

Konzeption und prototypische Implementierung konsumenten- zentrierter Gastronomie- Transaktionen mit Smartphones

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Dominik Gruber

Matrikelnummer 0926288

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Thomas Grechenig

Wien, 02.05.2012

(Unterschrift Verfasser)

(Unterschrift Betreuer)



Konzeption und prototypische Implementierung konsumenten- zentrierter Gastronomie- Transaktionen mit Smartphones

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Dominik Gruber

0926288

ausgeführt am

Institut für Rechnergestützte Automation

Forschungsgruppe Industrial Software

der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Thomas Grechenig

Wien, 02.05.2012

Erklärung

Dominik Gruber
Robert-Hamerling-Gasse 30/3
1150 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, am 02.05.2012

Dominik Gruber

Danksagung

Ohne meine Eltern, *Wolfgang und Elisabeth Gruber*, die mich in jeglicher Hinsicht immer zu 100 % unterstützt haben, hätte ich dieses Studium niemals absolvieren können. Danke für all die Möglichkeiten, die ihr mir dadurch eröffnet habt! Ich bin euch dafür mehr dankbar, als Worte es ausdrücken können.

Dank gilt meinem Betreuer *Thomas Grechenig*, der mir die Chance gab, ihn von diesem Konzept zu überzeugen und mir dadurch ermöglichte, an einem spannenden Projekt zu arbeiten.

Besonderer Dank gilt außerdem *Gerald Madlmayr*, der mir mit Rat und Tat stets freundlich und hilfreich zur Verfügung gestanden ist und immer an diese Idee geglaubt hat.

Für seinen Input zu sämtlichen Aspekten des Designs möchte ich mich herzlich bei *Clemens Schrammel* bedanken, der auf meine Fragen und Bitten immer rasch reagiert hat.

Weiters möchte ich allen Personen danken, die meinem Aufruf zur Teilnahme an der soziologischen Studie gefolgt sind, und besonders bedanken möchte ich mich bei den Personen, die sich für den Usability Test zur Verfügung gestellt und wertvolles Feedback geliefert haben: *Mariana Bernasconi*, *Martin Leonhartsberger-Schrott*, *Manuel Mayrhofer*, *Philipp Naderer* und *Luiza Puiu*.

Großer Dank gilt *Ernesto Gelles*, dessen ursprüngliche Idee ich zu dem hier vorliegenden System entwickeln konnte.

Abschließend möchte ich meiner Freundin *Mariana Bernasconi* danken, die mich stets unterstützt und immer dafür Verständnis gezeigt hat, dass ich oftmals mehr Zeit mit dieser Arbeit als mit ihr verbracht habe.

Inhaltsverzeichnis

Erklärung	iii
Danksagung	iv
Kurzfassung	viii
Abstract	ix
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Methodik	2
2 State-of-the-Art	4
2.1 Smartphones	4
2.1.1 Verbreitung	5
2.1.2 Betriebssysteme	5
2.1.3 Leistung	7
2.1.4 Nutzung	7
2.1.5 Tablets	8
2.2 Gastronomie	9
2.2.1 Analyse des Geschäftsprozesses Gastronomiebesuch . .	9
2.2.2 Identifizierung der Typen von Gastronomiebetrieben .	11
2.3 Wissenschaftlicher Kontext	11
2.4 Bestehende Systeme	13
2.4.1 POS-Systeme	13
2.4.2 Terminals	14
2.4.3 Bezahl-Systeme	15
2.4.4 Chipkarten	16
2.5 Online-Systeme	16
2.5.1 Kategorisierung	16
2.5.2 Evaluierung Online-Gesamtsysteme	17
2.5.3 Gesamtüberblick	21
3 Konzept Eigensystem	25

3.1	Grundgedanke	25
3.1.1	Art des Systems	25
3.1.2	Abwicklung einer Bestellung	26
3.1.3	Benutzer-Funktionalität	27
3.1.4	Vorteile des Systems	28
3.2	Architektur	28
3.2.1	Web Service	30
3.2.2	Smartphone-Client	31
3.2.3	Website	34
3.2.4	Administrations-Client	34
3.3	Bezahlung	35
3.3.1	Traditioneller Weg	35
3.3.2	Kreditkarte	36
3.3.3	Bankeinzug	38
3.3.4	paybox	39
3.3.5	E-Wallet	39
3.4	Check-In	40
3.5	Missbrauch	41
4	Soziologische Studie	43
4.1	Methodik	43
4.2	Durchführung	44
4.3	Auswertung	44
4.4	Ergebnis	51
5	Prototypische Implementierung	52
5.1	Entwicklungsprozess	52
5.2	Web Service	53
5.2.1	Konzeption	53
5.2.2	Architektur	54
5.2.3	Schnittstellen	56
5.2.4	Datenmodell	61
5.3	Smartphone-Client	63
5.3.1	Konzeption	63
5.3.2	Architektur	64
5.3.3	User Interface	68
6	Evaluierung Prototyp	75
6.1	Definition der Testmethode	75
6.2	Durchführung	76
6.3	Ergebnisse	77
6.3.1	Prototyp	77
6.3.2	Konzept	83

Inhaltsverzeichnis	vii
7 Zusammenfassung und Ausblick	85
A Fragenkatalog der soziologischen Studie	87
B Usability Test	94
B.1 Aufgaben	94
B.2 Fragebogen	97
Literaturverzeichnis	99
Literatur	99
Online-Quellen	101

Kurzfassung

In den letzten Jahren ist die Zahl jener Person, die einen leistungsstarken Computer mit Zugang zum Internet in ihrer Hosentasche tragen, rasant gestiegen. Sowohl in Europa als auch in Nordamerika sind mit Stand Mitte 2011 mehr als die Hälfte aller eingesetzten Mobiltelefone Smartphones – Tendenz steigend.

Diese Entwicklung eröffnet neue Möglichkeiten für diverse Industrien. Darunter auch für die Gastronomie, der sich diese Arbeit widmet. Es wird die Frage behandelt, in welcher Form die Geschäfts-Transaktionen mit aktuellen technischen Mitteln wie Smartphones durchgeführt werden können.

Zu Beginn wurde untersucht, welche technischen Lösungen aus Sicht des Konsumenten bereits existieren. Auf dieser Basis wurde ein Konzept für ein neuartiges System entwickelt, dessen Kern ein Smartphone Client ist. Ein wesentliches Feature ist, dass Benutzer während ihres Aufenthalts in einem Gastronomiebetrieb angebotene Artikel bestellen und bezahlen können. Der Hauptvorteil aus Sicht des Konsumenten ist der Zeitgewinn, da der Ablauf des Besuchs nun maßgeblich selbst gesteuert werden kann und die Abhängigkeit vom Servicepersonal, das nicht jederzeit verfügbar ist, entfällt. Einzig der zeitliche Aufwand, der im Restaurant für die Zubereitung und Zusammenstellung der Speisen anfällt, kann nicht beeinflusst werden.

Dieses Konzept wurde durch eine soziologische Studie auf Interesse untersucht. Als Kernergebnis wurde ermittelt, dass vier von fünf Personen in zumindest einer Art von Gastronomiebetrieb gerne eine Bestellung über ihr Smartphone tätigen würden und neun von zehn Befragten würden die Möglichkeit nutzen, Essen zur späteren Konsumation im Betrieb vorzubestellen.

Daraufhin wurde ein Prototyp des Smartphone Clients für das Betriebssystem *iOS* entwickelt, der die wesentlichen Features des Konzepts abbildet. Dieser Prototyp wurde einem Usability Test unterzogen, um einerseits Mängel festzustellen und andererseits Meinungen zum Konzept von Personen einzuholen, die bereits einen Prototyp des Clients, der einem realen System nahe kommt, benutzt haben. Der Großteil gab im Anschluss an den Test an, das Angebot eines Betriebs eher in Anspruch nehmen zu wollen, wenn dieser ein derartiges System einsetzt.

Abstract

Over the last few years there has been a rapid rise in the number of people carrying powerful computers in their pockets. By mid 2011, more than half of all mobile phones used in Europe and North America were smartphones.

The range of possibilities and opportunities opened up by this recent development is immense. This thesis focuses on the gastronomy industry in terms of addressing business transactions utilizing smartphones and similar technological advances.

At the beginning of this work researches and reviews were undertaken concerning products currently on the market which are used to simplify transactions with a focus on the consumer. Based on this research a concept for an innovative system with its core in smartphone client usage was developed. One of the key features of this system is that consumers are able to order and pay for their food without interaction with a waiter. The main advantage for consumers is to save time. The only element that can not be influenced is the time it takes to prepare the food.

A sociological study was then carried out researching the user acceptance of this concept. Four out of five participants in the study stated that they would be interested in ordering food with their smartphone while in a restaurant. Nine out of ten participants would preorder food for later consumption in a restaurant.

Once the interest in this concept has been established, a prototype of the smartphone client has been developed for the *iOS* operating system. With all significant features in place, a usability study was then undertaken. One of the aims of the aforementioned study was to find flaws in the prototype, while another aim was to receive valuable feedback not only on the usability but also on the concept as a whole. The majority of the surveyed expressed a greater willingness to choose a restaurant partaking in such a system.

Kapitel 1

Einleitung

Smartphones sind nicht nur Mobiltelefone. Smartphones sind die Geräte, die die Bezeichnung „Personal Computer“ rein von der Bedeutung her am meisten verdient hätten.

In Österreich sind 56 % aller eingesetzten Mobiltelefone Smartphones, in Nordamerika sind es bereits 63 % [72] [112]. Ein leistungsstarker Computer mit permanentem Internetzugang ist nun ständiger Wegbegleiter eines großen Teils der Bevölkerung dieser Nationen. Diese hohe und weiterhin steigende Verbreitung birgt enormes Potential für die Veränderung von Abläufen unterschiedlichster Art.

Diese Arbeit wählt die Geschäftsprozesse der Dienstleistungssparte Gastronomie und behandelt die Frage, wie diese mit Smartphones abgewickelt werden können. Dazu werden bestehende Lösungen analysiert, und auf dieser Basis wird ein innovatives Konzept entwickelt, das mit einem funktionalen Prototypen getestet wird.

1.1 Motivation und Problemstellung

Die Abläufe bei der Inanspruchnahme der Services eines Gastronomiebetriebs sind im Wesentlichen seit Jahrzehnten unverändert. Durch den Einsatz von Smartphones, also den Geräten, die Konsumenten beim Besuch eines Betriebs bereits bei sich tragen, könnten diese Abläufe neu gestaltet werden. Speziell steigern ließe sich die zeitliche Effizienz.

Der größte Zeitverlust entsteht momentan dadurch, dass beim Aufenthalt im Betrieb die Abwicklung der Geschäfts-Transaktionen komplett über das Servicepersonal erfolgt. Der Konsument muss warten, bis ihm die Speisekarte übergeben wird. Nach Auswahl der Artikel muss dieser warten, bis ihm ein Mitarbeiter zur Aufnahme der Bestellung zur Verfügung steht. Für die Abwicklung einer Bezahlung, die im Falle einer Gaststätte in der Regel nach der Konsumation der Speisen erfolgt, entsteht erneut Wartezeit, bis dafür Personal bereitsteht. All dies kann mangels sofortiger Verfügbarkeit

des Personals zu zeitlichen Verzögerungen führen und macht die Dauer des Aufenthalts in einem Gastronomiebetrieb schwer planbar.

Es existiert außerdem keine zentrale Plattform, auf der sich potentielle Kunden über die von Betrieben angebotenen Speisen sowie konkrete Angebote, wie z. B. Mittagsmenüs, informieren können. Weiters können Speisen nicht zur Konsumation im Betrieb für einen fixen Zeitpunkt vorbestellt werden, wofür in dieser Arbeit der Begriff „Dine-In“ verwendet wird. Würden alle Abläufe wie Auswahl des Betriebs und der gewünschten Artikel sowie deren Bestellung und Bezahlung bereits vor dem tatsächlichen Besuch stattfinden, könnte der Konsument sich im Betrieb mit dem eigentlichen Grund seines Erscheinens befassen: dem Essen.

Basierend auf dieser Ausgangssituation wird ein Konzept für ein System mit Smartphone-Clients entwickelt, das sich der beschriebenen Problematik annimmt. Es sollen die Prozesse aus Sicht der Konsumenten an mobile Endgeräte angepasst werden, um ein vorteilhaftes System für Kunden und Händler zu schaffen.

1.2 Methodik

Als Grundlage beschreibt Abschnitt 2.1 den technischen Entwicklungsstand von Smartphones sowie deren Verbreitung und das Nutzungsverhalten.

Als erster Schritt zum eigenen System werden die unterschiedlichen Abläufe im Geschäftsprozess zur Durchführung einer Bestellung analysiert und abgebildet. Darauf basierend werden die verschiedenen Typen von Gastronomiebetrieben definiert. Die Erkenntnisse werden durch Besichtigungen von Lokalitäten überprüft und in Abschnitt 2.2 präsentiert.

Durch Recherche werden aktuelle relevante technische Lösungen, die sich auf das Problem beziehen, identifiziert. Dabei kommt eine Bottom-Up-Vorgehensweise zum Einsatz: Es werden verschiedene Systeme einzeln betrachtet, die sich durch Merkmale wie innovative Features oder hohe aktive Benutzung auszeichnen. Als Ergebnis dieser Analyse entsteht eine gesammelte Feature-Liste aller betrachteten Anwendungen.

Basierend auf der abstrakten Analyse der Geschäftsprozesse sowie Innovationen gegenüber bestehenden Systemen wird in Kapitel 3 ein Konzept für ein neuartiges System entwickelt. Dieses soll den gesamten Ablauf für alle Arten von definierten Gastronomiebetrieben abbilden und dem Konsumenten einen Smartphone-Client zur Verfügung stellen.

Die Grundidee des Konzepts dient als Hypothese für eine soziologische Studie, die als webbasierte Umfrage durchgeführt wird. Es soll ermittelt werden, ob Interesse an einem derartigen System und den vorgeschlagenen Features besteht. Zusätzlich soll das aktuelle Verhalten hinsichtlich der Auswahl und Nutzung von Angeboten der Gastronomie erhoben werden, um mit diesen Erkenntnissen das Konzept an das bestehende Verhalten anzupassen. In

Kapitel 4 wird die Methodik der Studie präsentiert, und es wird beschrieben, wie das Ergebnis das Konzept beeinflusst.

Auf Basis des durch die Studie verfeinerten Konzepts wird ein funktionaler Prototyp des Smartphone-Clients implementiert. Als Plattform dafür dient das Betriebssystem *iOS*. Zur Anlieferung der Daten und zur besseren Simulation des Benutzererlebnisses wird ebenfalls ein Prototyp des Web Services entwickelt. In Kapitel 5 wird auf den Entwicklungsprozess sowie die konkrete Architektur eingegangen, und es wird erklärt, welche Herausforderungen zu bewältigen sind.

Dieser Prototyp wird einem Usability Test unterzogen. Einerseits soll damit die Software auf Mängel überprüft werden, und andererseits sollen die Benutzer im Anschluss an den Test darüber Auskunft geben, ob sie ein derartiges System auch im realen Einsatz verwenden würden. Da den Probanden der Prototyp für einen Test zur Verfügung gestellt wird, ist deren Meinung aussagekräftiger als eine rein hypothetische Annahme wie bei der soziologischen Studie. Die Methodik sowie die Ergebnisse sind in Kapitel 6 abgebildet.

Im finalen Kapitel 7 werden die Ergebnisse zusammengefasst und Schlüsse gezogen, ob ein derartiges System von Konsumenten akzeptiert werden würde. Abschließend werden weitere Schritte definiert, die nach Entwicklung von Konzept und Prototypen unternommen werden können, um ein marktreifes Produkt zu erschaffen.

Zu Gunsten der leichteren Lesbarkeit wird in der gesamten Arbeit sowohl für die männliche als auch für die weibliche Form personenbezogener Hauptwörter die männliche Form verwendet. Es wird jedoch deutlich darauf hingewiesen, dass sich die Arbeit gleichermaßen an beide Geschlechter richtet.

Kapitel 2

State-of-the-Art

Das folgende Kapitel zeigt den aktuellen Stand der Themengebiete, die für diese Arbeit relevant sind.

Zuerst werden als technische Grundlage Smartphones und deren Leistungsfähigkeit betrachtet. Weiters werden Studien zur aktuellen Verbreitung sowie zum Verhalten der Benutzer präsentiert.

Die Gliederung von Gastronomiebetrieben sowie die Abläufe im Gastronomiewesen mit Kundenbezug werden im darauf folgenden Abschnitt 2.2 behandelt. Es werden die Abläufe des Bestellprozesses beschrieben und basierend auf den Unterschieden die verschiedenen Typen von Betrieben definiert.

Anschließend wird der wissenschaftliche Kontext der Arbeit in Abschnitt 2.3 vorgestellt. Die letzten Abschnitte zeigen technische Entwicklungen, die in der Gastronomie bereits zum Einsatz kommen. Zuerst werden allgemeine Systeme beschrieben, und abschließend werden Online-Gesamtsysteme konkreter evaluiert.

2.1 Smartphones

Oxford Dictionaries Pro definiert „Smartphone“ als [82]

„a mobile phone that is able to perform many of the functions of a computer, typically having a relatively large screen and an operating system capable of running general-purpose applications.“

Mit Stand Ende 2011 ist es darüber hinaus Industriestandard, dass Smartphones mit einem kapazitiven Touchscreen sowie einer Kamera zur Aufnahme von Fotos und Videos ausgeliefert werden. Weiters verfügen die Geräte bei entsprechendem Mobilfunktarif, der im Normalfall gemeinsam mit dem Smartphone verkauft wird, über permanenten Zugang zum Internet.

„Apps“, der allgemein übliche Name für Software für Smartphones, existieren inzwischen für viele Anwendungsbereiche der traditionellen PC-Software. Dies beginnt bei klassischen Gebieten wie Office-Paketen (z. B. *Pages*,

Numbers und *Keynote* für *iOS*) und Spielen (z. B. *Angry Birds*) und reicht bis zu Videoschnitt- und Audiotbearbeitungsprogrammen (z. B. *iMovie* und *GarageBand* für *iOS*), die in den meisten Fällen aber nicht über alle Features einer PC-Version verfügen. Darüber hinaus werden spezielle Apps als Zugänge für Services angeboten, die auf Desktop-PCs hauptsächlich über eine Website genutzt werden. Ein Beispiel dafür ist das soziale Netzwerk *Facebook*. Im Durchschnitt installieren Benutzer 65 Apps pro Gerät [52].

Ein wesentlicher Trend in diesem Bereich ist Cloud Computing [54]. Darunter ist zu verstehen, dass Inhalte wie Kontakte, Kalendereinträge und Daten verschiedener Apps zusätzlich auf Servern über das Internet gespeichert und automatisch mit allen Geräten des Benutzer synchronisiert werden. Auch komplette Backups eines Smartphones sind möglich.

2.1.1 Verbreitung

Weltweit wurden im zweiten Quartal 2008 über 32 Millionen Smartphones verkauft [56]. Im vierten Quartal 2011 stieg dieser Wert um mehr als das Vierfache auf 149 Millionen [57].

Allein im Jahr 2011 wurden weltweit 472 Millionen Smartphones verkauft, was 31 % aller in diesem Jahr verkauften Mobiltelefone entspricht. Im zweiten Quartal 2008 betrug dieser Anteil nur 11 %. Hier muss allerdings der Unterschied zwischen den einzelnen Kontinenten betrachtet werden. Die Penetration in Nordamerika und Europa ist mit 63 % bzw. 51 % deutlich höher als im weltweiten Durchschnitt, da in den restlichen Märkten der Durchschnitt zwischen 17 % und 19 % liegt [112].

Auch innerhalb der einzelnen Nationen gibt es Unterschiede. Die deutsche Telekom verkündete, dass von Januar bis September 2011 61 % ihrer verkauften Mobiltelefone Smartphones waren [107]. In Österreich besitzen 56 % aller Nutzer eines Mobiltelefons ein Smartphone, wie eine im September 2011 veröffentlichte Studie der *Mobile Marketing Association Austria* angibt [72].

Inzwischen sind bei Abschluss eines neuen Mobilfunktarifs Smartphones bereits günstig erhältlich, in Österreich sogar kostenlos [26]. Dadurch ist der Anreiz, statt eines herkömmlichen Mobiltelefons ein Smartphone zu wählen, größer, wodurch sich ableiten lässt, dass der Trend weiterhin in Richtung Smartphones gehen wird.

2.1.2 Betriebssysteme

Mit Stand viertes Quartal 2011 dominieren die zwei Betriebssysteme *Android* (46 % in den USA) und *iOS* (30 %) klar den Markt, gefolgt von *BlackBerry OS* (15 %), *Windows Mobile* (5 %) bzw. dessen Nachfolger *Windows Phone 7* (1 %) und sonstigen (3 %) [74].

	Anzahl Apps	Downloads	Stand
<i>Android</i>	370.000	10 Mrd.	Nov. 11 [93] / Dez. 11 [28]
<i>iOS</i>	500.000	25 Mrd.	März 12 [30]
<i>BlackBerry OS</i>	15.000	1 Mrd.	Nov. 10 [110] / Juli 11 [108]
<i>Symbian</i>	48.000	2 Mrd.	Juni 11 [76] / Aug. 11 [27]
<i>Win. Phone 7</i>	50.000	<i>unbekannt</i>	Dez. 11 [113]

Tabelle 2.1: App Stores der Betriebssysteme

In Österreich zeichnet sich mit 38 % für *Android* und 23 % für *iOS* ein ähnliches Bild [72]. Nur *BlackBerry OS* hat mit 7 % deutlich weniger Bedeutung.

Das Unternehmen hinter *BlackBerry OS*, *Research In Motion*, hat bekannt gegeben, ab 2012 ihr neues Betriebssystem *BBX* einsetzen zu wollen [45]. Auf diesem sollen auch Apps, die für *Android* entwickelt wurden, mittels einer speziellen Laufzeitumgebung ohne Anpassungen seitens der Entwickler eingesetzt werden können.

Nokia setzte auf sein eigenes Betriebssystem *Symbian*, bevor Anfang 2011 entschieden wurde, in Zukunft primär *Windows Phone 7* von *Microsoft* zu verwenden [109].

Ebenfalls auf dem Markt befindet sich *webOS* von *HP* (bzw. ursprünglich *Palm*), das trotz guter Kritiken nur wenig Marktanteil erzielen konnte. Im Dezember 2011 kündigte *HP* an, es unter der Open Source Lizenz veröffentlichten zu wollen [62]. Die Zukunft des Betriebssystems ist unklar, aber es scheint unwahrscheinlich, dass ein signifikanter Marktanteil erreicht werden kann.

Noch klarer wird die Dominanz von *Android* und *iOS*, wenn ein Blick auf die Anzahl der veröffentlichten Apps und deren Downloads in den jeweiligen App Stores, den offiziellen Vertriebskanälen für Apps, geworfen wird. Zu den Zahlen in Tabelle 2.1 muss angemerkt werden, dass diese Betriebssysteme teils auch auf Tablets (siehe Abschnitt 2.1.5) im Einsatz sind und diese Zahlen für sämtliche Geräte kombiniert gelten.

Der Unterschied zwischen *Android* und *iOS* ist ein wesentlicher. *Android* ist ein von *Google* veröffentlichtes Open Source Betriebssystem, das dadurch auf Geräten unterschiedlichster Hersteller wie z. B. *Samsung*, *HTC* oder *Motorola* zum Einsatz kommt. Mit Stand Ende 2011 sind über 60 verschiedene *Android*-Smartphones erhältlich [29]. Diese Geräte sind in ihrer Konfiguration hinsichtlich Größe, Auflösung und Verhältnis des Bildschirms sowie Hardware-Leistung unterschiedlich, weswegen auch die Preise für ein vertragsfreies Gerät zwischen 100 € und 679 € schwanken. Das Betriebssystem selbst wird oft eigens für das jeweilige Gerät angepasst, was speziell im Testprozess eine weitere Hürde für Software-Entwickler darstellt.

iOS wurde von *Apple* entwickelt und kommt nur auf den vom Unternehmen auf den Markt gebrachten Geräten zum Einsatz. Unter dem Namen *iPhone* erschien seit der Einführung im Jahr 2007 jährlich nur ein neues Smartphone, das auch in der fünften Generation noch unverändert hinsichtlich Bildschirmgröße sowie -verhältnis ist. Die Auflösung dessen wurde allerdings in der vierten Generation auf 960 x 640px verdoppelt. Dies erleichtert bei der Entwicklung von Apps sowohl den UI-Design- als auch den Testprozess wesentlich, da das UI nicht für unterschiedlichste Auflösungen entwickelt oder angepasst werden muss. Displays von *Android*-Smartphones sind beispielsweise sowohl mit 480 x 320px (*HTC Wildfire S* [63]) als auch 1280 x 720px (*Samsung Galaxy Nexus* [58]) am Markt.

