

Software Engineering Projekt

Gruppe Einkaufsapp

18.Oktober 2015

Inhaltsverzeichnis

 Projektdokumentation	4
Gruppenmitglieder	4
 Einleitung	5
 1 Vorbetrachtung	7
1.1 Problembeschreibung	7
1.2 Entscheidung zur Projektdurchführung	7
1.3 Funktionen	8
1.3.1 Grundfunktionen der App	8
1.3.2 Hauptfunktionen der App	9
1.3.3 Zusatzfunktionen der App	9
1.4 Projektorganisation	10
1.4.1 Kick-Off-Meeting	10
1.4.2 Soll-Analyse	11
1.4.3 Ist-Analyse	13
1.4.4 Anforderungsanalyse	14
1.4.5 Arbeitsplanung	15
1.5 Designer	15
1.6 Entwickler	16
1.7 Dokumentation	17
1.7.1 Scrum	18
1.8 Sicherheit	19
 2 Systemarchitektur/-landschaft	20
2.1 Modell, View, Controller	21
2.2 MongoDB	22
2.2.1 OpenShift Server	23
2.2.2 Basis Tools	24
 3 Durchführungsphase	25
3.1 Registrierung	26
3.1.1 Login	27
3.2 Entwickler	27
3.3 Einkauf	28
3.4 Nutzerverwaltung	29
3.5 Auswertung	30

4	Problemzusammenfassung	31
4.1	Usability der App	31
4.2	Organisation Projektmanagement	32
5	Projektabschluss	33
5.1	Fertiges Produkt	33
5.2	Aussichten	33
5.3	Zusammenfassung	34
6	Lesson learned	35
	Quellen	36
	Internetquellen	36
	Anhang	37
	Glossar	38

Abkürzungsverzeichnis

Tabellenverzeichnis

Projektdokumentation

Gruppenmitglieder

Projektleiter

Markus Hube

Entwicklung

Sebastian Kiepsch

Michael Hein

Eric Sorgalla

Viktor Fuchs

Florian Schmitt

Design

Florian Graupeter

Moritz Karsten

Moritz Schaub

Jannis Grohs

Daniel Sawadenko

Dokumentation

Huong Dang

Thomas Elias

Annika Köstler

Einleitung

Diese Dokumentation soll einen näheren Einblick in den Umfang, den Nutzen, den Ablauf und das Ergebnis unseres Softwareprojekts 'EinkaufsApp' geben.

Die EinkaufsApp dient dem Nutzer seine alltäglichen Einkaufserlebnisse, hinsichtlich der besuchten Läden und gekauften Produkte zu dokumentieren und eine Übersicht über seine Finanzen zu erhalten. Gleichzeitig soll sie als kleines Nachschlagewerk fungieren, welches Überblick über Preis und Angebot bestimmter Produkte bietet. Der alltägliche Einkauf wird hinsichtlich des Monitoring der Finanzen und Produktauswahl aufgrund der Funktionalitäten der EinkaufsApp erleichtert.

Die Dokumentation umfasst die kompletten Phasen der Vorbetrachtung, Planung und Entwicklung der EinkaufsApp mit den jeweiligen Ideen, Tasks und angefertigten Dokumenten und fungiert als Leitfaden für alle Projektmitglieder durch das gesamte Projekt.

Zudem wurde eine Einteilung des Projektes in Definitionsphase, Planungsphase, Durchführungsphase und Abschlussphase als angemessen empfunden und in diesem Dokument angewandt. Diese Dokumentation ist parallel zur Durchführungsphase entstanden.

1 Vorbetrachtung

Die Vorbetrachtung beinhaltet alle vorbereitenden Aktivitäten, die vor der Entwicklung der Applikation getätigt wurden. Dazu gehören die konkrete Problembeschreibung, der darauffolgende Lösungsansatz und die Zielsetzungen für die Umsetzung der Entwicklung.

