Duale Hochschule Baden-Württemberg Mannheim

Web-Entwicklung Portfolioprüfung

3. Semester

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Verfasser:	Dominic Viola
Matrikelnummer:	siehe Moodle Abgabe
Kurs:	WWI-19-DSB
Studiengangsleiter:	siehe Moodle Abgabe
Prüfender Dozent:	siehe Moodle Abgabe
Abgabeschluss:	28.02.2021

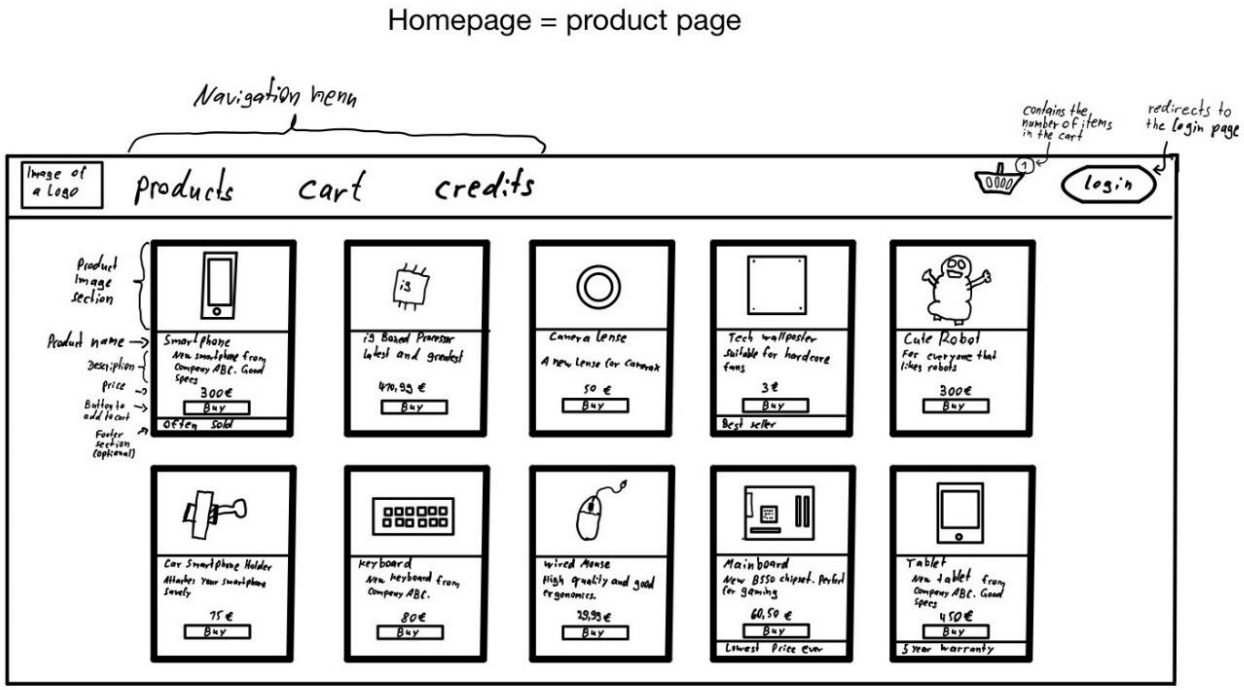
Requirements

Die Anforderungen dieser Ausarbeitung lauten wie folgt:

