



DS&OR Lab  
Fachgebiet Wirtschaftsinformatik  
Fakultät Wirtschaftswissenschaften  
Universität Paderborn

---

Dokumentation

**„Entscheidungsunterstützung bei der  
Einkaufsplanung“**

---

vorgelegt bei:  
Prof. Dr. Leena Suhl

betreut durch:  
Daniel Müller

vorgelegt von:

Name	Matrikelnummer	E-Mail
Ahmad Hashemi	6785702	haschemi.ahmad@gmail.com
Felix Klösener	6687318	f.kloesener@gmx.de
Timo Schäfers	6657244	timo.schaefers@gmail.com
Agnes Schützenmeister	7008900	agnes@schuetzenmeister.eu

Paderborn, 31. August 2016

---

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis.....</b>	<b>II</b>
<b>Tabellenverzeichnis.....</b>	<b>III</b>
<b>Abkürzungsverzeichnis .....</b>	<b>IV</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Problematik .....	1
1.2 Zielsetzung .....	1
1.3 Vorgehensweise .....	1
<b>2 Analyse der Ausgangssituation.....</b>	<b>3</b>
2.1 Markt- und Wettbewerbsanalyse.....	3
2.2 Datenbeschaffung.....	3
<b>3 Das mathematische Modell .....</b>	<b>6</b>
3.1 Aufstellung des Modells.....	6
3.1.1 Definition der Entscheidungsvariablen.....	6
3.1.2 Zielfunktion.....	7
3.1.3 Restriktionen .....	7
3.1.4 Unzulässigkeit (Infeasibility).....	8
3.1.5 Implementierung des Modells.....	8
3.1.6 Subtour-Eliminierung .....	9
3.2 Test und Überprüfung des Java Programmes.....	10
<b>4 Die App .....</b>	<b>13</b>
4.1 App-Konzipierung.....	13
4.1.1 Funktion .....	13
4.1.2 Schematischer Aufbau .....	13
4.1.3 Datenmodell.....	15
4.2 Mock-up .....	17
4.3 Prototyp .....	19
<b>5 Fazit &amp; Ausblick .....</b>	<b>22</b>
<b>Literaturverzeichnis .....</b>	<b>VI</b>
<b>Eidesstattliche Erklärung.....</b>	<b>VII</b>

---

## Abbildungsverzeichnis

Bild 1-1: Projektplan des Teams Einkaufsplanung.....	2
Bild 1-2: Zeitliche Planung und Meilensteine des Projektes .....	2
Bild 3-1: Subtour-Problem.....	9
Bild 3-2: Funktionsradius des mathematischen Modells .....	10
Bild 4-1: Funktionen der Smartphone-App - Ergebnis des Team-Brainstormings .....	13
Bild 4-2: Schematischer Aufbau der dritten App-Oberfläche - Historie und Statistik .....	14
Bild 4-3: Datenmodell.....	16
Bild 4-4: Sketch-Design.....	18
Bild 4-5: Wireframe-Design .....	18
Bild 4-6: Vergleich geeigneter Tools zur Erstellung eines Prototypen .....	19
Bild 4-7: Ausschnitt aus dem Prototyp .....	20

## **Tabellenverzeichnis**

Tabelle 3-1: Auflistung aller relevanten Knoten .....	11
Tabelle 3-2: Darstellung der Entfernung aller Knoten zueinander .....	11
Tabelle 3-3: Auflistung der Produktpreise bzgl. der verschiedenen Supermärkte .....	12
Tabelle 4-1: Stärken und Schwächen von balsamiq .....	18

---

## Abkürzungsverzeichnis

bzw.	beziehungsweise
ca.	circa
d.h.	das heißt
f.	folgende
ff.	fortfolgende
gem.	gemäß
ggf.	gegebenenfalls
Hrsg.	Herausgeber
i. H .v.	in Höhe von
i. S. d.	im Sinne des / der
Kap.	Kapitel
LEH-Geschäfte	Lebensmitteleinzelhandel-Geschäfte
max.	maximal
mind.	mindestens
o.g.	oben genannte
s.	siehe
sog.	so genannt
TPP	Traveling-Purchaser-Problem
TSP	Travelling Salesman Problem
u.a.	unter anderem
v.	von
vgl.	vergleiche
z.B.	zum Beispiel

# 1 Einleitung

In den vergangenen Jahren hat sich die Einkaufswelt vor allem durch die Digitalisierung und die Art wie Menschen „shoppen“ stark verändert. Heutzutage ist es selbstverständlich geworden mit dem Laptop oder Smartphone online Preise zu vergleichen und einzukaufen.

## 1.1 Problematik

In Europa und vor allem in Deutschland ist bei der Beschaffung von Produkten bzw. Lebensmitteln der Preis ein bedeutendes Kriterium, um Geld zu sparen. Konsumenten bzw. Endverbraucher schauen beim Produktkauf zunächst auf den Preis. Bevor sich der Verbraucher für ein Produkt und anschließend zum Kauf entscheidet, macht er sich vorab Gedanken, wo es das Produkt eventuell günstiger geben könnte. Es werden Preise und Produkteigenschaften mit anderen Supermärkten in Prospekten und Online verglichen. Der Kunde schaut sich beispielsweise die Angebote in Prospekten von „Lidl“, „Aldi“ und „Netto“ an und kauft, entsprechend seinem Nutzerverhalten oder erstellten Einkaufszettel die Produkte dort, wo sie am günstigsten sind. Dies kostet Zeit und auch Fahrtkosten, wenn der Supermarkt am anderen Ende der Stadt liegt.

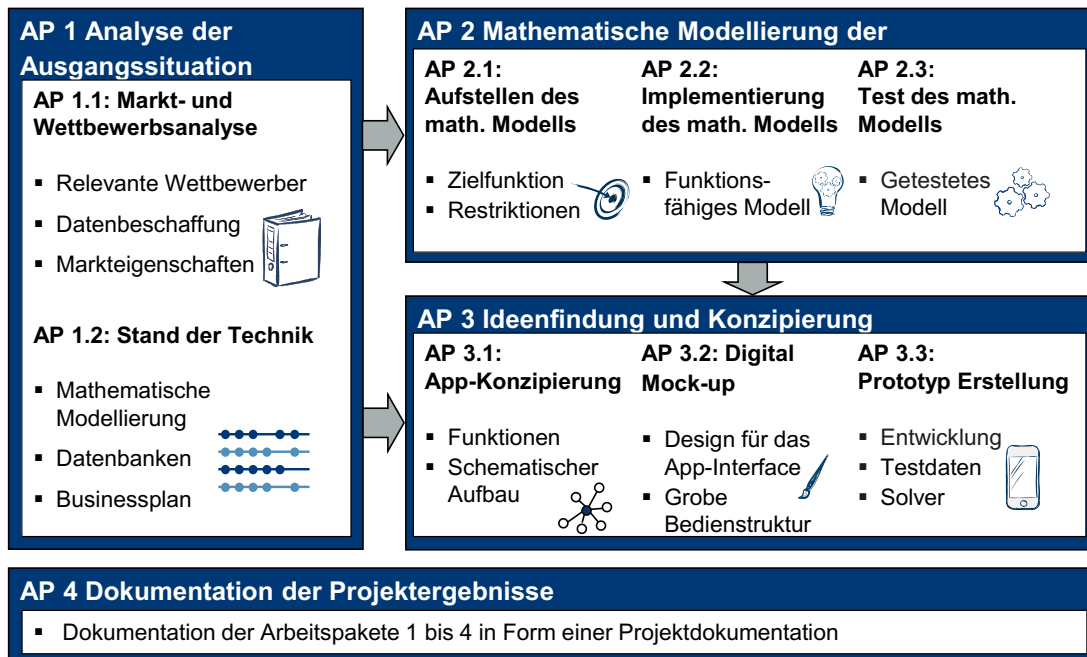
## 1.2 Zielsetzung

Das Ziel des vorliegenden Projekts ist es ein „Tool“ bzw. eine „App“ für den Bereich Paderborn zu entwickeln, um den Einkauf von Konsumenten zu optimieren. Das heißt, die zu entwickelnde „App“ für das Smartphone soll dem Nutzer exakt anzeigen können, wie er die Produkte seines erstellten Einkaufszettels am günstigsten einkaufen und erreichen kann. Dabei werden sowohl die Preise und die Wegkosten berücksichtigt sowie der preisgünstigste Weg kalkuliert, wenn mehrere Produkte in verschiedenen Supermärkten zu kaufen sind.

## 1.3 Vorgehensweise

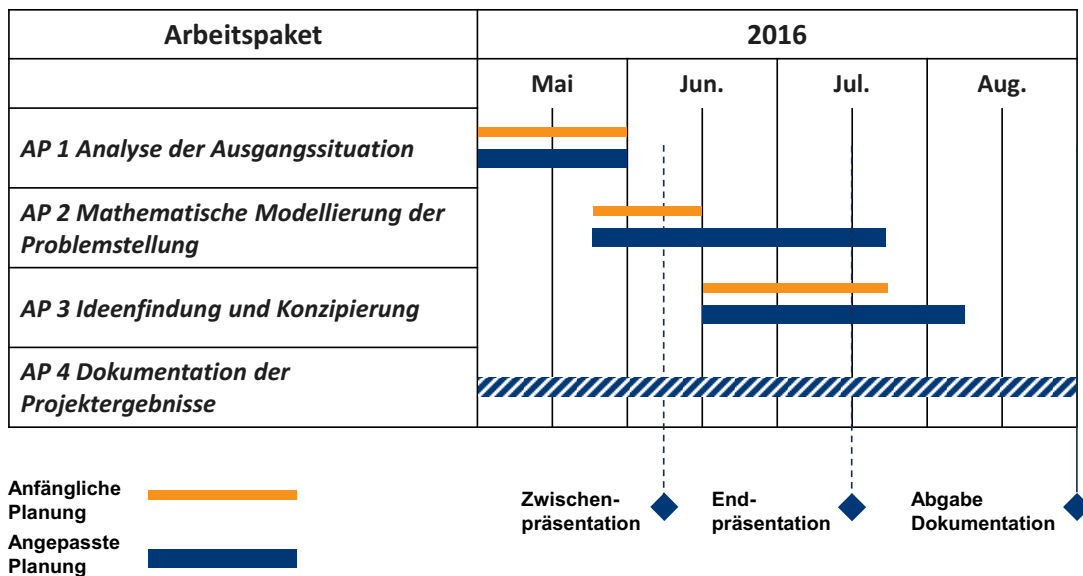
Für den erfolgreichen Abschluss des Projektes und um das o.g. Ziel zu erreichen, müssen im Vorfeld einige Überlegungen bzw. Projektannahmen getroffen werden. Zunächst gilt es sich damit zu beschäftigen, ob bereits ein entsprechendes „Tool“ oder auch eine „App“ bzw. „Website“ zur Einkaufsoptimierung existiert. Eine Analyse soll Aufschluss darüber geben, welche Produkte zu welchen Preisen in unterschiedlichen Discountern im Raum Paderborn angeboten werden. Hierfür wird eine Datenbank über die Einkaufsläden und ihre Produkte benötigt. Des Weiteren ist die Entfernung der unterschiedlichen Supermärkte zu ermitteln. Zur Lösung des Problems wird ein mathematisches Modell aufgestellt, welches sowohl die Kosten für die Strecke berücksichtigt, als auch die Kosten für den Einkauf minimiert. Zum Abschluss soll das hier aufgeführte Konzept anhand eines Mock-up getestet werden. Eine Empfehlung gibt Aufschluss darüber, wie das weitere Vorgehen und eine Umsetzung aussehen könnten. Abschließend möchten wir über Probleme während der Projektarbeit berichten.

