

BACHELOR-THESIS

ZUR ERLANGUNG DES AKADEMISCHEN GRADES B.SC.

Zukunftsorientierte Entwicklung von Webseiten für Smartwatches nach aktuellen Standards

vorgelegt in Hamburg, November 2019 von

Kevin Brandao da Graca

Matrikel-Nr. 

1. Prüfer: Prof. Dr. Andreas Plaß

2. Prüfer: Prof. Hebecker

HOCHSCHULE FÜR ANGEWANDTE

WISSENSCHAFTEN HAMBURG

Fakultät Design, Medien und Information

Department Medientechnik

Inhaltsverzeichnis

1	Einleitung	5
1.1	Relevanz von Webseiten auf mobilen Endgeräten	5
1.2	Motivation	6
2	Individuelle Technologie der Geräte	8
2.1	Technische Merkmale	8
2.2	Aktueller Stand	9
2.3	Zukünftige Technologie: Smartwatch	10
2.4	Nutzungsverhalten	12
3	Aktuelle Webstandards	14
3.1	HTML und CSS	14
3.2	JavaScript	16
4	Anforderungsanalyse einer Webanwendung	18
4.1	Definition der Nutzer	19
4.2	Beschreibung der funktionalen Anforderungen	20
4.3	Nicht funktionale Anforderungen	26
4.4	Gestaltungsrichtlinien	27
5	Entwicklung einer für Smartwatch optimierten Webanwendung	29
5.1	Konzept und Gestaltungsphase	29
5.1.1	User Stories	30
5.1.2	Desktop	31
5.1.3	Smartwatch	34
5.2	Programmierungsphase	39
5.3	Fehlende Standards	45
6	Zusammenfassung	47
6.1	Auswertung des Prototyps	47
6.2	Ausblick	48
	Glossar	50

Inhaltsverzeichnis

Abkürzungsverzeichnis	54
Abbildungsverzeichnis	55
Code Beispiele	56
Literaturverzeichnis	57

Abstract

The first smartphone with a full-featured browser was released on 09 November 2007 in Europe. For developers, this was the starting point for dealing with the development of mobile websites. Several years have passed since then and mobile device development has become routine. Terms like responsive design or mobile first are no longer unknown for web developers.

This bachelor thesis deals with the future of mobile web development. The focus is mainly on the development of websites for smartwatches. Developing a website involves different phases.

The thesis begins with a market and behavioral analysis to show which devices are currently in focus of developers and how these devices are different from smartwatches. It then examines the current web standards used for mobile web development. In the main part of the work, a web application is developed as a prototype. This prototype should contain different components of a website and be optimized for a smartwatch. The finished prototype will be evaluated and the methods used in the development will be described in the final part of the thesis.

Zusammenfassung

Das erste Smartphone mit einem voll funktionsfähigen Browser kam am 09 November 2007 in Europa auf den Markt. Dies war für Entwickler der Startschuss sich mit der Entwicklung von mobilen Webseiten zu beschäftigen. Seitdem sind einige Jahre vergangen und das Entwickeln für mobile Endgeräte ist zur Routine geworden. Begriffe wie Responsive Design oder auch mobile First sind für Webentwickler keine Fremdwörter mehr.

Diese Bachelorarbeit beschäftigt sich mit der zukünftigen Entwicklung von mobilen Webseiten. Dabei liegt der Fokus vor allem auf der Entwicklung von Webseiten für Smartwatches. Zur Entwicklung einer Webseite gehören unterschiedliche Phasen.

Die Arbeit beginnt mit einer Markt- und Verhaltensanalyse die aufzeigen soll für welche Geräte aktuell entwickelt wird und wie diese Geräte sich von Smartwatches unterscheiden. Anschließend werden die aktuellen Webstandards untersucht, die für die mobile Webentwicklung genutzt werden. Im Hauptteil der Arbeit wird eine Webanwendung als Prototyp entwickelt. Dieser Prototyp soll unterschiedliche Bestandteile einer Webseite beinhalten und für die Nutzung auf einer Smartwatch optimiert sein. Der fertige Prototyp wird ausgewertet und die bei der Entwicklung genutzten Methoden werden im Schlussteil der Arbeit beschrieben.

1 Einleitung

1.1 Relevanz von Webseiten auf mobilen Endgeräten

Die erste Webseite im World Wide Web wurde 1991 von Tim Berners-Lee veröffentlicht (Dino Grandoni 2017). Zu Beginn konzentrierte sich die Webentwicklung ausschließlich auf die Ausrichtung von Webseiten für stationäre Desktops. Parallel zur Entwicklung des Internets veränderten sich auch die benutzten Technologien in der Webentwicklung. Mit dem ersten Mobilfunkstandard Global System for Mobile Communications (kurz: GSM) begann das mobile Internet zu entstehen. Das mobile Internet bestand zu diesem Zeitpunkt aus Textlinks, WAP 2.0/XHTML Browsern und wurde aufgrund der langen Ladezeiten und der fehlenden Rechenleistung der damaligen Geräte kaum genutzt (Ryte 2018).

„A breakthrough internet communications device“ (Ryan Block 2007). - Ein bahnbrechendes Internet-Kommunikationsgerät. Mit diesen Worten präsentierte Steve Jobs am 09.01.2007 eine der drei Haupteigenschaften des ersten iPhones. Das iPhone verfügte über den ersten voll funktionsfähigen Browser und im Vergleich zu anderen Geräten, über eine hohe Rechenleistung (Ryan Block 2007). Damit war das erste iPhone einer der Meilensteine des mobilen Internets und wird gewissermaßen als Geburtsstunde bezeichnet. Dabei waren die ersten mobilen Webseiten separate eigenständige Webseiten, die zum Beispiel mit einem m. vor der Domain gekennzeichnet wurden. Das Problem der separaten mobilen Webseiten war, dass die Anzahl an verschiedenen Displaygrößen nicht abgedeckt werden konnte (Ryte 2018).

Eine einheitliche Darstellung mobiler Webseiten war nicht möglich, weshalb ein dynamisches Design entwickelt wurde. Dies wurde Responsive Design genannt und zeichnet seit etwa 2010 den Standard mobiler Webseiten (Ryte 2018). Beim Responsive Design werden Design-Elemente so gesteuert, dass sich die Darstellung der Webseite an die Browsergröße/Displaygröße anpasst. Seit 2013 ist der Wechsel zu responsivem Design beschleunigt spürbar. Diverse Branchen rüsten ihre Webseiten auf und veröffentlichen sie erneut mit einem responsivem Design (Wikipedia 2019).

Stand 2019 stehen wir vor der fünften Generation des Mobilfunks (Informationszentrum-Mobilfunk 2019) und aktuell basieren nahezu alle veröffentlichten Webseiten auf einem Responsive Design. Ein Grund dafür ist zum Beispiel der Anstieg der Internetnutzung über

1 Einleitung

ein mobiles Gerät. Die Internetnutzung über mobile Geräten überstieg Oktober 2016 zum ersten Mal weltweit die über Desktops (StatCounter 2016). Aktuell ist die Internetnutzung über ein mobiles Gerät gestiegen und liegt ca. bei 60% (Eric Enge 2019). Somit sind mobile Webseiten nicht mehr wegzudenken und genauso relevant wie Webseiten für Desktops. Webseiten die nicht für mobile Geräte optimiert sind werden zum Beispiel von Google bestraft. Sie werden im Google Ranking weiter unten eingestuft oder erst gar nicht angezeigt (Google Developer 2019).

Ein weiterer Aspekt für die Relevanz von mobilen Webseiten ist die Nachfrage der Nutzer. Eine Studie von Google ergab, dass 72% der Teilnehmer eine Webseite mit mobiler Optimierung wieder besuchen würden. 50% der Teilnehmer gaben an, dass Sie eine Webseite ohne mobile Optimierung nicht wieder besuchen (Masha Fisch 2012).

1.2 Motivation

2011 im Rahmen meines Realschulabschlusses mit teil qualifizierenden Medienkenntnissen beschäftigte ich mich zum ersten Mal mit der Entwicklung von Webseiten. Zu diesem Zeitpunkt sammelte ich erste Erfahrungen in diesem Bereich und konnte die Erfahrungen in meiner anschließenden schulischen Ausbildung als Screen Designer erweitern. Während der Entwicklung einer Smartphone Anwendung im Zuge meines Bachelorstudium merkte ich viele parallelen zur mobilen Webentwicklung. Spannend fand ich dabei vor allem Techniken aus der Anwendungsentwicklung für die mobile Webentwicklung zu adaptieren.

Neben meinem Studium an der HAW arbeite ich als Front-/Backend Developer, bei der Firma Homepage Helden. Dort habe ich den Trend zur mobilen Webseite vor allem im Jahr 2017 wahrgenommen. Ursprünglich war die Anpassung für mobile Endgeräte eine zusätzliche Dienstleistung, die nicht im Paketpreis enthalten war. Im September 2017 hat sich die Geschäftsführung dazu entschlossen, die Anpassung für mobile Geräte im Topseller **Business-Paket** zu standardisieren. Ein Grund dafür waren die Erwartungen der Kunden. Viele sind davon ausgegangen, dass eine Webseite standardmäßig auch auf einem mobilen Gerät funktioniert.

Im Jahr 2019 werden bei uns so gut wie alle Webseiten für mobile Geräte optimiert. Die Anpassungen dafür sind zur Routine geworden und laufen trotz individueller Webseiten oft nach dem gleichen Schema ab. Mit immer kleiner werdenden Displays verschwinden die Inhalte der Webseite oder weiten sich nach unten hin aus. Mit dieser Methode können die Anpassungen vor allem schnell und wirtschaftlich vorgenommen werden.

1 Einleitung

Diese Arbeit beschäftigt sich mit der Entwicklung einer Smartwatch optimierten Webseite. Am Ende der Arbeit soll die folgende Forschungsfrage beantwortet werden: Welche Methoden ergeben sich bei der Entwicklung einer Webanwendung, die für Smartwatches optimiert ist?

2 Individuelle Technologien der Geräte

2.1 Technische Merkmale

In diesem Teil der Arbeit möchte die Technischen Merkmale der Geräte hervorheben, die einen bei der Entwicklung von mobilen Webseiten beeinflussen. Die Arbeit bezieht sich dabei bewusst nicht auf technologische Merkmale, die nur in individuellen Situationen gebraucht werden. Die Kamera ist eines dieser technologischen Merkmale. In Kombination mit Augmented Reality Anwendung kann die Kamera in der Webentwicklung verwendet werden, aber bei regulären Webseiten wird diese nicht gebraucht. Die Arbeit geht auf technologische Merkmale ein, die bei der Entwicklung einer Webseite nicht ignoriert werden können.

Interessant für die Entwicklung sind folgende Merkmale Displayeigenschaften, Mobilfunkstandard und Bedienmöglichkeiten. Beim Display ist der Einfluss offensichtlich. Im Vergleich zum Desktop wird dieser bei Mobilien Geräten wesentlich kleiner. Neben der Displaygröße ist auch die Auflösung, Ausrichtung und Form des Displays relevant. Zum Beispiel sollten die genutzten Bilder, der jeweiligen Displayauflösung angepasst sein. Die Ausrichtung ist z.B. für den so genannten **Above the Fold** Bereich wichtig. Der **Above the Fold** Bereich definiert den Teil einer Webseite, der für den Nutzer ohne zu Scrollen sichtbar ist (Vera Scheuering 2014). Dieser Bereich wird oft so gestaltet, dass dort die wichtigsten Elemente platziert werden. Wenn das Format nun vom Hochformat in das Querformat wechselt, müssen die Positionierung der Inhalte überarbeitet werden.

Bei der Form des Display ist es ähnlich. Runde Displays oder auch Displays mit abgerundeten Ecken können Darstellungsprobleme verursachen. Der aktuelle Mobilfunkstandard oder auch der genutzte Mobilfunktarif setzt voraus, wie schnell die Webseite geladen wird. Im Idealfall berücksichtigt den Mobilfunkgeschwindigkeit und optimiert diverse Dateigrößen. Vor allem bei einem geringen Datenvolumen ist es für den Nutzer nicht von Vorteil, wenn die Webseite mehrere Megabyte groß ist. Die Bedienmöglichkeiten eines mobilen Gerätes können ganz unterschiedlich sein und sind im Vergleich zum Desktop komplett anders. Einen einfachen Klick mit der Maus kann durch einen Touch leicht ersetzt werden, aber in einigen Fällen könnten andere Bedienmöglichkeiten Vorteile bieten.

2.2 Aktueller Stand

Um einen möglichst guten Überblick über den aktuellen Stand zu bekommen, untersucht diese Arbeit die technischen Merkmale der am häufigsten verkauften bzw. genutzten Geräte. Dabei bezieht sich die Arbeit bei der Untersuchung nur auf Smartphones und Tablets, weil diese Gerätetypen zu den bewährtesten gehören, um das Internet mobil zu nutzen. Dies ging aus einer Pressemitteilung des Statistischen Bundesamtes hervor. Laut der Studie gehört das Smartphone zum bevorzugten Gerät, um online zu gehen (Statistisches Bundesamt 2018).

Im letzten Quartal 2018 gehörten Samsung, Apple und Huawei zu den Herstellern mit den weltweit meisten Smartphone-Verkäufern. Dies ging aus der Gartner Statistik hervor, die im Februar 2019 veröffentlicht wurde (Gartner 2019). Im Katalog der drei Hersteller findet man eine große Anzahl verschiedener Geräte. Um diese Anzahl einzuschränken, untersucht die Arbeit nur Smartphones, die am häufigsten genutzt werden und die neusten Flaggschiffe der drei Hersteller. Mit dieser Einschränkung können die technischen Merkmale für die aktuellsten und gängigsten Smartphones aufgestellt werden. Diese Geräte haben die folgenden technischen Merkmale:

- Displaygröße: Höhe 1280px bis 3040px, Breite 720px bis 1440px
- Displayauflösung: 2 DPR bis 4 DPR
- Displayform: Notch, abgerundete Displayecken & -kanten, hole punch Display
- Mobilfunkstandard: LTE bis LTE+
- Bedienmöglichkeiten: Multi-Touch, Sprachsteuerung, 3D Touch

(David Smith 2019, Device Atlas 2019, James Kielty 2019, Inside Digital 2019)

Aus der Studie des Statistischen Bundesamtes geht hervor, dass 46% aller Teilnehmer in den letzten drei Monaten für den Internetzugang ein Tablet nutzten. Im Vergleich dazu haben 87% der Teilnehmer ein Smartphone genutzt (Statistisches Bundesamt 2018). Vergleicht man die beiden Märkte miteinander, ist der weltweite Absatz von Tablets deutlich geringer als der des Smartphones. Somit ist der Markt für Tablets deutlich kleiner und die Absatzzahlen sinken seit 2014 kontinuierlich (Statista 2019). Im Bezug auf die Tablets gibt es keine aussagekräftigen Statistiken, über die meistgenutzten Geräte. Um dennoch einen technischen Überblick zu erhalten, werden die aktuellsten Geräte der drei erfolgreichsten Hersteller behandelt. Zu den erfolgreichsten Herstellern gehören Apple, Samsung und amazon.com (IDC 2018).

Die aktuellen Tablets der drei Hersteller haben folgende technische Merkmale:

- Displaygröße: Höhe 1280px bis 2732px, Breite 720px bis 2048px
- Displayauflösung: 2 DPR bis 4 DPR
- Displayform: Notch, abgerundete Displayecken & -kanten, hole punch Display
- Mobilfunkstandard: LTE bis LTE+
- Bedienmöglichkeiten: Multi-Touch, Sprachsteuerung, 3D Touch

(Areamobile 2019)

2.3 Zukünftige Technologie: Smartwatch

Mit dem stetig wachsenden technologischen Fortschritt, stellt sich die Frage was als nächstes kommt. Diese Frage zu beantworten ist jedoch nicht so leicht. Es gibt immer wieder technologische Fortschritte die von unterschiedlichen Unternehmen angekündigt werden. Ebenso bleiben aus Konkurrenzgründen auch technologische Fortschritte hinter verschlossenen Türen, damit der Hersteller als Pionier heraussticht. Was als nächstes kommt kann selten vorhergesehen werden. Aktuell stehen Foldable Screens, Wearables, oder auch das neue 5G-Netz weit oben im Trend. Bei Foldable Screens geht es um ein Display, das sich erweitern lässt, indem es auseinander gefaltet wird (Jerri Ledford 2019). Somit hat man ein Smartphone, das mit einem regulären Smartphone Display genutzt werden kann oder auch mit einem Display in der Größe eines Tablets. Wearables sind Computertechnologien, die am Körper getragen werden (Prof. Dr. Oliver Bendel 2019). Vor allem die Uhren und Brillen waren in der Vergangenheit besonders präsent. Mit dem neuen 5G Netz soll das mobile Internet verbessert werden. Es soll parallel zum 4G-Netz betrieben werden und bietet eine schnellere und konstante Internetverbindung (Informationszentrum-Mobilfunk.de 2019).