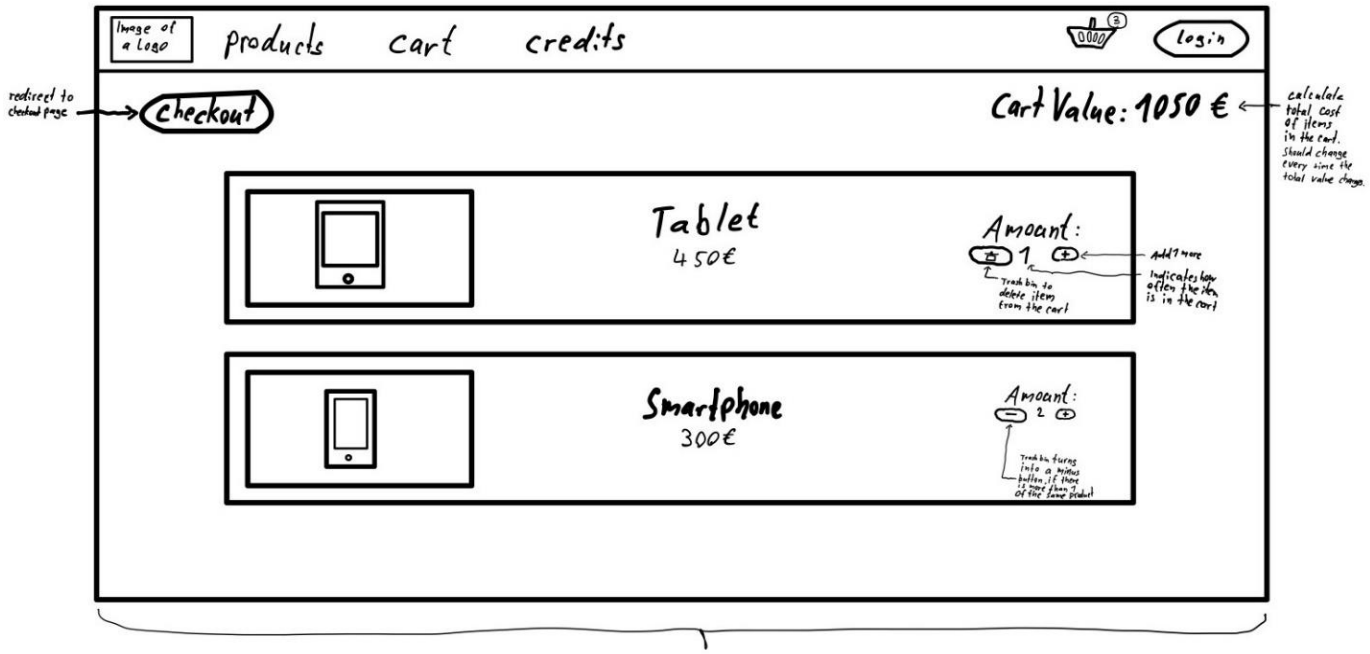
- Erstellung eines Onlineshops mit mindestens **drei Seiten**, diese Seiten sind: Produktseite, Warenkorbseite und eine Checkoutseite**.**
- Die **Produktseite** soll mindestens 10 unterschiedliche Produkte anzeigen, die in beliebig Reihenfolge und beliebig oft in den Warenkorb gelegt werden können.
- Die **Warenkorbseite** soll diese Produkte anzeigen und die Möglichkeit bieten diese wieder aus dem Warenkorb zu löschen.
- Die **Checkoutseite** soll ein Kontaktformular beinhalten, die Eingabefelder für Name, Adresse, E-Mail und zum Absenden beinhalten muss.
- Es soll keine **Datenbank** Anbindung erfolgen.
- Die Webseite muss mit der **React Library** umgesetzt werden. Dabei sollen sowohl functional als auch class based Components eingesetzt werden, die einen State besitzen, Props übergeben bekommen und mit Hilfe des BrowserRouters eingebunden werden können.
- Visuelle Elemente sollen mit dem Flexbox und dem Grid Layout angeordnet werden. Des Weiteren sollen Elemente aus einer Liste dynamisch angezeigt werden und es muss mindestens einmal conditional rendering durchgeführt werden
- Das Projekt soll mit GitHub versioniert werden und mindestens 10 Commits aufweisen

Paperprototype

Basierend auf den Requirements wurde ein Paperprototype der einzelnen Seiten angefertigt, der als Grundlage für die Realisierung der Webseite dienen soll. Beschreibungen an den Elementen sollen die Funktion dieser verdeutlichen. Die Beschreibungen und Inhalte sind auf Englisch, da die Webseite in englischer Sprache sein wird.

