

LB-2 2019 18l: iet-133

Name:

Vorname:

		Punkte	Note
maximale Punktzahl = 34 Zeitbudget = ca. 90 Minuten	Note = (Punkte * 5 / Maximale Punktzahl) + 1		

Gewichtung der Leistungsbeurteilung 2-2019: 70% der Gesamtnote

Erlaubte Hilfsmittel:

- › Schriftliche Unterlagen + beliebige Bücher
- › Internetrecherche
- › Modulunterlagen
- › bmLP1
- › selbsterstellten Lösungen im Modul 133

Nicht erlaubt sind Musterlösungen aus früheren Prüfungen, Plagiate und Hilfestellungen durch Dritte. Nichtbeachtung führt zur Note 1.

Screens und Netzwerkverkehr können aufgezeichnet und nachträglich zur Bewertung und Analyse beigezogen werden.

Lieferobjekt

Sie komprimieren nach Ablauf der Prüfungszeit Ihre Rails-App aeroport (gesamtes Verzeichnis) und benennen das Archiv mit **<Klasse>_<Nachname>_<Vorname>_LB2_2019.zip**.

Zum Beispiel: /home/vmadmin/workspace/aeroport

└──┬──
18A_Maurer_Ralph_LB2_2019.zip

Im Archiv enthalten ist ein Ordner Screenshots, der Bilder aller Views enthält.

Kopieren Sie das Archiv von der LP1 auf Ihren HOST-PC und verschieben Sie das Archiv in <smb://10.9.4.12/sh-exams/<ihre Klasse>>. Den Pfad müssen Sie im ubuntu Dateien Explorer mit «+ Andere Orte» (Button in der Seitenleiste) erfassen. Authentifizieren Sie sich mit dem IET-LOGIN.

Sie sehen Ihre Abgabearchiv nur wenige Sekunden. Vergewissern Sie sich bei der Lehrperson, ob Ihr Lieferobjekt korrekt gesichert ist. Sie tragen die Verantwortung für die ordentliche Prüfungsabgabe.

Nach Abschluss der LB können Sie LP1 die LP ausschalten und die Prüfung abschliessen.

Bei Fragen wenden Sie sich an die Lehrperson.

Auftrag

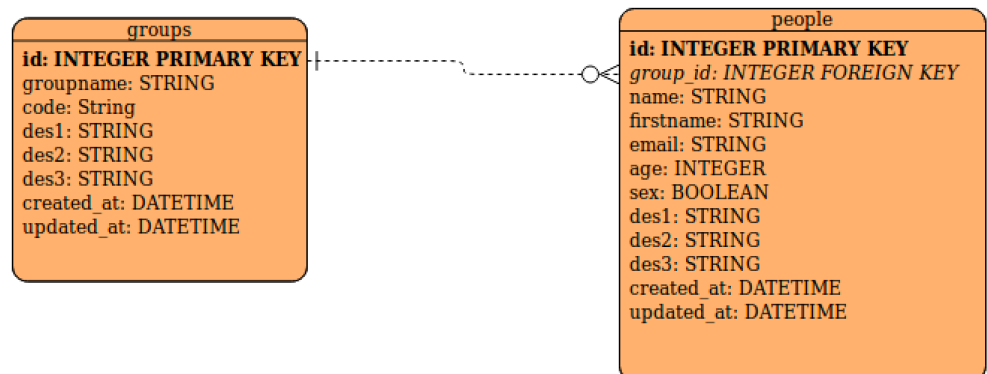
Sie bauen die Belegung von Passagieren in verschiedenen Flugzeugen.

Vorbereitungen

1. Erstellen Sie eine neue Rails App mit Namen BudgetBook. Nutzen Sie das Verzeichnis `/home/vmadmin/workspace/aeroporto`
`rails new .`
2. In der Datei `config/puma.rb` erfassen Sie den Port 5050.
`port ENV.fetch("PORT") { 5050 }`
3. Wenn Sie nun <http://localhost:5050> im Browser aufrufen, erscheint das Default-Bild der neuen Rails App.

