

Inhalt

Vorwort	13
Bevor es richtig losgeht – Installation eines lokalen Webservers	15

1 Die Konzeptionsphase 23

1.1 Quick and dirty: Warum eine App?	23
1.1.1 Native Apps vs. WebApp	23
1.1.2 Eine App, sie zu knechten, sie alle zu binden – oder so ähnlich	24
1.1.3 Immer auf dem neuesten Stand – Trends	26
1.1.4 Sicherheit	29
1.2 Die Suche nach dem Lichtschalter – App-Ideen finden	30
1.2.1 Tipp #1 – Liebling, ich habe die Ideen geschrumpft	31
1.2.2 Tipp #2 – Ideen und Probleme sofort festhalten	31
1.2.3 Tipp #3 – Ideen und Probleme verändern sich	31
1.2.4 Tipp #4 – Ideen teilen	31
1.2.5 Tipp #5 – Ideen vertiefen oder verwerfen	32
1.2.6 Eine Idee weiterentwickeln	32
1.3 Jeder ist Endbenutzer – stellen Sie ihn in den Mittelpunkt	35
1.3.1 Die App als 5-Sterne-Luxushotel	35

2 Das technische Grundgerüst 45

2.1 HTML5 – Definition und aktueller Stand	45
2.1.1 HTML als Standard	46
2.1.2 HTML5 als Marketinginstrument	46
2.1.3 Der Aufbau einer einfachen HTML5-Seite	46
2.1.4 Semantische Elemente in HTML5	49
2.1.5 CSS3 – Grundlagen und Neuerungen	51
2.1.6 Was ist eigentlich CSS?	51
2.1.7 Animieren mit »webkit-animation«	73
2.1.8 Automatische Silbentrennung	81
2.1.9 Media Queries – Definitionen für bestimmte Bildschirmgrößen	82
2.1.10 Komplexe Layouts einfach gebaut – mit CSS Flexbox	85
2.1.11 Von der Webseite zur WebApp	96

2.2	Crashkurs JavaScript	102
2.2.1	Die JavaScript-Syntax	103
2.2.2	Variablen und mathematische Operationen	104
2.2.3	Objekte	106
2.2.4	Arrays	107
2.2.5	Kontrollstrukturen	108
2.2.6	DOM-Manipulation	110
2.3	Funktionen und Events	115
2.3.1	EventListener	120
2.3.2	It's Hammertime!	121
2.3.3	Testen von TouchEvents	123
2.3.4	Eine Zeichen-App mit Canvas und TouchEvents	125
2.4	Xhr und Json	135
2.4.1	XMLHttpRequest	136
2.4.2	Daten abrufen mit JSON	138
2.4.3	Limitationen und Sicherheit	142

3 Design – das Feeling einer App

3.1	UI für Mobile und Desktop	146
3.1.1	Webdesign	146
3.1.2	App Design	147
3.1.3	Touch Design	147
3.1.4	Design Patterns	149
3.1.5	Die drei Prinzipien von Marissa Mayer	156
3.1.6	Designsprachen	159
3.1.7	Responsive Design	163
3.1.8	Fazit	166
3.2	Material Design	166
3.2.1	Gestaltungsprinzipien	167
3.2.2	Gestaltungsmittel	169
3.2.3	Unterschiede zu Flat Design	177
3.2.4	Frameworks und Beispiele	178
3.3	Multiscreen-Apps	180
3.3.1	Fangen Sie klein an – Mobile First	180
3.3.2	Muss es im ersten Schritt überall sein?	182
3.4	Inspiration	183

