

Duale Hochschule Baden-Württemberg Mannheim

Projektdokumentation

Webprogrammierung

Studiengang Wirtschaftsinformatik

Data Science

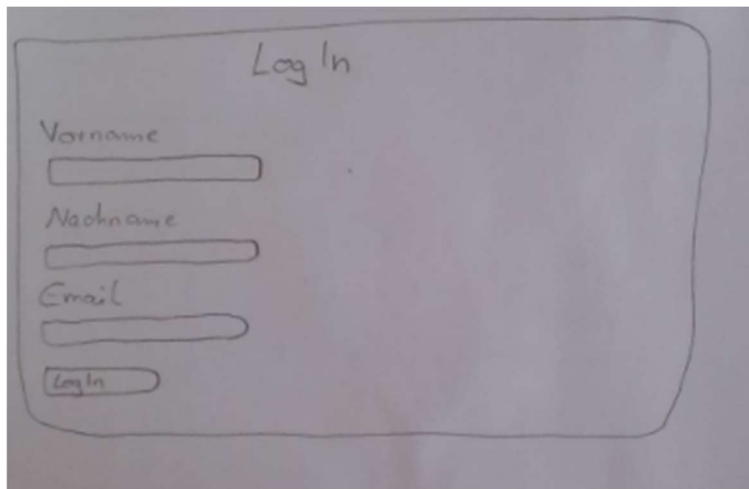
Verfasser(in):	Hannah Schult (5373022)
Kurs:	WWI-20-DSB
Vorlesung:	Webprogrammierung
Dozent:	Jonas Heuer
Bearbeitungszeitraum:	21.01.2022 – 28.02.2022

Inhalt

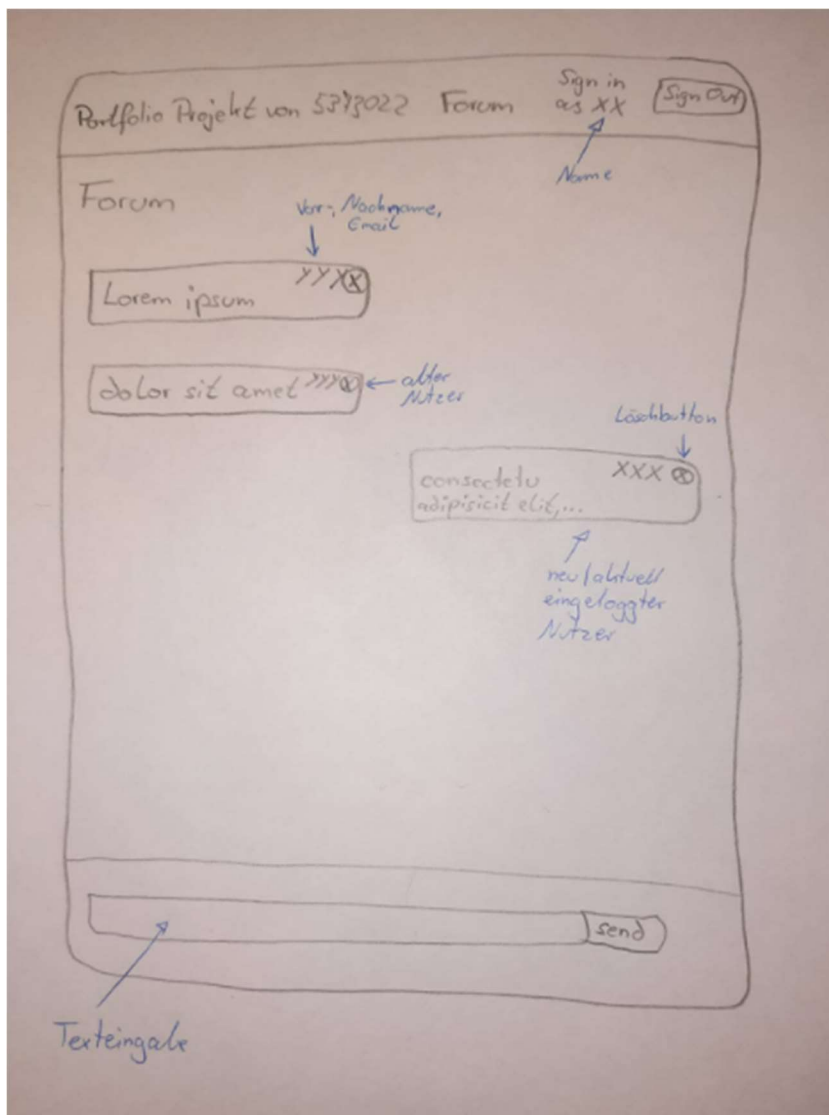
1	Paper Prototyping.....	1
2	Klassendiagramm.....	2
3	Requirements.....	3
4	Konzept.....	4
5	Resümee.....	6

