Entwicklung einer Einkaufslisten-App

Markus Hülß
E-Mail: huma1200@stud.hs-coburg.de

Malcolm Kögler
E-Mail: koma1200@stud.hs-coburg.de

Tanja Mögn
E-Mail: mota1200@stud.hs-coburg.de

Daniel Müller
E-Mail: muda1200@stud.hs-coburg.de

***Abstract*—Mangels Verfügbarkeit von multiplattformfähigen Apps zur Verwaltung von Einkaufslisten entstand die Idee, eine solche App zu entwickeln. Um den Entwicklungsaufwand möglichst gering zu halten und nicht für jede Plattform separat eine App programmieren zu mssen, entschieden wir uns für das Framework Phonegap. Dieses Paper beschäftigt sich mit der Konzeption und Entwicklung der App.**

I. Einleitung

Immer präsent und nicht mehr aus dem Alltag wegzudenken, unterstützt uns das Smartphone schon heute bei der Organisation und Kommunikation. Vor einigen Jahren musste vor dem Einkauf noch schnell ein Einkaufszettel geschrieben werden. Aber wie schnell war der Zettel in der Tasche spurlos verschwunden. Da das Smartphone praktisch immer bei einem ist, bietet es sich als Möglichkeit an, einen Einkaufszettel zu speichern. Das schafft den Vorteil den Einkaufszettel jeder Zeit zu editieren oder ihn mit Freunden zu teilen. Ziel dieser Arbeit ist es, eine App zur Verwaltung von Einkaufszetteln zu entwickeln. Diese soll es ermöglichen, erstellte Einkaufszettel mit anderen zu teilen. Die jeweilige Plattform des App-Benutzers darf dabei kein Hindernis sein.

Die Projektgruppe wurde in zwei Teams aufgeteilt. Ein Team war für die Frontend-Entwicklung zuständig, das andere für das Backend. Der Programmcode wurde mittels des Versionierungssystems GIT verwaltet. Grafiken wurden mit Balsamiq und MS Visio 2013 erstellt. Die Kommunikation erfolgte teils persönlich, teils über E-Mails.

Zu Beginn wird die Konkurrenzanalyse beschrieben. Anschlieend wird die Notwendigkeit von Mockups erlutert. Darauf folgend werden die verwendeten Technologien in Frontend und Backend vorgestellt. Zuletzt werden in einer Zusammenfassung die Projekterfolge und Misserfolge aufgezeigt. Ein Ausblick zeigt die weiteren geplanten Entwicklungsschritte.

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. Der Vergleich erfolgte anhand der Kriterien Plattformunabhängigkeit bzw. Multiplattformfähigkeit, Multiuserfähigkeit der Einkaufslisten, sowie eine zentrale Datenhaltung mit Sychronisierung der Daten.

Am Markt gibt es mehrere Apps, die eine Alternative zum Papiereinkaufszettel darstellen. Bezüglich der Plattformunabhängigkeit wurde festgestellt, dass es derzeit nur eine App gibt, die eine native Implementierung für die oben genannten Plattformen bereitstellt. Andere Lösungen bieten eine reduzierte Multiplattformfähigkeit in dem Sinne, dass ein Webservice zu Verfügung gestellt wird. ¨ber diesen kann auf die Daten zugegriffen werden. Die zentrale Datenhaltung wird bei den meisten Apps durch ein Backbone gelöst. Multiuserfähigkeit wird von wenigen Apps implementiert. Diese ist teilweise darauf beschränkt, dass eine Einkaufsliste per E-Mail verschickt wird.

III. Mockups

Damit jedes Teammitglied eine Vorstellung davon hat, wie die App später aussehen soll, war ein Prototyp notwendig. Zu diesem Zweck wurden Mockups für das Frontend in Teamarbeit entworfen.

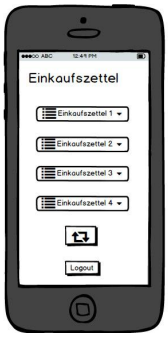

Abbildung 1: Einzelansicht

In Abbildung 1 ist ein Beispiel für ein Mockup zu sehen. Die Abbildung 1 zeigt die Ansicht eines Einkaufszettels mit den zugehörigen Steuerelementen.

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. Dies soll sowohl den Zeitaufwand als auch potenzielle Fehlerquellen verringern. Der Browser dient als Grundlage für die Darstellung. Somit ist eine Darstellung auf mobilen Geräten und Desktop-PCs möglich. Dieser Abschnitt befasst sich mit den beiden Komponenten "Phonegap" und "jQuery mobile".