1.1 Problembeschreibung

Die steigende Vielfalt an Produkten und die Preisschwankungen der Anbieter führen den Konsumenten zu einer Unübersichtlichkeit über die Angebotsvielfalt und der damit verbundenen Ausgaben. Wie in der Einleitung beschrieben, bietet die EinkaufsApp die Möglichkeit einen Überblick über getätigte Einkäufe zu schaffen, um vor allem die finanziellen Ausgaben pro Woche, Monat oder Jahr zu tracken und Preise gleicher Produkte von unterschiedlichen Anbietern zu vergleichen. Diese App soll zudem noch dabei helfen den finanziellen Überblick zu behalten und eine Hilfe für alle Konsumenten sein, die sich öfter fragen, wo ihre Lieblingsprodukte am günstigsten angeboten werden und wie oft sie diese Produkte im Monat kaufen. Zusätzlich gibt es eine Gruppenfunktion, die bestimmten angelegten Gruppen, z.B WG-Mitgliedern, die Möglichkeit bietet, die Ausgaben pro Person zu tracken, was die manuelle Kalkulation am Ende eines Monats erspart.

1.2 Entscheidung zur Projektdurchführung

Die EinkaufsApp soll die EANs (European Article Number) der Produkte, die Konsumenten bei ihren Einkäufen in den Warenkorb legen und bezahlen, speichern. Sie soll es zudem ermöglichen die Preise der Produkte und die damit verbundenen Kosten auf Gruppen oder einzelne Personen aufzuteilen und im Ergebnis eine finanzielle Auswertung aufzeigen. Unser Ziel mit der Umsetzung des Projektes ist es, eine App zu entwickeln, die eine Lösung für das in der Problemstellung genannte Problem darstellt. Die Produktvielfalt der verschiedenen Produkthanbieter wird vereinfacht dargestellt, der Konsument sieht auf einen Blick seine Ausgaben und Gruppenmitglieder müssen nicht noch manuell nach dem Einkauf jegliche Preise zusammenrechnen.

1.3 Funktionen

Unsere Ziele des Projektes sind untergliedert in Grundidee und Systemstruktur:

- Für GPS eine Abfrage an Google Maps implementieren
- Karte mit Standort verschiedener Läden dem Benutzer anzeigen
- Automatisch generierte Einkaufsliste
- GS1 verwaltet alle EANs: 1worldsync soll Schnittstellen für den Zugriff bereitstellen
- 95 Euro kostet Zugang für 1worldsync um ihre Schnittstelle zu verwenden
- Facebook, Twitter usw. zur Registrierung verwenden
- Artikel zuweisen nach Einkauf auf Gruppenmitglieder
- was passiert falls kein Empfang vorhanden ist (Offline-Version)
- Kostenteilung unter Leuten, die nicht in Datenbank und nicht in Gruppe sind und eventuelle Trennung bestimmter Sachen unter den Leuten in der Gruppe z.B. Frauen wollen Sekt, Männer Bier
- Einkaufsbewertung –> wie gefällt der Laden? Wie zufrieden war man mit dem Einkauf? Alles erhalten? Parkplatz vorhanden?
- Benutzerverwaltung: Profilbild, Account löschen, etc.–> unter Einstellung auf Homescreen

Ideen: Out of Focus

- zwei oder mehr Leute können gleiche Liste verwenden und diese abarbeiten sodass die anderen Änderungen sofort sehen
- Einkaufsliste vorher schreiben und diese abarbeiten –> gescannte Produkte werden abgehakt

1.3.1 Grundfunktionen der App

- Informationen über Artikel via EAN aus DB
- von aktiven Einkäufen unabhängige, regelmäßige Kosten zu erfassen.