Ein Projektplan wurde auf Basis dieser inhaltlichen Ausführungen erstellt und ist in Bild 1-1 abgebildet.



**Bild 1-1: Projektplan des Teams Einkaufsplanung**

Aus diesem Projektplan wurde die folgende zeitliche Planung und die Meilensteine des Projektes abgeleitet, dies ist in Bild 1-2: Zeitliche Planung und Meilensteine des Projektes dargestellt.



**Bild 1-2: Zeitliche Planung und Meilensteine des Projektes**

## 2 Analyse der Ausgangssituation

Dieses Kapitel beschäftigt sich mit potenziellen Konkurrenten und den Markteigenschaften. Das bedeutet, die Markt- und Wettbewerbsanalyse gibt Aufschluss darüber, ob die von uns geplante App zur Einkaufsoptimierung bereits auf dem Markt existiert. Des Weiteren wird erklärt, wie die Daten für die konzipierte Einkaufs-App beschafft werden.

### 2.1 Markt- und Wettbewerbsanalyse

Die Marktanalyse ist insbesondere für die Einführung eines neuen Produktes von Bedeutung. Sie dient als Orientierungshilfe und beschäftigt sich mit dem Zielmarkt [BEE04, S. 332]. Es findet eine Marktbeobachtung statt, um herauszufinden, wie sich das Kaufverhalten, Märkte und die Bedürfnisse der einzelnen Abnehmer entwickeln. Schlussendlich können erfolgreiche Produkte nur solche sein, welche die Konsumentenbedürfnisse besser befriedigen als Produkte von der Konkurrenz [Kre1987, S. 90]. Zur langfristigen Sicherung des Markterfolges ist eine Konkurrenzanalyse unerlässlich. Nur so können rechtzeitig Bedrohungen und Chancen erkannt werden, um dann mit entsprechenden Maßnahmen reagieren zu können [Kai13, S. 20 f.].

Recherchen im Internet haben gezeigt, dass es viele Apps gibt, die einen Einkaufszettel verwalten. Zudem werden Produkte und Angebote angezeigt, welche am günstigsten sind. Zu diesen Apps zählen u.a. „Sparpionier“ [Spa16-ol], „Besorger“ [Bes16-ol] und „Bring“ [Bri16-ol]. Außerdem informiert „Apriku“ auf einer Angebotsplattform über Wochen- und Tagesangebote sowie über Produkte am Ende der Mindesthaltbarkeit. Ziel ist es, Geld zu sparen und dabei zu helfen, die Wegwurfrate von Lebensmitteln nachhaltig zu reduzieren [Apr16-ol].

Die Analyse hat gezeigt, dass es bisher noch keine App gibt, welche eine kostenminimale Einkaufsplanung unter Berücksichtigung der Strecke und der jeweiligen Marktpreise durchführt.

Dieses Projekt fokussiert sich zunächst auf den Raum Paderborn. Hierzu ist eine Marktübersicht der Supermärkte erforderlich. Ebenso müssen die Preise der einzelnen Produkte herausgesucht werden. Im Abschnitt 2.2 wird die Beschaffung der Daten näher erläutert.

### 2.2 Datenbeschaffung

Für eine funktionsfähige App ist die Beschaffung einer Datengrundlage von essentieller Bedeutung. Im Kontext der mathematischen Modellierung der Einkaufsplanung zählen dazu beispielsweise folgende Informationen:

1. Orte der jeweiligen Lebensmitteleinzelhandel-Geschäfte (LEH-Geschäfte) sowie die eigenen Standortdaten
2. Entfernung zwischen den LEH-Geschäften bzw. den eigenen Standorten
3. Kürzeste Route zwischen einer beliebigen Anzahl an LEH-Geschäften und den eigenen Standorten inklusive Kartenmaterial
4. Aktuelles Produktangebot (inkl. Fotos und Zusatzangaben) sowie die Preise in den jeweiligen LEH-Geschäften

Die Schwierigkeit der Datenbeschaffung wird maßgeblich davon beeinflusst inwieweit die Daten frei verfügbar sind und wie stark die Daten im Zeitverlauf schwanken. Weiterhin kann es sein, dass Daten aktiv verschleiert werden, um die Datengenerierung zu erschweren.



**Punkt 1)** bezieht sich ausschließlich auf die für die Routenplanung notwendigen Informationen (Adressen der LEH-Geschäfte, eigene Standorte). In diesem Fall haben alle Beteiligten ein großes Interesse daran die Daten bereitzustellen. Die Nutzer der App möchten ihre Einkaufsplanung möglichst effizient gestalten und stellen daher die notwendigen Daten zur Verfügung. Die Betreiber der LEH-Geschäfte sind selbstverständlich ebenso bereit die Adressen zu veröffentlichen, damit vor dem Hintergrund ihres wirtschaftlichen Erfolgs möglichst viele Kunden ihre Einkäufe dort erledigen. Darüber hinaus handelt es sich hierbei um wenig schwankende Daten, da eine Neueröffnung oder Schließung eines LEH-Geschäftes nicht häufig stattfindet. Trotzdem müssen diese Daten in regelmäßigen Zeitabständen überprüft und dementsprechend angepasst werden, um eine dauerhafte Aktualität der Datenbasis zu gewährleisten.

Die **Punkte 2) und 3)** nehmen Bezug auf die Verarbeitung dieser Daten in Form von einer Routenplanung für den Einkauf. Diese Daten müssen ständig individuell generiert werden. Das benötigte Kartenmaterial sowie die Software zur Routenplanung sind nicht frei verfügbar – aber verfügbar. Eine Möglichkeit zur Generierung dieser Daten für einen gewerblichen Rahmen bietet bspw. die Google Maps API, die genau für diesen Anwendungsfall konzipiert wurde [Google16-ol]. Die Nutzung dieser Dienstleistung erfordert selbstverständlich einen entsprechenden Kapitaleinsatz, allerdings gibt es praktisch keine andere Möglichkeit als diese Fremdleistung in Anspruch zu nehmen.

**Punkt 4)** thematisiert die wichtigste und zugleich am schwierigsten zu beschaffende Datengrundlage für die Einkaufsplanung. Sowohl das Produktangebot als auch die Preise unterliegen einer ständigen Änderung aufgrund eines wachsenden bzw. sich ändernden Sortiments und den wöchentlich bzw. täglich wechselnden Angeboten. Darüber hinaus sind die LEH-Geschäfte nicht daran interessiert ihr gesamtes Sortiment sowie die Preise frei verfügbar zu machen. Genau das Gegenteil ist der Fall – sie bemühen sich die Produktpreise durch verschiedene Verschleierungstaktiken maximal zu verzerren, sodass der Kunde zum einen Preiserhöhungen nicht bemerkt und zum anderen die Vergleichbarkeit unter den Märkten nur mit hohem Aufwand möglich ist (vgl. Coca-Cola in 1,25l Flaschen im Aldi bzw. 1l Flaschen im Rewe/Edeka) [Vzh12-ol].

Aufgrund der geringen Kooperationsbereitschaft der LEH-Betreiber gibt es zahlreiche Internetseiten, die durch Mithilfe einer großen und interaktiven Community versuchen eine Preistransparenz für sämtliche LEH-Geschäfte herzustellen:

- <http://www.preisvergleich-lebensmittel.com>
- <http://www.supermarktcheck.de>
- <http://www.discounter-preisvergleich.de>
- <http://sparpionier.com/index.php>

Alle Anbieter stehen dabei denselben Herausforderungen entgegen. Eine Preistransparenz für jedes LEH-Geschäft in ganz Deutschland bedeutet die millionenhafte und regelmäßige Sammlung von Preisinformationen. Ein Großteil der Internetseiten versucht diese Herausforderung dadurch zu lösen, dass sie den Nutzern der Website die Möglichkeit geben, Produktpreise online zu veröffentlichen. Diese kollektive Datensammlung führt allerdings dazu, dass die Preisdatenbank zum einen zeitlich sehr asynchron gefüllt wird und zum anderen häufig die Vollständigkeit nicht gegeben ist. Eine saubere Datensammlung kann daher nur für die wöchentlichen Angebote gewährleistet werden, welche von den LEH-Geschäften veröffentlicht werden.

Ein anderer Ansatz zur Preisbeschaffung besteht darin, die Preise aus den Online-Shops der verschiedenen LEH-Geschäfte auszulesen. Hier sind die Produkte inklusive Preis, Zusatzinformationen und Produktbildern systematisch sortiert. Allerdings existieren zwei schwerwiegende Probleme, weshalb diese Herangehensweise nicht vollständig zum Erfolg führt. Auf der einen Seite gibt es nur von Rewe (rewe.de) und Edeka (edeka24.de) einen Online-Shop, der nahezu das gesamte Sortiment beinhaltet und auf der anderen Seite wird nicht gewährleistet, dass die Online-Preise vollständig mit den Markt-Preisen übereinstimmen [Stern15-ol]. Daher wird das Problem nicht vollständig gelöst und die Preisinformationen sind nicht zuverlässig.

Zusammenfassend kann dementsprechend für Punkt 4) festgehalten werden, dass keine Lösung existiert, die die benötigten Daten bereitstellen könnte. Vielmehr besteht die Gefahr, dass dieses Problem nur mit unverhältnismäßig großem Aufwand gelöst werden kann, weshalb eine wirtschaftliche Umsetzung der App nicht möglich sein könnte. Daher muss im Rahmen einer Geschäftsmodell-Konzipierung ein besonderes Augenmerk auf dieses Problem gelegt werden.

In Kapitel 3 wird das Problem der Optimierung genauer analysiert. Das heißt, die grundlegende Problematik wird anhand eines wissenschaftlichen Modells betrachtet. Ziel ist es herauszufinden, wie ein Modell aussehen könnte, welches sowohl die LEH-Geschäfte als auch die Kosten der Strecke, Produkte und Preise berücksichtigt. Auch wie sich das Problem mathematisch modellieren und anwenden lässt, wird genauer anhand von Variablen, Zielfunktionen und Restriktionen erklärt und beschrieben.

### 3 Das mathematische Modell

In Kapitel 3 (Arbeitspaket 2) wird der gesamte Ablauf der Konzipierung des mathematischen Modells zur optimalen Einkaufsplanung beschrieben. Das Aufstellen des Modells und die Implementierung des mathematischen Modells in Java wird in Kapitel 3.1 näher erklärt. Das Arbeitspaket 2 wird mit dem Kapitel 3.2 abgeschlossen, in welchem die Beschaffung der Testdaten sowie die Testdurchläufe des Modells erläutert werden.

