

Duale Hochschule Baden-Württemberg Mannheim

Portfolio-Prüfung
Web-Programmierung

Studiengang Wirtschaftsinformatik
Studienrichtung Data Science

Verfasser(innen):	Steven Schneider
Matrikelnummer:	9147121
Kurs:	WWI21DSB
Studiengangsleiter:	Prof. Dr. Bernhard Drabant
Prüfer:	Jonas Heuer
Bearbeitungszeitraum:	20.01.2023 – 28.02.2023

Inhaltsverzeichnis

Paper Prototype	3
<i>Kalender</i>	<i>3</i>
<i>To-Do Liste.....</i>	<i>3</i>
Klassendiagramm	4
Requirements	5
Konzept	6
Resümee	7

Paper Prototype

Kalender

header

weekday names

week 4-5 per month

→ number

→ To-Do Link

To-Do Liste

number + link

header

form input field

add button

checkbox

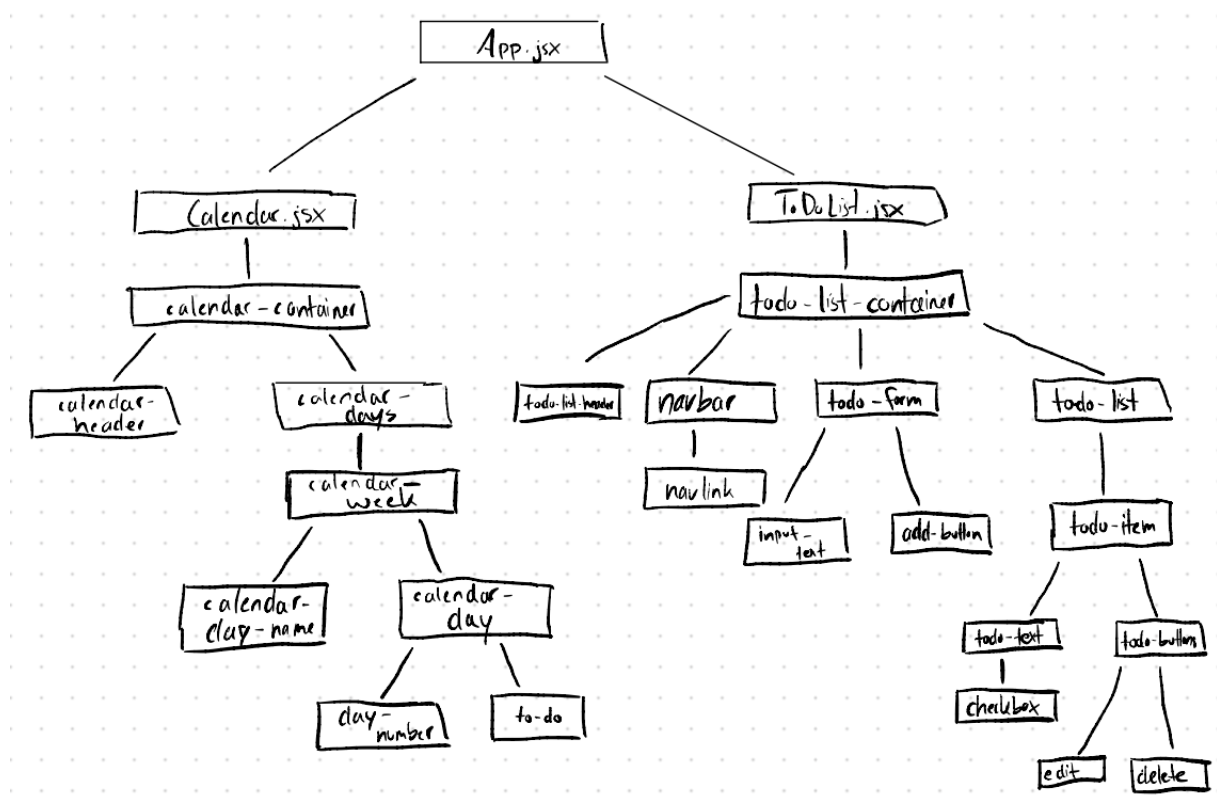
to-do buttons

to-do text

to-do item

list

Klassendiagramm



Requirements

Die Anforderungen an die zu erstellende Web-App wurden vom Dozenten Jonas Heuer vorab festgelegt. Die Web-App soll Folgendes beinhalten:

- Flexbox oder Grid
- Verwendung separater CSS-Dateien
- React.Router
- Conditional & List-Rendering
- useState
- useParams
- Ausschließlich functional based Components
- Props & Two-Way Binding
- Sinnvolles Naming
- Codequalität
- Funktionalität

Mit diesen Komponenten gilt es eine Web-App zu erstellen, die aus mindestens zwei Seiten besteht. Die Startseite soll ein Kalender in einer Monatsansicht sein. Bei Klick auf einen der Tage soll der Nutzer in eine Tagesansicht gelangen, in der die To-Dos des Tages sichtbar sind. In der To-Do Liste sollen neue To-Dos hinzugefügt, gelöscht oder abgehakt werden können. Die Anzahl der To-Dos soll in der Monatsansicht im Feld des jeweiligen Tages dargestellt werden. Außerdem gilt es eine Navigationsleiste zu erstellen, die zur Navigation zwischen Monats- und Tagesansicht dienen soll.