1.3.2 Hauptfunktionen der App

- Es soll möglich sein für jemand anderen oder eine Gruppe (z.B. WG) etwas einzukaufen.
- Marktfindung über GPS
- Anzeige versch. Märkte im Umkreis
- Login über FB, TWitter o. ä. - Benutzerverwaltung
- Artikelzuweisung nach Einkauf auf Gruppenmitglieder

1.3.3 Zusatzfunktionen der App

- Einkaufsbewertung
- Offline Funktion
- automatisch generierte Einkaufsliste
- Eine Web Site ist momentan out of scope, wäre aber eine sinnvolle Ergänzung für die Ausgabe von Statistiken und die Benutzerverwaltung.
- Auswertung
 - a) Durchschnittliche Tageskosten eines Zeitraumes (z.B. Woche oder Monat)
 - b) Maximal oder Minimalpreis innerhalb eines Zeitraumes (z.B. Woche oder Monat)
 - c) Eine Grafik, die den Ausgabenverlauf innerhalb eines Zeitraumes angibt
 - d) Eine Extrapolation regelmäßig gekaufter Artikel (Ersatz des „Einkaufzettels“)
 - e) Das persönliche Tracking der allgemeinen Ausgaben

1.4 Projektorganisation

Die Projektgruppe der EinkaufsApp teilte sich am 02. Oktober 2015 in die Untergruppen Dokumentation, Design und Entwicklung auf. Der Projektleiter und in diesem Falle auch Projektmanager wurde ebenso an diesem Tag ernannt. Als Projektmanager war er nun für die Team- und Projektorganisation zuständig, wozu das Einhalten der Projekt- und Meilensteinplanung und das Erfüllen der Projektziele hoch priorisiert wurden. Jegliche Unterhaltung basierte auf Mailverkehr oder fand durch Telefonkonferenzen statt. Jede Untergruppe musste sich selbst organisieren und wöchentlich ein Update dem Projektleiter zukommen lassen. Jeden Montag fanden Status-Telekonferenzen statt, wo sich alle Teammitglieder zusammen fanden und über den aktuellen Stand der Untergruppen informierten und über aufgekommene Probleme diskutierten. Die einzelnen Aktivitäten der Untergruppen werden in den Unterpunkten 1.5 - 1.7 noch genau erläutert.

1.4.1 Kick-Off-Meeting

Am 02. Oktober 2015 fand das erste Meeting mit der gesamten Projektgruppe statt. In diesem Kick-Off-Meeting traf man Absprachen über das weitere Vorgehen und die Projektumsetzung der Ideen und Ziele, welche im Kapitel 1.3 ausführlich beschrieben wurden. Es wurde über die weitere Kommunikationsform abgestimmt und ebenso fest gelegt, dass wöchentlich Telefonkonferenzen innerhalb der Untergruppen zum weiteren Vorgehen des Projektes statt finden werden, sodass die genannten Projektziele bis zum festgelegten Datum umgesetzt werden können. Die Untergruppen einigten sich außerdem auf Tools, die effizient und sinnvoll zur Umsetzung der anstehenden Aktivitäten und zum Einhalten der Projektziele verwendet wurden. Die einzelnen Tools der Untergruppen werden ausführlich im Abschnitt Organisationstools aufgezählt und definiert.

1.4.2 Soll-Analyse

In dem hier angeführten Kapitel werden konkrete Ziele für das bevorstehende Projekt formuliert, die auf den oben aufgeführten Funktionen der Applikation basieren. (Tabelle der Grundfunktionen, Die Umsetzungsmöglichkeiten, Prioritäten)

Funktion	nötige Umsetzungsaspekte	Priorität
Datenbank	Aufstellen einer MongoDB	Prio 1

