

Entwicklung einer Einkaufslisten-App

Markus Hülß

Email: huma1200@stud.hs-coburg.de

Tanja Mögn

Email: mota1200@stud.hs-coburg.de

Malcolm Kögler

Email: koma1200@stud.hs-coburg.de

Daniel Müller

Email: muda1200@stud.hs-coburg.de

Abstract—Heutzutage hält die Nutzung von Smartphones immer mehr Einzug in den Alltag eines Menschen. Aus diesem Gedanken wuchs die Idee eine Einkaufslisten App zu entwickeln, die selbstgeschriebene Zettel berflüssigt macht. Wichtige Anforderungen sind Plattformunabhängigkeit und die Möglichkeit Listen mit mehreren Nutzern zu teilen. Diese Arbeit zeigt den Ablauf des Projekts auf und erläutert die Verwendeten Technologien.

I. EINLEITUNG

Fast jeder Mobiltelefon Nutzer hat heutzutage ein Smartphone. Immer präsent und nicht mehr aus dem Alltag wegzu-denken, unterstützt es uns heute Tag für Tag in vielerlei Hinsicht. Ohne Chat oder Kalender geht es fast nicht mehr. Ein Blick in den App Store zeigt immer mehr Applikationen, die das Smartphone als Allrounder immer besser ausstattet. Vor einigen Jahren hat man vor dem Einkauf noch schnell einen Einkaufszettel geschrieben. Aber wie schnell war der Zettel doch in der Tasche spurlos verschwunden, obwohl man sich so sicher war, den Zettel eingepackt zu haben. Heutzutage bietet sich das Smartphone als Möglichkeit an, einen Einkaufszettel zu speichern. Das bietet unter anderem die Möglichkeit den Einkaufszettel zu jeder Zeit zu erweitern, oder ihn mit anderen zu teilen. Ziel dieser Arbeit ist es, eine Plattform übergreifende Einkaufszettel App zu entwickeln. Damit kann man seinen Einkaufszettel mit der Familie oder mit den WG-Mitbewohnern teilen. Entwickelt wird die App mit einer zentralen Datenhaltung auf einem Server. Das Frontend wird als Web App mittels Cordova realisiert. Das Projekt wurde mittels des Wasserfallmodells entwickelt. Zunächst wurden die Anforderungen erhoben. Anschließend hat sich die Projektgruppe in zwei Teams aufgeteilt. Ein Team war für die Frontend-Entwicklung zuständig, das andere für das Backend. Der Programmcode wurde mittels des Versionierungssystem GIT verwaltet. Grafiken wurden mit Balsamiq und MS Visio 2013 erstellt. Die Kommunikation erfolgte teils persönlich, teils über E-Mails. In der vorliegenden kurzen schriftlichen Ausarbeitung des Themas werden zunächst die Anforderungen näher beschrieben, welche die Grundlage für die Entwicklung bilden. Daraufaufgehend werden die verwendeten Technologien in Frontend und Backend im Zusammenhang mit dem Nutzen in der App vorgestellt.

Zuletzt werden in einer Zusammenfassung die Projekterfolge und Misserfolge aufgezeigt.

II. KONKURRENZANALYSE

Um sich einen Überblick über vorhandene Apps zu verschaffen, wurde eine Konkurrenzanalyse erstellt. Diese Konkurrenzanalyse beschränkt sich auf native Apps und lässt Cloud-basierte Lösungen außen vor. Zu diesem Zweck wurden die Top-Apps aus den App-Stores für die Plattformen Android, IOS und Windows Phone miteinander verglichen. Um die Apps miteinander vergleichen zu können, wurden drei Merkmale definiert. Diese Merkmale waren Plattformunabhängigkeit bzw. Multiplattformfähigkeit, die Möglichkeit Einkaufslisten mit anderen Leuten zu teilen sowie eine zentrale Datenhaltung mit Synchronisierung der Daten über mehrere Geräte hinweg.

Am Markt gibt es mittlerweile mehrere Apps die darauf spezialisiert sind Einkaufslisten zu verwalten. Nur eine App bietet eine native Implementierung für die oben genannten Plattformen. Einige Apps bieten eine reduzierte Multiplattformfähigkeit in dem Sinne, dass ein Webservice zu Verfügung gestellt wird über dem auf die Daten zugegriffen werden kann. Die meisten Apps bieten ein Backbone zur zentralen Datenhaltung. Nur wenige Apps bieten die Möglichkeit Einkaufslisten mit anderen zu teilen. Diese Fähigkeit ist teilweise darauf beschränkt eine Einkaufsliste per E-Mail zu verschicken.

Einzig allein eine App deckt alle drei definierten Merkmale ab. Für die Plattform Windows Phone ist keine App mit den definierten Merkmalen vorhanden. Bei Android und IOS ist die Situation etwas besser, allerdings ist auch hier Potenzial vorhanden.