Native Apps müssen für *Android* in der Programmiersprache *Java* und für *iOS* in *Objective-C* mit jeweils eigenem Framework entwickelt werden. Es existieren Lösungen von Drittherstellern, wie z. B. *Appcelerator Titanium*, um diesen doppelten Entwicklungsprozess abzukürzen. Diese hinken aktuell aber sowohl bezüglich Qualität als auch Popularität den nativ entwickelten Apps hinterher.

2.1.3 Leistung

Tabelle 2.2 zeigt eine Übersicht der Spezifikationen des mit Ende 2011 jeweils leistungsstärksten *Android*- bzw. *iOS*-Smartphones. Diese Zahlen zeigen deutlich, dass es sich bei diesen Smartphones nicht nur um Mobiltelefone, sondern um leistungsstarke Computer handelt.

Forscher des *Oak Ridge National Laboratory* haben Benchmark-Tests mit dem im März 2011 veröffentlichten Tablet *iPad 2* (siehe Abschnitt 2.1.5) durchgeführt. Dieses ist von der Spezifikation ähnlich zu dem später erschienenen Smartphone *iPhone 4S*. Es wurde geschätzt, dass ein Linpack Benchmark von 1,5 bis 1,65 Gigaflops (Milliarden von mathematischen Operationen pro Sekunde) erreicht werden kann und dadurch das Gerät 1994 noch als bester Höchstleistungsrechner der Welt gegolten hätte [77].

2.1.4 Nutzung

Im Jahr 2011 waren mobile Geräte, das inkludiert keine Tablets, für 8,5 % des gesamten Internet Traffics verantwortlich. Im Vergleich waren es 2010 4,3 % und 2009 nur 1,6 % [67].

Über das Verhalten von Smartphone-Besitzern hat *Google* in Partnerschaft mit *Ipsos OTX MediaCT* Ende 2010 eine Studie mit über 5.000 Personen durchgeführt [60].

Nur 11 % geben an, während des Tages ihr Gerät ausschließlich für das Tätigen und Empfangen von Anrufen zu verwenden. 89 % verwenden somit den Internetzugang, Textnachrichten, verschiedene Apps etc.

	<i>Galaxy Nexus</i> [58]	<i>iPhone 4S</i> [32]
Hersteller	<i>Samsung</i>	<i>Apple</i>
Betriebssystem	<i>Android</i>	<i>iOS</i>
Taktfrequenz Prozessor	1,2 GHz (Dualcore)	1,0 GHz (Dualcore)
Hauptspeicher	1 GB RAM	512 MB RAM
Interner Speicher	16 oder 32 GB	16, 32 oder 64 GB
Display: Größe	4,65"	3,5"
Display: Auflösung	1280 x 720px	980 x 640px
Internet Mobilfunknetz	21,6 Mbps (down) 5,8 Mbps (up)	14,4 Mbps (down) 5,8 Mbps (up)
Kamera	5 MP mit Blitz	8 MP mit Blitz
Videoaufnahme	1080p	1080p

Tabelle 2.2: Spezifikationen aktueller Smartphones

Den Internetzugang verwenden 89 % zur Kommunikation, 82 % zur Konsumation von Information, 75 % zur Navigation, 65 % zum Entertainment und 45 % zum Verwalten und Planen ihres Lebens. Darunter fallen Finanzen (34 %) und auch Platzreservierungen in Gastronomiebetrieben (20 %).

Als Einsatzort geben 93 % ihr Zuhause, 87 % unterwegs und – speziell für diese Arbeit interessant – 73 % Restaurants und 54 % Kaffeehäuser an. 48 % benutzen ihr Smartphone auch während des Essens.

Als meist besuchte Art von Websites werden mit 77 % Suchmaschinen genannt. Auf Platz 2 der damit gesuchten Inhalte befinden sich mit 51 % Restaurantinformationen, nur geschlagen von Nachrichten (57 %).

22 % tätigten bereits Einkäufe über eine App direkt auf ihrem Smartphone. In den veröffentlichten Top-8-Kategorien der erworbenen Güter befinden sich dabei keine in Zusammenhang mit Gastronomie oder Essen.

Einen Ermässigungs-Coupon lösten schon 28 % mittels ihres Geräts ein.

Als letzte Zahl sei genannt, dass bereits 95 % ihr Smartphone zur Beschaffung von Informationen über Plätze rund um ihren Standort verwendet haben. Unter diese Plätze fallen neben Sehenswürdigkeiten und Geschäften natürlich auch Gastronomiebetriebe.

2.1.5 Tablets

Mit dem im April 2010 veröffentlichten *iPad* trat *Apple* eine Welle an Veröffentlichungen von Tablets los.

Oxford Dictionaries Pro definiert „Tablet PC“ als [83]

„a small portable computer that accepts input directly on to its screen rather via than [sic] a keyboard or mouse.“

Tablets sind größer als Smartphones, aber zeichnen sich ebenfalls dadurch aus, dass ein kapazitiver Touchscreen einen großen Teil der Vorderseite des Geräts einnimmt. Aktuelle Displaygrößen liegen zwischen 7" und 10".

Im Gegensatz zu Smartphones ist die Möglichkeit der Telefonie über das Mobilfunknetz zumeist nicht gegeben, aber Zugriff auf das Datennetzwerk ist bei entsprechendem Modell möglich.

Eine Studie von *Gartner* gibt an, dass bis September 2011 weltweit 63,6 Millionen Tablets verkauft wurden. Weiters wird geschätzt, dass von allen im Jahr 2011 verkauften Tablets 73,4% auf das *iPad* mit seinem Betriebssystem *iOS* entfallen werden, womit dieses, gefolgt von diversen *Android*-Tablets mit 17,3%, deutlicher Marktführer bleiben wird. [55] Somit ist zu sehen, dass *iOS* und *Android* auch diesen Sektor dominieren.

2.2 Gastronomie

In diesem Abschnitt wird zu Beginn analysiert, wie der Geschäftsprozess eines Gastronomiebesuchs ohne den Einsatz von aktuellen technischen Hilfsmitteln typischerweise aussieht. Zur besseren Veranschaulichung wird die Seite des Konsumenten und die des Anbieters gesondert betrachtet. Ausgehend von dieser Beschreibung des traditionellen Ablaufs wird dessen Abstraktion entwickelt.

Anschließend werden, basierend auf den Parametern in den Transaktionen, die einzelnen Typen der Gastronomiebetriebe definiert.

2.2.1 Analyse des Geschäftsprozesses Gastronomiebesuch

Ablauf Konsument

Zu Beginn muss der Konsument sich für einen Betrieb entscheiden, bei dem er Artikel erwerben will. Im herkömmlichen Sinn wird der Konsument durch Mundpropaganda, Marketing-Maßnahmen wie Postwurfsendungen/Handzettel oder durch Zufall auf einen Gastronomiebetrieb aufmerksam.

Platzreservierungen werden direkt im Lokal oder via Telefon vorgenommen und sind üblicherweise möglich, wenn Sitzplätze zum Verzehr der Speisen im Betrieb vorhanden sind.

Wenn der Konsument die Speisekarte vor dem Besuch im Lokal zu sehen bekommt, dann entweder deswegen, weil sie beim Eingang ausgehängt ist, oder er Printwerbung erhalten hat, in der diese abgedruckt wurde. Bei Fastfood-Betrieben ist das Angebot in den meisten Fällen über alle Niederlassungen in einem Staat ident, weswegen der Kunde auch so mit dem Angebot vertraut sein könnte. Im Lokal werden Speisekarten durch das Personal in der Form eines Printmediums ausgehändigt und können durch angebrachte Aushänge ergänzt oder ersetzt werden.

Sobald der Konsument auf Basis der Speisekarte die gewünschten Artikel ausgewählt hat, wird die Bestellung an einen Mitarbeiter übermittelt. Bei Aufenthalt im Betrieb wird direkt vom Sitzplatz aus oder an einer Theke bestellt.

Alternativ können Bestellungen auch via Telefon übermittelt werden, um diese entweder zu einem definierten Zeitpunkt selbst abzuholen oder zur Zustellung an eine Adresse.

Nach erfolgreicher Bestellung nimmt der Kunde die bestellten Artikel in Empfang. Beim Aufenthalt im Betrieb werden die Produkte entweder an den Sitzplatz gebracht oder, falls die Bestellung an einer Theke erfolgt, muss im Normalfall gewartet werden, bis die Artikel übergeben werden. Sollten diese nicht binnen kürzester Zeit verfügbar sein, werden sie entweder verspätet an den Platz zugestellt oder müssen zu einem späteren Zeitpunkt von der Theke abgeholt werden.

Verzehrt der Konsument die Speisen im Gastronomiebetrieb, besteht die Möglichkeit, weitere Artikel zu bestellen, wofür bei Bedarf wieder die Speisekarte angefordert werden muss.

Bonussysteme, wie z. B., dass jede 10. Bestellung eines bestimmten Artikels oder von Waren einer bestimmten Gruppe kostenlos ist, werden über Stempelkarten abgewickelt.

Die Bezahlung der bestellten Produkte kann direkt nach der Bestellung, bei Entgegennahme der Artikel oder nach deren Konsumation erfolgen und wird mittels Bargeld, Kreditkarte oder Bankomarkarte abgewickelt.

Durch eine abstrakte Betrachtung sind folgende Schritte ermittelt worden, die je nach Art von Gastronomiestätte und Form der Bestellung in ihrer Reihenfolge variieren und auch kombiniert in Aktion treten können:

- Gastronomiebetrieb auswählen
- Speisekarte ansehen
- Platzreservierung vornehmen
- Bestellung aufgeben
- Artikel in Empfang nehmen
- Bezahlung tätigen

Ablauf Anbieter

Ohne den Einsatz von technischen Systemen muss der gesamte Ablauf der Bestellung seitens des Betreibers direkt über das Personal abgewickelt werden.

Der Betreiber der Gastronomiestätte muss sein Angebot veröffentlichen. Das wird in Form der Speisekarte erfolgen, aber nach Möglichkeit auch durch verschiedene Marketing-Aktivitäten.

Speisekarten müssen den Kunden bei Bedarf an den Platz gebracht werden, sofern diese sich dort nicht permanent befinden. Wenn sich das Angebot ändert oder falls aktuelle Aktionen, wie z. B. Mittagsmenüs, veröffentlicht werden sollen, müssen alle Exemplare manuell adaptiert werden.

Platzreservierungen und Bestellungen müssen mündlich aufgenommen, schriftlich auf Papier vermerkt und anschließend an die richtige Stelle im Unternehmen, wie die Küche oder die Bar, übermittelt werden. Technische Lösungen zur Unterstützung dieser Abläufe sind zum Teil schon im Einsatz (siehe Abschnitt 2.4.1).

Auch die Bezahlung muss durch einen Mitarbeiter persönlich entgegengenommen werden.

Wird auch diese Seite des Ablaufs ganz allgemein betrachtet, können die nachfolgenden Punkte erkannt werden:

- Angebot veröffentlichen
- Reservierung entgegennehmen
- Bestellung entgegennehmen
- Speisen zubereiten
- Artikel ausliefern
- Bezahlung entgegennehmen

Diese können ebenfalls je nach Art von Gastronomiebetrieb in ihrer Reihenfolge variieren oder kombiniert auftreten.

2.2.2 Identifizierung der Typen von Gastronomiebetrieben

In Tabelle 2.3 werden die verschiedenen Arten von Gastronomiebetrieben definiert, und der typische Ablauf einer Bestellung wird aufgeschlüsselt.

Der Teilprozess „Gastronomiebetrieb auswählen“ wird dabei außen vor gelassen, da sich dieser zwischen den Typen nicht unterscheiden muss. Für die Beschreibung der Teilprozesse wird die allgemein üblichste Form verwendet, wobei diese in der Praxis je nach Betrieb und Ansprüchen des Konsumenten individualisiert wird.

Nicht alle existierenden Betriebe lassen sich genau einer Kategorie zuordnen, sondern stellen oft Variationen eines Typs dar oder umspannen mehrere. Ein Beispiel sind Gaststätten, die gleichzeitig einen Lieferservice betreiben.

2.3 Wissenschaftlicher Kontext

Sämtliche wissenschaftlichen Publikationen, die im Zusammenhang mit technischen Lösungen für die Gastronomie stehen, behandeln nur Teilbereiche des Geschäftsprozesses, so wie die im folgenden Abschnitt 2.5 vorgestellten kommerziellen Online-Systeme, und basieren teilweise auf veralteter Technologie.

	Gaststätte	Fastfood-Betrieb	Lieferservice	Club
Reservierung möglich?	Ja	Nein	Nein	Nein
Speisekarte (Ort)	an Sitzplatz	an zentralem Ort	Flyer	an zentralem Ort
Speisekarte (Art)	Print	Aushang, Print	Print	Aushang, Print
Bestellung (Ort)	am Sitzplatz	an zentralem Ort	Fernkommunikation	an zentralem Ort
Form der Warteschlange	-	Reihe(n)	-	Traubenform
übliche Bestellschritte	2 (Getränke, Speisen)	1	1	1+
Auslieferung (Ort)	an Sitzplatz	an zentralen Ort	Hauszustellung	an zentralen Ort
Bezahlung (Zeitpunkt)	nach Konsumierung	nach Bestellung	bei Anlieferung	nach Bestellung
Bonussysteme üblich?	zum Teil	zum Teil	zum Teil	Nein
Trinkgeld üblich?	Ja	Nein	Ja	zum Teil

Tabelle 2.3: Arten der Gastronomiebetriebe

Beispielsweise dient die in [6] vorgestellte App nur zum Auffinden von Gastronomiebetrieben, die Algorithmen in [17] nur zur besseren Empfehlung von Restaurants und die Konzepte in [21] und [4] nur zur Bestellung in Restaurants mittels Computer Monitoren bzw. einem multi-touch-fähigen Tisch. Das in [25] vorgestellte System behandelt nur die Lieferung von Speisen und behandelt nicht mehr zeitgemäße Technologien, wie z. B. *PDA*s (*Personal Digital Assistant*) und *WAP* (*Wireless Application Protocol*)

Für diese Arbeit eignet sich eine genauere Untersuchung kommerzieller Lösungen besser, da diese durch den Markteinsatz besser an reale Bedürfnisse von Gastronomiebetrieben bzw. Konsumenten adaptiert sind.

2.4 Bestehende Systeme

Wenn man altbewährte technische Entwicklungen wie Telefon, Kreditkarte und Bankomatkarte dem in Abschnitt 2.2 vorgestellten traditionellen Ablauf zurechnet, lassen sich aktuelle Entwicklungen in folgende Kategorien einteilen:

- POS-Systeme
- Terminals
- Bezahl-Systeme
- Chipkarten
- Online-Systeme

Diese werden nachfolgend betrachtet. Online-Systeme werden auf Grund ihres Umfangs im eigenen Abschnitt 2.5 behandelt.

2.4.1 POS-Systeme

Inzwischen haben sich in Gastronomiebetrieben POS-Systeme etabliert, die der Betreiber einsetzt und mit denen ausschließlich die Mitarbeiter interagieren.

Im Allgemeinen dienen diese zur Abwicklung des Prozesses am „*Point of Sale*“ (POS). Im klassischen Handel ist ein POS-System ein Kassensystem. Für den Einsatz in der Gastronomie bieten diese Systeme oftmals erweiterte Funktionalitäten. Sie dienen dazu, den Ablauf, der in Abschnitt 2.2.1 beschrieben wurde, zu unterstützen.

Wie im Handel werden auch in der Gastronomie Waren über das System erfasst, deren Bezahlung anschließend über dieses abgewickelt wird.

Bedient wird dieses System in der Gastronomie üblicherweise durch Terminals mit Touchscreens, die im Lokal verteilt sind. Zusätzlich sind in Gaststätten inzwischen immer öfters mobile Handhelds im Einsatz, mit denen Bestellungen direkt am Tisch durch einen Angestellten aufgenommen werden können. Ohne diese Geräte muss die Bestellung vermerkt und anschließend

an einem Terminal eingegeben werden, wodurch ein zusätzlicher Zeitaufwand entsteht.

Eine typische zusätzliche Funktionalität von POS-Systemen für die Gastronomie ist die Fähigkeit, das Tisch-Layout des Restaurants abzubilden und so Bestellungen optisch richtig zuzuordnen. Auch Reservierungen können so direkt erfasst werden.

Eine andere angebotene Erweiterungsmöglichkeit ist die Implementierung von logistischen Abläufen. So werden die Waren der Bestellung nicht nur erfasst, sondern auch direkt an die richtige Stelle im Lokal zur Zubereitung übergeben. So sieht z. B. der Angestellte an der Bar auf einem Screen nur jene Getränke, für deren Zubereitung er verantwortlich ist, und in der Küche scheinen nur die bestellten Speisen auf. In der Praxis kann, je nach Anforderung, ein komplexeres Szenario implementiert werden.

Für die Bezahlung wird abschließend die Rechnung ausgedruckt. Die Möglichkeiten der Zahlung variieren nach Implementierung. Standard sind Bargeld, Bankomat- und Kreditkarte. Auf weitere Möglichkeiten wird in Abschnitt 2.4.3 eingegangen.

Ergänzt werden die genannten Funktionalitäten durch umfangreiches Reporting.

Das POS-System stellt somit den zentralen Knotenpunkt für die Verwaltung des Ablaufs im Betrieb dar. Etablierte Lösungen für Gastronomie-POS-Systeme stammen von den Unternehmen *MICROS*, *Aloha* und *Positouch*.

2.4.2 Terminals

Unter Terminals werden stationäre Geräte verstanden, die im Betrieb angebracht sind und mit denen der Konsument interagieren kann.

Diese Entwicklung ist nicht mehr ganz aktuell. Beispielsweise eröffnete schon vor über einem Jahrzehnt das Lokal *Point* in Amstetten, Österreich, welches auf jedem Tisch Touchscreens angebracht hatte. Bestellungen konnten ausschließlich über diese vorgenommen werden, und die Artikel wurden anschließend direkt an den Tisch gebracht. Bezahlt werden musste weiterhin direkt bei einem Angestellten über herkömmliche Wege.

Trotzdem hat sich diese Entwicklung noch nicht durchgesetzt, und Feldversuche mit erweiterter Funktionalität finden immer noch statt. Beispielsweise sind in der Fastfood-Filiale *McDonald's* in der Mariahilfer Straße 85-87, 1060 Wien, seit Anfang 2010 sechs stationäre Terminals, genannt „*easy order*“, zur Bestellung angebracht [42]. Über die Terminals wird nicht nur die Bestellung zusammengestellt und übermittelt, sondern auch direkt via Kredit- oder Bankomatkarte bezahlt. Eine Bestellung ohne eine dieser Karten ist nicht möglich. Die Ausgabe erfolgt weiterhin an der Theke. Die Zuordnung der Bestellung erfolgt über einen Bon mit Bestellnummer, der vom Terminal am Ende des Vorgangs ausgedruckt wird. Auf einem Bildschirm über der Theke scheinen die Nummern der zur Abholung bereiten Bestel-

lungen auf. Dieser ist allerdings nur von einem kleinen Teil der verfügbaren Sitzplätze sichtbar, weswegen im Bereich der Theke gewartet werden muss. Die Artikel des angeschlossenen Services *McCafé* sind nicht via Terminal verfügbar, sondern müssen bei der eigenen Theke bestellt werden. Mit Stand 15. Dezember 2011 sind diese Terminals nur in 9 der 174 Filialen Österreichs (5 %) im Einsatz [68].

Zwei Beispiele für aktuelle Lösungen für Terminals, die direkt an den Tischen angebracht werden, sind *eMenu* vom israelischen Unternehmen *Conceptic* [43] und *E la Carte* [47]. Gastronomen können diese Systeme erwerben und in ihrem Betrieb einsetzen.

2.4.3 Bezahl-Systeme

Der Einsatz von alternativen Bezahlssystemen ist nicht rein auf die Gastronomie beschränkt, weswegen hier nur kurz auf zwei aktuelle Beispiele eingegangen wird.

NFC

Near field communication (NFC) ist eine Technologie, die für neue Bezahl-Systeme vielfach zur Anwendung kommt. Mittels *NFC* können zwei Geräte, in diesem Szenario ein Bezahl-Terminal des Betreibers und ein Mobiltelefon des Konsumenten, auf knapper Distanz miteinander kommunizieren. So können Bezahl-Informationen ausgetauscht und Transaktionen durchgeführt werden.

Ein Beispiel für eine aktuelle Implementierung ist *Google Wallet* [59]. In Österreich sind das auf bargeldlosen Zahlungsverkehr spezialisierte Unternehmen *Hobex* und die *Raiffeisen Bankengruppe Österreich* eine Partnerschaft eingegangen und haben öffentliche Feldversuche mit kontaktlosem Payment am 1. August 2011 gestartet [94]. Ausgegeben werden Kreditkarten mit NFC-Chip, die mit Terminals, die in verschiedensten Unternehmen in ganz Österreich installiert wurden und weiterhin werden, kommunizieren.

Starbucks Cards

Als abschließendes Beispiel dient der alternative Ansatz der Kaffeehaus-Kette *Starbucks*, die nach der Gliederung in Abschnitt 2.2.2 den Fastfood-Betrieben zugeordnet werden kann.

Diese bietet in ihren Niederlassungen in den USA *Starbucks Cards* an. Diese können erworben werden und dienen als Gutscheine, die in der Form von Scheckkarten ausgegeben werden. Im September 2009 hat *Starbucks* begonnen, diese Karten in einem mobilen System abzubilden. Besitzer können die Nummer ihrer *Starbucks Card* in einer App eingeben und das Guthaben so mit ihrem Benutzerkonto verknüpfen. Ihr Guthaben kann auch via Kreditkarte aufgeladen werden. Zusätzlich ist ein Bonusprogramm integriert,

mit dem *Stars* gesammelt und gegen Belohnungen, wie kostenlose Getränke, eingetauscht werden können.

Beim Erwerb von Artikeln in einer Filiale kann ein 2D-Code am Display des mobilen Geräts angezeigt werden, welcher von einem Scanner an der Kassa erfasst wird. Daraufhin wird der Betrag automatisch vom Guthaben des Benutzers abgebucht.

Mit Stand Dezember 2011 ist das System in knapp 9.000 Filialen in den USA im Einsatz und als App für *iOS* und *Android* verfügbar [99]. Im Jahr 2011 wurden 26 Millionen Transaktionen in Gesamthöhe von \$ 110 Millionen über dieses mobile System vorgenommen, was aber nur 5% der \$ 2,4 Milliarden Umsatz via Geschenkkarten entspricht [41].

2.4.4 Chipkarten

Ein alternatives System zur Abwicklung der Bestellung sind Chipkarten-Systeme, die von Fastfood-Ketten eingesetzt werden. Die Grundidee ist, dass jedem Besucher bei dem Eintritt in das Lokal eine Chipkarte ausgehändigt wird. Bei Bestellung an einer Theke wird der zahlbare Betrag auf die Chipkarte aufgebucht, indem der Kunde diese an ein dort installiertes Terminal hält. Beim Verlassen des Lokals wird die Chipkarte wieder abgegeben und der Gesamtbetrag, der während des Besuchs aufgebucht wurde, muss bezahlt werden.

Im Einsatz sind derartige Systeme beispielsweise in Niederlassungen der Fastfood-Ketten *Vapiano* und *Eurest* sowie in Kantinen diverser Unternehmen.

2.5 Online-Systeme

2.5.1 Kategorisierung

Online-Systeme können in verschiedensten Ausprägungen auftreten. Grundsätzlich ist zwischen speziell entwickelten Lösungen für einzelne Betriebe bzw. Ketten, nachfolgend genannt Einzelsysteme, und Systemen, die Informationen verschiedenster Betriebe kombinieren, nachfolgend genannt Gesamtsysteme, zu unterscheiden.

Neben Lösungen, deren Umsetzung ein reines UI für Desktop-Browser bietet, sind momentan Umsetzungen für Smartphones und Tablets in der Form von Apps oder auch Websites mit speziell angepasstem Layout am Vormarsch. Diese Entwicklung hat neben der rasant ansteigenden Anzahl an Smartphones ihre Begründung darin, dass die Entscheidung für die Auswahl eines Lokals oft unterwegs und nicht an einem Ort, an dem ein PC verfügbar ist, getroffen wird. Weiters bieten diese mobilen Geräte mehr Möglichkeiten, da sie auch in den Betrieb mitgebracht und dort zur Abwicklung des Prozesses verwendet werden können.

Einzelsysteme

Begonnen hat alles mit der klassischen Website eines Betreibers, die Basisinformationen, wie die Speisekarte und Adressen der Niederlassungen, öffentlich zugänglich macht. Es existieren auch Lösungen, die weitere Funktionalitäten wie Reservierungen anbieten, aber das ist nicht die Norm.

Eine Recherche hat ergeben, dass mit Stand Dezember 2011 ein Großteil der Gastronomiebetriebe über eine Website verfügt, aber kaum welche von mobilen Systemen Gebrauch machen. Vereinzelt bieten Ketten mit einer hohen Anzahl von Niederlassungen, wie z. B. *Pizza Hut* in den USA, umfassende Lösungen für mobile Plattformen mit Reservierung, Vorbestellung und Bezahlung an, aber das sind Einzelfälle.