Modell

In Ihrer App enthalten ist eine Datenbank (sqlite) mit folgenden Tabellen, Attributen und Beziehungen. Achten Sie auf die Bezeichner, diese müssen genau der Vorgabe entsprechen.



Eine Person kann in einer Gruppe (Group) sein. In der Tabelle **people** im Attribut **group_id** ist die Gruppenzugehörigkeit der Person gespeichert. Achten Sie darauf, dass die plurale Schreibform von Person in Englisch **people** ist.

Namenskonvention

Achten Sie auf die Einhaltung der Rails gängigen Namenskonvention!

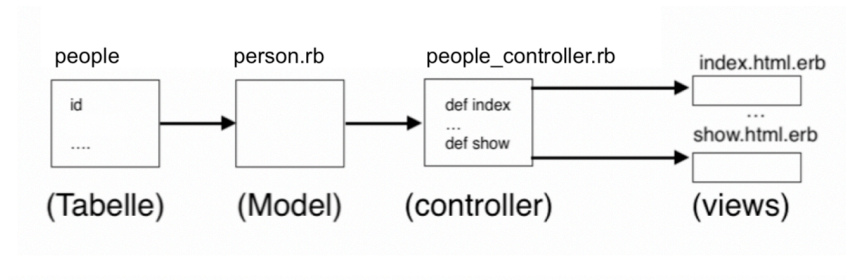


Tabelle groups

Model

```
Group(id: integer, groupname: string, code: string, des1: string,
des2: string, des3: string, created_at: datetime, updated_at:
datetime)
```

id	Primärschlüssel wird von Rails automatisch erstellt und verwaltet
groupname	Name der Gruppe (In unserem Beispiel eine Gruppe von Passagieren eines bestimmten Fluges z.B. ZRH-TLX Swiss)
code	Ein Gruppencode (In unserem Beispiel der Code des Fluges z.B. LX 3005)
des1	Ein Beschreibungsfeld 1 (description 1). In unserem Beispiel wird das Feld für den Startzeit (DEPARTURE Datetime) angewandt z.B. ZURICH ZRG / BERLIN TLX
des2	Ein Beschreibungsfeld 2 (description 2). In unserem Beispiel wird das Feld für den Startzeit (DEPARTURE Datetime) angewandt z.B. 08:30AM SATURDAY-30-MAY-2020.
des3	Ein Beschreibungsfeld 3 (description 3). In unserem Beispiel wird das Feld für den Ankunftszeit (ARRIVAL Datetime) angewandt z.B. 10:15AM SATURDAY-30-MAY-2020.
created_at	Das Erstellungsdatum wird von Rails automatisch erstellt und verwaltet
updated_at	Das Änderungsdatum wird von Rails automatisch erstellt und verwaltet

Tabelle poeople

Model

```
Person(id: integer, group_id: integer, name: string, firstname:
string, email: string, age: integer, sex: boolean, des1: string,
des2: string, des3: string, created_at: datetime, updated_at:
datetime)
```

id	Primärschlüssel wird von Rails automatisch erstellt und verwaltet
group_id	Fremdschlüssel zur Tabelle groups
name	Nachname
firstname	Vorname
email	Vorname
age	Alter

sex	Geschlecht: 1 = weiblich, 0 = männlich
des1	Ein Beschreibungsfeld 1 (description 1). In unserem Beispiel wird das Feld für die Passportnummer angewandt.
des2	Ein Beschreibungsfeld 2 (description 2). In unserem Beispiel wird das Feld für das Geburtsdatum angewandt.
des3	Ein Beschreibungsfeld 3 (description 3). In unserem Beispiel wird das Feld für die Gefahrenstufe des Passagiers angewandt. Z.B. Lovely, Crazy, Wanted, usw.
created_at	Das Erstellungsdatum wird von Rails automatisch erstellt und verwaltet
updated_at	Das Änderungsdatum wird von Rails automatisch erstellt und verwaltet