- 1.1) Die App soll in Echtzeit die monetären Ausgaben einer Person speichern, sowie ausgewertet wiedergeben.
- 1.2) Hierfür soll es möglich sein:
 - a) bei einem Einkauf Informationen über einen Artikel von einem Etikett via Barcodescanner einzulesen, beziehungsweise bei bestehender EAN Nummer aus einer Datenbank zu laden und aus diesen Argumenten einen Einkauf zu erstellen
 - b) sonstige Kosten aufzunehmen, die nicht mit einem EAN Code in Verbindung gebracht werden können.
 - c) von aktiven Einkäufen unabhängige, regelmäßige Kosten zu erfassen.
- 1.3) Es soll möglich sein für jemand anderen oder eine Gruppe (z.B. WG) einzukaufen.
- 1.4) Die Daten werden zentral in einer, über das Internet erreichbare, Datenbank gespeichert.
- 1.5) Die App soll primär ein einfaches Front End bereitstellen, um Informationen zu sammeln und zu verwalten
- 1.6) Eine Web Site ist momentan out of scope, wäre aber eine sinnvolle Ergänzung für die Ausgabe von Statistiken und die Benutzerverwaltung.
- 1.7) Die Möglichkeiten der Auswertung sind vielfältig und können in Listen oder Diagrammen dargestellt werden.
- 1.8) Auswertungsbeispiele:
 - a) Ausgaben innerhalb eines bestimmten Zeitraumes (z.B. Woche oder Monat)

- b) Maximal oder Minimalpreis innerhalb eines Zeitraumes (z.B. Woche oder Monat)
 - c) Eine Grafik, die den Ausgabenverlauf innerhalb eines Zeitraumes darstellt
 - d) Eine Extrapolation regelmäßig gekaufter Artikel (Ersatz des „Einkaufzettels“)
 - e) Das persönliche Tracking der allgemeinen Ausgaben
- 1.9) Außerdem nicht personenbezogene Auswertungen:
- a) über beliebteste Artikel
 - b) beliebteste Märkte
 - c) Durchschnittspreise eines Artikels
- 2.1) Ein online verfügbarer Server, auf dem seinerseits ein Datenbank Server und ein Web Server läuft
- a) Als Datenbank-Server wird MongoDB verwendet
 - b) Als Web Server wird Apache verwendet
- 2.2) Auf dem Web Server befindliche PHP-Skripte stellen die Verbindung zur Datenbank her.
- 2.3) Aus der Android-App heraus wird mittels HTTP-Post eine Anfrage an die PHP-Skripte geschickt und die Antwort im JSON-Format wieder an die App zurück geschickt.

1.4.3 Ist-Analyse

Was haben wir? Skillliste!

1.4.4 Anforderungsanalyse

Was benötigen wir?

1.4.5 Arbeitsplanung

Designer: Florian Graupeter:

- Konzeption Grundstruktur
- Wie ist der Ablauf bei der App Nutzung, welche Sonderfälle müssen an welchen Punkten beachtet werden, wie ist die generelle Struktur

Moritz Karsten:

- Konzeption Funktionalitäten und Informationsfluss einzelner Ansichten
- welche Knöpfe soll es geben, welche Informationen werden angezeigt, welche Informationen werden zwischen zwei Ansichten ausgetauscht

Moritz Schaub:

- Visuelles Design
- Farben, Formen und Anordnung, von Butten und Feldern
- eventuelles Logodesign + neuer Name für App

Jannis Grohs:

- Quality of Service (Konsistenz in allem beachten und Vorgaben einhalten)
- Zusammenfassung über Erwartungshorizont und Ausarbeitung zu Fragen
- Einfachheit und Intuition im Design beachten, zuverlässige Fehlerbehandlung beachten

ToDo's für alle Designer:

- Flussdiagramm erstellen und erweitern
- Erwartungshorizont der App definieren
- Anfangsstruktur für die Entwickler festlegen

Entwickler: Hier fehlen leider noch die einzelnen Aufgabenbereiche von Vielen! Unbedingt nachtragen!