1 Paper Prototyping

LogIn Seite / Startseite

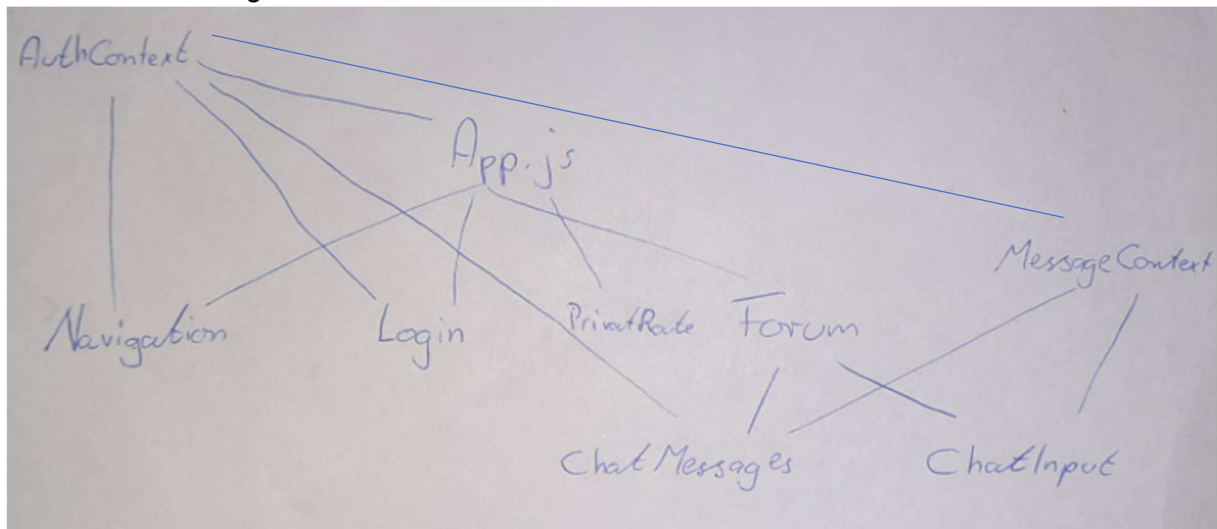


Forum

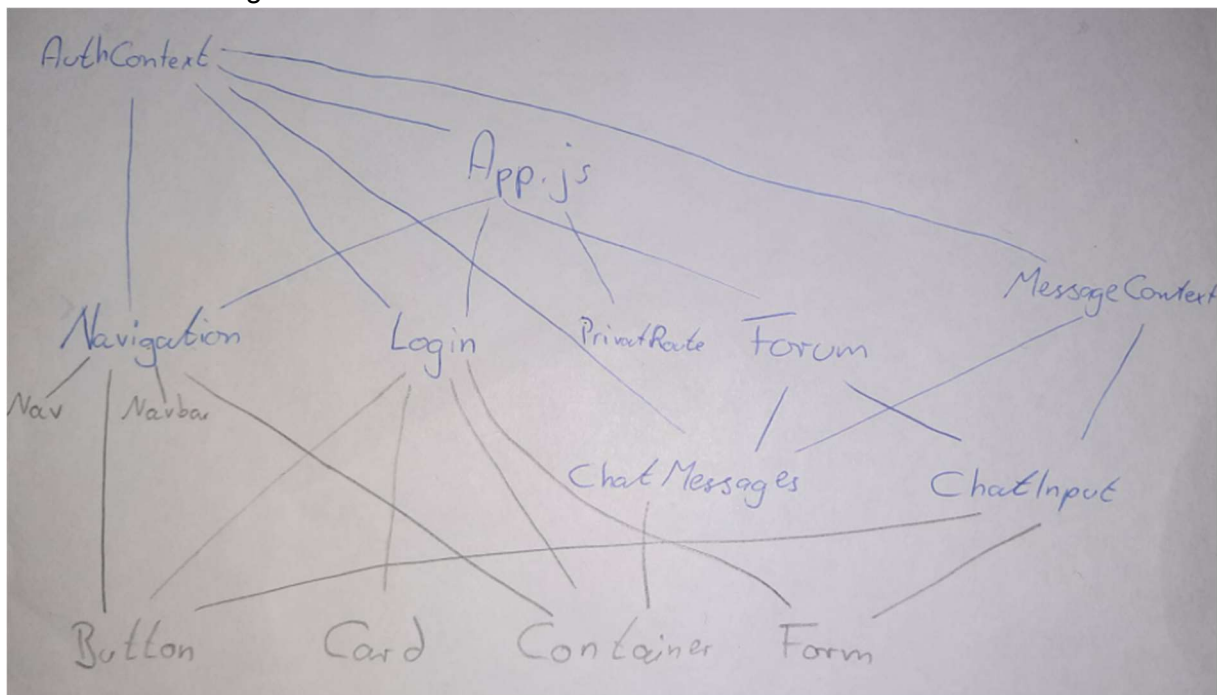


2 Klassendiagramm

1. Klassendiagramm



2. Klassendiagramm



Das erste Klassendiagramm ist das Klassendiagramm mit den Komponenten, die ich selbst gemacht habe.

