

Software Engineering Projekt

Dual Wirtschaftsinformatik Jahrgang 2013
EinkaufsApp

04. Januar 2015

Inhaltsverzeichnis

Einleitung	7
1 Vorbetrachtung	8
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
2 Durchführungsphase	19
2.1 Architektur	19
2.2 Registrierung	20
2.3 Login	23
2.4 Marktauswahl	24
2.5 Einkaufsverwaltung	27
2.6 Nutzerverwaltung	29
2.7 Auswertung	30
3 Problemzusammenfassung	35
3.1 Usability der App	35
3.2 Besondere Herausforderungen des Projektes	36
4 Projektabschluss	37
4.1 Aussichten	37
5 Lesson learned	38
Quellen	39
Anhang	41

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 : Ist- Analyse

Tabelle 3: Meilensteinplanung

Tabelle 4: Auswertung-Diagramme

Tabelle 5: Aussichten

Abbildungsverzeichnis

Abbildung 01: Include-Beziehung

Abbildung 02: Extend-Beziehung

Abbildung 03: ER - Modell

Abbildung 04: Aktivitätsliste

Abbildung 05: Klassen-Diagramm

Abbildung 06: Flussdiagramm-Login

Abbildung 07: Marktauswahl

Abbildung 08: Einzeleinkauf

Abbildung 09: Gruppenverwaltung

Abbildung 10: Aktivitätsdiagramm Auswertung

Abbildung 11: Zustandsdiagramm

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, Durchführungsphase, 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 Angebot 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 Trackingtool dienen. Funktioniert dieses einwandfrei, kann eine weitere Funktionalität implementiert werden, nämlich der automatischen Einkaufslistengenerierung. Diese soll nun vergangene Einkäufe auswerten und anhand dieser Vorschläge für 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 Name innerhalb der Gruppe
- 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

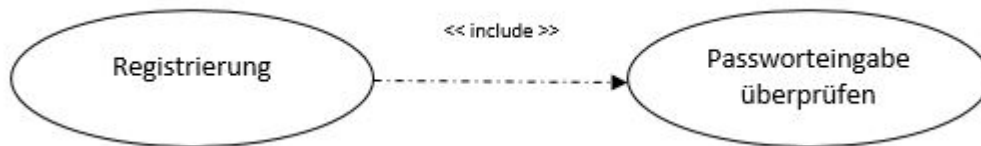


Abbildung 1: Include-Beziehung

Extend-Beziehung:



Abbildung 2: 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.

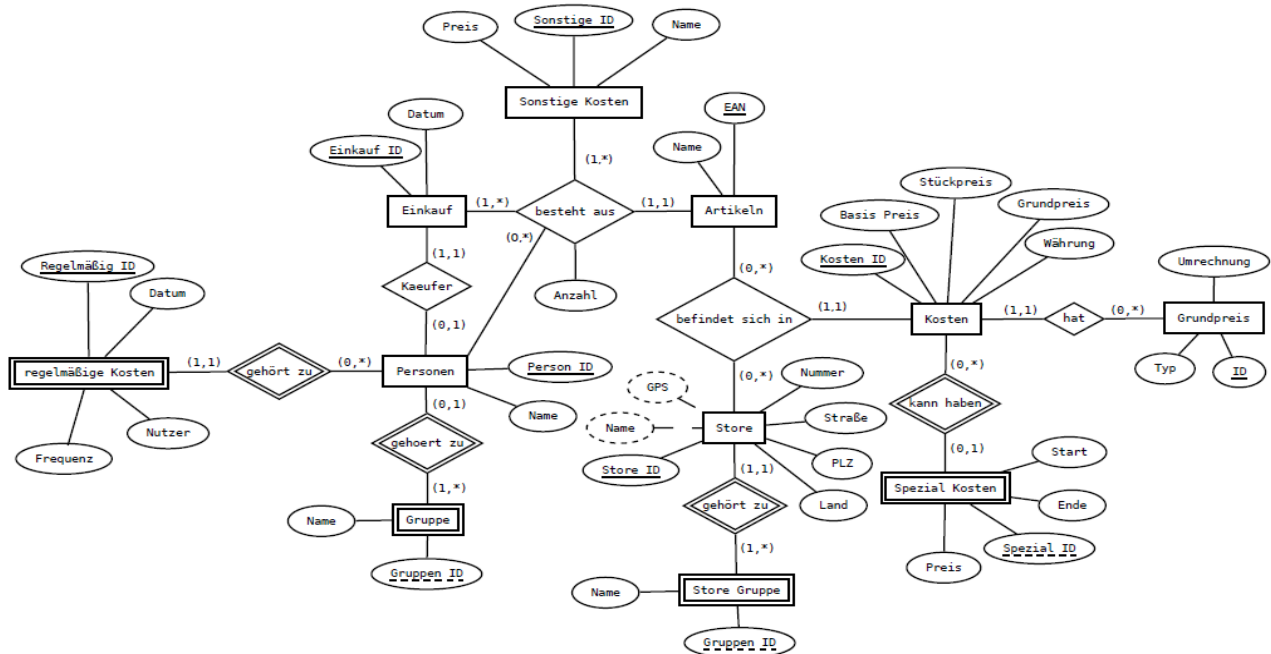


Abbildung 3: ER - Modell

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	
Thomas Elias	gute Kenntnisse in Projektkoordination	Aufgrund seiner guten Kenntnisse im Bereich Projektkoordination wird Thomas Elias in der Dokumentation arbeiten und für die Meilensteinplanung verantwortlich sein.
	Anforderungen und Arbeitspakete definieren	
	Erfahrungen Customizing von Dokumenten-Layouts	
	Kommunikation zwischen versch. Abteilungen zum Transparent-Machen der Informationen	
Victor Fuchs	gute Excelkenntnisse	Victor Fuchs wird in der Gruppe Entwicklung arbeiten und hierbei die benötigten Konzeptdiagramme erstellen
	gute Kenntnisse im Bereich Rechnungswesen-Controlling	
Sebastian Kiepsch	1 Jahr T-Systems, Expert Services	Sebastian Kiepsch wird wegen seiner ausgeprägten Programmiererfahrung bei der Entwicklung tätig sein.
	1 Jahr T-Systems, Digital Division	
	Programmiererfahrung (Javascript, HTML, CSS, C#)	
	Frameworks: Node.JS, Express.JS, Ionic	
Florian Graupeter	Application Management bei T-Systems BTG 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 4: Aktivitätsliste

Meilensteinplanung

Mithilfe der erstellten Meilensteinplanung wurde das pünktliche Erreichen der Projektziele geplant. Jedes Team konnte sich an dieser Tabelle orientieren und seine Aufgaben

der Reihe nach ordnen und ausführen.

