

Prüfung: Projektbericht

User Interface Entwicklung

Benno Steinkamp

855624

Inhalt

Konzeption.....	3
Grundideen und Voraussetzungen.....	3
Darstellung der Übersicht der Aufgaben.....	3
Darstellung der einzelnen Aufgaben	4
Gestaltung der Navigation.....	4
Gestaltung der Contact Page.....	4
Gestaltung der About Page	4
Technische Umsetzung.....	4
Auflistung der Aufgaben.....	4
Detailansicht einer Aufgabe	5
Übergang von Auflistung zu Detailansicht	5
Navigation vertikales scrollen	6
Navigation horizontales Scrollen.....	6
Contact	6
Abbildungen	7

Konzeption

Grundideen und Voraussetzungen

Aus den Vorgaben legte ich mir zunächst folgende Ideen zugrunde. Meine Projektarbeit würde ich als Onepage Webseite erstellen. Außerdem habe ich mich entschieden keine Landingpage zu erstellen, sondern die Darstellung der Ergebnisse als Einstiegspunkt für die Webseite zu nutzen. Die einzelnen Abschnitte der Webseite sollten außerdem nicht wie bei Onepage Seiten üblich untereinander, sondern nebeneinander angeordnet sein, damit die Seite nicht nur aus Listen von Listen besteht.

Die Navigation sollte sich am oberen Rand des Bildschirms befinden und intuitiv verständlich sein sich jedoch nicht mit dem restlichen Inhalt der Webseite bei einem „Seitenwechsel“ verschieben.

Darstellung der Übersicht der Aufgaben

Die zentrale Aufmerksamkeit der Webseite musste, wie durch die Aufgabenstellung bestimmt auf den Ergebnissen der zuvor über das Semester gelösten Aufgaben liegen. Aus diesem Grund war hier angebracht mit der Konzeption dieser Darstellung zu beginnen. Die simpelste Form der Darstellung, in Form einer Liste, wurde bereits durch die Aufgabenstellung ausgeschlossen. Trotz dieser Vorgabe entschied ich mich aus Gründen der Usability meine Darstellung der Projekte auf einem Listenkonzept basieren zu lassen. Diese Darstellung ist für den Nutzer intuitiv am leichtesten Nachzuvollziehen. Die Grundidee bestand darin die Aufgaben als Übersicht mit einem kurzen Auszug aus der Herangehensweise sowie einem Screenshot des Ergebnisses nebeneinander in Kachelform in einer Zeile darzustellen.

Um die Listendarstellung weiter aufzubrechen wurde die Reihenfolge jedes zweiten Eintrages umgekehrt, also stand dort das Bild rechts und der Text links daneben. Da bei Aufgabe 4 drei verschiedene Webseiten erstellt wurden, entstand hier ein Problem, da im jetzigen Layout lediglich Platz für ein Bild pro Zeile, bzw. pro Textabschnitt eingeräumt wurde. Zur Lösung dieses Problems wurden drei Möglichkeiten der Anpassung ausgearbeitet.

Möglichkeit 1: Aufgabe 4 nimmt zwei Zeilen ein, wobei die untere Zeile aus zwei Bild-Kacheln und die obere Zeile, wie bei den anderen Aufgaben aus einer Text-Kachel und einer Bild-Kachel besteht.¹

Möglichkeit 2: Aufgabe 4 nimmt drei Zeilen ein. Auf der einen Seite befindet sich dann eine dreimal so große Textkachel, wie bei den anderen Aufgaben, während sich auf der anderen Seite die drei Screenshots der Webseiten befinden.²

Möglichkeit 3: Aufgabe 4 wird genauso wie die anderen Aufgaben dargestellt jedoch wird das Bild durch eine Slideshow ähnlich der aus Aufgabe 6 ersetzt.³

Eine weitere Anpassung der Listendarstellung wurde zeitgleich vorgenommen:

Anstatt alle Kacheln (Bild und Text) gleich groß zu lassen⁴ wurde die Größe der Text-Kacheln verringert und die der Bild-Kacheln vergrößert. Zunächst auf ein 1:2 Verhältnis, was heißt das die Text-Kacheln 4 von 12 Spalten einnahmen und die Bilder die restlichen 8⁵. Später

¹ Siehe Abb.1

² Siehe Abb.2

³ Siehe Abb.3

⁴ Siehe Abb.4

⁵ Siehe Abb.5

wurde das Verhältnis auf 5 von 12 und die restlichen 7 reduziert⁶. Dies trennte in Kombination mit der Seitenverkehrten Anordnung jeder zweiten Zeile die einzelnen Zeilen voneinander ab. Dies ist der Fall, da nun zwischen den einzelnen Zeilen eine durchgezogene Lücke bleibt, während zwischen Text und Bild nun keine über die Länge der Seite mehr Vorhanden ist.