Unternehmen wie *Fresh Intermedia* [53] oder *MenuDrive* [69] bieten Systeme für mobile Speisekarten mit Bestellfunktion an, die für Betreiber von Gastronomiebetrieben adaptiert und eingesetzt werden können.

Der Nachteil all dieser Lösungen aus Sicht des Konsumenten ist, dass für jeden Betrieb bzw. jede Kette eine eigene Website aufgerufen oder eine eigene App installiert werden muss. Auch bei der Wahl des Lokals, außer innerhalb einer Kette, wird der Konsument nicht unterstützt. Für die Betreiber ist dies ebenfalls kein optimaler Zustand, da so die Existenz ihres Lokals mangels Präsenz in einer zentralen Plattform unter Umständen schwerer wahrgenommen wird.

Gesamtsysteme

Gesamtsysteme bilden mehr als nur einen Betrieb bzw. eine Kette ab. Der Inhalt kann hier entweder von Benutzern oder vom Betreiber des Systems bzw. dessen Kunden gepflegt werden.

Durch die Zusammenlegung der Inhalte mehrerer Betriebe eröffnen sich neue Möglichkeiten im Vergleich zu Einzelsystemen, wie z. B. die Abgabe von Bewertungen, die anschließend dem Konsumenten bei der Wahl des Lokals helfen.

2.5.2 Evaluierung Online-Gesamtsysteme

Um einen Überblick darüber zu schaffen, welche Online-Gesamtsysteme am Markt sind und welche Features abgedeckt werden, wurde eine Ist-Analyse durchgeführt. Die nachfolgend präsentierten Systeme sind durch Recherche ermittelt worden. Sofern ein System entweder signifikante Aktivität der Nutzer, einen umfangreichen aktuellen Datenbestand oder Alleinstellungsmerkmale, wie innovative Features, aufweisen konnte, wurde es ausgewählt.

Dadurch entsteht eine gesammelte Feature-Liste der einzelnen Systeme, die als Ausgangspunkt für das eigene Konzept dient.

Qype

Dieser Dienst wurde 2006 in Hamburg, Deutschland gegründet und wirbt mit dem Slogan „Entdecken. Empfehlen.“ [95]. Angemeldete Benutzer können wichtige Punkte in einer Stadt eintragen, bewerten und finden. *Qype* ist nicht auf die Gastronomie beschränkt, sondern enthält auch andere Orte wie z. B. Hotels und Einkaufsmöglichkeiten.

Prinzipiell agiert dieser Dienst inzwischen weltweit, der große Teil des Inhalts ist aber auf Städte in Europa bezogen.

Neben der Website mit Desktop- und Mobile-Interface werden mit Stand Dezember 2011 auch Apps für *iOS*, *Android*, *Windows Phone 7*, *BlackBerry OS* und *Symbian* angeboten. Ein spezielles Interface für Tablets steht nicht zur Verfügung.

Als zentrale Features, die auch im Zusammenhang mit Gastronomiebetrieben stehen, wurden folgende identifiziert:

- Verwaltung der Orte durch die Benutzer (Stammdaten inkl. Öffnungszeiten, Kategorisierung)
- Eigentümer von Betrieben können den Eintrag für sich beanspruchen und verwalten
- Positionsbestimmung (manuell bzw. automatisch auf Mobiltelefonen) und Anzeige der Orte in der Umgebung (inkl. Maps-Ansicht)
- Manuelle Suche nach Orten (Filterung nach Preisklasse und anderen Kriterien)
- Automatische Ortsempfehlungen in Bezug auf die aktuelle Position und andere angesehene Orte
- Bewertung von Orten (Skala, Textreview)
- Bild-Upload für Orte
- Favoriten-Funktion
- Community-Features (Profil, Kontakte, Messages etc.)

Yelp

Yelp basiert auf einem ähnlichen Konzept wie *Qype* und wurde 2004 ins Leben gerufen [118]. Die Funktionalität und die Feature-Liste sind großteils ident. Obwohl der Dienst seinen Sitz und Ursprung in den USA hat, scheint auch Europa durch die Benutzer inzwischen gut erfasst zu sein. Mit Stand 15. Dezember 2011 sind in *Yelp* 5.409 Einträge in der Kategorie „Speis’ & Trank“ für Wien zu finden, *Qype* bietet im Vergleich 6.068 Plätze unter „Essen und Trinken“.

Erwähnenswert ist, dass *Yelp* auch eine *webOS* App und eine *iOS* App mit einem Interface für das *iPad* anbietet. Für *Symbian* hingegen steht keine App zur Verfügung.

Alfred

Im Juli 2011 veröffentlichte das Unternehmen *Clever Sense* die *iOS* App *Alfred*. Diese App dient zum Auffinden von Gastronomiebetrieben und bedient sich hierfür selbst entwickelter Such-Algorithmen, die mit künstlicher Intelligenz arbeiten [39]. *Alfred* empfiehlt dem Benutzer Betriebe und lernt, ob der Benutzer an der Empfehlung Gefallen gefunden hat und lässt dies in zukünftige Vorschläge einfließen.

Im Moment ist *Alfred* auf Gastronomiebetriebe in den USA beschränkt. *Clever Sense* plant sowohl die Region als auch die Art der Betriebe auszuweiten. Die Datenbank im Hintergrund füllt das Unternehmen selbst. Dazu bedienen sie sich eines Web-Crawlers, der das Internet nach Bewertungen durchforstet, diese analysiert und 200 bis 400 Attribute pro Betrieb vergibt. So werden auch einzelne Speisen separat bewertet.

Im Dezember 2011 wurde die App für *Android* veröffentlicht, und kurz darauf wurde bekannt gegeben, dass *Clever Sense* von *Google*, dem Unternehmen hinter *Android*, gekauft wurde [105]. Ob der Service weiterhin so betrieben oder ob die Funktionalität in Produkte von *Google*, wie *Google Places*, einem *Qype*-ähnlichen Service, oder gar *Android* selbst, integriert wird, ist zum jetzigen Zeitpunkt nicht bekannt.

OpenTable

OpenTable ist ein Service, der es ermöglicht, Platzreservierungen in den am System teilnehmenden Restaurants vorzunehmen. Der Dienst wurde 1998 gestartet und gibt mit Stand November 2011 an, monatlich für eine Million Reservierungen in den 20.000 teilnehmenden Betrieben verantwortlich zu sein [106]. Von diesen sind 12.000 in den USA, der Rest in 18 anderen Staaten. In Österreich sind das aktuell nur 17 Betriebe, somit weniger als 0,01 % aller Teilnehmer [80].

Eine Reservierung über den Dienst ist für den Konsumenten kostenlos, Betreiber zahlen pro getätigter Reservierung. *OpenTable* bietet den Gastromomen ein System, mit dem generell Platzreservierungen verwaltet werden können. Dieses integriert sich mit den Online-Reservierungen, aber auch auf anderem Weg abgegebene Reservierungen können verwaltet werden. Eine Integration mit verschiedenen POS-Systemen ist ebenfalls möglich, um Reportings, wie z. B. über den durchschnittlichen Umsatz pro Reservierung, zu generieren [79].

Für den Kunden bietet *OpenTable* ein umfangreiches Spektrum an Zugängen: Website mit Desktop- und Mobile-Interface, Apps für *iOS* (*iPhone* und *iPad*), *Android* (Smartphone UI), *BlackBerry OS*, *webOS* und *Windows Phone 7* [81].

Menupages

Menupages bietet Gastronomen die Möglichkeit, kostenlos die Speisekarte ihres Betriebs im System zu veröffentlichen. Normale Benutzer können keinen Eintrag vornehmen oder verwalten. Dieser Dienst ist mit Stand Dezember 2011 sowohl in acht Städten in den USA als auch in London und Paris verfügbar [70].

Menupages ermöglicht es, wie andere Systeme, mittels verschiedener Kriterien nach Restaurants zu suchen und diese zu bewerten, nur wird die Auswahl des Lokals durch die verfügbare Speisekarte unterstützt. Reservierungen und Bestellungen sind nicht möglich.

Neben der Website mit Desktop-Interface sind Apps für *iOS (iPhone)* und *Android* verfügbar.

Snapfinger

Auch *Snapfinger* bietet eine Datenbank aller teilnehmenden Restaurants mit Speisekarten. Jedoch dient dieser Service vornehmlich zum Bestellen von Speisen, die auch individuell angepasst werden können. Die bestellten Artikel können je nach Restaurant abgeholt und/oder zugestellt werden.

Bestellt werden kann mit Stand Dezember 2011 bei über 28.000 Restaurants in den USA über die Website (Desktop- und Mobile-Interface) sowie eine *iOS (iPhone)* und *Android* App [98].

Weitere Bestelldienste

Es existieren weitere Systeme, die auf demselben Konzept wie *Snapfinger* basieren. Beispiele dafür sind:

- *GrubHub* (über 15.000 Restaurants in 19 amerikanischen Städten) [61]
- *Seamless* (über 7.500 Restaurants in 37 amerikanischen Städten) [97]
- *Lieferheld* (über 5.000 Restaurants in Deutschland) [65]
- *NetKellner* (über 2.000 Restaurants in Österreich, Deutschland, Polen, Ungarn, Tschechien) [73]
- *mjam* (über 500 Restaurants in Österreich) [71]
- *Lieferservice.at* (über 400 Restaurants in Österreich) [66]

Das Geschäftsmodell und die Art der Zugänge mögen bei all diesen Diensten variieren, die Basisfunktionalität ist aber dieselbe.

Durch diese konkurrierenden Anbieter ergibt sich eine Fragmentierung des Marktes. Manche Betriebe sind in Service X, aber nicht in Service Y vertreten und umgekehrt. Abhilfe will ein Service namens *ordr.in* schaffen, das mehrere dieser Services kombiniert und an Drittanbieter eine API zur Integration in eigene Apps zur Verfügung stellt. Mit Stand Oktober 2011

kooperiert *ordr.in* mit 72 Services in den USA und bietet dadurch die Möglichkeit, von über 7.000 Restaurants zu bestellen [104].

Tabbedout

Tabbedout ist ein System zum Bezahlen von Rechnungen in Gastronomiebetrieben. Es existieren Apps für *iOS* und *Android*.

Ein Benutzer kann einem Mitarbeiter eines teilnehmenden Betriebs einen durch die App generierten Code zeigen, der diesen im POS-System eingeben kann. Dadurch kann der Benutzer die aktuelle Rechnung jederzeit am Smartphone einsehen. Die Abgabe von Bestellungen erfolgt weiterhin auf traditionellem Weg, aber über die App kann die Rechnung jederzeit bezahlt werden.

Mit Stand Dezember 2011 ist *Tabbedout* in 328 Betrieben in den USA verfügbar [103].

BiteHunter

Dieser Dienst aggregiert spezielle Angebote von über 1.000 verschiedenen Quellen und zeigt dem Benutzer damit, in welchen Restaurants in seiner Umgebung er günstig speisen kann. *BiteHunter* ist in den USA verfügbar und bietet neben der Website mit Stand Dezember 2011 eine *iOS* App [37].

2.5.3 Gesamtüberblick

Wenn die Lösungen mit den Teilschritten des Geschäftsprozesses aus Sicht des Konsumenten aus Sektion 2.2.1, erweitert um den Punkt „Feedback“, verglichen werden, ergibt sich die in Tabelle 2.4 dargestellte Übersicht. Es ist zu erkennen, dass kein System den Geschäftsprozess vollständig abbildet.

Tabelle 2.5 bietet eine Übersicht über die Zugänge, die die evaluierten Systeme mit Stand Dezember 2011 anbieten.

Feature-Liste

Werden die Funktionalitäten aller Systeme kombiniert, ergibt sich folgende Feature-Liste:

- Datenbank von Gastronomiebetrieben
 - Verwaltung entweder durch
 - * System-Betreiber
 - * Community
 - * Community mit speziellem Recht für Lokal-Besitzer
 - Inhalt
 - * Stammdaten: Name, Adresse, Küche, Öffnungszeiten etc.

- * Bilder
- * Speisekarte
- Suche/Empfehlungen
 - * Positionsbestimmung und Anzeige der Orte in der Umgebung
 - * Manuelle Suche durch Kriterien
 - * Automatische Empfehlung via A.I.-basiertem Algorithmus
 - * Automatische Empfehlung via aktuell angesehenen Orten, Position etc.
- Abgabe von öffentlich einsehbarem Feedback
 - * nach erfolgreicher Bestellung bzw. Reservierung
 - * jederzeit
- Favoriten-Funktion
- Reservierung von Plätzen
- Bestellung von Speisen mit Möglichkeit der Individualisierung für
 - Lieferung
 - Abholung
- Bezahlung
- Community-Features

	Lokal finden	Speisekarte ansehen	Reservierung	Bestellung für			Bezahlung	Feedback
				Lieferung	Abholung	Dine-In		
<i>Qype</i>	X	-	-	-	-	-	X	
<i>Yelp</i>	X	-	-	-	-	-	X	
<i>Alfred</i>	X	-	-	-	-	-	X	
<i>OpenTable</i>	X	-	X	-	-	-	X	
<i>MenuPages</i>	X	X	-	-	-	-	X	
<i>Snappfinger</i>	X	X	-	X	X	X	-	
<i>GrubHub</i>	X	X	-	X	X	-	X	
<i>Seamless</i>	X	X	-	X	X	-	X	
<i>Lieferheld</i>	X	X	-	X	-	-	X	
<i>NetKellner</i>	X	X	-	X	-	-	-	
<i>myjam</i>	X	X	-	X	-	-	X	
<i>Lieferservice.at</i>	X	X	-	X	-	-	X	
<i>Tabbedout</i>	X	-	-	-	-	X	-	
<i>BiteHunter</i>	X	-	-	-	-	-	-	

Tabelle 2.4: Abdeckung der Schritte des Geschäftsprozesses durch die evaluierten Online-Gesamtsysteme

	Website - UI		iOS		Android		BlackBerry		Symbian		webOS		Windows	
	Desktop	Mobile	iPhone	iPad	Smartph.	Tablet	OS	OS	Symbian	webOS	Phone 7	Phone 7		
<i>Qype</i>	X	X	X	-	X	-	X	X	X	-	X	-	X	
<i>Yelp</i>	X	X	X	X	X	-	X	X	X	X	X	X	X	
<i>Alfred</i>	-	-	X	X	X	-	-	-	-	-	-	-	-	
<i>OpenTable</i>	X	X	X	X	X	-	X	X	-	X	X	X	X	
<i>MenuPages</i>	X	-	X	-	X	-	-	-	-	-	-	-	-	
<i>Snapfinger</i>	X	X	X	-	X	-	-	-	-	-	-	-	-	
<i>GrubHub</i>	X	-	X	-	X	-	-	-	-	-	-	-	-	
<i>NetKellner</i>	X	X	-	-	-	-	-	-	-	-	-	-	-	
<i>mjam</i>	X	-	X	-	-	-	-	-	-	-	-	-	-	
<i>Tabbedout</i>	-	-	X	-	X	-	-	-	-	-	-	-	-	
<i>BiteHunter</i>	X	-	X	-	-	-	-	-	-	-	-	-	-	

Tabelle 2.5: Zugänge zu evaluierten Online-Gesamtsystemen

Kapitel 3

Konzept Eigensystem

Zu Beginn wird die Grundidee des Konzeptes präsentiert. Im darauf folgenden Abschnitt wird die Architektur entwickelt, und die geplanten Zugänge zum System werden vorgestellt.

Die darauf folgenden beiden Abschnitte widmen sich zwei wesentlichen Problemstellungen:

- Art der Bezahlung
- Vorgangsweise zum Check-In

Im letzten Abschnitt werden die potentiellen Formen des Missbrauchs betrachtet und welche Maßnahmen dagegen gesetzt werden können.

Dieses Konzept ist bereits durch die Ergebnisse der soziologischen Studie (Kapitel 4) sowie der Evaluierung des Prototypen (Kapitel 6) verfeinert worden. Am Ende der jeweiligen Kapitel wird beschrieben, welche Punkte der Ergebnisse konkret Einfluss auf dieses Konzept hatten.

3.1 Grundgedanke

3.1.1 Art des Systems

Für den Konsumenten ist es am einfachsten, ein Online-Gesamtsystem einzusetzen, da sich dieser dadurch nicht für jeden Betrieb mit einem neuen System vertraut machen muss, weswegen ein solches entwickelt werden soll. Weiters ist dadurch aus wirtschaftlicher Sicht der Markt potentieller Kunden größer.

Die Datenbank der Gastronomiebetriebe soll durch den System-Betreiber verwaltet werden. Aufgenommen werden sollen nur teilnehmende Betriebe. Dadurch verliert das System zwar an Attraktivität hinsichtlich der Funktionalität zur Entdeckung von Betrieben und Angeboten, aber es wird ein konsistentes Erlebnis gewährleistet, indem alle gelisteten Unternehmen die Funktionalitäten des entwickelten Systems unterstützen.

Diese Betriebe sollen ihren Eintrag selbst verwalten können. Dieser soll neben den relevanten Stammdaten, wie Adresse und Öffnungszeiten, auch Bilder des Lokals sowie die Speisekarte inkl. Abbildungen und Preisen zur Bestellung enthalten.

3.1.2 Abwicklung einer Bestellung

Im Gegensatz zu den bestehenden Lösungen soll der gesamte Bestellprozess abgebildet werden. Weiters soll das System auch einsetzbar sein, wenn sich der Konsument bereits im Lokal befindet, um so Auslieferung der Speisekarte sowie Bestellung und Bezahlung der Artikel über dieses abwickeln zu können. Prinzipiell soll die komplette Feature-Liste von Abschnitt 2.5.3 in einem System zusammengefasst werden.

Zur Entscheidung der Auswahl des Betriebs soll der Benutzer durch die in der Feature-Liste genannten Punkte, wie z. B. Anzeige der Betriebe im Umfeld des eigenen Standorts, unterstützt werden. Auf Grund der Komplexität wird von einem A.I.-basiertem Empfehlungssystem in dieser Arbeit abgesehen.

Im Fall einer Gaststätte soll der Benutzer Sitzplätze reservieren können. Nach Möglichkeit soll das Layout des Lokals sichtbar sein, um so direkt den gewünschten Tisch auswählen zu können.

Hält sich der Benutzer im Betrieb auf und will über das System eine Bestellung tätigen, muss er einen sogenannten Check-In durchführen. Das bedeutet, dass er bekannt gibt, dass er im Betrieb ist und welchen Sitzplatz er gewählt hat. Letztgenannte Option ist nur dann verfügbar, wenn Speisen auf den Tisch zugestellt werden. Details zur Vorgehensweise zum Check-In werden in Abschnitt 3.4 behandelt.

Zur Bestellung ist am Client die Speisekarte sichtbar. Diese soll aktuelle Aktionen, wie z. B. Mittagsmenüs, enthalten. Die Speisen sollen hinsichtlich der Zutaten individualisiert werden können.

Ist der Benutzer nicht im Lokal, kann er je nach Angebot des Betriebs die Bestellung zur Lieferung, Abholung oder zum Dine-In abgeben. Dine-In ist bei keinem der betrachteten Gesamtsysteme eine Möglichkeit.

Zur Orientierung in der oft umfangreichen Speisekarte sollen die Top 10 Speisen angezeigt werden. Dies kann sich entweder auf die Anzahl der Bestellungen oder die durchschnittliche Bewertung beziehen. Weiters soll der Benutzer persönliche Favoriten anlegen können.

Die Bezahlung der Bestellung soll je nach Betrieb individualisiert möglich sein. Es soll die Option bestehen, direkt bei Abgabe der Bestellung im System zu zahlen oder erst bei persönlicher Übernahme auf anderem Weg. Die Möglichkeit der Vergabe von Trinkgeld soll bestehen. In Abschnitt 3.3 wird auf den Bezahlvorgang genauer eingegangen.

Es sollen Bonusprogramme integriert werden. Dabei handelt es sich eine elektronische Umsetzung der bekannten „Pizzapässe“, die eine kostenlose

Konsumation jeder zehnten Pizza in einem bestimmten Restaurant ermöglichen. Dieses Konzept lässt sich auf andere Warengruppen mit anderen Intervallen umlegen und kann auch mehrere Niederlassungen eines Betriebs umfassen. Der Benutzer soll jederzeit seinen aktuellen Status aufrufen können, und Vergünstigungen sollen automatisch bei der Bezahlung angewandt werden. Dadurch entsteht für den Benutzer kein zusätzlicher Aufwand, und die Komplexität wird für ihn gering gehalten, da er sich nicht damit beschäftigen muss, welche Bonussysteme zu welchem Zeitpunkt anwendbar sind.

Nach der Bezahlung soll der Benutzer für ein gewisses Zeitfenster die Möglichkeit haben, den Betrieb sowie die bestellten Speisen in Form einer Skala und einem Text-Review zu bewerten. Weiters soll je ein Foto zu jeder konsumierten Speise hochgeladen werden können. Die Bewertungsskala könnte auch nur zweistufig sein, womit quasi die Frage „Würden Sie den Betrieb nochmals besuchen?“ bzw. „Würden Sie den Artikel nochmals bestellen?“ beantwortet werden würde. Anstatt die Gesamtbewertung eines Betriebs bzw. einer Speise als einfachen Durchschnitt der Bewertungen zu errechnen, könnte diese darauf basieren, wie oft Benutzer tatsächlich nochmals in einem Betrieb bestellen bzw. denselben Artikel nochmals konsumieren.

Bei der Abholung einer Bestellung muss eine Identifikation des Benutzers erfolgen. Das könnte wie bisher auf dem herkömmlichen Weg erfolgen, so dass die Person, die zur richtigen Zeit in den Betrieb kommt und Kenntnis über die Bestellung hat, als Inhaber der Bestellung anerkannt wird. Sollte mehr Sicherheit gefordert sein, könnte ein Code am Smartphone-Client angezeigt werden, der zur Identifikation dient.

3.1.3 Benutzer-Funktionalität

Eine Bestellung soll nur mit einem aktiven Benutzer-Konto möglich sein. Dies dient zur Vorbeugung von Missbrauch des Systems. Zusätzlich können so die Daten des Benutzers zur besseren Individualisierung verwendet werden, um, z. B. basierend auf bisherigen Bestellungen, bestimmte Speisen oder Betriebe zu empfehlen.

Auf Community-Features, wie das Versenden privater Nachrichten, soll verzichtet werden. Dafür sollen Benutzer in ihrem Profil Angaben zu Kontaktmöglichkeiten in sozialen Netzwerken hinterlegen können.

Die durchgeführte soziologische Studie (Kapitel 4) hat ergeben, dass die Empfehlung von Bekannten ein wesentlicher Faktor für die Entscheidung zur Nutzung eines Gastronomieangebots ist. Das soll im System durch das bekannte Konzept von Freunden abgebildet werden, wobei hier zwei verschiedene Ansätze existieren.

Im ersten Fall muss eine Freundschaft im System von beiden Benutzern bestätigt werden und spiegelt in der Regel eine tatsächliche Bekanntschaft wieder. Beim „Follower“-Ansatz des sozialen Netzwerks *Twitter* ist diese Beziehung einseitig und zeigt nur, dass ein Benutzer Interesse an der Aktivität

des anderen hat. Je nach Einstellung können Benutzer andere Benutzer ohne deren Bestätigung als Freunde markieren. Bei diesem Ansatz muss keine tatsächliche Bekanntschaft existieren, sondern zeigt nur ein zumindest einseitiges Interesse am anderen Benutzer.

Beide Konzepte lassen sich theoretisch mischen, wie dies z. B. von *Facebook* erfolgt, aber darauf soll wegen der steigenden Komplexität für Benutzer verzichtet werden.

Abhängig von der Wahl des Konzepts können verschiedene Features umgesetzt werden. Bei dem klassischen Schema können die bestehenden sozialen Beziehungen beispielsweise dazu genutzt werden, um gemeinsame Termine für einen Besuch in einem Betrieb zu vereinbaren. Beim „Follower“-Ansatz kann darauf abgezielt werden, interessante Personen ausfindig zu machen, um deren Aktivität in Form von verfassten Reviews zu verfolgen, um neue Betriebe zu entdecken.

Für dieses System soll der „Follower“-Ansatz zum Einsatz kommen, um dadurch den Fokus darauf zu legen, Benutzern bei der Entdeckung interessanter Betriebe zu helfen und dabei nicht auf den Kreis der eigenen Bekannten eingeschränkt zu sein.

3.1.4 Vorteile des Systems

Als wesentlicher Vorteil dieses Systems für Konsumenten wird gesehen, dass im Vergleich zu momentanen Inanspruchnahmen des Services von Gastronomiebetrieben Zeit gespart werden kann. Zusätzlich ist die Dauer des Besuchs besser abschätzbar, da die Abhängigkeit von der zeitlichen Verfügbarkeit des Personals wegfällt.

Aus Sicht der Betreiber kann bei hoher Adaptierung des Systems durch die Konsumenten Personal eingespart werden. Servicepersonal müsste nur noch für die Auslieferung der Speisen zuständig sein, und selbst dieser Schritt könnte durch die Konsumenten abgewickelt werden, wenn das System sie informiert, sobald ihre Speisen an einem gewissen Ort im Betrieb zur Abholung bereit sind.