#### 3.1 Aufstellung des Modells

Als Grundlage und Lösung für die mathematische Modellierung, der zu erstellenden Einkaufs-App, dient das klassische Travelling-Purchaser-Problem (TPP). Es handelt sich um ein kombinatorisches Optimierungsproblem. Das heißt, eine bestimmte Person, die sich an einem Ort befindet möchte weitere Orte besuchen, um dort Produkte zu kaufen. Die Kosten der Reise bzw. Fahrt zwischen den Lebensmitteleinzelhandelsgeschäften (LEH-Geschäften) und die Preise der angebotenen Artikel im jeweiligen LEH-Geschäft sind bekannt. Die Aufgabe besteht darin, in Bezug auf einen gegebenen Einkaufszettel eine Route zu finden, um die kombinierten Kosten des Einkaufes und der Reise zu minimieren.

Das Travelling-Purchaser-Problem (TPP) gilt als ein NP-schweres Problem. Das Travelling Salesman Problem (TSP) ist ein Sonderfall dieses Problems [GI05, S. 315]. Das Problem lässt sich mit Hilfe eines Graphen mit Knoten und Kanten abbilden und auf das Projekt übertragen. Das bedeutet, die Knoten repräsentieren die Orte bzw. die LEH-Geschäfte und die Kante zwischen zwei Knoten die zurückzulegende Strecke.

##### 3.1.1 Definition der Entscheidungsvariablen

Angenommen wird, dass  $V$  die Menge von Knoten ist (alle berücksichtigten LEH-Geschäfte und der Knoten des Wohnhauses).  $E$  repräsentiert die Menge von Kanten, die die verschiedenen LEH-Geschäfte verbinden. Da eine Verbindung zwischen allen LEH-Geschäften besteht wird der Graph als vollständig angenommen – d.h. es gibt genau eine Kante zwischen jedem beliebigen Knotenpaar. Jede Kante im Graph erhält eine binäre Variable.

$$x_{ij} \ (n \times n) = \begin{cases} 1 & \text{wenn Kante } (i, j) \text{ auf Route ist} \\ 0 & \text{anderenfalls} \end{cases} \\ \forall i, j \in V$$

Des Weiteren lässt sich noch eine weitere binäre Variable definieren, um zu bestimmen, welcher Artikel in welchem LEH-Geschäft verkauft wird.

$$y_{kj} \ (m \times n) = \begin{cases} 1 & \text{wenn } k. \text{ Artikel in } j. \text{ Supermarkt gekauft wird} \\ 0 & \text{anderenfalls} \end{cases} \\ \forall k \in A \text{ und } j \in V$$

Um diese zwei Variablen zu verbinden, wird eine zusätzliche Variable benötigt. Diese bestimmt, welche LEH-Geschäfte in der Lösung ausgewählt werden.

$$a_j = \begin{cases} 1 & \text{wenn } j. \text{ Supermarkt ausgewählt wird} \\ 0 & \text{anderenfalls} \end{cases}$$

Da die Kanten ungerichtet sind, d.h.  $x_{ij} = x_{ji}$ , reicht es, wenn nur die Kanten mit  $i < j$  im Modell eingefügt werden.

### 3.1.2 Zielfunktion

Der Zweck des Modells ist die Minimierung der kombinierten Kosten von zwei Termen. Auf der einen Seite die Summe der Fahrtkosten bzw. der gesamten Distanz der Route und auf der anderen Seite die Kosten des Einkaufes. Dazu werden zwei Matrizen als Input berechnet. Zu Beginn wird die  $c_{ij}$ -Matrix bestimmt, welche die Distanz zwischen den Knoten  $i$  und  $j$  enthält. Der Matrixeintrag  $c_{23}$  steht beispielsweise für die Distanz zwischen dem zweiten und dritten LEH-Geschäft und wie oben angedeutet gilt  $c_{ij} = c_{ji}$ . Weiterhin wird die Matrix  $d_{kj}$  definiert, welche die Preise der Artikel  $k$  in den jeweiligen LEH-Geschäften  $j$  enthält.

$$\text{minimiere } \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \sum_{k \in A} \sum_{j \in V} d_{kj} y_{kj}$$

Es ist zu berücksichtigen, dass der erste Term die Einheit Kilometer aufweist und der zweite Term die Einheit Euro. Daher wird der erste Term mit einem Einheitskostensatz multipliziert, um diesen auch in Euro abbilden zu können. Angenommen wird, dass jeder Kilometer einer Fahrt durchschnittlich 0,10 € kostet.

### 3.1.3 Restriktionen

Jedes LEH-Geschäft bzw. jeder Knoten auf der ausgewählten Route soll nur einmal besucht werden. Aus diesem Grund darf jeder Knoten der Route und die Knoten des Hauses genau eine eingehende und eine herausgehende Kante aufweisen. D.h. für jeden Knoten  $i$  auf der Route sollen genau zwei  $x_{ij}$  der binären Variablen „1“ betragen. Diese Restriktionen lassen sich folgendermaßen definieren:

$$\text{Haus Restriktion} \quad \sum_{j \in V} x_{Hj} = 2$$

$$\text{Supermarkt Restriktion} \quad \sum_{j \in V} x_{ij} = 2 \quad \forall i \in V$$

Diese Restriktionen stellen sicher, dass der Käufer jedes LEH-Geschäft sowie sein Haus genau einmal anläuft und verlässt.

Zum Erreichen der Optimalität soll auf Basis der Artikel- und Wegkosten, die bei den unterschiedlichen LEH-Geschäften anfallen, ein Kompromiss gefunden werden. D.h. die Artikel könnten nur in LEH-Geschäften gekauft werden, die sich auf der kürzesten Route befinden, obwohl es sich im Hinblick auf die Artikelkosten eventuell nicht um die preiswerteste Möglichkeit handelt. Die mathematische Formulierung dieser Restriktionen sehen wie folgt aus:

$$\text{artikel Restriktion 1:} \quad \sum_{k \in A} y_{kj} \leq M \quad \forall j \in V$$

$$\text{artikel Restriktion 2:} \quad \sum_{k \in A} y_{kj} \geq a_i \quad \forall j \in V$$

$M$  bezeichnet eine so große Zahl, welche der linke Term nie übertreffen kann.

Es ist sicherzustellen, dass am Ende des Einkaufsvorganges der Einkaufszettel des Benutzers vollständig abgedeckt ist. D.h. alle vom Benutzer ausgewählten Artikel sollen mittels des vorgeschlagenen Einkaufsplans gekauft werden.

$$\text{Einkaufszettel Restriktion} \quad \sum_{k \in A} y_{kj} = e_k \quad \forall k \in A$$

Dabei repräsentiert  $e_k$  den eingegebenen Einkaufszettel als Input.

Es wäre sinnvoll bzgl. der Umsetzung, wenn der Benutzer einschränken könnte wie viele LEH-Geschäfte er maximal besuchen möchte oder wie viel Zeit er für den Einkaufsvorgang investieren möchte. Die Formulierung dieser Einschränkung sieht wie folgt aus:

$$\text{Restriktion der Höchsten Anzahl von Supermärkte} \quad \sum_{j \in V} a_j \leq \text{Märkte}$$

### 3.1.4 Unzulässigkeit (Infeasibility)

Ein Optimierungsproblem wird als unzulässig (eng. infeasible) bezeichnet, wenn es keine Lösung gibt, die alle Restriktionen erfüllt, genauer gesagt, wenn keine zulässige Lösung erstellt werden kann. Da jede real modellierte Operation gegenüber den Einschränkungen der Realität treu bleiben muss, deutet eine zutreffende Unzulässigkeit meistens auf eine Problematik zwischen den formulierten Restriktionen hin. Ein eindeutiger Ansatz zur Behebung dieses Problems ist zu entdecken, damit aufgedeckt wird zwischen welchen Restriktionen das Problem vorliegt. In einem nächsten Schritt muss das Modell an diese eventuellen Sonderfälle angepasst werden. In 3.1.3 wurden zwei Restriktionen formuliert, welche sicherstellen, dass jeder Knoten auf der Route zwei Kante besitzt (eine eingehende und eine herausgehende Kante). Angenommen es wird zur Berechnung des optimalen Einkaufsplans nur ein LEH-Geschäft ausgewählt, dann entstehen zwei Knoten (Haus und LEH-Geschäft), was zu einem Widerspruch bzgl. der festgelegten Restriktionen führt. Es ist lediglich eine Kante erforderlich, um die zwei Knoten zu verbinden. D.h. es gibt einen Widerspruch zwischen den zwei beschriebenen Restriktionen bzgl. des beschriebenen Falls. Die zugehörigen Restriktionen und die Objektfunktion sollen so angepasst werden, dass die Knoten nur eine Kante brauchen und die Kosten dieser einzigen Kante doppelt eingerechnet werden – einmal für die Hinfahrt und einmal für die Rückfahrt.

$$\sum_{j \in V} x_{Hj} = 1$$

$$\sum_{j \in V} x_{ij} = a_i \quad \forall i \in V$$

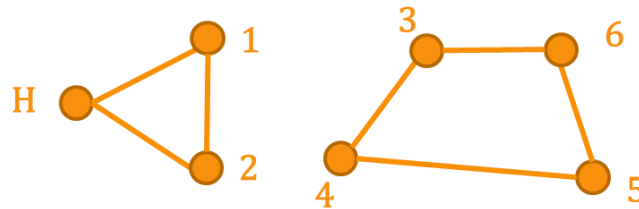
$$\sum_{i \in V} \sum_{j \in V} 2c_{ij}x_{ij} + \sum_{k \in A} \sum_{j \in V} d_{kj}y_{kj}$$

### 3.1.5 Implementierung des Modells

Im letzten Kapitel wird die Formulierung des Problems als eine integer Programmierung beschrieben. Zur Implementation des Projekts wurde Gurobi und Java als Interface des Solvers angewendet. Zur Implementierung der Subtour-Eliminierungsrestriktionen wurde die so genannte „Lazy Constraints“ durch die „Callback Methode“ von Gurobi dargestellt.

### 3.1.6 Subtour-Eliminierung

Nur mit den formulierten Restriktionen würden eventuell Routen erzeugt werden, die nicht miteinander verbunden wären. Die folgende Abbildung 3-1 stellt beispielweise einen Graphen mit sechs Knoten und zwei unverbundenen Subtours dar. Die erste Subtour führt durch die Knoten 1, 2 und das Wohnhaus. Die zweite Subtour führt durch die Knoten 3, 4, 5 und 6. Es wird berücksichtigt, dass jeder auf der Route liegende Knoten und der Knoten des Wohnhauses genau zwei Kanten aufweist, jedoch steht kein Pfad für den Käufer zur Verfügung, um die Reise zu allen Knoten durchführen zu können.