Diese Entscheidung schloss aber die erste Möglichkeit zum Lösen des Problems von Aufgabe 4 aus, da die Bilder der unteren Zeile dann verschiedene Breiten und damit aufgrund des festgesetzten Seitenverhältnisses verschiedene Höhen. Dies passt nicht in das Layout. Deshalb habe ich mich letztendlich für die zweite Möglichkeit entschieden Aufgabe 4 darzustellen, da es mir wichtig war alle drei Ergebnisse gleichzeitig darzustellen.

Darstellung der einzelnen Aufgaben

Die detaillierte Darstellung der einzelnen Aufgaben, sowie die Verlinkung und Erklärung der Herangehensweise wird mithilfe von Popups umgesetzt. So kann sich die detaillierte Darstellung wie eine eigene Seite verhalten, ohne die Seite zu verlassen. In der detaillierten Darstellung wird oben die volle Herangehensweise dargestellt. Darunter befindet sich ein einfacher Button, welcher die Seite in einem neuen Tab öffnet. Ganz unten befindet sich eine Live-Vorschau der Webseite in einem iframe.

Gestaltung der Navigation

Für die Navigation wurde eine konventionelle Form gewählt, sowie ein Dropdown Menü für die Auswahl bestimmter Aufgaben direkt aus der Navigation. Bei einem Seitenwechsel verschiebt sich die Seite horizontal um so drei verschiedene Unterseiten anzuzeigen. Außerdem sollten alle Übergänge sanft und nicht sprunghaft sein.

Gestaltung der Contact Page

Hier werden lediglich simpel die Kontaktdaten neben einem Foto dargestellt. Für die Daten werden Symbole aus einem Icon Font⁷ verwendet was die Beschriftung universell verständlich, und räumlich kompakter macht und somit nicht von den relevanten Daten ablenkt.

Gestaltung der About Page

Hier wird lediglich ein kleiner Infotext zur Webseite angezeigt, sowie Verlinkungen zu den verwendeten Frameworks, sowie zum Icon Set. Die Links sind übersichtlich in einer Tabelle nebeneinander dargestellt.

Technische Umsetzung

Auflistung der Aufgaben

Da die Nutzung des Bootstrap Frameworks erlaubt war, wurde dieses für die Umsetzung des Konzeptes an allen verwendbaren Stellen genutzt. Die Auflistung der einzelnen Aufgaben wurde mithilfe von „row“ Elementen und dem in Bootstrap eingebautem Raster umgesetzt. So wurde auch die Breite der einzelnen Bilder und Textblöcke am Raster ausgerichtet. Dabei nehmen die Textblöcke jeweils 5 von 12 Spalten ein und die Bilder jeweils 8. Für die tatsächlichen Textkacheln wurde die Bootstrap Klasse „card“ verwendet sowie die dazugehörigen „title“, „body“ und „text“ Klassen. Lediglich das den Aufgabentext enthaltende Element wurde mit einer selbst geschriebenen Klasse gestylt. Diese orientiert sich an einem Tutorial für eine Textbox⁸ welche Text automatisch ab einer bestimmten Zeilenzahl kürzt und dann „...“ dahinter anzeigt. Dies wird erreicht, indem ein das

⁶ Siehe Abb.6

⁷ <https://mfglabs.github.io/mfglabs-iconset/>

⁸ <http://hackingui.com/front-end/a-pure-css-solution-for-multiline-text-truncation/>

Pseudoelement „:before“ mit dem Inhalt „...“ versehen wird, und dann an die linke untere Ecke des Textes gesetzt. Als nächstes wird das Pseudoelement „:after“ dazu genutzt, die Punkte abzudecken, wenn der Text weniger Zeilen lang ist, als durch die maximale Höhe vorgegeben.