Weiters kann das System für Marketing-Maßnahmen verwendet werden, um Konsumenten mittels spezieller Angebote, wie z. B. Rabattgutscheinen, das Service des Betriebs schmackhaft zu machen. Feedback, das durch die Benutzer über das System gegeben wird, kann zur Verbesserung des Services verwendet werden.

3.2 Architektur

Durch den starken mobilen Charakter des Systems soll der Fokus auf Implementierungen für mobile Plattformen gelegt werden. Websites spielen aber weiterhin eine wesentliche Rolle, weswegen auch eine solche entwickelt werden sollte. Ob dabei aus technischer Sicht dieselben Features der

Smartphone-Clients realisiert werden können, muss ihm Rahmen einer separaten Evaluierung festgestellt werden. Auf die Website wird in Abschnitt 3.2.3 eingegangen.

Aus Sicht der Benutzer wäre es am besten für jedes aktiv benutzte Smartphone-Betriebssystem eine Implementierung zu schaffen. In Abschnitt 2.1.2 wurde gezeigt, dass im Moment *Android* und *iOS* am häufigsten im Einsatz sind, und auch bei bestehenden Systemen sind Clients für diese beiden Betriebssysteme am verbreitetsten (siehe Tabelle 2.5). Somit sollte die Priorität auf die Entwicklung eines Clients für diese zwei Plattformen gelegt werden. In Abschnitt 3.2.2 wird dieser Client genauer konzipiert.

Wenn für eine Plattform kein nativer Client zur Verfügung steht, sollte die Nutzung des Dienstes über die Website möglich sein. Dafür muss sich die Website entweder durch ein spezielles Mobile User Interface oder über *Responsive Web Design (RWD)* an die Benutzung über ein Smartphone anpassen. Unter *RWD* ist zu verstehen, dass sich das Layout einer Seite automatisch an die Größe des Bildschirms anpasst und sichergestellt wird, dass der Inhalt in jeder Bildschirmauflösung gut repräsentiert wird.

Wie das Ergebnis der soziologischen Studie (Kapitel 4) deutlich gezeigt hat, haben Konsumenten ihre Tablets bei einem Besuch eines Gastronomiebetriebs kaum bei sich. Deswegen ist die Entwicklung eines Tablet-UIs als optional anzusehen, da wenig Nutzungspotential für das wesentliche Feature der direkten Bestellung im Betrieb besteht.

Denkbar wäre die Anbringung von stationären Tablets auf Tischen der Betriebe. Diese müssten allerdings robust verarbeitet sein, um den speziellen Anforderungen gerecht zu werden. Getränke, die am Tisch verschüttet werden, oder die Bedienung mit schmutzigen Fingern dürfen das Tablet in seiner Funktionalität nicht einschränken. Beispielsweise ist für das Terminal-System *E la Carte* eigens ein Tablet namens *Presto* für den stationären Einsatz auf Restauranttischen entwickelt worden [48].

Zur Verwaltung der Daten muss ein Administrations-Client entwickelt werden. Dieser muss einerseits Betrieben ermöglichen, ihre Stammdaten, wie z. B. die Öffnungszeiten, zu bearbeiten und andererseits den System-Betreibern die Möglichkeit geben, teilnehmende Betriebe und Benutzer zu verwalten. Dafür soll eine Website mit Desktop-UI zum Einsatz kommen. Auf dessen Konzeption wird in Abschnitt 3.2.4 eingegangen.

In dieser Arbeit wird davon ausgegangen, dass in den Gastronomiebetrieben POS-Systeme im Einsatz sind, die über eine Schnittstelle verfügen, mittels derer das zu entwickelnde System mit dem POS-System kommunizieren kann.

Es muss eine Server-Umgebung entwickelt werden, die sowohl mit den verschiedenen Clients als auch den POS-Systemen der Gastronomiebetriebe kommuniziert. Diese dient zur Verwaltung der systemeigenen Daten und auch dazu, dass Clients nicht direkt mit POS-Systemen kommunizieren müssen.

Somit erfolgt die Verwaltung der Bestellungen weiterhin über das POS-System. Eine Stornierung einzelner Artikel soll am Client nicht möglich sein, um Problemen vorzubeugen, falls der Artikel schon in der Zubereitungsphase ist, worüber das POS-System in der Regel nicht informiert ist. Deswegen müsste eine derartige Stornierung mündlich beim Personal erfolgen, das je nach Möglichkeit den Artikel regulär im POS-System stornieren kann. Über die Schnittstelle wird das zu entwickelnde System über die Stornierung informiert. Diese Information wird an den Client weitergegeben, auf dem dadurch die Anzeige der bestellten Artikel entsprechend aktualisiert wird.

Diese Server-Umgebung ist Thema des nun folgenden Abschnitts.

3.2.1 Web Service

Die Server-Umgebung kommuniziert über ein Web Service mit den Clients, wozu auch die Website gezählt wird. Im Hintergrund werden Daten mit den POS-Systemen der teilnehmenden Betriebe ausgetauscht. Dies geschieht rein über die Server-Umgebung, Clients kommunizieren nie direkt mit diesen. Das ermöglicht, Anpassungen vornehmen zu können, ohne dass Konsumenten ihre Client-Apps einem Update unterziehen müssen.

Es wird die Annahme getroffen, dass folgende Schnittstellen zu den POS-Systemen verfügbar sind:

- Abruf des Tischlayouts des Betriebs
- Abfrage der Verfügbarkeit für Reservierungen
- Anlegung bzw. Stornierung von Reservierungen
- Abfrage der Artikel (Speisekarte)
- Eröffnung einer Bestellung
- Hinzufügen von Artikeln zu einer bestehenden Bestellung
- Bezahlung einer Bestellung
- Abfrage, ob Rechnung bezahlt worden ist
- Stornierung einer Bestellung

Zusätzlich zu diesen genannten Schnittstellen zum POS-System müssen den Clients folgende Schnittstellen zur Verfügung stehen:

- Benutzer-Registrierung
- Benutzer-Login
- Abfrage von Betrieben, sortiert nach der Entfernung zum Standort des Benutzers
- Abfrage von Betrieben, die kürzlich vom Benutzer oder seinen Freunden besucht wurden
- Suche nach Betrieben, Speisen und Benutzern
- Abfrage der Stammdaten eines Betriebs

- Check-In in einen Betrieb
- Abfrage vergangener Bestellungen eines Benutzers
- Abfrage des Standes von Bonusprogrammen
- Abgabe eines Reviews zu einer Bestellung
- Abfrage von Reviews zu einem Betrieb oder von einem Benutzer
- Markierung von Speisen als Favoriten und Abfrage dieser
- Hinzufügen und Entfernen von Freunden

3.2.2 Smartphone-Client

Wie schon in Abschnitt 3.2 argumentiert wurde, sollte ein nativer Client zumindest für *Android* und *iOS* entwickelt werden.

Da im Rahmen dieser Arbeit ein Prototyp des Smartphone-Clients entwickelt wird (Kapitel 5), wird der Aufbau des Clients an dieser Stelle ausführlicher beschrieben. Die Struktur des Clients soll für alle verschiedenen Implementierungen ident sein. Nur das Design soll dem allgemeinen Stil des jeweiligen Betriebssystems angepasst werden.

Startscreen

Beim Start der App soll der Standort des Benutzers automatisch ermittelt werden. Sollte dies nicht möglich sein, soll entweder die zuletzt ermittelte Position herangezogen oder der Benutzer zu einer manuellen Eingabe aufgefordert werden.

Basierend auf dem ermittelten Standort sollen umliegende Betriebe, sortiert nach der berechneten Entfernung, angezeigt werden. In einer Liste sollen tagesaktuelle Angebote dieser Betriebe angezeigt werden, und es soll die Möglichkeit geben, zur Suche zu gelangen.

Der Benutzer soll bei jedem Start automatisch mit seinem Account eingeloggt werden. Dann sollen ihm in weiteren Ansichten seine favorisierten Betriebe sowie Betriebe, zu denen Freunde kürzlich Reviews verfasst haben, angezeigt werden. Wichtig ist, dass über einen Benachrichtigungsmechanismus daran erinnert wird, wenn abgeschlossene Besuche noch nicht bewertet wurden. Es soll kein Zwang zur Bewertung bestehen, aber es soll eine bewusste Entscheidung sein, wenn keine vorgenommen wird.

Ist der Benutzer nicht eingeloggt, soll die Möglichkeit zur Registrierung und zum Login bestehen.

Benutzer-Funktionalität

Zur Registrierung sollen so wenige Daten wie möglich angegeben werden müssen. Dies dient dazu, interessierte Personen nicht abzuschrecken und den

Prozess so kurz und einfach wie möglich zu gestalten. Speziell auf Smartphones werden lange Eingaben oft ungern getätigt.

Vorgesehen werden sollen Felder für E-Mail-Adresse, Passwort, Vor- und Nachname sowie optional PLZ und Wohnort. Letzteres dient v.a. zur Auffindung von interessanten Personen im eigenen lokalen Umfeld.

Zum Login ist die Angabe der E-Mail-Adresse und des Passworts erforderlich. Die Zugangsdaten sollen in der App gespeichert und der Benutzer soll bei jedem Start automatisch eingeloggt werden.

Auf dem Profil eines Benutzers sollen die Anzahl der von ihm verfassten Reviews sowie die Benutzer, denen er folgt bzw. die ihm folgen, angezeigt werden.

Es soll möglichst einfach sein, bestehende Freunde von sozialen Netzwerken zu finden, weswegen es die Möglichkeit geben soll, den eigenen Account mit *Facebook* und *Twitter* zu verknüpfen. So können Benutzer, die ein Freund in einem dieser sozialen Netzwerke sind, gefunden und vorgeschlagen werden, wenn sie ihren Account ebenfalls verknüpft haben.

Detailansicht Gastronomiebetrieb

Der wichtigste Bereich der Applikation ist die Detailansicht eines Gastronomiebetriebs. Hier soll zwischen den Ansichten

- Stammdaten mit Reviews
- Sitzplatzreservierung
- Speisekarte
- Warenkorb

gewählt werden können.

Die erste Ansicht, Stammdaten, soll alle wichtigen Informationen auf einen Blick zeigen. Das umfasst den Namen sowie ein Bild des Betriebs, die Adresse, die Art der Bestellmöglichkeiten (Check-In, Dine-In, Lieferung, Abholung), die Typen der angebotenen Speisen und die Öffnungszeiten. Weiters soll der Benutzer in dieser Ansicht seinen Stand bei den angebotenen Bonusprogrammen sofort sehen können. Eine Liste von Reviews soll die bisher abgegebenen Bewertungen, absteigend sortiert nach Datum, anzeigen.

Wenn in einem Betrieb Sitzplatzreservierungen möglich sind, soll eine Ansicht mit dem Tischlayout angezeigt werden. Spezielle Bereiche im Betrieb, wie z. B. ein Raucherbereich oder ein Barbereich, in dem ausschließlich Getränke serviert werden, sollen entsprechend gekennzeichnet werden. Nach der Eingabe des gewünschten Zeitpunkts sollen alle verfügbaren Tische für den Benutzer auswählbar sein. Nach der Selektion der gewünschten Tische kann durch Eingabe des Passworts die Reservierung abgeschickt werden.

Für die Speisekarte sollen die Artikel in Gruppen gegliedert werden, um die Navigation zu erleichtern. Angeführt werden soll die Gruppenliste von den dynamischen Kategorien „Favoriten“, „Angebote“ und „Top 10“. In der

Liste der Speisen einer einzelnen Gruppe soll zusätzlich zu dem Namen und dem Preis jeweils ein kleines Bild sichtbar sein. Von dieser Liste soll die Möglichkeit bestehen, einen Artikel entweder direkt dem Warenkorb hinzuzufügen oder weitere Informationen aufzurufen. Sollten die Zutaten einer Speise individualisierbar sein, soll dazu vor der Aufnahme in den Warenkorb ein eigener Dialog erscheinen. In der Detailansicht einer Speise sollen größere Bilder, die Zutaten sowie die Bewertungen sichtbar sein.

Im Warenkorb soll die Anzahl der ausgewählten Artikel veränderbar sein. Bevor es zur eigentlichen Bestellung kommt, muss angegeben werden können, ob diese zum Dine-In, zur Abholung oder zur Lieferung bestimmt ist. Je nach ausgewählter Art müssen anschließend Eingaben wie Zeitpunkt, Lieferadresse und Art der Bezahlung erfolgen. Bei der Bestellung zum Dine-In soll zusätzlich eine Reservierung vorgenommen werden können.

Bei einer Bestellung sollen automatisch die verfügbaren Bonusprogramme zur Anwendung kommen. Angenommen im Rahmen eines Bonusprogramms wird jede 10. Speise einer Art kostenlos angeboten, ist im Warenkorb zu sehen, wie viele Speisen im Rahmen dieses Programms bisher bestellt worden sind, die Anzahl der Posten, die durch diese Bestellung hinzukommen und ob bereits ein Preisabzug erfolgt.

Hält sich der Benutzer in einem teilnehmenden Betrieb auf und will von dem System Gebrauch machen, muss über die App ein Check-In durchgeführt werden. Dieser soll ebenfalls von dieser Ansicht aus möglich sein. Sollte die Bestellung nicht direkt zum Sitzplatz zugestellt werden, muss der Konsument über den Client darüber informiert werden, wenn es an einem zentralen Ort zur Abholung bereit ist. Im Falle der Zustellung an den Sitzplatz muss dieser im Rahmen des Check-Ins ermittelt werden. Die möglichen Mechanismen zum Check-In werden in Abschnitt 3.4 behandelt.

Hat der Kunde seine Bestellung zum Dine-In aufgegeben, muss er bei seinem Erscheinen im Betrieb einen Check-In durchführen. Da der Betrieb nun über seinen Aufenthalt informiert ist, können ihm die bestellten Artikel übergeben werden bzw. kann er darüber informiert werden, dass die Artikel zur Abholung bereit stehen.

War der Check-In erfolgreich, soll die Ansicht zur Sitzplatzreservierung nicht mehr angezeigt werden. Dafür muss eine Gesamtübersicht der aktuellen Bestellung sichtbar sein. Alle vom Warenkorb aus bestellten Artikel sollen in dieser Ansicht gesammelt angezeigt werden. Benachrichtigungen über zur Abholung bereit stehende Artikel sollen direkt bei den jeweiligen Posten aufscheinen.

Ist der Besuch beendet, kann der Benutzer die Bestellung bezahlen und dadurch einen Check-Out durchführen. Die jeweiligen Bonusprogramme kommen auch hier zur Anwendung.

Nach einer erfolgreich durchgeführten Bestellung soll der Benutzer am Startscreen eine Benachrichtigung erhalten, dass er nun eine Bewertung abgeben kann.

3.2.3 Website

Die Website soll im Wesentlichen dieselben Funktionalitäten wie der soeben beschriebene Smartphone-Client anbieten. Es soll eine Lokalisierung vorgenommen werden, nahe gelegene Betriebe sollen angezeigt werden, und es sollen Sitzplatzreservierungen und Bestellungen möglich sein.

Neben einem Desktop-UI soll, wie bereits in Abschnitt 3.2 argumentiert wurde, für Smartphones, für deren Betriebssystem kein nativer Client existiert, ein Mobile-UI entwickelt werden. Alternativ kann diese Problematik auch via *Responsive Web Design* gelöst werden.

Ob über die Website ein Check-In ermöglicht werden soll, hängt von dem gewählten Mechanismus zum Check-In ab und davon, ob alle Features technisch realisiert werden können.

3.2.4 Administrations-Client

Der Administrations-Client soll zwei Arten von Zugängen ermöglichen. Einerseits für Gastronomiebetriebe und andererseits für die Betreiber des Systems selbst. Für diesen Client soll eine Website mit Desktop-UI zum Einsatz kommen.

Betrieb

Ein großer Teil der administrativen Arbeit wird vom POS-System übernommen. Daten, wie z. B. die Speisekarte, werden von diesem abgefragt (Abschnitt 3.2).

Über diesen Client sollen die Stammdaten verwaltet werden. Betriebe sollen Daten, wie Adresse, Öffnungszeiten und Bilder, jederzeit verändern können. Hier soll ebenfalls bestimmt werden, welche Bestellmöglichkeiten (Check-In, Dine-In, Lieferung, Abholung) möglich sind.

Sollte das POS-System die Speicherung von Bildern zu Artikeln der Speisekarte nicht ermöglichen, soll es hier möglich sein, Bilder hinzuzufügen. Auch denkbar sind erweiterte Informationen wie Nährwertangaben.

Bonusprogramme sollen über diesen Client verwaltet werden. Dazu können bestimmte Artikel oder Artikelgruppen selektiert werden, um anzugeben, welcher Rabatt bei welcher bestellten Anzahl zum Tragen kommen soll.

Betreiber sollen mittels des Administrations-Clients die Möglichkeit haben, Reviews an den System-Betreiber zu melden, die als ungerecht oder unseriös empfunden werden. Die Entscheidung über die tatsächliche Entfernung soll beim System-Betreiber liegen.

Ebenfalls implementiert werden soll ein umfangreiches Reporting, z. B. über die Anzahl der bestellten Artikel, Frequenz der Nutzung zu bestimmten Zeiten und Rückkehr-Raten von Benutzern.

System-Betreiber

Der System-Betreiber muss die Betriebe verwalten und diese neu anlegen bzw. deaktivieren können. Natürlich können auch dieselben Daten, wie sie die Betriebe selbst sehen, bearbeitet werden.

Da der System-Betreiber von den teilnehmenden Betrieben eine monetäre Gegenleistung erwarten wird, z. B. in der Form eines fixen Prozentsatzes pro Bestellung, soll die Abrechnung und Rechnungslegung zwischen diesen Parteien über den administrativen Client erfolgen können.

Neben den Betrieben sollen ebenfalls die Benutzer verwaltet werden können, um diese auch bei Bedarf sperren oder verwarnen zu können.

Wie im vorigen Abschnitt beschrieben wurde, soll der System-Betreiber gemeldete Reviews sehen können, um darüber zu entscheiden, ob diese tatsächlich aus dem System entfernt werden sollen.

Auch für den System-Betreiber sollen diverse Nutzungsstatistiken ersichtlich sein.

3.3 Bezahlung

Bei Dine-In, Lieferung und Abholung wird jeweils die komplette Bestellung verrechnet, bei Check-In werden die bestellten Artikel am Ende des Aufenthalts gesammelt bezahlt.

Die Bezahlung soll prinzipiell über das System erfolgen. Es ist aber auch zu überlegen, weiterhin eine Zahlung über den herkömmlichen Weg beim Personal mit Bargeld, Kreditkarte etc. zuzulassen. Dieses Szenario wird in Abschnitt 3.3.1 behandelt. Erfolgt die Zahlung über das System, ist dies im POS-System ersichtlich und kann so durch den Betreiber überprüft werden. Bei der Bezahlung beim Personal rechnet dieses selbst mittels des POS-Systems ab.

Missbrauchsfälle, wie z. B. das Verlassen des Lokals durch den Kunden ohne zu bezahlen, werden eigens in Abschnitt 3.5 behandelt.

Für die Art der Bezahlung bieten sich verschiedene Alternativen an. Nachfolgend werden diese vorgestellt und auftretende Vor- und Nachteile besprochen.

3.3.1 Traditioneller Weg

Wie die durchgeführte soziologische Studie (Kapitel 4) ergeben hat, ist Bargeld im Moment das am häufigsten verwendete Zahlungsmittel im Zusammenhang mit Gastronomiebetrieben.

Prinzipiell wäre es möglich, die Option anzubieten, weiterhin beim Personal zu bezahlen. Das würde bedeuten, dass bei Abschluss der Bestellung, z. B. bei Übergabe der Lieferung oder vor Verlassen des Betriebs, weiterhin eine Transaktion mittels Bargeld, Kreditkarte etc. stattfinden würde, und dieser

Teil des Geschäftsprozesses unverändert bliebe. Bei Aufenthalt des Konsumenten im Lokal würde das für ihn einen erhöhten Zeit- und für den Betrieb einen erhöhten Personalaufwand bedeuten. Da der Kunde aber selbst diese Option gewählt hätte, kann der für ihn erhöhte Zeitaufwand vernachlässigt werden.

Als wesentlicher Vorteil dieser Art der Bezahlung ist das Vertrauen in die Form der Transaktion anzusehen. Der Konsument hat selbst Kontrolle über den Zahlungsverkehr und muss kein Vertrauen in die korrekte Vorgangsweise einer elektronischen Transaktion oder die Form der Verarbeitung seiner Zahlungsdaten durch ein neues System haben.

Durch die Ergebnisse der Studie kann davon ausgegangen werden, dass bei Wahl dieser Option großteils mit Bargeld bezahlt werden würde. Dadurch entsteht für den Betreiber ein weiterer Vorteil, da im Vergleich zu anderen Formen der Bezahlung keine Transaktionsgebühren anfallen. Die Kosten von Cash-Handling bewegen sich je nach Industrie zwischen 0,3 % und 3,0 % [96], wobei die Gastronomie im unteren Bereich anzusiedeln ist. Je nach Anbieter liegen Transaktionsgebühren für Zahlungen mittels Kreditkarte in Österreich mit Stand Mai 2009 zwischen 1,21 % und 2,48 % des Transaktionswerts zzgl. 0,10 € bis 0,15 € pro Transaktion [114]. Zahlungen mittels Maestro liegen bei 0,30 % bis 0,39 %, ebenfalls zzgl. 0,10 € bis 0,15 € pro Transaktion.

Ein Nachteil ist die erhöhte Gefahr von Missbrauchsfällen, die in Abschnitt 3.5 gesondert behandelt werden.

3.3.2 Kreditkarte

Die Bezahlung mittels einer Kreditkarte ist v.a. online gebräuchlich und sollte somit ermöglicht werden.

Grundsätzlich gilt es zu entscheiden, ob die dafür notwendigen Daten selbst gespeichert und verarbeitet werden sollen, oder ob auf das Service eines Drittanbieters zurückgegriffen wird. Will man Kreditkartenzahlungen selbst akzeptieren, wird ein spezielles Bankkonto, ein sogenanntes Händlerkonto, benötigt.

Bei Drittanbietern muss zwischen zwei Integrationsmöglichkeiten unterschieden werden. Die eine Form von Anbietern arbeitet für den Benutzer unsichtbar, indem die Daten weiterhin in der eigenen Software eingegeben und im Hintergrund an das Service weitergeleitet werden (z. B. *WorldPay*, *Wirecard*). Die alternative Form leitet den Benutzer auf das Angebot des Drittanbieters weiter, die Eingabe erfolgt dort, und er muss eventuell sogar über einen eigenen Account beim Anbieter verfügen (z. B. *PayPal*, *Clickand-Buy*).

PCI-Standards

Sobald Kreditkartendaten in der eigenen Applikation eingegeben oder verarbeitet werden, unabhängig davon, ob ein eigenes Händlerkonto existiert oder dies im Hintergrund über einen Drittanbieter abgewickelt wird, muss die Applikation einem internationalen Sicherheitsstandard der „*Payment Card Industry*“ (*PCI*) gerecht werden. Ansonsten wird das Händlerabkommen mit dem jeweiligen Kreditinstitut verletzt [88].

Durchgeführt wird die Zertifizierung von dem *PCI Security Standards Council (PCI SSC)*, einer 2006 gegründeten Institution der Kreditkarten-Unternehmen *American Express*, *Discover Financial Services*, *JCB International*, *MasterCard Worldwide* und *Visa Inc.* [92].

Abhängig davon, welche Aktionen durchgeführt werden, muss einer der folgenden Standards erfüllt werden: [90]

- *PCI PIN Transaction Security (PCI PTS)*
„The PCI PTS applies to manufacturers who specify and implement device characteristics and management for personal identification number (PIN) entry terminals used for payment card financial transactions.“
- *PCI Payment Application Data Security Standard (PCI PA-DSS)*
„The PA-DSS is for software developers and integrators of applications that store, process or transmit cardholder data as part of authorization or settlement. It governs these applications that are sold, distributed or licensed to third parties.“
- *PCI Data Security Standard (PCI DSS)*
„The PCI DSS applies to any entity that stores, processes, and/or transmits cardholder data. It covers technical and operational system components included in or connected to cardholder data. If your business accepts or processes payment cards, it must comply with the PCI DSS.“

Bei *PCI DSS* existieren darüber hinaus verschiedene Stufen der Konformität („Compliance Levels“). Diese werden von jedem Kreditkarten-Unternehmen anders definiert und sind von verschiedenen Faktoren, wie z. B. der Anzahl der jährlich durchgeführten Transaktionen, abhängig [49].

Diese Standards betreffen nicht nur die jeweilige Software, sondern auch die Hardware, auf denen sie ausgeführt wird, wie z. B. die Server-Umgebung oder das mobile Gerät.