**Bild 3-1: Subtour-Problem**

Damit diese Problematik nicht auftritt, müssen zusätzliche Restriktionen zum Modell hinzugefügt werden, um das Auftreten solcher Lösungen zu eliminieren.

$$\sum_{i,j \in S, i \neq j} x_{ij} \leq |S| - 1 \quad \forall S \subset \{\text{ausgewählte Knoten bzw. Supermärkte}\} \\ S \neq \emptyset$$

Diese Restriktion stellt sicher, dass die Anzahl der Kanten in jeder nicht leeren Teilmenge  $S$  der ausgewählten LEH-Geschäfte bzw. die Anzahl auf der Route liegenden Knoten kleiner als die Anzahl der Knoten der Teilmenge sein soll. Wenn die Anzahl der Kanten tatsächlich gleich  $S$  wäre, würde eine Subtour entstehen. Beispielsweise hat die obere Abbildung die Teilmenge der Knoten  $S = \{H, 1, 2\}$  drei Kanten auf dieser Route.  $x_{H1} = 1$ ,  $x_{H2} = 1$ ,  $x_{12} = 1$ , dann:

$$\sum_{i,j \in \{H,1,2\}, i \neq j} x_{ij} = 3 > 2 = |\{H, 1, 2\}| - 1$$

Auf diese Weise wird die Subtour-Eliminierungsrestriktion verletzt.

Allerdings gibt es bzgl. der Implementierung solcher Restriktionen zwei Probleme. Als erstes ist die ursprüngliche Anzahl der ausgewählten LEH-Geschäfte bzw. die auf der Route liegenden Knoten nicht bestimmt. Als nächstes vervielfacht sich die Anzahl der Restriktionen exponentiell. Wenn das Paket von Knoten eine Menge von  $n$  hätte, gäbe es  $2^n - 2$  Teilmengen von  $S$  und die jeweilige Teilmenge braucht eine dazugehörige Restriktion.

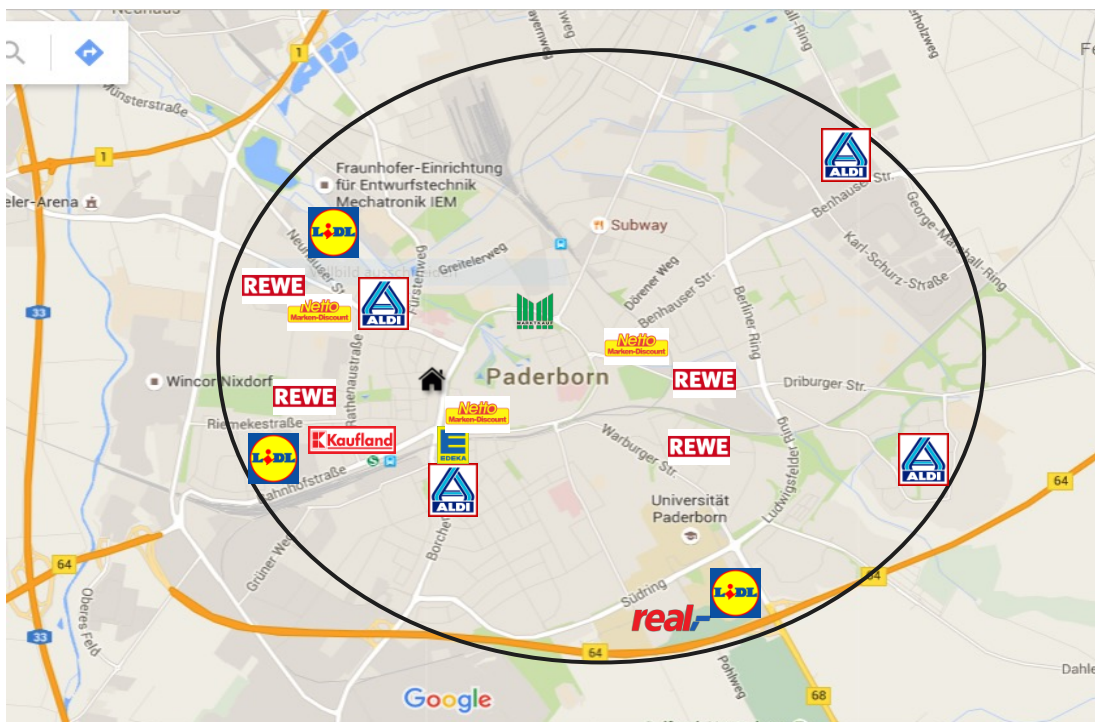
Zur Behebung dieses Problems werden diese Restriktionen zum Modell durch einen so genannten "Lazy" Modus generiert und addiert – für jedes  $S$  wird diese Restriktion als „Lazy“ eingefügt. Das Modell besitzt ursprünglich keine Subtour-Eliminierungsrestriktion. Es wird abgewartet ob Gurobi eine zulässige Lösung übergibt, welche die anderen Einschränkungen erfüllt. Dann wird der erste Zyklus bzw. die Reihenfolge von Knoten, welche die übergebene Route enthält, berechnet. Wenn die Länge dieser Zyklen gleich der Anzahl der ausgewählten Knoten ist, so ist das Modell gelöst und die Lösung ist zulässig. Ansonsten weist der Zyklus eine Subtour auf und es müssen die zugehörigen Restriktionen zur Eliminierung der Subtours als „Lazy Constraint“ addiert werden. Im Anschluss daran

löst Gurobi das neu entstandene Modell. Dieser Prozess wird iterativ fortgesetzt bis die optimal übergebene Lösung von Gurobi keine Subtour enthält – d.h. bis alle eingefügten Subtour-Eliminierungsrestriktionen erfüllt sind.

### 3.2 Test und Überprüfung des Java Programmes

In Kapitel 3.2 werden die Testdatenbeschaffung und die Testdurchläufe in Java näher beschrieben. Zum Testen des Modells soll eine geeignete Menge an Supermärkten definiert werden und ein Radius, in welchem sich diese sinnvolle Menge an LEH-Geschäften befindet. Die Testdurchläufe dienen zur Überprüfung und Sicherstellung der korrekten Funktionsweise des Programmes. Es wurden mehrere Durchläufe zu verschiedenen Zeitpunkten durchgeführt und unterschiedliche Probleme detektiert.

In einem ersten Schritt wurde in Paderborn ein relevantes Nutzungsgebiet ausgewählt – in diesem Bereich wurden 18 verschiedene LEH-Geschäfte identifiziert (siehe Bild 3-2).



**Bild 3-2: Funktionsradius des mathematischen Modells**

Des Weiteren ist ein Wohnort und ein Arbeitsort festgelegt worden. Diese weisen jeweils einen geeigneten Standort in Bezug auf den gewählten Nutzungsradius auf. Das Modell soll nicht nur einen kreislaufartigen Einkauf realisieren können (Start: Wohnort – Ziel: Wohnort), sondern auch einen Einkauf von einem beliebigen Standort mit dem Wohnort als Ziel umsetzen (Start: Arbeitsort – Ziel: Wohnort). Folglich besitzt das Modell 20 unterschiedliche Knoten (siehe Tabelle 3-1).

**Tabelle 3-1: Auflistung aller relevanten Knoten**

	WOHNORT	REWE	ALDI	LIDL	NETTO	EDEKA	MARKTKAUF	KAUFLAND	REAL
		Filiale 1	Filiale 5	Filiale 9	Filiale 12	Filiale 15	Filiale 16	Filiale 17	Filiale 18
Adresse	Imadstraße 4 33102 Paderborn	Klöcknerstraße 38-44 33102	Rathenastraße 9 33102	Neuhäuser Straße 127 33102	Jahnstraße 6 33098	Borchener Straße 25 33098	Detmolder Straße 4 33102	Riemekestraße 37 33102	Husener Straße 121 33100
Öffnungszeiten	07:00 - 21:30	08:00 - 21:00	08:00 - 21:00	07:00 - 21:00	07:00 - 22:00	07:00 - 21:00	07:30 - 21:00	07:00 - 22:00	08:00 - 22:00
Telefon	05251 310060	0800 7234870	0800 4353361	0800 4353361	0800 2000015	05251 778093	05251 180420	05251 1840530	05251 689080
	ARBEITSORT	Filiale 2	Filiale 6	Filiale 10	Filiale 13				
Adresse	An der Talle 27-31 33102 Paderborn	Eisener Straße 20 33102	Borchener Straße 25 33098	Bahnhofstraße 84 33102	Marienpl. 16 33098				
Öffnungszeiten	07:00 - 22:00	08:00 - 20:00	07:00 - 21:00	07:00 - 20:00					
Telefon	05251 1335	0800 7234870	0800 4353361	0800 2000015					
		Filiale 3	Filiale 7	Filiale 11	Filiale 14				
Adresse		Driburger Straße 44 33100	Pohlweg 110 33100	Warburger Straße 130 33100	Benhauser Straße 4 33100				
Öffnungszeiten		08:00 - 22:00	09:00 - 20:00	07:00 - 21:00	07:00 - 21:00				
Telefon		05251 680613	0800 7234870	0800 4353361	0800 2000015				
		Filiale 4	Filiale 8						
Adresse		Warburger Straße 52 33098	Senefelderstraße 13 33100						
Öffnungszeiten		07:00 - 22:00	08:00 - 20:00						
Telefon		05251 680672	0800 7234870						

In einem nächsten Schritt wurden die Entfernungen (in km) zwischen den verschiedenen Knoten bestimmt und in einer Tabelle gesammelt. Darüber hinaus wurden die Entfernungen mit einem Einheitskostensatz von 0,10 € pro km multipliziert, welcher einen groben Kostensatz für den Benzinpreis darstellen soll. Diese Entfernungsmatrix dient als Grundlage für das Modell und wird bei der Implementierung als  $c_{ij}$ -Matrix bezeichnet (siehe Tabelle 3-2).

