Aufgabenverteilung bei der Fallstudie Kinoticketbuchungssystem  
Team: SuperDuperTrooperDeluxe3D

Front-End: Manuel Meixner, Patrick Honert, Hendrik Reidel

* Startseite: Manuel Meixner
* Filmübersicht: Hendrik Riedel
* Header(3x)/Footer: Manuel Meixner
* Filminformationen: Hendrik Riedel
* Buchungsseite Hendrik Riedel
* Warenkorb: Hendrik Riedel, Manuel Meixner
* Login/Register: Patrick Honert
* CheckoutSeiten (4x): Hendrik Riedel
* Kontakt und Preise: Patrick Honert
* Datenschutz, Impressum: Alle oben genannten

Back-End: Munir Abobaker, Mohammed Chalabi, (Oliver Gärtner[Unit Tests])

* Datenbankstruktur: Mohammed Chalabi
* Serveraufbau und Routing: Munir Abobaker
* Funktionen des Back-Ends: Mohammed Chalabi, Munir Abobaker
* (Unit Tests: Oliver Gärtner)

Koordination (Schnittstellen) zwischen Front- und Backend: Hendrik Riedel, Munir Abobaker

Design, Präsentation, Dokumentation, User Story Mapping, Mockuperstellung: Lea Noll

Ausgeschieden: Saleh Chaaban, Oliver Gärtner

Anmerkung: In dieser Gruppe wurden während der Bearbeitungszeit 9 Nachklausuren geschrieben. Bei einigen von uns ging es dabei darum ob sie exmatrikuliert werden oder nicht. Daher war die Belastung in diesem Semester bei uns relativ hoch.

Leider wurde Oliver Gärtner zum Anfang der Klausurenphase exmatrikuliert ohne uns ein Wort davon zu berichten. Als wir dies zum Ende der Klausurenphase erfuhren, entschied das Team ohne meine Einstimmung, die Unit Tests nicht fertig zu stellen. Ich habe diesem Dokument die „Unit Tests“ angehangen, die ich während der Erstellung des Front-Ends erstellt habe.

Schöne Grüße Hendrik Riedel

var expect  = require('chai').expect;

var request = require('request');

const app = require('../app.js');

it('Main page status', function(done) {

    request('<http://localhost:3000>' , function(error, response, body) {

        expect(response.statusCode).to.equal(200);

        done();

    });

});

it('Films status', function(done) {

  request('<http://localhost:3000/films>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

/\*it('FilmItem status', function(done) {

  request('<http://localhost:3000/filmItem>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

}); \*/

it('Contact status', function(done) {

  request('<http://localhost:3000/contact>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('Preis status', function(done) {

  request('<http://localhost:3000/pricing>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('login status', function(done) {

  request('<http://localhost:3000/login>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('register status', function(done) {

  request('<http://localhost:3000/register>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('check status', function(done) {

  request('<http://localhost:3000/check>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('noCheck status', function(done) {

  request('<http://localhost:3000/noCheck>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('cashReservation status', function(done) {

  request('<http://localhost:3000/cashReservation>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

/\*it('booking status', function(done) {  //param übergeben

  request('<http://localhost:3000/booking>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

}); \*/

/\*it('payment status', function(done) {  //param übergeben

  request('<http://localhost:3000/payment>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

}); \*/

it('Impressum status', function(done) {

  request('<http://localhost:3000/impressum>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

it('Datenschutz status', function(done) {

  request('<http://localhost:3000/datenschutz>' , function(error, response, body) {

      expect(response.statusCode).to.equal(200);

      done();

  });

});

/\*it('OK, creating a new note works', (done) => {

  request(app).post('/register')

    .send({ username: 'NOTE', password: "AAA" })

    .then((res) => {

      const body = res.body;

      expect(body).to.contain.property('\_id');

      expect(body).to.contain.property('username');

      expect(body).to.contain.property('password');

      done();

    })

    .catch((err) => done(err));

}); \*/