Im Juni 2011 hat das *PCI SSC* bekannt gegeben, vorerst keine Apps für Smartphones mit *PCI DA-DSS* zu zertifizieren, so lange nicht ausdrücklich festgestellt worden ist, dass das jeweilige Smartphone intern die Daten sicher verarbeitet [91]. Mit Stand Januar 2012 erfolgte zu diesem Thema keine weitere Stellungnahme.

Die Kosten der Zertifizierung sind davon abhängig, welcher Standard erfüllt werden muss, sowie im Falle von *PCI DSS* von dem eingestuften Level. Berichten zufolge beginnen diese Kosten bei mehreren 10.000 \$ bis mehrere 100.000 \$ (*PCI PA-DSS*) [87] und reichen bis zu mehreren Millionen Dollar (*PCI DSS*) [89]. Diese Kosten setzen sich aus den notwendigen Investitionen zur Erfüllung des Standards sowie den regelmäßigen Überprüfungen zusammen. Eine Studie besagt, dass die Kosten einer Sicherheitslücke und eines darauf folgenden Datenraubs die Kosten der Zertifizierung aber um das bis zu 20-fache übersteigen können [46].

Wahl der Methode

Somit bleibt zu evaluieren, mit welcher Lösung langfristig der höchste wirtschaftliche Gewinn erzielt werden kann.

Zu Beginn wird die Lösung über die Eingabe der Kreditkartendaten bei einem Drittanbieter die beste Variante sein. Somit muss nur dieser einem der vorgestellten Standards entsprechen. Neben der unmittelbaren Kostenersparnis ist als weiterer Vorteil auch die Einrichtung in der Regel binnen weniger Tage abgeschlossen. Wenn der Drittanbieter über einen bekannten Markennamen verfügt, wie das z. B. *PayPal* tut, steigt dadurch zusätzlich das Vertrauen der Kunden in den Vorgang der Bezahlung.

Bei großem Erfolg des Systems muss berechnet werden, ob die Kosten einer *PCI*-Zertifizierung bei einer eigenen Lösung durch die niedrigere Gebühr pro einzelner Transaktion, gerechtfertigt werden. Für eine Implementierung im Smartphone Client bleibt außerdem die Entscheidung des *PCI SSC* bezüglich der Sicherheit der einzelnen Geräte abzuwarten.

Für die Server-Umgebung, auf dem das Web Service sowie die Website betrieben werden, könnte *AWS (Amazon Web Services)* eingesetzt werden. Diese bieten eine skalierbare Cloud-Computing-Plattform, die im Dezember 2010 „*Level 1 Compliance*“ mit *PCI DSS* erreicht hat [40].

3.3.3 Bankeinzug

Noch weiter verbreitet als die Kreditkarte ist das Bankkonto, weswegen auch eine Zahlung mittels Bankeinzug optimal wäre.

In Europa wurde im November 2009 ein einheitliches Lastschriftverfahren namens *SEPA-Lastschrift (Single Euro Payments Area)* realisiert [36]. Dadurch kann von einem österreichischen Konto ein Bankeinzug von Konten anderer teilnehmender Länder durchgeführt werden. Mit Stand Januar 2012 sind das die 27 *EU*-Länder sowie Liechtenstein, Norwegen, Island, Monaco und die Schweiz [120].

Dafür muss eine Creditoren-Kennung (*CID*) bei der eigenen Bank beantragt und jedem Kunden muss eine Mandats(referenz-)nummer zugeordnet und mitgeteilt werden. Anstatt der in Österreich üblichen Kombination aus

Kontonummer und Bankleitzahl müssen nun *IBAN* (*International Bank Account Number*) und *BIC* (*Business Identifier Code*) gespeichert und angegeben werden. Kunden müssen der eigenen Bank ein unterzeichnetes Mandat übergeben, das ein mittels *CID* identifiziertes Unternehmen zum Einzug vom Konto berechtigt. Sollte ein Einzug ohne Vorlage dieser Berechtigung erfolgen, kann die Abbuchung bis zu 13 Monate nach Fälligkeitsdatum beanstandet werden. Bei Vorlage eines Mandats ist dies nur innerhalb von acht Wochen möglich. [36]

In Österreich existiert außerdem der e-Payment Standard *eps* [101]. Im Rahmen einer Online-Bezahlung wird der Konsument dabei auf sein gewohntes Online-Banking System weitergeleitet und bestätigt dort selbst die Zahlung des Betrags von seinem Konto. An einer internationalen Lösung für Bezahlverfahren mittels Mobiltelefonen wird mit Stand Januar 2012 noch gearbeitet [120].

3.3.4 paybox

Der Anbieter *paybox* bietet einen Bezahlservice für österreichische Mobilfunkkunden [84].

Alle privaten Vertragskunden der Mobilfunkanbieter *A1*, *T-Mobile*, *Orange* und *tele.ring* können den Dienst automatisch nutzen, der Rest gegen eine jährliche Gebühr von 19 € (Stand: 31.12.2011). Unter der Angabe der Mobilfunknummer können Zahlungen getätigt werden, die *paybox* über das Bankkonto des Kunden abrechnet.

Zur Absicherung der Transaktion wird bei jedem Vorgang an die Nummer des Kunden ein Anruf zur Autorisierung getätigt oder eine SMS geschickt. Bei einem Anruf muss die Transaktion durch die Eingabe eines PINs bestätigt werden, bei einem SMS muss „JA“ geantwortet werden [85]. Das *ÖIAT* (*Österreichische Institut für angewandte Telekommunikation*) ermittelte im Rahmen einer im Jahr 2009 veröffentlichten Studie, dass *paybox* zu diesem Zeitpunkt das sicherste elektronische Zahlungsmittel war [119].

Eine Integration in eine Smartphone-App ist möglich und für den Kunden praktisch, da die Rufnummer automatisch ermittelt werden kann und dadurch die manuelle Eingabe entfällt.

Pro Transaktion fallen auch hier, ähnlich zu einer Abrechnung via Kreditkarte, Gebühren für den Zahlungsempfänger an.

3.3.5 E-Wallet

Anstatt jede Transaktion einzeln direkt mit dem Kunden zu verrechnen, besteht die Möglichkeit, ein E-Wallet zu schaffen. Darunter ist zu verstehen, dass der Kunde über ein aufladbares Guthaben beim System verfügt und jede Transaktion gegen dieses Guthaben verrechnet wird.

Der Vorteil für den Kunden liegt darin, dass das potentielle Risiko sinkt und bei Verlust oder Missbrauch seiner Daten maximal das vorhandene Guthaben verbraucht werden kann.

Für den System-Betreiber sinken dadurch die Transaktionsgebühren für Bezahlvorgänge, da weniger oft mit externen Unternehmen abgerechnet wird. Wenn als Zahlungsmittel für Bestellungen exklusiv ein E-Wallet eingesetzt werden würde, würde auch die Sicherheit gegenüber Missbrauch steigen, wie in Abschnitt 3.5 demonstriert wird.

Anstatt eines eigenen Systems könnte auch ein externer Anbieter, wie z. B. das österreichische Unternehmen *paysafecard*, herangezogen werden [86].

3.4 Check-In

Verschiedene zur Durchführung des Check-Ins denkbare Mechanismen werden nachfolgend vorgestellt.

Manuell

Als einfachste Form der Implementierung könnte der Kunde den Betrieb sowie bei Bedarf seinen Sitzplatz angeben. Dies ist allerdings ein potentielles Sicherheitsrisiko und wird in Abschnitt 3.5 ausführlicher behandelt.

Location-based

Der Client könnte durch die Funktion der Lokalisierung am Smartphone erkennen, wenn sich ein Benutzer in oder um einen Betrieb aufhält. Dies ist allerdings nicht zuverlässig und ordnet dem Nutzer nicht automatisch einem Tisch zu.

Ende November 2011 hat *Google* eine Lösung für Indoor-Navigation im Zusammenhang mit ihrem populären Service *Google Maps* vorgestellt [78]. Diese erkennt nach eigenen Angaben zwar das Stockwerk und soll den Standort auf einige Meter genau ermitteln können, aber für die fehlerfreie Zuordnung zu einem Sitzplatzes ist das zu ungenau. Dieses Feature wird beispielsweise durch den Barometer, wie er bereits im *Samsung Galaxy Nexus* verbaut ist, unterstützt.

Insofern ist diese Möglichkeit nur als Hilfestellung zu einem manuellen Check-In zu betrachten.

Terminal

Es könnten im Betrieb, entweder auf jedem Tisch oder nur an bestimmten Orten, Terminals zum Check-In angebracht werden. Zur Interaktion mit den Clients könnte eine Kommunikation via *NFC* (Abschnitt 2.4.3) oder das Abfotografieren von 2D-Barcodes dienen. *NFC* scheint für den Benutzer einfacher, wird aber momentan nur von wenigen Smartphones unterstützt.

Das Abfotografieren des 2D-Barcodes kann theoretisch von beiden Instanzen vonstattengehen. Entweder kann der Terminal einen Code anzeigen, damit der Client ihn abfotografieren kann, oder der Terminal kann über eine Kamera einen 2D-Barcode vom Display des Benutzers einlesen.

Durch diese Kommunikation können Benutzer identifiziert und vorgenommene Reservierungen sowie Dine-In Bestellungen erkannt werden.

Ist der Terminal zentral angebracht, muss über diesen, sofern keine Reservierung vorhanden ist, die Auswahl des Tisches vorgenommen werden.

Ein Terminal am Tisch wäre komfortabel und könnte zusätzlich für reguläre Besucher des Betriebs die Verfügbarkeit basierend auf aktuellen Reservierungen anzeigen (z. B. „Reserviert ab 20:00 Uhr“). Auch Benachrichtigungen darüber, wenn eine Speise zur Abholung bereit ist, könnten auf diesem angezeigt werden. Diese Geräte müssten dem Einsatzgebiet entsprechend robust verarbeitet sein.

2D-Barcode auf Sitzplatz

Es könnte auch ohne Terminal ein 2D-Barcode via einem Printmedium am oder um einen Tisch angebracht werden, der abfotografiert werden muss, um den Check-In am Tisch durchzuführen. Dies birgt allerdings Sicherheitsrisiken, da der Code nicht dynamisch generiert werden kann und so als Bild gespeichert werden könnte, um gefälschte Check-Ins durchzuführen.

3.5 Missbrauch

Die grundsätzliche Problemstellung ist, dass sichergestellt werden muss, dass alle Bestellungen bezahlt werden.

Würde der Einsatz von Bargeld bzw. anderen herkömmlichen Bezahloptionen erlaubt werden, könnten Bestellungen zur Lieferung mit einer falschen Adresse getätigt werden. Möglich wäre auch eine Bestellung zur Abholung oder zum Dine-In, ohne dass der Kunde jemals im Betrieb erscheint.

Selbst wenn diese Optionen zur Bezahlung verboten werden würden, könnte mit Rahmen des Check-Ins Missbrauch geschehen. Kunden könnten regulär über das System bestellen und anschließend den Sitzplatz verlassen ohne zu bezahlen. Im herkömmlichen Geschäftsprozess würde das die Aufmerksamkeit des Personals auf sich ziehen, aber sobald dieses daran gewöhnt wäre, dass Kunden nicht bei ihnen bezahlen, kann auch nicht erwartet werden, dass sie bei jedem Kunden den Status im POS-System abrufen. Auch ein falscher Check-In wäre bei Wahl eines unsicheren Mechanismus möglich, um Waren zu bestellen, obwohl sich der Benutzer nicht im Betrieb befindet.

Nachfolgend werden denkbare Lösungswege behandelt.

Exklusiver Einsatz eines E-Wallets

Wäre eine Bestellung nur mit genügend Guthaben auf dem E-Wallet möglich, könnte jeder Artikel automatisch von diesem abgebucht werden. Somit würde jeder Artikel, auch während eines Check-Ins, sofort bezahlt werden.

Ein Nachteil wäre, dass Kunden dieses E-Wallet bereits vor ihrer ersten Bestellung mit Guthaben aufladen müssten, was eine Hemmschwelle zur allgemeinen Benutzung darstellen könnte.

Reservierung von Beträgen

Sowohl bei Kreditkarten als auch bei *paybox* ist es möglich, einen gewissen Betrag zu reservieren. Das bedeutet, dass der Betrag nicht sofort verrechnet wird, aber der Verfügungsrahmen des Kunden um den fälligen Betrag verringert wird, sofern dieser ausreicht. Sollte dies nicht der Fall sein, schlägt die Reservierung des Betrags und somit die Bestellung fehl.

Sollte der Benutzer also eine Leistung in Anspruch nehmen, ohne anschließend zu bezahlen, kann der reservierte Betrag abgebucht werden, und durch die Reservierung ist sicher gestellt worden, dass der Betrag gedeckt ist.

Dieser Mechanismus könnte bei Einzelbestellungen im Rahmen eines Check-Ins zum Einsatz kommen.

Eindeutige Identifikation von Benutzern

Um im Bedarfsfall Benutzer belangen zu können, müssten diese eindeutig identifiziert sein. Dies scheint schwer, da dieser Prozess umständlich wäre (z. B. Übermittlung eines amtlichen Dokuments), und je einfacher er gestaltet werden würde, desto mehr Raum für potentielle Täuschung würde sich öffnen.

Kapitel 4

Soziologische Studie

Zur Evaluierung des im vorigen Kapitel vorgestellten Konzeptes wurde eine soziologische Studie durchgeführt.

4.1 Methodik

Die Umfrage sollte mittels eines Online-Formulars durchgeführt werden. Im Gegensatz zu persönlicher Befragung oder per Telefon liegt der Vorteil darin, dass eine Umfrage dieser Art einfacher sowie schneller abzuwickeln ist und die Rücklaufquote höher ausfallen sollte. Als Nachteil ist in Kauf zu nehmen, dass die Antworten auf die gegebenen Möglichkeiten beschränkt sind und nicht in die Tiefe gefragt werden kann.

Da das konzipierte System großteils auf die mobile Nutzung abzielt, definierte sich die Zielgruppe als Besitzer von Smartphones, die im Durchschnitt mindestens zweimal pro Monat das Angebot von Gastronomiebetrieben in Anspruch nehmen. Alter und Einkommen waren somit nur indirekte Faktoren.

Der Fragenkatalog wurde unter Zuhilfenahme der Ratschläge von [100] und [102] erstellt. Im Allgemeinen ist darauf geachtet worden, dass die Fragestellungen einfach und verständlich waren und keine Antworten oder Tendenzen suggeriert wurden. Die Antworten sollten alle Möglichkeiten abdecken, ohne zu umfangreich zu werden. Auch die gesamte Umfrage wurde so kurz wie möglich gehalten, damit die Beantwortungsdauer 5 Minuten nicht überschreitet. Bei der Durchführung der Umfrage wurde der Befragte immer über seinen aktuellen Fortschritt bei der Beantwortung der Fragen informiert, wie in Abbildung 4.1 oben zu sehen ist.

Gestartet wurde mit Fragen, die das Interesse wecken, aber gleichzeitig einfach zu beantworten sind. In diesem Fall behandelten die Fragen Themen wie die eingesetzte Technologie.

Wichtig war die logische Gruppierung. Zuerst beschäftigten sich die Fragen mit der aktuellen Nutzung von Gastronomieangeboten und welche Fak-

toren Einfluss auf die Auswahl eines Betriebs haben. Es folgten Fragen über das vorgestellte Konzept, um das Interesse an den vorgestellten Möglichkeiten zu erheben.

Bei Fragen, die als Antwortmöglichkeit eine Skala hatten, ist darauf geachtet worden, dass die Anzahl der Werte gerade war, damit immer eine Tendenz erkennbar war und neutrale Antworten vermieden wurden.

Abgeschlossen wurde mit Fragen zur Person.

Vor der Durchführung wurde die Umfrage an einer kleinen Gruppe getestet, damit allfällige Schwächen und unlogische Abläufe gefunden und behoben werden konnten.

Der vollständige Fragenkatalog findet sich in Anhang A.

4.2 Durchführung

Die Studie wurde mit Hilfe des Services www.rationalsurvey.com online durchgeführt. Der Link zu der Umfrage wurde mittels des sozialen Netzwerks *Facebook* sowie per E-Mail nach dem Schneeballprinzip verteilt. Im Zeitraum vom 29. Juli bis 02. August 2011 füllten 55 Personen den Fragebogen vollständig aus.

4.3 Auswertung

Es wird eine Auswahl der interessantesten Schlüsse präsentiert, die aus den Umfragedaten gewonnen werden konnten.

Population

Teilgenommen haben 55 Personen, davon 37 Männer (67 %) bzw. 18 Frauen (33 %). 35 Personen (63,5 %) gaben an, zwischen 19 und 25 Jahren alt zu sein, 18 (33 %) zwischen 26 und 35 Jahren und 2 (3,5 %) zwischen 36 und 50 Jahren. Der grobe Altersdurchschnitt der Teilnehmer lag somit bei 25,5 Jahren.

Smartphones

Je 25 Personen, somit je 45 %, sind im Besitz eines Smartphones mit *Android* bzw. *iOS* als Betriebssystem. Eine Person verwendet *Windows Phone 7*, die übrigen vier *Symbian*, wobei davon drei angeben, unzufrieden zu sein, und auf *Android* oder *iOS* wechseln wollen. Je ein Anwender (4 % der Benutzer) ist unzufrieden mit *Android* bzw. *iOS* und will auf das jeweils andere System wechseln.

Somit lassen sich *Android* und *iOS* als klar gleichwertige Plattformen identifizieren. Diese Zahlen entsprechen auch in etwa der weltweiten Verteilung der Betriebssysteme, wie in Abschnitt 2.1.2 gezeigt wurde.

Gastronomie für Smartphone-Besitzer

Seite 1 / 5 | Technologie Einsatz Page 1

1) Welches Betriebssystem ist auf Ihrem Smartphone im Einsatz?

- Android
- BlackBerry OS
- iOS (iPhone)
- Symbian (Nokia)
- webOS
- Windows Phone 7
- Sonstiges: (Bitte angeben)

2) Sind Sie mit dem Betriebssystem zufrieden?

- Ja
- Nein, ich plane einen Wechsel auf: (bitte angeben)

3) Besitzen Sie ein Tablet?

- Ja, ein iPad
- Ja, und zwar: (bitte angeben)

- Nein, aber ich plane eine Anschaffung in den nächsten sechs Monaten
- Nein

Abbildung 4.1: Umfrage mit Rational Survey

Tablets

11 der 55 Befragten (20 %) sind im Besitz eines Tablets und 9 weitere (16 %) geben an, in den nächsten sechs Monaten einen Kauf tätigen zu wollen. Somit lässt sich eine zu erwartende Steigerung von der Verbreitung von Tablets ablesen. 10 der 11 Tablet-Besitzer (91 %) kauften ein *iPad* (*iOS*), der einzige Ausreißer war ein *ViewSonic gTablet* (*Android*). Auch diese Zahlen entsprechen in etwa der weltweiten Verbreitung von Tablets.

Von den zehn *iPad*-Besitzern verwenden zwei *Android* am Smartphone, wohingegen die Person mit dem *Android*-Tablet das Betriebssystem auf beiden Geräten einsetzt. Von neun Personen, die eine Anschaffung eines Tablets planen, verwenden je vier *Android* bzw. *iOS* auf ihrem Smartphone. Die übrige Person besitzt ein Smartphone mit *Windows Phone 7*.

Interessant ist, dass mit einer Ausnahme alle Tablet-Besitzer männlich sind. Von denen, die eine Anschaffung planen, sind 7 männlich, 2 weiblich.

Acht der elf Tablet-Besitzer (73 %) geben an, ihr Gerät zu keiner Zeit während eines Gastronomiebesuchs bei sich zu haben. Zwei Personen (18 %) würden es immer bei sich haben, eine Person (9 %) nur zur Mittagszeit.

Nutzung von Gastronomie-Angeboten

An einem Arbeitstag nehmen die Befragten zur Mittagszeit großteils nicht-gastronomische Angebote in Anspruch. 28 Personen (51 %) versorgen sich u.a. über einen Supermarkt, 22 (40 %) bringen Speisen von Zuhause mit und 11 (20 %) bereiten auch Speisen am Arbeitsplatz zu. 20 (36 %) nehmen das Angebot einer Kantine wahr, 8 (15 %) haben ein Stamm-Restaurant, 16 (30 %) besuchen verschiedene Restaurants und 13 (24 %) nehmen Lieferdienste in Anspruch (Abbildung 4.2).

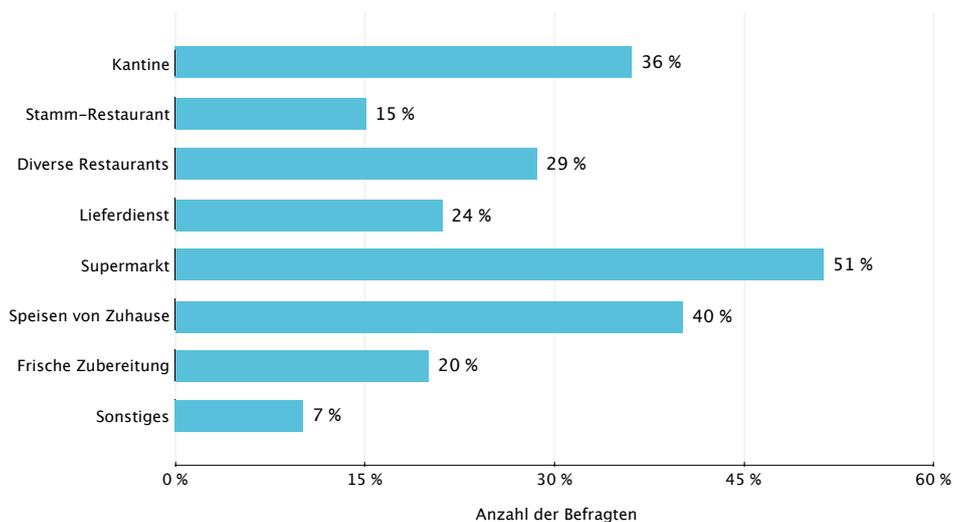


Abbildung 4.2: Versorgung zu Mittag am Arbeitsplatz

Von den 20 Personen, die angeben, mittags eine Kantine zu besuchen, tun 12 (60 %) das im Durchschnitt über 10 Mal pro Monat. Von diesen 20 Personen geben auch nur 6 (30 %) an, mittags andere gastronomische Angebote in Anspruch zu nehmen. Somit lässt sich erkennen, dass andere gastronomische Angebote auf wenig Interesse stoßen, wenn eine Kantine genutzt wird. Es kann angenommen werden, dass dies der Fall ist, da diese in der Regel günstiger und örtlich näher sind als reguläre gastronomische Angebote. Auch ist das Essen meist schon vorbereitet, was den Aufenthalt zusätzlich verkürzt.

Wie Abbildung 4.3 zeigt, ist die Kantine auch der Gewinner bei den im Durchschnitt am öftesten besuchten Gastronomiebetrieben. Rechnet man die Angabe „mehr als 10 Mal pro Monat“ mit 13 Besuchen, besuchen alle Personen im Durchschnitt 3,86 Mal im Monat eine Kantine, gefolgt von einem Club/einer Bar mit 3,70 Mal und einem Restaurant zu Mittag mit 3,39 Mal. Auch die anderen Angeboten werden im Durchschnitt 2 bis 3 Mal im Monat in Anspruch genommen.

Die Bezahlung wird bevorzugt mit Bargeld getätigt, wie 45 Personen (82 %) angeben.

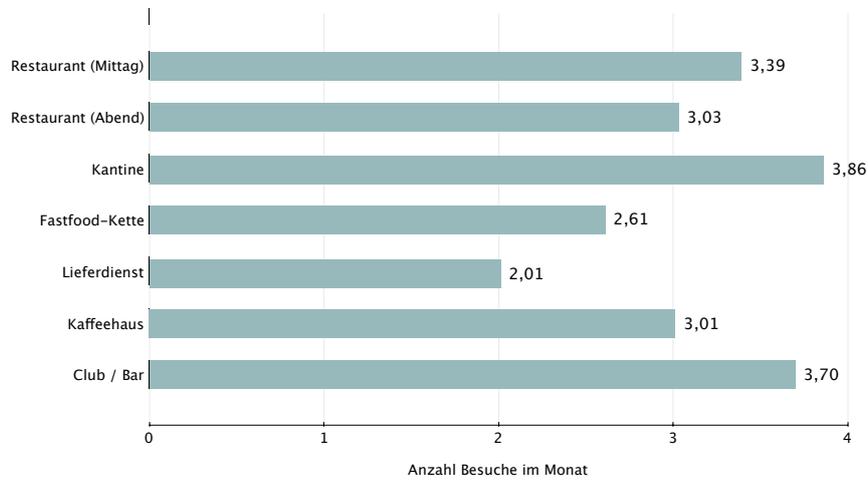


Abbildung 4.3: Anzahl Besuche von Gastronomiebetrieben pro Monat

Die Gesamtkosten halten 38 Personen (69 %) von einer häufigeren Inanspruchnahme der Gastronomie-Angebote ab. Als zweithäufigster Grund wird von 25 Befragten (45 %) genannt, dass der Aufenthalt im Lokal zu viel Zeit kostet. Alle Antworten sind in Abbildung 4.4 zu sehen.