All diese unterschiedlichen Technologien in einer Bachelorarbeit zu behandeln würde zu keinem Ziel führen, deshalb beschäftigt sich die Arbeit mit dem Bereich der Wearables. Eine Technologie, die sich in den letzten Jahre etabliert hat. Ein Pionier Produkt war unter anderem die Google Glass. Eine Datenbrille die, die typischen Funktionen eines Smartphones direkt vor das Auge projiziert. Nach der Ankündigung im April 2012 war die Begeisterung groß und sorgte für einen großen Wirbel um die Google Glass. Doch 7 Jahre später zeigte die Google Glass Probleme auf, an denen die neue Technologie gescheitert ist. Zu den Problemen gehörten zu geringe Fortschritte, verbunden mit einer zu langen Wartezeit, Eingriffe in die Privatsphäre von fremden Personen und die Unterschiede zwischen der angekündigten Vision und des umgesetzten Produktes. Vor allem das Problem mit der Privatsphäre hatte keine Lösung und somit wird die Google Glas nur in Unternehmen eingesetzt. Heute wird nur von einem Nachfolger spekuliert, der auch für den Normalverbraucher geeignet ist

2 Individuelle Technologien der Geräte

(Google Watch Blog 2019). Eine andere Technologie die mehr Erfolg erzielen konnte sind die Smartwatches. Sie machen rund 59,1% des gesamten Wearables Marktes aus (IDC 2018). Diverse unterschiedliche Geräte wurden bereits veröffentlicht. Darunter auch viele Nachfolger mit immer optimierterer Technik.

Zu Beginn waren Smartwatches vor allem in Kombination mit einem Smartphone relevant und dienten als Benachrichtigungsgerät, Fitness Tracker oder boten dank unterschiedlicher Applikationen nette Extrafunktionen (Sarah Silbert 2019). Heute ist Apple der erfolgreichste Hersteller der Smartwatch und Ihr aktuellstes Gerät hat deutlich mehr Funktionen (IDC 2018). Die Apple Watch Series 4 funktioniert nun auch als eigenständiges Gerät, welches ohne Smartphone dennoch Internetzugang hat, dank der Cellular Funktion. Cellular bedeutet dabei soviel wie Mobilfunk und bietet die Möglichkeit zu telefonieren, Nachrichten zu senden und das Internet in LTE Geschwindigkeit zu nutzen (Apple 2012). Ein aktueller Nachteil dieser Funktion ist, dass die Auswahl der Mobilfunkanbieter eingeschränkt ist. Die Cellular Funktion wird in Deutschland aktuell nur von O2, Telekom und Vodafone unterstützt (Apple 2019). Dennoch ist es ein Anfang Smartwatches zukünftig als unabhängiges Gerät mit Internetzugang zu nutzen. Das Nutzen von Webseiten auf einer Smartwatch steht dabei in den Startlöchern. Das aktuelle Watch OS Betriebssystem von Apple bietet mit der neusten Version erstmals die Möglichkeit ausgewählte Webseiten anzuzeigen. Dabei sind die Webseiten nicht über einen eigenständigen Browser aufrufbar, sondern können über Verlinkungen in einer Email oder Textnachricht geöffnet werden (Ben Otterstein 2018). Wird eine Webseite nicht unterstützt, erscheint eine Meldung, dass die Webseite nicht auf diesem Gerät anzeigen werden kann.

Zukünftig könnte ein eigenständiger Browser folgen, der für die Smartwatches optimiert ist. Die Konkurrenz ist da Apple ein Stück voraus. Das Android Wear Betriebssystem verfügt bereits über einen Browser. Der Wear Internet Browser kurz WIB vom Hersteller appfour ist bereits für Android Wear erhältlich. Die Akzeptanz des Browser ist aktuell durchwachsen. In Betrachtung der direkten Kundenbewertungen im Google Play Store, erhält der Browser eine durchschnittliche Bewertung. Kritikpunkte sind dabei oft, dass der Browser noch nicht ausreichend optimiert ist. Geht man dennoch von über 500 tausend geschätzten Installationen aus, ist die Nachfrage sehr groß (Google Play 2019). Das gibt Anlass eine Webseite zukünftig für eine Smartwatch zu optimieren.

2.4 Nutzungsverhalten

Im folgenden Abschnitt werden die genaueren Details der Smartwatches untersucht und wie diese sich vom aktuellen Standard unterscheiden. Der erfolgreichste Hersteller Apple hat aktuell 6 verschiedene Modelle auf dem Markt, mit jeweils zwei unterschiedlichen Größen (IDC 2018). Die neueste Apple Watch Series 4 hat jeweils eine Displaygröße von 368 x 448 Pixel und 324 x 394 Pixel. Beide Varianten haben eine DPR von 2 und ein rechteckiges Display mit abgerundeten Ecken (Abb. 2.1) (VIZ Devices 2018). Und die Series 4 erreicht maximale das LTE Mobilfunknetz (Apple 2019).

Die Smartwatch Modelle die mit dem Android Wear Betriebssystem laufen sind wesentlich vielfältiger, weil diese von unterschiedlichen Herstellern produziert werden. Die Displayform ist rechteckig oder komplett rund, wie bei einer traditionellen Uhr. Runde Displays können dabei eine permanent eingblendete **Chin** haben. Die **Chin** ist eine Einsparung im Display und verringert den Viewport einer Anwendung bzw. Webseite (Android Developers 2019). Mit Hilfe einer Liste der neuesten Wear OS Geräte, die von Google erstellt wurde, werden die verschiedenen Displaygrößen untersucht. Die rechteckigen Displays gehen von 160 x 160 bis 186 x 186 Pixel, die runden von 213 x 213 bis 228 x 228 Pixel und runde Display mit einer **Chin** gehen von 205 x 185 Pixel bis 240 x 220 Pixel (Abb. 2.1) (Wear OS by Google 2019).

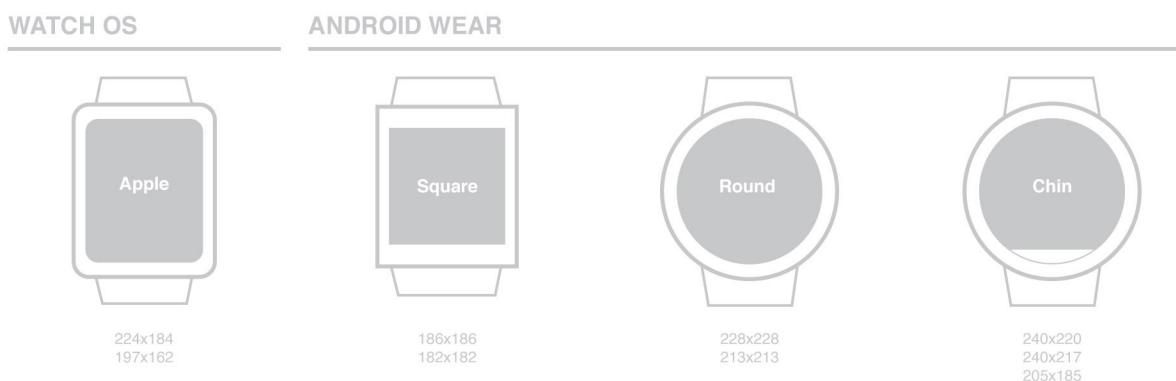


Abbildung 2.1: Technische Zeichnungen von verschiedenen Smartwatch-Varianten, eigene Grafik

Die Displays sind idR. hochauflösend und verfügen über eine DPR von 2 oder höher. Dies ist nur eine grobe Übersicht, weil die Liste nicht alle Geräte auflistet. Dennoch reichen diese Werte, um einen Überblick zu erhalten, in welchen Dimensionen sich die Displaygrößen befinden. Die unterschiedlichen Android Wear Uhren bieten teilweise keine Mobilfunkverbindung an und andere Modelle wie z.B. die Huawei Watch 2 bieten LTE. Die Smartwatches werden mit Multi-Touch bedient und bekannte Funktionen wie Scrollen, durch Ziehen auf dem Display oder zur vorherigen Seite wechseln, durch das Wischen von rechts nach links funktionieren auch.

2 Individuelle Technologien der Geräte

Die Apple Watch Modelle verfügen über eine **Digital Crown**. Eine **Digital Crown** ist ein Drehrad an der Seite des Gerätes, ähnlich wie bei einer analogen Armbanduhr. Die **Digital Crown** bietet unterschiedliche Funktionen, unter anderem kann eine Webseite mit Hilfe der **Digital Crown** gescrollt werden.

Im Vergleich zum aktuellen Standard sind die Displays von Smartwatches wesentlich kleiner, haben unter anderem ein komplett rundes Display und oftmals ein Seitenverhältnis von 1:1. Anders als zum aktuellen Standard gibt es keine große Tastatur, mit der unterschiedliche Daten eingegeben werden. Das Apple **WebKit** für die Apple Watch hat unterschiedliche Eingabemasken. Diese füllen bei der Nutzung das gesamte Display und die Eingabemaske ist für das gewünschte Eingabefeld angepasst (Abb. 2.2) (Apple Developer 2018).

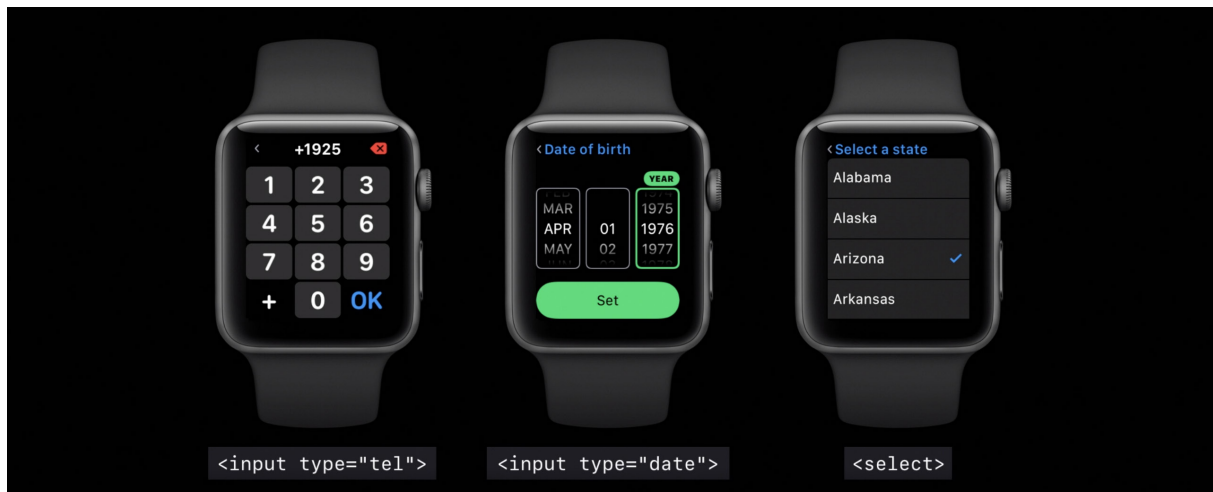


Abbildung 2.2: Unterschiedlichen WebKit Eingabemasken (Apple Developer 2018)

Auch der WIB bietet die Möglichkeit unterschiedliche Eingabemasken zu nutzen. Beide Smartwatch Systeme setzen vor allem auf Spracherkennung, wenn es um Eingabe von Text geht. Displayauflösungen und Mobilfunkstandards der Smartwatches sind im Vergleich zum aktuellen Standard identisch. Smartwatches ohne eigenen Mobilfunk können entweder über die Internetverbindung des verbundenen Smartphones Webseiten aufrufen oder durch eine WLAN Verbindung aufgerufen werden. Das Apple **WebKit** bietet standardmäßig eine eigene Skalierung der Webinhalte. Die Größe der Webseite wird dabei ca. um 50% reduziert. Somit sind auch nicht optimierte Webseiten auf der Apple Watch nutzbar. Bei einer optimierten Webseite, sollte diese Skalierung deaktiviert werden (Apple Developer 2018).

3 Aktuelle Webstandards

Das Internet war zu Beginn nur ein riesiges Netzwerk aus Computern, die miteinander verbunden waren, aber das World Wide Web machte diese Technologie erst für jeden nutzbar. Das World Wide Web ist eine Sammlung von unterschiedlichen Webseiten, die mithilfe eines Browsers dargestellt werden (Google Arts & Culture 2019). Um die Prinzipien und die Langlebigkeit des Webs zu garantieren, wurde das World Wide Web Consortium (kurz W3C) gegründet. W3C ist eine internationale Gemeinschaft, die Webstandards entwickelt. Das Ziel der Standards ist es das Web optimal nutzen zu können (W3C 2019). Die Webstandards bestehen aus HTML und CSS, zwei Kerntechnologien, die im nächsten Abschnitt genauer beschrieben werden (W3C 2016).

3.1 HTML und CSS

Vereinfacht kann man sagen HTML ist für die Struktur einer Webseite verantwortlich und CSS für das Visuelle. Gemeinsam bilden HTML und CSS die Grundlage für die Erstellung von Webseiten und Webanwendungen (W3C 2016).

Genauer ist HTML eine textbasierte Auszeichnungssprache und steht für Hypertext Markup Language und wird weltweit zur Erstellung von Webseiten genutzt. Ein HTML-Dokument besteht aus einer Dokumententypdeklaration (`doctype`), einem Kopfteil (`HEAD`) und einem Körperteil (`BODY`). Die Dokumententypdeklaration steht ganz am Anfang der Datei und beschreibt, welche Dokumenttypdefinition verwendet wird. HTML5 ist der neueste Dokumenttyp und der aktuelle Standard für eine zukunftsreicher Struktur einer Webseite. Im `HEAD` und `BODY` Bereich können verschiedene Tags genutzt werden, um unterschiedliche Elemente zu erzeugen. Der `HEAD` Bereich wird nicht im Anzeigebereich dargestellt. Er enthält hauptsächlich technische und dokumentarische Informationen. Im `HEAD` Bereich wird zum Beispiel der Titel einer Seite definiert oder unterschiedliche Metadaten werden gesetzt und es werden Stylesheets oder Skripte eingebunden. Der `BODY` Bereich enthält den eigentlichen Inhalt einer Webseite. Unter anderem können die verschiedenen Tags genutzt werden, um zum Beispiel Überschriften, Tabellen, Listen, Bilder, Videos usw. darzustellen (Wikipedia 2019).

3 Aktuelle Webstandards

CSS steht für Cascading Style Sheets und ist eine Sprache, die zur Beschreibung der Darstellung von Webseiten genutzt wird. Unter anderem können unterschiedliche Farben, Layouts und Schriftarten von CSS gesteuert werden. Dafür wählt man das gewünschte Element mit Hilfe von unterschiedlichen Selektoren aus und kann anschließend die Werte der Eigenschaften des Elementes bearbeiten. Als Selektoren können z.B. die HTML Tags direkt genutzt werden oder auch Klassen und IDs, die in HTML jedem Element vergeben werden können (W3C 2016). Dabei kann die Klassen beliebig oft wiederholt werden und IDs sollten als einzigartige Identifikation genutzt werden. Die veränderbaren Eigenschaften der Elemente sind zahlreich. In der dritten Generation von CSS (CSS3) gibt es ca. 377 verschiedene Eigenschaften die verändert werden können, um z.B. ein individuelles Layout umzusetzen (Meiert 2019).

Für die Entwicklung von mobilen Webseiten bieten beide Sprachen unterschiedliche Optionen. Mit den folgenden HTML Meta-Element (Code 3.1) kann beeinflusst werden, wie die Webseite auf dem Endgerät geladen wird.

```
1.<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Code 3.1: HTML Viewport Metatag (Jay Kang 2018)

Das erste Attribut `name` beschreibt, um welches Meta-Element es sich handelt. Der Wert `viewport` gibt dem Browser Anweisungen zum Steuern der Abmessungen und Skalierung der Webseite. Das zweite Attribut `content` beinhaltet die genaueren Anweisungen. Mit dem Wert `width = device-width` wird die Seitengröße an die Bildschirmbreite angepasst. Der Wert `initial-scale = 1.0` legt fest welche Zoomstufe zu Beginn eingestellt werden soll. Mit einem weiteren Wert `user-scalable=no` könnte z.B. das Zoomen deaktivieren oder auch aktivieren werden (Jay Kang 2018).

In CSS können verschiedene Media Queries genutzt werden, um eine Webseite für mobile Geräte zu optimieren. Media Queries sind Abfragen aus denen Bedingungen erstellt werden, die erfüllt werden müssen, damit die CSS Einstellungen übernommen werden. Die Bedingungen können sich z.B. auf die Displaybreite, -höhe oder -auflösung beziehen. Dabei können die unterschiedlichen Abfragen miteinander kombiniert werden (SelfHtml 2019). Der folgende Code 3.2 zeigt eine Bedingung, die erfüllt wird für Geräte mit einer maximalen Breite von 420px, einer Höhe von mindestens 500px und einer Displayauflösung von 2DPR.