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

Tabelle 3: 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:

Einige konkrete Sicherheitsvorkehrungen lauten:

- 1) Eine Session läuft nach 10 Minuten ohne Request an den Server ab.
- 2) Jedes Passwort wird als Einweg-Hash in der DB gespeichert.
- 3) Es wird eine Authentifizierung durchgeführt, das heißt ein Sicherstellen, dass die beteiligten Parteien auch tatsächlich die sind, die sie vorgeben zu sein.
- 4) Data Security: Eine weitere wichtige Technik zur Sicherheit von Webanwendungen spielt das SSL-Protokoll (Secure Socket Layer). Bei Verwendung dieser Protokolle wird vor der eigentlichen HTTP-Kommunikation ein geschützter Kanal aufgebaut, sodass die Nutzerdaten für Dritte nicht zugänglich sind. Die Nutzerdaten werden in einem normalen Web-Formular eingetragen (Login-Screen) und mittels POST-Request an den Server gesendet. Da der TCP-Kanal abgesichert ist, haben Dritte keinen Zugriff auf die Nutzdaten innerhalb des POST-Requests, d.h. Nutzernamen und Passwort werden gesichert übertragen.

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. Generell wurde in der Entwicklung nach dem Modell-View-Controller Prinzip gearbeitet. Nachdem die Prozesslogik der Designer stand („Modell“), wurden die einzelnen Views, also die visuelle Darstellung, der zu implementierten Funktion, die der Nutzer am Ende sieht, entwickelt und welche schlussendlich über die „Controller“ mit der Logik versehen wurden. Jedes einzelne View hat dabei seine eigene Route.

Einleitung

Die genannten Hauptprozesse stehen, wie in dem folgenden Diagramm zu sehen ist, in Relation. Dabei wurden die Views¹ als einzelne Klassen dargestellt:

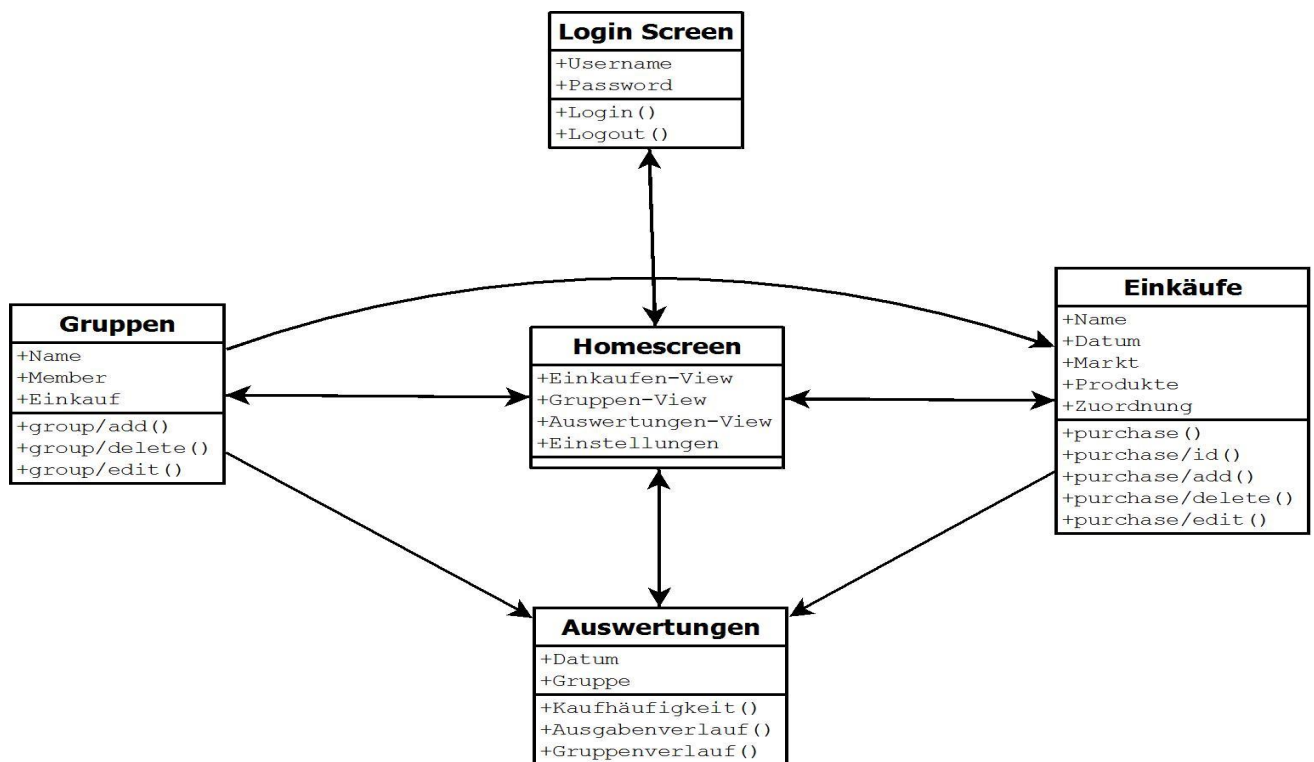


Abbildung 5: Klassendiagramm

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.

¹In der EinkaufsApp werden Views verwendet, da die Systemarchitektur auf der Model-View-Controller Architektur basiert.