Sebastian Kiepsch:

- Einführung in die Technik- Präsentation vorbereiten
- Modulplan erstellen
- bereits erstellten Code kommentieren

Michael Hein:

- Erstellung einer Benutzerverwaltung mit User Interface
- bereits erstellten Code kommentieren

Eric Sorgalla:

- Einführung in die Technik- Präsentation vorbereiten
- Modulplan erstellen
- Kanban-Board erstellen

Viktor Fuchs:

- Namenskonventionen ausarbeiten - Variablen und Funktionen

Florian Schmitt:

- HTML-Code der Proto.io- App exportieren und durchgehen

offene Aufgaben:

- aktualisiertes DB Modell erstellen
- EAN-Kategorien recherchieren und Tabelle füllen
- neues DB Modell implementieren
- UML Diagramm erstellen

Dokumentation:

Thomas Elias:

- Projektplan erstellen
- Skillliste der einzelnen Projektmitglieder erstellen
- Aktivitätsliste aller Projektmitglieder definieren
- Ausarbeiten allgemeiner Informationen für Handbuch

Huong Dang:

- Projektseitige Dokumentation
- Dokumentation der Emails, Meetings und Telefonkonferenzen
- Ausarbeiten der Aktivitäten des Einkaufsprozesses für Handbuch

Annika Köstler:

- Schreiben der endgültigen Dokumentation
- Ausarbeiten der Installationsanleitung

1.4.6 Scrum

Rahmenbedingungen für Profil und Einkaufslisten

1.5 Sicherheit

2 Systemarchitektur/-landschaft

2.1 Modell, View, Controller

2.2 MongoDB

2.2.1 OpenShift Server

2.2.2 Basis Tools

3 Durchführungsphase

In diesem Abschnitt erklären wir, wie die EinkaufsApp nach und nach aufgebaut wurde und gehen insbesondere auf die einzelnen Punkte wie Registrierung, Login, Einkauf, Nutzerverwaltung und die Auswertung ein. (Idee: Roadmap, s. Flussdiagramm) (Was haben die Designer für die App geplant und was können die Entwickler davon umsetzen) -> nachfolgende Struktur erklären

Einleitung

3.1 Registrierung

Bei der EinkaufsApp muss sich jeder Nutzer mittels einer Email-Adresse und einem Kennwort bei der App registrieren. Eine Registrierung ist bei dieser App unentbehrlich, da für jeden Nutzer ein Profil angelegt wird und er mit diesem Profil seine Produkte einscannen kann und seine Finanzen im Blick behalten kann. In den drei folgenden Unterpunkten wird der Prozess der Registrierung aus Seiten der Designer, Entwickler und der Dokumentation beschrieben.

Design

Die Designer befassten sich mit der Optik bei der Registrierung und erstellten als erstes einen Welcome-Screen, wo der Nutzer sich aussuchen kann, ob er sich neu registrieren oder nur einloggen will. Entscheidet sich der Nutzer für die Registrierung werden ihm die Eingabefelder "Email-Adresse", "Nutzername", "Passwort" und "Passwort wiederholen" angezeigt, die er ausfüllen muss. Nachdem er diese ausgefüllt hat, klickt er auf den Button "Registrierung bestätigen" und es werden die eingegebenen Daten kontrolliert: Die Email-Adresse und der Nutzername werden auf Einmaligkeit und die Passwort-Anfragen auf Gleichheit überprüft. Wenn die Daten alle korrekt eingegeben wurden, wird der Nutzer auf den Login Screen weiter geleitet um sich dann anzumelden. Doch falls bei bestimmten Daten Unstimmigkeiten auftreten, erscheinen folgende Fehlermeldungen:

1. Bei einer Nicht-Übereinstimmung der Passwörter taucht eine Meldung auf, die den Nutzer auf den Fehler hinweist und verweist ihn dazu die Passwörter neu einzugeben. Die Email-Adresse und der Nutzername bleiben in diesem Fall voreingetragen in den Eingabefeldern stehen.
2. Bei bereits existierender Email-Adresse wird der Nutzer, durch eine Fehlermeldung, darauf hingewiesen und muss diese ändern. Hierbei bleibt nur der Nutzername voreingetragen stehen.
3. Bei bereits existierendem Nutzernamen erscheint eine Meldung, die den Nutzer darauf hinweist, dass der Name bereits vergeben ist. Der Nutzer sieht dann nur seine Email-Adresse noch voreingetragen und muss einen neuen Nutzernamen eintragen.