**Tabelle 3-2: Darstellung der Entfernung aller Knoten zueinander**

	Arbeitsort	Wohnort	Filiale 1	Filiale 2	Filiale 3	Filiale 4	Filiale 5	Filiale 6	Filiale 7	Filiale 8	Filiale 9	Filiale 10	Filiale 11	Filiale 12	Filiale 13	Filiale 14	Filiale 15	Filiale 16	Filiale 17	Filiale 18
Arbeitsort		0,5	0,56	0,42	0,5	0,56	0,44	0,64	0,71	0,38	0,35	0,59	0,64	0,46	0,53	0,41	0,64	0,39	0,54	0,65
Wohnort			0,15	0,15	0,25	0,2	0,11	0,15	0,3	0,45	0,19	0,13	0,32	0,14	0,04	0,19	0,15	0,13	0,07	0,28
Filiale 1				0,12	0,4	0,32	0,14	0,27	0,48	1,02	0,21	0,08	0,74	0,12	0,22	0,34	0,27	0,32	0,08	0,47
Filiale 2					0,34	0,34	0,05	0,3	0,51	0,53	0,09	0,2	0,46	0,07	0,18	0,29	0,3	0,22	0,14	0,43
Filiale 3						0,25	0,3	0,32	0,26	0,27	0,38	0,35	0,19	0,33	0,28	0,1	0,32	0,14	0,32	0,27
Filiale 4							0,44	0,35	0,19	0,48	0,64	0,39	0,12	0,45	0,35	0,31	0,35	0,32	0,36	0,19
Filiale 5								0,26	0,41	0,5	0,09	0,15	0,43	0	0,15	0,25	0,26	0,18	0,1	0,39
Filiale 6									0,27	0,47	0,26	0,14	0,29	0,2	0,1	0,21	0	0,19	0,11	0,25
Filiale 7										0,5	0,48	0,36	0,14	0,42	0,33	0,28	0,27	0,29	0,33	0
Filiale 8											0,62	0,53	0,41	0,55	0,5	0,25	0,5	0,32	0,5	0,48
Filiale 9												0,24	0,5	0,11	0,22	0,33	0,34	0,25	0,18	0,47
Filiale 10													0,5	0,19	0,27	0,41	0,38	0,38	0,14	0,59
Filiale 11														0,44	0,34	0,29	0,34	0,31	0,35	0,13
Filiale 12															0,17	0,28	0,27	0,2	0,08	0,4
Filiale 13																0,22	0,18	0,16	0,11	0,31
Filiale 14																	0,3	0,12	0,3	0,31
Filiale 15																		0,19	0,11	0,25
Filiale 16																			0,29	0,33
Filiale 17																				0,31
Filiale 18																				

Des Weiteren wird für die Simulation bzw. das Testen des Modells eine Matrix mit Produktpreisen pro LEH-Geschäft benötigt, welche als  $d_{kj}$ -Matrix bezeichnet wird (siehe Tabelle 3-3). Hierzu wurden 26 Produkte mit realistischen Produktpreisen ausgewählt – die Produktpreise und Verfügbarkeit in den LEH-Geschäften wurden sinnvoll bestimmt und spiegeln die realistischen Produktpreise wieder. Um eine Nicht-Verfügbarkeit in einem LEH-Geschäft zu simulieren, wurde ein sehr hoher Produktpreis von 10.000€ festgelegt – z.B. kann das Produkt 24 (Carolinen Medium 12L) nicht in den vier Aldi Märkten gekauft werden.



**Tabelle 3-3: Auflistung der Produktpreise bzgl. der verschiedenen Supermärkte**

Artikel	Rewe				Aldi				Lidl			Netto			Edeka	Marktkauf	Kaufland	Real
	Filiale 1	Filiale 2	Filiale 3	Filiale 4	Filiale 5	Filiale 6	Filiale 7	Filiale 8	Filiale 9	Filiale 10	Filiale 11	Filiale 12	Filiale 13	Filiale 14	Filiale 15	Filiale 16	Filiale 17	Filiale 18
1 Apfel 1kg	1,3	1,3	1,3	1,3	1	1	1	1	1,2	1,2	1,2	1	1	1	1,1	1,2	1,3	1,1
2 Zitrone 2 Stück	1,6	1,6	1,6	1,6	1,2	1,2	1,2	1,2	1	1	1	1,4	1,4	1,4	1,5	1,7	1,7	1,1
3 Rispentomaten 500g	1,5	1,5	1,5	1,5	1	1	1	1	1,3	1,3	1,3	1,6	1,6	1,6	1,5	1,4	1,4	1,3
4 Paprika 500g	2	2	2	2	1,6	1,6	1,6	1,6	1,4	1,4	1,4	1,3	1,3	1,3	2	2	2,1	1,5
5 Speisemöhren 1kg	0,9	0,9	0,9	0,9	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,8	0,8	0,8	1	1	1	0,9
6 Bananen 1kg	1,4	1,4	1,4	1,4	1,1	1,1	1,1	1,1	1,3	1,3	1,3	1,1	1,1	1,1	1,3	1,2	1,1	1
7 Limetten 4 Stück	1,5	1,5	1,5	1,5	1,3	1,3	1,3	1,3	1,5	1,5	1,5	1,7	1,7	1,7	1,6	1,7	1,7	1,4
8 Milch 4L	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	2,8	2,8	2,8	2,9	2,9	2,9	3,4	3,5	3,1	3
9 Margarine 500g	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,7	0,7	0,7	0,7	0,7	0,7	0,9	0,8	0,8	0,5
10 Käse 150g	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,7	1,7	1,7	1,6	1,6	1,6	1,6	1,5	1,5	1,4
11 Salami	1,7	1,7	1,7	1,7	1,5	1,5	1,5	1,5	1,6	1,6	1,6	1,6	1,6	1,6	1,7	1,7	1,7	1,6
12 Rapsöl 750ml	2,5	2,5	2,5	2,5	2	2	2	2	2,2	2,2	2,2	2,3	2,3	2,3	2,2	2,1	2	1,8
13 Marmelade	2	2	2	2	1,8	1,8	1,8	1,8	1,6	1,6	1,6	1,7	1,7	1,7	2,1	2,2	2,1	1,7
14 Frischkäse 125g	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	1	1	1	1,1	1,1	1,1	1	1,2	1,1	0,8
15 Joghurt 450g	0,9	0,9	0,9	0,9	0,8	0,8	0,8	0,8	1	1	1	0,9	0,9	0,9	1,2	1,3	1,4	0,9
16 Nudeln 380g	2,5	2,5	2,5	2,5	2,3	2,3	2,3	2,3	2,2	2,2	2,2	2,2	2,2	2,2	2,4	2,6	2,3	2
17 Toast	1	1	1	1	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,8	0,8	0,8	1	1	1	0,9
18 Brot	2,5	2,5	2,5	2,5	2,3	2,3	2,3	2,3	2	2	2	2,2	2,2	2,2	2	2	2,4	2,2
19 Funny-Frisch Chipsfrisch	1,5	1,5	1,5	1,5	1,3	1,3	1,3	1,3	1,5	1,5	1,5	1,4	1,4	1,4	1,5	1,5	1,4	1,3
20 Kinder Schokolade 125 g	1,1	1,1	1,1	1,1	10000	10000	10000	10000	10000	10000	10000	1	1	1	1	0,9	1,1	10000
21 Haribo 200g	0,7	0,7	0,7	0,7	10000	10000	10000	10000	0,6	0,6	0,6	0,5	0,5	0,5	0,8	0,9	0,7	0,6
22 Prinzenrolle	1,6	1,6	1,6	1,6	10000	10000	10000	10000	1,3	1,3	1,3	1,4	1,4	1,4	1,7	1,9	1,8	1,4
23 Coca Cola 1L	1	1	1	1	0,9	0,9	0,9	0,9	0,8	0,8	0,8	10000	10000	10000	1	1,1	1,1	0,8
24 Carolinen Medium 12L	6,5	6,5	6,5	6,5	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	6	6,3	7,5	10000
25 Carolinen Apfelschorle 12L	10	10	10	10	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	9	10,5	11	10000
26 Krombacher 4 L	7	7	7	7	5	5	5	5	10000	10000	10000	10000	10000	10000	6	7,5	6	10000

Im weiteren Verlauf soll kurz auf zwei detektierte Probleme innerhalb der Testdurchläufe eingegangen werden.

Es wurde festgestellt, dass das Programm keine optimale Einkaufsplanung ermittelt bzw. ausgibt, sobald nur ein LEH-Geschäft ausgewählt wird oder die Wahl nur eines LEH-Geschäftes zur optimalen Einkaufsroute führen würde. Jedoch soll das Programm auch eine Route für die Wahl nur eines LEH-Geschäftes ermitteln oder auch nur ein LEH-Geschäft auswählen, falls es am Optimalsten ist nur ein LEH-Geschäft aufzusuchen (bei z.B. einer sehr kleinen Einkaufsliste). Dieses Problem konnte von Ahmad behoben werden, indem wir verschiedene Änderungen im Modell vorgenommen haben.

Wie oben beschrieben muss das Problem von entstehenden Subtouren berücksichtigt werden. Hierbei wurden als optimale Einkaufsroute zwei nicht verbundene Einkaufsrouten ausgewählt – dies bedeutet, dass z.B. eine Einkaufsroute im Wohnort gestartet wird und über Filiale 1 und 2 zurück zum Wohnort führt, in einer zweiten nicht verbundenen Route wird in Filiale 3 gestartet und über Filiale 4 und 5 in Filiale 3 zurückgekehrt. Dieses Ergebnis stellt keine korrekte Lösung des Modells zur optimalen Einkaufsplanung dar. Deshalb wurde von Ahmad eine Schleife eingebaut, welche zur Eliminierung von Subtouren führt (siehe Kapitel 3.1.6).

Aufgrund des fehlenden Fachwissens ist es uns leider nicht möglich eine tatsächlich funktionierende App mit allen technischen Anforderungen zu erstellen. In Kapitel 4 ist daher eine mögliche Version dargestellt, wie nach unseren Vorstellungen die Einkaufs-App aussehen und funktionieren könnte.