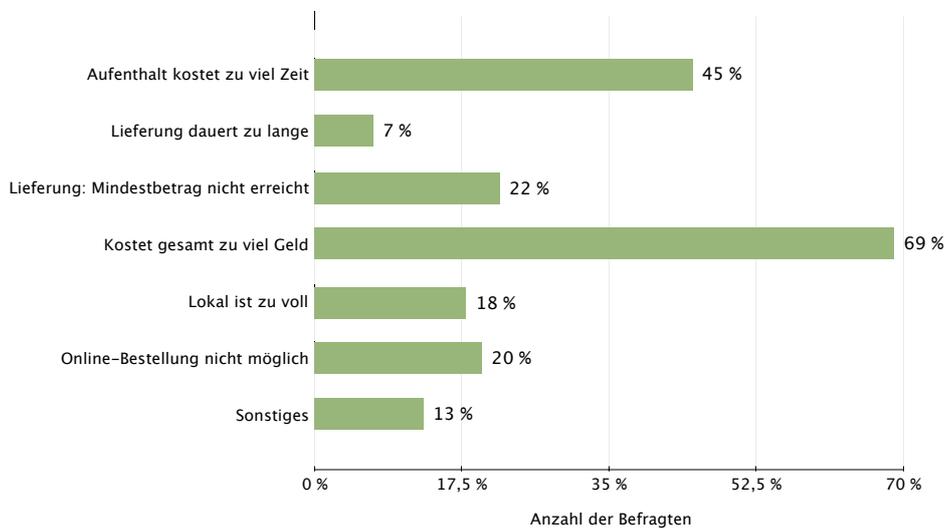


Abbildung 4.4: Faktoren gegen Nutzung von Gastronomie-Angeboten (Mehrfachnennung)

Auswahl des Lokals

Auf die Frage wie die Aufmerksamkeit auf ein Lokal gelenkt wird (Abbildung 4.5), geben 51 von 55 Personen (93 %) an, dass die Empfehlung von Bekannten eine Rolle spielt. Auch der Zufall ist für 43 (78 %) der Befragten ein Faktor. Abgeschlagen teilen sich Werbung/Postwurfsendung und Location-based Services (z. B. *Google Places, Qype*) Platz 3 mit je 14 Personen (25 %), die diese Punkte als relevant identifizieren.

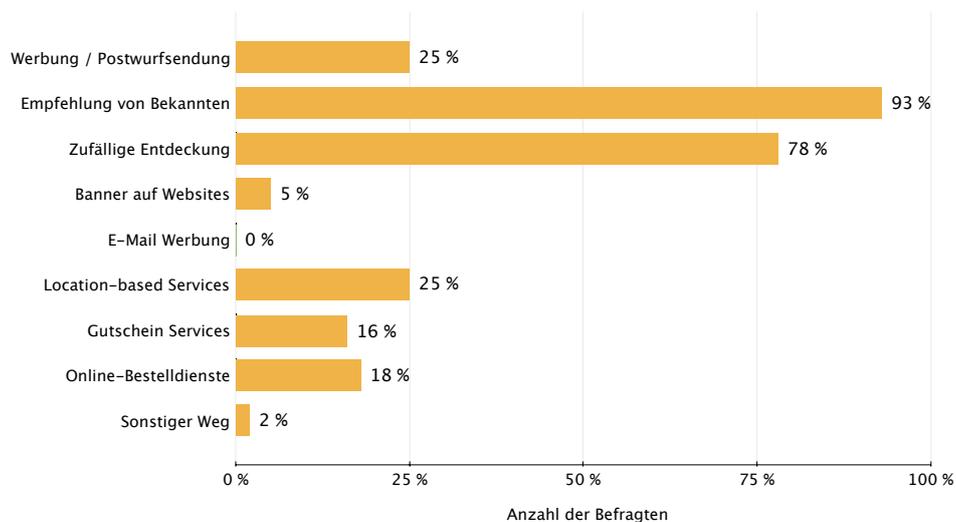


Abbildung 4.5: Erregung von Aufmerksamkeit für einen Gastronomiebetrieb (Mehrfachnennung)

Auch bei der eigentlichen Auswahl des Lokals (Abbildung 4.6) ist 46 (84 %) der Befragten die Meinung von Bekannten entweder sehr wichtig (26) oder wichtig (20). Übertroffen wird diese Angabe nur von den angebotenen Speisen selbst, die 53 (96 %) als (sehr) wichtig empfinden.

Werden die Angaben „sehr wichtig“ mit 4, „wichtig“ mit 3, „wenig wichtig“ mit 2 und „unwichtig“ mit 1 gewichtet und davon der Durchschnitt ermittelt, reihen sich nach den zwei genannten Faktoren, die mit 3,56 (Angebotene Speisen) und 3,20 (Kritik von Bekannten) gewichtet wurden, die Preisklasse (3,07) und der Eindruck von außen (2,87) auf Platz 3 und 4 ein.

An letzter Stelle befinden sich die Bezahlungsmöglichkeiten (1,65), was daher rühren wird, dass die deutlich bevorzugte Zahlungsmethode, Bargeld, im Allgemeinen überall akzeptiert wird.

Interesse an speziellen Angeboten, wie z. B. Mittagsmenüs, haben 36 (65 %) der Befragten. 26 von diesen 36 Personen (72 %) informieren sich über die Website und 19 (53 %) über die Aushänge eines Betrieb. 8 Personen (22 %) geben an, sich gerne mehr darüber informieren zu wollen, aber keinen effizienten Weg zu kennen.

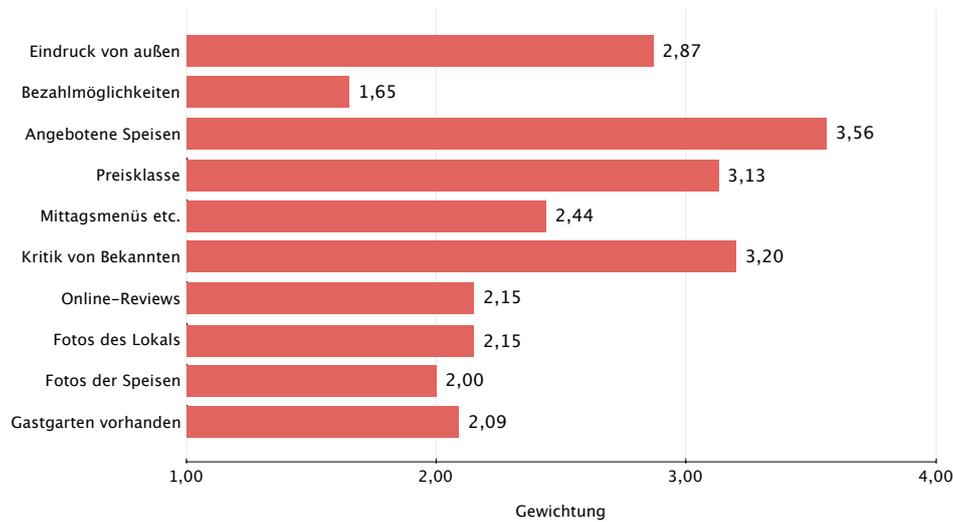


Abbildung 4.6: Wichtigkeit von Faktoren für einen Gastronomiebesuch

Nutzung von Services

Im Durchschnitt reservieren die 55 Befragten 0,74 Mal pro Monat Sitzplätze in einem Gastronomiebetrieb. Telefonisch bestellt wird 1,38 Mal. Dem gegenüber steht eine Online-Bestellung von 1,01 Mal pro Monat.

Bei der Bestellung ist ein wesentlicher Unterschied zwischen Männern und Frauen festzustellen. Männer bestellen im Durchschnitt 1,28 Mal pro Monat telefonisch und 1,23 Mal online, beide halten sich hier die Waage. Frauen hingegen bestellen 1,58 Mal telefonisch und im Vergleich nur 0,56 Mal online.

16 Personen (31 %) geben an, gerne online bestellen zu wollen, wobei 12 (75 %) kein System dazu kennen und 4 (25 %) kein Vertrauen in die ihnen bekannten haben. Die 14 Personen (25 %), die kein Interesse an einer Online-Bestellung haben, bestellen mit 1,50 Mal pro Monat am häufigsten telefonisch, ganz knapp vor den 16 Personen, die gerne online bestellen würden, mit 1,47 Mal. Die Personen, die bereits online bestellen, nutzen den telefonischen Weg trotzdem 1,26 Mal pro Monat.

Mjam.net bzw. *willessen.at* – derselbe Dienst unter anderem Namen – hat sich als die am häufigsten verwendete Online-Bestellplattform herausgestellt. 15 Personen (60 % der Online-Besteller) machen davon Gebrauch. Die Benutzer sind der Plattform treu, und nur eine Person gibt an, zusätzlich andere Online-Systeme zu verwenden. Auch die 4 Personen, die *NetKellner.at* verwenden, nutzen ansonsten kein anderes Online-Bestellsystem.

Interesse an weiteren Services

46 der 55 Befragten (83 %) signalisierten Interesse an einer Sitzplatz-Reservierung über ein (mobiles) Online-System. Werden die Antworten erneut von 4 („Ja, auf jeden Fall“) bis 1 („Nein, keinesfalls“) gewichtet, beträgt der Durchschnitt 3,04.

Interesse an einer Bestellung im Voraus zur späteren Konsumation im Lokal hätte ca. die Hälfte aller Befragten, am ehesten bei einer Kantine (2,65), gefolgt von Restaurant (2,58) und Fast food Kette (2,35). Allerdings haben nur 4 der 55 Personen (7 %) für alle drei Arten von Betrieben eine negative Tendenz angegeben. Somit hätten 51 Personen (93 %) in zumindest einer Art von Betrieb Interesse an der Bestellung im Voraus.

Von diesen 51 Personen würden 33 (65 %) entweder mit Sicherheit (8) oder eventuell (27) sofort bei Bestellung bezahlen.

Eine grundsätzlich negative Tendenz hinsichtlich einer Bestellung via Smartphone bzw. Tablet direkt im Lokal zeigen nur 8 der 55 Befragten (15 %). Alle anderen signalisieren zumindest in einer Art von Betrieb Interesse. Führend sind Kantine (2,67), Fast food Betrieb (2,62) und Restaurant (2,58). Klar abgeschlagen sind Kaffeehaus und Club/Bar mit im Durchschnitt je 2,20.

Bei der Tendenz bezüglich Bezahlung via Smartphone bzw. Tablet zeigt sich ein ähnliches Bild wie bei der Bezahlung bei der Bestellung im Voraus. Über alle Arten hinweg beträgt der Wert 2,47. Führend ist die Kantine mit 2,62 – Schlusslicht der Club/die Bar mit 2,20.

Die Zahlungsinformationen vor der Bezahlung im System hinterlegen würde nur eine Person „auf jeden Fall“. Gesamt wird die Frage im Durchschnitt mit 2,31 beantwortet. Wesentlicher Grund, der von der Hinterlegung abhält, ist das mangelnde Vertrauen. Von 33 Personen (60 %) mit negativer Tendenz geben 18 (55 % der 33 Personen) an, nur Diensten zu vertrauen, die sie bereits kennen, und 10 (30 %) müssten zuerst positive Erfahrungen von Bekannten oder aus den Medien hören. 5 (15 %) bezahlen aus Prinzip nicht online.

27 Befragte (49 %) würden bei der Bezahlung über Online-Systeme kein Trinkgeld geben, 8 (15 %) weniger als normal und 20 (36 %) denselben Betrag wie bei normaler Bestellung.

Integrierte Bonussysteme, wie dass z. B. jede 10. Speise einer Art kostenlos ist, stößt mit 3,53 auf hohes Interesse.

Nach erfolgreicher Abwicklung des Bestellprozesses würden 29 (53 %) direkt mobil eine Bewertung abgeben, die auch Bilder enthalten könnte. 4 (14 % der 29 Personen) nur bei Zufriedenheit, 9 (31 %) nur bei Unzufriedenheit und 16 (55 %) immer. Somit besteht keine unmittelbare Gefahr, dass Reviews einseitig gefärbt werden.

Community Features, wie z. B. private Nachrichten, erregten mit 1,85 wenig Interesse.

4.4 Ergebnis

Die Existenzberechtigung für das in Kapitel 3 vorgestellte System ist durch die Studie bestätigt worden.

83 % der befragten Personen signalisierten Interesse an einer Sitzplatzreservierung, 93 % haben angegeben, in zumindest einer Art von Betrieb die Möglichkeit Essen vorzubestellen, um es direkt im Betrieb zu konsumieren, nutzen zu wollen, und 65 % dieser Personen würden bei Bestellung auch sofort bezahlen. Ebenfalls in mindestens einer Art von Betrieb würden 85 % aller Befragten während des Aufenthalts Essen über ein mobiles System bestellen wollen.

Deutlich beeinflusst worden ist das Konzept durch das Ergebnis, dass die Meinung von Freunden und Bekannten bei der Entscheidung für einen Gastronomiebetrieb eine große Rolle spielt. Deswegen wurde die soziale Komponente wesentlich stärker ausgearbeitet als sie ursprünglich geplant war.

Kapitel 5

Prototypische Implementierung

Aus Sicht der Konsumenten liegt der größte Neuheitswert des Konzepts darin, dass mittels des Smartphones in den Geschäftsprozess eines Gastronomiebetriebs eingegriffen werden kann. Daher wurde ein Prototyp dieses Clients für das Betriebssystem *iOS* entwickelt, mit dem anschließend ein Usability Test durchgeführt wurde, dem sich Kapitel 6 widmet.

Damit das Erlebnis für den Benutzer möglichst realistisch ist, durften die präsentierten Daten nicht statisch oder zufällig sein, und die Ladezeiten sollten ebenfalls der Praxis entsprechen. Deswegen wurde zusätzlich ein Prototyp des Web Services entwickelt, auf den in Abschnitt 5.2 eingegangen wird.

5.1 Entwicklungsprozess

Der gesamte Entwicklungsprozess erfolgte agil, aber ohne dabei einer bestimmten Methode wie z. B. *Extreme Programming (XP)* oder *Scrum*, zu folgen. Agil bedeutete in diesem Fall, dass vor dem Beginn der Implementierung sogenannte User Stories definiert worden sind. Diese beschreiben in kurzer Form jeweils einen gewissen Prozess, den der Benutzer mittels der Software durchführen können soll. Im Zuge der Implementierung werden nur User Stories abgearbeitet. Ein Gegensatz zu traditioneller Softwareentwicklung ist, dass das Datenmodell flexibel ist und nicht komplett in einem Schritt entwickelt wird.

Die in Tabelle 5.1 präsentierten User Stories wurden aus Sicht des Benutzers des Smartphone Clients definiert. Für die Umsetzung mussten je nach Bedarf auch die zugehörigen Schnittstellen des Web Service implementiert werden.

Auf die Implementierung von manchen Features wurde auf Grund des geringen Neuheitswertes oder mangelnder Relevanz für den Prototypen verzichtet. Beispiele dafür sind die Benutzerregistrierung und das Auffinden von Freunden über soziale Netzwerke.

Nr.	User Story	Abhängigkeit
#01	User-Login	-
#02	Startseite	#01
#03	Benutzer-Profil: Stammdaten	#01
#04	Benutzer-Profil: Freundeslisten	#03
#05	Betrieb: Stammdaten	-
#06	Betrieb: Reviews	#05
#07	Betrieb: Speisekarte: Liste	#05
#08	Betrieb: Detail Speise, Zu Warenkorb	#06
#09	Betrieb: Reservierung	#05
#10	Betrieb: Warenkorb	#05
#11	Betrieb: Bestellung senden	#11
#12	Abgabe Review	#11
#13	Check-In: Ansicht	#05
#14	Check-In: Notifications	#13
#15	Suche	#03, #05, #07
#16	Favoriten Betrieb, Speise	#05, #08

Tabelle 5.1: User Stories zur Entwicklung des Prototypen

Auf die Entwicklung von automatisierten Tests wurde für den Prototypen verzichtet. Während der Entwicklung wurden sämtliche neuen Features und deren Integration mit den bestehenden manuell getestet.

5.2 Web Service

5.2.1 Konzeption

Das Web Service wurde für die prototypische Implementierung bewusst einfach gehalten und dessen Fokus liegt in der Kommunikation mit dem Smartphone Client. Auf die Implementierung einer Schnittstelle zu einem POS-System wurde verzichtet, und auf Sicherheit sowie Verhalten bei hoher Last wurde wenig Rücksicht genommen.

Die Wahl der Technologie fiel auf die Scriptsprache *PHP* sowie das Datenbank System *MySQL* in der jeweils aktuellen Version 5.3 bzw. 5.5. Beide Technologien sind im Web weit verbreitet und werden aktiv weiterentwickelt.

Für *PHP* existieren diverse Frameworks, deren Aufgabe es ist, Entwicklern die Arbeit zu erleichtern, indem bestimmte Funktionalitäten von ihnen angeboten oder übernommen werden. Für diesen Prototypen wurde jedoch keines dieser Frameworks eingesetzt, um auf diese zusätzliche Schicht von Komplexität zu verzichten.

5.2.2 Architektur

Das *World Wide Web Consortium (W3C)* definierte „Web service“ im Jahr 2004 wie folgt [116]:

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

SOAP

SOAP ist ein Protokoll für den Austausch von strukturierter Information im *XML*-Format [115]. Der Begriff war ursprünglich ein Akronym für „*Simple Object Access Protocol*“, das aber seit Version 1.2 nicht mehr gebraucht wird.

Die verfügbaren Schnittstellen mit den notwendigen Parametern und den verarbeiteten Datenstrukturen werden über eine *WSDL*-Datei (*Web Services Description Language*) offen gelegt [117]. Die Struktur dieser Datei ist ebenfalls im *XML*-Format, und Clients können diese direkt verwenden, um zu erfahren, wie mit dem Web Service kommuniziert werden kann.

SOAP wird jedoch von dem *iOS*-Framework auch in der aktuellen Version 5 nicht unterstützt. Deswegen wurde beim Design des Web Services auf die Alternative *REST* (Representational state transfer) zurückgegriffen. Auch populäre Web Services, wie z. B. die der sozialen Netzwerke *Facebook* und *Twitter* oder des Foto Services *Flickr*, verwenden dieses Pattern.

REST

REST ist ein Architekturstil und im Gegensatz zu *SOAP* kein klar definiertes Protokoll. Der Begriff *REST* wurde von *Roy Fielding*, einem der Autoren der Spezifikationen 1.0 und 1.1 des *Hypertext Transfer Protocols (HTTP)*, in seiner Dissertation im Jahr 2000 definiert [9].

Neben den *REST*-Komponenten (Origin Server, Gateway, Proxy, User Agent) und den *REST*-Verbindungssteilen (Client, Server, Cache, Resolver, Tunnel) werden die *REST*-Datenelemente als zentrales Architektur-Element beschrieben. Das Hauptelement davon sind die Ressourcen, die durch einen Identifikator klar ansprechbar sind und in einer bestimmten Repräsentation (inkl. zugehöriger Meta- und Kontrolldaten) ausgeliefert werden. Bei Ressourcen kann es sich z. B. um Datenobjekte oder Bilddateien handeln.

Fielding beschreibt sechs Eigenschaften, die einen *REST*-Dienst definieren:

1. Client-Server: Eine klare Trennung der Aufgabenbereiche von Server (Datenspeicherung) und Client (User Interface) und die dadurch ermöglichte eigenständige Entwicklung beider Komponenten.
2. Stateless: Die Kommunikation erfolgt zustandslos, und jede Anfrage des Clients muss alle notwendigen Informationen enthalten, so dass der Server keinen Gebrauch des Kommunikationskontexts machen muss.
3. Cache: Es muss in der Antwort auf eine Anfrage implizit oder explizit ausgewiesen werden, ob die Daten vom Client zwischengespeichert werden dürfen.
4. Uniform Interface: Verwendung einer einheitlichen Schnittstelle zwischen den Clients und dem Server-System.
5. Layered System: Die Server-Seite kann aus mehreren Komponenten bestehen, die unterschiedliche Aufgaben, wie z. B. Caching, übernehmen können. Für den Client ist nicht sichtbar, mit welcher Komponente er kommuniziert.
6. Code-On-Demand: Als optionale Eigenschaft wird beschrieben, dass der Server auf Anfrage Programm-Code ausliefern kann, der am Client ausgeführt wird.

Was in der Dissertation nicht explizit definiert wurde, aber inzwischen Standard ist, ist, dass die Anfragen von den *HTTP*-Request-Methoden *GET*, *POST*, *PUT* und *DELETE* Gebrauch machen, um anzugeben, in welcher Form die offen gelegten Ressourcen verändert werden sollen. Werden die Objekte in textueller Form repräsentiert, kann oftmals über einen Parameter in der URL bestimmt werden, in welchem Format (z. B. *JSON*, *XML*) diese erfolgen soll.

Im Gegensatz zu *SOAP* verfügt *REST* über keine maschinenlesbare Beschreibungssprache wie *WSDL*, weswegen die Dokumentation auf anderem Weg, z. B. über eine Website, offen gelegt und jeder Aufruf manuell implementiert werden muss.

Aufbau

Welche Ressourcen definiert wurden und wie diese ausgeliefert bzw. manipuliert sowie gespeichert werden, wird in den folgenden beiden Abschnitten beschrieben.

Zur Authentifizierung eines Benutzers wurde eine gewöhnliche Anmeldung mittels Loginname und Passwort implementiert. Zur einfacheren Handhabung wurden *PHP*-Sessions verwendet, wodurch sich der Benutzer einmal authentifizieren muss und dieser Status für die Dauer seiner folgenden Anfragen gespeichert bleibt. Damit wird zwar „Stateless“ als einzige *REST*-

Eigenschaft verletzt, aber diese Form der Implementierung hält den Aufwand sowohl auf Client- als auch auf Server-Seite geringer. Sollte es gewünscht sein, dass diese Bedingung nicht verletzt wird, könnten die Login-Daten mit jeder Anfrage erneut gesendet werden. Darauf wurde aber bei diesem Prototyp verzichtet.

Für die optionale Eigenschaft „Code-On-Demand“ besteht einerseits kein Bedarf und andererseits erlaubt *Apple* diese Vorgehensweise aus Sicherheitsgründen für *iOS* Apps nicht [31].

Auf die Implementierung einer Registrierung wurde auf Grund des geringen Neuheitswerts sowie des nicht vorhandenen Mehrwerts für die Usertests verzichtet. Die Daten der Benutzer werden zwar in der Datenbank abgelegt, jedoch sind diese Datensätze manuell zu verwalten.

Für die Auslieferung der Daten wurde ausschließlich das *JSON*-Format (*JavaScript Object Notation*) verwendet, da dieses sowohl von dem *Android*- als auch dem *iOS*-Framework einfach und effizient verarbeitet werden kann. Die Implementierung von Alternativrepräsentationen würde für den Prototypen somit keinen Mehrwert schaffen. Nur Bilddateien werden als Binärdaten ausgeliefert. Ein Beispiel für eine Antwort im *JSON*-Format findet sich im folgenden Abschnitt.

5.2.3 Schnittstellen

Folgende Schnittstellen werden durch das entwickelte Web Service bereit gestellt. Aus den URLs sind die verschiedenen Ressourcen einfach ablesbar.

```
GET /lists/places/excerpt
GET /lists/places/friends
GET /lists/places/favorites

GET /place/:id
GET /place/:id/image
GET /place/:id/menu
GET /place/:id/table_plan
GET /place/:id/taken_tables
GET /place/:id/reviews
POST /place/:id/favorite/create
POST /place/:id/favorite/destroy

GET /order/:id
POST /order/create
POST /order/open
POST /order/:id/add_items
POST /order/:id/pay

POST /reservation/create

POST /check_in/create
POST /check_in/:id/set_order_dinein
POST /check_in/:id/check_out
```

```

GET /meal/:id
GET /meal/:id/image
GET /meal/:id/reviews
POST /meal/:id/favorite/create
POST /meal/:id/favorite/destroy

GET /menu_category/:id/image
GET /cuisine/:id/image

GET /review/:id
POST /review/create

GET /user/:id
GET /user/:id/reviews
GET /user/:id/image
GET /user/:id/followers
GET /user/:id/friends
POST /user/:id/follow/create
POST /user/:id/follow/destroy

POST /account/auth
GET /account/lists/notifications
GET /account/lists/orders
GET /account/lists/orders/to_review

GET /search

```

Wie zu sehen ist, wurde auf die *HTTP*-Request-Methoden *PUT* und *DELETE* verzichtet und durch *POST* mit entsprechendem URL-Parameter ersetzt. Hiermit wird einer aktuellen Entwicklung im Design von *REST*-Schnittstellen gefolgt, die von populären APIs, wie z. B. der von *Twitter* [111], angeführt wird. Alle Anfragen, die rein zur Abfrage von Daten dienen, erfolgen via *GET* und alle Anfragen, die der Manipulation von Daten dienen, via *POST*.