4	Schneller zur eigenen App mit Frameworks	191
4.1	Responsive Apps mit Twitter Bootstrap	192
4.1.1	Grundlagen	193
4.1.2	Die Instagram-App	198
4.1.3	Nützliche Links, Vorlagen und Erweiterungen	207
4.2	Apps erstellen mit dem Ratchet-Framework	210
4.2.1	Grundprinzipien von Ratchet	211
4.2.2	Vorbereitungen	211
4.2.3	Grundstruktur Ihrer App	213
4.2.4	Komponenten	214
4.2.5	Themes	217
4.2.6	Rapid Prototyping mit Push.js	220
4.2.7	Ein Kontaktformular mit Ratchet und Formspree	225
4.2.8	Fazit	230
4.3	Bewegte Prototypen mit Framer.js	230
4.3.1	Was ist Framer.js?	232
4.3.2	Vorbereitungen	232
4.3.3	Grundlagen	234
4.3.4	Take it to the next level	241
4.3.5	Framer Studio für Mac OS	244
4.3.6	Vorschau des Prototyps über Apps für Android und iOS	248
4.3.7	Das automatische Importieren von Grafiken mit Framer Generator auf Mac OS	249
4.3.8	Abschließende Worte zu Framer.js	251
5	Positionsbestimmung	253
5.1	Die Positionsbestimmung mit HTML5	255
5.1.1	Weitere Eigenschaften der Positionsbestimmung	257
5.2	Die Where-to-Eat-App	258
5.2.1	Position auf einer statischen Karte eintragen	260
5.2.2	Interaktive Karten mit der Google Maps JavaScript API einbinden	267
5.2.3	Das App-Grundgerüst anlegen	269
5.2.4	Die Navigationsleiste	271
5.2.5	Liste der Restaurants laden und ausgeben	272
5.2.6	Den Abstand zwischen zwei Koordinaten berechnen	276

5.2.7	Die Optionen	278
5.2.8	Der letzte Schliff	280
5.3	More Fun mit Geodaten	282
5.3.1	Yelp-API	282
5.3.2	Foursquare-API	283
5.3.3	Google Places	283
5.3.4	Flickr-API	285

6 Auslesen des Bewegungssensors mit JavaScript 287

6.1	Diese Daten liefert der Bewegungssensor	288
6.2	Vorüberlegungen zur App »Shake it like a Polaroid picture«	291
6.3	Shake it like a Polaroid picture – die Umsetzung	292
6.3.1	Die HTML-Datei	292
6.3.2	Laden eines Bildes aus dem Fotodienst Flickr via JSON	295
6.3.3	Die CSS-Datei	301
6.3.4	Das Laden eines Polaroids	303
6.3.5	Sahnehäubchen – eine leichte Rotation des Polaroids	307
6.3.6	Jetzt aber mit extra viel Sahne – das Gesamtbild anzeigen und mithilfe des Bewegungssensors bewegen	309
6.4	Die Anzeige eigener Bilder mit dem »input«-Element	314
6.4.1	Einführung in den Dateiupload	315
6.4.2	Erweitern der Shake-it-WebApp um eigene Fotos – Übersicht	316

7 Offline – damit eine App in jedem Winkel der Welt funktioniert 327

7.1	Abhilfe schafft das Cache-Manifest	328
7.1.1	Die Struktur der Cache-Manifest-Datei	329
7.1.2	Ihre erste Cache-Manifest-Datei	330
7.1.3	Wann werden welche Daten gecacht?	331
7.1.4	Die Cache-Manifest-Datei im Einsatz	332
7.2	Der Local Storage – die Offlinedatenbank	333
7.2.1	»localStorage«-Funktionen	334
7.2.2	Temporäre Speicherung von Daten im Session Storage	334