```
1.@media all
2.  and (max-width: 420px)
3.  and (min-height: 500px)
4.  and (min-device-pixel-ratio: 2)
5.{ /* Hier folgen die CSS Einstellungen */ }
```

Code 3.2: Exemplarisches CSS Media Query

3.2 JavaScript

JavaScript ist eine Skriptsprache, die vor der Ausführung nicht verarbeitet (z.B. kompiliert) werden muss. Der geschriebene JavaScript Code wird beim Aufrufen der Webseite ausgeführt oder auch als Reaktion auf ein vom Benutzer ausgelöstes Ereignis. Diese Ereignisse können z.B. ein Mausklick sein oder auch das Verändern der Browsergröße. Diese Ereignisse können auf verschiedenen Elementen verfolgt werden.

Anders als bei CSS hat JavaScript nicht so eine simple Auswahl an Selektoren, um Elemente auszuwählen. Bei JavaScript kann man für die Auswahl der Elemente unter anderem die `document.getElementById("hier die ID")` Funktion nutzen. Diese Funktion gibt ein Element aus dem HTML Dokument zurück, welches die angegebene ID hat (MDN web docs 2019).

Mit JavaScript können Webseiten dynamischer gestaltet werden. JavaScript bietet die Möglichkeit Inhalte zu verändern, nachzuladen oder zu generieren und somit können die Möglichkeiten von HTML und CSS erweitert werden. Ebenso ermöglicht JavaScript Informationen über die Benutzerumgebung zu erhalten. Unter anderem könnte somit der aktuelle Standort des Nutzers auf der Webseite wiedergegeben werden. Durch solche Funktionen sind Webseiten mittlerweile so interaktiv wie Softwareanwendungen (W3C 2016).

Um die Nutzung von JavaScript zu vereinfachen wird in dieser Arbeit die jQuery-Bibliothek genutzt. JQuery ist eine schnelle, 30KB kleine und funktionsreiche JavaScript-Bibliothek. Unter anderem werden Dinge wie das Durchsuchen und Bearbeiten von HTML-Dokumenten, die Ereignisbehandlung und die Nutzung von Animation vereinfacht. Mit einer benutzerfreundlichen API, die in einer Vielzahl von Browsern funktioniert, können diverse vorgefertigte JavaScript-Funktionen genutzt werden. Zum Beispiel können mit der Hilfe von jQuery, die gleichen Selektoren wie in CSS zur Auswahl von Elementen genutzt werden. Mit der Zeile `$("h1.element")` wird jedes h1 Tag aus dem HTML-Dokument, mit der Klasse `element` ausgewählt. Ein weiterer Vorteil von jQuery ist die vielseitige Erweiterbarkeit (jQuery 2019). Somit können unterschiedliche jQuery Plugin APIs genutzt werden. Diese Plugins bieten komplexe Funktionen, um z.B. einen Slider zu erstellen oder eine interaktive Galerie (Noël Bossart 2010).

Das folgende Code Beispiel zeigt, wie JavaScript in Kombination mit jQuery, für die Entwicklung von mobilen Webseiten genutzt werden kann (Code 3.3 auf S. 17).

3 Aktuelle Webstandards