A. Phonegap

Die auf dem Markt vorhandenen Applikationen entwickeln für die Plattformen Android, iPhone und Windows Phone jeweils separate native Apps. Das erfordert tiefgehendes Know-how für die einzelnen Plattformen. Zudem wird die Behandlung von Fehler- und Change-Requests aufwändig, da diese für jedes System separat entwickelt werden müssen. Aus diesem Grund wird bei der Entwicklung der App auf Phonegap gesetzt. Phonegap ist ein Framework zum Erstellen von nativen Anwendungen für die gängigsten mobilen Plattformen durch die Verwendung von HTML5. Das Konzept von Phonegap ist, zum einen das Frontend der Anwendung über eine angezeigte Webseite im Browser zu realiseren. Zum anderen stellt es zusätzlich Javasript API's bereit, die einem Zugriff auf bestimmte Hardware eines Smartphones, wie beispielsweise Kamera oder Sensoren ermöglichen. Wichtig ist, dass man damit auf plattformabhängige Events reagieren kann, zum Beispiel auf eine schwache Funkverbindung des Geräts. Somit sind Funktionalitäten nativer Apps gegeben, ohne sich intensiv mit der Architektur der Zielplattformen zu beschäftigen. Der Buildprozess von Phonegap erstellt aus den Komponenten eine native App, welche sich zugehörigen App Store verkaufen lässt.

B. jQuery Mobile

Ein weiters Problem ist, dass sich die App nicht durch die Vielzahl an unterschiedlichen Browsern und Displayauflösungen der Endgeräte beinflussen lässt. Die Anwendung soll ein einheitliches Design, unabhängig von Browser und Auflösung darstellen. Der Nutzer der Anwendung muss auf allen Browsern, mobil und Desktop, das gleiche Design sehen. Um diese Anforderung zu erfüllen, wird auf das jQuery Mobile Framework zurückgegriffen. Es ermöglicht eine einfache Umsetzung von Webseiten mit einem responsive Design. Dieses Frameworkt wird von einer großen Anzahl an unterschiedlichen Browsern unterstützt (sowohl mobile als auch non-mobile). Somit wird mit einer einzigen Version des Frontends eine große Menge an Endgeräten unterstützt. Dies trägt zu einer schnellen Entwicklung der Anwendung bei. Des weiteren 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. Damit verbessert sich die Performance und Traffic wird gespart. (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 Webserver. Bei der Datenhaltung wird PHP und MySQL genutzt. Die Kommunikation zwischen Frontend und Backend erfolgt ber JSON nach dem REST-Paradigma . Dieser Abschnitt befasst sich mit den Komponenten PHP, MySQL und JSON.

A. PHP

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

B. MySQL

MySQL ist ein weit verbreitetes relationales Datenbanksystem und kam vor allem wegen seiner Plattformunabhängigkeit und der kostenfreien Nutzung unter der OpenSource-Lizenz zur Umsetzung der Datenbank in Frage. Ein weiteres Kriterium, das fr MySQL sprach, war die Möglichkeit damit verschiedene Speicherssubsystemen zu nutzen. Durch die Verwendung von InnoDB stehen Rollback und Transaktionssicherungsmöglichkeiten zur Verfügung. Das ist ein wichtiger Gesichtspunkt im Bezug auf Datensicherheit.

C. JSON

Aufgrund der Verwendung des auf Javascript basierenden Frameworks JQuery-Mobile, lag die Verwendung von JSON nahe. Damit ist es knnen Daten vom Backend ohne spezielle Konvertierung in der Anwendung verwendet werden. Mithilfe von PHP lässt sich so die Schnittstelle nach dem REST-Programmierparadigma gestalten.

VI. Architektur

Damit die einzelnen Komponenten ordnungsgemäß zusammenarbeiten wird die passende Architektur benötigt. Aufgrund der verwendeten Technologien wird eine Drei-Schichten-Architektur verwendet. Somit lässt sich eine Trennung von Präsentation und Anwendungslogik realisieren.

Die Abbildung 2 zeigt den genauen Aufbau der Architektur. Ein Client sendet seine Aktion per HTTP-Request an den Webserver. Dieser holt die geforderten Daten aus der Datenbank oder ändert diese. Die von der Datenbank erhaltenen Daten werden in das JSON-Format konvertiert.