Controller

Gemäss dem MVC-Pattern werden im Controller alle Interaktionen zwischen View und Model (Programmlogik) programmiert. Programmlogik in den Views ist nicht erlaubt.

groups_controller.rb

Der Controller `groups_controller.rb` beinhaltet alle Methoden für die Auflistung aller Gruppen (Flüge) sowie die Erstellung und Löschung einer einzelnen Gruppe (ein Flug). Die dazugehörigen Views ist: `index` im Ordner `views/groups/`

Benötigte Methoden und vorgegebene Bezeichner in der Klasse GroupsController:

```
› def index  
› def create  
› def group_params  
› def destroy
```

people_controller.rb

Der Controller `people_controller.rb` beinhaltet alle Methoden für die Auflistung aller Personen (Passagiere) sowie die Erstellung, Aktualisierung und Löschung einer einzelnen Person (Passagier). Die dazugehörigen Views sind: `index`, `new`, `edit` im Ordner `views/people/`

Benötigte Methoden und vorgegebene Bezeichner in der Klasse PeopleController:

```
› def index  
› def new  
› def create  
› def person_params  
› def edit  
› def update  
› def destroy
```

View

groups#index (<http://localhost:5050/categories>)

Die View groups#index soll alle erfassten Flüge auflisten. Sind keine Flüge erfasst, bleibt die Liste leer.

- › Mit den Textfeldern FLIGHT, CODE, FROM/TO, DEPARTURE Datetime und ARRIVAL Datetime und dem Button « Save Flight» muss ein neuer Flug erfasst werden können. Der neu erfasste Flug muss auf groups#index angezeigt werden.
- › Mit dem Button «Löschen» muss der entsprechende Flug entfernt werden können.
- › Der Button «Passengers» für auf die view people#index.
Die Liste der Flüge wird an letzter Position mit der Anzahl erfasster Flüge abgeschlossen.
- › Erfassen Sie drei bis fünf Flüge

AEROPORT

Flights Aeroport

FLIGHT

CODE

FROM/TO

DEPARTURE Datetime

ARRIVAL Datetime

Save Flight

Flights	CODE	FROM/TO	DEPARTURE Datetime	DESTINATION Datetime	
ZRH-TLX SWISS	LX 3005	ZURICH ZRH / BERLIN TLX	08:30AM SATURDAY-30-MAY-2020	10:15AM SATURDAY-30-MAY-2020	Delete
TKY-SYD JPAIR	JP 6567	TOKYO / SYDNAY	18:30PM SATURDAY-04-APRIL-2020	19:50PM SATURDAY-05-APRIL-2020	Delete
MOS-MOS RUAIR	RU 99.9	MOSCOW / MOSCOW	18:30PM SATURDAY-11-APRIL-2020	18:30PM SATURDAY-11-APRIL-2020	Delete
LHR-ZRH SWISS	LX 0404	LONDON LHR / ZURICH ZRH	08:30AM SATURDAY-04-APRIL-2020	10:15AM SATURDAY-04-APRIL-2020	Delete
ISB-ZRH SWISS	LX 9991	ISLAMABAD ISB / ZURICH ZRH	11:30AM MONDAY-20-APRIL-2020	22:30PM MONDAY-20-APRIL-2020	Delete

Number of flights: 5

Passengers

people#index (<http://localhost:5050/people>)

Die view people#index ist die Startseite und unter <http://localhost:5050> erreichbar.

Die view `people#index` listet alle Passagiere mit den Spaltenbezeichner auf.