```
1.$(".open-nav").click(function(){
2.    $(this).find(".hamburger").toggleClass("is-active");
3.    $(".nav").slideToggle( "fast" );
4.});
5.
6.$(".nav ul li").each(function(){
7.    var hasSubnav = $(this).find("ul").length;
8.    if ( hasSubnav >= 1 ){
9.        $(this).prepend("<div class='showSub' />");
10.    }
11.});
12.
13.$(".showSub").click(function){
14.    $(this).toggleClass("open");
15.    $(this).parent("li").toggleClass("open");
16.    $(this).parent("li").children(".sub-menu").slideToggle("fast");
17.});
```

Code 3.3: JavaScript Beispiel für eine mobile Navigation

Die erste Zeile des Codes startet mit einer Ereignisabfrage. Diese Funktion wird durchgeführt sobald das Element mit der Klasse `open-nav` geklickt wird. Die Funktion sorgt dafür, dass dem geklickten Element eine Klasse hinzugefügt oder weggenommen wird und öffnet oder schließt die Navigation mit einer Slide Animation. Mit der extra Klasse für das geklickte Element, kann dafür gesorgt werden, dass sich das Element bei einer geöffneten oder geschlossenen Navigation verändert. So kann sich das Element bei einer geöffneten Navigation in ein Kreuz umwandeln, damit der Nutzer weiß, dass mit dem Element die Navigation wieder geschlossen werden kann. Auf einer mobilen Webseite ist es sinnvoll die Navigation nicht dauerhaft einzublenden, um Platz zu sparen. Diese Funktion bietet die Möglichkeit, die Navigation nur bei Gebrauch zu öffnen.

In der sechsten Zeile beginnt eine `foreach`-Schleife. Die Schleife geht durch jeden Punkt der Navigation und sucht nach Punkten, die eine ein Untermenü haben. Diesen Punkten wird ein Element hinzugefügt, welches genutzt wird, um in der mobilen Version der Navigation, das Untermenü zu öffnen. So hat ergibt sich eine dynamische Art, die extra Elemente nur den Menüpunkten mit einem Untermenü zu geben.

Die 13 Zeile ist der ersten Zeile ähnlich. Eine Ereignisabfrage für Elemente mit der Klasse `showSub`. Sobald das Element geklickt wird, wird diesem Element und das übergeordnete Element eine Klasse hinzugefügt oder weggenommen. Dies dient der selben Funktion, wie in der ersten Zeile beschrieben. Beim Klicken des Elementes wird auch das dazugehörige Untermenü des übergeordneten Elementes mit einer Animation geöffnet oder geschlossen. Somit kann der Nutzer das Menü auch nach belieben ein oder ausblenden. In Kombination mit HTML und CSS hat man so eine voll funktionsfähige Navigation, die auf mobilen Geräten nutzbar ist.

4 Anforderungsanalyse einer Webanwendung

Für diese Bachelorarbeit wird eine Webanwendung entwickelt, die für Smartwatches optimiert ist. Die Webanwendung wird eine exemplarische Webseite, von einem fiktiven Versandunternehmen darstellen. Diese Branche bietet der Anwendung einen ausreichenden Umfang und die Anwendung könnte ein realen Nutzen bieten. In diesem Kapitel geht es um die erste Phase des Entwicklungsprozesses. Der Entwicklungsprozess einer Webanwendung beginnt mit der Ausarbeitung einer Anforderungsbeschreibung. In der ersten Phase wird eine Anforderungsanalyse durchgeführt. In der Anforderungsanalyse werden die wichtigsten Informationen über den Anwendungsbereich und die geforderten Funktionen einer Webanwendung zusammengefasst. Die Zusammenfassung wird anschließend durch entsprechende Beschreibungen formalisiert. Diese Beschreibungen dienen bei der Entwicklung als Grundlage für die zu entwickelnde Webanwendung. Bei der Entwicklung kann man somit verstehen, welche Anforderungen die Webanwendung erfüllen muss. Diese Anforderungen können auch genutzt werden, um die Version der Webanwendung zu bewerten.

Die weiteren Phasen des Entwicklungsprozesses werden unter Berücksichtigung der Anforderungsbeschreibung durchgeführt. Somit ist das Ziel der Anforderungsbeschreibung, dass möglichst ein genaues Gesamtbild der Webanwendung beschrieben wird. Im Bereich der Softwareentwicklung ist es üblich in der Planungsphase ein Lastenheft auszuarbeiten, welches eine detaillierte Beschreibung der Anforderungen an eine neue Anwendung enthält (Dr. Veikko Krypczyk 2016). In Anlehnung daran wird in diesem Kapitel eine Anforderungsanalyse erstellt, die auf den Onlineartikel von Dr. Veikko Krypczyk aufbaut (Dr. Veikko Krypczyk 2016). Für die zu erstellende Webanwendung, soll die Anforderungsanalyse als Endergebnis folgende Aspekte beinhalten:

1. Definition der Nutzer
2. Beschreibung der Funktionalen Anforderungen
3. Nicht funktionale Anforderungen, wie Benutzerfreundlichkeit, Anpassbarkeit, Sicherheit
4. Gestaltungsrichtlinien

4.1 Definition der Nutzer

In diesem Abschnitt werden die möglichen Nutzer einer Lieferservice-Webanwendung definiert. Die Nutzer werden in zwei verschiedene Nutzergruppen unterteilt. Es gibt die Gruppen mit nicht registrierten Nutzern und registrierten Nutzern. Die Gruppe der nicht registrierten Nutzer besteht aus dem Nutzer Empfänger. Zu der anderen Gruppe, mit den registrierten Nutzern gehören Versender, Verkäufer, Lieferanten und Administratoren. Die beiden Gruppen unterscheiden sich in den Zugriffsrechten. Registrierte Nutzer erhalten entsprechend ihrer Zugriffsrechte spezielle Daten oder können Daten der Anwendung verändern. Zur Verdeutlichung der Abgrenzung der unterschiedlichen Nutzergruppen, folgt eine allgemeine Beschreibung der Nutzer mit den entsprechenden Zugriffsmöglichkeiten.

Empfänger

Ein Empfänger ist kein registrierter Nutzer und ist nicht im System aufgeführt. Als nicht registrierter Nutzer hat der Empfänger, einen eingeschränkten Zugriff auf die Inhalte und Funktionen der Anwendung. Der Empfänger hat die Möglichkeiten Sendungsaufträge zu verfolgen und sich über die Angebote des Unternehmens zu informieren. In einer weiteren Ansicht hat der Empfänger die Möglichkeit den Verkäufer zu kontaktieren, um Fragen oder ähnliches zu klären.

Versender

Der Versender ist ein registrierter Nutzer und besitzt somit ein Benutzerprofil im System. In seinem Benutzerprofil sind wichtige Angabe hinterlegt, dazu gehören Kontaktdaten, Zahlungsmittel und Lieferadressen. Diese Angaben kann der Versender in seinem Benutzerprofil verändern. Zusätzlich zu den Möglichkeiten des Empfängers, kann der Versender Versandmarken kaufen und diese mit den gespeicherten Lieferadressen ausfüllen. Die Bezahlung der Versandmarken erfolgt durch die angegebenen Zahlungsmittel. Mit einer extra Ansicht kann man sich als Versender registrieren.

Verkäufer

Die Verkäufer besitzen Benutzerprofile in den nur die Namen hinterlegt sind. Die Verkäufer dienen als Zwischenstück zwischen den Versender und dem Lieferanten. Dafür kann der Verkäufer die verkauften Versandmarken einsehen und diesen einen Status zuweisen. Der Status **fehlende Zahlung** soll die Versandmarke auf eine Warteliste schieben. Mit dem Status **Storniert** kann die verkaufte Versandmarke automatisch rückerstattet werden. Der Status **in Bearbeitung** erzeugt aus der Versandmarke einen Auftrag. Der Verkäufer kann diese

4 Anforderungsanalyse einer Webanwendung

Aufträge den Lieferanten zuweisen. Vor Beginn der Auftragsbearbeitung, kann der Verkäufer dem Lieferanten den Auftrag wieder entziehen. Um die richtigen Lieferanten auszuwählen erhält der Verkäufer einen Vorschlag oder kann sich anhand der Lieferzonen der Lieferanten, selber einen auswählen. Der Verkäufer kann den Versender über auftretende Probleme informieren.

Lieferant

Die Lieferanten haben wie die Verkäufer ein simples Benutzerprofil. In diesem Profil ist der Name und die Versandzone hinterlegt. Ein Lieferant hat Zugang zu den Aufträgen, die ihm vom Verkäufer zugewiesen wurden. Aus den Aufträgen erhält der Lieferant die Adresse des Senders und die Lieferadresse. Sobald der Lieferant mit der Bearbeitung des Auftrages begonnen, kann er den Aufträgen einen Status zuweisen. Anderes als die Status der Versandmarken, dienen diese als Kennzeichnung des Fortschritts der Bearbeitung. Der erste Status heißt **Auftrag erhalten**. Ist dieser Status gesetzt, kann der Verkäufer den Auftrag nicht mehr verändern. Die anschließenden Status beschreiben die einzelnen Etappen, die bei der Sendungsverfolgung angezeigt werden.

Administrator

Der Administrator oder kurz Admin hat kein Benutzerprofil, ist aber ist dennoch ein registrierter Nutzer. Der Admin steht bei der Hierarchie der Nutzer an oberster Stelle. Als Admin kann man die registrierten Nutzer verwalten. Dazu gehört das Anlegen von Verkäufer- oder Lieferantenprofile und das bearbeiten oder löschen von bereits vorhandenen Profilen. Der Admin hat die Möglichkeiten neue Seiten basierend auf vorhandene Seiten zu erstellen, Inhalte von vorhandenen Seiten zu löschen oder bearbeiten und den Aufbau der Navigation zu verändern.

4.2 Beschreibung der funktionalen Anforderungen

In diesem Abschnitt werden die wichtigsten funktionalen Anforderungen beschrieben. Die Beschreibungen der Funktionen werden ausgehen von den unterschiedlichen Nutzern beschrieben. Anhand der identifizierten Nutzer leiten sich die benötigten Seiten für die Webanwendung ab. Somit können die festgelegten Anforderungen direkt den Seiten zugewiesen werden.

Öffentliche Seiten für die Nutzer Empfänger und Versender

Die hier aufgelisteten Seiten sind unabhängig von der registrierten Nutzergruppe erreichbar. Die Seiten richten sich vor allem an die Nutzer Empfänger und Versender. Die beschriebenen Seiten sollen untereinander verlinkt sein, damit man jede Seite von überall aus erreichen kann.

Startseite: Die Startseite dient als Einstiegspunkt der Webseite. Diese sollte besonders leicht und intuitiv erreichbar sein. Von dort aus sollen die Nutzer über Inhalte der Unterseiten informiert werden. Es soll die Möglichkeit bestehen diese Inhalte bei Interesse aufzurufen. Die Kernfunktion der Startseite ist die Sendungsverfolgung. Der Empfänger oder Versender kann einen Sendungsverlauf nachverfolgen. Der Sendungsverlauf basiert dabei auf den Status der Aufträge. Dem Nutzer sollen unterschiedliche Anzeigen geboten werden, um den Sendungsverlauf nachvollziehen zu können.

Inhaltliche Unterseiten: Die Unterseiten mit Inhalten können in der Anzahl variieren. Die Seiten dienen dazu, die Nutzer über Angebote, Neuigkeiten usw. zu informieren.

Kontaktseite: Über die Kontaktseite soll der Empfänger und Versender die Möglichkeit haben, den Verkäufer zu kontaktieren. Die Kontaktaufnahme soll ohne vorherige Registrierung funktionieren. Dem Verkäufer soll dennoch möglich sein, die Kontaktanfrage zu beantworten.

Anmeldung und Registrierung als Versender: Auf dieser Seite kann sich ein Nutzer als Versender anmelden oder registrieren. Für die Anmeldung soll eine E-Mail Adresse und ein Passwort genutzt werden. Nach der Anmeldung wird man auf das Benutzerprofil des Versender weitergeleitet. Für die Registrierung soll der Nutzer eine vorhandene E-Mail Adresse angeben und ein sicheres Passwort. Zur Registrierung gehört auch das Eintragen der Angaben, die im Benutzerprofil gespeichert werden.

Funktionale Anforderungen:

- Inhalte anzeigen
- Inhalte vorstellen
 - Gekürzte Textausschnitte anzeigen
 - Auf die jeweilige Seite verlinken
- Seiten verlinken
- Sendung verfolgen
 - Eingabe für die Sendungsnummer
 - Sendungsablauf ausgeben
 - Etappen des Sendungsauftrages erhalten

4 Anforderungsanalyse einer Webanwendung

- Kontaktformular
- Versender Anmeldeformular
 - Anmelddaten kontrollieren
 - Weiterleitung auf das Benutzerprofil
- Versender Registrierungsformular
 - Anmelddaten erstellen
 - Kontaktdaten, Zahlungsdaten und Lieferadressen speichern

Geschützte Seiten für den Nutzer Versender

Die hier aufgeführten Seiten sind nur für den Nutzer Versender zugänglich. Um diese Seiten aufrufen zu können, muss ein Nutzer sich vorher angemeldet haben. Das Navigieren durch die Seiten, soll von jeder Seite aus möglich sein.

Benutzerprofil: Auf diese Seite wird man nach der erfolgreichen Anmeldung weitergeleitet. Auf der Seite kann der Versender seine Angaben verwalten. Dazu gehört das Anpassen der Anmelddaten, Kontaktdaten und Zahlungsmethoden. Auf dieser Seite können auch neue Lieferadressen erstellt, bearbeitet oder gelöscht werden.

Versandmarken kaufen: Auf dieser Seite kann der Versender unterschiedliche Versandmarken kaufen. Jede Versandmarke soll mit einer Beschreibung und einem Preis beschriftet sein. Um eine Versandmarke zu kaufen, muss der Versender die Marke auswählen und anschließend den Kauf bestätigen. Nachdem der Kauf bestätigt wurde, soll die Zahlung automatisch ablaufen. Dafür werden die im Profil vorliegenden Zahlungsmethoden genutzt.

Versandmarken Übersicht: Diese Seite listet die vom Versender gekauften Versandmarken auf. Der Versender kann die Versandmarken auswählen und diese mit einer Lieferadresse ausfüllen. Die ausgefüllten Versandmarken sollen anschließend druckbar sein. Sobald die Versandmarke gedruckt wurde, gehen die Informationen an den Verkäufer, damit dieser den weiteren Ablauf vorbereiten kann.

Funktionale Anforderungen:

- Seiten verlinken
- Vorhandene Anmeldung kontrollieren
- Profil bearbeiten
 - Anmelddaten, Kontaktdaten, Zahlungsmethoden ändern
 - Lieferadressen erstellen, bearbeiten und löschen

4 Anforderungsanalyse einer Webanwendung

- Versandmarken kaufen
 - Preise und Beschreibung anzeigen
 - Kaufbestätigung aufrufen
 - Zahlung abrufen
- Gekaufte Versandmarken auflisten
- Gekaufte Versandmarken nutzen
 - Lieferadressen eintragen
 - Versandmarke ausdrucken
 - Versandmarke an Verkäufer weiterleiten
- Logout

Geschützte Seiten für den Nutzer Verkäufer

Der Nutzer Verkäufer soll über verschiedene Listen die versandbereiten Versandmarken verwalten können. Die Listen sollen auf eine simple Art miteinander verbunden sein.

Bereitliegende Versandmarken: Sobald der Versender bereit für den Versand ist und die Versandmarke gedruckt hat, soll der Verkäufer mit der weiteren Verarbeitung der Bestellung weiter machen. Auf dieser Seite werden dem Verkäufer alle bereits gedruckten Versandmarken aufgelistet. In dieser Liste kann der Verkäufer die Versandmarken bearbeiten. Zur Bearbeitung gehört das vergeben des Statuts und das zuweisen des Lieferanten. Wie im Abschnitt 4.1 beschrieben, kann der Verkäufer die Versandmarke mit dem Status **Storniert** wieder löschen und mit dem Status **in Bearbeitung** einen Auftrag, aus der Versandmarke erzeugen. Die Aufträge werden in einer anderen Liste aufgeführt. Mit dem Status **fehlende Zahlung** kann die Bearbeitung aufgeschoben werden. Die aufgeschobenen Versandmarken werden in einer anderen Liste gespeichert. Für die Zuweisung der Lieferanten, soll ein automatischer Vorschlag gestellt werden. Dieser Vorschlag soll aus der Versandzone des Lieferanten und der Lieferadresse erzeugt werden. Wird der Status verändert, erhält der Versender eine automatisch versendete E-Mail. Bei unerwarteten Problemen soll der Verkäufer den Versender über die jeweilige Versandmarke erreichen können.

Auftragsliste: In dieser Liste werden die Aufträge angezeigt. Als Verkäufer kann man hier die Aufträge rückgängig machen, wenn die sich noch nicht in Bearbeitung befinden.

Wartende Versandmarken: In dieser Liste befinden sich die Versandmarken mit dem Status **fehlende Zahlung**. Der Verkäufer kann die Versandmarken bei eingehender Zahlung wieder zurück in die vorherige Liste verschieben.

Funktionale Anforderungen:

- Listen verbinden
- Bereitliegende Versandmarken auflisten
- Versandmarken bearbeiten
 - Status vergeben (in Bearbeitung, Storniert, fehlende Zahlung)
 - Lieferant zuweisen
 - Automatischer Vorschlag des Lieferanten
- Versender kontaktieren
 - Automatische E-Mails nach Statusänderung
 - Extra Kontaktmöglichkeit
- Erkennung des Status der Aufträge
- Nicht bearbeitete Aufträge rückgängig machen
- Aufgeschobene Versandmarken zurückschieben
- Logout

Geschützte Seite für den Nutzer Lieferant

Der Lieferant benötigt für seine Tätigkeit nur eine Seite der Webanwendung.

Auftragsliste: In der Auftragsliste der Lieferanten sind die ihm zugewiesenen Aufträge aufgelistet. Dort kann der Lieferant den Status des Auftrages verändern. Mit dem beginnenden Status **Auftrag erhalten**, kann der Verkäufer den Auftrag nicht mehr rückgängig machen. Mit den weiteren Status definieren die Anzeige der Sendungsverfolgung. Jeder Status steht dabei für eine Etappe. Es gibt folgende Etappen Auftrag erhalten, Sendung abgeholt, Im Verteilerzentrum, In Zustellung und Paket zugestellt. Die letzte Etappe kann variieren, wenn der Empfänger nicht anzutreffen war. Wie bei den Versandmarken soll der Versender mit automatischen E-Mail über den Fortschritt informiert werden.

Funktionale Anforderungen:

- Zugehörige Aufträge auflisten
- Status der Aufträge vergeben
- Automatische E-Mails nach Statusänderung
- Logout

Geschützte Seiten für den Nutzer Administrator

Die Seiten des Admins sind wie die vorherigen Seiten auch miteinander verlinkt.

Nutzerverwaltung: Auf dieser Seite sind die verschiedenen Nutzer aufgelistet. Diese sind nach Nutzerart sortiert. Der Admin kann die Passwörter der Nutzer neu vergeben und die Nutzer löschen.

Nutzer anlegen: Der Admin kann auf dieser Seite neue Verkäufer und Lieferanten anlegen. Beim anlegen eines Nutzer muss die Nutzerart, der Nutzernamen und ein Passwort vergeben werden. Um einen Lieferanten zu erstellen, muss die Versandzone angegeben werden.

Seitenübersicht: Diese Seite listet alle öffentlichen Seiten der Nutzer Empfänger und Versender auf. Die Liste der Seite gibt den Aufbau der Navigation wieder. Die Listenpunkte sollen verschiebbar sein, damit man so den Aufbau der Navigation steuern kann. Von dort aus kommt man zur Bearbeitung der aufgelisteten Seiten.

Seite erstellen/ bearbeiten: Diese Seite soll je nach Situation anders aufgebaut sein. Wenn man eine neue Seite erstellt, kann man unterschiedliche Inhaltselemente nutzen, um die Seite mit Inhalten zu befüllen. Der Titel der erstellten Seite dient als Title in der Navigation und in der Url. Wenn man eine Seite bearbeitet, dann werden die bereits genutzten Inhaltselemente angezeigt. Diese können nach belieben verändert, dupliziert oder gelöscht werden. Um eine Seite final zu erstellen oder um Änderungen wirkend zu machen, muss eine Bestätigung erfolgen.

Funktionale Anforderungen:

- Nutzer auflisten
 - Sortierung nach Nutzer Art
- Nutzer bearbeiten
 - Neue Passwörter vergeben
 - Nutzer löschen
- Nutzer anlegen
 - Nutzerart (Verkäufer, Lieferant), Name, Passwort angeben
 - Bei Lieferanten, Versandzone angeben
- Öffentliche Seiten auflisten
 - Sortiert nach dem Aufbau der Navigation
- Aufbau der Navigation über Liste ändern

4 Anforderungsanalyse einer Webanwendung

- Seite erstellen
 - Inhaltselemente auswählen
 - Titel vergeben
 - Erstellung bestätigen
- Seite bearbeiten
 - Vorhandene Inhaltselemente anzeigen
 - Elemente verändern, duplizieren oder löschen
 - Änderungen bestätigen
- Logout

Öffentliche Seite für die Nutzer Verkäufer, Lieferant und Administrator

Die hier erwähnte Seite soll über eine nicht gelistete Url aufrufbar sein. Somit sollen nur Nutzer die im Zusammenhang mit dem Unternehmen stehen, diese Seite aufrufen.

Backend Login: Diese Seite ist für die Anmeldung der Nutzer Verkäufer, Lieferant und Administrator zuständig. Für die Anmeldung soll ein Name und ein Passwort genutzt werden. Es soll erkannt werden um welchen Benutzer es sich handelt, um die richtige Weiterleitung zu ermöglichen. Der Verkäufer wird auch die Seite Bereitliegende Versandmarken weitergeleitet. Ein Lieferant wird auf die Auftragsliste weitergeleitet. Und der Admin wird auf die Seitenübersicht geleitet.

Funktionale Anforderungen:

- Anmeldeformular
 - Anmeldedaten kontrollieren
 - Nutzerart bestimmen
 - Richtige Weiterleitung wählen (Bereitliegende Versandmarken, Auftragsliste, Seitenübersicht)

4.3 Nicht funktionale Anforderungen

In diesem Abschnitt werden die generell geltenden Anforderungen an die Webanwendung beschrieben. Dabei geht es um Benutzerfreundlichkeit, Anpassbarkeit, Verfügbarkeit, Sicherheit usw.. Beginnend mit der Benutzerfreundlichkeit soll die Webanwendung eine einfache Erlernbarkeit bieten. Dafür sollen bekannte Standards wie Menüs, Links und Buttons zur Benutzerführung gewählt werden. Damit der Nutzer die jeweiligen Funktionen richtig deuten kann, sollen Orientierungshilfen genutzt werden. Überschriften können

4 Anforderungsanalyse einer Webanwendung

zum Beispiel als Orientierungspunkt dienen. Es muss auf eine fehlerfreie Umsetzung der Webanwendung geachtet werden, um dem Nutzer eine hohe Korrektheit der Anwendung zu bieten. Internationale Nutzer sollen die Möglichkeit haben ohne die Sprache zu verstehen, die Sprache der Anwendung zu wechseln.

Das Thema der Anpassbarkeit beginnt mit der gewünschten Skalierbarkeit. Die Webanwendung soll eine möglichst effiziente Skalierbarkeit bieten. Das heißt bei einer Verdopplung der Nutzzahlen, soll die Hardwareleistung nicht analog dazu verdoppelt werden. Die Hardwareleistung soll möglichst effizient genutzt werden und nicht linear zu der Nutzeranzahl steigen. Die Anwendung soll Plattform und Browser unabhängig funktionieren. Auf gängigen Betriebssystemen und Browsern soll es keine funktionale Einschränkung geben. Besonderheit für die Arbeit ist, dass die Anwendung nicht nur für aktuelle mobile Geräte optimiert werden soll, sondern in Zukunftsaussicht auf für Smartwatches. Somit soll die Webanwendung bei der Nutzung auf einer Smartwatch, dem Nutzer weiterhin alle Funktionen bieten.

Die Verfügbarkeit der Webanwendung soll es den Nutzern ermöglichen zur gleichen Zeit im System angemeldet zu sein. Die Nutzer Verkäufer und Lieferant sollen parallel zu einander die Aufträge bearbeiten. Der veränderte Status eines Auftrages soll den jeweiligen Nutzern möglichst direkt angezeigt werden. Um eine kurze Ladezeit zu erreichen sollen die genutzten Grafiken optimiert werden und die Antwortzeiten des Server sollen konstant und schnell erfolgen.

Die Sicherheit der Webseite betrifft zu einem großen Punkt die registrierte Nutzergruppe. Die jeweiligen Nutzer haben als ein Passwort im System hinterlegt welches nicht einsehbar sein soll. Durch eine Verschlüsselung bei der Registrierung eines Nutzers, soll es nicht mehr möglichst schwierig sein das Passwort anschließend herauszufinden, ohne das Passwort zu kennen. Dabei sollen die aktuellen Datenschutzrichtlinien eingehalten werden. Um Spam durch die Formulare zu verhindern, soll ein zuverlässige Captcha Prüfung bei jedem Formular vorhanden sein. Die automatisch generierten E-Mails an den Versender, soll klar und deutlich im Zusammenhang mit dem Unternehmen zu betrachten sein. Der Versender soll diese E-Mails von unseriösen E-Mails unterscheiden können.

4.4 Gestaltungsrichtlinien

Die Gestaltungsrichtlinien sollen offen und einfach gehalten werden. Gewünscht ist ein modernes, gut aufgeräumtes Layout. Verschiedene Bereiche sollen visuell voneinander getrennt sein. Das Logo des Unternehmen soll präsent sein und klar als Logo erkennbar. Ebenso ist zu beachten, dass die Farben blau und rot überwiegend vertreten sind. Diese

4 Anforderungsanalyse einer Webanwendung

sollen dem Nutzer, Treue und Energie suggerieren. Die Gestaltung soll sich stark von der Konkurrenz differenzieren. Auf Fotografien soll verzichtet werden. Illustrationen sollen den Aufbau der Inhalte auflockern. Wichtig ist es einen roten Faden bei der Gestaltung zu behalten. Die unterschiedlichen Seiten sollen einheitlich zueinander passen. Für die Webanwendung wird die Open Sand Schriftart gewünscht. Bei der Entwicklung der Gestaltung sollen die angeforderten Funktionen immer im Mittelpunkt stehen. Diese sollen nicht funktional von der Gestaltung beschränkt werden. Neben dem Layout für die Desktop-Version der Webanwendung, soll ein weiterer Entwurf erstellt werden. Der Entwurf soll die Webanwendung auf einer Smartwatch darstellen. Dabei sollen die geforderten Funktionen weiterhin bestehen bleiben.

5 Entwicklung einer für Smartwatch optimierten Webanwendung

In diesem Kapitel der Arbeit wird ein Teil der Webanwendung entwickelt. Die Bachelorarbeit bietet nicht die zeitliche Anforderung, um die gesamte Anwendung zu entwickeln. Für die weiteren Entwicklungsphasen werden die im Kapitel 4 aufgelisteten Anforderungen berücksichtigt. Die entwickelte Anwendung wird als simpler Prototyp dargestellt und ist keine komplett dynamische Webseite, sondern eine statische Seite, die nur als Exemplar dient. Der Prototyp stellt die Startseite der Anwendung dar und ist somit für die Nutzer Empfänger und Versender geeignet. Diese werden im weiteren Verlauf der Arbeit als Besucher gekennzeichnet.

Bevor die nächste Phase mit der Gestaltung beginnt, wird ein Konzept für den Prototypen geschrieben. Dieses Konzept soll die Anforderungen an die Webanwendung zusammenfassen, um die Gestaltungsphase und die Programmierphase zu erleichtern. Ist das Konzept fertig, dann beginnt die Gestaltungsphase mit der Desktop-Version. Anschließend wird die Gestaltung der Smartwatch-Version erstellt. Die Gestaltung wird unter Berücksichtigung der funktionalen Anforderungen und der Gestaltungsrichtlinien erstellt. Sobald die beiden Design-Versionen abgeschlossen sind, beginnt die Programmierphase des Prototyps. Beginnend wird diese Phase mit der Desktop-Version. Schritt für Schritt wird die Desktop-Version immer weiter an kleinere Displaymaße angepasst, bis hin zu den Maßen der kleinsten Smartwatch. In dieser Arbeit werden die verschiedenen Codeabschnitte beschrieben, die für die Smartwatch-Optimierung gebraucht wurden. Für die Simulation der Displaygrößen und der Touch-Bedienung, wird der **Device Mode Emulator** von Google Chrome genutzt.

5.1 Konzept und Gestaltungsphase

Wie in der Kapiteleinleitung erwähnt, geht es in diesem Abschnitt, um die Erstellung eines Konzeptes und um die Gestaltungsphase. Den Aufbau der Inhalte und die unterschiedlichen Funktionen sind von verschiedenen Webseiten übernommen. Unter anderem von DHL, DPD und Hermes (DHL Paket GmbH 2019, DPD Deutschland GmbH 2019, Hermes Germany

GmbH 2019). Im folgenden Abschnitt 5.1.1 wird das Konzept für den Prototypen definiert. Aus dem Konzept leiten sich die funktionalen Anforderungen ab, die bei den Entwicklungsphasen berücksichtigt werden sollen.

5.1.1 User Stories

Das Konzept für den Prototypen wird mit Hilfe von User Stories dargestellt. Die unterschiedlichen Stories werden aus der Besuchersicht geschrieben und beschreiben die Anforderungen an den Prototypen. Jede Story ist dabei ein kleiner, leicht verständlicher Text und enthält die Rolle des Benutzers, die Anforderung an den Prototypen und den Nutzen der Anforderung. Mit Hilfe der Stories wird die Gestaltung und die Programmierung des Prototyps erstellt.

- Als Besucher der Webseite möchte ich, dass die Webseite sich an unterschiedliche Geräte anpasst, um die Webseite auf Desktops und mobilen Geräten zu nutzen.
- Als Besucher möchte ich einfach zur Startseite navigieren können, um zurück zur Startseite zu gelangen.
- Als Besucher möchte ich in der Lage sein, zu einer bestimmten Unterseiten navigieren zu können, damit ich die Unterseiten erreiche, die mich interessieren.
- Als Besucher möchte ich mich auf der Webseite orientieren können, damit ich weiß auf welcher Seite ich mich befinde.
- Als Besucher kann ich mit meiner Sendungsnummer eingeben, um den Versandstatus zu erfahren.
- Als Besucher möchte ich eine simple Ansicht des gesamten Versandstatus haben, um alle Schritte und die noch fehlenden Schritte auf einen Blick zu erfahren.
- Als Besucher möchte ich den aktuellen Versandstatus über eine separate Ansicht erhalten, um den aktuellen Stand direkt zu erkennen.
- Als Besucher möchte ich ein detaillierten Versandstatus mit Datum, Zeit, Ort und Status haben, um den genauen Versandablauf nachvollziehen zu können.
- Als Besucher möchte ich über weitere Angebote des Unternehmens informiert werden, damit ich mich über die Angebote bei Interesse mehr informieren kann.
- Als Besucher möchte ich die offiziellen Sozialen Netzwerke erreichen können, um mich dort über weitere Neuigkeiten zu informieren.
- Als Besucher kann ich die Sprache der Webseite verändern, um die Webseite vollständig verstehen zu können.

5.1.2 Desktop

Die Webseite wird in drei verschiedene Bereiche unterteilt. Es existiert eine Kopfzeile (Header), den Inhalt der Seite (Content) und eine Fußzeile (Footer). Der Header befindet sich im **Above the Fold** Bereich und beginnt auf der linken Seite mit dem Logo, welches auf die Startseite verlinkt. Der Besucher hat somit eine einfache Möglichkeit zurück zur Startseite zu gelangen. Neben dem Logo auf der rechten Seite, ist ein klassisches Dropdown-Menü. Kennzeichen eines Dropdown-Menüs ist, dass das Submenü zu Beginn ausgeblendet ist und erst angezeigt wird, sobald der dazugehörige Hauptpunkt ausgewählt ist. Das Auswählen der einzelnen Menüpunkte und die aktuelle Seite wird durch eine Farbe hervorgehoben. Somit kann der Besucher sich auf der Webseite orientieren und kann alle Unterseiten erreichen (Abb. 5.1).



Abbildung 5.1: Die Kopfzeile der Desktop-Version, eigene Grafik

Nach dem Header kommt ein flächenfüllender Banner. Der Banner gehört zum Content und füllt den restlichen **Above the Fold** Bereich. Auf dem Banner ist ein Werbetext und unterschiedliche Illustrationen, die simpel animiert sind. Der Banner dient nur als Eyecatcher und Gestaltungsmittel, welcher für den Besucher keine weiteren Funktionen bietet (Abb. 5.2).



Abbildung 5.2: Der Banner der Desktop-Version, eigene Grafik

5 Entwicklung einer für Smartwatch optimierten Webanwendung

Um eine Sendung zu verfolgen, folgt nach dem Banner ein Formular. Das Formular hat eine Überschrift, ein Eingabefeld, ein Text und einen Such- bzw. Absende-Button. Die Überschrift dient als Orientierungspunkt und der Text informiert den Besucher über die genaue Nutzung. Das Eingabefeld ist für die Eingabe der Sendungsnummer und akzeptiert bei der Eingabe nur Zahlen. Nach der Eingabe, der Sendungsnummer kann die Suche mit dem Button gestartet werden oder bei fokussiertem Eingabefeld, mit der Eingabetaste. Anschließend soll der Sendungsverlauf angezeigt werden. Der Besucher hat somit die Möglichkeit seine Sendungsnummer einzugeben, um den Versandstatus zu erfahren (Abb. 5.3)

The image shows a dark blue header with the text 'Sendung verfolgen' in white. To the right is a white input field with the placeholder text 'Sendungsnummer eingeben' and a red search icon. Below the input field is a small disclaimer in white text: 'Bitte trage eine gültige Sendungs-/Quittungs-/Auftrags-Nr. ein. Beachte, dass Sendungen frühestens 2 Stunden nach Paketscheiterstellung und nur für einen Zeitraum von 90 Tagen nach Beauftragung angezeigt werden.'

Abbildung 5.3: Das Formular für die Sendungsverfolgung, eigene Grafik

Das nächste Element zeigt schrittweise den Sendungsverlauf. Die verschiedenen Stationen, die eine Sendung durchläuft, werden mit Piktogrammen dargestellt. Die Linien zwischen den Piktogrammen zeigen, ob die Sendung auf den Weg zur nächsten Station ist. Der aktuelle Stand oder bereits abgeschlossene Stationen bzw. Wege werden farbig markiert. Die verbleibenden grauen Elemente zeigen dem Besucher die fehlenden Stationen an. Der Besucher kann somit auf einem Blick nachvollziehen, an welchem Versandpunkt sich die Sendung befindet und wie viele Punkte noch folgen (Abb. 5.4).



Abbildung 5.4: Die erste Ansicht des Sendungsverlaufes, eigene Grafik

Nach dem Sendungsstatus anhand von Piktogrammen, folgt eine genauere Beschreibung über den aktuellen Status. Der Besucher erhält die Information, wann die Abfrage gestartet wurde, um welche Sendungsnummer es sich handelt und den aktuellen Status als Text. Um es optisch ansprechend zu gestalten, wird das Piktogramm der aktuellen Station neben dem Text dargestellt. Der Besucher hat somit eine Ansicht, die ihm direkt über den aktuellen Versandstatus informiert (Abb. 5.5 auf S. 33).

5 Entwicklung einer für Smartwatch optimierten Webanwendung



Abbildung 5.5: Die zweite Ansicht des Sendungsverlaufes mit dem aktuellen Status, eigene Grafik

Als nächstes kommt eine Tabelle, die den detaillierten Sendungsverlauf anzeigt. In der Tabelle erhält der Nutzer Informationen über die Zeit, den Ort und den Status der Sendung. Anderes als im vorherigen Bereich, stehen hier auch Informationen zu den bereits vergangenen Stationen. Der Besucher hat somit eine detaillierte Übersicht über den gesamten Versandablauf und kann diesen genau nachvollziehen (Abb. 5.6).

Schritt	Datum/ Uhrzeit	Ort	Status
1	26.08.2019 16:15	---	Die Sendung wurde vom Versender eingereicht und der Versand wird vorbereitet. Das passende Verteilzentrum wird gesucht
2	26.08.2019 21:30	---	Die Sendung wird in das Verteilzentrum geliefert und anschliessend einsortiert.
3	27.08.2019 08:01	Hamburg	Die Sendung befindet sich im Verteilzentrum und wird in das Zustellfahrzeug geladen.

Abbildung 5.6: Die dritte Ansicht des Sendungsverlaufes mit dem genauen Ablauf, eigene Grafik

Die letzten Elemente im Content sind drei Teaser, die auf drei Spalten verteilt sind. Jeder Teaser hat ein Piktogramm, eine Überschrift, ein Textauszug und ein Button. Das Piktogramm ist passend zur Überschrift und lockert die Gestaltung auf. Die Überschrift und der Textauszug beziehen sich auf die verlinkte Unterseite. Der Button verlinkt auf die Unterseite. Die Teaser informieren den Besucher über weitere Angebote und der Besucher hat die Möglichkeit mehr über die Angebote zu erfahren (Abb. 5.7).



Abbildung 5.7: Ansicht mit drei Teaser auf drei Spalten verteilt, eigene Grafik

5 Entwicklung einer für Smartwatch optimierten Webanwendung

Als letztes folgt der Footer. Der Footer beginnt mit einem fünfspaltigen Element. In den ersten vier Spalten wird das Menü wiederholt. Somit kann der Besucher die Seite wechseln, ohne wieder hochscrollen zu müssen. In der letzten Spalte sind unterschiedliche Netzwerke verlinkt, mit den passenden Icons. Als nächstes kommt der Sprachwechsler, mit dem die Sprache der Webseite umgeschaltet werden kann. Das Logo, die rechtlichen Links und der Copyright vermerk schließen die Webseite ab. Der Besucher hat somit im Footer die Möglichkeit auf die Sozialen Netzwerke zu gelangen und kann die Sprache der Webseite wechseln (Abb. 5.8).



Abbildung 5.8: Die Fußzeile der Desktop-Version, eigene Grafik

5.1.3 Smartwatch

Für die Smartwatch-Version werden die Abschnitte der Webseite reduziert und die Nutzung verändert. Anders als bei der Desktop-Version sind die unterschiedlichen Abschnitte immer genauso groß, wie das Display. So kann der Besucher alle Informationen des jeweiligen Abschnitts sehen, ohne auf der Seite scrollen zu müssen. Um zwischen den Elementen zu wechseln, wischt man von unten nach oben über den Bildschirm. Dabei wird die normale Scrollfunktion verändert, sodass die unterschiedlichen Elemente am Bildschirmrand einrasten. Um möglichst viel Fläche der Bildschirme zu nutzen, werden die Inhalte auf der horizontalen und vertikalen Achse zentriert. Somit werden keine Inhalte aufgrund der runden Bildschirme abgeschnitten.

Der Header hat im Vergleich zur Desktop-Version ein Button statt der Navigation. Die Navigation ist permanent ausgeblendet und kann per Klick auf den Button eingeblendet werden. Sobald die Navigation geöffnet ist, füllt die Navigation das gesamte Display und ein Menüpunkt steht im Fokus. Klickt man nun auf ein Menüpunkt, wird immer auf die Unterseite des fokussierten Menüpunktes verlinkt. Somit hat der Nutzer nicht die Schwierigkeit den richtigen Punkt mit dem Finger zu treffen. Durch auf und ab Wischen können Menüpunkte

5 Entwicklung einer für Smartwatch optimierten Webanwendung

gewechselt werden. Der fokussierte Menüpunkt ist dabei immer vertikal zentriert und nicht transparent, wie die anderen Menüpunkte. Die Navigation kann mit einem extra Button auf der linken Seite geschlossen werden. Zu Beginn zeigt die Navigation nur die erste Ebene an. Das jeweilige Untermenü kann mit dem Pfeil neben dem Hauptpunkt geöffnet werden. Sobald man diesen drückt, öffnet sich das Untermenü und funktioniert nach dem selben Prinzip. Klickt man bei geöffnetem Untermenü den Button auf der linken Seite, dann schließt sich das Untermenü und der Besucher sieht wieder die erste Ebene der Navigation. Diese Variante der Navigation ist platzsparend und bietet auf der Smartwatch die selben Möglichkeiten, wie auf einem Desktop. Aus der Benutzersicht hat mein weiterhin die Möglichkeit sich an der Navigation zu orientieren und auf eine beliebige Unterseite zu navigieren (Abb. 5.9).



Abbildung 5.9: Die Kopfzeile der Smartwatch-Version mit der Navigation, eigene Grafik

Anderes als bei der Desktop-Version startet der Content nicht mit dem Banner, sondern direkt mit dem Formular. Der Banner wird für die Smartwatch-Version ausgeblendet, weil dieser unter anderem nur den Werbenutzen hat und dieser für den Besucher keinen weiteren Nutzen bietet. Dem Besucher fehlen somit keine Anforderungen an die Webseite.

Beim Formular verändert sich hauptsächlich die Größen. Das Eingabefeld und die Schriften werden kleiner. Wird auf das Eingabefeld geklickt, dann öffnet sich die jeweilige Eingabemaske der Smartwatch, mit der die Ziffern eintippen werden können. Der Informationstext ist auf eine Zeile beschränkt, damit die Elemente nicht aus dem Bildschirmrand fallen. Damit der Text dennoch lesbar bleibt, wird der Text von links nach rechts animiert. Ähnlich wie bei einem alten Autoradio bewegt sich der Text und der nicht sichtbare Teil kommt zum Vorschein. Auf diese Weise kann der Platz gespart werden und dennoch den Text lesen. Der Besucher hat somit die selben Möglichkeiten wie auf der Desktop-Version und die Webseite passt sich dem mobilen Gerät an (Abb. 5.10 auf S. 36).

5 Entwicklung einer für Smartwatch optimierten Webanwendung

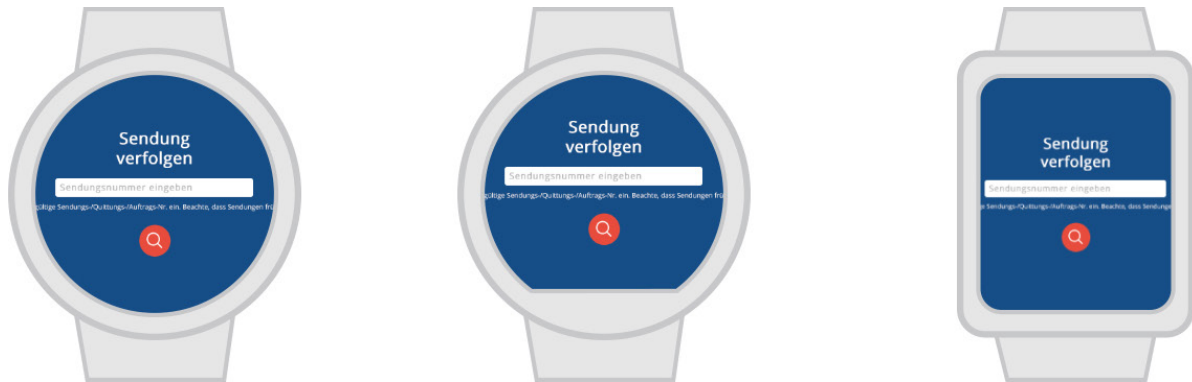


Abbildung 5.10: Das Formular für die Sendungsverfolgung der Smartwatch-Version, eigene Grafik

Im Bereich mit den Piktogrammen wird nur noch das Piktogramm der aktuellen Station angezeigt. Die anderen Stationen können anhand von Zahlen unter dem Piktogramm nachvollzogen werden. Jede Station bekommt eine Zahl. Die aktuelle und die bereits abgeschlossenen Stationen sind fettgedruckt. Die jeweiligen Schritte bzw. Wege sind rund um das Piktogramm ablesbar. Der Rand eines Kreises wurde dafür geviertelt und jedes Stück steht für einen Schritt. Analog zur Desktop-Version ist der aktuelle Schritt und die bereits abgeschlossenen Schritte farblich hervorgehoben. Der Besucher kann somit immer noch auf einem Blick nachvollziehen, wie viele Stationen die Sendung bereits bewältigt hat und wie viele noch folgen werden (Abb. 5.11).



Abbildung 5.11: Die erste Ansicht des Sendungsverlaufes der Smartwatch-Version, eigene Grafik

Im Bereich der genaueren Beschreibung über den aktuellen Versandstatus, sind die Schriftgrößen kleiner geworden und das Wort Sendungsnummer wurde mit Nr. abgekürzt. Somit wird der Platz gespart und der gesamte Text passt auf die Ansicht. Wenn der Text des aktuellen Status zu lang wird oder das Display zu klein, kann die Zeilenanzahl beschränkt werden und mit der selben Animation, wie beim Informationstext des Formulars gearbeitet werden. Dem Besucher fehlt somit keine Information, die in der Desktop-Version vorhanden ist und die Anforderung aus der User Story ist weiterhin erfüllt. (Abb. 5.12 auf S. 37).

5 Entwicklung einer für Smartwatch optimierten Webanwendung



Abbildung 5.12: Die zweite Ansicht des Sendungsverlaufes der Smartwatch-Version, eigene Grafik

Die Tabelle mit allen Versandinformationen wird nicht mehr als Tabelle dargestellt. Die Legende der Tabelle verschwindet und jede Tabellenzeile wird zu einem einzelnen Bereich. Durch Wischbewegungen von links nach rechts oder rechts nach links kann zwischen den Bereichen gewechselt werden. Somit kann sich die Mühe gespart werden, die gesamten Informationen auf einer Ansicht unterzubekommen und der Besucher kann durch eine intuitive Bedienung alle Informationen erhalten. Der Wechsel der Bereiche ist horizontal gelöst und nicht vertikal, damit die Bereiche noch zusammengehörig wirken und nicht als separate einzelne Elemente der Webseite. Ebenso muss der Besucher auch nur ein Abschnitt weiter wischen, wenn er nicht nach diesen Informationen gesucht hat. Im Vergleich zur Desktop-Version fehlen dem Besucher keine Informationen und der gesamte Versandablauf kann weiterhin nachvollziehbar (Abb. 5.13).



Abbildung 5.13: Die dritte Ansicht des Sendungsverlaufes der Smartwatch-Version, eigene Grafik

Bei den Teasern die auf drei Spalten aufgeteilt waren, habe ich die selbe Aufteilung wie bei der Tabelle gewählt. Jede Spalte wurde zu einem eigenen Bereich und die Bereiche können, wie bei der Tabelle durch Wischbewegungen gewechselt werden. Die Piktogramme und die Schriften sind kleiner geworden. Der Text auf dem Button ist verschwunden und wurde mit einem Pfeil ersetzt.

5 Entwicklung einer für Smartwatch optimierten Webanwendung

Somit wird der Platz eingespart und der Besucher weiß dennoch, dass der Button auf eine andere Unterseite verlinkt. Bezüglich der User Stories, hat der Benutzer weiterhin die Möglichkeit sich über weitere Angebote zu informieren (Abb. 5.14).

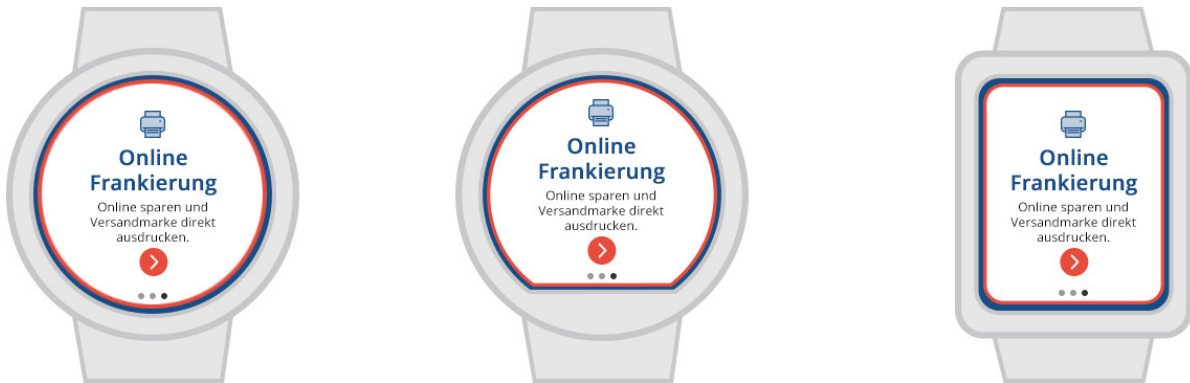


Abbildung 5.14: Die Ansicht der drei Teaser auf der Smartwatch-Version, eigene Grafik

Im Footer ist die wiederholte Navigation verschwunden. Diese hatte nur den Mehrwert, dass der Besucher nicht nach oben scrollen muss, um die Seite zu wechseln. Auf diesen Mehrwert wird verzichtet, um den Platz einzusparen. Die verlinkten Icons sind kleiner geworden. Die Möglichkeit für den Besucher die Sozialen Netzwerke zu erreichen bleibt weiterhin vorhanden. Der Sprachwechsler hat keine Veränderung erhalten, wird der Bildschirm aber zu klein oder es sind mehrere Sprachen vorhanden, dann könnte der Text neben den Flaggen ausgeblendet werden. Der Besucher hätte somit weiterhin die Möglichkeit, die Sprache zu wechseln und muss die passende Sprache nur anhand der Flagge erkennen. Um weiteren Platz einzusparen sind die rechtlichen Links aus dem Footer verschwunden und sind mit in die Navigation gekommen (Abb. 5.15).



Abbildung 5.15: Die Fußzeile der Smartwatch-Version mit der Navigation, eigene Grafik

5.2 Programmierungsphase

Dieser Abschnitt beschäftigt sich mit der genaueren Umsetzung der Smartwatch-Version. Die verschiedenen Code-Abschnitte werden erklärt und die genutzten Plugins werden beschrieben. Die Gestaltung der programmierten Version kann an einigen Stellen vom Layout abweichen. Die Gründe dafür werden im Abschnitt 5.3 erklärt. Die Webseite basiert auf den Webstandards, die im Kapitel 3 erklärt wurden. Grundlegend wurde für die Erkennung der Smartwatches in CSS und JavaScript die selbe Art von Abfrage genutzt. Die Abfrage in CSS sieht wie folgt aus (Code 5.1).