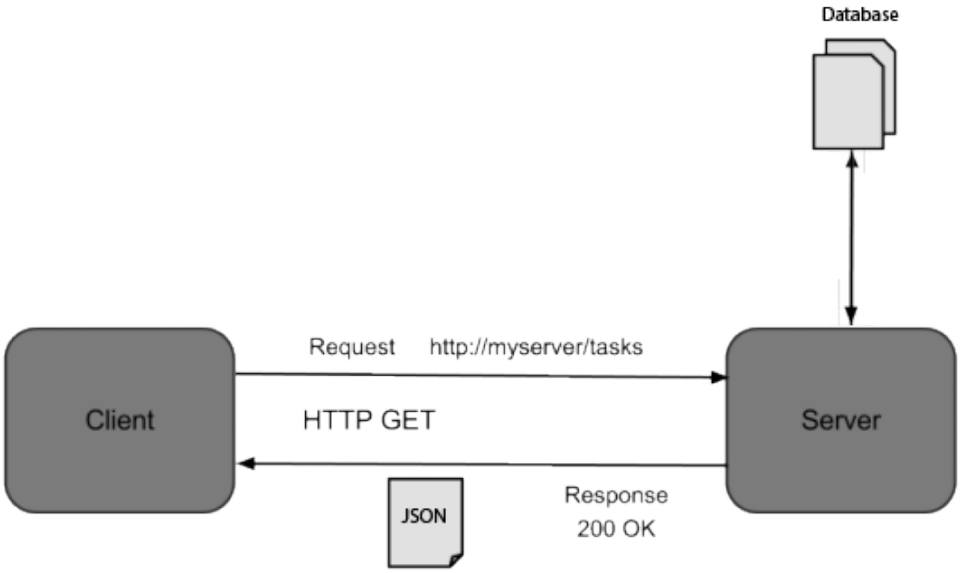

Abbildung 2: 3-Wege-Architektur für die App

Danach werden sie als Inhalt des HTTP-Response an den Client gesendet. Der genauere Ablauf ist im Abbildung 3 ersichtlich.

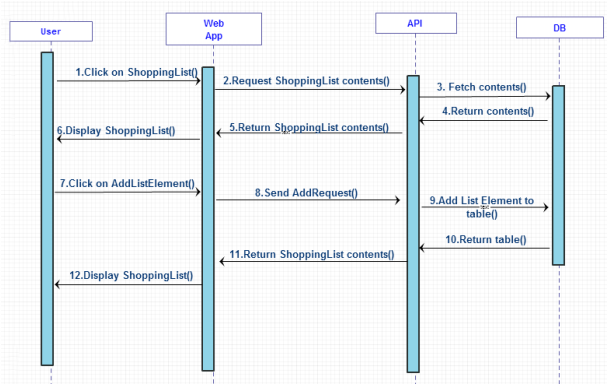

Abbildung 3: Sequenzdiagramm

VII. Fazit

Der Vorteil von Phonegap ist, dass ohne große Rücksichtnahme auf die Plattform entwickelt werden konnte. Der Einsatz von JQuery mobile schafft eine einheitliche Darstellung der mobilen Web-App auf allen Geräten. Die Einbindung von anderen Frameworks ist mit Komplikationen verbunden. Der Grund dafür ist, dass viele Javascript-Frameworks den DOM-Tree manipulieren. Somit musste auf Frameworks wie AngularJS, welche eine Entwicklung nach dem MVC Pattern ermöglicht hätten, verzichtet werden.

VIII. Ausblick

Die App besitzt bereits alle notwendigen Funktionen, um eine Einkaufsliste zu erstellen, zu editieren und zu teilen. In der weiteren Entwicklung zeigen sich die Vorteile, die von Phonegap geboten werden. So kann beispielsweise ein Offline-Modus entwickelt werden, damit die App auch ohne permanente Internetverbindung nutzbar ist. Des Weiteren ist an eine intelligentere Unterstützung der App im Alltag gedacht. Die App soll beispielsweise den User benachrichtigen, falls ein anderer User in einer gemeinsamen Einkaufsliste einen Eintrag geändert oder hinzugefügt hat. Ein weitere Verbesserung wäre auch die permanente Lokalisierung des Users und eine Reaktion der Einkaufslisten-App, wenn sich der User in der Nähe oder in einem Einkaufsladen befindet.

[1, S.7]

References

[1] Bundesministerium fr Bildung und Forschung. It-ausstattung der allgemein bildenden und berufsbildenden schulen in deutschland. http://www.schulen-ans-netz.de/neuemedien/fakten/dokus/it-ausstattung-2003.pdf, 2003.