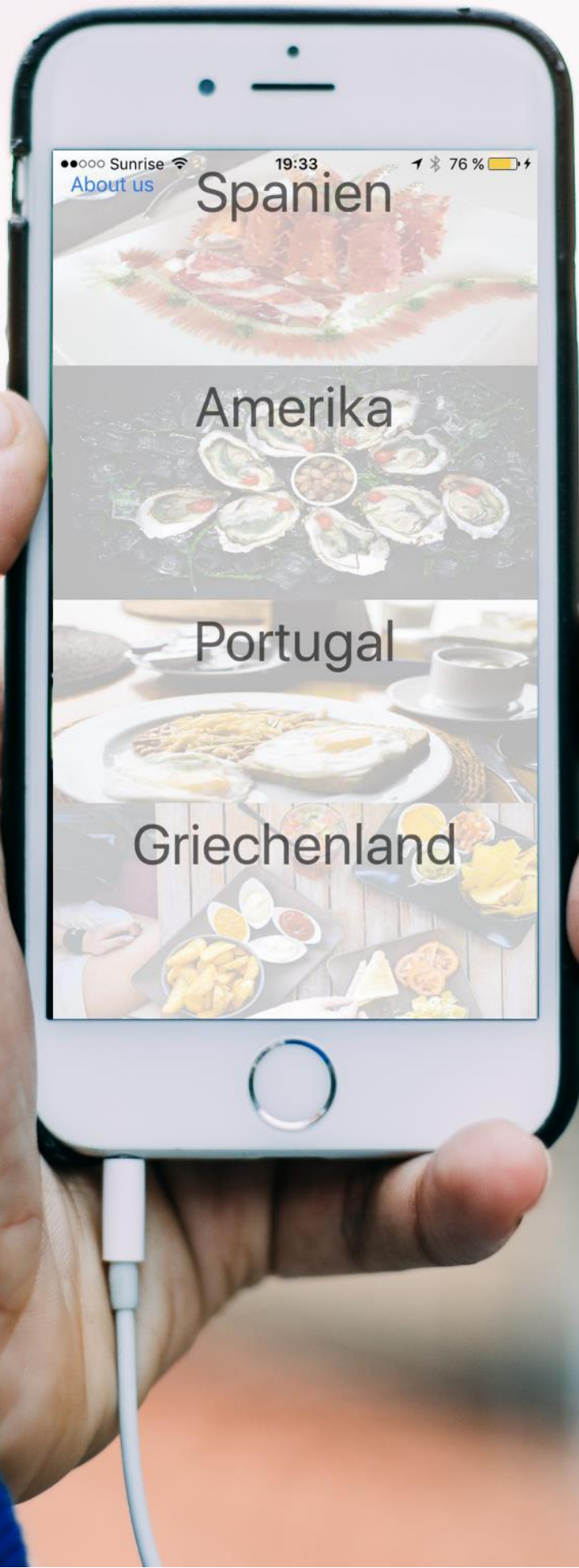




KATW

Kitchen around the World

Philipp Brügger

OZ Buelen

12.06.2017

Roland Bruderer



Designed by Philipp in Unterwasser

Inhaltsverzeichnis

1. Einleitung	4
2. Hauptteil	5
2.1 Idee	
2.2 Vorbereitung	6
2.3 Programmierung	7
2.4 Beispiel	8
2.5 Testphase	9
2.6 Publikation	10
2.7 Kompatibilität	11
3. Reflexion	12
4. Interessante Fakten	14
5. Quellenverzeichnis	15
6. Downloadlink	16

1. Einleitung

Applikationen, wir brauchen sie in unserem Alltag um zu telefonieren, den Busfahrplan anzuschauen, Essen zu bestellen, mit unseren Freunden zu chatten und viele weitere täglich zu bewältigende Aufgaben. Dabei sollten sie uns den Alltag erleichtern und unsere Produktivität steigern.

Persönlich benutze ich jeden Tag Snapchat, eine App mit der man Freunden schnell und unkompliziert Fotos senden kann, welche dann nur für einen von sich selbst festgelegten Zeitraum sichtbar sind, um so meinen Alltag visuell mit ihnen zu teilen. Deshalb nahm es mich einmal wunder, was die Entwickler bei Snapchat den ganzen Tag arbeiten und ob man wirklich ein Computernerd sein muss um programmieren zu können.

Über diese kleinen Programme auf unseren Smartphones wollte ich mehr erfahren und aufgrund meines grossen Interesse in Sachen Informatik beschloss ich eine eigene App zu entwickeln, gestalten sowie zu programmieren.

Im Sommer werde ich mit meiner Lehre als Informatiker, Fachbereich Applikationsentwicklung, starten. Deshalb sah ich dies als gute Vorbereitung für meine zukünftige Ausbildung und auch eine gute Chance in die Computer-Welt von Apple einzutauchen.