```
1. @media all
2.   and (max-width: 480px)
3.   and (max-height: 480px)
4. { /* Hier folgen die CSS Einstellungen */ }
```

Code 5.1: CSS Media Query welches für Smartwatches wirkt

Sobald die Displaymaße kleiner oder gleich 480x480px sind, dann wird der folgende Code ausgeführt. Somit werden alle Geräte, die in Abschnitt 2.4 vorgestellt wurden mit enthalten und die Geräte aus Abschnitt 2.2 sind davon nicht betroffen. Die weiteren Codebeispiele, die in diesem Abschnitt erklärt werden, durchlaufen diese Abfrage und erfolgen nur bei positiver Rückmeldung der Abfrage.

Jetzt geht es um die Aufteilung der Abschnitte und die veränderte Scrollfunktion. Für die Aufteilung der Abschnitte, erhält jeder Abschnitt eine gleiche Klasse gegeben. Über diese Klassen wird die Höhe und die Breite der Abschnitte definiert. Für die Breite und Höhe werden die CSS Eigenschaften `width` und `height` verändert. Diese wurden auf `100vw` und `100vh` gesetzt. Diese Werte stehen für `Viewport Width` und `Viewport Height`. Die Breite und Höhe der Abschnitte passen sich somit immer der Displaygröße an.

Die veränderte Scrollfunktion wurde mit dem jQuery Plugin **Scrollify** umgesetzt. Das Plugin kommt von dem Entwickler Luke Haas und sorgt dafür, dass die Abschnitte an den richtigen Positionen einrasten. Das Plugin ist für die Bedienung mit Touchscreens optimiert und ist vollständig konfigurierbar. Für den Prototypen reicht die Standardkonfigurierung mit der extra `section` Option. Mit dieser Option werden die Abschnitte ausgewählt, die am Bildschirmrand einrasten sollen. Zur Auswahl der Abschnitte werden die bereits hinzugefügten Klassen genutzt (Luke Haas 2019). Im späterem Verlauf der Arbeit werden zwei Methoden von diesem Plugin genutzt.

Spezifisch für die Apple Wasche Modelle, muss die automatische Skalierung der Webinhalte deaktiviert werden. Die Webseite ist für die Größe der Smartwatches optimiert und benötigt deshalb keine automatische Skalierung der Webinhalte. Um die automatische Skalierung zu

deaktivieren, kann man den Code 3.1 nutzen. Ausschlaggebend ist dabei der Wert `initial-scale = 1.0`. Dieser Wert liegt standardmäßig beim **WebKit** der Apple Watches auf 0.49. Um anschließend die richtigen Werte für die Abfrage im Code 5.1 zu erhalten, wird ein neuer Meta-Tag benötigt. Dieser Meta-Tag lautet `disabled-adaptations` und benötigt das `content` Attribut mit dem Wert `watch` (Apple Developer 2018). Mit diesen Meta-Tags ist die Webseite so konfiguriert, dass der Browser eine optimierte Webseite anzeigt.

Als nächstes geht es um den Aufbau der Navigation. Der Code dafür basiert auf den Beispiel Code 3.3. Dieser Code wird für die Nutzung auf der Smartwatch erweitert. Der Code für den Aufbau der Navigation wird in mehrere Codeabschnitte ausgeteilt, um diese einfacher zu erklären. Wichtig dabei ist, dass die verschiedenen Abschnitte nur im Zusammenhang funktionieren.