III. MOCKUPS

Zu Beginn der Entwicklung wurde eine Möglichkeit gesucht schnell einen Prototypen der App zu entwickeln. Zu diesem Zweck wurden Mockups für das Frontend erstellt. Diese wurden in Teamarbeit entworfen und somit hatte jeder im Team eine Vorstellung wie die App später aussehen soll. In Abbildung 1 ist ein Beispiel für ein solches Mockup zu sehen. Die Abbildung 1 zeigt die Ansicht eines Einkaufszettels mit den zugehörigen Steuerelementen.

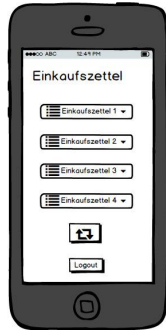


Abbildung 1: Einzelansicht

IV. FRONTEND

Ein wichtiges Ziel ist es eine plattform- und auflösungsunabhängige Darstellung zu ermöglichen. Dabei soll das Frontend nicht für jede Plattform separat entwickelt werden, um sowohl Zeitaufwand, als auch potentielle Fehlerquellen zu verringern. Die Verwendung des Browser dient als Grundlage. Somit ist eine einfache Darstellung auf mobilen Geräten und Desktop PC's möglich. Dieser Abschnitt befasst sich mit den beiden Komponenten "Apache Cordova" und "jQuery mobile", welche die Umsetzung der Anforderungen ermöglichen.

A. Apache Cordova

Die auf dem Markt vorhandenen Applikationen entwickeln für Android, iPhone und Windows Phone jeweils separat. Das erfordert tiefgehendes KnowHow für die einzelnen Plattformen und weitergehende separate Fehler und Change-Request Behandlung. Aus diesem Grund wird bei der Entwicklung der App auf Apache Cordova, im Weiteren nur noch Cordova genannt, gesetzt. Cordova ist ein Framework zum Erstellen von nativen Anwendungen für die gängigsten mobilen Plattformen durch die Verwendung von HTML5. Das Konzept von Cordova ist auf einer Seite das Frontend einer Anwendung über eine angezeigte Webseite im Browser zu realisieren. Auf der anderen Seite stellt es zusätzlich Javasript API's bereit die einem Zugriff auf bestimmte Hardware wie der Kamera oder bestimmte Sensoren ermöglichen. Auch kann auf Events wie eine schwache Funkverbindung reagiert werden. Somit sind Funktionalitäten nativer Apps gegeben, ohne sich intensiv mit der Architektur jeglicher Zielplattformen auseinanderzusetzen. Zusätzlich baut Cordova am Ende aus diesen Komponenten eine native App, welche sich in dem dementsprechenden Store anbieten lässt.

B. jQuery Mobile

Ein zusätzliches Problem neben den unterschiedlichen Plattformen, ist die Vielzahl an unterschiedlichen Browsern und Displayauflösungen der Endgeräte. Die Anwendung soll ein einheitliches Design, unabhängig von Browser und

Auflösung besitzen. Ein Benutzer soll sich nicht bei einem Wechseln zwischen einem normalen und einem mobilen Browser bei der Bedienung umgewöhnen müssen. Um dieser Anforderung nachzukommen wird auf das jQuery Mobile Framework zurückgegriffen. Es ermöglicht eine einfache Umsetzung von Webseiten mit einem responsive Design und bietet eine große Auswahl unterstützter Browser (sowohl mobile, als auch non-mobile). Somit wird mit einer einzigen Version des Frontends eine große Menge an Endgeräten unterstützt und trägt zu einer schnellen Entwicklung der Anwendung bei. Zusätzlich ist es mit dem Multi-Page-Layout Konzept möglich mit einer einzigen HTML-Seite alle Ansichten einer App darzustellen. Dadurch muss nicht bei jedem Wechsel einer Ansicht eine neue Seite heruntergeladen werden, womit sich die Performance verbessert und Traffic gespart wird. (Bisherige Quelle bisher nur die jQuery Mobile Website)

V. BACKEND

Ein weiteres wichtiges Ziel ist es eine endgeräteunabhängige Datenspeicherung zu gewährleisten. Dafür ist ein Backend das von überall über das Internet zu erreichen ist vonnöten. Für die Umsetzung des Projekts entschieden wir uns für einen einfachen Webserver und die Nutzung von PHP und MySQL zur Datenhaltung und für JSON nach dem ReST-Standart zur Kommunikation zwischen Anwendung und Backend. Dieser Abschnitt befasst sich mit den Komponenten PHP, MySQL und JSON, welche die Umsetzung der Anforderungen ermöglichen.

A. PHP

PHP wird im Backend verwendet um die Schnittstelle zwischen Anwendung und Datenbank auf dem Webserver zu stellen. PHP unterstützt unzählige Datenbanktypen und ein breites Spektrum an Funktionalitäten zur Konvertierung in das JSON Format.