Da Bootstrap selbst ohne Anpassung bereits stark responsiv ausgerichtet ist wechseln die nebeneinanderstehenden Bilder und Text automatisch untereinander. Dies geschieht bei dem durch Bootstrap vorgegebenen Breakpoint von 992px Bildschirmbreite und garantiert eine gute Usability auf möglichst vielen Geräten. Lediglich die Zeilenzahl nach denen der Text zu jeder Aufgabe abgeschnitten wird musste mithilfe von „@media“ Blöcken einzeln an die verschiedenen Bootstrap Breakpoints abgestimmt werden. So werden bei einer Bildschirmbreite von mehr als 1200px 11 Zeilen angezeigt. Bei mehr als 992px 9 Zeilen, dann bei mehr als 768px 13 Zeilen, hier wechseln die Bilder und Texte untereinander und werden deshalb im Vergleich zu den Vorherigen Einstellungen breiter. Mein Ziel war hierbei, die Bilder und Textkacheln etwa gleich groß zu halten, weshalb in dieser breite mehr Zeilen dargestellt werden können. Bei mehr als 576px werden wieder nur 9 Zeilen dargestellt. Bei allen kleineren Bildschirmbreiten werden standardmäßig die ersten 10 Zeilen gezeigt.

Für Aufgabe vier wurde eine weitere Hilfsklasse, welche die Zeilenzahl bei den breiteren Bildschirmen um das dreifache erhöht um damit zur höhe der drei Bilder zu passen. Bei den schmalere Bildschirmen wurden die Zeilenzahlen übernommen.

Detailansicht einer Aufgabe

In jeder Zeile befindet sich außer den beiden Elementen für die Text- und Bildkachel auch ein div Element, welches die Detailansicht jeder Aufgabe darstellt. Es belegt die gesamte Fläche des Bildschirms und kann gescrollt werden. Innerhalb dieses Elements befindet sich die Inhalte der Detailansicht. Da das äußere Element jeder Detailansicht aber zunächst mit „display:none;“ ausgestattet ist, werden sie beim Laden der Webseite nicht Angezeigt. Innerhalb der Detailansicht wurden wieder die von Bootstrap bereitgestellten Zeilen- und Spaltenklassen genutzt, um oben eine Textspalte für die Herangehensweise zu präsentieren. Der sich darunter befindende Button wurde auch mit den Bootstrap Standardklassen gestylt. Seine Funktion ist lediglich das nächste aktive iframe-Element zu finden und dann die Adresse in einem neuen Tab zu laden. Darunter befindet sich besagter iframe, welcher eine Live-Vorschau des Aufgabenergebnisses darstellt. Mithilfe von Bootstrap hat der iframe immer ein Seitenverhältnis von 16:9 und sollte deshalb auch auf schmalere Geräten erkennbar dargestellt werden.

Für Aufgabe 4 musste außerdem ein weiterer Button eingefügt werden, welcher die Auswahl zwischen den drei erstellten Webseiten ermöglicht. Hierzu bot sich der Radiobutton an, da er für die Auswahl einer Option aus mehreren gemacht ist. Dieser wurde wieder mit Bootstrap gestylt und mithilfe von jQuery funktionsfähig gemacht. Wird einer der Buttons angeklickt sucht dieser nach den unter ihm platzierten iframes und macht dann nur den iframe sichtbar der seiner id entspricht und macht die beiden anderen unsichtbar.

Übergang von Auflistung zu Detailansicht

Der Übergang zwischen der Auflistung und der Detailansicht einer Aufgabe wurde mithilfe von JavaScript und jQuery gelöst. Klickt man bei einer Aufgabe auf das bestehende Bild oder die Überschrift des bestehenden Textes, so löst das einen Event-Handler aus, welcher zuvor mit jQuery eingerichtet wurde. Da der relative Pfad zwischen einer Überschrift, bzw. einem Bild und dem Element der Detailansicht für jede Aufgabenzeile gleich ist, lässt sich dann der style der jeweiligen Detailansicht auf „display:block“ ändern, wodurch die Detailansicht auf dem ganzen Bildschirm dargestellt wird.

Um wieder auf die Ausgangsseite zurückzukommen muss man nun lediglich außerhalb des inneren Containers klicken. Der äußere Container, welcher sich über den gesamten Bildschirm erstreckt überprüft dann, ob er direkt angeklickt wurde und macht in diesem Fall alle Detailansichten wieder unsichtbar. Ist der äußere Container nicht sichtbar, was der Fall ist, wenn der Bildschirm schmaler als 576px ist oder die Seite auf Smartphone aufgerufen wird, wird unter der Live-Vorschau ein roter Button angezeigt, der auch alle Detailansichten schließt.

Navigation vertikales scrollen

Da ein glattes Scrollen bei der Verwendung der Navigationsleiste deutlich eleganter aussieht als ein Sprung mit einem lokalen href, wurde auch dies mithilfe von jQuery umgesetzt. Dabei sucht der angeklickte Link jeweils das Element mit der in seinem href angegebenen id, findet dann seine x-Position auf der Seite heraus und scrollt dann glatt mithilfe der „window.scrollTo“ Funktion an diese.