Konzept

Die Web-App besteht wie gefordert aus zwei Views. Einer Kalenderansicht, die als Startseite dient und den aktuellen Monat darstellt und einer Tagesansicht, in der es möglich ist, To-Dos für den jeweiligen Tag zu hinterlegen. Beide Views werden in der App.jsx als Routen festgelegt.

Die Kalenderansicht besteht aus einer Überschrift, die den Namen des aktuellen Monats und das aktuelle Jahr angibt. Die Daten, die zur Bestimmung des aktuellen Monats benötigt werden, können über die Moment-Library importiert werden. Mithilfe von flexbox soll der typische Aufbau eines Kalenders dargestellt werden. Die Namen der Wochentage sollen zur besseren Orientierung über den Kalendertagen platziert werden. Für jeden Tag einer Woche wird eine Spalte erstellt, insgesamt also sieben Spalten für Montag bis Sonntag. Außerdem wird für jede Woche eines Monats eine Reihe mit jeweils sieben Spalten dem Kalender hinzugefügt. Zum Schluss wird die Nummer des Tages in dem jeweiligen Feld des Kalenders angegeben und darunter wird ein Text platziert, der die Anzahl der hinterlegten To-Dos des Tages angibt. Mithilfe der Map-Funktion kann sichergestellt werden, dass die Nummern in die richtigen Felder platziert werden. Um bei einem Klick auf den Tag in die richtige Tagesansicht zu springen, wird ein Link erstellt, dem der entsprechende Pfad hinterlegt wird.

An der Spitze der Tagesansicht soll sich eine Navigationsleiste befinden, die sich farblich vom Rest der Webseite abhebt. Ein Link, der links in der Navigationsleiste platziert wird, soll dem Nutzer wieder zur Kalenderansicht zurückbringen. Darunter befindet sich die Überschrift der To-Do Liste. Der Kern der To-Do Liste besteht aus dem Input-Formular, in dem der Nutzer die To-Dos eintragen und mit Klick auf dem Button „Add“ abspeichern kann. Diese To-Do Items werden nun darunter angezeigt und können darauffolgend bearbeitet, abgehakt oder gelöscht werden. Um zu gewährleisten, dass die Items auch nach aktualisieren der Seite gespeichert sind, wird der lokale Speicher des Browsers verwendet. Um die einzelnen Tagesansichten mit ihren zugehörigen To-Dos voneinander zu unterscheiden, muss mit einer ID gearbeitet werden. Am besten eignet sich dazu voraussichtlich die Nummer des Tages, bzw. das Datum.

Das Design der Web-App soll schlicht und funktional gehalten werden. Hierzu werden separate CSS-Dateien erstellt, die die Styles von Kalender und To-Do Liste festlegen und verfeinern.

Resümee

Die Erstellung der Kalender-Komponente ging gut von der Hand, wenn auch nicht ohne Probleme. Gerade die Berücksichtigung der Anzahl an To-Dos für jeden Tag war nicht ohne Weiteres realisierbar. Schlussendlich war die Lösung, den lokalen Speicher über eine ID aufzurufen. Allerdings ist anzumerken, dass in der derzeitigen Logik die To-Dos nur mit dem Tag als ID im lokalen Speicher hinterlegt werden. Wenn beispielsweise für den 27. Tag eines Monats To-Dos hinterlegt werden, dann werden diese To-Dos auch für 27. Tag der nächsten Monate angezeigt. Dieser Bug konnte im Rahmen dieser Portfolio-Prüfung nicht mehr behoben werden und ist auch für die Aufgabenstellung irrelevant, da das Ziel war, nur für einen Monat eine Ansicht zu erstellen. Die Lösung wäre vermutlich die ID über die To-Dos gespeichert werden, nicht mehr nur mit dem Tag anzusteuern, sondern über das komplette Datum.

Die Tagesansicht gestaltete sich im Vergleich zur Kalenderansicht deutlich schwieriger. So war eine Reihe von Funktionen zu erstellen, die die gewünschten Funktionen (Hinzufügen, Abhaken und Löschen) abdecken. Zusätzlich wurde noch eine Möglichkeit zur Bearbeitung eines To-Dos geschaffen. Wie auch schon bei der Kalenderansicht zuvor, war das Data Handling die komplexeste Aufgabe, die es zu lösen galt. Anfangs wurden To-Dos nach jedem Aktualisieren der Seite gelöscht. Als das gelöst war, wurden für jeden Tag die gleichen To-Dos angezeigt. Schlussendlich konnte mithilfe von useParams der Parameter „day“ für das Data Handling verwendet werden. So wurden schließlich für jeden Tag die richtigen To-Dos abgespeichert und angezeigt.

Die finale Version der Web-App erfüllt die im Kapitel Requirements genannten Anforderungen und ist wie gewünscht zu Nutzen. Auf der Startseite, also der Kalenderansicht, erhält der Nutzer einen Überblick über den aktuellen Monat und der Anzahl an hinzugefügten To-Dos. Durch einen Klick auf einen der Tage gelangt man in die dazugehörige Tagesansicht. Dort sind die gewünschten Funktionen Hinzufügen, Abhaken und Löschen der To-Dos möglich. Verbesserungswürdig ist, neben dem bekannten Bug, das Design der Web-App, welche vor allem auf der Tagesansicht sehr funktional und schlicht ist. Schön wäre hier noch wenn in der Überschrift der Tag in dem man sich befindet angezeigt werden würde.