Als Leitfrage, welche mich durch das halbe Jahr begleiten sollte, wählte ich: «Ist es möglich in ca. 4 Monaten eine funktionierende App zu erschaffen und im AppStore zu veröffentlichen?». Diese Frage stellte sich mir immer wieder. Es konnte zu keinem Zeitpunkt gesagt werden, ob ich mein Ziel erreichen werde.

In dieser Arbeit, versuche ich zu verdeutlichen, welchen Aufwand eine App mit sich bringt und möchte dazu bewegen, bevor man über eine abgestürzte oder fehlerhafte Applikation schimpft, einfach mal an die Arbeit und das Engagement des Entwicklers dahinter zu denken.

Ausserdem sollte sie auch als kleine Hilfestellung dienen, für solche, die mit der Applikationsentwicklung anfangen wollen oder ebenfalls ein Projekt in diese Richtung machen wollen.



2. Hauptteil

2.1 Idee

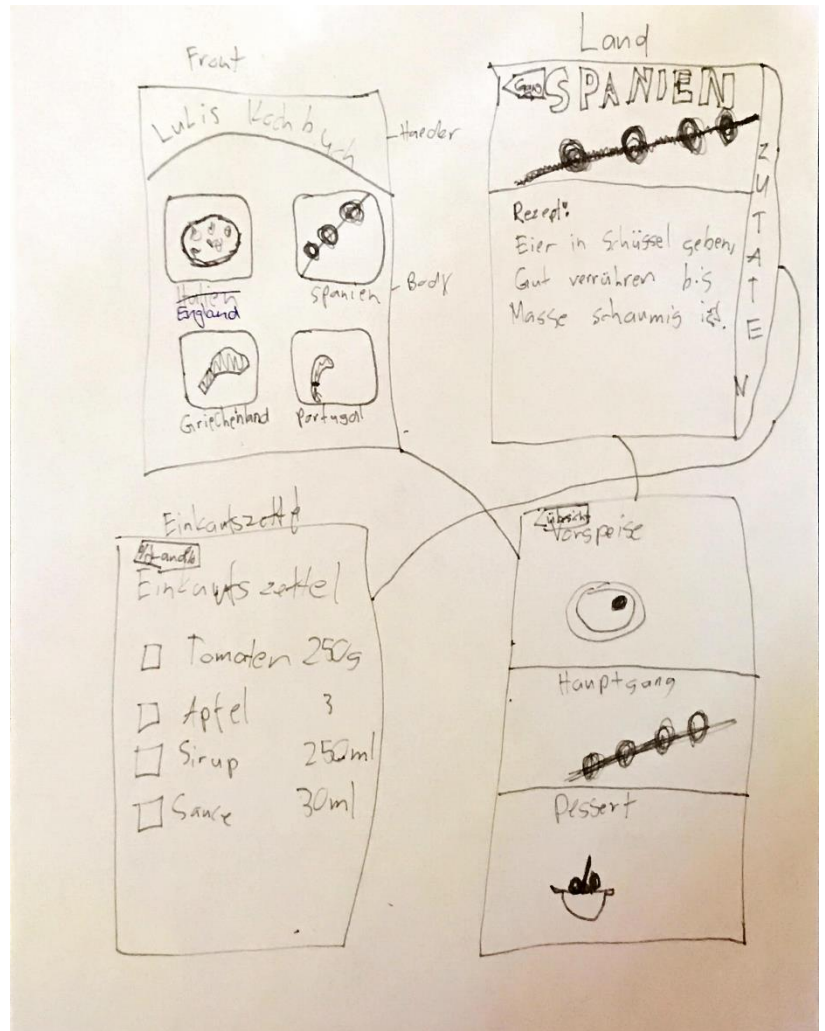
Für jedes Projekt braucht man anfangs eine gute Idee. Mir war es wichtig, nicht einfach nur eine Projektarbeit zu schreiben, sondern auch etwas zu erschaffen, dass man brauchen konnte. Durch mein grosses Interesse an der Informatik kam mir schnell die Idee eine Applikation zu erstellen. Jedoch gefiel mir der Gedanke nicht, wie einige Schüler vor mir, ein faciles Quiz zu kreieren, mit dem ein paar Fragen beantwortet werden können um sich anschliessend über das angezeigte Resultat zu freuen.

Als ich mich bei Kollegen erkundigte, welche Pläne sie schmiedeten, hörte ich, dass jemand ein eigenes Kochbuch kreieren wollte. Mir gefiel der Gedanke, eine «Kochbuch-App» zu programmieren, um so den Alltag von Menschen leichter zu machen und ihnen

die lästige Rezeptsuche, welche mir persönlich jedes Mal die Lust zu kochen nimmt, zu ersparen. Schnell kamen mir Ideen zur visuellen Umsetzung meines Planes (siehe Bild).