Entwicklung

Die Entwickler befassen sich mit den Funktionen der Applikation und sorgen bei der Registrierung dafür, dass alle Daten ordentlich geprüft und in die Datenbank eingepflegt werden. Als Datenbank wird die MongoDB genutzt und via RoboMongo gemanagt. Als Programmiersprache wird in allen, auch den

folgenden Unterpunkten JavaScript verwendet. Beim Registrieren wird nach Eingabe der Daten durch den Nutzer die Datenbank abgefragt, ob bereits ein solcher Nutzernamen und eine solche Email-Adresse bereits existiert. Zudem wird noch überprüft, ob die eingetragenen Passwörter übereinstimmen und ob die Email-Adresse im richtigen Format eingetragen wurde. Sobald alles übereinstimmt und weder Nutzernamen, noch Email-Adresse in der Datenbank bereits existieren, wird der Nutzer angelegt und das Passwort als Hash in der Datenbank gespeichert. Das bedeutet, dass das eingetragene Passwort als ein kleinerer Wert abgebildet und dieser gespeichert wird.

Dokumentation

–; Probleme beschreiben Eine ganz wichtige und entscheidende Rolle spielt hierbei der Datenschutz und die Datensicherheit. Alle eingetragenen Daten des Nutzers müssen abgesichert sein, sodass kein Dritter sich an diesen Daten vergreifen kann.

3.1.1 Login

Nachdem sich der Nutzer erfolgreich registriert hat, muss er sich nun um die EinkaufsApp nutzen zu können mittels seiner Email-Adresse und dem Passwort anmelden. Auch dieser Teil ist unentbehrlich für die Nutzung der App.

Design

Die Aufgabe der Designer bestand hier, einen anschaulichen und einfach bedienbaren Login-Screen für den Nutzer zu kreieren. Zuallererst wird der Nutzer, wie bereits bei der Registrierung schon erwähnt auf einen Welcome-Screen geleitet, wo er die Wahl trifft zwischen Login und Registrierung. Bei bereits abgeschlossener, erfolgreicher Registrierung wählt der Nutzer nun den Login aus, um sich anzumelden. Es folgt dann ein weiterer Screen mit den Eingabefeldern Email-Adresse und "Passwort". Nach dem Eintragen der Nutzerdaten klickt der Nutzer auf den AnmeldenButton um sich einzuloggen. Auch hierbei werden die Daten überprüft und wenn mindestens eine Eingabe falsch war, erscheint eine Fehlermeldung die besagt, dass die Email-Adresse oder das Passwort unbekannt sind. Der Nutzer muss seine Daten nun erneut eintragen und seine Email-Adresse wird dabei automatisch voreingetragen. Wenn der Login erfolgreich war, wird der Nutzer auf den Home Screen weiter geleitet. Bei Vergessen des Passwortes gibt es eine Funktion, die es ermöglicht, das Kennwort zurücksetzen zu lassen. Der Nutzer klickt auf den Button "Passwort vergessen" und gelangt auf einen neuen Screen. Dieser Screen ermöglicht es dem Nutzer seine Email-Adresse einzutragen und mittels dem Button Zurücksetzen ein neues Passwort an die eingetragene Mail senden zu lassen. Nachdem die Mail zur Passwortrücksetzung gesendet wurde und die Email-Adresse des Nutzers bekannt war, informiert ein Ausgabefenster den Nutzer, dass die Mail erfolgreich verschickt wurde. Falls die E-Mail-Adresse nicht existiert, wird ebenso ein Ausgabefenster angezeigt mit der Information, dass die Email-Adresse nicht bekannt ist und der Nutzer kann es erneut versuchen. Wenn der Nutzer eine Mail erhalten hat und auf den darin enthaltenen Link klickt, gelangt er auf einen neuen Screen, der ihn dazu auffordert ein neues Passwort einzugeben. Dieses muss er wieder bestätigen und dann gelangt er wieder ins Hauptmenü der EinkaufsApp.