Navigation horizontales Scrollen

Die Horizontale Scroll Funktion der Webseite wurde mithilfe von jQuery und mehreren div-Elementen erreicht. Die drei Unterseiten befinden sich nebeneinander in einem inneren div Container. Dieser befindet sich wiederum in einem äußeren div Container, welcher mit der Eigenschaft „overflow:hidden“ ausgestattet ist. Da der äußere div Container dieselbe Größe hat, wie jede der drei enthaltenen Seiten wird jeweils immer nur eine angezeigt. Der innere Container hat jedoch die dreifache breite, um so alle drei Seiten aufzunehmen. Um nun eine der Seiten anzuzeigen, wird der innere Container relativ zum Äußeren verschoben. Damit von jeder Unterseite zu jeder anderen Unterseite navigiert werden kann wurden Hilfsklassen für jeden möglichen Übergang erstellt. Diese wurden dann mit CSS-Animationen versehen, wenn nicht von einer Unterseite zur selben gewechselt wird, wobei dann kein Übergang nötig ist. Die Hilfsklassen werden, mithilfe von jQuery, bei einem Klick auf die in der Navigation befindlichen Links auf den inneren Container angewendet. Hierzu muss sich das Skript Global die jetzt aktive Unterseite merken, um dann bei Auslösung die entsprechende Hilfsklasse für den Übergang zu finden.

Diese Art, die Unterseiten auf einer Webseite gleichzeitig darzustellen, zieht aber den Nachteil mit sich, dass die höhe der Webseite immer die Höhe der höchsten Webseite behält, egal welche Unterseite gerade aktiv ist. Damit sich die Höhe der gesamten Webseite immer an der Höhe der aktiven Unterseite ausrichtet werden die inaktiven Unterseiten auf 1px Maximalhöhe begrenzt.

Contact

Die Contact-Seite wurde wieder mithilfe des Bootstrap Raster Modells an das Konzeptionierte Layout angepasst. Die Kontaktdaten wurden mithilfe einer Tabelle an den Symbolen aus dem Icon Set ausgerichtet. Sie sind, wie die Textkacheln aus der Übersicht, in einem „card“ Element aus Bootstrap enthalten.

Die jeweiligen Daten wurden außerdem mit relevanten Hyperlinks belegt. So führen die Adresszeilen jeweils zu einem Google Maps Standort, die Telefonnummer öffnet, bei Nutzung auf einem Smartphone, sofort die entsprechende App zum Anrufen, bzw. auf einem Computer eine entsprechende App (z.B. Skype). Die Mailadresse öffnet beim Klicken, sofern eingerichtet, ein Mailprogramm. Auch die Twitter Verlinkung führt sofort zum entsprechenden Profil. Da das „target“ Attribut der Weblinks auf „_blank“ gesetzt wurde, öffnen alle Weblinks immer einen neuen Tab anstatt im jetzigen Tab weiterzuleiten.