Die grundlegendste aller Fragen, nämlich für welches Betriebssystem ich diese programmieren möchte, war für mich schnell geklärt. Die Wahl fiel auf das Apple's iOS. Davon abgesehen, dass die Schweiz die höchste Anzahl iPhones pro Kopf aufweist, war ausserdem entscheidend, dass Apple einem sehr viele Möglichkeiten gibt, die Applikation zu testen und zu optimieren.

Schnell zeigte sich jedoch, dass die zum Programmieren benötigte Entwicklerumgebung (z.B. Xcode; ein Programm welches einem schon viele vorgefertigte Bausteine zur Verfügung stellt) nur auf einem Mac OS installiert werden kann. Dies stellte mich als Windowsnutzer, vor eine der grössten Herausforderungen, welche ich wahrscheinlich in dieser Arbeit meistern musste.

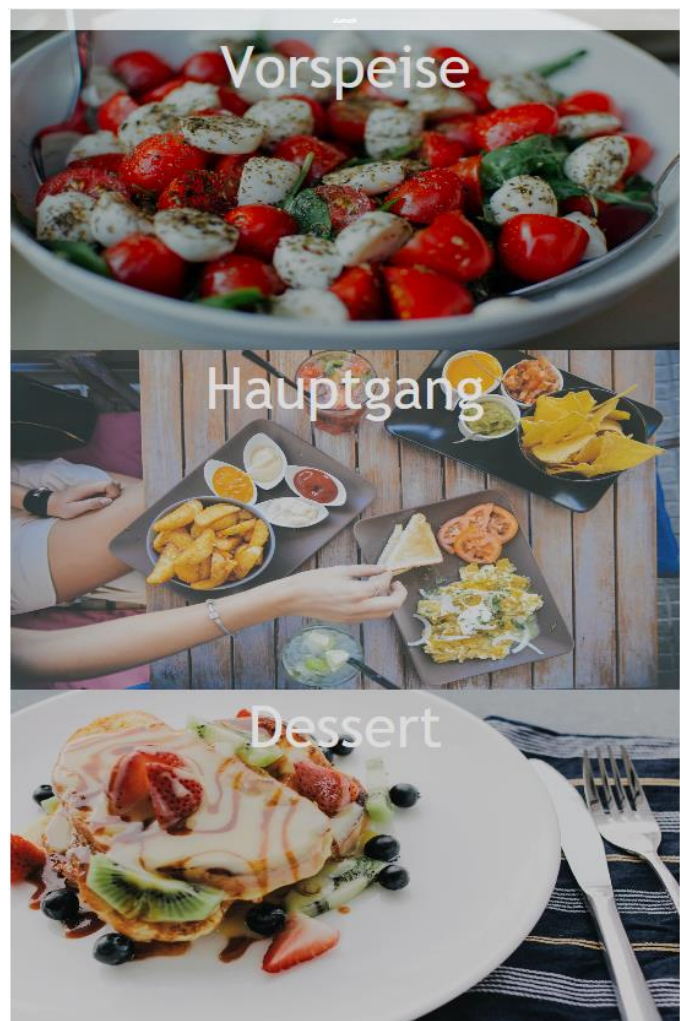


Erste Skizze wie meine App aussehen sollte

Der Wechsel auf ein grössten Teils unbekanntes Betriebssystem war schwieriger als gewohnt. Auch das wohl berühmteste Problem stellte sich mir schnell. Wo ist die rechte Maustaste? Dank Google schuf ich dieses Problem so schnell aus der Welt, wie es sich mir in den Weg stellte. Durch die Begeisterung eine App zu programmieren, waren Probleme wie diese, keine allzu grossen Hürden.