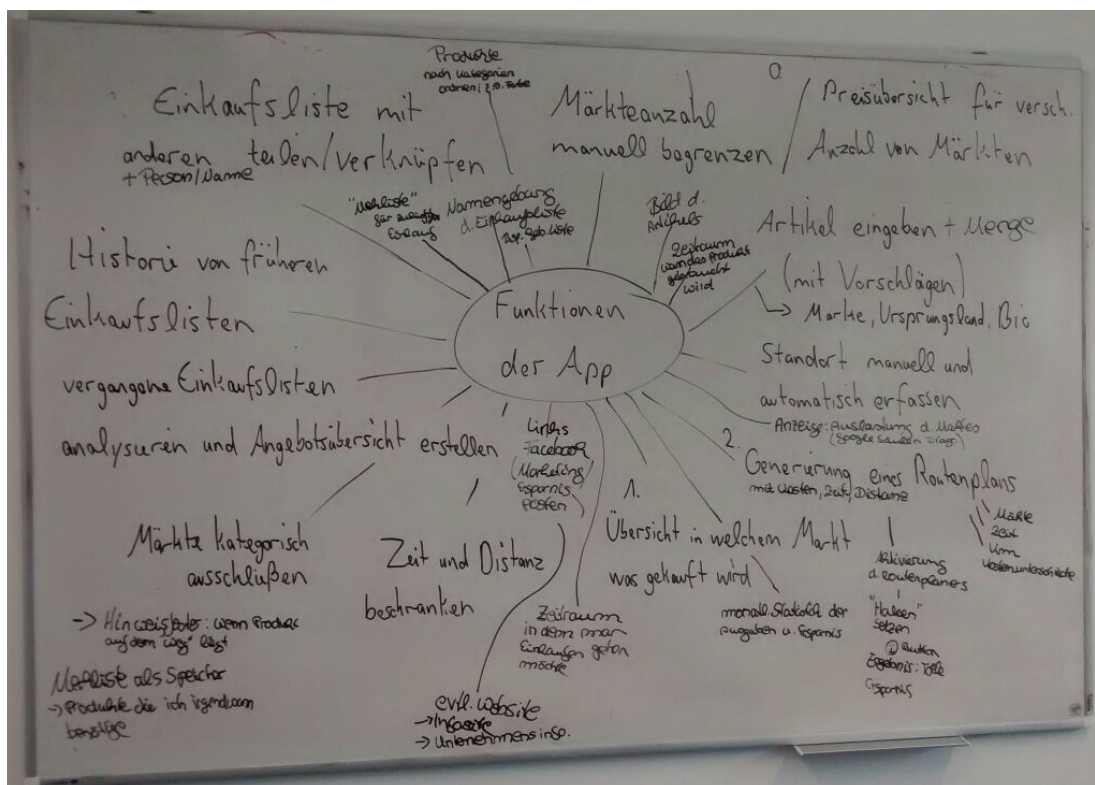
## 4 Die App

### 4.1 App-Konzipierung

In diesem Kapitel wird der gesamte Prozess des dritten Arbeitspaketes bzgl. der Ideenfindung und Konzipierung einer Smartphone App dargestellt. In 4.1.1 wird die erste Phase der App-Konzipierung beschrieben, welche das Brainstorming zur Findung der elementaren Funktionen der App und der Erarbeitung einer ersten groben Skizzierung der App-Oberfläche umfasst. Darüber hinaus wird in Kapitel 4.1.3 das Datenmodell dargestellt, welches die Beziehungen zwischen den verschiedenen Objekten der App beschreibt. Im nächsten Schritt wird in Kapitel 4.2 auf Basis dieser Ausarbeitungen ein Mock-up mittels „balsamiq“ erstellt. Dieses dient als Vorlage für die dritte Phase der Konzipierung (Kapitel 4.3), welche die Implementierung eines App Prototypen mittels „InVision“ behandelt.

#### 4.1.1 Funktion

In einem ersten Schritt wurde ein Brainstorming bzgl. der gesamten Funktionen einer entsprechenden Smartphone-App durchgeführt. Jedes Teammitglied sollte alle Vorstellungen und Gedanken bzgl. einer App nennen und diese auf einem Whiteboard sammeln (siehe Bild 4-1). Im Anschluss wurden vier Bedienungsoberflächen definiert – 1. Einkaufsliste, 2. Merkliste, 3. Historie/Statistiken, 4. Einstellungen. Die verschiedenen Funktionen und Ideen, welche aus dem Brainstorming hervorgegangen sind wurden diesen vier Oberflächen zugeordnet.

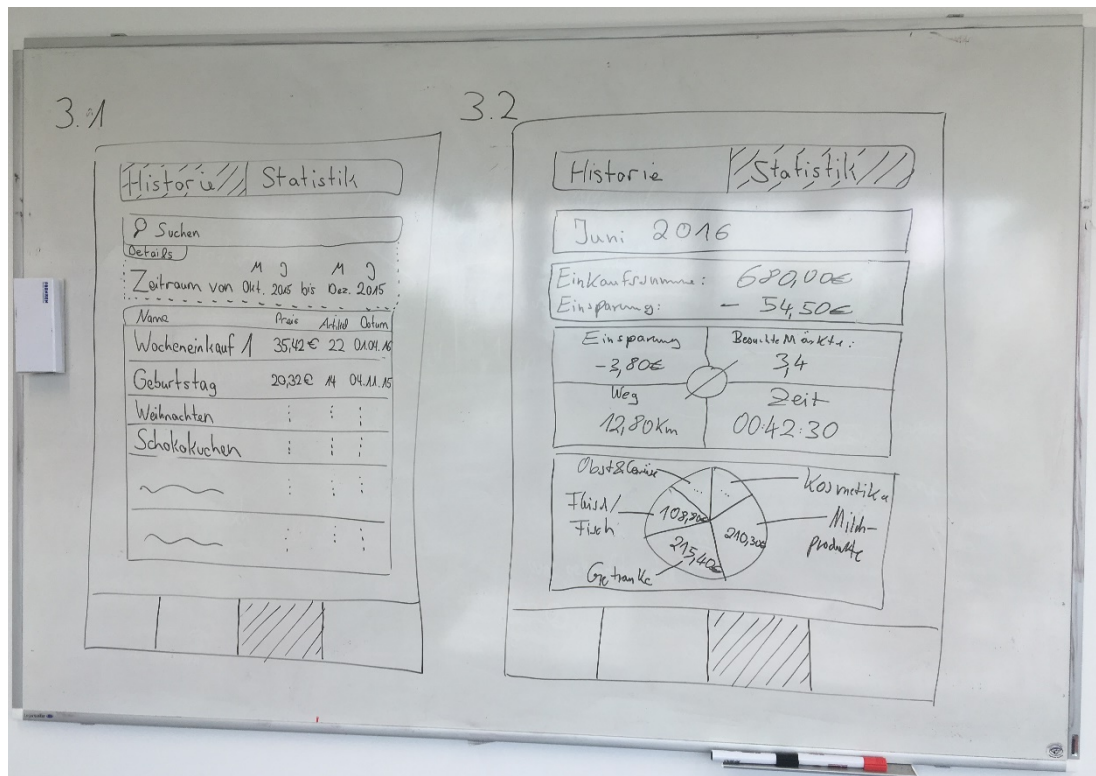


**Bild 4-1: Funktionen der Smartphone-App - Ergebnis des Team-Brainstormings**

#### 4.1.2 Schematischer Aufbau

In einem nächsten Schritt wurde auf Basis der vier Oberflächen und der zugewiesenen Funktionen ein erster grober schematischer Aufbau der App-Oberfläche auf dem Whiteboard erarbeitet. Für jede Oberfläche und die jeweiligen Unterbereiche wurde eine solche

Skizze angefertigt und nach den Vorstellungen des Teams ausgearbeitet. In Bild 4-2 ist beispielhaft die Skizze für die 3. App-Oberfläche – Historie/Statistik – dargestellt.



**Bild 4-2: Schematischer Aufbau der dritten App-Oberfläche - Historie und Statistik**

Abschließend wurden diese Zeichnungen dokumentiert, um diese weiterfolgend als Vorlage für die Konzipierung eines Mock-up zu nutzen.

Wie diese Daten und Informationen für die zu erstellende App in Verbindung stehen und gespeichert werden, soll das im nächsten Abschnitt erläuterte Datenmodell zeigen.

### 4.1.3 Datenmodell

Der Entity-Relationship-Ansatz (ERM) oder auch Objektbeziehungsmodell dient als Informationssystem und beschreibt einen Ausschnitt der realen Welt. Im Rahmen der Modellierung werden die folgenden Typ-Ausprägungen definiert:

1. Entitätstyp: beliebige Objekte
2. Beziehungstyp: Verknüpfung oder Zusammenhang zwischen den Entitäten
3. Attribut: die Eigenschaft

Die Entitäten werden identifiziert und die Beziehungen, welche für das Unternehmen bzw. die App wichtig sind, werden miteinander in Verbindung gesetzt. Weiterhin wird der Entwurf einer Datenbank auf einfache und verständliche Art und Weise grafisch dargestellt [CK1991, S. 35 ff.].

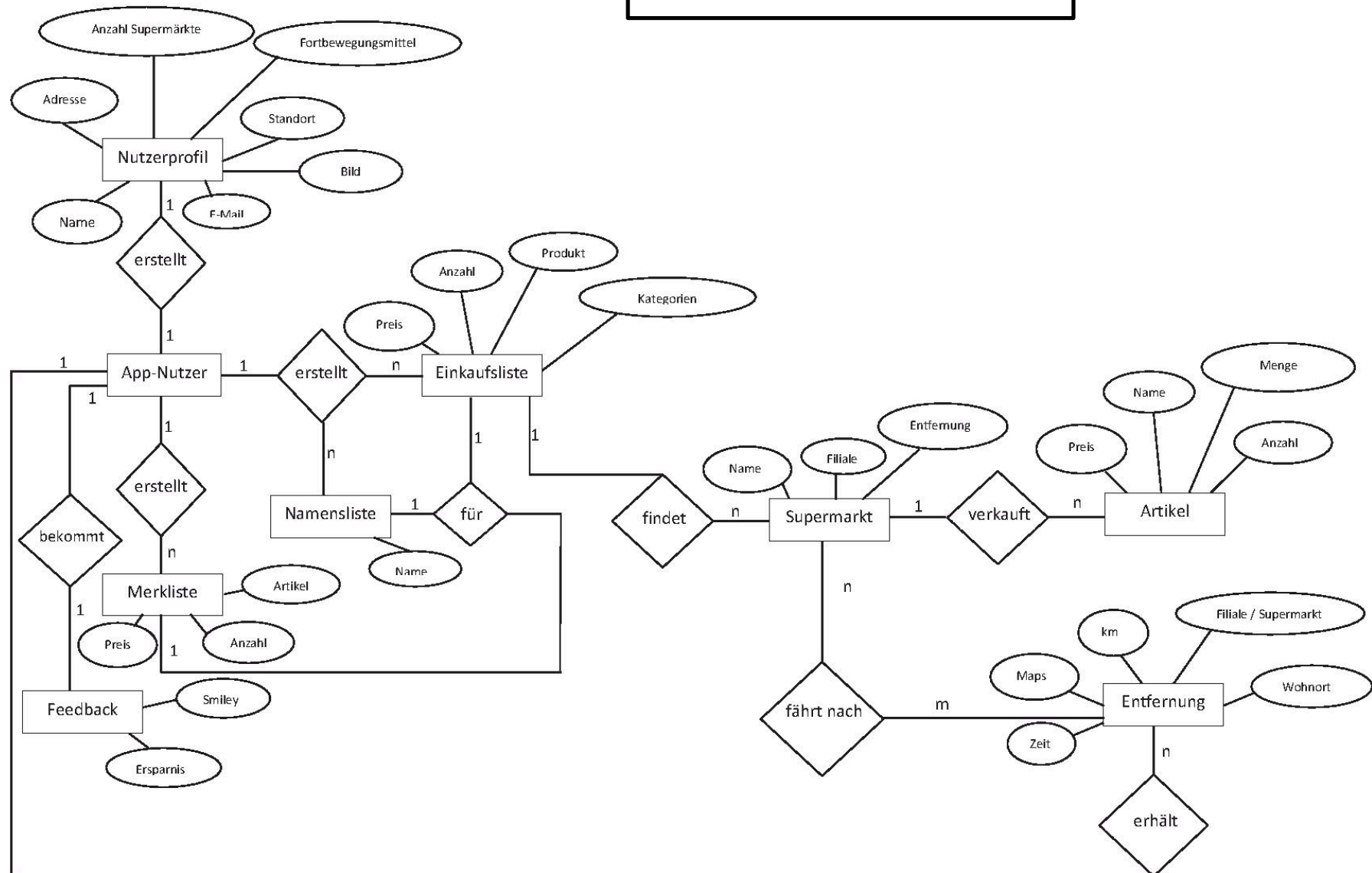
Das folgende Entity-Relationship-Diagramm (Bild 4-3) zeigt den Informationsprozess bzw. die Funktion der erstellten Einkaufs-App. Der Funktionsbaum ist netzwerkartig aufgebaut und stellt hierarchische Beziehungen der einzelnen Funktionen in fachlich logischer Sicht dar. Die Rechtecke beschreiben das Objekt bzw. die Entität, die Kreise die Eigenschaften und die Rauten die bestehende Beziehung.

