

BACHELORARBEIT

Analyse der Auswirkung von Progressive Web Apps auf bestehende Web Apps

durchgeführt am
Studiengang Informationstechnik & System-Management
an der
Fachhochschule Salzburg GmbH

vorgelegt von
Refik Kerimi



Studiengangsleiter: FH-Prof. DI Dr. Gerhard Jöchl
Betreuer: DI Norbert Egger BSc

Salzburg, September 2018

Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich die vorliegende Bachelorarbeit ohne unzulässige fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und alle aus ungedruckten Quellen, gedruckter Literatur oder aus dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte gemäß den Richtlinien wissenschaftlicher Arbeiten zitiert, bzw. mit genauer Quellenangabe kenntlich gemacht habe. Diese Arbeit wurde in gleicher oder ähnlicher Form weder im In- noch im Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt und stimmt mit der durch die Begutachter beurteilten Arbeit überein.

Salzburg, am 1.09.2018



Refik Kerimi

1410555043

Matrikelnummer

Allgemeine Informationen

Vor- und Zuname:	Refik Kerimi
Institution:	Fachhochschule Salzburg GmbH
Studiengang:	Informationstechnik & System-Management
Titel der Bachelorarbeit:	Analyse der Auswirkung von Progressive Web Apps auf bestehende Web Apps
Schlagwörter:	PWA, Manifest, Service Workers, Push Notification, Cach API
Betreuer an der FH:	DI Norbert Egger BSc

Kurzfassung

Abstract

Danksagung

Danken möchte ich vor allem meinem Betreuer für die Unterstützung bei dieser Bachelorarbeit.

Besonderer Dank gilt auch meiner Familie und Freunden, die uns während des Studiums in allen Belangen immer unterstützt haben.

Inhaltsverzeichnis

Abkürzungsverzeichnis	i
Abbildungsverzeichnis	ii
Tabellenverzeichnis	iii
Listingverzeichnis	iv
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
2 Grundlagen	3
2.1 Geschichte Applikationsentwicklung	3
2.2 Mobile APP	4
2.2.1 Native Apps	4
2.2.2 Webapplikationen	4
2.2.3 Progressive Web Apps	4
2.2.4 Unterschiede zwischen Web Apps PWA und Native Apps	4
3 Basistechnologien	5
3.1 PWA vs.Native	5
3.1.1 Vorteile/ Nachteile	5
3.2 Aufbau PWA	5
3.2.1 Web App Manifest	5
3.2.2 Service Worker	5
3.2.3 Push Notifikation	5
3.2.4 Cache API	5
3.2.5 Geolocation	5
3.2.6 Camera API	5
3.2.7 Browser	5
3.2.8 Datenbank	5
4 Entwurf	6
4.1 Model Native	6
4.2 Model PWA	6

4.3	Anforderungsanalyse	6
5	Implementierung	7
5.1	Umsetzung der Anforderungen	7
5.2	Ausgewählte Programmiersprache und IDE	7
5.3	Manifest	7
5.3.1	Aufbau	7
5.3.2	Implementierung	7
5.4	Aufbau Service Worker	7
5.4.1	Aufbau	7
5.4.2	Implementierung	7
5.5	Zugriff Cache API	7
5.5.1	Aufbau	7
5.5.2	Implementierung	7
5.6	Zugriff Device API	7
5.6.1	Aufbau	7
5.6.2	Implementierung	7
5.7	Fetch API	7
5.7.1	Aufbau	7
5.7.2	Implementierung	7
5.8	IndexedDB	7
5.8.1	Aufbau	7
5.8.2	Implementierung	7
6	Funktionstest/Validierung	8
6.1	Ausgangsbedingung und Ausgrenzung	8
6.2	Komponententest	8
6.2.1	Add to Homescreen	8
6.2.2	Funktion Service Worker	8
6.2.3	Cache	8
6.2.4	Kamera	8
6.2.5	Geolocation	8
7	Fazit	9

Abkürzungsverzeichnis

PWA	Progressive Web Application
SP	Smart Phone
JS	Java Script
SHP	Smart Home Prototypen

Abbildungsverzeichnis

2.1 Bildtext [?] 3

Tabellenverzeichnis

Listings

1 Einleitung

Durch die Markteinführung des Smart Phones(SP) hat sich unser Leben gravierend geändert. Nicht nur unsere Kommunikation, sondern unser Leben im Allgemeinen, ist durch dieses kleine Wundergerät erleichtert worden. Wir haben ständig das SP im Einsatz, zum Organisieren, zum Spielen, zum Musik hören, um unsere Kontakte zu pflegen und ab und zu wird es auch zum Telefonieren verwendet. Das SP hat nicht nur unseren Alltag beeinflusst, sondern auch das Internet und die Entwicklung von Webapplikationen. Kurz nach der Erfindung des smarten Handys kam ein weiterer Markt hinzu, der sich parallel dazu entwickelt hat. Es wurden neue Berufe gegründet wie z.B.: der Native App Entwickler. Native Apps werden speziell an das Betriebssystem angepasst und können somit im Gegensatz zu einer Standard Web Applikation die Ressourcen eines Mobilen Gerätes optimal nutzen. Das Ganze benötigt natürlich eigene Entwickler die sich auf die jeweiligen Plattformen spezialisieren. Dies führt zu höheren Entwicklungskosten, wenn man das Produkt auf verschiedenen Plattformen betreiben will. In den letzten Jahren wurde auch, durch die immer besseren werdenden Browser, die Webapplikation stetig weiter verbessert und durch Technologien wie den Progressive Web Apps sind diese heute schon in der Lage mit den Native Apps zu konkurrieren.