2.2 Vorbereitung

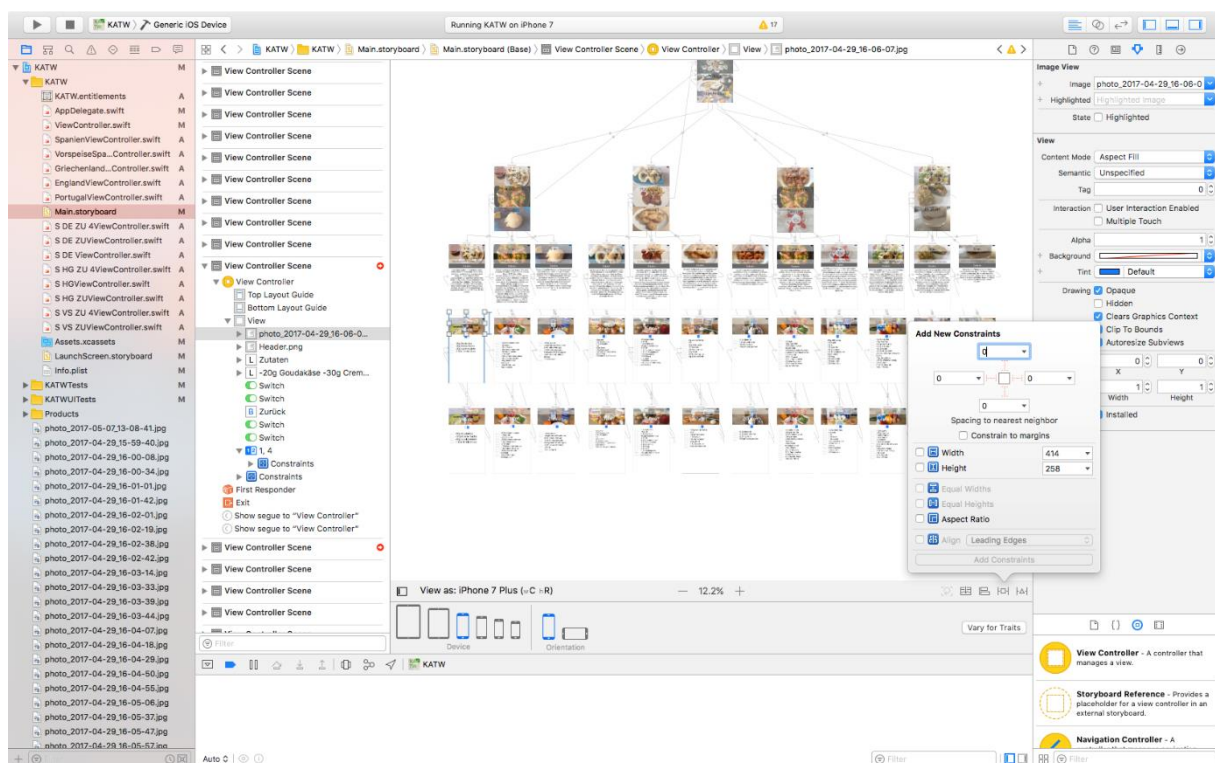
Um eine iOS- App zu programmieren benötigt es, wie beim Kochen, einige Vorbereitungen. Zum einen musste ich ein MacBook, einen iMac oder sonst ein Mac OS fähiges Gerät auftreiben. Nach langer Suche bot mir mein Onkel seinen Mac mini leihweise an. Danach musste ich mir Wissen über die Programmierung von Swift aneignen. Ausserdem benötigte ich eine Mitgliedschaft beim Apple Developer Programm (eine eigens von Apple entwickelte Umgebung für App-Entwickler sowie App-Designer) um meine App im App Store zu veröffentlichen oder mit TestFlight (Erklärung und mehr dazu, kommt dann später) zu testen. Um herauszufinden wie die App optisch aussehen sollte, erstellte ich eine Website als optischer Entwurf (siehe Bild) um allfällige Fehler frühzeitig zu erkennen. Damit ich schlussendlich starten konnte, musste ich die gesamte Entwicklerumgebung von Apple installieren. Nachdem ich alles erfolgreich eingerichtet habe, begann ich endlich mit dem Programmieren.



Optischer Entwurf (Website)

2.3 Programmierung

Swift ist eine von Apple entwickelte Programmiersprache, welche vor allem für die Einfachheit und die vielen Hilfsmöglichkeiten, wie das Developer-Forum (Entwickler-Forum), bekannt ist. Um diese zu erlernen lud ich mir «Swift Playgrounds» aus dem App Store auf mein iPad herunter. Diese App gibt einem die Möglichkeit, Swift auf spielerische Weise zu erlernen, in dem man einem niedlichen kleinen Männchen hilft, Juwelen zu sammeln. Um dies zu tun musste man verschiedene Befehle eingeben wie z.B. «moveForward()» um nach vorne zu gehen. Auf der Entwicklerumgebung Xcode (siehe Bild) begann ich langsam mich mit dem Ganzen vertraut zu machen. Eine grosse Herausforderung war es, all die ganzen Kleinigkeiten, die Mac OS zu Windows unterscheiden, zu entdecken und zu verstehen. Da es zu komplex und zu kompliziert wäre, auf die genaue Programmierung einzugehen, verzichte ich an dieser Stelle darauf und versuche stattdessen, mit einem kleinen Beispiel einen Teil der Programmierung verständlich zu erklären. Um das nachfolgende Beispiel zu verstehen, ist es erforderlich, die KATW-App schon einmal gesehen zu haben.