7.3	Die Offline-To-do-App	335
7.3.1	Welche Anforderungen muss Ihre To-do-App erfüllen?	336
7.3.2	Der Wireframe der To-do-App	336
7.3.3	Das Grundgerüst mit Bootstrap	338
7.3.4	Aufgaben hinzufügen und abhaken mit JavaScript	340
7.3.5	Aufgaben speichern mit »Local Storage«	344
7.3.6	CSS-Anpassungen	347
7.3.7	Cache-Manifest	348
7.4	Checkliste zum Überprüfen der Offlinefähigkeit einer WebApp	349
8	Backend als Webservice	351
8.1	Übersicht und Auswahl	351
8.1.1	Parse	352
8.1.2	Firebase	353
8.1.3	Orchestrate	354
8.1.4	Heroku	355
8.1.5	Meteor	357
8.2	Eine App mit Firebase-Backend entwickeln	358
8.2.1	Die Firebase-To-do-Liste	362
8.2.2	Daten speichern per JavaScript-Client	364
8.2.3	Erweiterte Features	368
8.3	Eine App für mehrere Benutzer	375
8.3.1	Authentifizierung mit E-Mail	377
8.3.2	Sicherheit mit Firebase	382
8.3.3	Zusätzliche Features	385
8.4	Deploy mit Firebase	395
8.5	Rechtliches	397
8.6	Fazit	397
9	Native Anwendungen und App Stores	399
9.1	Die Vor- und Nachteile einer nativen App	399
9.1.1	Die Vorteile einer nativen App	399
9.1.2	Die Nachteile einer nativen App	400

9.2	Hybrid Mobile Apps – eine WebApp wird zur nativen Anwendung	401
9.2.1	Appcelerator Titanium	401
9.2.2	PhoneGap bzw. Cordova	402
9.2.3	Ionic	403
9.2.4	Appium	403
9.2.5	Trigger.io	404
9.3	Apps mit Cordova	404
9.3.1	Vorbereitung	404
9.3.2	Ein Spiel mit Cordova	407
9.3.3	Grundlagen	411
9.3.4	Runde für Runde – JavaScript, die Erste	414
9.3.5	Die Positionierung einer Zielscheibe	415
9.3.6	Das Resultat einblenden	416
9.3.7	Hinzufügen von nativen Funktionen	418
9.3.8	Lasset die Duelle beginnen	421
9.4	Die Erstellung der App mit PhoneGap Build	422
9.4.1	Apple-Zertifikate und Profile	423
9.4.2	Eine App ID erstellen	427
9.4.3	Geräte registrieren	428
9.4.4	Profile für Development und Distribution erstellen	429
9.4.5	Die Erstellung eines Keystore für die Kompilierung der Android-App	432
9.4.6	Ihre App mit PhoneGap kompilieren	435
9.5	Native Apps mit React Native	441
9.5.1	Installation	442
9.5.2	Ihr erstes Projekt mit React Native	443
9.5.3	Eine native Buchliste mit React	444
9.5.4	Wie geht's weiter?	450

10 Testen mit einem Testpublikum und die Veröffentlichung in einem App Store

453

10.1	Zum Testen für Freunde und Familie: TestFairy	453
10.1.1	Was ist TestFairy?	454
10.1.2	Das Hochladen einer iOS-App-Datei	455
10.1.3	Das Hochladen einer Android-App-Datei	459
10.1.4	Die Testphase	461

10.2 App-Marketing	462
10.2.1 Name	463
10.2.2 Beschreibung und Keywords	466
10.2.3 Icon, Screenshots und Produktvideo	467
10.2.4 Eine Landing Page gestalten	475
10.3 Die Veröffentlichung Ihrer App in einem App Store	481
10.3.1 Das Veröffentlichen einer App in Apples App Store	482
10.3.2 Das Veröffentlichen einer App in Google Play	488
10.4 Ihre App bekannt machen	491
10.4.1 Mit den Nutzern in Kontakt bleiben – Customer Care	495
10.5 Abschließende Worte	496
 <hr/>	
Anhang: Weiterführende Informationen	497
A.1 Szene und Start-ups	497
A.2 JavaScript und nützliche Tools	498
A.3 iOS, Apple	498
A.4 Android, Google	499
A.5 Technologie und Gadgets	499
A.6 Design und Inspiration	499
A.7 Sonstiges	500
A.8 Interessante Twitter-Kontakte	500
 <hr/>	
Index	503