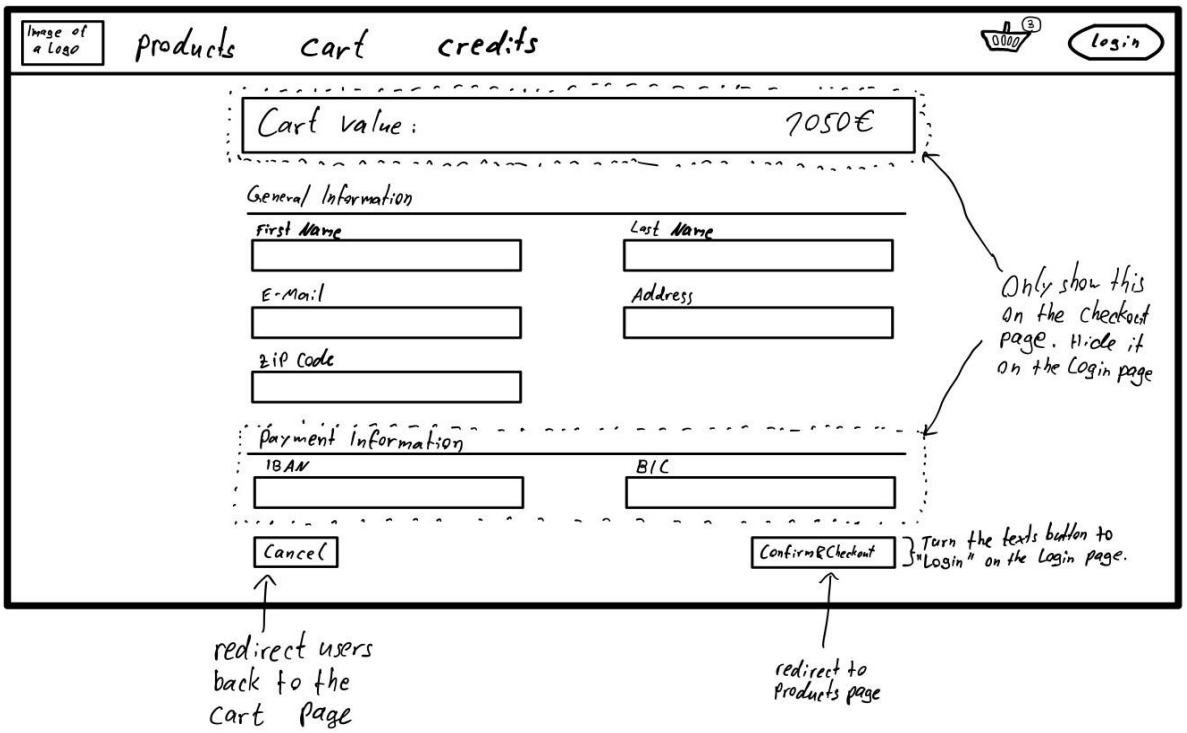


Cart page

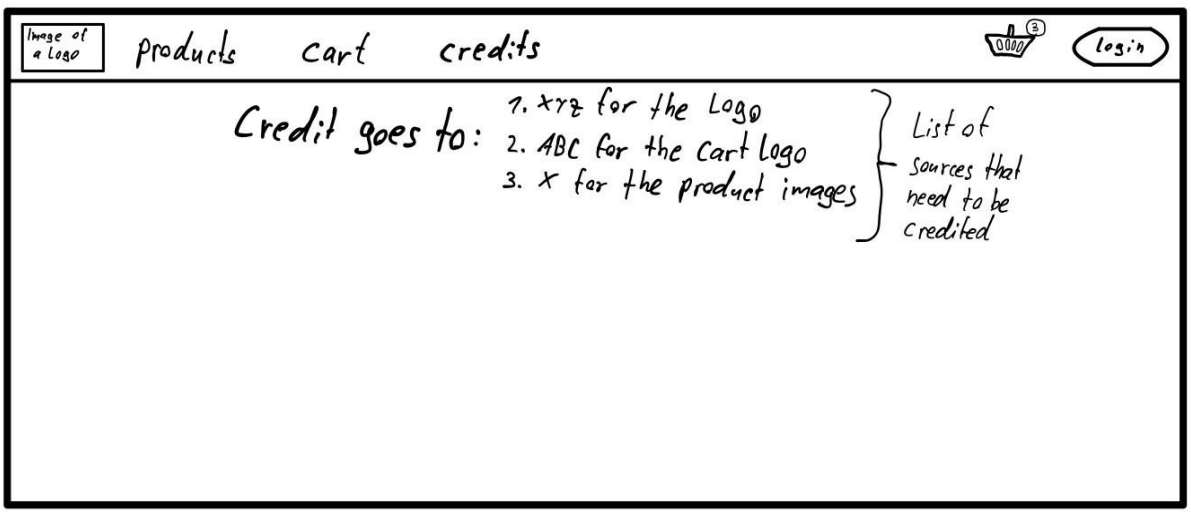


If the cart is empty: show a text, that informs the user about the cart being empty