Je nach Bedarf werden die Anfragen um zusätzliche Parameter ergänzt. Ein Beispiel für eine Suchanfrage:

```
GET /search?q=mario&scope=place
```

Die Antwort im *JSON*-Format:

```

{
  "results": [
    {
      "id": "1",
      "name": "Pizzeria Vesuvio da Mario",
      "street": "Arnsteingasse 30",
      "zip": "1150",
      "city": "Wien",
      "country": "Austria",
      "food_delivery": true,

```

```
    "food_pickup": true,
    "food_dinein": true,
    "contact_tel": "+43 1 892 38 71",
    "contact_email": "",
    "contact_www": "http://www.pizzeria-vesuvio-da-mario.at/",
    "nr_reviews": 2
    "images": [
      "/images/place/1_1.jpg",
      "/images/place/1_2.jpg"
    ]
  }
],
"success": true
}
```

Alle Anfragen sind nach demselben Prinzip aufgebaut. Wenn eine Anfrage über *POST* erfolgt, wird das betreffende Objekt nach Durchführung der beauftragten Manipulation zurückgeliefert.

Bestellung

Eine Serie von Anfragen, die etwas genauere Erklärung bedarf, ist die der Bestellungen.

Stellt der Benutzer am Client eine Bestellung zum Dine-In, zur Abholung oder zur Lieferung zusammen, kann die gesamte Bestellung mit allen Informationen wie Artikel, Adresse und Bezahlung in einem Schritt übermittelt werden.

```
POST /order/create
```

Befindet sich der Benutzer aber im Betrieb und hat mittels der Anfrage

```
POST /check_in/create
```

erfolgreich einen Check-In durchgeführt, zählen alle über die Dauer des Besuches bestellten Artikel zu einer Bestellung. Um Artikel in mehreren Schritten übermitteln zu können, um z. B. das Dessert nach Konsumation der Hauptspeise zu bestellen, muss die Schnittstelle für die Bestellung in mehrere Schritte aufgeteilt werden. Mittels

```
POST /order/open
```

wird eine Bestellung initialisiert, zu der über

```
POST /order/:id/add_items
```

laufend Artikel hinzugefügt werden können. Nach jedem dieser Aufrufe erhält der Client den aktuellen Gesamtbetrag der Bestellung unter Berücksichtigung der verfügbaren Bonusprogramme.

Für den finalen Schritt zum Abschluss und zur Bezahlung der Bestellung, kommt folgende Schnittstelle zum Einsatz:

```
POST /order/:id/pay
```

Dadurch wird eine Bestellung als abgeschlossen markiert und über die angegebene Bezahlmethode abgerechnet. Auf die implementierten Zahlungsmethoden wird im folgenden Abschnitt eingegangen.

Zur Sicherheit muss der Client den bezahlbaren Betrag hier erneut übergeben, der aus der Antwort der zuletzt erfolgten `/order/:id/add_items` Abfrage stammen sollte. Dies ist notwendig, damit der Benutzer auch tatsächlich den Betrag, der ihm im Client angezeigt wird, bezahlt. Der Server überprüft den übergebenen Wert mit dem berechneten Gesamtbetrag und führt bei Erfolg die Bezahlung durch.

Abgefragt wird eine Bestellung via

```
GET /order/:id
```

wobei hier am Server auf Basis der Session überprüft wird, ob der Benutzer die entsprechende Berechtigung besitzt, um diese Bestellung abzurufen. Momentan können Benutzer nur ihre eigenen Bestellungen sehen, aber sollte die Implementierung des vorgestellten Administrations-Clients ebenfalls von diesem Web Service Gebrauch machen, müssten Benutzer mit mehr Berechtigungen eingeführt werden.

Optionen zur Bezahlung

Für den Prototypen wurden drei Möglichkeiten zur Bezahlung implementiert.

Benutzer können angeben, dass sie wie bisher mit Bargeld bezahlen wollen. Damit wird zwar, wie in Abschnitt 3.5 erläutert wurde, die Gefahr des Missbrauchs erhöht, was aber für den Prototypen vernachlässigt werden kann.

Als weitere Option wird die Kreditkarte angeboten. Dabei wird ebenfalls auf Sicherheit keine Rücksicht genommen, sondern rein auf die Simulation eines möglichst komfortablen Benutzererlebnisses gesetzt. Dazu können Benutzer ihre Kreditkarteninformationen im Client eingeben und zu der Bezahlung einer Bestellung übermitteln. Diese Daten werden anschließend am Server gespeichert und sind bei späteren Bestellungen unter der Angabe der vom Server vergebenen ID verwendbar.

Die letzte und ebenfalls praxisnahe Möglichkeit ist die Bezahlung mittels des österreichischen Services *paybox*, das in Abschnitt 3.3.4 vorgestellt wur-

de. Der Client übermittelt dazu die Mobilfunknummer, mit der abgerechnet werden soll. Dies wird beim Einsatz des Smartphone-Clients üblicherweise die eigene Nummer sein, die auf verschiedene Wege ermittelt oder alternativ manuell eingegeben werden kann. Das Web Service kommuniziert daraufhin mit dem Web Service von *paybox*, in dem Fall über eine *SOAP*-Schnittstelle, und übermittelt anschließend dem Client, ob die Abrechnung und somit die Bezahlung der Bestellung erfolgreich war.

Zu Demonstrationszwecken einer Kommunikation mit einem *SOAP* Web Services anhand einer *WSDL*-Datei nachfolgend die entwickelte *PHP*-Klasse, die zur Abrechnung mittels *paybox* dient.

```
1 class PayboxClient {
2
3     // Instanzvariable zur Kommunikation
4     private $SOAPClient = null;
5
6     // Konstruktor
7     function __construct() {
8
9         // Array mit Optionen zur Instanzierung
10        $options = array('location' => 'https://www.paybox.at/service/path',
11                        'login' => '12345', 'password' => '12345', 'uri' => 'http://www.
12                        service.com', 'trace' => '1');
13
14        // Angabe des Pfades zur WSDL-Datei des Web Services
15        $wsdl = './wsdl/paybox.wsdl';
16
17        // Instanzierung des Clients mit der PHP-Klasse SoapClient
18        $this->SOAPClient = new SoapClient($wsdl, $options);
19    }
20
21    public function pay($payer, $payee, $amount, $currency) {
22
23        // Aufruf folgender Methode, deren Existenz durch das WSDL-File
24        // bekannt ist
25        // Transaction payment(String language, boolean isTest, String payer
26        // , String payee, String caller, Integer amount, String currency,
27        // Integer paymentDays, String timestamp, String posId, String orderId,
28        // String text, String sessionId, com.rise.AdditionalProperty[]
29        // additionalProperties)
30        $response = $this->SOAPClient->payment(null, true, $payer, $payee,
31        null, $amount, $currency, null, null, null, null, null, null, null, null
32        );
33
34        // Rueckgabe der erhaltenen Antwort
35        return $response;
36    }
37 }
38 }
```

Diese Klasse wird folgendermaßen verwendet:

```
1 private function payOrder($order, $payment_type, $cc, $paybox_caller) {
2
3   // (...)
4
5   if ($order['payment_type'] == "paybox") {
6
7     $payboxClient = new PayboxClient();
8     $pbRes = $payboxClient->pay($paybox_caller, "+4300080098765",
9       floatval($order['sum_total']), "EUR");
10
11    if ($pbRes->error != "" && $pbRes->error != 0) {
12      return false;
13    }
14
15    $paybox['authorization'] = $pbRes->authorization;
16    $paybox['transaction_id'] = $pbRes->idTransaction;
17    $paybox['caller_nr'] = $paybox_caller;
18  }
19  // (...)
20 }
```

5.2.4 Datenmodell

Nachdem im Zuge des Designs der Schnittstellen die Ressourcen identifiziert wurden, wurde dadurch auch die wesentliche Struktur des Datenmodells vorgegeben. Da große Teile des Modells wenig Neuheitswert besitzen, wird im Anschluss nur auf Teilbereiche eingegangen, die einen größeren Bezug zum System haben. Es werden die Problemstellungen und der gewählte Lösungsweg beschrieben.

Öffnungszeiten

Die Öffnungszeiten der einzelnen Betriebe dürfen nicht nur in textueller Form gespeichert werden, sondern müssen auch verwendbar sein, um ermitteln zu können, ob ein Betrieb zu einer bestimmten Zeit geöffnet hat. Das ist notwendig, um z. B. eine Bestellung zum Dine-In nur für einem Zeitpunkt abgeben zu können, an dem der Betrieb geöffnet hat.

Weiters sind nicht nur die Öffnungszeiten des Betriebs relevant, sondern auch die Zeiten der Verfügbarkeit von Küche und Lieferservice, die sich von den Öffnungszeiten unterscheiden können.

Der Fall, dass manche Speisen nur zu bestimmten Tageszeiten verfügbar sind, wurde für den Prototypen außer Acht gelassen und ließe sich nach demselben Schema lösen.

Die Speicherung der Zeiten erfolgt mit folgender naheliegenden Tabelle.

```
CREATE TABLE `eat_place_hours` (
  `id` INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `id_place` INTEGER NOT NULL,
  `day` ENUM('mo', 'tu', 'we', 'th', 'fr', 'sa', 'su') NOT NULL,
  `type` ENUM('open', 'delivery', 'kitchen') NOT NULL,
  `start` TIME NOT NULL,
  `end` TIME NOT NULL,
  FOREIGN KEY (`id_place`) REFERENCES `eat_place`(`id`) ON DELETE
    CASCADE
);
```

Sollten Öffnungszeiten eine Tagesgrenze überschreiten, lässt sich das dadurch lösen, dass diese in zwei Datensätze aufgeteilt werden. Zum Beispiel wird eine Angabe wie „Samstag, 20:00 - 02:00 Uhr“ für die Speicherung zu „Samstag, 20:00 - 23:59:59 Uhr“ und „Sonntag, 00:00 - 02:00 Uhr“ umgewandelt.

Das wesentliche Problem stellen aber unregelmäßige Ausnahmen, wie z. B. Betriebsurlaube und Feiertage, dar. Prinzipiell könnten Feiertage wie ein zusätzlicher Wochentag behandelt werden, aber das wirft das Problem auf, dass manche Feiertage unterschiedlich behandelt werden, abhängig davon, ob sie auf einen Wochentag oder ein Wochenende fallen. Somit müssten zwei Feiertagstypen als Wochentag eingeführt werden. Das wirft aber wiederum das Problem auf, dass jeder Staat eigene Feiertage hat, die nicht alle regelmäßig auftreten und somit manuell verwaltet oder von einem externen Service abgefragt werden müssen. Auch behandelt nicht jeder Betrieb jeden Feiertag gleich. Beispielsweise haben etliche Betriebe an allen Feiertagen geöffnet, mit Ausnahme von Weihnachten und Silvester.

Auf Grund der zu erwartenden Komplexität durch die hohe Wahrscheinlichkeit von Unregelmäßigkeiten wurde entschieden, Feiertage und sämtliche anderen Änderungen als Sonderfälle der regelmäßigen Zeiten zu behandeln. Ein Sonderfall kann bedeuten, dass ein Betrieb zu einer bestimmten Zeit, z. B. einem Feiertag, entgegen der regelmäßigen Verfügbarkeit entweder geöffnet oder geschlossen hat.

```
CREATE TABLE `eat_place_hours_exception` (
  `id` INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `id_place` INTEGER NOT NULL,
  `type` ENUM('open', 'delivery', 'kitchen') NOT NULL,
  `start` DATETIME NOT NULL,
  `end` DATETIME NOT NULL,
  `available` BOOLEAN NOT NULL DEFAULT false,
  FOREIGN KEY (`id_place`) REFERENCES `eat_place`(`id`) ON DELETE
    CASCADE
);
```

Betreiber sollen über den zu entwickelnden Administrations-Client direkt Einträge vornehmen können. Denkbar ist auch eine automatisierte Eintragung durch definierbare Regeln, wie z. B., dass automatisch an jedem Feiertag geöffnet sein soll.

Reservierung: Tischlayout

Da in diesem Prototypen nicht mit Daten von einem POS-System gearbeitet wird, musste eine eigene Schnittstelle für Reservierungen entwickelt werden.

Anstatt für das Tischlayout eine komplette Grafik vom System zeichnen zu lassen, wurde entschieden, eine externe Grafik einzusetzen. Dies lässt auch größere Freiheiten für die Gestaltung und die damit erfolgende Individualisierung. Dieser Plan kann neben den Tischen zusätzliche Informationen, wie z. B. die Kennzeichnung von Raucher- und Nichtraucher-Bereichen, beinhalten.

Damit auf diesem Tische ausgewählt werden können, werden die klassischen Formen der Geometrie, das Rechteck und die Ellipse, unterstützt, um die Tische auf der Grafik zu markieren.

```
CREATE TABLE `eat_place_table_plan` (
  `id` INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `id_place` INTEGER NOT NULL,
  `id_table` INTEGER NOT NULL,
  `seats` INTEGER NOT NULL,
  `shape` ENUM('rect', 'oval') NOT NULL,
  `x` INTEGER NOT NULL,
  `y` INTEGER NOT NULL,
  `width` INTEGER NOT NULL,
  `height` INTEGER NOT NULL,
  FOREIGN KEY (`id_place`) REFERENCES `eat_place`(`id`) ON DELETE
    CASCADE
);
```

Der Client verwendet diese Daten, um die Grafik entsprechend zu annotieren.

5.3 Smartphone-Client

5.3.1 Konzeption

Der Prototyp wurde für das Betriebssystem *iOS* mit einem User Interface für das *iPhone* implementiert.

Die einzige Möglichkeit zur Erstellung einer nativen App besteht darin, die Entwicklung mittels des offiziellen *iOS*-Frameworks in der Programmiersprache *Objective-C* vorzunehmen. *Objective-C* ist Erweiterung der Programmiersprache *C* um das Konzept der Objektorientierung und wird haupt-

sächlich für Software, die für *iOS* oder *Apples* Desktop-Betriebssystem *Mac OS X* entwickelt wird, eingesetzt.

Der Aufbau der App folgt der in Abschnitt 3.2.2 beschriebenen Struktur. Die Daten werden von dem im vorigen Abschnitt vorgestellten Web Service bezogen.

Als Name des Prototypen wurde „*eatmeal*“ gewählt.

5.3.2 Architektur

Grundsätzlich sieht das *iOS*-Framework eine Implementierung nach dem *Model-View-Controller (MVC)* Schema vor [35]. Das Konzept wurde von *Trygve Reenskaug* im Jahr 1979 als logische Trennung des Datenmodells (*Model*), des User Interfaces (*View*) sowie der Programmlogik (*Controller*) entwickelt [16], deren erste Implementierung im Jahr 1987 von *Steve Burbeck* beschrieben wurde [2].

Model und *View* kommunizieren nie direkt, sondern nur über die Zwischenschicht des *Controllers*. Dadurch können diese Teile, sobald die Schnittstellen definiert wurden, unabhängig voneinander entwickelt und auch ausgetauscht werden. Das ermöglicht beispielsweise eine Auswechslung des User Interfaces, ohne einer Änderung an der Programmlogik oder am Datenmodell.

Model

Das Datenmodell der App besteht im Wesentlichen aus den Ressourcen, die auch das Web Service offenlegt.

Auf Persistierung der Daten am Client wurde verzichtet, da der Benutzer für alle wesentlichen Aktionen über eine Verbindung zum Internet verfügen muss und so die Daten immer aktuell vom Web Service abgerufen werden können.

Zum automatischen, schnellen Login werden Benutzername und Passwort in der *Keychain* abgelegt. Diese ist ein zentraler Mechanismus von *iOS*, der zur sicheren Speicherung von Daten dient.

Controller

Als *Controller* werden all jene Elemente verstanden, die der Verarbeitung der Daten dienen. Das sind z. B. die Schnittstelle zum Web Service und die Mechanismen zur Evaluierung und Verarbeitung der Benutzereingaben.

View

Das User Interface macht von Standard-Ansichten und -Konzepten Gebrauch, die ein Benutzer des Betriebssystems kennt und somit gewohnt ist. Dadurch wird die Einarbeitungszeit in die App möglichst kurz gehalten. Ein Beispiel

für eine Ansicht ist die Klasse `UITableView`, einer tabellarischen Ansicht von Daten mit Scrolling. Diese Elemente werden durch grafische Gestaltung individualisiert, aber darauf wird für den Prototypen kein Fokus gelegt.

Trennung von Controller und View

iOS trennt zwischen den Hauptklassen `UIViewController` und `UIView`. Letztere ist für die Darstellung von Elementen verantwortlich, und sämtliche UI-Elemente, wie z.B. `UILabel` und `UIImageView`, erben von ihr. Für die Verschachtelung der Views und deren Befüllung mit korrekten Daten dienen `UIViewController`. Ein Beispiel für ein derartiges Paar ist die zuvor genannte View `UITableView` mit dem Controller `UITableViewController`.

Wie zu sehen ist, sind `UIViewController` eng mit den eigentlichen Ansichten verknüpft und beinhalten diverse Logik zur Darstellung der Daten, was nach dem *MVC*-Schema noch dem „Model“ zuzuordnen ist. Diese `UIViewController` enthalten üblicherweise aber auch davon unabhängige Methoden, um Daten zu laden oder Eingaben zu validieren, deren eigentlicher Platz im „Controller“ wäre und somit eigentlich eine Klasse `UIViewControllerController` existieren müsste.

Um diese Trennung zu erzielen und damit auch die Möglichkeit zu schaffen, nachträglich ein zweites User Interface für das *iPad* einfach implementieren zu können, wurden diese Methoden in zwei Klassen getrennt. Zum Beispiel beinhaltet `PlaceListVC` die Controller-Methode

```
- (void)loadPlaces:(PlaceListType)listType;
```

sowie die zugehörigen Datenstrukturen. Von dieser Klasse erbt `PlaceListVC-iPhone`, welche Methoden enthält, um die geladenen Daten in einer `UITableView` anzuzeigen und somit ein `UITableViewController` ist. Sollte ein Interface für das *iPad* hinzugefügt werden wollen, müsste eine Klasse `PlaceListVC-iPad` entwickelt werden, die ebenfalls von `PlaceListVC` erbt und die Daten in einer für ein Tablet passenden Ansicht aufbereitet.

Alle entwickelten Klassen im Prototypen folgen diesem Schema.

Aufteilung der Aufgaben

Die Berechnung der Öffnungszeiten und der anwendbaren Bonusprogramme finden am Client statt. Dazu werden sämtliche notwendigen Angaben komplett über das Web Service übergeben. Das birgt den Vorteil, dass nicht bei jeder Aktion des Benutzers, die eine Neuberechnung erforderlich macht, eine Anfrage an das Web Service erfolgen muss. Einerseits entsteht dadurch keine Wartezeit für den Benutzer und andererseits hält das die Last des Web Services geringer. Ein Beispiel dafür ist die Aktualisierung der angewandten Bonusprogramme, sobald der Benutzer einen Artikel zum Warenkorb hinzu-

fügt bzw. dessen Anzahl ändert. Bei der schlussendlichen Übermittlung zum Web Service überprüft dieses die Berechnungen.

Check-In

Für den Prototypen wurde keine der in Abschnitt 3.4 vorgestellten Methoden zum Check-In implementiert oder simuliert. Erst wenn basierend auf den verschiedenen Kriterien wie Machbarkeit und Sicherheit eine für die Praxis denkbare Lösung ermittelt worden ist, macht es Sinn, diese durch einen Prototypen zu testen. In der aktuellen Version kann sich der Benutzer zur Simulation einfach durch einen Klick auf einen Button und der Eingabe seines Passworts bei einem Betrieb einchecken.

Für die Auslieferung von Benachrichtigungen über zur Abholung bereite Speisen werden zur Simulation am Gerät lokale Notifications verwendet und keine Push-Notifications. Das bedeutet, dass das Gerät selbst die Notifications ausliefert und diese nicht vom Web Service über einen Push-Mechanismus an das Gerät übermittelt werden. Für den Benutzer ist vom User Interface kein Unterschied zu erkennen, weswegen diese Art der Implementierung für den Prototypen völlig ausreichend ist.

Diese Notifications dürfen nicht mit den normalen Benachrichtigungen für einen Benutzer verwechselt werden. Letztere sind einfach eine über das Web Service abgefragte Liste mit Nachrichten, die z. B. darüber informieren, dass für eine Bestellung ein Review vergeben werden kann. Die Notifications hingegen sind ein zentraler Mechanismus des Betriebssystems und können auch empfangen werden, wenn die App nicht aktiv ist.

Externe Programmbibliotheken

Für bestimmte Aufgaben, die das *iOS*-Framework nicht direkt ermöglicht, kamen externe Libraries zum Einsatz.

Zur Kommunikation mit dem Web Service wurde *ASIHTTPRequest* verwendet [44]. Diese Bibliothek ist im Wesentlichen ein Wrapper des *CFNetwork* Frameworks, einem Teil des *iOS*-Frameworks, aber sie erleichtert die Kommunikation, wie z. B. durch simplen Upload von Dateien, wesentlich.

Zur Verarbeitung der Antworten im *JSON*-Format wurde die Bibliothek *JSONKit* gewählt [50]. Das *iOS*-Framework stellt für diese Aufgabe erst seit der aktuellen Version 5 die Klasse `NSJSONSerialization` zur Verfügung [33]. Da aber Abwärtskompatibilität zu *iOS* 4 gewünscht ist, wurde dieses externe Framework verwendet. Außerdem hat ein unabhängig durchgeführter Test ergeben, dass *JSONKit* im Durchschnitt rund 20% schneller arbeitet als `NSJSONSerialization` [38].

Die nachfolgende Funktion zeigt, wie eine klassische Kommunikation mit einem Web Service dadurch aussieht.

```

1 + (User *)getUser:(NSString *)userId {
2
3 // Erstellung der URL ala
4 // http://api.service.com/user/1?client_version=1.0
5 NSString *urlStr = [NSString stringWithFormat:@"%s/%s/%s", [
    Configuration getBaseUrl], userId, [Configuration
    getBaseUrlAdditionalParameters]];
6
7 // Start der GET-Anfrage mittels der ASIHTTPRequest Library
8 ASIHTTPRequest *request = [ASIHTTPRequest requestWithURL:[NSURL
    URLWithString:urlStr]];
9 [request setShouldContinueWhenAppEntersBackground:YES];
10 [request startSynchronous];
11
12 // Initialisierung des angefragten User-Objekts mit "null"
13 User *user = nil;
14
15 // Abfrage ob die Anfrage erfolgreich war
16 NSError *error = [request error];
17 if (!error) {
18
19 // Parsen der Antwort via JSONKit
20 NSDictionary *dictionary = [[request responseData]
    objectFromJSONDataWithParseOptions:JKParseOptionStrict error:&error
    ];
21
22 // Parsen erfolgreich und Key "success" auf "true"
23 if (!error && [[dictionary objectForKey:@"success"] boolValue]) {
24
25 // Verarbeitung der Daten in einer eigenen Methode
26 NSArray *userArr = [self parseDictionaryArrayAsUsers:[NSArray
    arrayWithObject:[dictionary objectForKey:@"user"]]];
27 if (![DataUtils isEmptyArray:userArr]) {
28     user = [userArr objectAtIndex:0];
29 }
30 }
31 }
32
33 // Retournierung des Objekts
34 return user;
35 }

```

Zur Abfrage, ob eine Verbindung zum Internet vorhanden sowie der Server des Web Services erreichbar ist, wurde die *Reachability* Klasse verwendet, die eigentlich ein Beispielcode von *Apple* ist [34]. Dies kürzt eine derartige Ermittlung auf folgendes Codestück:

```

1 Reachability *r = [Reachability reachabilityWithHostName:@"http://api.
    service.com"];
2 BOOL connectionAvailable = ([r currentReachabilityStatus] !=
    NotReachable);

```

Zum Einsatz des *Singleton* Patterns wurde die Klasse `FTSWAbstractSingleton` von *Shawn Erickson* eingesetzt [51]. Unter diesem Schema wird verstanden, dass in der ganzen App von einer gewissen Klasse maximal eine Instanz existieren kann.

Dieses Pattern wurde für die Benutzer-Verbindung zum Web Service verwendet. Die *Singleton* Klasse `UserConnection` ist für den Login zuständig und enthält anschließend das `User`-Objekt des eingeloggten Benutzers, das dadurch in der gesamten Anwendung zentral verfügbar ist.

```
1 // Login
2 BOOL loginSuccessful = [[UserConnection sharedInstance]
    loginWithUsername:self.nameTextField.text password:self.
    passwordTextField.text];
3
4 // Sukzessive Verwendung in jedem anderen Teil der App, z.B.
5 NSString *userId = [UserConnection sharedInstance].user.itemId;
```

Dieses Pattern wäre auch einsetzbar, wenn ein zentraler Cache implementiert werden soll, auf den alle Klassen Zugriff haben. Dieser könnte beispielsweise Bilder, die von mehreren Ansichten oft benutzt werden, im Speicher halten. Dabei muss darauf Acht gegeben werden, dass die Implementierung entweder Zugriff von mehreren Threads ermöglicht oder sicherstellt, dass nur von einem bestimmten Thread zugegriffen wird.