```
1.$('.open-nav').click(function(){
2.  if (docWidth <= 480 && docHeight <= 480) {
3.    $('.nav').css("left", "0px");
4.    $("body").css("height", "100vh");
5.    $("body").css("overflow", "hidden");
6.    $("body").css("touch-action", "none");
7.    $("html").css("overflow", "scroll");
8.    $("html").css("-webkit-overflow-scrolling", "touch");
9.
10.   $.scrollify.disable();
11.
12.   $('.open-nav').toggleClass('is-active');
13.   $('.open-nav').find('.hamburger').toggleClass('is-active');
14. }
15.});
```

Code 5.2: Code Abschnitt für das Öffnen des Menüs der Smartwatch-Version

Der Code 5.2 wird benötigt, um das Menü der Smartwatch-Version zu öffnen. Der Code beginnt mit einer Ereignisabfrage. Sobald das Element `open-nav` geklickt wurde, folgt eine `if`-Bedingung. Die `if`-Bedingung kontrolliert, ob die Maße des Displays, der einer Smartwatch entsprechen. Wird die Bedingung erfüllt, öffnet sich die erste Ebene des Menüs. Anders als im Code 3.3, wird das Menü nicht mit einer Slide Animation geöffnet. Initial ist das Menü nicht ausgeblendet, sondern befindet sich außerhalb des Viewports. Statt der Slide Animation, wird das Menü von der rechten Seite aus, in den Viewport geschoben. Diese Variante wurde gewählt, weil das Menü für eine spätere Funktion nicht ausgeblendet sein darf. Die Zeilen 4 bis 8 passen den `body` und den `html` Tag an. Die Zeilen bezwecken, dass das Scrollen auf der Webseite nicht mehr möglich ist. Dadurch wird die Benutzerfreundlichkeit optimiert, weil sich die Abschnitte hinter dem Menü, nicht beim wechseln der Menüpunkte verändern. Die 10 Zeile dient dem selben Effekt. In dieser Zeile wird eine Methode des **Scrollify** Plugins genutzt. Mit der Methode `disable()` wird das Plugin deaktiviert und somit auch das

Einrasten der Abschnitte (Luke Haas 2019). Die Zeilen 12 und 13 verändern das `open-nav` Element, wie im Code 3.3 in ein Kreuz. Das Kreuz symbolisiert den Nutzer, dass das Menü mit dem Element geschlossen werden kann (Code 5.2 auf S. 40).

```
1.$('.nav ul li').each(function(){
2.  var hasSubnav = $(this).find('ul').length;
3.  if ( hasSubnav >= 1 ){
4.    $(this).find("> a").after('<div class="showSub" />');
5.  }
6.});
7.
8.$('ul.sub-menu').each(function(){
9.  $(this).appendTo('nav');
10.});
11.
12.$('.nav ul li .showSub').click(function(){
13.  var liClass = $(this).parent().attr('class').split(' ')[0];
14.  $('ul.sub-menu.'+liClass).css('left', '0px').addClass('openSubMenu');
15.});
```

Code 5.3: Code Abschnitt für das Öffnen des Untermenüs der Smartwatch-Version

Der Code 5.3 wird für das Untermenü der Smartwatch-Version genutzt. Die `foreach`-Schleife aus der ersten Zeile ist vergleichbar, mit der `foreach`-Schleife aus dem Code 3.3. Weil die Untermenüs für die Smartwatch-Version getrennt von ihren Hauptmenüpunkten angezeigt werden, müssen die Positionen der Untermenüs verändert werden. Dies geschieht in der 8. Zeile. Dort beginnt eine `foreach`-Schleife, die jedes Untermenü aus dem jeweiligen Hauptmenüpunkt entfernt und in die übergeordnete Navigation einfügt. Um ein Untermenü zu öffnen, muss das `showSub` Element geklickt werden. Die dazugehörige Funktion beginnt ab Zeile 12. Sobald das Element geklickt wurde, wird in der 13. Zeile, die erste Klasse des Hauptmenüpunktes gespeichert. Die Klasse wird genutzt, um das richtige Untermenü anzuzeigen. In der 14. Zeile wird das passende Untermenü in den Viewport geschoben (Code 5.3).

```
1.$('.open-nav').clone().prependTo('.nav');
2.$('.nav .open-nav').attr('onclick', 'closeSmartMenu()');
3.
4.function closeSmartMenu() {
5.  if($('.ul.sub-menu.openSubMenu').length) {
6.    $('.ul.sub-menu.openSubMenu').css('left', '100%');
7.    $('.ul.sub-menu.openSubMenu').removeClass('openSubMenu');
8.  }
9.  else {
10.    $('.nav').css('left', '100%');
11.    $('body').css('height', 'unset');
12.    $('html').css("overflow", "auto");
13.    $('html').css('-webkit-overflow-scrolling', 'auto');
14.
15.    $.scrollify.enable();
16.
17.    $('.open-nav').toggleClass('is-active');
18.    $('.open-nav').find('.hamburger').toggleClass('is-active');
19.  }
20.}
```

Code 5.4: Code Abschnitt für das Schließen der Menüs

Der Code 5.4 wird benötigt, um das Menü wieder zu schließen. Das erste Problem welches sich dabei ergibt ist, dass das ursprüngliche Element, welches zum schließen des Menüs genutzt wurde, vom Menü überdeckt wird. Das entsprechende Element `open-nav` wird in der ersten Zeile dupliziert und in die Navigation eingefügt. Somit ist das Element auch bei einem geöffneten Menü noch sichtbar. Das `open-nav` Element wird in der Smartwatch-Version nicht nur für das Schließen des Hauptmenüs genutzt, sondern auch, um das Untermenü zu schließen. Somit benötigt das duplizierte `open-nav` Element eine eigene Ereignisabfrage. In der zweiten Zeile wird diese Abfrage, in Form einer Funktion, dem Element hinzugefügt. Wird das duplizierte `open-nav` Element geklickt, dann wird die `closeSmartMenu()` Funktion ausgeführt.

Die Funktion beginnt in der 5 Zeile mit einer `if`-Bedingung. Die Bedingung kontrolliert, ob ein Untermenü geöffnet ist. Dazu wird geschaut, ob ein Element mit der Klasse `openSubMenu` existiert, die beim öffnen des Untermenüs vergeben wird. Ist dies der Fall, dann wird das offene Untermenü geschlossen und die Klasse `openSubMenu` wird entfernt. Trifft die Bedingung nicht zu, dann wird das Hauptmenü geschlossen. Dafür wird das Hauptmenü wieder zurück aus den Viewport geschoben. Die Zeilen 11 bis 13 ermöglichen wieder das Scrollen auf der Webseite. In der 15 Zeile wird die `enable()` Methode des **Scrollify** Plugins genutzt. Die `enable()` Methode aktiviert das **Scrollify** Plugin, welches beim öffnen des Menüs deaktiviert wurde (Luke Haas 2019). Mit der Aktivierung des Plugins, können die Abschnitte wieder am Displayrand einrasten. In den Zeilen 17 und 18 wird das `open-nav` Element, welches die Form eines Kreuzes hat, wieder in die Ursprungsform geändert (Code 5.4 auf S. 41).

```
1.$('.nav > ul').slick({
2.  vertical: true,
3.  infinite: false,
4.  autoplay: false,
5.  speed: 500,
6.  arrows: false,
7.  dots: false,
8.  draggable: true,
9.  verticalSwiping: true,
10. centerMode: true
11.});
```

Code 5.5: Code Abschnitt für die Slick Slider Konfiguration

Für das Wechseln der Menüpunkte wird das jQuery Plugin **Slick** genutzt. Das Plugin wurde vom Entwickler Ken Wheeler entwickelt. Mit dem Plugin können voll konfigurierbare Slider erstellt werden. Das Plugin bietet dafür eine umfangreiche Auswahl an Einstellungen und funktioniert mit jedem HTML Element (Ken Wheeler 2019). Der Code 5.5 zeigt die Einstellungen, die für das Menü gewählt wurden. In der ersten Zeile wird definiert, welche Elemente vom Plugin verändert werden. Mit dem Selector `.nav > ul` werden alle

5 Entwicklung einer für Smartwatch optimierten Webanwendung

Menüs, die sich in der Navigation befinden, zu einem Slider. Die Menüpunkte innerhalb der Menüs sind anschließend Teil eines Sliders. Standardmäßig sind die **Slick** Slider horizontal angeordnet. Für das Menü wird eine vertikale Anordnung der Menüpunkte benötigt. Dies wird in der zweiten Zeile eingestellt. Für die weitere Konfiguration des Menüs, sind die Zeilen 9 und 10 relevant. In der 9 Zeile wird die `verticalSwiping` Option aktiviert. Die Option bewirkt, dass das fokussierte Menüpunkt, durch vertikale Wischbewegungen wechselbar wird. Durch die `centerMode` Option in der 10 Zeile, wird das fokussierte Menüpunkt, in der Mitte des Viewports positioniert (Code 5.5 auf S. 42).

```
1.$('.nav ul li a').click(function(e){
2.  e.preventDefault();
3.  var clickedNav = $(this).parent().parent();
4.  var location = clickedNav.find('.slick-current a').text();
5.  console.log(location);
6.});
```

Code 5.6: Exemplarischer Code für das Verlinken der Menüpunkte

Für das Wechseln der Unterseiten, benötigt die Smartwatch-Version eine veränderte Ereignisabfrage. Der Code 5.6 zeigt wie diese Ereignisabfrage aussehen könnte. Weil es sich beim Prototypen um keine vollständige Webseite handelt, muss der Code 5.6 für eine vollständige Webseite noch überarbeitet werden.

In der ersten Zeile beginnt die Ereignisabfrage. Sobald ein Menüpunkt geklickt wurde, wird die Funktion ausgeführt. Zu Beginn der Funktion wird in der zweiten Zeile das Standardereignis des Menüpunktes deaktiviert. Beim Klicken eines Menüpunktes wird der Nutzer auf den im Menüpunkt hinterlegten Link weitergeleitet. Für den Prototypen wird dieses Ereignis nicht benötigt. Wie im Abschnitt 5.1.3 beschrieben, soll auf die Unterseite des fokussierten Menüpunktes weitergeleitet werden. In der 3 und 4 Zeile wird der fokussierte Menüpunkt gesucht. Dafür wird in der 4 Zeile das Menü, des geklickten Menüpunktes gespeichert. Dies könnte das Hauptmenü sein oder auch eines der Untermenüs. Anschließend wird nach dem Menüpunkt gesucht, welches die Klasse `slick-current` hat. Diese Klasse wird von dem **Slick** Plugin vergeben und wird dem aktiven Slide zugeordnet. Für den Prototypen ist dieser Slide, der fokussierte Menüpunkt. In der 4 und 5 Zeile wird für den Prototypen, der Text des Menüpunktes in der Web-Konsole ausgegeben. Für eine vollständige Webseite würde die Funktion `text()` mit der Funktion `attr()` gewechselt werden. Mit der `attr()` Funktion, kann das `href` Attribut des Menüpunktes gespeichert werden. Das `href` Attribut beinhaltet die verlinkte Seite. In der 5 Zeile würde die Konsolenausgabe, mit einer Weiterleitung gewechselt werden. Eine Weiterleitung ist zum Beispiel mit dem Code `window.location = url;` möglich. In der Variable `url` ist der gespeicherte Link hinterlegt (Code 5.6).

5 Entwicklung einer für Smartwatch optimierten Webanwendung

```
1.$('.footer .outro ul li').each(function(){
2.  $(this).appendTo('.nav ul.main-menu li.fourth .sub-menu');
3.});
```

Code 5.7: Verschiebung der rechtlichen Links aus dem Footer

Mit dem Code 5.7 werden die rechtlichen Links aus dem Footer, in das Menü verschoben. Dafür wird eine foreach-Schleife genutzt. Die Schleife geht durch jeden Link und fügt diesen Link dem Menü hinzu (Code 5.7).