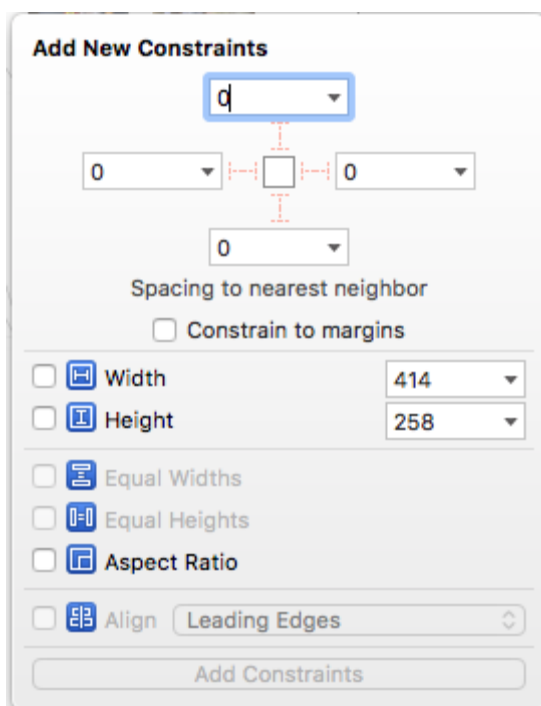


Screenshot der Entwicklerumgebung

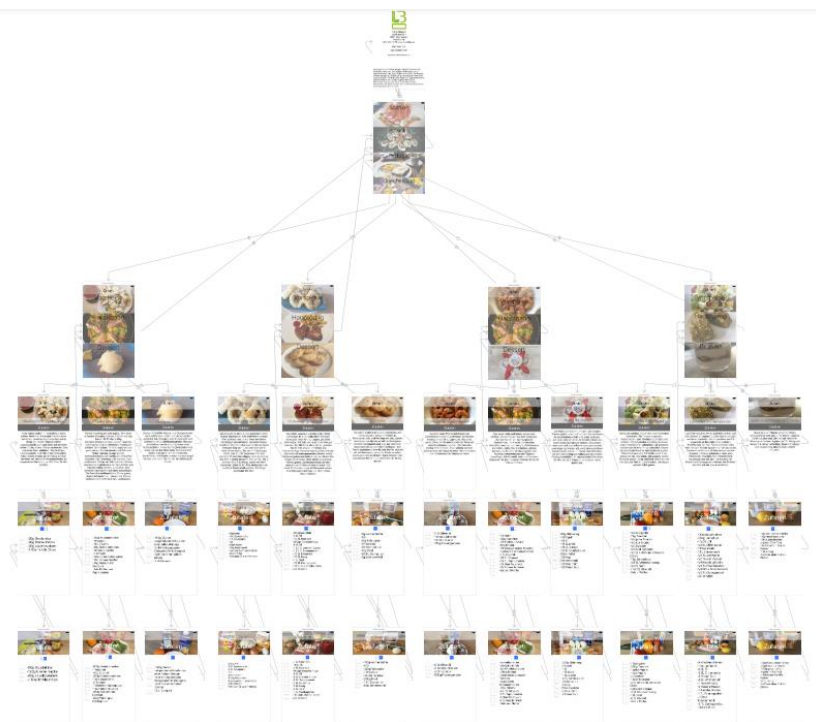


2.4 Beispiel

Die ganze App besteht aus verschiedenen Seiten, welche View Controller genannt werden (siehe Bild). Die KATW-App besteht aus ganzen 42 View Controllern, was überdurchschnittlich viel für eine App ist. Man könnte es mit einer Webseite vergleichen, welche ja auch verschiedene Seiten und Unterseiten hat, wie z.B. «Kontakt» und «Über uns». Diese Seiten bestehen dann auch wieder aus verschiedenen «Bausteinen» wie einem Bild, dem Text, dem «Zurück»-Button. Wenn man die App nun auf dem iPhone betrachtet, will man, dass es möglichst schön aussieht und nichts verzogen ist. Deshalb definiert man die genaue Position dieser Bausteine mit so genannten «Constraints». Man sagt z.B., dass der Abstand zum rechten und linken Rand 0 ist, damit das definierte Objekt genau bis zum Rand geht (siehe Bild) und nicht zu kurz ist oder allenfalls überlappt und nicht mehr sichtbar ist. Dies klingt zwar recht überflüssig, ist jedoch von grosser Bedeutung, da fast jede neue iPhone-Generation eine neue Displaygrösse hat und es ansonsten beim Drehen des Gerätes ein riesiges Durcheinander aller Bausteine geben würde.