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 und 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:

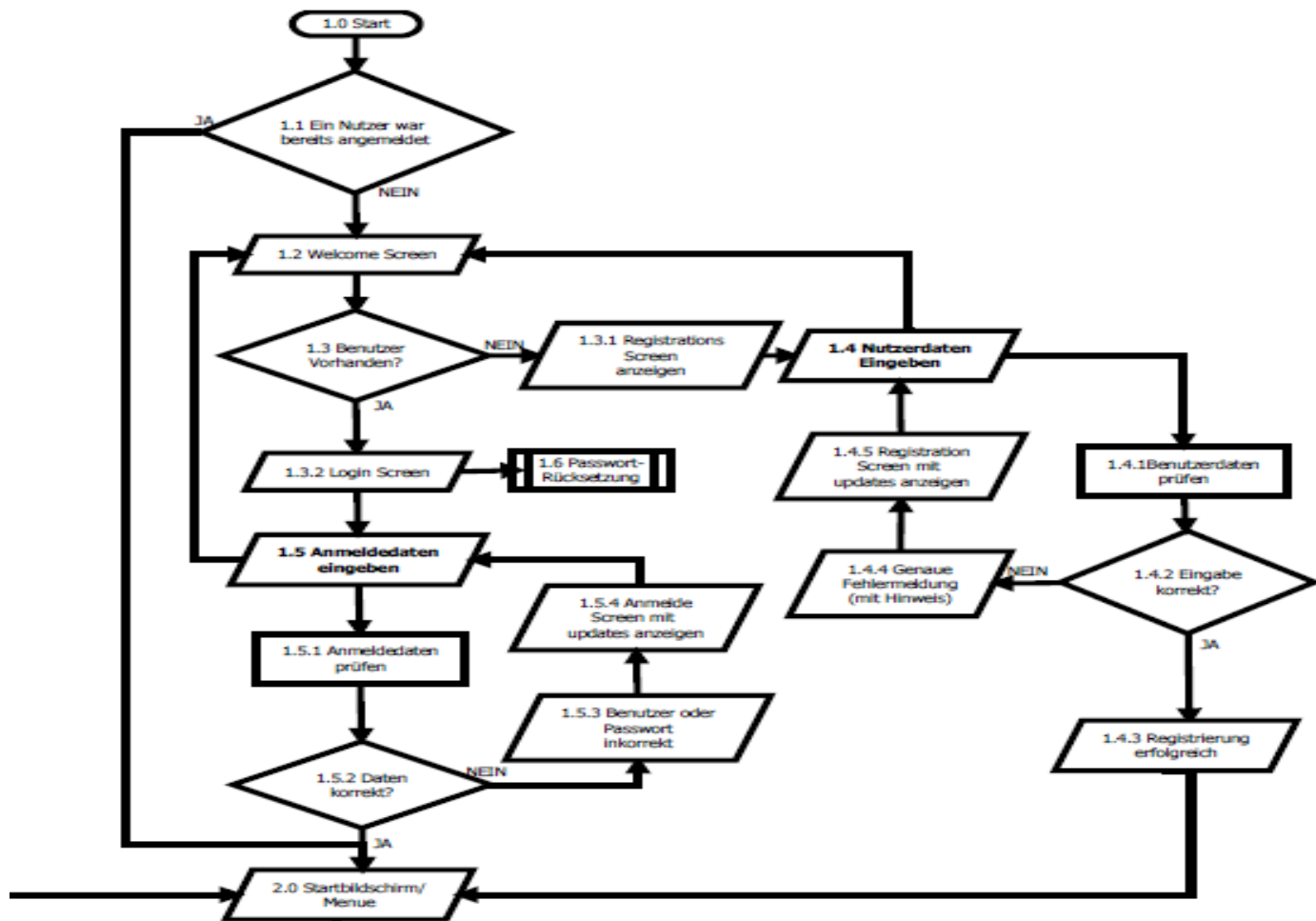


Abbildung 6: Flussdiagramm Login

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

Entwicklung

Die Entwickler befassten sich mit den Funktionen der Applikation und sorgten bei der Registrierung dafür, dass alle Daten ordnungsgemäß geprüft und in die Datenbank eingepflegt werden.² 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.³

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

²Genaueres zu der Datenstruktur können Sie dem angehängten Angebotsdokument entnehmen.

³Die 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:

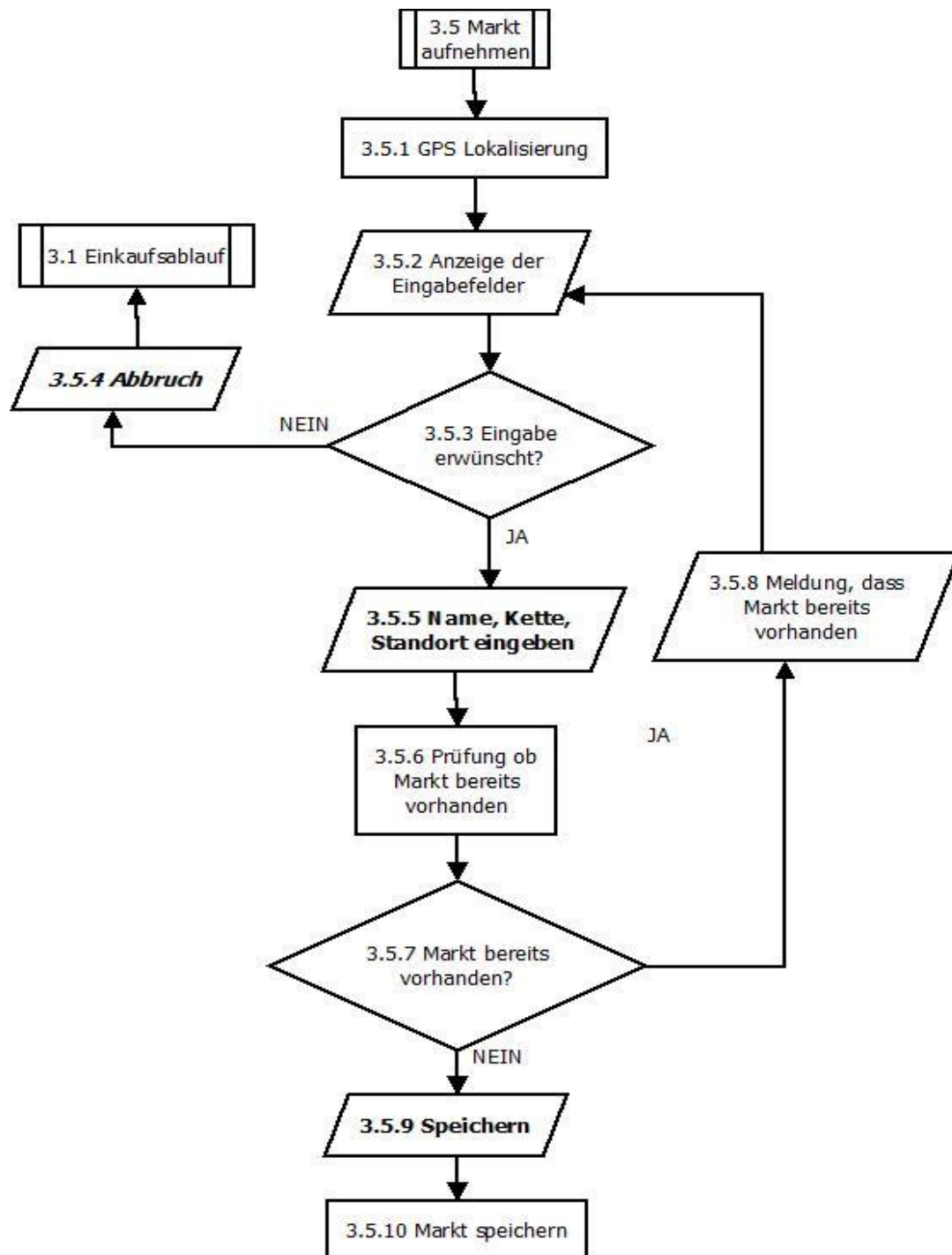


Abbildung 7: Markt-Auswahl

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. 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.

2.5 Einkaufsverwaltung

Design

Aktivitätsdiagramm „Einkauf einlesen“

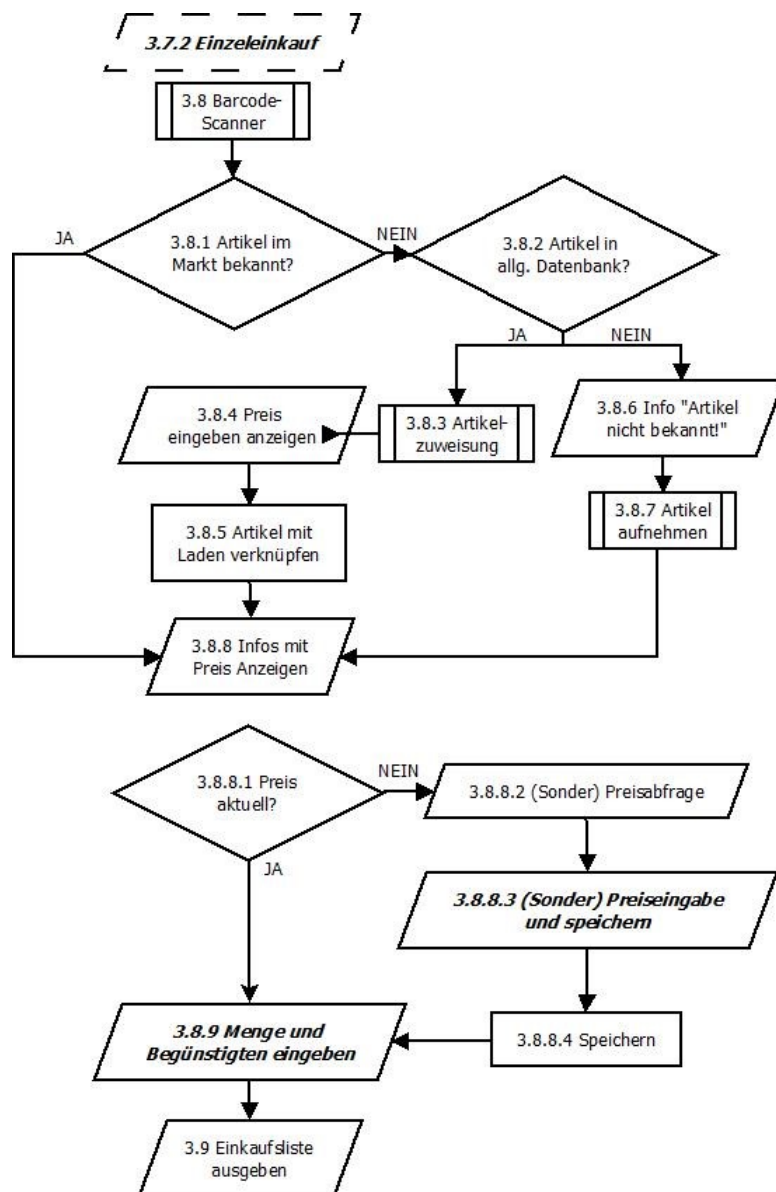


Abbildung 8: Flussdiagramm-Einzeleinkauf

Das Design Team hat ein Flussdiagramm zur Einkaufsverwaltung erstellt, das den Ausbau der Systemarchitektur berücksichtigt und als Orientierung für die Entwickler gilt. Zwecks Übersichtlichkeit ist nur ein Teil des Flussdiagramms enthalten⁴.

Gleichzeitig wird ein Prototyp in Proto.io entwickelt, mit der Zielsetzung, die Hauptfunktionalität der Anwendung abzubilden. Das Tool gibt bereits im frühen Projektstadium, vor der eigentlichen Entwicklung, ein Gefühl für das Endprodukt.

Insbesondere bei der Umsetzung der Designvorgaben durch die Entwicklung ist es eine große Herausforderung sich untereinander abzustimmen und speziell die Umsetzbarkeit

⁴Anm. d. Autors: Das vollständige Flussdiagramm befindet sich im Anhang

von Designvorgaben zu prüfen.