Abbildungen

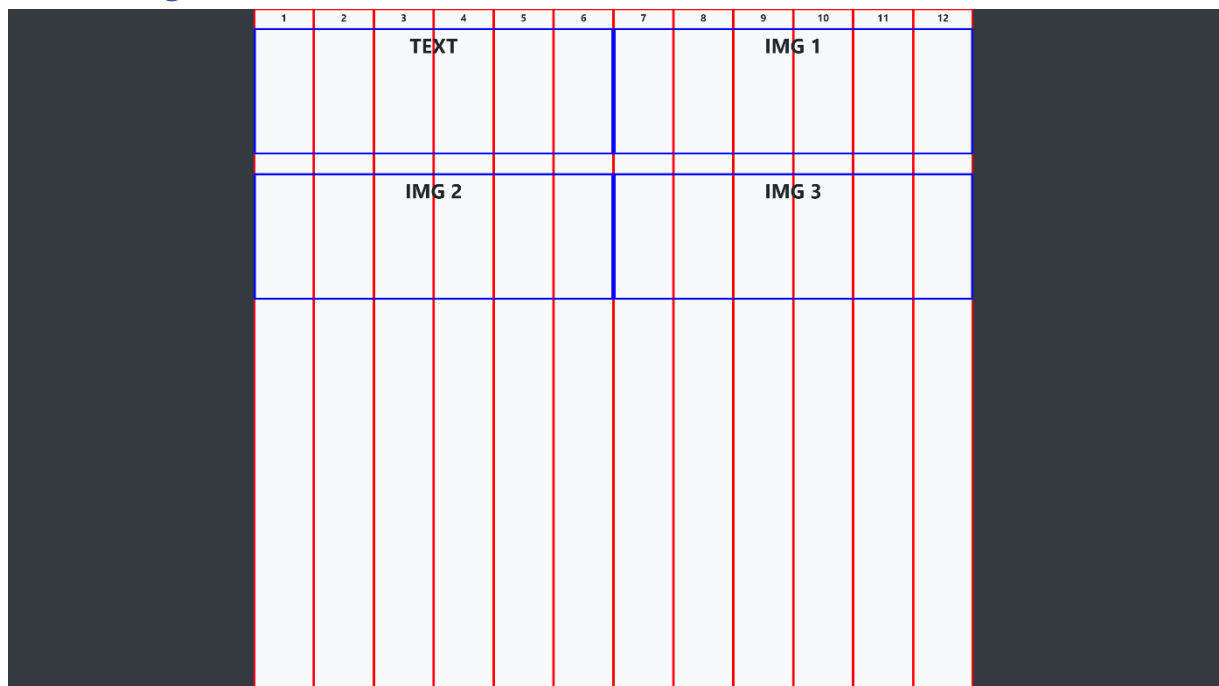


Abb. 1: Wireframe Lösung 1

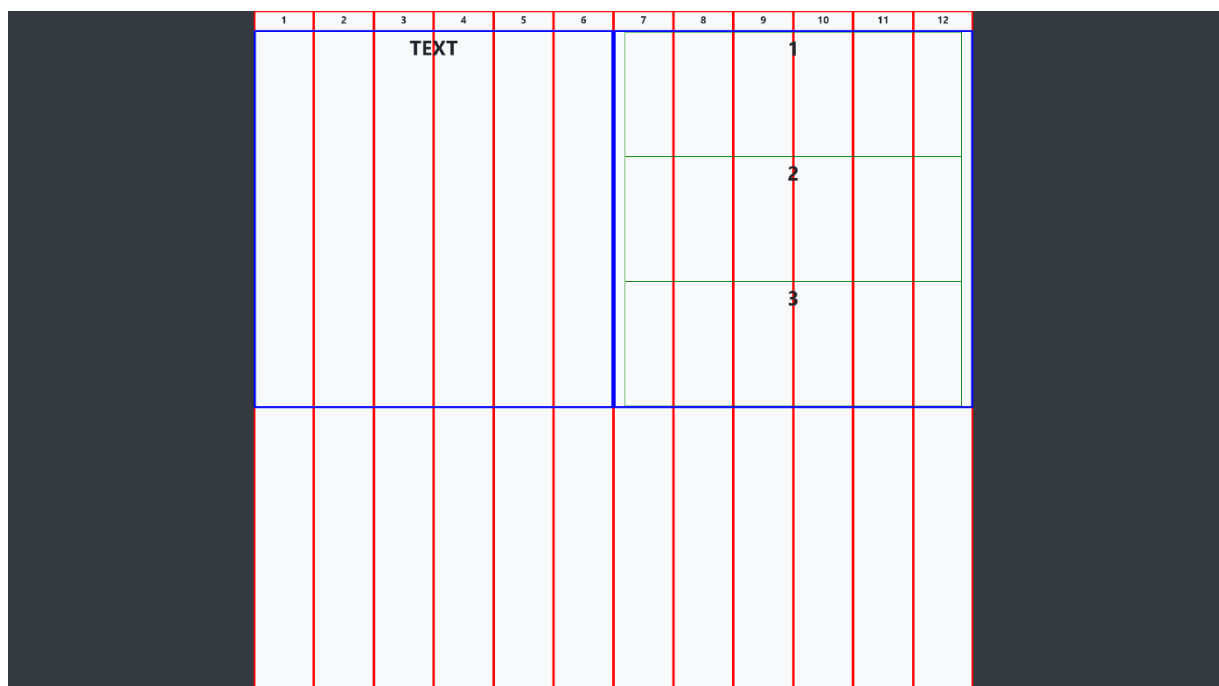


Abb. 2: Wireframe Lösung 2

1	2	3	4	5	6	7	8	9	10	11	12
	TEXT						IMG				
			IMG						TEXT		
	TEXT						IMG				

Abb.5: Wireframe 4-8

1	2	3	4	5	6	7	8	9	10	11	12
		TEXT						IMG			
			IMG						TEXT		
		TEXT						IMG			

Abb. 6: Wireframe 5-7