Der App-Nutzer kann beispielsweise ein Profil erstellen. Eigenschaften des Nutzerprofils sind u.a. der Name, die E-Mail Adresse, der Standort, das Fortbewegungsmittel und die Anzahl der anzufahrenden Supermärkte. Diese Eigenschaften werden in einer Datenbank gespeichert.

Weiterhin hat der App-Nutzer die Möglichkeit Merklisten sowie Namenslisten für die Einkaufsliste zu erstellen. Die Merkliste ist für bevorstehende Einkäufe gedacht und speichert Produkte, die der Nutzer in Zukunft kaufen möchte. Eine Namensliste kann zum Beispiel für einen Wocheneinkauf oder einen Geburtstag erstellt werden. Die Einkaufsliste enthält die Eigenschaften der Produkte sowie Informationen bezüglich der Preise und Mengen.

Nachdem die Einkaufsliste erstellt ist, findet die App entsprechende LEH-Geschäfte, welche die Artikel der erstellten Einkaufsliste verkauft. Darüber hinaus erhält der App-Nutzer Informationen zur Entfernung der LEH-Geschäfte.

Weiterhin bestimmen die Kardinalitäten die Anzahl an Entitäten, welche in Beziehung zueinanderstehen. Insgesamt gibt es drei solcher Kardinalitäten. Hier kann es eine 1:1 Beziehung geben, eine 1:n oder eine n:m Beziehung [Kel05, S. 15 ff.]. Beispielsweise kann ein Artikel in mehreren LEH-Geschäften verkauft werden. Jedes LEH-Geschäft verkauft mehrere Artikel. Somit gibt es verschiedene Kombinationen zwischen den beteiligten Entitäten, wobei der Beziehungstyp diese Anzahl beschränkt.

**Entity-Relationship-Diagramm****Bild 4-3: Datenmodell**

## 4.2 Mock-up

Zu diesem frühen Zeitpunkt im Produktentstehungsprozess ist es nicht sinnvoll direkt mit der Programmierung der App zu starten, da es noch zahlreiche Änderungen am Design und am Funktionsumfang geben wird. Vielmehr sollte durch eine einfache computergenerierte Veranschaulichung der App potenziellen Nutzern die Möglichkeit gegeben werden, die App zu testen und Verbesserungen vorzuschlagen. Dadurch entsteht ein Prozess, der den Bedienkomfort schrittweise erhöht und möglicherweise auch zur Implementierung weiterer nützlicher Features führt. Darüber hinaus können potenzielle Investoren einfacher von dem Konzept der App überzeugt werden [Bal16-ol].

Daher wurde aufbauend auf den Skizzen aus Kapitel 4.1.2, welche die grundlegende Bedienstruktur bzw. den gesamten Funktionsumfang der App zeigen, ein Mock-up erstellt. Zur Anfertigung wurde das Tool „balsamiq“ genutzt, welches genau für diesen Anwendungsfall konzipiert wurde.

Das Tool beinhaltet verschiedene Vorlagen, die zur Erstellung von Desktop-Apps, Mobile Apps, Web Sites und vielen weiteren Anwendungen eingesetzt werden können. Nach Auswahl der entsprechenden Vorlage entsteht das Design sowie die Bedienstruktur durch vorgefertigte User Interface (UI) Komponenten (ON/OFF Switch, iPhone Menü, Check Box, etc.), die per Drag & Drop an der entsprechenden Stelle positioniert werden können. Zur Erweiterung des Detailgrades enthält das Tool außerdem eine große Anzahl an Icons, die in hochauflösender Qualität zur Verfügung stehen. Darüber hinaus besteht die Möglichkeit auch eigene Bilder hinzuzufügen, die entweder aus den eigenen Dateien ausgewählt werden können oder direkt per Internet-Link eingefügt werden können. Zur Implementierung der Bedienstruktur können verschiedene Elemente mit Links versehen werden, um durch Klicken automatisch auf eine andere Seite zu springen.

Auf diese Weise entsteht auf eine sehr zeiteffiziente Art und Weise ein Mock-up der App, welches einen sehr realitätsnahen Eindruck des Designs, des Funktionsumfangs sowie der Bedienstruktur ermöglicht. Änderungen und Ergänzungen können sehr einfach hinzugefügt werden, sodass die App kontinuierlich verbessert werden kann. Eine weitere nützliche Zusatzfunktion besteht darin, dass die App automatisch in zwei verschiedene Designs konvertiert werden kann. Das sehr einfache und wie eine Bleistiftskizze wirkende Sketch-Design unterstützt ein schnelles Brainstorming optimal und das saubere und realitätsnahe Wireframe-Design eignet sich optimal für Präsentationszwecke. Darüber hinaus bietet das Tool die Möglichkeit in einen Präsentationsmodus zu wechseln. Dieser enthält zusätzliche Funktionen wie z.B. einen großen Zeiger, um konkret auf einzelne Elemente hinzuweisen. Des Weiteren gibt es zwei verschiedene Export-Möglichkeiten (PNG und pdf), um das Design der App mit anderen zu teilen. An dieser Stelle muss allerdings berücksichtigt werden, dass die Qualität der Darstellung durch den Export erheblich abnimmt, wodurch eine Verwendung in Präsentationen nicht empfehlenswert ist.

Insgesamt wurden 22 Seiten der App angefertigt, auf denen die wichtigsten Funktionen enthalten sind und auf diese Weise einen guten Einblick in das Konzept der App ermöglichen. Als Beispiel ist eine Seite jeweils im Sketch (Bild 4-4) sowie im Wireframe-Design (Bild 4-5) dargestellt. Zusammenfassend stellt Tabelle 4-1 die Stärken und Schwächen von balsamiq gegenüber.



Bild 4-4: Sketch-Design



Bild 4-5: Wireframe-Design

Tabelle 4-1: Stärken und Schwächen von balsamiq



## Stärken:

- Intuitive Bedienbarkeit
- Sehr große Anzahl an Icons
- Einfaches Hochladen von eigenen Fotos
- Automatische Konvertierung von Sketch- in Wireframe-Design
- Einfache Verlinkung zwischen verschiedenen Seiten
- Präsentationsmodus mit nützlichen Features (Pfeil, Kommentare, etc.)



## Schwächen:

- Export-Funktion in sehr schlechter Qualität (in Bearbeitung)
- Kalibrierung von eigenen Fotos mit sehr geringem Funktionsumfang (kein Beschneiden, kein Sperren der Seitenverhältnisse)
- Nicht auf aktuellstem Stand (iPhone 5 aktuellstes Design)
- Keine Möglichkeit das Mock-up direkt auf dem Smartphone zu betrachten



### 4.3 Prototyp

Im folgenden Kapitel wird die Umsetzung eines ersten Prototyps der App beschrieben. Aufbauend auf dem zuvor erstellten Mock-up besteht das Ziel dieses Arbeitspaketes darin, die App auf dem Smartphone auf realitätsnahe Art und Weise testen zu können. Dadurch können verschiedene Personen die App vor dem Hintergrund der intuitiven Benutzung beurteilen und Verbesserungsvorschläge einbringen. Zur Unterstützung der Prototyp-Erstellung wurden vier verschiedene Tools untersucht, die alle die Möglichkeit bieten, eine App auf dem Smartphone zu testen. Allerdings unterscheiden sich die Tools im Hinblick auf den zeitlichen Aufwand und den benötigten Programmierkenntnissen stark voneinander.



**Bild 4-6: Vergleich geeigneter Tools zur Erstellung eines Prototypen**

Da sich der Reifegrad der App in einer frühen Phase befindet und daher noch mit einigen Änderungen zu rechnen ist, sollte die Flexibilität und die Geschwindigkeit der Anpassung möglichst hoch sein. Da diese Kriterien gemäß der Vor- und Nachteile, die in der obigen Grafik dargestellt sind (Bild 4-6), am besten von „InVision“ erfüllt werden, wird dieses Tool für die Erstellung des Prototyps verwendet [Inv16-ol], [Sol16-ol], [MIT16-ol], [xCo16-ol].

Die grundlegende Funktionsweise des Tools besteht darin vorgefertigte Designs verschiedener Oberflächen einer App (Website etc.) miteinander zu verknüpfen und durch die zugehörige InVision-App auf dem Smartphone anzuzeigen.

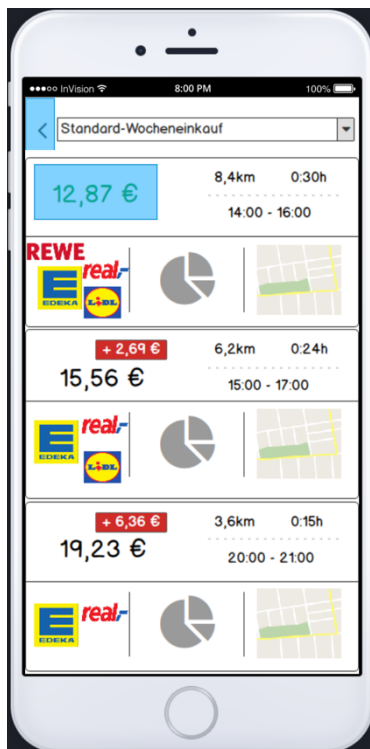
Zunächst wird ein neues Projekt erstellt, wobei die Wahl zwischen einer Website, einer Smartphone-App und vielen weiteren Oberflächen besteht. Nach Auswahl der Smartphone-App wird das entsprechende Endgerät festgelegt, damit das Tool die notwendige Skalierung des Bildschirms berücksichtigt. Eine Tabelle mit der Angabe der verschiedenen Abmessungen (Breite ist am wichtigsten, Länge kann noch angepasst werden) für die verschiedenen Gerätegrößen wird leider nicht bereitgestellt – allerdings lassen sich diese Angaben sehr schnell in unterschiedlichen Blogs im Internet finden. Nun werden die unter-



schiedlichen Ansichten der App benötigt, um das Projekt mit Inhalt zu füllen. Die Importmöglichkeiten reichen dabei von .jpeg-Formaten über das .png-Format bis hin zur direkten Anbindung an AdobePhotoshop. An dieser Stelle können die im Mock-up erstellten Designs verwendet werden. Allerdings muss zuvor die Skalierung angepasst werden, um eine einwandfreie Darstellung auf dem Smartphone zu gewährleisten. Anschließend können die Ansichten der App über eine Exportfunktion von balsamiq als .png-Datei exportiert werden und per Drag & Drop bei InVision eingefügt werden. Schließlich können die verschiedenen Ansichten nun durch Verlinkung miteinander verknüpft werden, sodass die Benutzung der App sehr realitätsnah getestet werden kann. Dazu wird eine Vorschau der Smartphone-App angezeigt und es lassen sich Bereiche auf dem Bildschirm definieren, die mit einer bestimmten Aktion belegt werden können, um auf eine andere Ansicht der App zu wechseln. Der Funktionsumfang orientiert sich hierbei sehr nah an den bekannten Gesten, die üblicherweise in Smartphone-Apps verwendet werden (Wischen, Doppel-Klick, einfacher Klick, etc.).