```
1.<input class="input" type="number" name="sendungsnummer"
  placeholder="Sendungsnummer eingeben" aria-label="Sendungsnummer">
```

Code 5.8: Eingabefeld für die Sendungsnummer

Das Eingabefeld für die Sendungsnummer wird mit dem Code 5.8 generiert. Mit dem `input` Tag wird das Eingabefeld erzeugt. Das Attribut `type` mit dem Wert `number` wird genutzt, um das Eingabefeld auf Zahlenwerte einzuschränken. Das Attribut wird auch benötigt, damit die für Zahlenwerte optimierte Eingabemaske angezeigt wird. Das `aria-label` Attribut wird spezifisch für das **WebKit** genutzt. Mit diesem Attribut wird dem Eingabefeld ein Titel übergeben. Die Eingabemasken des **WebKits** überdecken den gesamten Viewport. Das `aria-label` wird dem Benutzer bei einer geöffneten Eingabemaske angezeigt, damit dieser nachvollziehen kann, um welches Eingabefeld es sich handelt (Code 5.8)(Apple Developer 2018).

```
1.var animateTextWidth = $('.animateText').width();
2.function loop() {
3.  $('.animateText').css('left', '0px');
4.  $('.animateText').delay(4000).animate ({
5.    left: '-'+animateTextWidth,
6.  }, 25000, 'linear', function() {
7.    loop();
8.  });
9.}
```

Code 5.9: Code Abschnitt für die Animation des Textes

Damit der Text nicht den gesamten Viewport von der Formularansicht (Abb. 5.10 auf S. 36) füllt, muss der Text angepasst werden. Dafür wird der Text mit Hilfe von CSS auf eine Zeile limitiert. Dafür wird die Eigenschaft `white-space` mit den Wert `nowrap` gefüllt. Damit der Nutzer den Text weiterhin lesen kann, wird der Text animiert, wie es im Abschnitt 5.1.3 beschrieben wurde. Der Code 5.9 ist für diese Animation zuständig. In der ersten Zeile wird die gesamte Breite des Textes in einer Variable gespeichert. Die Variable wird für die Animation in invertierter Form benötigt. Eine sich wiederholende Funktion beginnt in der zweiten Zeile. In dieser Funktion wird die Animation ausgeführt. Die Funktion wiederholt sich, damit die Animation nicht aufhört. In der 4. Zeile wird die `delay()` Funktion genutzt,

um vor Beginn der Animation eine Pause zu erzeugen. Die Pause dauert 4 Sekunden und wird benötigt, damit der Nutzer nachvollziehen kann, wo der Textanfang ist. Wäre die Pause nicht vorhanden, dann würde sich der Text permanent von rechts nach links bewegen. Für die Animation wird die CSS Eigenschaft `left` verändert. Diese Eigenschaft verändert sich von 0 bis hin zum Wert der invertierten Variable. Dies geschieht über einen Zeitraum von 25 Sekunden. Der Zeitraum wird in der 6 Zeile in Millisekunden angegeben. In der dritten Zeile wird die Eigenschaft zurück gesetzt, damit die Animation beim nächsten Durchlauf wieder von 0 aus an beginnt. Sobald die Animation abgeschlossen ist, wird in der 7 Zeile, die Funktion `loop()` wiederholt (Code 5.9 auf S 44).