Entwicklung

Da das Projekt enorm zeitkritisch war sind Design und Implementation stark parallelisiert abgelaufen. Um innerhalb von zwei Monaten ein lauffähiges und qualitätsgesichertes Software Produkt zu erstellen bedarf es eines überschaubaren und für alle Mitglieder transparenten Projektes.

Im Verlauf des Projektes wurde eine zusätzliche Organisationseinheit zwischen dem Gesamtprojekt und den einzelnen Entwicklern eingefügt, um einen besseren Überblick zu gewährleisten und schneller Entscheidungen treffen zu können.

Durch eine klare Aufgabentrennung, die anhand der einzelnen Module der EinkaufsApp abgegrenzt wurde, wurden die verschiedenen Kenntnisse der Entwickler berücksichtigt. Zum Schluss mussten schließlich die Schnittstellen angepasst und die Kompatibilität der einzelnen Module geprüft werden

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

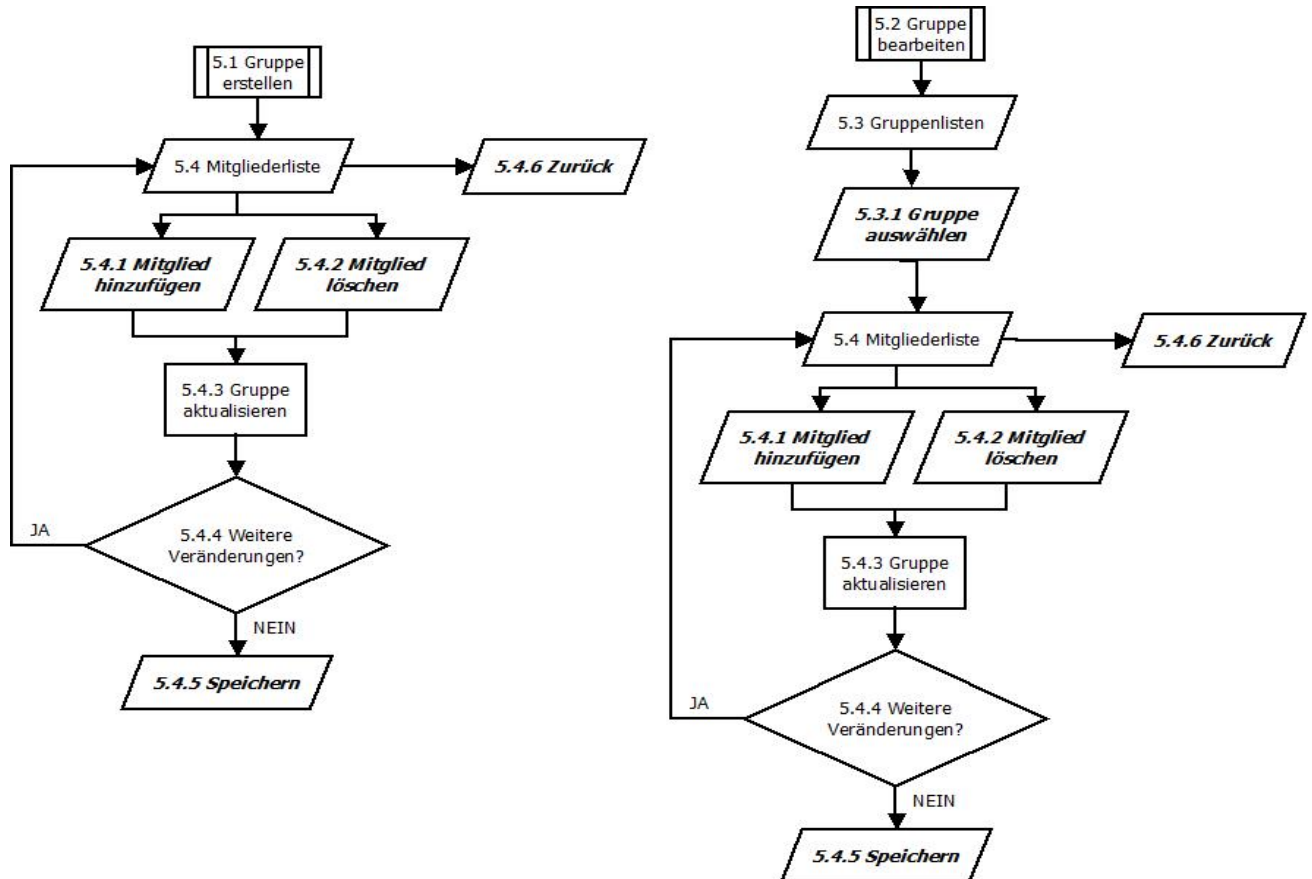


Abbildung 9: Gruppenverwaltung

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.