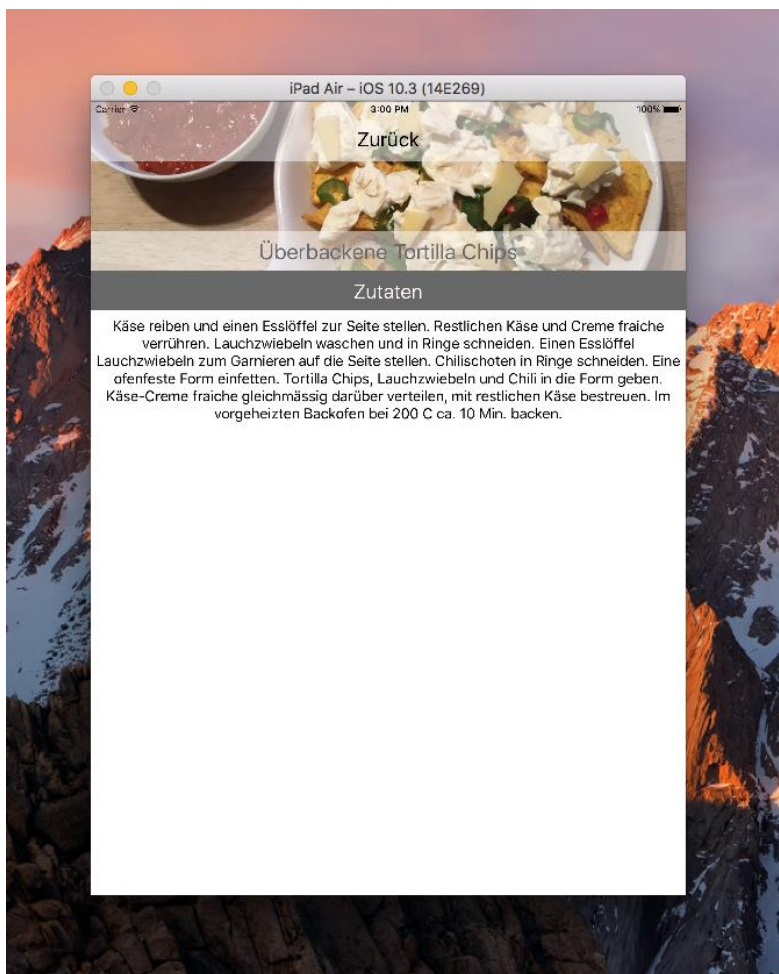
Definition der Position eines Objektes



Übersicht aller View Controller

2.5 Testphase

Nach jeder neu programmierten Linie, wollte man natürlich sofort wissen, welche Auswirkungen diese hatte. Deshalb konnte man die App mit einem Simulator testen. Der Simulator war eigentlich ein iPhone oder iPad, welches nur in der Entwicklungsumgebung existierte, jedoch genau mit einem echten Gerät übereinstimmte. Somit konnte man Fehler und nicht Gewolltes wieder ändern, ohne extra die ganze Applikation auf das iPhone zu laden. Als die App langsam Gestalt annahm, installierte ich sie immer wieder auf mein iPhone und iPad um diese genau zu überprüfen und auszuprobieren. Um die Meinung von anderen Testern zu erfahren, hatte ich mit TestFlight von Apple eine einfache Möglichkeit dies zu tun. Ich konnte meine App ganz einfach auf iTunes Connect (Plattform um Applikationen zu veröffentlichen und anderen Testern zugänglich zu machen) hochladen. Die von mir ausgewählten iPhone Besitzer einladen, welche die App TestFlight herunterladen mussten und einen per E-Mail erhaltenen Code eingeben konnten, um KATW als Beta (Kennzeichnung einer noch nicht fertiggestellten App) herunterzuladen. Sobald die Tester einen Fehler gefunden haben, teilten sie mir den mit, damit ich ihn in der nächsten Version beheben konnte.