Dabei gilt (das Zeichen $:=$ ist eine formale Schreibweise für Definition):

- › Nr := Person.id
- › FLIGHT := Person.group.groupname
- › NAME := Person.name
- › FIRST := Person.firstname
- › EMAIL := Person.email
- › AGE := Person.age
- › SEX := Person.sex
- › PASSPORT := Person.des1
- › BIRTH := Person.des2
- › RATED := Person.des3

Mit dem Button edit können Sie einen bestehenden Passagier aktualisieren und gelangen auf [people#edit](#).

Mit dem Button Löschen können Sie einen bestehenden Passagier löschen und gelangen auf `people#index`.

Mit dem Button ADD Passenger kommen Sie auf die people#new View

Mit dem Button FLIGHTS kommen Sie auf die groups#index View

Geben Sie am Ende der Passagierliste folgende Daten aus:

- Anzahl erfasster Passagiere (NUMBER OF PEOPLE)
- Durchschnittsalter aller Passagiere (AVERAGE AGE)
- Anzahl Passagiere pro Flug (FLIGHTS)

[illegible]

Sobald ein Passagier als WANTED klassiert ist (reated) soll in rot der Text **WANTED Police is coming!** Angezeigt werden.

people#new (<http://localhost:5050/people/new>)

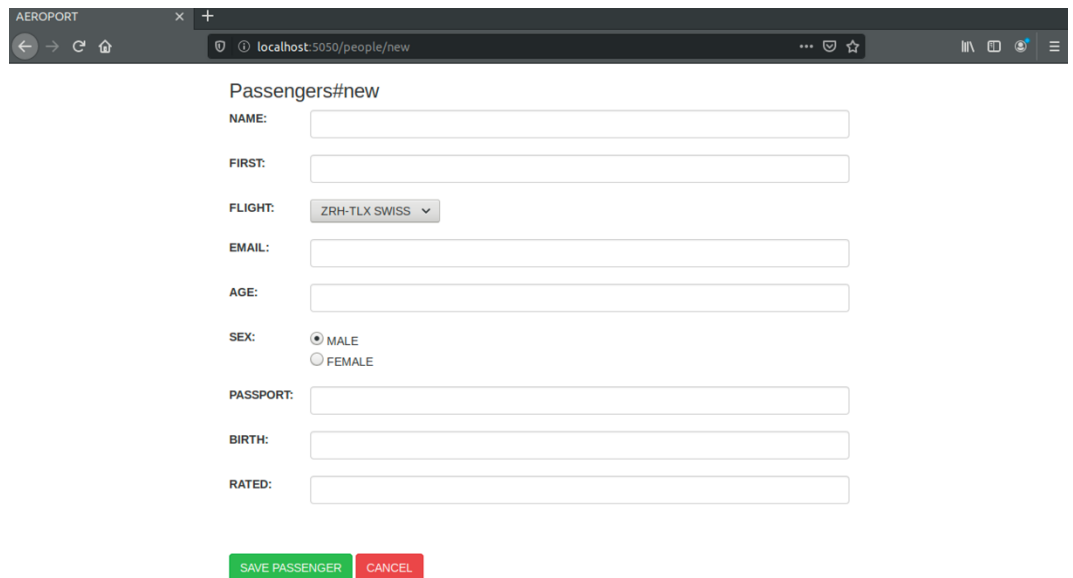
Die View `people#new` beinhaltet:

Ein Formular mit den Feldern NAME, FIRST, FLIGHT, EMAIL, AGE, SEX, PASSPORT, BIRTH und RATED, wobei

- › NAME := `Person.name`
- › FIRST := `Person.firstname`
- › FLIGHT := `Person.group.groupname`; dargestellt als `collection_select` Element
- › EMAIL := `Person.email`
- › AGE := `Person.age`
- › SEX := `Person.sex`; dargestellt als `radio_button` Element
- › PASSPORT := `Person.des1`
- › BIRTH := `Person.des2`
- › RATED := `Person.des3`

Einen «SAVE PASSENGER»-Buttonsolol die view `people#index` rendern. Der neu erfasste Passagier muss auf `people#index` angezeigt werden

Ein «CANCEL»-Button führt auf `people#index` zurück



The screenshot shows a web browser window with the address bar displaying `localhost:5050/people/new`. The page title is "Passengers#new". The form contains the following fields and controls:

- NAME:
- FIRST:
- FLIGHT:
- EMAIL:
- AGE:
- SEX: ☒ MALE ☐ FEMALE
- PASSPORT:
- BIRTH:
- RATED:

At the bottom of the form, there are two buttons: a green "SAVE PASSENGER" button and a red "CANCEL" button.