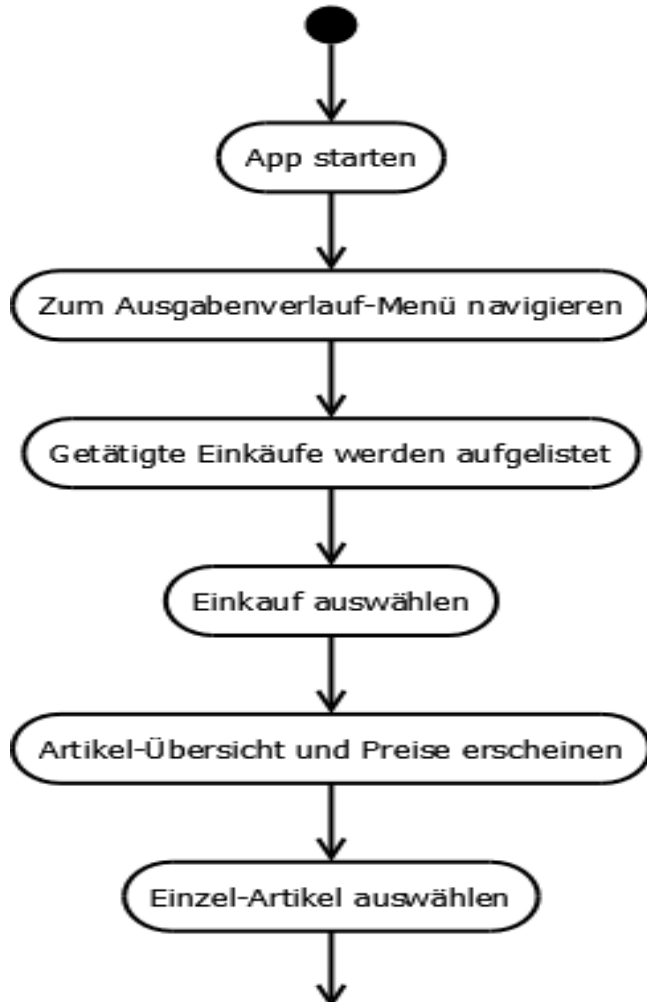


Abbildung 10: Aktivitätsdiagramm-Ausgabenverlauf

Zustandsdiagramm

„Ausgabenverlauf anzeigen“

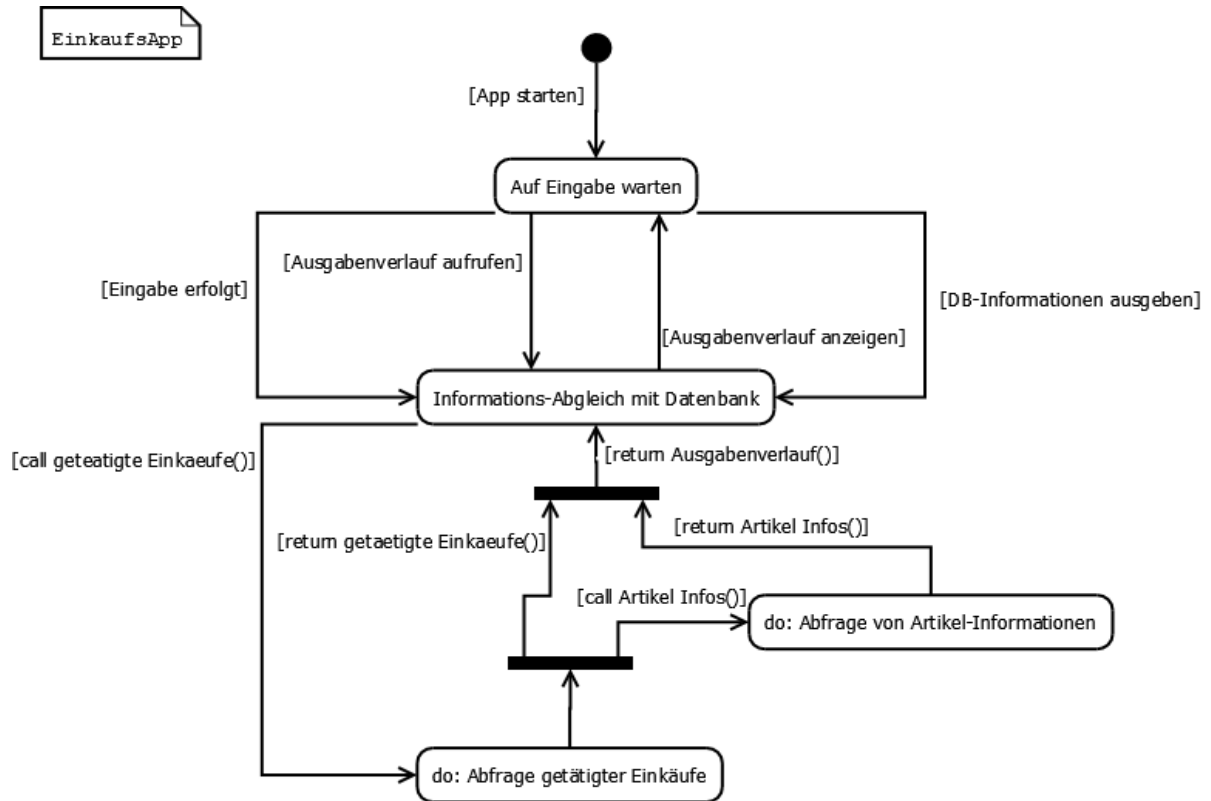


Abbildung 9: Zustandsdiagramm

Design

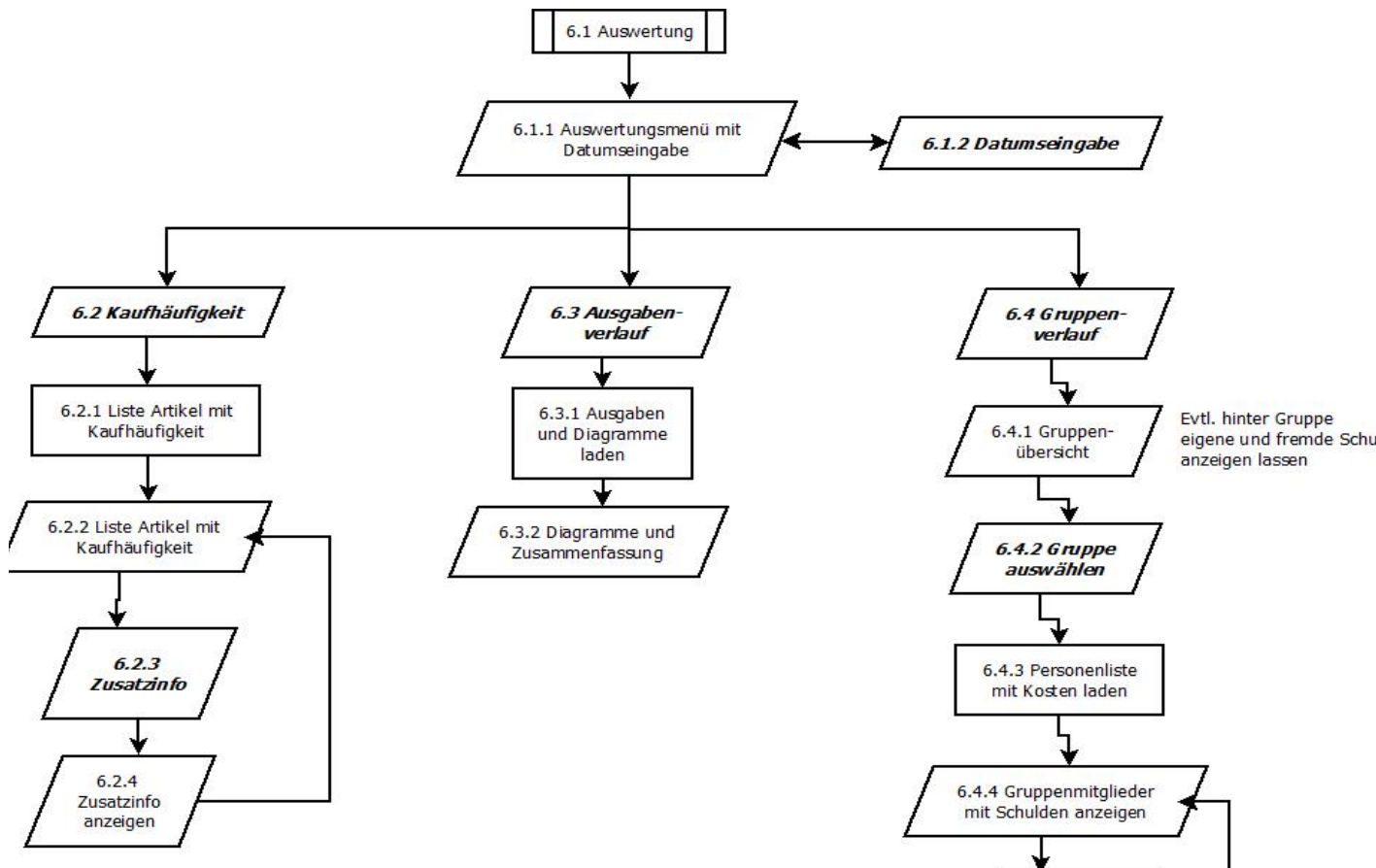


Tabelle 4: Auswertung

Auswertung der Diagramme

In diesem Abschnitt wird noch einmal genauer auf die UML-Diagramme⁵ eingegangen, welche mit dem Enterprise Architect erstellt wurden. Der Enterprise Architect eignet sich besonders gut zum Modellieren des Softwaresystems, da das Programm nach UML Standards arbeitet. Diese Funktionalität war besonders hilfreich beim Erstellen der UML-Diagramme.

Zudem bietet das Tool Vorlagen und Objekte für viele verschiedenen UML-Diagramme an und ist einfach zu bedienen. Gerade bei der ersten Erstellung eines UML-Diagramms bietet das Programm viele Hilfestellungen und ist optimal auf die Erstellung solcher Diagramme ausgelegt. Außerdem ist das Programm für den Anwender individuell anpassbar, um auch speziellen Anforderungen gerecht zu werden.

⁵Die gesamten UML-Diagramme befinden sich in dem Anhang der Dokumentation und werden vereinzelt in den Unterkapiteln schon dargestellt.

In der nachfolgenden Tabelle sind die Eigenschaften der in dem Projekt genutzten Diagramme beschrieben und gegenübergestellt:

	Eigenschaften	Was ist dargestellt?	fehlende Aspekte
Aktivitätsdiagramm (Activity-Diagramm)	Beschreibung zeitliche Abläufe, Reihenfolge der Abläufe	Aktionen Beziehung der Aktionen	Beschreibt keinen Akteur