1.1 Motivation

Wie im vorigen Kapitel beschrieben werden native Applikationen für ein bestimmtes Betriebssystem optimiert. Diese haben dann den Vorteil die Hardware des Gerätes nutzen zu können. Dadurch sind komplexe Anwendungen realisierbar. Doch diese sind relativ kostspielig und auf Grund der Vielzahl der diversen Apps in den Appstores nicht gerade lukrativ. Durchschnittlich werden (genaue Prozentzahl ermitteln) monatlich pro Nutzer heruntergeladen und benötigen viel Speicherplatz auf dem Gerät. Die Progressive Web Application(PWA) vereint die Vorteile von native App und von Webanwendungen und gibt dem Nutzer ein Gefühl, dass man es mit einer auf das System angepassten Anwendung zu tun hat.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist es, einen Smart Home Prototypen(SHP) zur demonstration zu entwickeln. Dem Prototypen werden die PWA typischen Features wie das hinzufügen auf dem Startbildschirm, Offline arbeiten, die Pushfunktionen, das Zugreifen auf Gerätefunktionen und das cachen über eine Clientseitig integrierte Datenbank, hinzugefügt. An diesen Features sollen die Vorteile, Nachteile, Entwicklung, Betrieb und User Experience betrachtet werden. Basictechnologien und verwendete Frameworks (z.B.: Java Script(JS),ReactJS, NodeJS, Yarn,...) werden in dieser Arbeit nicht behandelt.

2 Grundlagen

Wie in Kapitel 1 beschrieben, hat der stetige Zuwachs von PWAs zum Umdenken bei der Planung und beim Entwickeln von Webapplikationen geführt. Zu Beginn jedes Projektes steht die Entscheidung an, welche Technologien und Tools zur Entwicklung verwendet sollen, um die bestmögliche Ergebnis zu erhalten. Wenn die falschen Methoden gewählt werden, kann das zu gravierenden Fehlern in der Applikation führen, die sich erst mit Fortdauer der produktiven Verwendung ersichtlich werden. Entscheidet man sich für eine Anwendung, die auf das Betriebssystem zugeschnitten ist oder doch für eine plattformübergreifende Webanwendung. Beide haben Vorteile und Nachteile und diese werden im Zuge dieser Arbeit betrachtet. Der Kern der Arbeit stellt, aber die von Google entwickelten (1) Progressive Web Apps da. Die PWA() sollen den Spagat zwischen diesen beiden Anwendungen zu schaffen oder kann diese neue Form der Appentwicklung die traditionellen Technologien gar zur Gänze ablösen?



Abbildung 2.1: Bildtext [?]

2.1 Geschichte Applikationsentwicklung

-
-
-

2.2 Mobile APP

2.2.1 Native Apps

2.2.2 Webapplikationen

2.2.3 Progressive Web Apps

2.2.4 Unterschiede zwischen Web Apps PWA und Native Apps

3 Basistechnologien

3.1 PWA vs.Native

In diesem Kapitel werden die Besonderheiten einer PWA erklärt.

3.1.1 Vorteile/ Nachteile

3.2 Aufbau PWA

3.2.1 Web App Manifest

3.2.2 Service Worker

3.2.3 Push Notifikation

3.2.4 Cache API

3.2.5 Geolocation

3.2.6 Camera API

3.2.7 Browser

3.2.8 Datenbank

4 Entwurf

In diesem Kapitel wird das Muster im Allgemeinen und die Anforderungen bzw. die Umsetzung der Progressive Web Applications betrachtet.

4.1 Model Native

4.2 Model PWA

4.3 Anforderungsanalyse

5 Implementierung

5.1 Umsetzung der Anforderungen

5.2 Ausgewählte Programmiersprache und IDE

5.3 Manifest

5.3.1 Aufbau

5.3.2 Implementierung

5.4 Aufbau Service Worker

5.4.1 Aufbau

5.4.2 Implementierung

5.5 Zugriff Cache API

5.5.1 Aufbau

5.5.2 Implementierung

5.6 Zugriff Device API

5.6.1 Aufbau

5.6.2 Implementierung

5.7 Fetch API

5.7.1 Aufbau

5.7.2 Implementierung

5.8 IndexedDB

5.8.1 Aufbau

5.8.2 Implementierung

6 Funktionstest/Validierung

6.1 Ausgangsbedingung und Ausgrenzung

6.2 Komponententest

6.2.1 Add to Homescreen

6.2.2 Funktion Service Worker

6.2.3 Cache

6.2.4 Kamera

6.2.5 Geolocation

7 Fazit