Checkout/Login page



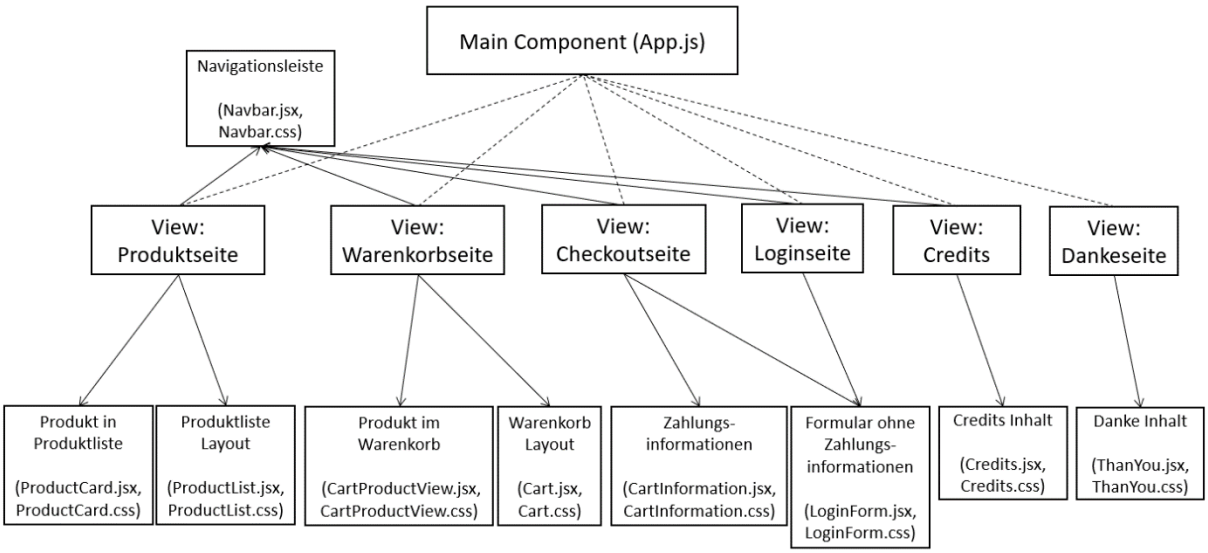
Credits page



Thank you page

Konzept

Das nachfolgende Klassendiagramm dient der Verdeutlichung der Views und den zugehörigen Components, jede Component hat dabei eine gleichnamige .css-Datei, welche für die visuelle Gestaltung der Inhalte zuständig ist. Diese sollen das Erscheinungsbild des Paperprototypes abbilden. Im Anschluss werden die technische Umsetzung und die Funktion der einzelnen Components genauer erläutert.



Main Component (App.js):

Ist eine functional Component, die alle anderen Components mit Hilfe des BrowserRouters einbinden kann. Sie speichert den State, der zwischen allen Components über Props ausgetauscht werden kann. Im State befindet sich eine Liste aller Produkte, eine Liste der Produkte im Warenkorb und ein Objekt, in dem die Login Daten des Benutzers gespeichert werden. Außerdem beinhaltet sie Funktionen, zum Beispiel um Produkte zur Warenkorbliste hinzuzufügen, die in mehreren Components verwendet werden und über Props an die Child-Components weitergegeben werden.

Navigationsleiste:

Die Navigationsleiste ist eine functional Component und dient der Navigation zwischen den Seiten. Sie besitzt ein Flexbox Layout mit folgenden Elementen: Ein Logo, das auf die Startseite weiterleitet, Navitems für Produkte, Warenkorb und Credits, einen Login Button und ein Warenkorb Icon. Wählt man eines dieser Elemente aus, so wird die entsprechende Seite mit Hilfe des BrowserRouters dargestellt. Das Warenkorb Symbol dient nicht nur als Link zum Warenkorb, sondern zeigt außerdem die aktuelle Anzahl von Produkten im Warenkorb an.

Produktliste Layout / Produkt in Produktliste:

Die Produktseite ist eine functional Component, die jedes Produkt in einer functional Card Component visualisiert. Das Productlist Layout ordnet die Cards in einem Grid Layout an, während die Produkte in der Produktliste ihren Inhalt mit einer Flexbox untereinander anordnet. Die Informationen zu den Produkten stehen in der Produktliste, die im State der Main Component gespeichert ist und über Props zugänglich ist. Zu jedem Produkt in dieser Liste wird eine Card gerendert (List Rendering). Die Cards beinhalten ein Bild, einen Titel, eine Beschreibung, einen Preis und einen Button mit dem das Produkt in einer Warenkorbliste im State gespeichert wird. Zusätzlich dient eine conditional gerenderte Footer Section in der Card dazu zusätzliche Informationen darzustellen.