Beim zweiten Klassendiagramm habe ich mit Bleistift noch die Komponenten ergänzt, die ich über Bootstrap importiert habe.

3 Requirements

Es wird eine Webseite mit zwei Seiten erstellt. Die erste Seite ist die LogIn Seite. Dort meldet man sich mit Vornamen, Nachnamen und Email-Adresse an. Wenn man auf den LogIn Button klickt wird der Vor- und Nachname als Parameter übernommen.

Nach Anmeldung landet man auf der zweiten Seite – dem Forum. Auf das Forum kann man durch die Authorisierung erst zugreifen bzw. sehen, wenn man eingeloggt ist. Es gibt einen LogOut Button und wenn man auf den klickt landet man wieder auf der LogIn Seite und kann sich mit einem anderen Namen einloggen.

Auf dem Forum kann der Nutzer Beiträge in Form von Texten posten. Die Texte werden mit Namen und Email-Adresse des Nutzers versehen. Der angemeldete Nutzer kann seine eigenen Texte selbst wieder löschen. Wenn der Nutzer sich abmeldet und mit anderem Namen wieder einloggt, dann stehen seine Texte immer noch im Forum.

Außerdem ist oben eine Navigationsleiste vorhanden, um zwischen Forum und LogIn hin und her zu switchen.

4 Konzept

Components

ChatInput

- ➔ Hier ist das Texteingabefeld mit der Funktion die Forumsbeiträge des Nutzers zu posten.
- ➔ Die Komponente „Form“ von Bootstrap ist das Texteingabefeld und mit klick auf den „Button“ von Bootstrap wird der Text gesendet.

ChatMessages

- ➔ Hier sind die geposteten Forumsbeiträge und werden hier mit Vor-, Nachnamen und Email-Adresse des Nutzers versehen.
- ➔ Die Beiträge werden mit der Komponente „Container“ von Bootstrap visualisiert.
- ➔ Mit klick auf die kleine Tonne kann der aktive Nutzer seinen Beitrag löschen.

Navigation

- ➔ Das ist die Navigationsleiste zur Navigation zwischen dem Forum und der Login Seite.
- ➔ Die Komponenten „Navbar“ und „Nav“ sind wieder von Bootstrap.
- ➔ Mit klick auf den „Button“ Sign Out kann man sich ausloggen.

PrivateRoute

- ➔ „Versteckt“ die Forumseite, solange, bis der Nutzer angemeldet ist.

Context

AuthContext

- ➔ Macht, dass der Vor-, Nachname und die Email-Adresse des Nutzers global verwendet werden können.

MessageContext

- ➔ Definiert die Funktionen rund um die Forumsbeiträge (getMessage, sendMessage, deleteMessage, compareMessage) und macht, dass die Funktionen global verwendet werden können.

Pages

Forum

- ➔ Hier können die Forumsbeiträge gepostet werden und die anderen Forumsbeiträge gelesen werden.

LogIn

- ➔ Hier ist die LogIn Seite, auf der sich der Nutzer mit Namen und Email Adresse einloggen kann.
- ➔ Für die visualisieren wurden wieder die Komponenten „Container“ und „Card“ von Bootstrap genutzt.
- ➔ Die drei Texteingabefelder für Vornamen, Nachnamen und Email bestehen aus der Komponente „Form“ aus Bootstrap.
- ➔ Mit Klick auf den „Button“ Log In kann der Nutzer sich einloggen.

5 Resümee

Ich fand das Projekt insgesamt anspruchsvoll. Die LogIn Seite und die Navigationsleiste mit Routing waren gut machbar, da wir dafür die Grundlagen in der Vorlesung gut vorbereitet hatten. Allerdings fand ich die Forumsbeiträge sehr schwer umsetzbar, da wir zwar im Geburtstagskalender einen Text posten konnten, aber dieser Text nicht gespeichert wurde. An dieser Stelle habe ich lange gebraucht das Forum zu programmieren.

Außerdem bin ich von der vorgegeben Ordnerstruktur abgewichen, da ich mit Bootstrap gearbeitet habe und da der Style schon direkt im Code ist. Außerdem finde ich es so übersichtlicher, wenn das benötigte css in der js.Datei zu finden ist. So konnte ich viel schneller die benötigten Stellen finden und bearbeiten.

Bei größeren Projekten würde ich trotzdem wieder die Ordnerstruktur verwendet, da man dann das Design vllt. öfter an anderen Stellen nochmal einsetzen kann. Aber bei diesem Projekt, bei dem ich insgesamt auch nicht besonders viel css benutzt habe, fand ich es einfachere ohne so viele css-Dateien.

Die Vorgabe zur Umsetzung einer flexbox oder grid Datei in einer extra css habe ich in der index.css umgesetzt.