KATW auf einem iPad Air-Simulator

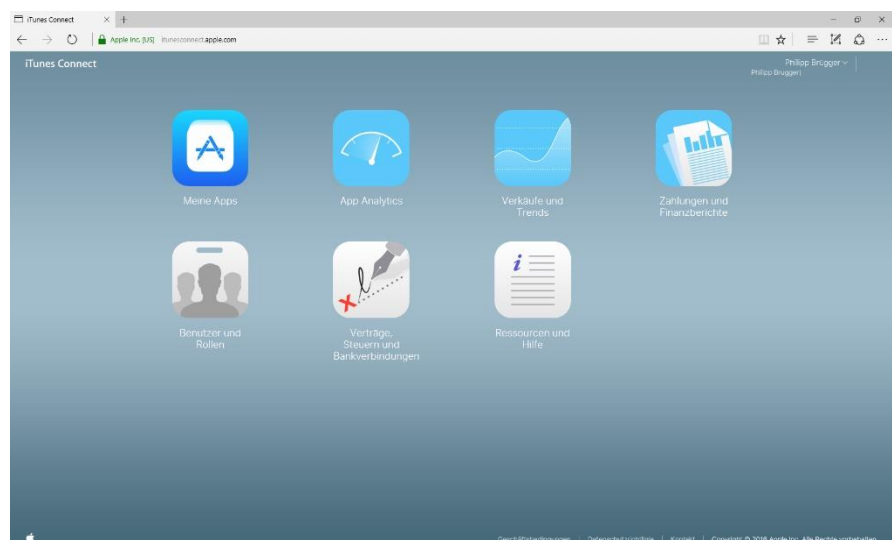


KATW auf einem iPhone 7-Simulator

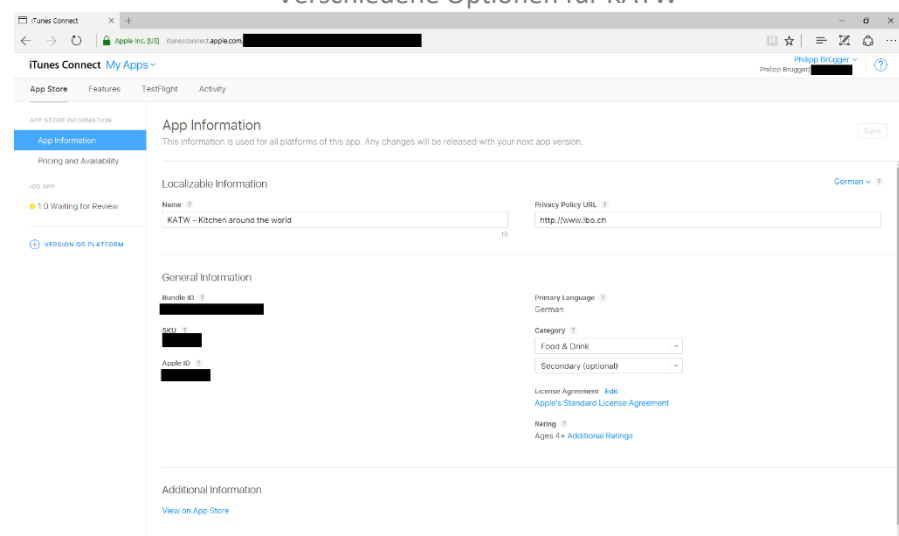
2.6 Publikation

Da ich die Applikation nicht nur für meine Tester und mich programmiert habe, war es für mich ein Muss, diese im Apple App Store zu veröffentlichen. Um dies zu tun, musste ich meine App erneut auf iTunes Connect hochladen, den Kaufpreis angeben, den Namen, die Kategorie, die Sprache, in welchen Ländern sie verfügbar sein sollte, die Kontaktdaten sowie auch die Website angeben. Danach füllte ich einen Fragebogen aus um das Mindestalter des Benutzers meiner App herauszufinden. Wichtig für meine App war auch das Icon (wie das Logo einer Firma). Ich gestaltete dies selbst mit verschiedensten Bildbearbeitungsprogrammen. Ausserdem war eine gute Beschreibung und interessante Screenshots wichtig. Die Bildschirmfotos mussten genau auf die strengen Richtlinien von Apple angepasst sein, wie z.B. die Grösse, welche ich extra noch mit Photoshop auf den Pixel genau anpassen musste. Nachdem ich all diese Arbeiten erledigt hatte, konnte ich endlich den Submit for Review-Button (Apple überprüft jede einzelne App) drücken. Diese Freude hielt jedoch nicht lange, da bald die Meldung «Rejected» auf meinem Bildschirm auftauchte. Dies bedeutete leider, dass Apple einen Fehler in

meinem Code entdeckt hatte und ich ihn nun beheben musste. Das Problem war, dass die App auf dem iPad nicht richtig angezeigt wird, was daran lag, dass die Definition der Objekte nur für das iPhone gedacht war, jedoch auch für das iPad benötigt wurde. Nun musste ich bei allen meinen 42 Seiten noch einmal jedes einzelne Objekt neu definieren. Weshalb es sich, wie beim Beispiel erwähnt, zeigt, dass Definition sehr wichtig ist.



Verschiedene Optionen für KATW



Publikationseinstellungen

2.7 Kompatibilität

Meine App habe ich für alle iOS Versionen ab 10.2 entwickelt, also ab dem iPhone 5. Jedoch sollte das Gerät schon eine 64- Bit Architektur haben, um die problemlose Nutzung sicher zu stellen. Hier eine Liste mit allen kompatiblen Geräten:

Gerät	iPhone 7 (Plus)	iPhone SE	iPhone 6S (Plus)	iPhone 6 (Plus)	iPhone 5s	iPhone 5c	iPhone 5
iOS	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2
64 Bit	Ja	Ja	Ja	Ja	Ja	Nein	Nein

Gerät	iPod Touch 6G	iPad 2017	iPad Air 2	iPad Air	iPad 4th gen	iPad mini 4	iPad mini 3
iOS	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2
64 Bit	Ja	-	Nein (höher)	Ja	Nein (höher)	Ja	Ja

Gerät	iPad mini 2	iPad Pro (10.5- und 12.9- inch 2nd gen)	iPad Pro (9.7- und 12.9- inch)	iPad Air	iPad 4th gen	iPad mini 4	iPad mini 3
iOS	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2	Ab 10.2
64 Bit	Ja	Ja	Ja	Ja	Nein	Ja	Ja

3. Reflexion

Das Ergebnis meiner Projektarbeit ist eine funktionierende App. Diese kann sich jeder aus dem AppStore auf sein iPhone herunterladen.