Außerdem können Bereiche oben und unten fixiert werden, um zum einen gewisse Bereiche einzufrieren und zum anderen um den gesamten Inhalt der Seite darzustellen. Falls der Inhalt zwischen diesen Bereichen größer ist als der verbleibende Sichtbereich des Smartphones, erstellt InVision automatisch einen Bereich, der später in gewohnter Weise durch Scrollen erreicht werden kann. Auf diese Weise entsteht sukzessiv eine detaillierte Simulation einer App, die in einer beliebigen Komplexität angefertigt werden kann. Zu Testzwecken kann ein Link zur InVisionApp auf dem Smartphone per SMS an eine beliebige Nummer verschickt werden. Nach Betätigung des Links wird das aktuelle Projekt automatisch in der App geöffnet und die Ergebnisse können direkt auf dem Smartphone betrachtet werden (Bild 4-7).

Die anschließenden Änderungen, die online auf der Website vorgenommen werden, können von jetzt an direkt in der App mit einem Wisch nach unten aktualisiert werden



**Bild 4-7:** Ausschnitt aus dem Prototyp

Zusammenfassend kann festgehalten werden, dass das Tool sehr intuitiv bedient werden kann und die Kriterien der Flexibilität und einer schnellen Umsetzung von neuen Ideen/Verbesserungsvorschlägen optimal erfüllt werden. Auch die Kombination mit dem balsamiq-Tool hat einwandfrei funktioniert, sodass eine Kombination beider Tools ausdrücklich empfohlen werden kann. Außerdem bietet InVision eine Vielzahl an weiteren Funktionen, die in unserem Projekt noch nicht voll ausgeschöpft wurden.

Eine zusätzliche herausragende Eigenschaft besteht darin, dass sich das Tool optimal zur Gruppenarbeit eignet – es können Teams gebildet werden, die sich online zu Live-Demos zusammenschalten können, bei denen eine Ansicht erscheint, die die App zeigt und den Mauszeiger der jeweiligen Mitglieder. Zusätzlich gibt es eine Chat-Funktion, mit der dann neue Ideen geteilt werden können oder die bisherigen Oberflächen beurteilt werden können. Auf diese Weise können Meetings ortsunabhängig sehr zügig durchgeführt werden, wodurch zeitaufwendige Missverständnisse vermieden werden können, die sich zu Lasten der Entwicklungszeiten auswirken würden.

## 5 Fazit & Ausblick

Abschließend ist zu sagen, dass das Projekt „Entscheidungsunterstützung bei der Einkaufsplanung“ erfolgreich abgeschlossen werden konnte.

Auch wenn alle vier Gruppenmitglieder in unterschiedlichen Fachrichtungen studieren, hat jeder bzw. jede auf seine eigene Art und Weise dazu beigetragen, das Projekt entsprechend voranzutreiben. Mehrmalige Gruppentreffen in der Woche und ständige Kommunikation waren gefordert. Kommuniziert wurde über eine eingerichtete WhatsApp Gruppe sowie mit unserem Gruppenleiter über Koala bzw. via E-Mail. Eine Dropbox diente zur Speicherung der Dokumente und Entwürfe. Getroffen haben wir uns im zur Verfügung gestellten Seminarraum des Lehrstuhls. In mehreren Stunden der Woche wurde diskutiert und erörtert wie die zu erstellende App entsprechend aussehen und funktionieren könnte. Am Whiteboard, sowie mittels Beamer und Laptop wurden in der Gruppe Daten und Fakten zur App sowie die mathematische Modellierung skizziert und erläutert. Eine Mind-Map gab zu Beginn der Projektphase Aufschluss, welche Funktionen die App haben sollte. Auch ein Projektplan und Meilensteinplan wurde für die bessere Planung und Organisation erstellt. In der Zwischenpräsentation und Endpräsentation wurden diese entsprechend vorgestellt. Hauptschwerpunkt sahen wir zu Beginn in der Modellierung des mathematischen Modells. Alle Gruppenmitglieder wurden integriert und informiert, wie und mit welchen Programmen zu arbeiten ist, um die geforderten Matrizen bzw. Restriktionen aufzustellen.

Ziele und Aufgaben die zu Beginn des Projekts definiert wurden, sind erfolgreich in der Gruppe umgesetzt worden. Dazu zählen u.a. die Marktanalyse und die Auswahl an Supermärkten sowie die Aufstellung und Implementierung des mathematischen Modells. Außerdem wurde ein Konzept für die Funktionsstruktur ausgearbeitet und durch ein Mock-up sowie einen Prototyp visualisiert.

Ebenfalls war die Motivation in der Gruppe sehr gut. Dies mag daran liegen, dass am Ende ein Prototyp entstanden ist, welcher sehr anschaulich dargestellt wird. Die Idee eine App zu erstellen, welche unter Berücksichtigung der Preise und der jeweiligen Distanzen zwischen den Supermärkten ein Kostenoptimum finden und anzeigen soll, ist von den Zuhörern während der Präsentationen positiv bewertet worden.

Chancen sehen wir beim mathematischen Modell in der optimalen Einkaufsplanung und bei der Berücksichtigung von Einkaufspräferenzen. Der Nutzer hat die Möglichkeit seinen individuellen Einkauf zu planen und anzeigen zu lassen.

Handlungsbedarf sehen wir in der Datenbeschaffung (siehe Abschnitt 2.2) von dauerhaft, aktuellen Produktpreisen. Hier erscheint es notwendig, ein Konzept zu entwickeln, welches nicht direkt den Einzelhandel mit einbindet. Hier könnte der Einkäufer selbst helfen, indem er nach seinem Einkauf die Kassenzettel zeitnah einscannt. So könnten die Produktpreise in unsere Datenbank gelangen und wären auf dem aktuellen Stand.

Des Weiteren sollte in der Zukunft die Smartphone-App vollständig konzipiert werden.

---

## Literaturverzeichnis

- [Apr16-ol]      Apriku: Was ist Apriku. Unter: <https://apriku.de/#!/uberuns>, letzter Abruf: 25.05.2016.
- [Bal16-ol]      balsamiq: Balsamiq Mockups. Unter: <https://balsamiq.com/products/mockups/>, letzter Abruf: 29.07.2016.
- [BEE04]      Berekoven, Ludwig / Eckert, Werner / Ellenrieder: Marktforschung. Methodische Grundlagen und praktische Anwendung. 10. Auflage. Gabler, Wiesbaden, 2004.
- [Bes16-ol]      Besorger. Unter: <http://www.besorger-app.de/de/>, Abruf: 27.05.2016.
- [Bri16-ol]      Bring. Unter: <https://www.getbring.com/#!/app>, Abruf: 27.05.2016.
- [CK1991]      Chen, Peter P.S. / Knöll, Heinz-Dieter: Der Entity Relationship-Ansatz zum Logischen Systementwurf. Datenbank- und Programmwurf. Bibliographisches Institut & F.A. Brockhaus AG, Mannheim, 1991.
- [GI05]      Grünert, Tore / Irnich, Stefan: Optimierung im Transport. Band 2: Wege und Touren. Shaker Verlag, Aachen, 2005.
- [Google16-ol]      Google: Google Maps API. Unter: <https://developers.google.com/maps/?hl=de>, letzter Abruf: 08.07.2016.
- [Inv16-ol]      InVision: Unter: <https://www.invisionapp.com>, letzter Abruf: 15.08.2016.
- [Kai13]      Kairies, Peter: So analysieren Sie Ihre Konkurrenz. Konkurrenzanalyse und Benchmarking in der Praxis. 9. Auflage. Expert, Renningen, 2013.
- [Kel05]      Kelter, Udo: Datenmodellierung mit ER-Modellen. 2005.
- [Kre1987]      Kreilkamp, Edgar: Strategisches Management und Marketing. Markt- und Wettbewerbsanalyse, Strategische Frühaufklärung, Portfolio-Management. Walter de Gruyter, Berlin, New York. 1987.
- [MIT16-ol]      MIT App Inventor: Unter: <http://appinventor.mit.edu/explore/>, letzter Abruf: 15.08.2016.
- [Sol16-ol]      ZURB Solidify: Unter: <http://www.solidifyapp.com>, letzter Abruf: 15.08.2016.
- [Spa16-ol]      Sparpionier. Unter: <http://sparpionier.com/index.php?PHPSESSID=nv0dadddbku09qclu6vou7lo55>, letzter Abruf: 27.05.2016.
- [Stern15-ol]      stern: Rewe dreht an der Preisschraube. Unter: <http://www.stern.de/wirtschaft/news/rewe-lieferservice--preise-wie-im-markt-sind-vorbei-3463740.html>, 22.01.2015, letzter Abruf: 08.07.2016.
- [Vzhh12-ol]      Verbraucherzentrale Hamburg: Verschleierungstaktik im Supermarkt. Unter: <http://www.vzhh.de/ernaehrung/248409/verschleierungstaktik-im-supermarkt.aspx>, 20.06.2012, letzter Abruf: 08.07.2016.
- [xCo16-ol]      xCode 8 (Swift 3): Unter: <https://developer.apple.com/xcode/>, letzter Abruf: 15.08.2016.

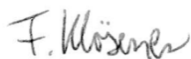
## Eidesstattliche Erklärung

Wir, Ahmad Hashemi, Felix Klösener, Timo Schäfers und Agnes Schützenmeister, versichern durch eigenhändige Unterschrift, dass wir die vorliegende Arbeit selbstständig und ohne unerlaubte Hilfe Dritter angefertigt haben. Alle Stellen, die inhaltlich oder wörtlich aus Veröffentlichungen stammen, sind kenntlich gemacht. Diese Arbeit lag nach unserem Informationsstand in gleicher oder ähnlicher Weise noch keiner Prüfungsbehörde vor und wurde bisher noch nicht veröffentlicht. Wir sind uns darüber bewusst, dass bei Abgabe einer falschen Erklärung die Prüfung als nicht bestanden gilt. Im dringenden Verdachtsfall kann unsere Arbeit unter Zuhilfenahme des Dienstes „Turnitin“ geprüft werden. Dabei erlauben wir die Ablage unserer Arbeit im institutsinternen Speicher. Unabhängig vom Ergebnis der Prüfung durch „Turnitin“ wird immer eine individuelle Prüfung und Bewertung der Arbeit vorgenommen. Darüber hinaus wird der Inhalt der Arbeit Dritten nicht ohne unsere ausdrückliche Genehmigung zugänglich gemacht.

Paderborn, 31. August 2016

---

Ahmad Hashemi



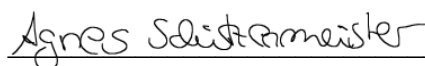
---

Felix Klösener



---

Timo Schäfers



---

Agnes Schützenmeister