Entwicklung

Die Entwicklung beschäftigt sich hierbei mit der Funktionsweise des Tests ob die eingegebene Email-Adresse und das dazugehörige Passwort korrekt eingegeben wurden. Dazu schaut die Applikation in der Datenbank nach und

überprüft die eingegebenen Daten mit den Daten in der Datenbank. Falls diese nicht übereinstimmen, wird der Nutzer nicht eingeloggt. Falls er sein Passwort vergessen hat, kann er dieses wieder zurücksenden lassen und erhält eine Email an die eingegebene Mail-Adresse, welche sich in der Datenbank befinden muss. Der Nutzer muss bei Erhalt der Nachricht diese öffnen und erhält einen Link mit einem Token, welcher nur eine Stunde gültig ist, um sein Kennwort zurücksetzen zu lassen. Wenn er diesen Link anklickt, prüft die Datenbank ob der Token richtig und noch gültig ist und leitet den Nutzer weiter zu einer Seite, wo er sein Kennwort ändern kann. Nachdem der Nutzer sein Passwort zweimal eintragen musste, ändert die Datenbank das Kennwort des Nutzer und speichert dieses.

Dokumentation

–¿ Probleme beschreiben

3.2 Einkauf

Gruppeneinkauf vereinfacht, Einkauf fortsetzen eingebaut und Passwort rücksetzen
Gruppenverwaltung wurde zu Verwaltung allgemein: Verwaltung von Gruppen UND KONTAKTE Kontaktverwaltung wird noch erstellt”

Design

Entwicklung

Dokumentation

–j Probleme beschreiben

3.3 Nutzerverwaltung

Design

Entwicklung

Dokumentation

– Probleme beschreiben

3.4 Auswertung

Design

Entwicklung

Dokumentation

– Probleme beschreiben

4 Problemzusammenfassung

4.1 Usability der App

4.2 Organisation Projektmanagement

5 Projektabschluss

5.1 Fertiges Produkt

5.2 Aussichten

nicht umgesetzte Ideen – siehe Excelliste

5.3 Zusammenfassung

6 Lesson learned

Quellen

Internetquellen

1. Ionic Framework: <http://ionicframework.com/>
2. Ionic Guide: <http://ionicframework.com/docs/guide/>
3. Ionic Getting Started: <http://ionicframework.com/getting-started/>
4. ngCordova - Plugin Seite <http://ngcordova.com/>
5. BarCode Scanner : Plugin <http://ngcordova.com/docs/plugins/barcodeScanner/>
6. Beispiel Projekt: <https://github.com/bastisk/suedm>
7. Editor: <http://brackets.io/>
8. Angular JS-Kurs: <https://www.codeschool.com/courses/shaping-up-with-angular-js>
9. Tutorial zum Routing: <https://scotch.io/tutorials/angular-routing-using-ui-router>
10. App-Projekt: <http://www.mobile2b.de/ablauf-app-projekt/>
11. Dokumentationshilfe: <http://www.tellsbells.de/dokuwebsite/tbdokumentation.pdf>
12. Dokumentationshilfe: <https://www.lecturio.de/magazin/projekte-dokumentieren/>
13. Open Source mit API über eine einfachen HTTP-GET-Request: <http://www.opengtindb.org/api.php>
14. Suchmaschine der Firma die GTIN-Nummern verwaltet: http://www.gepir.de/v31/V31_client/gtin.aspx

Organisationstool- Übersicht

- Allgemeine Ablage: GitHub
- Diskussionsrunden: Slack
- Informationsaustausch: via Email
- Diagramme zeichnen: via Dia
- Kreieren von Web-Prototypen: proto.io
- Datenbanken und Datenbankenverwaltung: MongoDB, RoboMongo

Anhang