B. MySQL

MySQL ist das weltweit verbreitetste relationale Datenbanksystem und kam vor allem wegen seiner plattformunabhängigkeit und der kostenfreien Nutzung unter der OpenSource-Lizenz zur Umsetzung der Datenbank in Frage. Bei der Auswahl spielte auch die Möglichkeit der Verwendung von verschiedenen Speichersubsystemen eine Rolle. Durch die Verwendung von InnoDB stehen Rollback und Transaktionssicherungsmöglichkeiten zur Verfügung, dies ist ein wichtiger Gesichtspunkt im Bezug auf Datensicherheit.

C. JSON

Aufgrund der Verwendung des javascriptbasierten JQuery-Mobile zur Erstellung einer Webapp lag die Verwendung von JSON als Format zum Datenaustausch zwischen Anwendung und Backend nahe. Dadurch ist es möglich die Daten vom Backend ohne vorherige Konvertierung in der Anwendung

weiter zu verwenden. Mithilfe von PHP lässt sich so die Schnittstelle nach dem ReST Programmierparadigma gestalten, die die geforderten Daten im JSON zurückliefert.

VI. ARCHITEKTUR

Damit die einzelnen Komponenten ordnungsgemäß so zusammenarbeiten um die Anforderungen zu erfüllen, benötigt es noch eine passende Architektur. Aufgrund der verwendeten Technologien wird eine Drei-Schichten-Architektur verwendet. Somit lässt sich eine Trennung von Präsentation und Anwendungslogik realisieren.

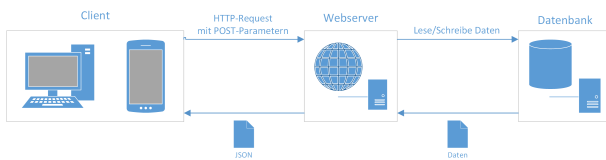


Abbildung 2: 3-Wege-Architektur für die App

Die Abbildung 2 zeigt den genauen Aufbau der Architektur. Ein Client sendet seine Aktion per HTTP-Request an den Webserver. Dieser holt geforderte Daten aus der Datenbank, oder ändert diese. Die von der Datenbank erhaltenen Daten werden in das JSON Format konvertiert. Danach werden sie als Inhalt des HTTP-Response an den Client gesendet. Der genauere Ablauf ist im folgenden Sequenzdiagramm ersichtlich.

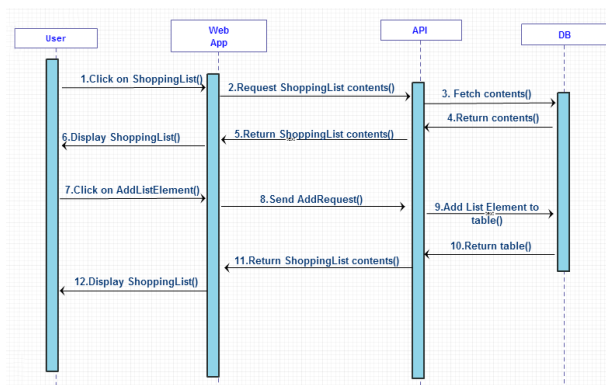


Abbildung 3: Sequenzdiagramm

VII. ZUSAMMENFASSUNG

Anmerkung Tanja: Hier ist vielleicht wichtig das wir reinschreiben, dass zwar Cordova und JQuery mobile ganz hübsch sind, es aber trotzdem noch nicht das wahre vom Ei ist. Außerdem sollten wir erwähnen wo es Probleme gab.

VIII. AUSBLICK (ANMERKUNG: VIELLEICHT NOCH ERWÄHNER INWIEFERN DIE OFFLINE BENUTZUNG REALISIERT WERDEN KANN, DA DAS GRUNDKONZEPT EINE PERSISTENTE DATENSPEICHERUNG IN EINER ZENTRALEN DATENBANK VORSIEHT.)

Die App HIER NAME EINFÜGEN besitzt bereits alle notwendigen Funktionen um eine Einkaufsliste zu erstellen, editieren und teilen. In der weiteren Entwicklung werden die Vorteile von Phonegap eingebaut. So wird beispielsweise ein Offline-Modus entwickelt, der es ermöglichen soll, die App auch ohne permanenten Internetverbindung zu nutzen. Das ist mit Phonegap möglich, da man mit diesem Framework die Qualität der Internetverbindung abfragen kann. Des weiteren ist eine intelligentere Unterstützung der App in den Alltag gedacht. Die App soll beispielsweise den User benachrichtigen, falls eine ein anderer User in einer gemeinsamen Einkaufsliste einen Eintrag geändert oder hinzugefügt hat. Hier ist wichtig das unbedingt der native Benachrichtigungsmodus des Systems verwendet wird um so viel Usability wie möglich zu gewährleisten. Ein weitere Verbesserung wäre auch die permanenten Lokalisierung des Users und eine Reaktion der Einkaufslisten-App, wenn sich der User in der Nähe oder in einem Einkaufsladen befindet. Hiermit hebt sich die HIER NAME EINFÜGEN von seinem Konkurrenten auf dem Markt ab.