Erfassen Sie fünf Passagiere.

people#edit (<http://localhost:5050/people/<id>/edit>)

DIE EDIT VIEW GIBT NUR 2 PUNKTE.

Geraten Sie in Zeitnot, lassen Sie die people#edit View weg.

Die View people#edit beinhaltet die korrekt ausgefüllten Felder NAME, FIRST, FLIGHT, EMAIL, AGE, SEX, PASSPORT, BIRTH und RATED, wobei

- › NAME := Person.name
- › FIRST := Person.firstname
- › FLIGHT := Person.group.groupname; dargestellt als collection_select Element
- › EMAIL := Person.email
- › AGE := Person.age
- › SEX := Person.sex; dargestellt als radio_button Element
- › PASSPORT := Person.des1
- › BIRTH := Person.des2
- › RATED := Person.des3

Einen «SAVE PASSENGER»-Buttonsolol die view people#index rendern. Der aktualisierte

The screenshot shows a web browser window with the address bar displaying 'localhost:5050/people/16/edit'. The page title is 'Passengers#edit'. The form contains the following fields and values:

- NAME: Xi Pong
- FIRST: Pi
- FLIGHT: ZRH-TLX SWISS (dropdown menu)
- EMAIL: Pi@Po.JP
- AGE: 17
- SEX: FEMALE (radio button selected)
- PASSPORT: JP))SEZZ
- BIRTH: 04.04.2003
- RATED: Funny

At the bottom of the form, there are two buttons: 'SAVE PASSENGER' (green) and 'CANCEL' (red).

Passagier muss auf people#index angezeigt werden

Ein «CANCEL»-Button führt auf people#index zurück

Bewertungstests

Name: _____

Vorname: _____

No.	Bewertungskriterium	Erreichte Pte.
1	<p>Server Start: Wechseln Sie im Terminal in das Verzeichnis <code>home/vmadmin/worksapce/BudgetBook</code></p> <p>Eingabe:</p> <pre>sudo kill \$(cat tmp/pids/server.pid) rails s</pre> <p>Erwartetes Resultat: Rails Server startet und gibt folgende zwei letzte Zeilen aus: * Listening on <code>tcp://0.0.0.0:5050</code> Use Ctrl-C to stop</p>	/1
2	<p>Model Klassen vorhanden Neues Terminal öffnen und in das Verzeichnis <code>home/vmadmin/worksapce/portfolio</code> wechseln.</p> <p>Eingabe:</p> <pre>ll app/models</pre> <p>Erwartetes Resultat: Folgende Dateien sind vorhanden: 1. <code>person.rb</code> (1P) → <code>person singular</code> 2. <code>group.rb</code> (1P) → <code>group singular</code></p>	/2
3	<p>Beziehungen zwischen Models sind definiert</p> <p>Eingabe 1:</p> <pre>cat app/models/group.rb</pre> <p>Erwartetes Resultat:</p> <pre>class Index < ApplicationRecord has_many :people (1 P) → people plural end</pre> <p>Eingabe 2:</p> <pre>cat app/models/person.rb</pre> <p>Erwartetes Resultat:</p> <pre>class Share < ApplicationRecord belongs_to :group (1 P) → group singular end</pre>	/2
4	<p>Model Schema 1</p> <p>Eingaben:</p> <pre>rails c Person.connection Person</pre> <p>Erwartetes Resultat:</p> <pre>=> Person(id: integer, group_id: integer, name: string, firstname: string, email: string, age: integer, sex: boolean, des1: string, des2: string, des3: string, created_at: datetime, updated_at: datetime) (1P)</pre>	/1