Anwendungsfalldiagramm (Use-Case-Diagramm)	Beschreibung Funktionalitäten (zu erbringende Dienste) aus Anwendersicht Bieten Gesamtüberblick über Systemfunktionen	Actors, Pre-/Post- Condition, Mainflow, Alternative Flow, Exceptinal flow	Beschreibt kein Verhalten, keine Abläufe, keine Reihenfolge
Klassendiagramm	Beschreibt Klassen und Beziehungen zueinander	Klassenname, Attribute, Operationen	Beschreibt keine Abläufe, keine Reihenfolge
Sequenzdiagramm	Beschreibung Verhalten und Interaktionen, Nachrichtenaustausch zwischen Ausprägungen	Akteur, Aktionen	Beschreibt kein Verhalten
Zustandsdiagramm (State Machine)	Zeigt Folge von Zuständen, die Objekt einnehmen kann, Ursachen der Zustandsänderungen	Objekt, Zustand	/

Tabelle 4: Auswertung-Diagramme

Das Sequenzdiagramm muss innerhalb des Designprozesses konkretisiert dargestellt werden, damit die Entwickler optimal implementieren können.

Entwicklung

Wie im Programmablaufplan beschrieben, gibt es im View „Auswertung“ die drei Optionen Kaufhäufigkeit, Ausgabenverlauf und Gruppenverlauf.

Kaufhäufigkeit

Die Funktionalität für diese Auswertung ist im Controller „PurchaseQuantityCtrl“ hinterlegt. Zuerst wird der Username abgefragt und es werden alle für die Auswertung relevanten Variablen auf 0 gesetzt. Anschließend wird ein Array mit den Artikeln angelegt und es wird nach der Kaufhäufigkeit sortiert. Es wird geprüft ob die UserID gleich der EinkaufsID ist und ob die Einkäufe im Zeitraum liegen. Wenn ja wird „Anzahl Einkäufe“ inkrementiert. Der gesamte Einkauf wird durchlaufen und die Gesamtausgaben aufaddiert. Falls ein Artikel schon im Array ist, wird die Gesamtanzahl aufgerechnet. Wenn alle Artikel erfasst sind werden dann die Preise ausgerechnet, gerundet und ausgegeben.