```
1.$('.box33').slick({
2.  infinite: false,
3.  autoplay: false,
4.  speed: 500,
5.  arrows: false,
6.  dots: true,
7.  draggable: true,
8.  slidesToShow: 1,
9.  slidesToScroll: 1
10.});
```

Code 5.10: Konfiguration des Slick Slider für die Teaser

Die Aufteilung der Tabellenspalten oder der Teaser wird mit dem **Slick** Plugin umgesetzt. Für die Teaser wird das Plugin wie im Code 5.10 konfiguriert. Im Vergleich zum Menü, werden die Elemente horizontal angeordnet und nicht vertikal. Mit der `slidesToShow` Option wird eingestellt, wie viele Elemente zur gleichen Zeit angezeigt werden. Mit der Option `dots` werden Elemente eingefügt, an denen der Nutzer erkennt, dass sich die Elemente wechseln lassen (Code 5.10).

Anpassungen die nicht in diesem Abschnitt genauer beschrieben wurden, erfolgten durch simple CSS Anpassungen. Unter anderem wurden diverse Schriften, Abstände etc. angepasst. Diese werden nicht beschrieben, weil die Arbeit dadurch keinen Mehrwert erhält.

5.3 Fehlende Standards

Die Test- und Wartungsphase wird in dieser Arbeit übersprungen, stattdessen geht es in diesem Abschnitt der Arbeit um zwei Probleme, die bei der Entwicklung des Prototyps aufgetaucht sind. Diese können nicht behoben werden, weil die aktuellen Standards nicht ausreichen. Diese Probleme müssten zu einem späteren Zeitpunkt optimiert werden, um ein einwandfreie Webseite zu erreichen, die für Smartwatches optimiert ist.

Das erste Problem bezieht sich auf die drei unterschiedlichen Displayformen. Das Layout der Webseite auf die Displayformen anzupassen ist möglich, aber die Abfrage, um das richtige Layout für die passende Displayform zu wählen fehlt. Die getätigte Abfrage im

Code 5.1 beinhaltet alle drei Displayformen. Dies führt zum Problem, dass die Webseite auf einem runden Display besser dargestellt wird, als auf einem rechteckigen. Um dieses Problem zu beheben, benötigt es eine neue Abfrage. Mit einem neuen CSS Media Query, der die Displayformen voneinander unterscheidet, könnte das jeweilige Layout dem richtigen Display zuweisen.

Das zweite Problem war, dass sich unter anderem Text nicht an runden Formen ausrichten. Das heißt der Viewport eines runden Displays wird nicht rund dargestellt, sondern als rechteckig. Dies führt zum Problem, dass Elemente die sich in den Ecken befinden, abgeschnitten werden. Texte zum Beispiel können bei runden Displays nicht am Bildschirmrand umbrechen.



Abbildung 5.16: Unterschiedliche Viewports in einem runden Display, eigene Grafik

Die Abbildung 5.16 zeigt unterschiedliche Arten eines Viewports. Die gelb markierten Bereiche zeigen an, wo sich die Webinhalte befinden. Aktuell richten sich die Inhalte wie in der linken Uhr an. Die Uhr in der Mitte zeigt den optimalen Zustand. Die Texte in einem runden Display passen sich der runden Displayform an. Für den Prototypen wurde die Lösung der rechten Uhr gewählt. Der ursprüngliche Viewport der linken Variante wurde dafür verringert. Zur Darstellung der Inhalte wird nur ein Teil des Displays genutzt. Somit können keine Inhalte aufgrund der Displayform abgeschnitten werden. Jedoch wird viel Fläche des Displays mit der Methode verloren. Um Elemente außerhalb eines Kreises richtig anzupassen, gibt es die CSS Eigenschaft `shape-outside`. Die CSS Eigenschaft `shape-inside` wird aktuell noch entwickelt. Diese Eigenschaft soll den Zustand der mittleren Uhr darstellen können (Abb. 5.16). Jedoch geht man davon aus, dass die heutigen Browser diese Eigenschaft nicht unterstützen werden (SelfHtml 2019).

6 Zusammenfassung

Dieses Kapitel bildet den Abschluss der Arbeit. Nach den verschiedenen Entwicklungsphasen, um einen Prototypen zu entwickeln, geht es im Abschnitt 6.1 um die Auswertung des Prototyps. Das Ziel der Auswertung soll sein, die zu Beginn der Arbeit gestellte Forschungsfrage zu beantworten: *Welche Methoden ergeben sich bei der Entwicklung einer Webanwendung, die für Smartwatches optimiert ist?*

Nach der Auswertung wird im Abschnitt 6.2 eine Zukunftsprognose aufgestellt. Die Zukunftsprognose beschäftigt sich mit dem Thema, wie die Entwicklung für Smartwatch optimierte Webseiten aussehen könnte. Dabei geht es um die Geräte, Webstandards, Testumgebungen und Notwendigkeit.

6.1 Auswertung des Prototyps

Der Prototyp wurde entwickelt und verfügt auf der Smartwatch-Version über die gleichen funktionalen Anforderungen die, die Desktop-Version bietet. Dafür mussten bei der Entwicklung unterschiedliche Punkte beachtet werden. Wie im Abschnitt 5.3 angesprochen, muss man bei der Entwicklung auf die verschiedenen Displayformen achten. Bei runden Displays sollte z.B. eine extra Optimierung vorgenommen werden. Somit ergibt sich die Methode, die Webinhalte nicht nur einmalig festzulegen, sondern für jede Displayform einzeln anzupassen.

Wie im Abschnitt 2.4 erwähnt, sind die Bildschirme der Smartwatches wesentlich kleiner, als die des aktuellen Standes. Bei der Gestaltung sollten somit nur Elemente eingebunden werden, die funktionale Anforderungen erfüllen. Eine Methode wäre es, die Elemente simple zu halten und nicht funktionale Elemente auszublenden. Mit dem kleinerem Bildschirmen verändern sich auch die Bedienungen. Ein gezieltes Element mit dem Finger zu treffen ist nicht leicht möglich, weil ein Finger schon reicht, um einen großen Teil der Fläche des Bildschirms zu verdecken. Beim Scrollen ist dies z.B. ein Problem, weil man nicht nachvollziehen kann, ob die gewünschten Inhalte angezeigt werden. Eine Methode um diese Problematik zu verhindern ist, dass der Nutzer beim scrollen eingeschränkt wird und nicht mehr die freie Scrollfunktion hat. Somit kann man dem Nutzer nur vordefinierte Bereiche anzeigen. Diese kann und muss der Nutzer anschließend nicht in die richtige Position

6 Zusammenfassung

bringen, um den gesamten Inhalt zu sehen. Eine weitere Methode ist es, dem Nutzer über seine Eingabe zu informieren. Im Bezug auf den Prototypen wurde diese Methode beim Aufbau des Menüs genutzt. Der Nutzer muss nicht gezielt einen Menüpunkt treffen, sondern kann sich den gewünschten Menüpunkt auswählen und anschließend auf einen beliebigen Menüpunkt klicken. Für die Nutzung von Eingabefeldern ergibt sich die Methode auf vordefinierte Eingabemasken zurück zu greifen. Wie im Abschnitt 2.4 werden diese von den jeweiligen Browsern zur Verfügung gestellt.

Für lange Inhalte kann ein Bereich des Bildschirms zu klein sein, um den gesamten Inhalt anzeigen zu können. Für den Prototypen wurde die Methode genutzt, die zusammengehörigen Inhalte auf der horizontalen Achse zu teilen und die nicht zusammengehörigen Bereiche auf der vertikalen Achse zu trennen. Somit ergeben sich verschiedene Ebenen. Die zusammengehörigen Inhalte können auf der selben Ebene vertikal gewechselt werden.

Allgemein betrachtet kann man für die Entwicklung von Smartwatch optimierten Webseiten viele Elemente der Anwendungsentwicklung übernehmen. Die Anwendungsentwicklung für Smartwatches wird schon seit der Smartwatch betrieben und ist viel Fortgeschrittener. Diese Methode kann einen vor der Entwicklung helfen, um die Bedienung der verschiedenen Komponenten zu testen. Betrachtet man den Verlauf der Entwicklungsphasen, sind unterschiedliche Methoden zusammengekommen, die bei der Entwicklung von Smartwatch optimierten Webseiten unterstützen können.

6.2 Ausblick

Mit Ausblick auf die Zukunft, kann die Entwicklung von Smartwatch optimierten Webseiten von unterschiedlichen Technologien profitieren. Damit diese Technologien entwickelt werden, muss die Relevanz dieser Webseiten gesteigert werden. Der WIB und das Apple WebKit bieten die Möglichkeit zukünftig die Nachfrage von Smartwatch optimierten Webseiten zu steigern. Steigt die Relevanz könnten standardisierte Browser folgen. Unabhängig vom Betriebssystem können die Webseiten nach ein und den selben Standard entwickelt werden. Ebenso können verschiedene Testumgebungen auf einem Desktop folgen. Aktuell ist es schwierig, die verschiedenen Displayformen bei der Entwicklung zu simulieren. Zukünftig könnte es eine Testumgebung geben die, die unterschiedlichen Displayformen simulieren kann. Bei der Entwicklung bleibt zukünftig dennoch das Problem, dass die unterschiedlichen Displayformen auch unterschiedliche Gestaltungsvarianten benötigen. Wie im Abschnitt 5.3 beschrieben, können angepasste Webstandards dieses Problem erleichtern. Bei der Programmierung könnten angepasste Webstandards helfen, um auf die unterschiedlichen Displayformen eingehen zu können.

6 Zusammenfassung

Im Bezug auf die Technologien der Gräte, kann man zukünftig vor allem von Leistungsoptimierungen ausgehen. Änderungen am Display, die bei der Entwicklung berücksichtigt werden müssen, sind eher unwahrscheinlich. Die Auflösungen werden vermutlich höher, aber die Displaymaße bleiben gleich. Die Displaymaße orientieren sich an reguläre Armbanduhren und diese Maße sind mittlerweile erreicht. Vereinzelt könnten Hersteller diese Orientierung nicht einhalten, dennoch wäre die Größe einer Uhr, die man am Handgelenk trägt beschränkt. Weiterhin könnten die Geräte in der Bedienung optimiert werden. Zukünftige Browser können z.B. auf Sprachsteuerung setzen. Somit wäre bei der Entwicklung eine Bedienung per Sprachbefehle zu beachten.

Glossar

API	API bezeichnet eine Schnittstelle zu einer Software, die Entwicklern Funktionen zur Verfügung stellt, durch die man Teile des Programmes steuern kann.
Above the Fold	Above the Fold definiert den Teil einer Webseite, der für den Nutzer ohne zu Scrollen sichtbar ist.
Apple WebKit	WebKit ist eine freie HTML-Rendering-Engine. Mit dem WebKit können Webseiten in Webbrowsern dargestellt werden.
Application	Application steht für Anwendung und beschreibt eine Software, die zahlreiche Funktionen erfüllen kann.
Browser	Browser steht umgangssprachlich für Webbrowser. Ein Browser ist eine Software, die Internetseiten aufrufen und entsprechend anzeigen kann.
Button	Ein Button bedeutet Schaltfläche und ist eine Fläche auf einer Webseite oder in einer Anwendung, die durch das Anklicken eine Aktion auslöst.
Captcha	Ein Captcha wird verwendet, um zu kontrollieren, ob ein Mensch oder eine Maschine eine Aktion durchgeführt hat. Captcha wird häufig für Internetformulare genutzt.
Chin	Die Chin ist eine Einsparung im Display einer Smartwatch und verringert den Viewport einer Anwendung bzw. Webseite.
Content	Content heißt übersetzt Inhalte und beschreibt alle Inhalte einer Webseite.
Desktop	Der Desktop ist die zentrale Arbeitsfläche eines Computers.
Device Mode Emulator	Der Device Mode Emulator ist eine Funktion von dem Google Chrome Browser. Mit Hilfe des Emulators kann man verschiedene mobile Geräte simulieren.

Digital Crown	Eine Digital Crown ist ein Drehrad an der Seite des Gerätes, ähnlich wie bei einer analogen Armbanduhr. Mit der Digital Crown kann man z.B. scrollen, ohne mit dem Finger das Display zu berühren.
Doctype	Doctype bedeutet auf deutsch Dokumententyp und definiert die Art eines Webdokumentes.
Domain	Eine Domain ist die Web-Adresse einer Webseite, mit dem der Nutzer die Webseite aufrufen kann.
Dropdown-Menü	Ein spezielles Menü im Web, welches sich durch seine dynamische Funktionsweise auszeichnet. Beim Dropdown-Menü sind die Unterpunkte zu beginn ausgeblendet und werden eingeblendet, sobald man sich mit der Maus über dem jeweiligen Hauptpunkt befindet.
Eyecatcher	Ein Eyecatcher ist in der Werbung und allen visuellen Medien ein Grafik- oder Text-Element, das die Aufmerksamkeit des Nutzer auf eine bestimmte Botschaft lenken soll.
Footer	Der Footer bezeichnet den unteren Teil einer Webseite.
Header	Der Header bezeichnet den oberen Teil einer Webseite.
Hole Punch Display	Ein Hole Punch Display hat eine runde Aussparung im Display. Die Aussparung bietet den Platz für die Frontkamera eines Gerätes.
Link	Link oder genau Hyperlink ist ein Querverweis in einem Hypertext. Sobald auf einen Link geklickt wird, wird automatisch das darin angegebene Ziel aufgerufen.
Logo	Ein Logo ist ein grafisches Zeichen, das z.B. ein Unternehmen repräsentiert.
Media Queries	Media Queries bedeutet auf Deutsch Medienabfrage. Mit Hilfe von Media Queries kann man in CSS die Darstellung eines Dokumentes an verschiedene Ausgabemedien anpassen.
Notch	Eine Notch ist eine Aussparung am oberen Displayrand eines Gerätes. Hinter der Aussparung befinden sich unterschiedliche Sensoren und die Frontkamera.

Glossar

Plugin	Ein Plugin ist eine Softwarekomponente, die man in eine bestehende Software einsetzen kann.
Prototyp	Ein Prototyp ist eine Erstversion oder auch ein Arbeitsmodell eines neuen Produktes. Anhand eines Prototyps kann man unter anderem die Machbarkeit des Designs beurteilen und Probleme erkennen.
Responsive Design	Ein Responsive Design beschreibt den Aufbau einer Webseite, die sich flexibel an der Größe des Browserfensters anpasst.
Slider	Ein Slider im Zusammenhang mit dem Web, steht für ein Element, welches nach einer bestimmten Zeit oder durch eine manuelle Eingabe seinen Inhalt wechselt.
Smartphone	Ein Smartphone ist ein Mobiltelefon mit einem Touchscreen und weitere zusätzliche Funktionen.
Smartwatch	Eine Smartwatch ist eine digitale Uhr, die ähnliches wie ein modernes Handy bedient wird und im Vergleich zu einer analogen Uhr diverse extra Funktionen bietet.
Submenü	Das Submenü oder auch Untermenü beschreibt ein Menü, welches sich innerhalb eines Menüs befindet.
Syntax	Syntax definiert in der Programmierung ein Regelsystem, welches vorgibt wie man seinen Code richtig aufbaut und schreibt.
Tablet	Ein Tablet ist ein kleiner, tragbarer Computer mit einem Touchscreen. Das Display des Tablets ist im Vergleich zum Smartphone wesentlich größer.
Tag	Ein Tag wird im Zusammenhang mit HTML zur Kennzeichnung eines Elementes genutzt.
Teaser	Ein Teaser ist ein kurzer Textausschnitt, welches die Neugierig eines Nutzers wecken soll.
Touch	Touch beschreibt die Interaktion des Nutzer mit einem Touchscreen. Ein Touch bedeutet dabei soviel wie z.B ein klick.

Glossar

Touchscreen	Ein Touchscreen ist eine Art von Display, die durch Berührungen mit dem Finger gesteuert werden kann.
User Stories	User Stories sind einfach formulierte Sätze, die aus der Nutzersicht, die Anforderungen an ein Softwareprodukt stellen.
Viewport	Der Viewport definiert die zur Verfügung stehende Displaygröße eines Gerätes.
Wearables	Wearables sind Computertechnologien die man am Körper trägt.

Abkürzungsverzeichnis

API	Application Programming Interface
CSS	Cascading Style Sheet
DPR	Display Pixel Ratio
GSM	Global System for Mobile Communication
HTML	Hypertext Markup Language
ID	Selektor für CSS
KB	Kilobyte
LTE	Long Term Evolution
OS	Operating System
px	Pixel
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
vh	Viewport Height
WIB	Wear Internet Browser
WLAN	Wireless Local Area Network
vw	Viewport Width
XHTML	Extensible Hypertext Markup Language

Abbildungsverzeichnis

2.1	<i>Technische Zeichnungen von verschiedenen Smartwatch-Varianten, eigene Grafik.</i>	12
2.2	<i>Unterschiedlichen WebKit Eingabemasken (Apple Developer 2018-45)</i>	13
5.1	<i>Die Kopfzeile der Desktop-Version, eigene Grafik</i>	31
5.2	<i>Der Banner der Desktop-Version, eigene Grafik</i>	31
5.3	<i>Das Formular für die Sendungsverfolgung, eigene Grafik</i>	32
5.4	<i>Die erste Ansicht des Sendungsverlaufes, eigene Grafik</i>	32
5.5	<i>Die zweite Ansicht des Sendungsverlaufes mit dem aktuellen Status, eigene Grafik</i>	33
5.6	<i>Die dritte Ansicht des Sendungsverlaufes mit dem genauen Ablauf, eigene Grafik</i>	33
5.7	<i>Ansicht mit drei Teaser auf drei Spalten verteilt, eigene Grafik</i>	33
5.8	<i>Die Fußzeile der Desktop-Version, eigene Grafik</i>	34
5.9	<i>Die Kopfzeile der Smartwatch-Version mit der Navigation, eigene Grafik</i>	35
5.10	<i>Das Formular für die Sendungsverfolgung der Smartwatch-Version, eigene Grafik</i>	36
5.11	<i>Die erste Ansicht des Sendungsverlaufes der Smartwatch-Version, eigene Grafik</i>	36
5.12	<i>Die zweite Ansicht des Sendungsverlaufes der Smartwatch-Version, eigene Grafik</i>	37
5.13	<i>Die dritte Ansicht des Sendungsverlaufes der Smartwatch-Version, eigene Grafik</i>	37
5.14	<i>Die Ansicht der drei Teaser auf der Smartwatch-Version, eigene Grafik</i>	38
5.15	<i>Die Fußzeile der Smartwatch-Version mit der Navigation, eigene Grafik</i>	38
5.16	<i>Unterschiedliche Viewports in einem runden Display, eigene Grafik</i>	46

Code Beispiele

3.1	<i>HTML Viewport Metatag (Jay Kang 2018)</i>	15
3.2	<i>Exemplarisches CSS Media Query</i>	15
3.3	<i>JavaScript Beispiel für eine mobile Navigation</i>	17
5.1	<i>CSS Media Query welches für Smartwatches wirkt</i>	39
5.2	<i>Code Abschnitt für das Öffnen des Menüs der Smartwatch-Version</i>	40
5.3	<i>Code Abschnitt für das Öffnen des Untermenüs der Smartwatch-Version</i>	41
5.4	<i>Code Abschnitt für das Schließen der Menüs</i>	41
5.5	<i>Code Abschnitt für die Slick Slider Konfiguration</i>	42
5.6	<i>Exemplarischer Code für das Verlinken der Menüpunkte</i>	43
5.7	<i>Verschiebung der rechtlichen Links aus dem Footer</i>	44
5.8	<i>Eingabefeld für die Sendungsnummer</i>	44
5.9	<i>Code Abschnitt für die Animation des Textes</i>	44
5.10	<i>Konfiguration des Slick Slider für die Teaser</i>	45

Literaturverzeichnis

Android Developers: *Android Developers Beitrag über die Layout Entwicklung für Smartwatches*, URL: <https://developer.android.com/training/wearables/ui/layouts>, 2019, letzter Zugriff: 29.07.2019

Apple: *Apple Beitrag über Cellular*, URL: <https://www.apple.com/de/shop/question/answers/readonly/tag/ipad/was-bedeutet-cellular-modell/QH9D279J2KACUCXH2>, 2012, letzter Zugriff: 19.07.2019

Apple: *Apple Watch Mobilfunkanbieter*, URL: <https://www.apple.com/de/watch/cellular/>, 2019, letzter Zugriff: 19.07.2019

Apple Developer: *Apple Developer Beitrag zur Entwicklung von Webinhalten für WatchOS*, URL: <https://developer.apple.com/videos/play/wwdc2018/239/>, 2018, letzter Zugriff: 10.09.2019

Areamobile: *Areamobile Datenbank mit Tablet-Daten*, URL: <http://www.aramobile.de/tablets>, 2019, letzter Zugriff: 12.07.2019

Ben Otterstein: *Artikel von MacLife über der Watch OS Betriebssystem*, URL: <https://www.maclife.de/ratgeber/watchos-5-rufen-webseiten-apple-watch-100106065.html>, 2018, letzter Zugriff: 19.07.2019

DHL Paket GmbH: *DHL Webseite*, URL: <https://www.dhl.de>, 2019, letzter Zugriff: 16.09.2019

DPD Deutschland GmbH: *DPD Webseite*, URL: <https://www.dpd.com/de/>, 2019, letzter Zugriff: 16.09.2019

David Smith: *Statistik über die meistgenutzten iPhones von David Smith*, URL: <https://david-smith.org/iosversionstats/>, 2019, letzter Zugriff: 12.07.2019

Literaturverzeichnis

Device Atlas: *Statistik über die populärsten Android Smartphones von DeviceAtlas*, URL: <https://deviceatlas.com/blog/most-popular-android-smartphones>, 2019, letzter Zugriff: 12.07.2019

Dino Grandoni: *Online Artikel von Huffpost über die weltweit erste Webseite*, URL: https://www.huffingtonpost.com/2012/08/06/worlds-first-website_n_1747476.html, 2017, letzter Zugriff: 06.07.2019

Dr. Veikko Krypczyk: *Onlineartikel über Lastenheft: Anforderungen in der Softwareentwicklung richtig definieren*, URL: <https://entwickler.de/online/development/lastenheft-anforderungen-softwareentwicklung-255339.html>, 2016, letzter Zugriff: 26.09.2019

Dr. Veikko Krypczyk: *Onlineartikel über Lastenheft: Anforderungen in der Softwareentwicklung richtig definieren*, URL: <https://entwickler.de/online/development/lastenheft-anforderungen-webprojekte-definieren-255382.html>, 2016, letzter Zugriff: 26.09.2019

Eric Enge: *Desktop vs Mobile Studie von Perficient*, URL: <https://www.perficientdigital.com/insights/our-research/mobile-vs-desktop-usage-study/>, 2019, letzter Zugriff: 07.07.2019

Gartner: *Pressemitteilung von Gartner*, URL: <https://www.gartner.com/en/newsroom/press-releases/2019-02-21-gartner-says-global-smartphone-sales-stalled-in-the-fourth-quart>, 2019, letzter Zugriff: 12.07.2019

Google Arts & Culture: *Google Arts & Culture Beitrag über das World Wide Web*, URL: <https://artsandculture.google.com/theme/eAJS4WcKh7UBIQ>, 2019, letzter Zugriff: 29.07.2019

Google Developer: *Google SEO-Überblick für Mobilgeräte*, URL: <https://developers.google.com/search/mobile-sites/mobile-seo/>, 2019, letzter Zugriff: 07.07.2019

Google Play: *Web Browser for Wear OS App aus dem Google Play Store*, URL: <https://play.google.com/store/apps/details?id=com.appfour.wearbrowser&hl=de>, 2019, letzter Zugriff: 20.07.2019

Google Watch Blog: *Artikel vom GoogleWatchBlog über das Comeback der Google Glass*, URL: <https://www.googlewatchblog.de/2019/03/google-glass-comeback-sieben/>, 2019, letzter Zugriff: 19.07.2019

Literaturverzeichnis

Hermes Germany GmbH: *Hermes Webseite*, URL: <https://www.myhermes.de/>, 2019, letzter Zugriff: 16.09.2019

IDC: *Tablet Marktanalyse von IDC*, URL: <https://www.idc.com/getdoc.jsp?containerId=prUS44425918>, 2018, letzter Zugriff: 12.07.2019

IDC: *Wearables Analyse von IDC*, URL: <https://www.idc.com/getdoc.jsp?containerId=prUS44276818>, 2018, letzter Zugriff: 19.07.2019

Informationszentrum-Mobilfunk: *Artikel über das 5g-Netz*, URL: <http://www.informationszentrum-mobilfunk.de/technik/funktionsweise/5g>, 2019, letzter Zugriff: 07.07.2019

Inside Digital: *Inside Digital Datenbank mit Smartphone-Daten*, URL: <https://www.inside-digital.de/handy/datenbank>, 2019, letzter Zugriff: 12.07.2019

jQuery: *Jquery Webseite*, URL: <https://jquery.com/>, 2019, letzter Zugriff: 15.09.2019

James Kielty: *Artikel von mobiForge über die Displayeigenschaften, der populärsten Smartphones*, URL: <https://mobiforge.com/research-analysis/viewport-resolution-diagonal-screen-size-and-dpi-for-the-most-popular-smartphones>, 2019, letzter Zugriff: 12.07.2019

Jay Kang: *SEOptimer Beitrag über den HTML Viewport*, URL: <https://www.seoptimizer.com/blog/viewport/>, 2018, letzter Zugriff: 10.09.2019

Jerri Ledford: *Artikel von Lifewire über Foldable Phones*, URL: <https://www.lifewire.com/what-is-a-foldable-phone-4178374>, 2019, letzter Zugriff: 24.07.2019

Ken Wheeler: *Slick Plugin von Ken Wheeler*, URL: <https://kenwheeler.github.io/slick/>, 2019, letzter Zugriff: 18.09.2019

Luke Haas: *Scrollify Plugin von Luke Haas*, URL: <https://projects.lukehaas.me/scrollify/>, 2019, letzter Zugriff: 17.09.2019

MDN web docs: *MDN web docs JavaScript Dokumentation*, URL: <https://developer.mozilla.org/de/docs/Web/API/Document/getElementById>, 2019, letzter Zugriff: 15.09.2019

Literaturverzeichnis

Masha Fisch: *Studie von Google Mobile Ads Marketing*, URL: <http://googlemobileads.blogspot.com/2012/09/mobile-friendly-sites-turn-visitors.html>, 2012, letzter Zugriff: 07.07.2019

Meiert: *Meiert Publikation mit einer Übersicht aller CSS-Eigenschaften*, URL: <https://meiert.com/de/publications/indices/css-properties/>, 2019, letzter Zugriff: 15.09.2019

Noël Bossart: *Namics Beitrag über Flexible & erweiterbare jQuery Plugins*, URL: <https://blog.namics.com/2010/08/create-flexible-jquery-plugins.html>, 2010, letzter Zugriff: 16.09.2019

Prof. Dr. Oliver Bendel: *Artikel vom Gabler Wirtschaftslexikon über Wearables*, URL: <https://wirtschaftslexikon.gabler.de/definition/wearables-54088/version-368816>, 2019, letzter Zugriff: 19.07.2019)

Ryan Block: *Engadget Beitrag über die Steve Jobs Apple Keynote die am 01.09.2007 vorgetragen wurde*, URL: https://www.engadget.com/2007/01/09/live-from-macworld-2007-steve-jobs-keynote/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xILmNvbS8&guce_referrer_sig=AQAAABVWK374oKZcKgQ4EwoNpQjUxY-H6344qIqXWJgkL-KkBVZp-Vc_wsRonqg_jLpHMUlaDRCattM2idMUA-8nfcZkbtBq0FPg8czoaL6uNeeSB_S7QIWtDt7w5JNZABeX-8jz1qVNw5tLrjxmooulHsAcWzDwQcxBTwoC2dDrYen, 2007, letzter Zugriff: 06.07.2019

Ryte: *Online Artikel von Ryte über Mobile First*, URL: https://de.ryte.com/wiki/Mobile_First, 2018, letzter Zugriff: 06.07.2019

Sarah Silbert: *Artikel über Smartwatches von Lifewire*, URL: <https://www.lifewire.com/an-introduction-to-smart-watches-3441381>, 2019, letzter Zugriff: 19.07.2019

SelfHtml: *SELFHTML-Wiki Beitrag über CSS/Media Queries*, URL: https://wiki.selfhtml.org/wiki/CSS/Media_Queries, 2019, letzter Zugriff: 10.09.2019)

SelfHtml: *SELFHTML-Wiki Beitrag über CSS/Eigenschaften/Positionierung/shape*, URL: <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Positionierung/shape#shape-inside>, 2019, letzter Zugriff: 14.09.2019

Literaturverzeichnis

StatCounter: *Internetnutzung über mobile Endgeräte übersteigt die über Desktops*, URL: <https://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>, 2016, letzter Zugriff: 07.07.2019

Statista: *Statistik über Absatzzahlen von Tablets, Smartphones und PCs von Statista*, URL: <https://de.statista.com/statistik/daten/studie/256337/umfrage/prognose-zum-weltweiten-absatz-von-tablets-pcs-und-smartphones/>, 2019, letzter Zugriff: 12.07.2019

Statistisches Bundesamt: *Pressemitteilung des Statistischen Bundesamtes*, URL: https://www.destatis.de/DE/Presse/Pressemitteilungen/2018/09/PD18_330_634.html, 2018, letzter Zugriff: 12.07.2019

VIZ Devices: *Apple Watch Series 4 (44mm) Displayeigenschaften*, URL: <https://vizdevices.yesviz.com/devices/apple-watch-series-4-44mm/>, 2018, letzter Zugriff: 26.07.2019

VIZ Devices: *Apple Watch Series 4 (40mm) Displayeigenschaften*, URL: <https://vizdevices.yesviz.com/devices/apple-watch-series-4-40mm/>, 2018, letzter Zugriff: 26.07.2019

Vera Scheuering: *Online Artikel von Ryte über den Above the Gold Bereich*, URL: <https://de.ryte.com/magazine/above-the-fold-webseiten-richtig-aufbauen>, 2014, letzter Zugriff: 12.07.2019

W3C: *W3C Beitrag über das World Wide Web Consortium*, URL: <https://www.w3.org/Consortium/>, 2019, letzter Zugriff: 29.07.2019

W3C: *W3C Beitrag über HTML & CSS*, URL: <https://www.w3.org/standards/webdesign/htmlcss>, 2016, letzter Zugriff: 29.07.2019

W3C: *W3C Beitrag über JavaScript Web APIs*, URL: <https://www.w3.org/standards/webdesign/script.html>, 2016, letzter Zugriff: 12.09.2019

Wear OS by Google: *WearOS by Google design guidelines*, URL: <https://designguidelines.withgoogle.com/wearos/resources/downloads.html>, 2019, letzter Zugriff: 16.09.2019

Wikipedia: *Responsive Webdesign Wikipedia Eintrag*, URL: https://de.wikipedia.org/wiki/Responsive_Webdesign, 2019, letzter Zugriff: 06.07.2019

Literaturverzeichnis

Wikipedia: *Hypertext Markup Language* Wikipedia Artikel, URL: https://de.wikipedia.org/wiki/Hypertext_Markup_Language, 2019, letzter Zugriff: 29.07.2019

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

01.11.2019, Hamburg

Kevin Brandao da Graca