-
- 5 **Model Schema 2** /1
Eingaben:
Group
Erwartetes Resultat:
=> Group(id: integer, groupname: string, code: string, des1: string, des2: string, des3: string, created_at: datetime, updated_at: datetime) (1P)
Bemerkung: Die rails console kann man exit verlassen.
-
- 6 **DB Schema** /1
Eingaben:
sqlite3 db/development.sqlite3
.tables
Erwartetes Resultat:
ar_internal_metadata groups people schema_migrations
(1P) Plural und Kleinschreibung!
Bemerkung: Die SQLITE Console kann man .exit verlassen.
-
- 7 **Controller categories_controller.rb** /2
Eingabe:
grep def app/controllers/groups_controller.rb
Erwartetes Resultat:
› def index (0.5P)
› def create (0.5P)
› def groups_params (0.5P)
› def destroy (0.5P)
Bemerkung: Reihenfolge kann variieren
-
- 8 **Controller people_controller.rb** /3
Eingabe:
grep def app/controllers/people_controller.rb
Erwartetes Resultat:
› def index (0.5P)
› def new (0.5P)
› def create (0.5P)
› def edit (0.5P)
› def update (0.5P)
› def destroy (0.5P)
› def people_params
-

AEROPORT

Flights Aeroport

FLIGHT

CODE

FROM/TO

DEPARTURE Datetime

ARRIVAL Datetime

Flights	CODE	FROM/TO	DEPARTURE Datetime	DESTINATION Datetime	
ZRH-TLX SWISS	LX 3005	ZURICH ZRH / BERLIN TLX	08:30AM SATURDAY-30-MAY-2020	10:15AM SATURDAY-30-MAY-2020	<input type="button" value="Delete"/>
TKY-SYD JPAIR	JP 6567	TOKYO / SYDNAY	18:30PM SATURDAY-04-APRIL-2020	19:50PM SATURDAY-05-APRIL-2020	<input type="button" value="Delete"/>
MOS-MOS RUAIR	RU 99.9	MOSCOW / MOSCOW	18:30PM SATURDAY-11-APRIL-2020	18:30PM SATURDAY-11-APRIL-2020	<input type="button" value="Delete"/>
LHR-ZRH SWISS	LX 0404	LONDON LHR / ZURICH ZRH	08:30AM SATURDAY-04-APRIL-2020	10:15AM SATURDAY-04-APRIL-2020	<input type="button" value="Delete"/>
ISB-ZRH SWISS	LX 9991	ISLAMABAD ISB / ZURICH ZRH	11:30AM MONDAY-20-APRIL-2020	22:30PM MONDAY-20-APRIL-2020	<input type="button" value="Delete"/>

Number of flights: 5

- › Die view groups#index listet alle Flüge mit den Spaltenbezeichnungen **FLIGHT, CODE, FROM/TO, DEPARTURE Datetime und ARRIVAL** auf. (1P)
- › 3 Testdatensätze sind erfasst (1P)
- › «Anzahl Flüge» ist korrekt am Ende der Tabelle (1P)
- › Flug erfassen funktioniert (neuer Testdatensatz X) (1P)
- › Flug löschen funktioniert (Testdatensatz X) (1P)
- › «Passengers» geht auf <http://localhost:5050/people/index> (1P)