Anfangs lernte ich auf meinem iPad mini Swift zu programmieren. Nach der Erstellung einer Website, mit der ich mir einen Überblick verschuf, fing ich an meine App zu programmieren und immer weiter zu verbessern bis ich mit dem Endergebnis zufrieden war. Ich stellte den Kontakt zu Firma Apple Inc. her, damit ich meine App testen konnte und dann später für alle öffentlich zugänglich im AppStore veröffentlichen konnte.

In der Testphase bekam ich immer wieder Rückmeldungen von Testern meiner App.

Es war eine sehr grosse Hilfe das ich mich nicht allzu gross um den Inhalt kümmern musste, da ich die gesamten Rezepte, sowie die Zutatenlisten schon fertig von meiner Mitschülerin Luana Gantenbein bekam.

Leider hatte ich nicht viele Ansprechpersonen bei Problemen, welche direkt die Programmierung betrafen, da in meinem Umfeld noch niemand eine iOS- App programmiert hat. Dies war eine ganz besondere Herausforderung, selber Probleme zu lösen und nicht nachfragen zu können wie man dies nun genau machen muss. Das gewohnte: «Chönd sie mol schnell cho helfe cho?» verstummte nach den ersten Malen schnell, da die Antwort stets: «Do musch öper andersch fröge, i chum do nöd druss» war.

Mit dem Entwickeln dieser App ist mir, finde ich zumindest, etwas gelungen, von dem ich vor einem Jahr noch nicht zu träumen gewagt hätte. Die Programmiersprache «Swift» aus eigener Kraft zu erlernen und sich Zugang zu einem Mac zu verschaffen, sowie auch die viele Zeit zu investieren, brauchte Ausdauer und viel Motivation, welche ich vor allem durch den Gedanken erlangte, die eigene App aus dem Apple AppStore herunter laden zu können. Mein jetziges Wissen musste ich mir mühsam aus dem Internet, Videos und weiteren Quellen zusammensuchen und immer aufpassen, damit ich keine falschen Informationen aufschnappte und so einen Fehler in meine App einbaute.

Die wichtigste Person, welche die ganze Projektarbeit überhaupt möglich gemacht hat, ist mein Onkel Paul Brügger, welcher mir seinen Mac mini zur Verfügung stellte, damit ich dieses «Wunder» vollbringen konnte. Ausserdem half mir mein Bruder Adrian Brügger bei der Definition der verschiedenen Objekte. Wichtig war für mich meine ganze Familie, welche mich trotz einigen Rückschlägen immer wieder neu zu motivieren wusste und mir viele nützliche Tipps gab.

Danke an Apple, den grössten Smartphonehersteller. Ohne das von ihnen entwickelte iPhone und dazu gehörige iOS und auch den riesigen Hilfen wie iTunes Connect, Xcode usw. wäre dieses Projekt nicht möglich gewesen.

Meine Fragestellung, ob die Entwicklung und Veröffentlichung einer App in nur 4 Monaten möglich sei, hat sich für mich klar beantwortet. Es ist möglich in dieser kurzen Zeit, ohne jegliche Vorkenntnisse in der Applikationsentwicklung, eine App für Apple's Betriebssystem zu programmieren.

Dank dieser Arbeit weiss ich nun, was es für eine grosse Anstrengung es ist, trotz guten Kenntnissen in der Informatik, eine Kochbuch- App zu programmieren.

Interessante Fakten

Nach der langen Entwicklungszeit wurde die neue Version iOS 11 präsentiert. Das heisst, dass die App ein weiteres Mal überarbeitet werden muss, damit sie dann auch auf dem neuesten Betriebssystem läuft.

Durch mein Apple Developer Program Membership habe ich nun Zugang zu den noch nicht veröffentlichten Beta-Versionen wie z.B. von iOS, Mac OS, tvOS usw., um meine App schon vor dem Release einer neuen iOS Version, auf dieser zu testen.

Die ganze App besteht aus 42 View Controllern.

Ich habe rund 60 Stunden an der ganzen Projektarbeit gearbeitet.

Die ganze App besteht aus ca. 550 Objekten.

Dies sind etwa 2'200 Constraints, welche ich definieren musste.

Quellenverzeichnis

Mac mini:

Paul Brügger

Playgrounds:

Apple Inc. (<https://www.apple.com/swift/playgrounds/>)

Xcode:

Apple Inc. (<https://developer.apple.com/xcode/>)

TestFlight:

Apple Inc. (<https://developer.apple.com/testflight/>)

Problemlösungen:

YouTube (<http://www.youtube.ch>)

Rezepte, Bilder und Zutatenliste:

Luana Gantenbei

Entwickler Mitgliedschaft:

Apple Developer Program Membership (<https://developer.apple.com/programs/>)

109.00 Fr. jährlich



Zu meiner App



Zu meiner Website