Warenkorb Layout / Produkt im Warenkorb:

Das Warenkorb Layout ist eine functional Component, sie greift über Props auf die Warenkorbliste zu und zeigt die Produkte darin in jeweils einer functional Component an. Ein Produkt im Warenkorb beinhaltet ein Bild, einen Titel, eine Beschreibung, einen Preis und eine Mengenanzeige. Die Mengenanzeige stellt dar, wie häufig das entsprechende Produkt im Warenkorb liegt, aber dient auch dazu mit einem Plus und einem Minus Button diese Anzahl zu verändern. Liegt die Menge eines Produktes bei 1 so kann die Menge nicht weiter reduziert werden und der Minus Button wird zu einem Mülleimer, der das Produkt aus der Warenkorbliste entfernt. Zusätzlich zu den Produkt Views zeigt die Warenkorbseite ganz oben den Gesamtpreis der Produkte im Warenkorb an und einen Checkout Button, der den Benutzer auf die Checkoutseite weiterleitet.

Formular ohne Zahlungsinformationen / Zahlungsinformationen:

Login- und Checkoutseite beruhen auf der gleichen class Component, die ein Formular ohne Zahlungsinformationen darstellt. Die Eingabefelder von Login und Checkout sind größten Teils identisch, auf der Checkout Seite müssen nur zusätzlich die Zahlungsinformationen eingeblendet werden. Die Zahlungsinformationen sind ebenfalls eine class Component, dessen Darstellung mit Hilfe von conditional rendering erfolgt, die Condition hierfür beruht auf dem URL-Pfad. Ist der Pfad also „/login“ so werden nur die Informationen zur Anmeldung abgefragt, ist der Pfad jedoch „/checkout“, so werden auch Zahlungsinformationen angezeigt. Sowohl die Zahlungsinformationen als auch das Formular verwenden ein Grid Layout. Die Eingaben des Benutzers werden bei Bestätigung im State der Main Component abgespeichert.

Danke Inhalt:

Hierbei handelt es sich um eine class Component, die nach Beendigung des Checkouts eine dankende Botschaft und Informationen zur Versanddauer für einige Sekunden angezeigt, währenddessen wird die Navigationsleiste ausgeblendet. Die Textelemente werden hierfür mit Flexbox angeordnet. Anschließend werden alle Informationen im State gelöscht und der Benutzer wird auf die Startseite zurückgeleitet.

Credits Inhalt:

Der Credits Inhalt ist eine functional Component und dient als Anerkennung der Künstler, deren Bilder und Icons verwendet werden. Die Informationen werden in einer geordneten Liste dargestellt und befinden sich in einer Flexbox.

Resümee

Allgemein war es möglich einen schnellen Überblick über die Aufgabenstellung und die benötigten Components zu erlangen. Es war allerdings zu Beginn schwierig zu verstehen, wie Props zwischen Components ausgetauscht und verwendet werden können. Nach einigen Tutorials war dies allerdings klar und ermöglichte einen einfacheren Überblick über Variablen im State, die in App.js gespeichert und über Props an die anderen Components weitergegeben werden konnten.

Die logische Strukturierung des Codes wurde dadurch erleichtert, dass React in einer JSX-Datei sowohl JavaScript als auch HTML Code zusammenführt. Gelegentlich war es jedoch schwierig die Syntax von JSX zu verstehen, beispielsweise wenn man eine map-Funktion verwendet hat um HTML Code zu returnen oder bei der Verwendung von React Hooks wie useState und useLocation. Nach einiger Übung wurde die Syntax jedoch intuitiv und der Umgang mit komplizierterer Syntax wie z.B. im Fall des Ternary Operators war auch kein Problem mehr.

Eine weitere Schwierigkeit offenbarte sich anfangs in der Gestaltung von Layouts mit CSS. Zwar war mir die Verwendung von Flexbox und Grid aus den Vorlesungen bekannt, jedoch war die Verschachtelung von Grid in Flexbox und andersherum etwas Unerprobtes. Mit Konzepten wie der Mengenangabe im Warenkorb war es notwendig umzudenken und Grid sowie Flexbox so anzuordnen, dass die Mengenangabe wie ein Element im Product View behandelt wird, aber gleichzeitig einen zentrierten Text und darunter in einer Reihe zwei Buttons und einer Zahl darstellt.

Letzten Endes hat das Projekt Spaß gemacht, war jedoch auch sehr fordernd und Zeit intensiv.