Nr	FLIGHT	NAME	FIRST	EMAIL	AGE	SEX	PASSPORT	BIRTH	RATED	EDIT	DESTROY
146640	ISB-ZRH SWISS	Affenhals	Musstaffa	AFF@AST.SU	20	MALE	BB #1#21U	10.09.1999	Funny	Edit	Löschen
87984	ZRH-TLX SWISS	Biancini	Hans-Ueli	HB@BH.CH	70	MALE	CH Z4GT34	16.06.1951	Aggressiv	Edit	Löschen
48880	ZRH-TLX SWISS	Bummle	Armando Jose	ajm75@gm.ru	45	MALE	LI 4TG51X	23.04.1975	Lovely	Edit	Löschen
117312	LHR-ZRH SWISS	Freundkol	Bambawa	Bam@Want.ba	22	MALE	SY BAS3XY	01.01.1997	Crazy	Edit	Löschen
136864	ZRH-TLX SWISS	Gobletti	Cobreto	GC@UHUH.UU	18	MALE	UU 2134HT	18.02.2001	WANTED Police is coming!	Edit	Löschen
78208	ZRH-TLX SWISS	Gogolov	Vreni	gv@gu.ch	20	FEMALE	CH U786T4	14.12.2000	Funny	Edit	Löschen
97760	LHR-ZRH SWISS	Hupfi	Barbara	HB2@BH.CH	30	FEMALE	CH ZU55ZU	19.11.1990	Complicated	Edit	Löschen
166192	LHR-ZRH SWISS	Li	Fi	Fo@fo.jp	40	MALE	JP #####	03.09.1980	Spass***	Edit	Löschen
127088	LHR-ZRH SWISS	Mills	Bill	Mills@Bills.uk	33	MALE	UK BR2341	06.09.1986	Lovely	Edit	Löschen
107536	LHR-ZRH SWISS	Pirelli	Mirelli	PM@MP.IT	40	MALE	IT IT77IT	14.07.1980	WANTED Police is coming!	Edit	Löschen
156416	ZRH-TLX SWISS	Xi Pong	Pi	Pi@Po.JP	17	FEMALE	JP))SEZZ	04.04.2003	Funny	Edit	Löschen

NUMBER OF PEOPLE: 11

AVERAGE AGE: 32.2727272727273

FLIGHT: 5 PASSENGERS ON ZRH-TLX SWISS
5 PASSENGERS ON LHR-ZRH SWISS
1 PASSENGERS ON ISB-ZRH SWISS
0 PASSENGERS ON TKY-SYD JPAIR
0 PASSENGERS ON MOS-MOS RUAIR

ADD Passenger FLIGHTS

- › 3-5 Testdatensätze sind erfasst (siehe Screen) (1P)
- › Anzahl erfasste Datensätze ist mit korrektem Resultat am Ende der Tabelle (1P)
- › Durchschnittsalter aller Passagiere ist mit korrektem Resultat am Ende der Tabelle (1P)
- › Anzahl Passagiere sind pro Flug mit korrektem Resultat am Ende der Tabelle (1P)
- › Einen Passagier erfassen funktioniert (neuer Testdatensatz X) (1P)
- › Passagier löschen funktioniert (Testdatensatz X) (1P)
- › «FLIGHTS» geht auf <http://localhost:5050/groups/index> (1P)
- › Sobald ein Passagier als WANTED gerated ist, erscheint roter Warntext (1P)

<http://localhost:5050/people/new/> entspricht:

/5

Passengers#new

NAME:

FIRST:

FLIGHT: ZRH-TLX SWISS ▾

EMAIL:

AGE:

SEX: ☒ MALE ☐ FEMALE

PASSPORT:

BIRTH:

RATED:

- › Alle Felder vorhanden (1P)
- › Gruppen als Dropdown (1P)
- › Sex als Radiobutton (1P)
- › Speichern funktioniert (1P)
- › Speichern führt auf people/index (1P)

<http://localhost:5050/people/edit/> entspricht:

/2

Passengers#new

NAME:

FIRST:

FLIGHT: ZRH-TLX SWISS ▾

EMAIL:

AGE:

SEX: ☒ MALE ☐ FEMALE

PASSPORT:

BIRTH:

RATED:

- › FLIGHT\$ ausgefüllt (1P)
- › Radiobutton korrekt ausgewählt (1P)

Total:

/36

Note:

Visum Experte: