

# Software Engineering Projekt

Gruppe Einkaufsapp

18.Oktober 2015

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>7</b>
<b>1 Vorbetrachtung</b>	<b>8</b>
1.1 Problembeschreibung . . . . .	8
1.2 Zielsetzung . . . . .	8
1.3 Vorbereitende Fragen . . . . .	9
1.4 Projektorganisation . . . . .	12
1.4.1 Anforderungskatalog . . . . .	13
1.4.2 Ist-Analyse . . . . .	14
1.4.3 Arbeitsplanung . . . . .	16
1.4.4 Agiles Projektmanagement . . . . .	17
1.5 Sicherheit . . . . .	18
<b>2 Durchführungsphase</b>	<b>19</b>
2.1 Architektur . . . . .	19
2.2 Registrierung . . . . .	20
2.3 Login . . . . .	23
2.4 Marktauswahl . . . . .	24
2.5 Einkaufsverwaltung . . . . .	27
2.5.1 Design . . . . .	27
2.6 Nutzerverwaltung . . . . .	28
2.7 Auswertung . . . . .	29
<b>3 Problemzusammenfassung</b>	<b>32</b>
3.1 Usability der App . . . . .	32
3.2 Organisation und Projektmanagement . . . . .	33
<b>4 Projektabschluss</b>	<b>34</b>
4.1 Fertiges Produkt . . . . .	34
4.2 Aussichten . . . . .	34
<b>5 Lesson learned</b>	<b>35</b>
<b>Quellen</b>	<b>36</b>
<b>Anhang</b>	<b>38</b>

# Abkürzungsverzeichnis

1. EAN - European Article Number
2. App - Applikation
3. WG - Wohngemeinschaft
4. ER - Entity Relationship
5. UML - Unified Modeling Language
6. SSL - Secure Socket Layer
7. GTIN - Global Trade Item Number

# **Tabellenverzeichnis**

Tabelle 1: Anforderungsanalyse

Tabelle 2 : Anforderungsanalyse

Tabelle 3: Aussichten

# Abbildungsverzeichnis

- Abbildung 1: Aktivitätsliste
- Abbildung 2: Meilensteinplanung
- Abbildung 3: Klassen-Diagramm
- Abbildung 4: Flussdiagramm-Login
- Abbildung 5: Marktauswahl
- Abbildung 6: Aktivitätsdiagramm Einkauf
- Abbildung 7: Gruppenverwaltung
- Abbildung 8: Aktivitätsdiagramm Auswertung
- Abbildung 9: Zustandsdiagramm
- Abbildung 10: Auswertung

# **Projektdokumentation**

## **Gruppenmitglieder**

### **Projektleiter**

Markus Hube

### **Teilprojektleitung - Entwicklung**

Eric Sorgalla

### **Entwicklung**

Sebastian Kiepsch

Michael Hein

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 des Softwareprojekts 'EinkaufsApp' geben. Die EinkaufsApp dient dem Nutzer dazu sein alltägliches Einkaufserlebnisse, hinsichtlich der besuchten Läden und gekauften Produkte zu tracken 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 der Nachverfolgung von Finanzen und Produktauswahl, durch die Funktionen der Einkaufs-App stark erleichtert.

Die Dokumentation umfasst die Phasen der Vorbetrachtung und Entwicklung der EinkaufsApp mit den jeweiligen Ideen, Tasks und angefertigten Dokumenten und dient als Reflexion aller Projektmitglieder über das gesamte Projekt. Zudem wurde eine Einteilung des Projektes in Vorbetrachtung, Durchführungs- phase, Problembeschreibung und Abschlussphase als angemessen empfunden und in diesem Dokument angewandt.

# 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 Problembeschreibung kann aus dem Pflichtenheft im Anhang entnommen werden.

## 1.2 Zielsetzung

Die EinkaufsApp soll die EANs (European Article Number), beziehungsweise die neuere GTIN (Global Trade Item Number), der Produkte, die von den Konsumenten gekauft werden, zusammen mit dem Datum, dem Einkaufsort und ihren Kosten speichern. Sie soll es zudem ermöglichen die Preise der Produkte und die damit verbundenen Kosten auf Gruppen oder einzelne Personen zu verteilen und im Ergebnis eine finanzielle Auswertung aufzeigen. Das Ziel des Projektes ist es, eine App zu entwickeln, die eine Lösung für die unter 1.1 dargestellten Herausforderungen bereitstellt. Die Vielfalt an Produkten wird vereinfacht dargestellt, der Konsument sieht auf einen Blick eine Zusammenfassung seiner Ausgaben, sowie den Finanzstatus innerhalb der Gruppen in denen er Mitglied ist. Die Nutzer erhalten eine automatisierte Auswertung über Einkaufsverläufe, entstandene Kosten und Artikel, welche auf Anfrage bestimmter, anderer Personen erworben wurden.

Zum jetzigen Zeitpunkt soll die zu entwickelnde Applikation vorerst als Tracking-tool dienen. Funktioniert dieses einwandfrei so kann eine weitere Funktionalität implementiert werden nämlich der automatischen Einkaufslistengenerierung. Diese soll nun vergangene Einkäufe auswerten und anhand dieser Einkaufslisten erstellen, die die größtmöglichen Sparpotentiale für den Nutzer bieten. Hier kann dieser vorab festlegen, in wie vielen Märkten er höchstens einkaufen möchte und wie viel Budget er für den jetzigen Einkauf zur Verfügung hat. Dadurch spart sich der Nutzer die manuelle Einkaufsanalyse und somit auch die dafür benötigte Zeit.



## 1.3 Vorbereitende Fragen

### 1. Wer arbeitet mit dem Softwaresystem?

Mit dem Softwaresystem kann jede Privatperson arbeiten, die ihren Einkauf digital dokumentieren und Auswertungen des eigenen Kaufverhaltens erhalten möchte. Des Weiteren hilft diese App jedem, der für Gruppen, z. B. Mitgliedern einer Wohngemeinschaft, Einkäufe tätigt und eine direkte Zuteilung der einzelnen Produkte zur jeweiligen Person wünscht. Die App richtet sich auch an Menschen, die mit Hilfe der Auswertung mögliche Sparpotenziale erkennen und wahrnehmen möchten.

### 2. Welcher Benutzer benötigt welche Funktionen?

Insgesamt werden in der EinkaufsApp drei verschiedene Nutzerrollen vorgesehen: Einerseits existiert der Standarduser, welche Rolle jeder Nutzer einnimmt nachdem er sich angemeldet hat. Wenn dieser wiederum in der EinkaufsApp eine Gruppe erstellt kann er zusätzlich die Rolle des Gruppenadministrators einnehmen und kann weitere Gruppenmitglieder hinzufügen. Somit bestehen die Rollen Standarduser, Gruppenadmin und Gruppenmitglied, welche im Folgenden genauer differenziert werden:

- Standarduser (ohne Gruppenzugehörigkeit):
  - Einkauf einlesen
  - Einkauf löschen
  - Neuen Einkauf starten
  - Gruppe erstellen
  - Auswertungen abrufen
  - Passwort ändern
  - Markt auswählen
  - Neuen Markt hinzufügen
  - Neue Artikel zum Datenbestand hinzufügen
- Gruppenadmin:

Dieser erbt die Funktionalitäten des Standardusers und kann darüber hinaus noch folgende Funktionen ausführen:

  - Gruppe löschen
  - Einkauf einem Gruppenmitglied zuordnen
  - Neue Mitglieder hinzufügen
  - Weitere Gruppenadmins festlegen
- Gruppenmitglied:

Dieser erbt ebenfalls die Funktionalitäten des Standardusers und kann darüber hinaus noch folgende Funktionen ausführen:

  - Einkauf einem Gruppenmitglied zuordnen

Hinweis: Ein Gruppenadmin ist gleichzeitig auch ein Gruppenmitglied.

**3. Welche Informationen müssen zu einer Person, dem Benutzer, gespeichert werden, um einen Geschäftsprozess, z. B. das für eine WG einkaufen, mit dem System abzuwickeln?**

Folgende Informationen müssen vom System gespeichert werden, damit ein Einkauf, für z. B. eine WG, stattfinden kann:

- Eindeutiger Name des User in der Gruppe
- Eindeutiger Gruppenname
- Zuordnung des Users zu der Gruppe
- Produktname
- Produktmenge
- Produktpreis
- Märkte (Name und Standort)
- Einkaufsdatum

**4. Welche im Szenario nicht genannten Funktionen werden von dem Softwaresystem benötigt, um heutigen Anforderungen zu entsprechen? Nennen Sie beispielhaft fünf Funktionen!**

- a. Separater Zugang für Anbieter, z. B. Supermärkte um Angebote einzupflegen, die der Käufer via Push-Notification bekommt
- b. Bewertung eines Marktes durch Käufer
- c. Anzeigen der Bewertung eines Marktes für alle Nutzer
- d. Erstellen eines monatlichen Auswertungsreports via Push-Notification
- e. Erstellen manueller Einkaufslisten vor dem Einkauf
- f. Nutzung der EinkaufsApp über Social Media (z. B. Twitter oder Facebook)

## 5. Was ist ein Anwendungsfall und welche Beziehungen zwischen Anwendungsfällen beschreibt der Standard?

Ein Anwendungsfall ist die Beschreibung eines Szenarios innerhalb einer gesamten Anwendung. Dabei beginnt in der Regel der Prozess mit einem Startzustand („Precondition“), dem Akteur, der Abwicklung („Main Flow“), und dem Zielzustand. Je nach Anwendungsfall bzw. Use Case werden die einzelnen Parameter unterschiedlich angegeben. Es ergeben sich die zwei verschiedenen Beziehungen „Include“ und „Extend“. Die Include-Beziehung im Anwendungsdiagramm beschreibt eine abhängige Zugehörigkeit eines Anwendungsfalls zu einem anderen. Im Gegensatz dazu beschreibt die Extend-Beziehung eine unabhängige Erweiterung eines Anwendungsfalls.

Als Beispiel stehen folgende Use Cases in der EinkaufsApp in Beziehung:

Include-Beziehung:



Extend-Beziehung:



## 1.4 Projektorganisation

Am 02. Oktober 2015 fand das erste Meeting mit der gesamten Projektgruppe der EinkaufsApp statt, hierbei wurden Absprachen über das weitere Vorgehen und die Projektumsetzung der Ideen und Ziele getroffen. Das gesamte Team teilte sich zur optimalen Zielerfüllung 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. Dazu gehört unter anderem das Einhalten der Projekt- und Meilensteinplanung und das Erfüllen der Projektziele. 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-Telefonkonferenzen statt, wo sich alle Teammitglieder zusammenfanden und über den aktuellen Stand der Untergruppen informierten und über aufgekommene Probleme diskutierten. Die Untergruppen einigten sich außerdem auf Tools, die effizient und sinnvoll zur Umsetzung der anstehenden Aktivitäten und zum Einhalten der Projektziele verwendet wurden.

### 1.4.1 Anforderungskatalog

In dem hier angeführten Kapitel werden konkrete Ziele für das bevorstehende Projekt formuliert, die auf den zuvor aufgeführten Funktionen der Applikation basieren. Die einzeln genutzten Tools die im Pflichtenheft, welches sich im Anhang befindet, festgehalten sind, werden hier den einzelnen Arbeitsgruppen zugewiesen.

Umsetzung	Tool	Definition
Entwickler - Backend	Node.JS	Tool, welches JavaScript als Serversprache ausführt um JavaScript Applikationen zu entwickeln
Entwickler - Backend	Express Webframework	Framework für die Webentwicklung auf Basis von Node.JS um folgende Aufgaben zu vereinfachen: Routes erstellen wo die API mit zusammenhängt und stellt die Schnittstelle zur Rendering Engin Jade bereit
Entwickler - Backend	Jade	Rendering Engin für Erstellung von Views, generiert HTML
Entwickler - Backend	Passport.JS	Modul für die Authentifizierung des Endnutzers in Express
Entwickler - Backend	Mongoose	Objektmodellierungstool für Node.JS - Übernimmt Datenbankbindung sowie damit verbundene Businesslogik  Mongoose schafft Datenmodelle, Controller greifen darauf zu und bestimmen was im View dargestellt wird
Entwickler - Backend	MongoDB	Open-Source Datenbank
Entwickler - App	Ionic	Framework mit HTML, CSS und JS Komponenten zur Entwicklung von Hybriden Apps
Entwickler - App	Cordova	Development Framework für die Entwicklung in Javascript anstatt mit den nativen Sprachen der Plattformen
Entwickler - App	Cordova Bar Code Scanner	verwendetes Plugin um Bar Codes zu scannen
Entwickler - App	Views	HTML Ansichten; Routes: Navigation und Addressierung in der App; Controllers: Business Logik der App; Factories: Datenbankbindung
Designer - logisch	Flussdiagramm via dia	Tool, mit dem Flussdiagramme gezeichnet werden können
Designer - optisch	Proto.io	Prototyperstellung der App
Dokumentation - Testing		Testen der implementierten Funktionen, Qualitätssicherung
Organisation - Versionierung	Github	Kommunikation innerhalb des Projektes und File Sharing
Designer und Entwickler - UML Diagramme	dia	

Tabelle 1: Anforderungsanalyse

### 1.4.2 Ist-Analyse

Aus vorangegangener Erfahrung mit dem Thema der automatisierte Unterstützung durch eine App bei Einkäufen im privaten Bereich ist ein Grobkonzept, eines ER-Modells, bereits in das Projekt überführt worden. Dieses wurde verwendet, um ein Grundverständnis beim designen und implementieren zu erzeugen.

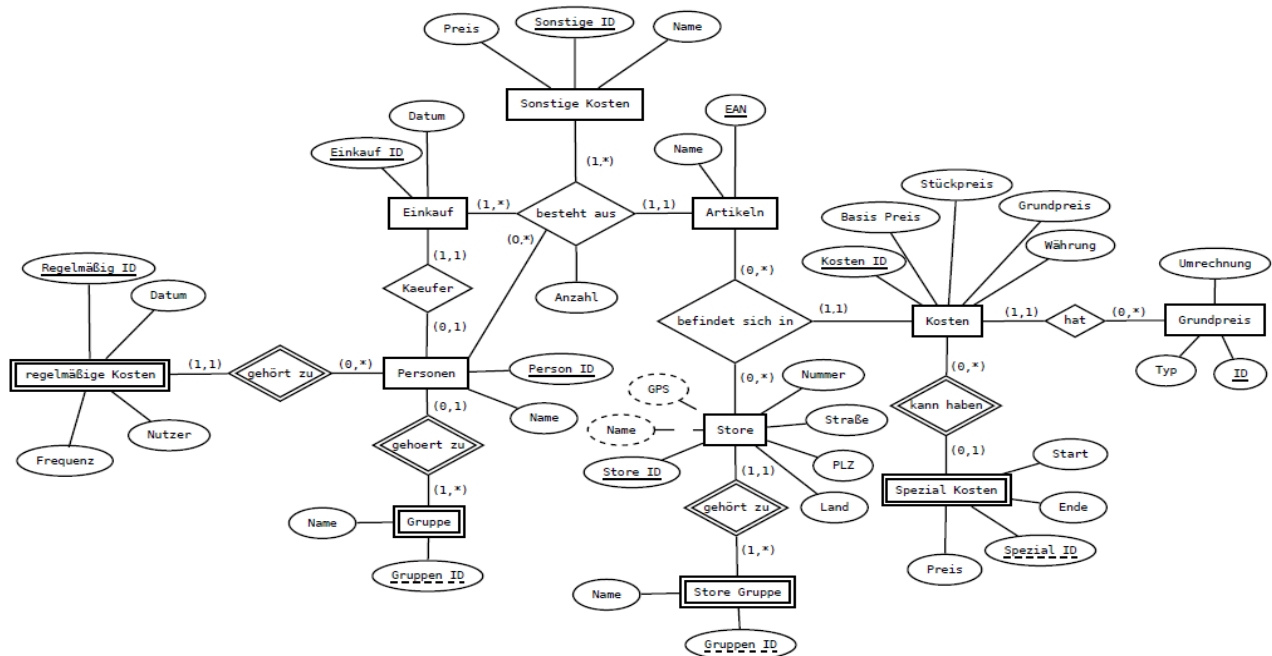


Abbildung 2: ER - Modell

Zu Beginn wurden die jeweiligen Kompetenzen der Projektmitarbeiter vor der Durchführung des Projektes aufgenommen. Aus diesen leiteten sich die Zugehörigkeiten jeder einzelnen Person in die Projektgruppen Dokumentation, Entwicklung und Design ab.

Name	Skills vor Projektstart	Teamzuordnung
Annika Köstler	Zwei Jahre Controlling	Annika Köstler wird aufgrund ihrer Kompetenzen im Bereich Protokollierung in der Gruppe Dokumentation arbeiten.
	Tools Latex	
	Grundkenntnisse VBA	
	Protokollierung von Meetings	
Eric Sorgalla	Grundkenntnisse (Java, C/C++, Javascript, HTML/CSS, VBA, SQL)	Eric Sorgalla wird aufgrund seiner Programmier- und Projektmanagement Erfahrung das Team Entwicklung leiten.
	1 Jahr Projektleitung ISIPT (nur kaufmännische Verantwortung)	
Huong Dang	Zwei Jahre Vertrieb	Aufgrund der regelmäßigen Quality Check Aufgaben im Betrieb arbeitet Huong Dang in der Gruppe Dokumentation.
	Tools LaTeX	
	Grundlagen VBA	
Jannis Grohs	Datenbanken (MYSQL, Apex)	Jannis Grohs wird aufgrund seines Know-Hows für das kontinuierliche Quality Check der Entwicklung und Designs zuständig sein.
	Programmiererfahrung (VBA, JAVA, Apex)	
	Projektmanagement	
	Design + Marketingtechniken	
Markus Hube	Zwei Jahre PMO der operational services	Da Markus Hube bereits zum Thema EinkaufsApp Vorarbeit geleistet und Erfahrung im Projektmanagement gesammelt hat, übernimmt er die Position des Projektleiters.
	Zwei Jahre Programmiererfahrung (VBA)	
	Bereits Vorarbeit zum Projekt geleistet	
Michael Hein	Zwei Jahre Applikations Administration	Durch die langjährige Programmiererfahrung wird Michael Hein bei den Entwicklern tätig sein.
	Java Erfahrung	
	VBA Erfahrung	
	Skript Programmierung	
Moritz Karsten	Zwei Jahre Projektansprechpartner Messe	Durch seine Erfahrung im Bereich Prozessablauf und Konzeptentwicklung wird
	Application Management	
Moritz Schaub	2 Jahre Co-Product Owner in iOS und Android Messaging Produkt in AGILER Entwicklung	Da Vorkenntnisse im Bereich Prototypenstellung von Apps bestehen, wird Moritz Schaub für das visuelle Design zuständig sein.
	Erstellung von komplexen Prototypen mit Proto.io auf Basis von HTML5	
	Vorbereitung, Durchführung von Design Thinking Workshops, Designstudien	
	gute Kenntnisse in Projektkoordination	
Thomas Elias	Anforderungen und Arbeitspakete definieren	Aufgrund seiner guten Kenntnisse im Bereich Projektkoordination wird Thomas Elias in der Dokumentation arbeiten und für die Meilensteinplanung verantwortlich sein.
	Erfahrungen Customizing von Dokumenten-Layouts	
	Kommunikation zwischen versch. Abteilungen zum Transparent-Machen der Informationen	
	gute Excelkenntnisse	
Victor Fuchs	gute Kenntnisse im Bereich Rechnungswesen-Controlling	Victor Fuchs wird in der Gruppe Entwicklung arbeiten und hierbei die benötigten Konzeptdiagramme erstellen
	1 Jahr T-Systems, Expert Services	
Sebastian Kiepsch	1 Jahr T-Systems, Digital Division	Sebastian Kiepsch wird wegen seiner ausgeprägten Programmiererfahrung bei der Entwicklung tätig sein.
	Programmiererfahrung (Javascript, HTML, CSS, C#)	
	Frameworks: Node.JS, Express.JS, Ionic	
	Application Management bei T-Systems BTG	
Florian Graupeter	Benutzeradministration im SAP	Florian Graupeter wird bei den Designern im logischen Design mitwirken.
Daniel Sawadenko	VBA Kenntnisse aus Beruf und Uni	Daniel Sawadenko wird aufgrund seiner Fähigkeiten im Team Design arbeiten.
	Datenbankkenntnisse (MS Access)	
	Kenntnisse der UML Erstellung	
Florian Schmitt	Scrum - Product Owner	Florian Schmitt arbeitet bei den Entwicklern aufgrund seiner Erfahrung im Bereich Software-Entwicklung
	Mobile Device Management	
	Software-Entwicklung	
	Projektmanagement	

Tabelle 2: Ist-Analyse

Insgesamt gibt es demnach drei Designer, fünf Entwickler und drei Dokumentatoren, die parallel den stetigen Quality Check durchführen.

### 1.4.3 Arbeitsplanung

Zu Beginn der Projektorganisation wurde von der Dokumentation ein grober Plan erstellt, der eine Einteilung der Teams in organisatorische Einheiten aufzeigt und einen Rahmen für die Planung der Aufgaben, beziehungsweise Arbeitspakete, vorgibt. Es wurde ein organisatorisches Grundgerüst geschaffen, das allen Gruppen als Orientierung dient und gleichzeitig zur eigenständigen Organisation, sowie Bearbeitung der Arbeitspakete motiviert:

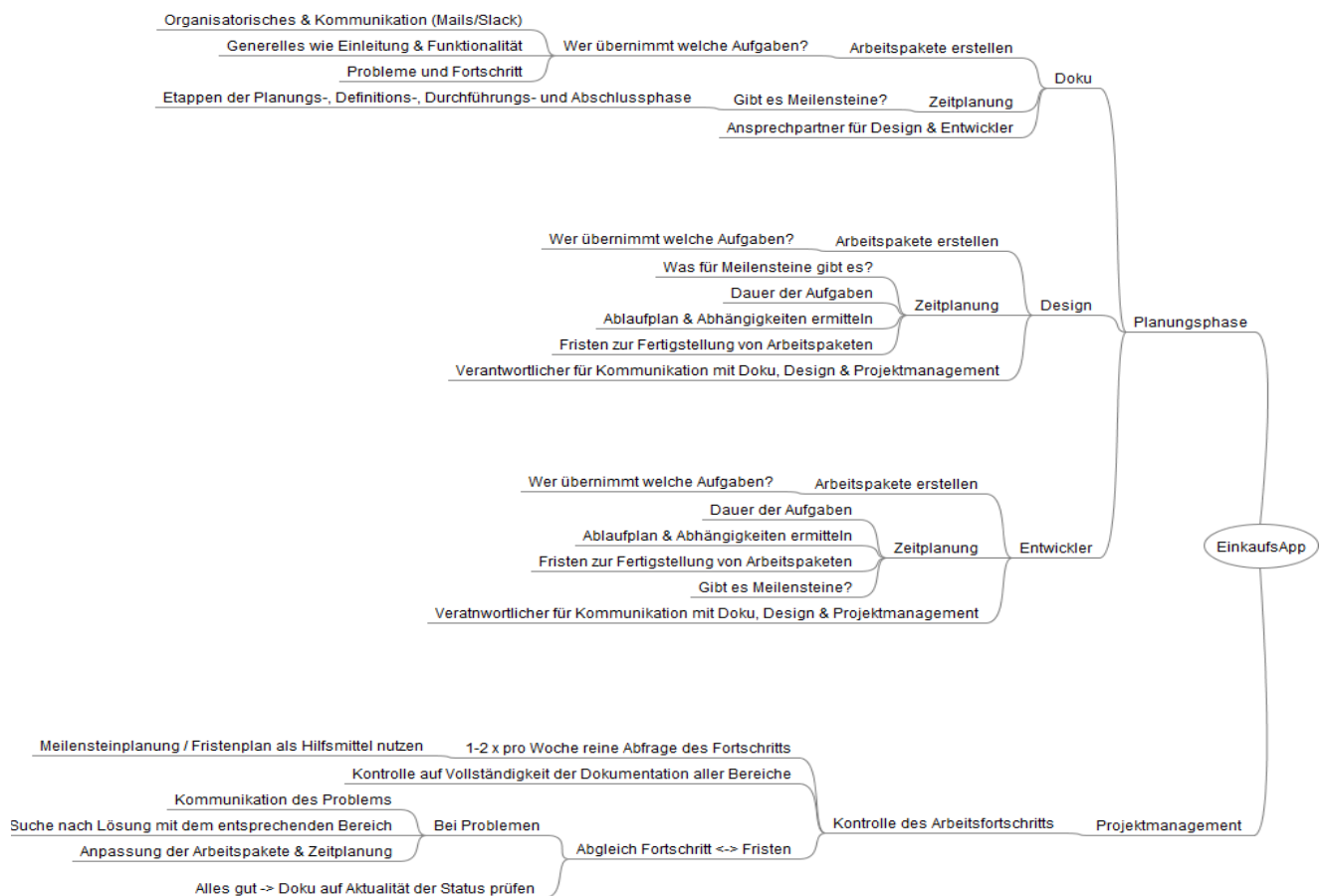


Abbildung 1: Aktivitätsliste



## Meilensteinplanung

Datum	Gruppe	Meilenstein
23.10.2015	Design	Fertigstellung Login Dia
30.10.2015	Development	Fertigstellung Login
06.11.2015	Development	Fertigstellung Menü
06.11.2015	Design	Fertigstellung Einkaufsmodul Dia
13.11.2015	Dokumentation	Handbuch Entwurf erstellen basiert auf Development Status
13.11.2015	Development	Fertigstellung Einkaufsmodul
13.11.2015	Design	Fertigstellung Auswertungsmodul
20.11.2015	Dokumentation	Fertigstellung Struktur der Doku und 1. Version des Handbuchs
20.11.2015	Development	Fertigstellung Auswertungsmodul
20.11.2015	Design	Fertigstellung Gruppenverwaltungsmodul Dia
20.11.2015	Alle	Evaluation aller Dokumente, die erstellt werden müssen (UMLs...)
27.11.2015	Development	Fertigstellung Gruppenverwaltungsmodul
04.12.2015	Dokumentation	Anfang Schreiben der finalen Doku und des Handbuchs
04.12.2015	Development	App Fertigstellung - Pen tests
04.12.2015	Alle	AZ Woche - Dokumentation, Fehlerbehebung, usw
11.12.2015	Alle	App finale Fertigstellung
18.12.2015	Alle	Dokufertigstellung und Handbuch

Abbildung 2: Meilensteine

### 1.4.4 Agiles Projektmanagement

Nachdem ein grober organisatorischer Rahmen für das Projekt von der Gruppe Dokumentation vorgegeben wurde, haben die einzelnen Gruppen, durch agile Projektmanagement-Methoden, ihre Arbeitspakete und Ablaufpläne festgelegt. Die Dokumentation erstellte Arbeitspakete für alle Gruppen und jedes Projektmitglied hat sich nach dem Pull-Prinzip – bekannt aus der Projektmanagement-Methode Kanban – seine Arbeitspakete abgeholt und eine Bearbeitungsfrist definiert. In der Gruppe der Entwickler wurde unter Zuhilfenahme des Tools „Trello“ die Planung und Durchführung der Arbeitspakete definiert. Trello ist ein Web-Dienst, der ein Board anbietet, um Arbeitspakete gemäß agiler Projektmanagement-Methoden zu bearbeiten und Arbeitsfortschritte transparent darzustellen. Die Designer haben ihre Arbeitspakete auf Basis eines Ablaufplanes verteilt. Es wurden die Phasen Prototyp-Entwurf, Prototyp-Review, Prototyp-Modifikationen und Prototyp-Test und Prototyp-Abnahme durchlaufen.

## 1.5 Sicherheit

Sobald Daten eines Nutzers für eine Applikation gespeichert werden, wird ein gewisser Standard an Sicherheit gefordert, damit keinen Dritten diese Daten zugänglich werden. In der EinkaufsApp wurden folgende Maßnahmen getroffen um dies gewährleisten zu können:

1. Der Nutzer startet durch die Eingabe von Benutzername und persönlichem Passwort eine Session, die ihn bei jeder Anfrage an den Server authentifiziert.
2. Alle Passwörter werden als Einweg-Hash in der Datenbank gespeichert. Bei der Anmeldung wird das Passwort dem gleichen Verfahren unterzogen und dann die resultierende Zeichenfolge mit der, in der Datenbank gespeicherten, abgeglichen.
3. Die Verwendung des SSL-Protokolls (Secure Socket Layer) sorgt für den Aufbau eines geschützten Kanals vor der eigentlichen HTTP-Kommunikation, so dass die Nutzerdaten für Dritte nicht zugänglich sind.

Dies bedeutet: Die Nutzerdaten werden in einem standard Web-Formular eingetragen (Login-Screen) und mittels POST-Request an den Server gesendet. Da der TCP-Kanal verschlüsselt ist, haben Dritte keinen Zugriff auf die vom Nutzer eingegebenen Daten innerhalb des POST-Requests, was wiederum die gesicherte Übertragung von Nutzernamen und Passwort bewirkt.

## 2 Durchführungsphase

In diesem Abschnitt wird die Funktionsweise der EinkaufsApp beschrieben. Dabei werden die einzelnen Hauptprozesse separat vorgestellt und deren technische Umsetzung erläutert. Die Hauptprozesse sind unterteilt in den Login und die Registrierung des Nutzers, den Einkaufsprozess, die Nutzerverwaltung und die Auswertung.

### Einleitung

Die genannten Hauptprozesse stehen, wie in dem folgenden Diagramm zu sehen ist, in Relation. Dabei wurden die Views als einzelne Klassen dargestellt:

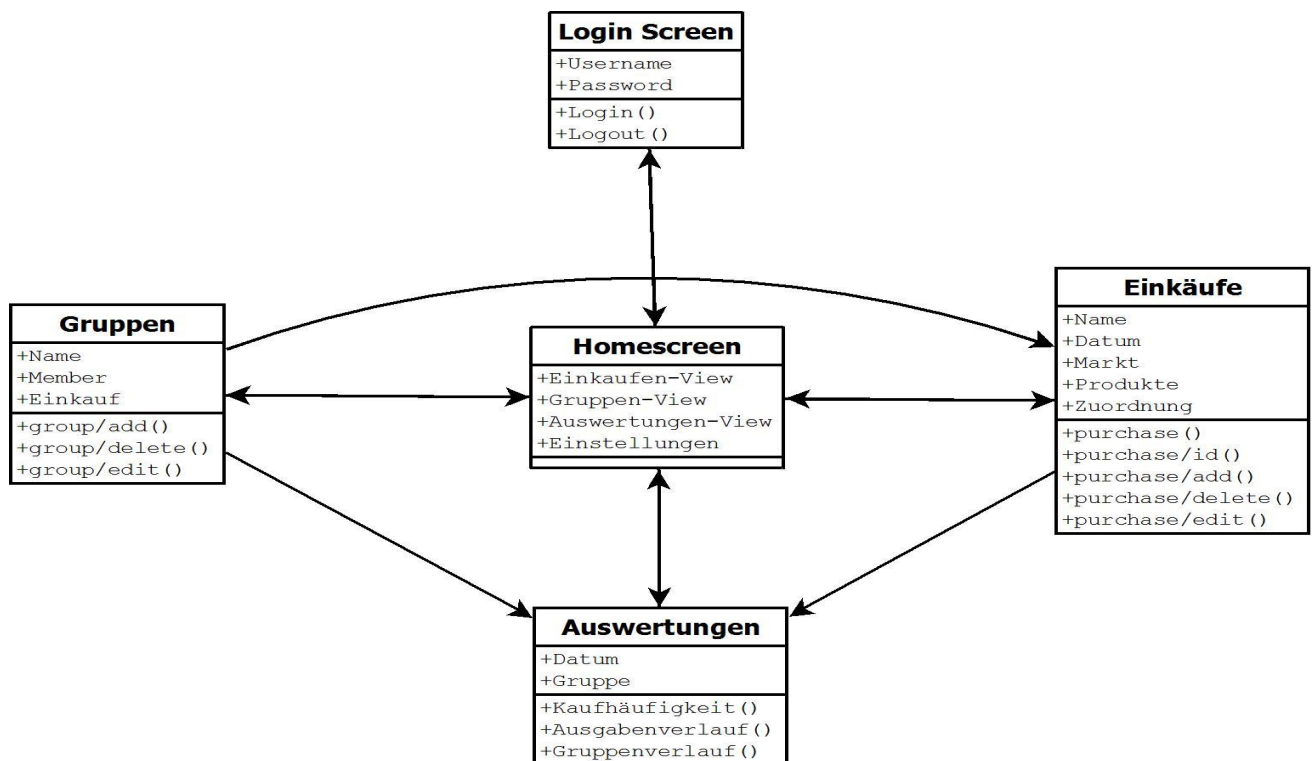


Abbildung 3: Klassendiagramm

### 2.1 Architektur

Die App selbst wurde auf Basis des Ionic Frameworks entwickelt, der Webserver wurde mittels Node.js umgesetzt und die zugrundeliegende Datenbank ist eine MongoDB. Die Entscheidung zu diesen Komponenten fand auf Grund ihres guten Zusammenspiels hinsichtlich Webapplikationen statt. Im ersten Schritt wurde die MongoDB gewählt, weil es für unsere Zwecke eine angemessene Flexibilität aufgrund der Schemalosigkeit bietet, aber auch die Skalierbarkeit einer NoSQL für den Aspekt der Zukunftssicherheit mit sich bringt. Node.js ist besonders gut geeignet, um weniger rechenintensive Aufgaben, wie das Handeln einer Anfrage und das Verweisen auf eine Ressource, auf einem Server umzusetzen. Ebenfalls wurde diese Entscheidung durch das gute Zusammenspiel mit MongoDB, nicht zuletzt durch Mongoose, bestärkt.

Das Ionic Framework wurde allem voran genutzt, weil ermöglicht, ein zeitgemäßes Desing zu erstellen und die Möglichkeit bietet, sowohl Android, als auch iOS-lauffähige Anwendungen zu erstellen, ohne dafür zusätzliche Entwicklungsarbeit zu leisten. Für detailliertere Ausführungen wird auf das Pflichtenheft verwiesen.

## 2.2 Registrierung

Um die EinkaufsApp zu nutzen, muss sich jeder Nutzer mittels einer E-Mail-Adresse und einem Kennwort für der App registrieren. Eine Registrierung ist bei dieser App unentbehrlich, da für jeden Nutzer ein Profil angelegt wird und diesem Profil innerhalb der Datenbank die Produkte und Finanzen zugeordnet werden. In den folgenden Unterpunkten wird der Prozess der Registrierung jeweils von den Designern und Entwicklern beschrieben.

### Design

Die Designer haben zu der Registrierung und zu dem Login, welcher im Punkt 3.4 behandelt wird, das folgende Flussdiagramm entworfen:

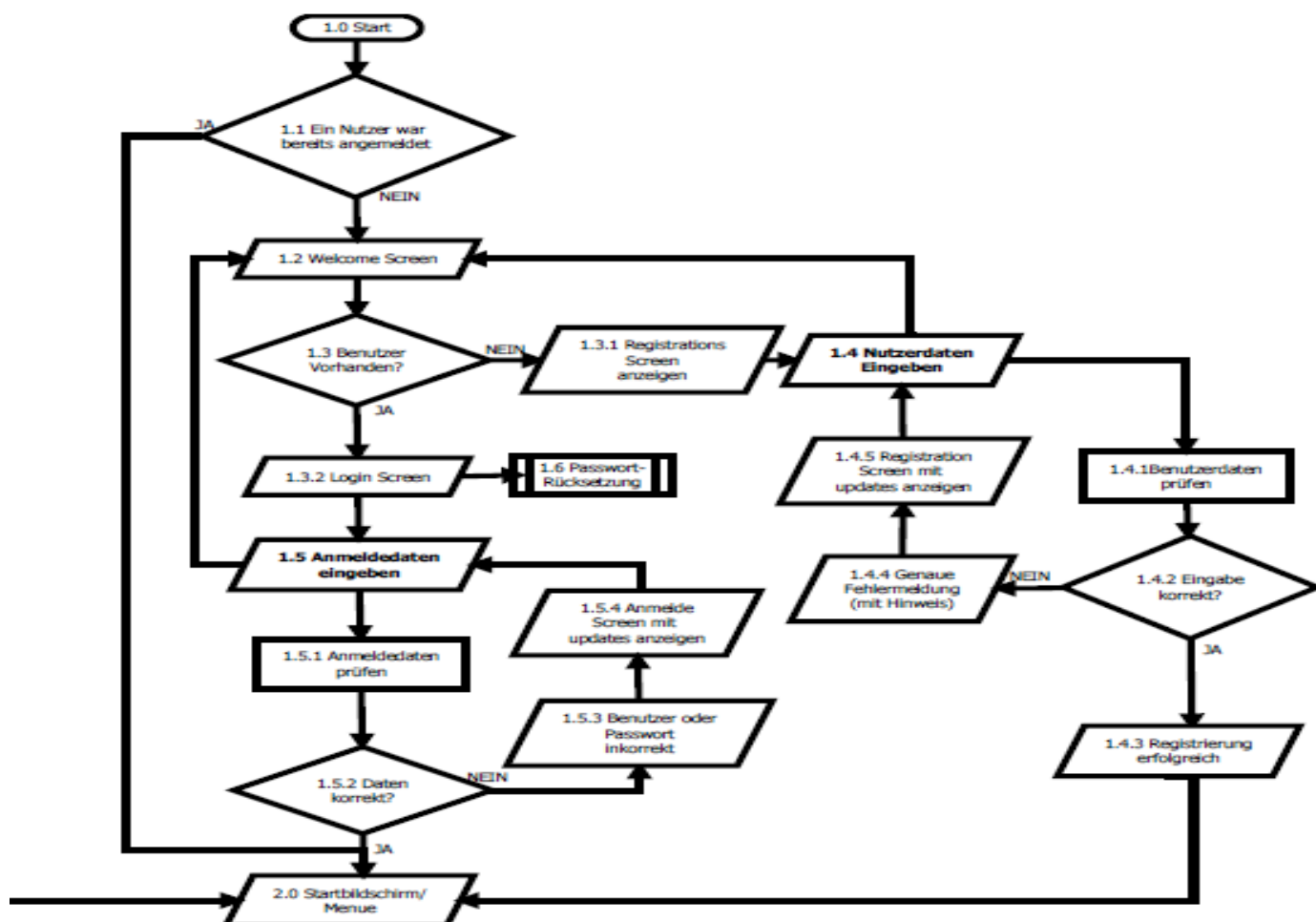


Abbildung 4: Flussdiagramm Login

Das Flussdiagramm beschreibt sukzessiv den Ablauf des Registrierungs- und des Loginvorgangs, welcher von den Entwicklern daraufhin technisch umgesetzt wurde.

## Entwicklung

Die Entwickler befassen sich mit den Funktionen der Applikation und sorgen bei der Registrierung dafür, dass alle Daten ordnungsgemäß geprüft und in die Datenbank eingepflegt werden. Als Datenbank wird für die App MongoDB genutzt, welche via RoboMongo gemanagt wird. Als Programmiersprache JavaScript.<sup>1</sup> verwendet.

Bei der Registrierung werden die einzelnen Benutzereingaben durch bestimmte Regeln in Hinblick auf Länge, E-Mail-Format, Eindeutigkeit, sowie Sicherheitskriterien bei der Passwortvergabe in der Applikation geprüft.<sup>2</sup>

Wenn alle Prüfungen erfolgreich waren, wird der Benutzer angelegt und das Passwort verschlüsselt in Form eines Hashs in der Datenbank gespeichert.

---

<sup>1</sup>Genaueres zu der Datenstruktur können Sie dem angehängten Pflichtenheft entnehmen.

<sup>2</sup>Die konkreten Regeln der Benutzereingaben können Sie in dem angehängten Handbuch entnehmen.

## **2.3 Login**

### **Design**

Das Flussdiagramm der Designer für den Login ist im Abschnitt 3.3 Registrierung zu finden.

### **Entwicklung**

Die Entwicklung beschäftigt sich mit der Funktionsweise des Logins und prüft hierbei, ob der Benutzer in der Datenbank existiert. Falls dies der Fall ist, wird das Passwort geprüft und bei korrekter Eingabe ist der Login erfolgreich durchgeführt. Wenn der Benutzer sein Passwort vergessen hat, kann er dieses zurücksetzen lassen. Hierbei bekommt er eine E-Mail an die im Userprofil hinterlegte E-Mail Adresse. Diese Funktionalität ist bereits implementiert, wird aber auf Grund der Sicherheitsbestimmungen der Testumgebung geblockt. Diese enthält ein Token womit es einen Benutzer ermöglicht wird sein Passwort zu ändern. Dieses Token ist genau eine Stunde gültig, danach verfällt es. In dem dann aufgerufenen Bildschirm muss der Nutzer nun sein Passwort zweimal eintragen. Daraufhin ändert die Datenbank das Kennwort des Nutzer und speichert dieses.

## **2.4 Marktauswahl**

Bevor der Einkaufsprozess beginnt muss der Nutzer einen Markt auswählen, hierfür wird der Standort des Nutzers ermittelt via GPS ermittelt und bereits registrierte Märkte in seiner Nähe angezeigt.

### **Design**

Die Designer haben hierzu, ebenso wie bei der Registrierung und dem Login ein Flussdiagramm erstellt:



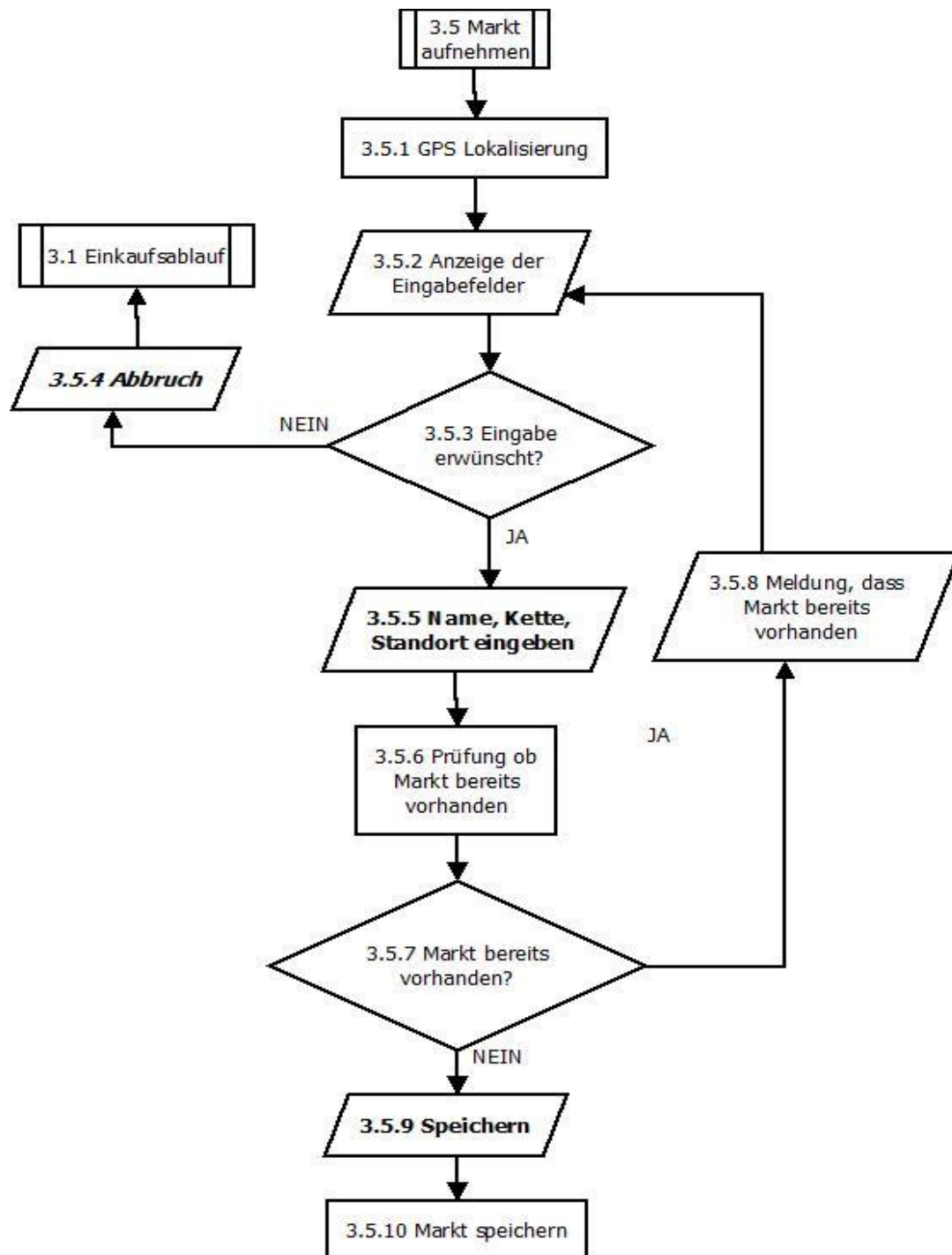


Abbildung 5: Markt-Auswahl

## Entwicklung

Der vom Design erstellte Ablauf der Marktauswahl wird direkt in den Einkaufsprozess mit eingebunden. Sobald der Nutzer die Option „Einkaufen“ wählt gelangt dieser in den

Marktauswahl-View.<sup>3</sup> Anders als im geplanten Ablauf der Designer, kann der Nutzer selber entscheiden, in welchem Markt er sich gerade befindet, indem er entweder in der direkt aufgeführten Marktliste einen auswählt, oder die Option „Markt hinzufügen“ nutzt. Dies hat technische Gründe, da zurzeit eine Lokalisierung der Märkte via GPS aus Zeitgründen noch nicht umgesetzt wurde.

---

<sup>3</sup>In der EinkaufsApp werden Views verwendet, da die Systemarchitektur auf der Model-View-Controller Architektur basiert.

## 2.5 Einkaufsverwaltung

### 2.5.1 Design

Aktivitätsdiagramm „Einkauf einlesen“

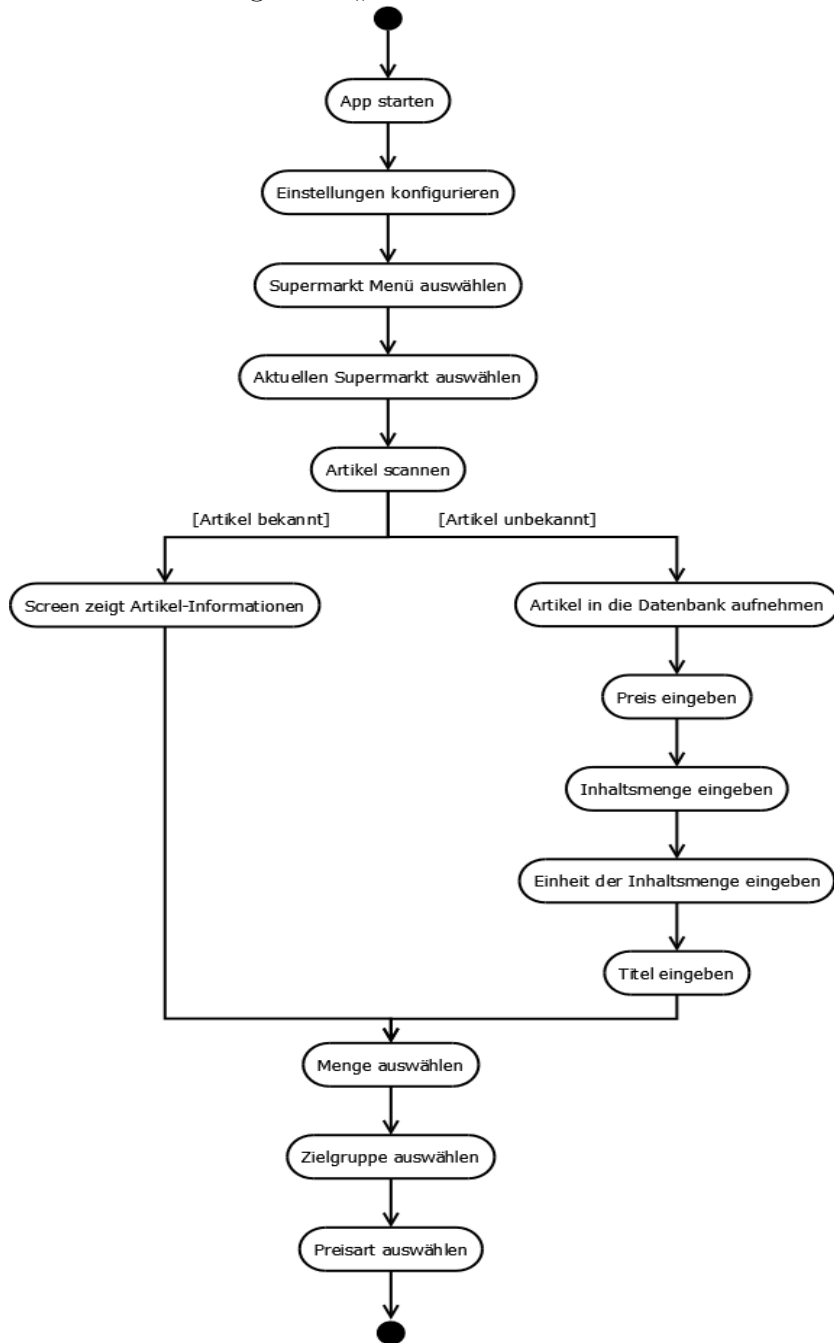


Abbildung 6: Aktivitätsdiagramm-Einkauf

### Entwicklung

## 2.6 Nutzerverwaltung

Die Nutzerverwaltung ermöglicht dem Nutzer die individuelle Zuweisung von anderen Accounts als Gruppenmitglieder zu bereits erstellten Gruppe. Ziel dessen ist die vereinfachte Finanzverwaltung während des Einkaufes, sowie die Möglichkeit komplexere Auswertung durchzuführen.

### Design

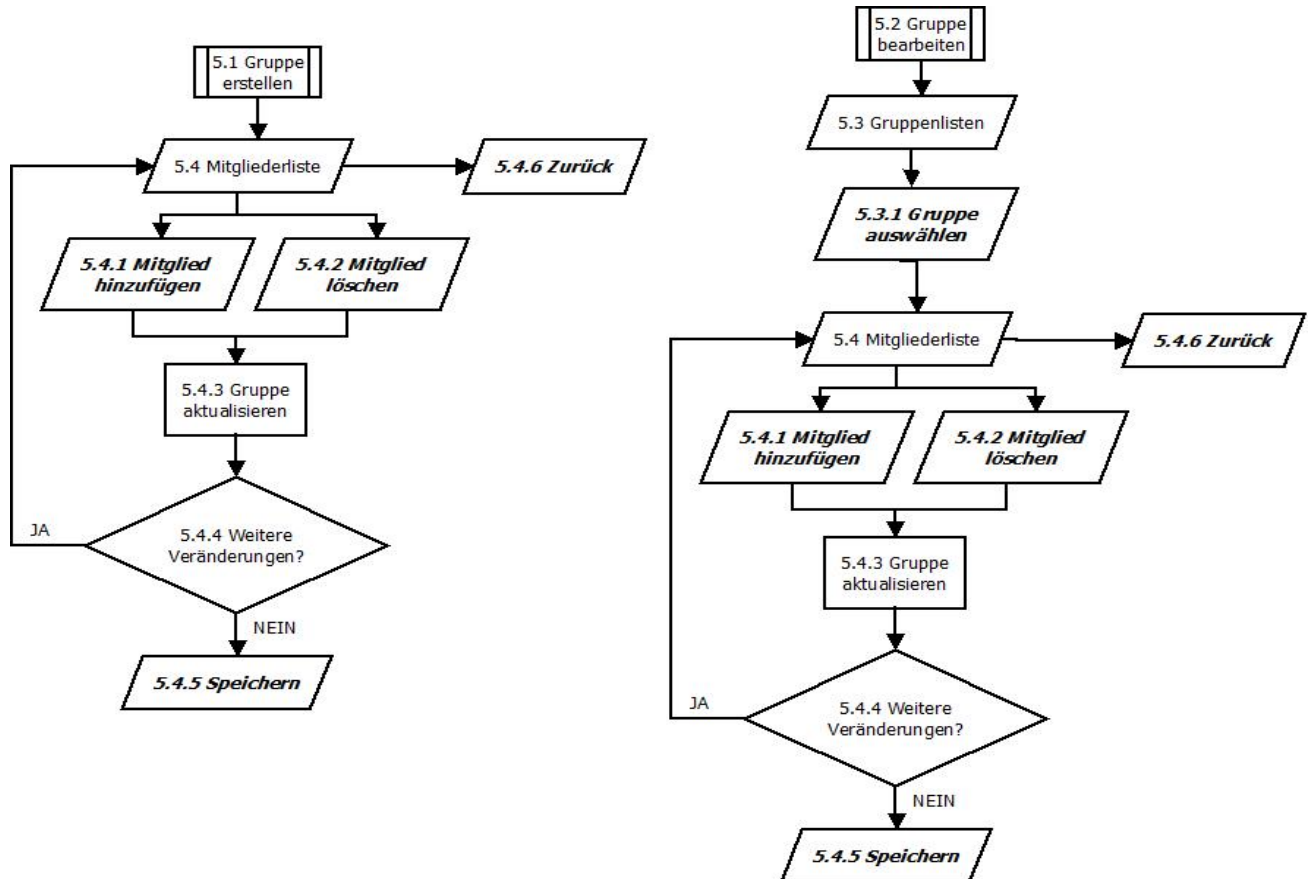


Abbildung 7: Gruppenverwaltung

Die Gruppenverwaltung wird in zwei Teile unterteilt. Der Erste beschreibt die Gruppenerstellung. Wie im Flussdiagramm zu sehen, kann eine Gruppe angelegt werden, indem zunächst der Nutzer die Menüoption Mitgliederliste wählt und hier eine Gruppe erstellt. Dabei können Mitglieder hinzugefügt bzw. auch gelöscht werden.

### Entwicklung

Die Entwicklung hat lediglich den Teil der eigentlichen Gruppenerstellung integriert. Der Nutzer legt einen Gruppennamen fest und kann daraufhin die registrierten Nutzer hinzufügen und ggf. wieder entfernen. Ist die Gruppe erstellt, kann gespeichert werden. Diese kann im Nachhinein noch nicht geändert werden, weil die Gruppe nicht angezeigt wird. Dies ist ein Problem, welches die Entwickler noch ausbessern werden.

## 2.7 Auswertung

Der Nutzer hat die Möglichkeit vergangene Einkäufe auswerten zu lassen. Es gibt die zeitliche Eingrenzung und eine Gruppenmitgliedereingrenzung, bei der alle vergangenen Einkäufe von bestimmten Gruppen zusammengefasst werden.

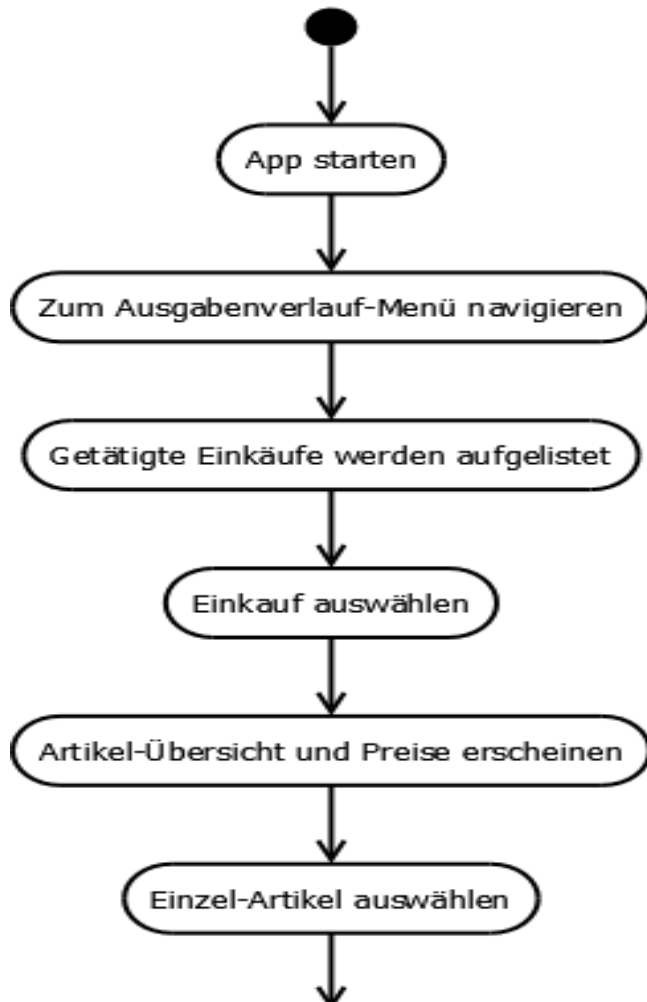


Abbildung 8: Aktivitätsdiagramm-Ausgabenverlauf

## Zustandsdiagramm

„Ausgabenverlauf anzeigen“

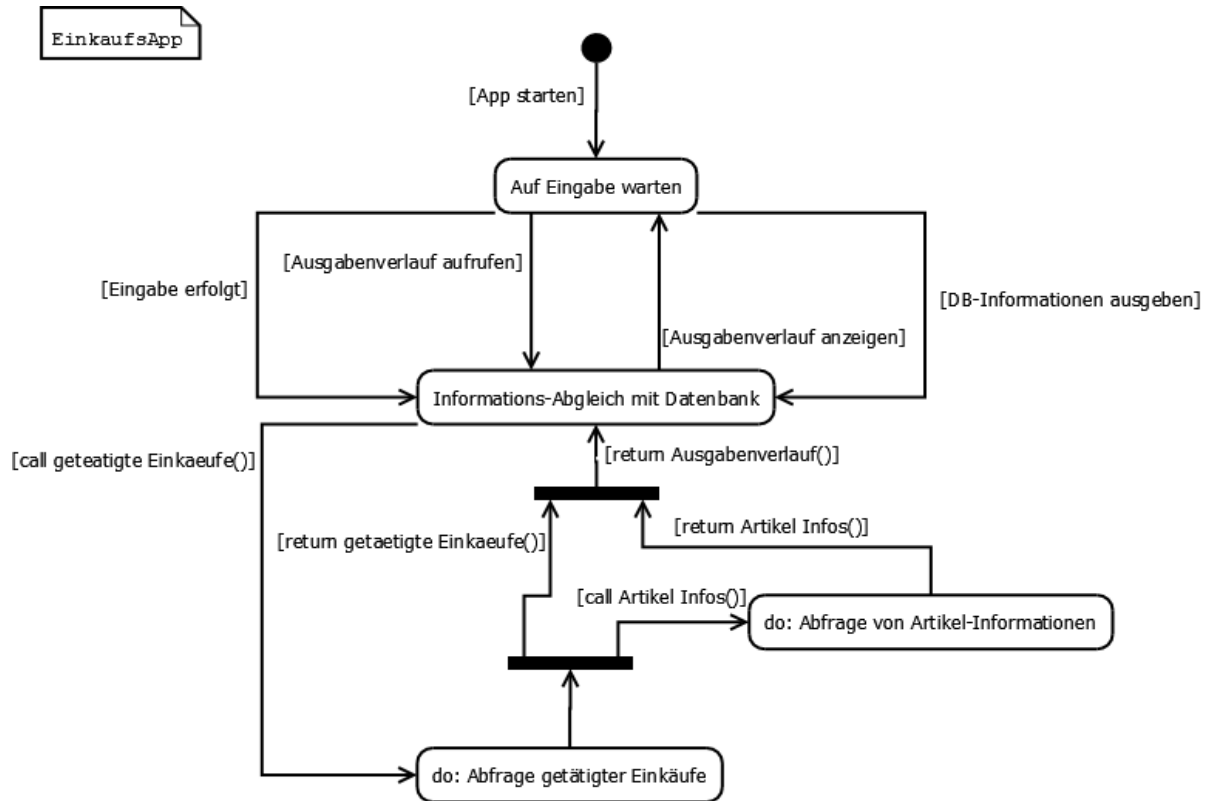


Abbildung 9: Zustandsdiagramm

## Design

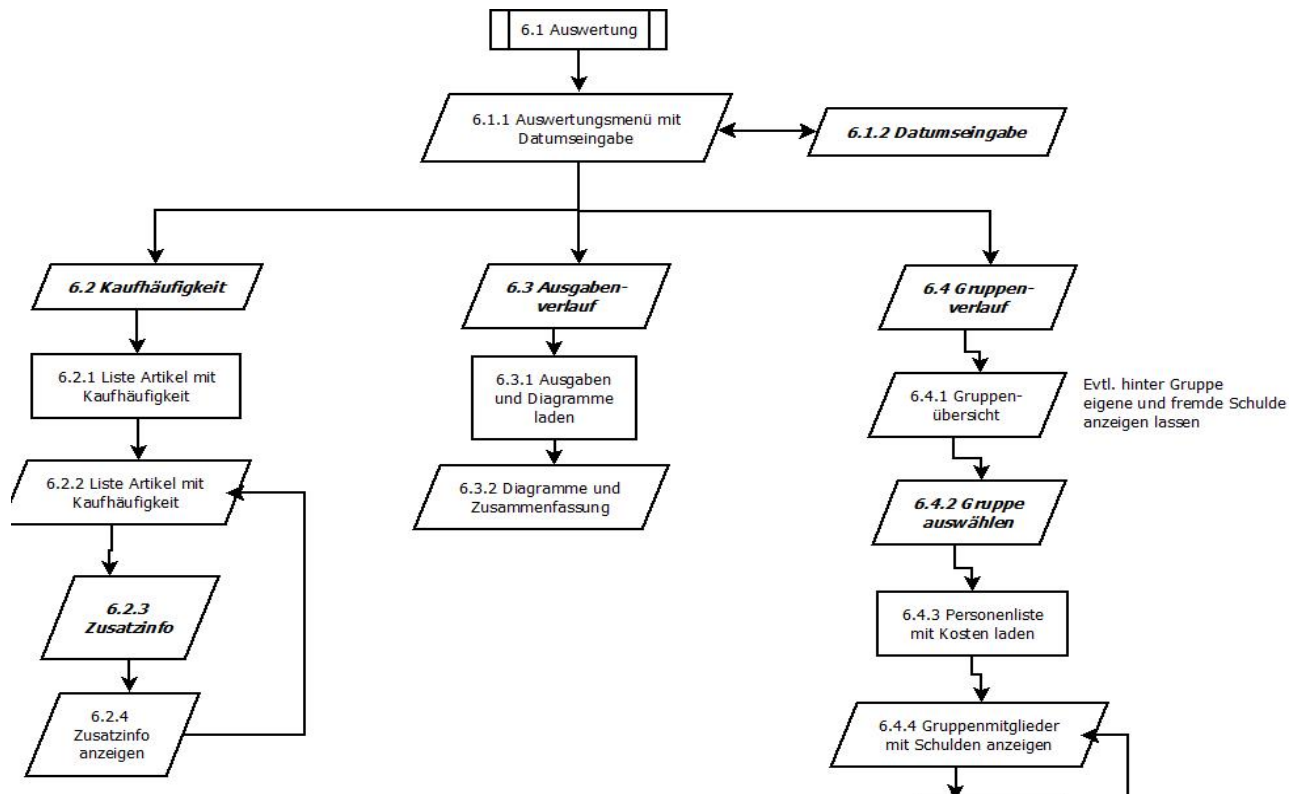


Abbildung 10: Auswertung

## Entwicklung

Wie im Programmablaufplan beschrieben, gibt es im View „Auswertung“ die drei Optionen Kaufhäufigkeit, Ausgabenverlauf und Gruppenverlauf. In jedem dieser Optionen muss ein Start- und Enddatum vom Nutzer festgelegt werden. Je nachdem welche Option ausgewählt wurde, sind die vom System gebende Informationen unterschiedlich. Da es aber weiterhin noch Probleme mit dem Einkaufsprozess gibt, kann die Auswertungsfunktion noch nicht genutzt werden.

## **3 Problemzusammenfassung**

### **3.1 Usability der App**

Zusammengefasst kann die Version 1.0 der EinkaufsApp Nutzer registrieren und anmelden, sowie den Einkaufsprozess durchführen. Außerdem können Gruppen erstellt und verwaltet werden. Der Auswertungsprozess wurde in Anbetracht der Zeit nicht umgesetzt. Hinsichtlich der Optik wurde kein Corporate Design verwendet, um die Benutzerfreundlichkeit dennoch zu gewährleisten, implementierten die Entwickler ein übersichtliches Interface im aktuellen Web Stil.



## 3.2 Organisation und Projektmanagement

Allem voran muss erwähnt werden, dass dieses Projekt enorm zeitkritisch war und daher die Planungsphase nahe zu vollständig übersprungen werden musste. Ebenfalls sind Design und Implementation stark parallelisiert abgelaufen. Um innerhalb von zwei Monaten ein lauffähiges und qualitätsorientiertes Software-Produkt zu erstellen, bedarf es eines überschaubaren und für alle Mitglieder transparenten Projektes, aber allem voran auch ein, gezielt auf die Herausforderungen, zugeschnittenes Team, welches die benötigten Skills und das Know-How bereits mitbringt. Ebenfalls ist es sehr wertvoll, wenn die Beteiligten bereits zuvor zusammen gearbeitet haben und untereinander mit Tools und Konventionen vertraut sind, oder zumindest in der Lage sind unter formalen Bedingungen zu arbeiten. Da naturgemäß in unserem Umfeld diese Voraussetzungen nur zu sehr geringen Teilen erfüllt werden können, ist auch ein sehr kritischer Projektablauf unausweichlich. Um diesen Schwierigkeiten entgegen zu wirken, wäre eine klarer Abgrenzung von Aufgabenbereichen und Skills und auch das Vertrauen in diese Entscheidungen notwendig gewesen. Unglücklicherweise fehlte auch hier wieder die bereits erwähnte Planungsphase, die aber auch aus Zeit- und Lokalisierungsgründen nicht umsetzbar war. Im weiteren Verlauf wurde dann eine zusätzliche Organisationseinheit zwischen dem Gesamtprojekt und den einzelnen Entwicklern eingefügt, um einen besseren Überblick zu gewährleisten und schneller Entscheidungen treffen zu können. An dieser Stelle lässt sich auch die obligatorische Diffizilität, die aus den unterschiedlichen Engagements entsteht erwähnen. Nach den vorangegangenen Ausführungen zu den kritischen Komponenten des Projektes sollte selbstredend sein, dass es ohne überdurchschnittlichen Einsatz jedes einzelnen Beteiligten unmöglich zu 100% in gegebener Zeit mit derartig begrenzten Ressourcen abgeschlossen werden kann. Nun sind naturgemäß die intrinsischen Beweggründe, sowie die Möglichkeiten der einzelnen Personen verschieden. Somit verbleibt die Aufgabe aller Beteiligten im Rahmen der gegebenen Bedingungen das bestmöglich Ergebnis zu erzielen.

## 4 Projektabschluss

### 4.1 Fertiges Produkt

### 4.2 Aussichten

Anforderungen an die App	Umgesetzte Funktionen
Produkte aus gepflegter Datenbank auswählen und für eigenen Einkauf nutzen	Datenbank muss manuell eingepflegt werden Jeder Nutzer sieht Produkte in Datenbank Um Datenbank zu füllen: EAN einscannen und Produktname und Preis dazu eingpflegen
Bestimmte Artikel, welche sich in einem bestimmten Markt im Angebot befinden, anzeigen	Ebenso manuelle Einpflegung von Angebotspreis und Angebotszeitraum für Produkt notwendig
Sonstige Kosten aufzunehmen, die nicht mit einem EAN Code in Verbindung gebracht werden (z.B. Miete, Strom-Kosten)	Noch nicht implementiert
Möglichkeit für jemand anderen oder Gruppe (z.B. WG) einzukaufen	Jedoch noch nicht möglich dass andere Gruppenmitglieder bereits eingekaufte Artikel sehen wenn Einkauf nicht abgeschlossen ist
Daten werden zentral in einer, über das Internet erreichbare, Datenbank gespeichert	Umgesetzt in MongoDB
Die App soll primär ein einfaches Front End bereitstellen, um Informationen zu sammeln und zu verwalten	Umgesetzt in gesamter EinkaufsApp
Möglichkeiten der Auswertung sind vielfältig und können in Listen oder Diagrammen umgesetzt werden	Umgesetzt durch 3 Auswertungsmöglichkeiten: Häufigkeit, Ausgabenverlauf, Gruppenverlauf
Außerdem nicht personenbezogene Auswertungen: a) über beliebteste Artikel b) beliebteste Märkte c) Durchschnittspreis	Noch nicht implementiert
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	Umgesetzt in EinkaufsApp
Verbindung zur Datenbank via JavaScript	Umgesetzt in EinkaufsApp
Aus der Android-App heraus wird mittels HTTP-Methoden(GET,POST,...) eine Anfrage an den Webserver geschickt und die Antwort im JSON-Format wieder an die App zurück geschickt.	Umgesetzt in EinkaufsApp
Funktion des Preisvergleichs in unterschiedlichen Märkten	Noch nicht implementiert
Funktion der Erstellung von Einkaufslisten vor dem Einkauf Artikelaufnahme in diese Liste Nachfolgende Bearbeitung dieser Liste	Noch nicht implementiert
Out of Scope: Anmelden via Facebook oder Twitter	Out of Scope, aber kann in späteren Versionen eingesetzt werden
Out of Scope: WebSite als Ergänzung für die Ausgabe von Statistiken und die Benutzerverwaltung	Out of Scope, doch im Internet kann man Website aufrufen --> aber Ausgabe wäre dort sinnvoll

Tabelle 3: Aussichten

## 5 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](http://www.gepir.de/v31/V31_client/gtin.aspx)

# Organisationstools

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

# Anhang

- Anhang [1]: Pflichtenheft
- Anhang [2]: Handbuch
- Anhang [3]: Installationsanleitung
- Anhang [4]: EER-Diagramm
- Anhang [5]: Proto.io Design
- Anhang [6]: Prototyp
- Anhang [7]: MVP
- Anhang [8]: Main PAP
- Anhang [9]: Datenfluss
- Anhang [10]: ER Modell
- Anhang [11]: MongoDB-Klassendiagramm
- Anhang [12]: Factories