Ausgabenverlauf

Hierfür wird der Controller „PurchaseTimelineCtrl“ gebraucht. Darin werden zuerst die Daten festgelegt und der Username abgefragt. Anschließend werden alle Einkäufe nach Datum sortiert sodass der älteste Einkauf zuerst im Array steht. Wenn der User auch der Einkaufsowner ist und der Einkauf im angegebenen Zeitraum liegt, wird ein neues Label erstellt. Der Tagespreis wird auf 0 gesetzt. Anschließend werden alle Ar-

tikel des Einkaufes durchlaufen und die Gesamtkosten aufaddiert. Die Gesamtkosten sind dann der Datenpunkt. Zuletzt werden die Daten dann an „charData“ übergeben und ein Diagramm erstellt.

Gruppenverlauf

Es wird der Controller „GroupPurchase-TimelineCtrl“ verwendet. Auch hier wird zuerst der Username abgefragt. Es wird geprüft, ob ein Label schon vorhanden ist. Wenn das nicht der Fall ist, wird dieses als Datum hinzugefügt. Nach diesem Lauf sind alle Einkaufsdaten erfasst. Nun wird mit „SelectionSort“ sortiert. Wie bei der Kaufhäufigkeits-Auswertung werden auch hier alle Einkäufe durchlaufen. Für jedes Label wird ein Datenpunkt angegeben. Alle Artikel für jeden Einkauf werden erfasst, daraufhin wird der Gesamtpreis ermittelt. Es wird die richtige Stelle im Array gesucht, welche von Gruppe und Datum abhängt. Nachdem alle Datenpunkte angegeben wurden ist das Array komplett. Auch hier wird zuletzt alles an „chart.js“ übergeben und ein Diagramm wird erstellt.

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 Besondere Herausforderungen des Projektes

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 Aussichten

In der folgenden Tabelle ist als kleine Übersicht gezeigt, welche Anforderungen an die EinkaufsApp umgesetzt wurden und welche noch nicht:

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 5: Aussichten

5 Lesson learned

Rückblickend war es eine besondere Herausforderung die gleiche Vorstellung der Aufgaben und Ziele im gesamten Team zu erreichen. Hierfür sind interne Abstimmungen innerhalb des Teams noch vor dem Projektstart sehr zu empfehlen. Ebenso ist eine im Vorfeld gut geplante und abgestimmte Organisationsstruktur unerlässlich. Da diese Vorarbeit hier nicht gegeben werden konnte, wurde zuerst auf eine flache Hierarchie mit einem hohen Teil an Selbstorganisation für jede Person gesetzt. Dies hatte das Ziel jede Person mit einem signifikanten Maß an Verantwortung und Einsicht in die sie umgebenden Projektgeschehnisse zu versehen, um die Qualität der geleisteten Arbeit zu sichern. Leider haben die Vorteile einer unkomplizierten und schnellen Kommunikation, sowie das flexible Arbeiten hier nicht überwogen. Demzufolge wurde eine weitere Organisationsebene über den Teilbereich "Entwicklung" eingefügt. Nachdem hier, abgetrennt von den anderen Arbeitsgruppen, kleinere Arbeitspakete erstellt wurden, ist die Produktivität spürbar gestiegen.

Weiterhin ist für das Arbeiten miteinander und das Verständnis der Ergebnisse ein gewisses Anteil an formalen Methoden sehr zu empfehlen. Es wurde vor allem in der Anfangsphase des Projektes relativ viel Zeit damit verbracht, sich über die Arbeitsmittel, das Vorgehen in den einzelnen Gruppen, die Ziele in den Gruppen und innerhalb des gesamten Projektes, sowie die Nachhaltigkeit und den Nutzen der verrichteten Arbeiten einig zu werden. Diese Herausforderung wäre ebenfalls bereits in der Planungsphase zu bewältigen.

Als ein weiteren sehr wichtigen Erfolgsfaktor ist das eigene Team und jedes Einzelne seiner Mitglieder als Stakeholder, zu betrachten. Hier können die Relevanz einzelner Personen für den Projekterfolg oder auch Ihre Motivation diesbezüglich stark variieren. Im Idealfall wird das Team, wie bereits zuvor bei den besonderen Herausforderungen erwähnt, konkret auf die spezifischen auftretenden Aufgaben zugeschnitten und dann im vollen Umfang mit einer qualitativ hochwertigen Lösung betraut.

Ebenfalls ist in diesem Projekt Zeit eine sehr Herausfordernde Komponente gewesen. Da hier ein fixer Abgabetermin unausweichlich ist und darüber hinaus alle Mitglieder nicht Vollzeit an diesem Projekt arbeiten, wirken Abwesenheiten der einzelnen Personen besonders schwer. Da dieses Risiko aber unabdingbar ist, kann es lediglich in der Planung berücksichtigt werden.

+Somit sind die kritischsten Punkte, die in einem Softwareprojekt wie diesem beachtet werden sollten, eine klare Definition der Aufgaben unter Berücksichtigung der zu Verfügung stehenden Zeit und der Personenressource, sowie ihrer Ausfallwahrscheinlichkeit. Allerdings auch die Zusammenstellung des Teams bezogen auf Motivation und Skills und nicht zuletzt eine ausreichende Planung des gesamten Projektes mit allen Beteiligten, um die Erwartungen und die notwendigen Arbeiten klar abzugrenzen, zu kommunizieren und auch entgegen zu nehmen.

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

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 [01]: Pflichtenheft
- Anhang [02]: Handbuch
- Anhang [03]: Installationsanleitung
- Anhang [04]: Angebot
- Anhang [05]: Proto.io Design
- Anhang [06]: Prototyp
- Anhang [07]: Datenfluss-PAP-EinkaufsApp
- Anhang [08]: Klassendiagramm
- Anhang [09]: MongoDB-Klassendiagramm
- Anhang [10]: Factories
- Anhang [11]: Aktivitätsdiagramm Ausgabenverlauf
- Anhang [12]: Aktivitätsdiagramm Einkauf
- Anhang [13]: Anwendungsfalldiagramm Ausgabenverlauf
- Anhang [14]: Anwendungsfalldiagramm Einkauf
- Anhang [15]: Sequenzdiagramm Ausgabenverlauf
- Anhang [16]: Sequenzdiagramm Einkauf
- Anhang [17]: Zustandsdiagramm