Klasse **Teilnehmerdaten**.
- Die Methode **getTeilnehmKey()** ermittelt über ein zugewiesenes Teilnehmerdatenobjekt den DatenKey und gibt ihn zurück.
- Die Methode **setTeilnehmerKey(newVal:int)** legt einen neuen Teilnehmerkey für einen Teilnehmer fest.

6.14 Teilnehmerdaten

- Diese Fachklasse stellt alle Daten die einem Teilnehmer zugeordnet werden können als Objekt dar
- Die Methode **Teilnehmerdaten(teilnehmerDatenKey: int, teilnehmerKey: int)** ist ein Konstruktor der TeilnehmerDatenklasse, sie initialisiert das erzeugte Objekt mit den beiden übergebenen Schlüsseln.
- Die Methode **Teilnehmerdaten()** ist der Destruktor des Objekts Teilnehmer.
- Die Methode **getTeilnehmerdatenKey():int** liefert den Key des Teilnehmerdatenobjekts zurück.
- Die Methode **getNachname():String** liefert den Nachnamen des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **getEmail():String** liefert den Nachnamen des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **getAdresse():Adresse** liefert die Adresse des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **getVorname():String** liefert den Vornamen des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **getSchulname():String** liefert den Schulnamen des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **getDatum():Datum** liefert das Datum und den Zeitstempel zurück, andem diese Version der Teilnehmerdaten entstanden ist.
- Die Methode **getWeitereTelefonnummern():String** liefert alle weiteren Telefonnummern des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **getHauptTel():String** liefert die Haupttelefonnummer des aktuellen Teilnehmerdaten-Objektes zurück.
- Die Methode **setnachname(newVal:String)** ermöglicht das Attribut Nachname in der aktuellen Instanz von Teilnehmerdaten zu ändern.
- Die Methode **setvorname(newVal:String)** ermöglicht das Attribut Vorname in der aktuellen Instanz von Teilnehmerdaten zu ändern.
- Die Methode **setAdresse(newVal:Adresse)** ermöglicht das Attribut Adresse in der aktuellen Instanz von Teilnehmerdaten zu ändern.
- Die Methode **setHauptTel(newVal:String)** ermöglicht das Attribut haupttelefonnummer in der aktuellen Instanz von Teilnehmerdaten zu ändern.

- Die Methode **setweitere_telefonnummern(newVal:String)** ermöglicht das Attribut `weitere_telefonnummern` in der aktuellen Instanz von Teilnehmerdaten zu ändern.
- Die Methode **setDatum(newVal:Datum)** ermöglicht das Attribut `Datum` in der aktuellen Instanz von Teilnehmerdaten zu ändern.
- Die Methode **setSchulname(newVal:String)** ermöglicht das Attribut `Schulname` in der aktuellen Instanz von Teilnehmerdaten zu ändern.
- Die Methode **setEmail(newVal:String)** ermöglicht das Attribut `Schulname` in der aktuellen Instanz von Teilnehmerdaten zu ändern.

6.15 I_DAO_Teilnehmer

- Interface der Schnittstelle zwischen Programm und Datenbank
- Die Methode **insert(teilnehmer: Teilnehmer*):boolean** fügt eine Teilnehmerobjekt der Datenbank hinzu
- Die Methode **update(teilnehmer: Teilnehmer*):boolean** prüft ob ein Teilnehmerobjekt in der Datenbank eingetragen ist.
- Die Methode **search(teilnehmer: Teilnehmer*):boolean**
- Die Methode **remove(id:int):boolean**
- Die Methode **select(name:String,ps:list<Teilenehmer*>):boolean**

6.16 Qt_DAO_Teilnehmer

- Realisiert die Schnittstelle zwischen Programm und Datenbank, implementiert die Methoden der `I_DAO_Teilnehmer` (für weitere Ausführungen siehe die erwähnte Klasse)

6.17 Organisator

- Die Organisatorklasse repräsentiert Organisatoren im Programm
- Die Methode **Organisator(password: String,email:String,isHauptOrganisator:boolean)** ist ein Konstruktor der Organisatorklasse, sie initialisiert das erzeugte Objekt mit den übergebenen Daten.
- Die Methode **~Organisator()** ist der Destruktor des Objekts Teilnehmer
- Die Methode **getPasswort():String** gibt das Attribut `Passwort` zurück
- Die Methode **setPasswort(newVal:String)** setzt das Attribut `Passwort` zum Organisatorobjekt

6.18 I_DAO_Teilnehmerdaten

- Interface der Schnittstelle zwischen Programm und Datenbank
- Die Methode **insert(teilnehmerdaten: Teilnehmerdaten*):boolean** fügt ein Teilnehmerdatenobjekt der Datenbank hinzu
- Die Methode **update(teilnehmerdaten: Teilnehmerdaten*):boolean** updated ein bereits gespeichertes Teilnehmerdatenobjekt
- Die Methode **search(teilnehmerdatenr: Teilnehmerdatenr*):boolean** prüft ob ein Teilnehmerdatenobjekt in der Datenbank eingetragen ist
- Die Methode **select(name:String,ps:list<Teilnehmerdaten*>):boolean**

6.19 Qt_DAO_Teilnehmerdaten

- Realisiert die Schnittstelle zwischen Programm und Datenbank, implementiert die Methoden der I_DAO_Teilnehmerdaten (für weitere Ausführungen siehe die erwähnte Klasse)

7 Alternativer Entwurf mittels einer zentralen Managerklasse