5.3.3 User Interface

Nachfolgend sind Screenshots des Prototypen in Originalgröße zu sehen, da diese am besten die Funktionsweise und die Abläufe im Client demonstrieren.



Abbildung 5.1: Icon



Abbildung 5.2: Startscreen



Abbildung 5.3: Start: Liste der umliegenden Betriebe. Rechts oben: Benachrichtigung (Abbildung 5.23)

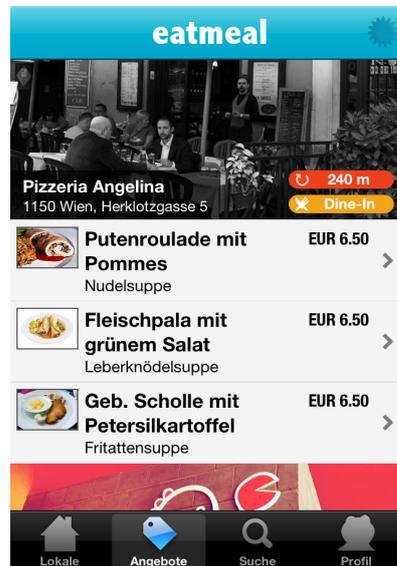


Abbildung 5.4: Start: Spezielle Angebote dieser Betriebe

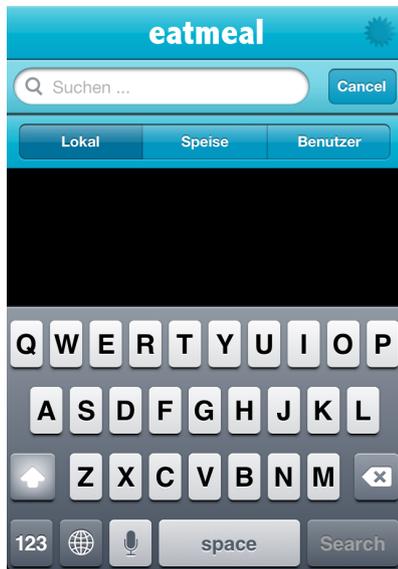


Abbildung 5.5: Suche



Abbildung 5.6: Benutzer-Profil



Abbildung 5.7: Betrieb: Stammdaten

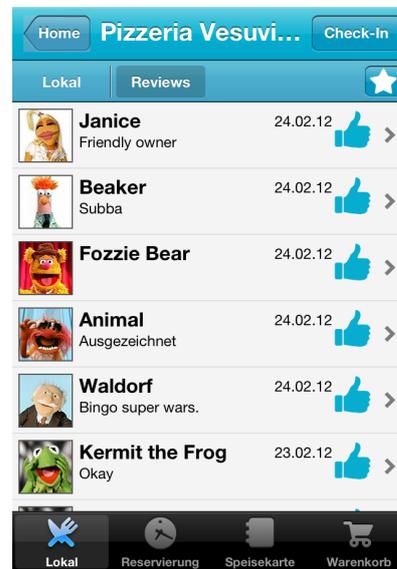


Abbildung 5.8: Betrieb: Reviews

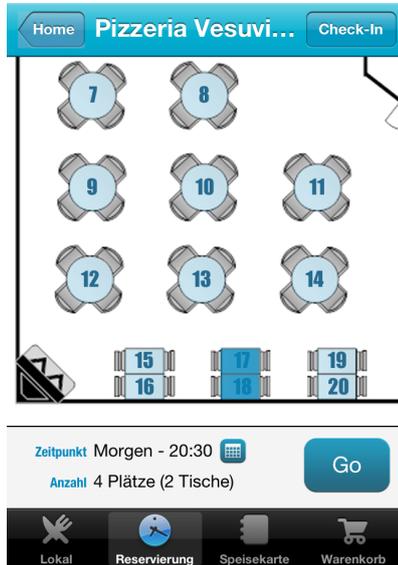


Abbildung 5.9: Tischlayout zur Reservierung



Abbildung 5.10: Speisekarte: Kategorien



Abbildung 5.11: Speisen mit markierten Favoriten

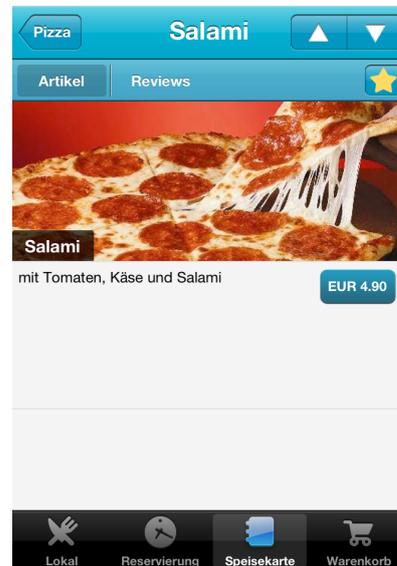


Abbildung 5.12: Detailansicht einer Speise



Abbildung 5.13: Hinzufügen zum Warenkorb mit spezieller Konfiguration



Abbildung 5.14: Warenkorb mit angewandtem Bonusprogramm



Abbildung 5.15: Bestellung zum Dine-In: Zeitpunkt

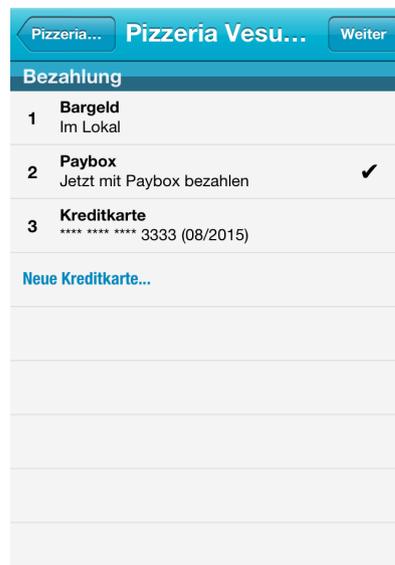


Abbildung 5.16: Bestellung zum Dine-In: Bezahlung



Abbildung 5.17: Bestellung zum Dine-In: Bestätigung



Abbildung 5.18: Check-In



Abbildung 5.19: Check-In: Warenkorb

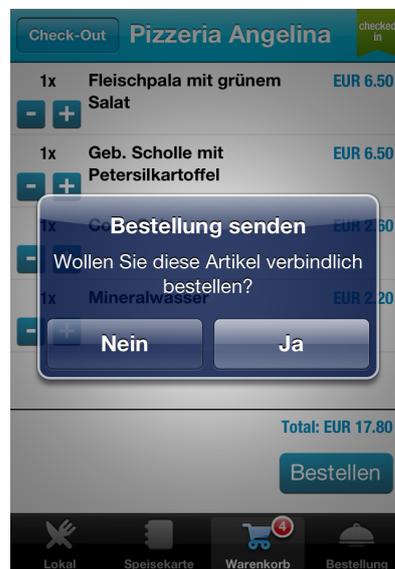


Abbildung 5.20: Check-In: Bestellung der Artikel des Warenkorbs

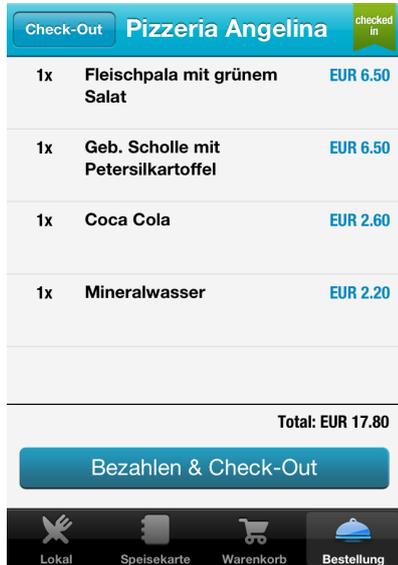


Abbildung 5.21: Check-In: Übersicht aller Artikel, die während dieses Aufenthalts bestellt worden sind



Abbildung 5.22: Empfang der systemweiten Notifications

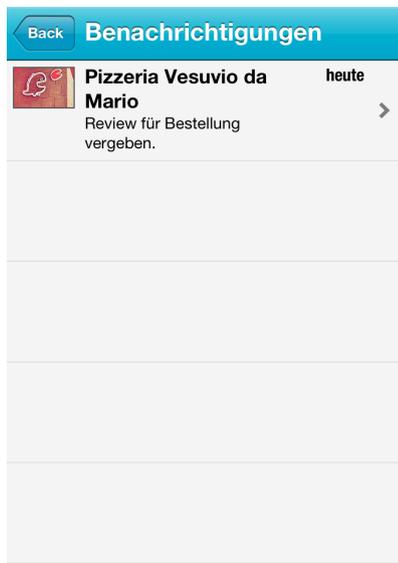


Abbildung 5.23: Benachrichtigungen



Abbildung 5.24: Abgabe Review

Kapitel 6

Evaluierung Prototyp

Anhand eines Tests durch Benutzer sollte einerseits der Prototyp auf Mängel getestet werden und andererseits im Rahmen einer anschließenden Befragung erneut das Interesse am Konzept evaluiert werden. Im Gegensatz zur zuvor durchgeführten Studie (Kapitel 4) waren die Befragten nun mit einem Prototypen des konzipierten Services vertraut und konnten daher konkretere Aussagen zur potentiellen Nutzung in einem realen Umfeld treffen.

6.1 Definition der Testmethode

Das *ISTQB* (*International Software Testing Qualifications Board*) definiert folgende vier Stufen von Tests: [64]

- Component Testing
- Integration Testing
- System Testing
- Acceptance Testing

Das Testen der einzelnen Komponenten („Component Testing“) sowie deren Zusammenspiel („Integration Testing“) fand bereits im Rahmen der Entwicklung statt. Dies geschah auf undokumentierter Basis durch regelmäßige manuelle Testläufe.

Im Rahmen eines Systemtests wird festgestellt, ob die Software den vor der Entwicklung definierten Anforderungen gerecht wird. Da es sich in diesem Fall um einen Prototypen handelt, dessen Anforderungen nie exakt spezifiziert worden sind, ist ein solcher Test nicht erforderlich.

Über den Akzeptanztest schreibt das *ISTQB* Folgendes: [64]

„The goal in acceptance testing is to establish confidence in the system, parts of the system or specific non-functional characteristics of the system. Finding defects is not the main focus in acceptance testing. Acceptance testing may assess the system’s

readiness for deployment and use, although it is not necessarily the final level of testing.“

In diesem konkreten Fall ist ein Usability Test, eine Form des Akzeptanztests, zum Einsatz gekommen. Dieser Typ setzt seinen Fokus auf den Umgang des Benutzers mit der Software und stellt fest, ob vorgegebene Aufgaben einfach bewältigt werden können. Im Mittelpunkt steht dabei klar das User Interface.

Jakob Nielsen, ein bekannter Usability Berater, gab in einer im Jahr 2000 veröffentlichten Studie an, dass nicht mehr als fünf Benutzer benötigt werden, um bei einem Usability Test die besten Ergebnisse zu erzielen. Mit diesen sollen aber so viele kleine Tests wie möglich durchgeführt werden. [75]

Als Testpersonen sollten potentielle Benutzer des Systems, das sind Besitzer eines Smartphones, herangezogen werden. Die Durchführung sollte in einer Laborumgebung geschehen, da keine Kooperation mit einem Gastronomiebetrieb abgeschlossen wurde, die es ermöglicht hätte, vor Ort zu testen. Als Hardware sollte ein aktuelles *iPhone* zum Einsatz kommen.

Zu Beginn eines jeden Tests soll der Testperson eine schriftliche Liste mit Aufgaben übergeben werden, die der Benutzer mittels der Software durchführen soll. Die Beschreibungen sollen kurz und verständlich sein. Während der Abarbeitung der Liste sollte mit der Testperson nicht kommuniziert werden, damit die Aufgaben wie in einem realen Umfeld selbständig gelöst werden. Die Testperson soll darum gebeten werden, während der Abarbeitung der Aufgaben laut mitzudenken, damit ersichtlich wird, welcher Gedankenablauf einer Handlung vorausgeht. Während der Durchführung soll die Testperson vom Testleiter beobachtet werden, um festzustellen, welche Lösungswege gewählt werden und welche Probleme beim Umgang mit der Software auftreten.

In einem abschließenden Gespräch soll ein Fragebogen beantwortet werden. In diesem Fall sollen nicht nur Fragen zur Benutzbarkeit des Prototypen gestellt werden, sondern vor allem welche zur potentiellen Nutzung eines derartigen Systems.

6.2 Durchführung

Es wurde festgelegt, dass die Gesamtlaufzeit eines Testlaufs inkl. Beantwortung des Fragebogens 30 Minuten nicht überschreiten soll.

Die ersten vier Aufgaben beschäftigten sich mit der Durchführung von Vorgängen wie z. B. einer Tischreservierung, einem Check-In und der Bestellung von Speisen. In der abschließenden fünften Aufgabe mussten voneinander unabhängige Fragen beantwortet werden. Die dazu notwendigen Informationen waren mittels der Software zu ermitteln.

Der Fragebogen gliederte sich in zwei Teilbereiche. Der erste Abschnitt beschäftigte sich mit der Handhabung des Systems und der zweite Teil mit dem Interesse am Konzept. Der Großteil der Antworten des Fragebogens

wurde in Form eines Interviews erhoben, um detaillierte Informationen und Meinungen zu erhalten.

Die Aufgabenliste sowie der Fragebogen sind vollständig in Anhang B zu finden.

Zur Optimierung der Tests wurde ein Pilottest durchgeführt. Im Anschluss an diesen wurden sowohl Adaptierungen am Prototyp sowie an der Methodik des Usability Tests vorgenommen.

Der eigentliche Test fand Ende Februar 2012 mit fünf Personen in einer Laborumgebung statt. Im Raum befanden sich jeweils nur der Testleiter sowie die Testperson, damit die anderen Testpersonen ihren Testlauf völlig unbeeinflusst durchführen konnten. Keine der Testpersonen hatte den Prototypen vor dem Testlauf gesehen. Drei Personen waren männlich, zwei weiblich und das Durchschnittsalter betrug 24,8 Jahre. Als Hardware diente ein *iPhone 4S*. Alle Personen waren auch privat Benutzer eines *iphones* und somit mit der Plattform gut vertraut.

6.3 Ergebnisse

Die Ergebnisse werden in zwei Abschnitten präsentiert. Der erste behandelt den Prototypen und der zweite die Meinung der Testpersonen zum Konzept.

6.3.1 Prototyp

Prinzipiell konnte jeder Teilnehmer alle Aufgaben selbstständig erfolgreich lösen. Nur einmal wurde ein Hinweis gegeben, und einmal wurde eine der Fragen der abschließenden Aufgabe falsch beantwortet. Auf diese sowie auf weitere Probleme mit dem User Interface wird später in diesem Abschnitt eingegangen.

Die Dauer der Lösung der einzelnen Aufgaben wurde zwar gemessen, aber sie wird nicht als Benchmark herangezogen, da Faktoren, wie das laute Mitdenken und die Überprüfung der Eingaben durch die Aufgabenstellung, sie zu sehr beeinflusst haben. Im Allgemeinen kann aber gesagt werden, dass kein Teilnehmer zur Bewältigung einer Aufgabe ungewöhnlich lange gebraucht hat.

Im Rahmen der anschließenden Befragung lobten Testpersonen das User Interface allgemein bzw. spezielle Bereiche wie die Reservierung oder die Startseite. Die App wurde für geübte *iPhone*-Besitzer als „leicht verständlich“ bezeichnet, da sie sich an allgemein üblichen Standards orientiert. Zwei Personen gefiel die Identität bzw. die farbliche Gestaltung der Anwendung. Ebenfalls zwei Personen fanden es hilfreich, dass Fotos zu allen Speisen vorgesehen sind und damit direkt ersichtlich ist, welche Erwartung an die bestellten Artikel gestellt werden kann. Die Möglichkeit der Abgabe eines Reviews mit Fotos sprach drei Personen besonders an.



Abbildung 6.1: Start: Problem mit den Markierungen rechts



Abbildung 6.2: Detailansicht eines Gastronomiebetriebs mit denselben Markierungen

Auf die Frage, ob eine Funktion vermisst wurde, haben zwei Personen geantwortet, dass sie gerne Trinkgeld gegeben hätten. Dieses Feature war im Konzept bereits vorgesehen, aber es wurde im Prototypen bewusst nicht implementiert. Weiters wurde die Funktionalität vermisst, die Liste der verfügbaren Gastronomiebetriebe nach Art der angebotenen Speisen und der Preiskategorie zu filtern.

Sämtliche sonstigen Kritiken bezogen sich direkt auf das User Interface, auf die nun eingegangen wird. Es wird jeweils beschrieben, wie diese Kritiken im Anschluss Einfluss auf den Prototypen hatten bzw. welche Lösungen zur Verhinderung des Problems möglich wären.

Startbildschirm

Wie in Abbildung 6.1 zu sehen ist, wurden die Gastronomiebetriebe auf der Startübersicht durch große Bilder repräsentiert. Diese Darstellung wurde, wie zuvor erwähnt, von einem Teilnehmer gelobt, ein anderer empfand diese jedoch als überladen.

Ein wesentliches Problem hatten aber alle Teilnehmer mit den Markierungen auf der rechten Seite, die anzeigen, ob ein Gastronomiebetrieb Bestellungen zum Dine-In, zur Lieferung oder zur Abholung anbietet. Diese Ansicht wurde auch in der Detailansicht eines Betriebs nochmals verwendet (Abbildung 6.2). Auf der Startübersicht wird zusätzlich die Entfernung zum ermittelten Standort des Benutzers als derartige Markierung angezeigt.



Abbildung 6.3: Alternative Version zur Startübersicht

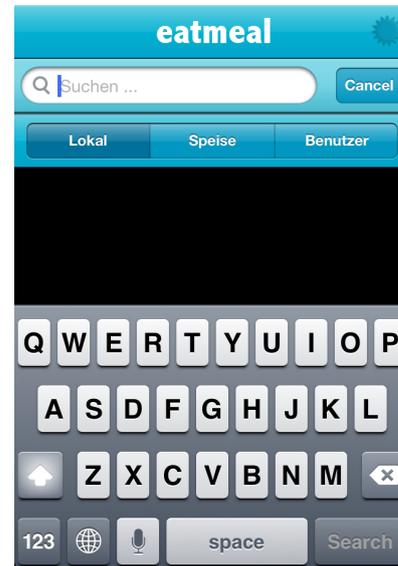


Abbildung 6.4: Suche mit Vorauswahl der Kategorie

Das erste Problem trat dadurch auf, dass Markierungen, die nicht zutrafen, trotzdem in grau angezeigt wurden. Einigen Teilnehmern war diese Farbcodierung nicht auf Anhieb klar.

Das zweite und wesentlichere Problem war, dass die Darstellung der Markierung von einigen Teilnehmern als Button missverstanden wurde. Zum Beispiel wurde versucht eine Bestellung zum Dine-In aufzugeben, indem in der Detailansicht des entsprechenden Betriebs wiederholt auf diese Markierung geklickt wurde.

Das erste Problem ist dadurch gelöst worden, dass die nicht zutreffenden Markierungen nicht mehr angezeigt werden. Zur Lösung des zweiten Problems müsste eine Darstellung gefunden werden, die zu weniger Missverständnissen führt. Im Rahmen der Entwicklung des Prototypen wurde jedoch auf dieses Experiment verzichtet.

Als Alternative zur Listenansicht mit den dominanten Bildern könnte ein Ansatz wie in Abbildung 6.3 gewählt werden.

Menüführung

Zwei Teilnehmer hatten Probleme mit der Menüführung. In diesem Zusammenhang wurde auch der einzige Hinweis bei der Durchführung der Tests gegeben.

Wie in Abbildung 6.1 und 6.2 zu sehen ist, wechselt der Inhalt des Menüs am unteren Bildschirmrand, wenn von der Startübersicht auf die Detailansicht eines Gastronomiebetriebs gewechselt wird.

Diese Praxis ist ungewöhnlich, da ein derartiges Menüelement in der gesamten App üblicherweise denselben Inhalt hat. In diesem Fall wurde jedoch durch die Fülle der Optionen diese Möglichkeit gewählt. Sämtliche anderen überlegten Optionen erschienen nicht praktikabel.

Die ursprüngliche Alternative wäre gewesen, kein Menü auf der Startübersicht zu haben und alle Elemente, die in dieser Version im Menü sind, anderweitig erreichbar zu machen. Einerseits war dies aber unübersichtlich und andererseits ist auch dieser Ansatz nicht üblich, da ein derartiges Menü in der Regel in der gesamten App zu sehen ist.

Eine weitere Alternative wäre, auf die Menüs komplett zu verzichten und sowohl für die Startübersicht als auch für die Detailansicht eines Gastronomiebetriebs eine weitere Übersichtsansicht einzuführen. Diese würde das Menü ersetzen und die einzelnen Features zugänglich machen. Diese Lösung würde aber für den Benutzer bedeuten, dass jeder Vorgang mehr Klicks benötigen würde, weswegen dieser Ansatz als nicht brauchbar eingestuft wurde.

Man könnte allerdings versuchen, die jetzige Lösung benutzerfreundlicher zu machen, indem die Menüs grafisch jeweils anders gestaltet werden und Verwechslungen so vorgebeugt wird. Beispielsweise könnte eines der zwei Menüs in einem Blau- statt dem Schwarz-Ton eingefärbt werden. Das vom *iOS*-Framework zur Verfügung gestellte Standard-Element für dieses Menü lässt ein anderes Aussehen zwar nicht zu, aber es könnte eine eigene Komponente entwickelt werden.

Es scheint hier keine optimale Lösung zu geben und jeder alternative Ansatz müsste einem weiteren Usability Test unterzogen werden. Bei den durchgeführten Tests war aber zu sehen, dass die Teilnehmer keine Probleme mit der jetzigen Menüführung hatten, sobald das Konzept einmal verstanden worden ist.

Suche

Bei der Eingabe des Suchbegriffs kann zwischen den Kategorien Gastronomiebetriebe, Speisen und Benutzer gewählt werden (Abbildung 6.4).

Eine Testperson hat das dafür eingesetzte Menüelement allerdings übersehen und deswegen die Frage, wie viele Betriebe eine bestimmte Speise führen, falsch beantwortet, da irrtümlich nach Betrieben anstatt Speisen gesucht wurde.

Ein Lösungsansatz wäre, keine Vorauswahl zu treffen und den Benutzer bei jeder Suche die Kategorie selbst wählen zu lassen. Sollte das vergessen werden, würde ein entsprechender Hinweis erscheinen.

Da dies für den Benutzer aber einen zusätzlichen Schritt bedeuten würde, ist ein anderer Weg zur Verhinderung dieses Problems gewählt worden. Beim Prototypen für den Usability Test wurde nach einer nicht erfolgreichen Suche die Meldung „Keine Ergebnisse“ ausgegeben. Da nun allerdings auch die Kategorie in dieser Meldung erwähnt wird, wie z. B. „Keine zutreffende

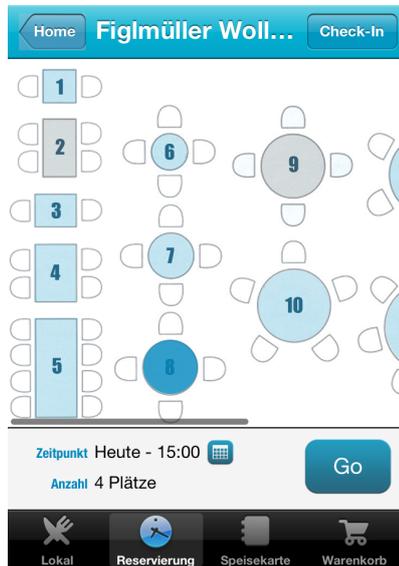


Abbildung 6.5: Reservierung



Abbildung 6.6: Datumswahl

Speise gefunden“, sollte der Benutzer darauf aufmerksam werden, dass in der falschen Kategorie gesucht wurde.

Reservierung

Im Zusammenhang mit der Ansicht zur Reservierung (Abbildung 6.5) traten verschiedene Probleme auf.

Eine Testperson empfand die farbliche Markierung bezüglich der Verfügbarkeit von Tischen als unklar und hätte sich eine Legende gewünscht. Verfügbare und somit durch den Benutzer selektierbare Tische waren hellblau markiert, bereits reservierte Tische hellgrau. Sobald der Benutzer einen Tisch ausgewählt hatte, erschien dieser dunkelblau. Eine Legende ist allerdings unüblich und auch diese Testperson konnte eine Reservierung durchführen, nachdem sie sich mit dem Interface vertraut gemacht hatte. Um es deutlicher zu machen, welche Tische verfügbar sind, könnten die bereits reservierten zusätzlich zur hellgrauen Darstellung mit einem roten X markiert werden.

Ein Problem kam allerdings für mehrere Personen zum Tragen. Bei dem Aufruf der Ansicht war bereits der nächstmögliche Zeitpunkt für eine Reservierung standardmässig ausgewählt. Dabei wurden auch die für diesen Zeitpunkt nicht mehr verfügbaren Tische hellgrau dargestellt. Mehrere Personen schenkten dem Datum anfangs keine Aufmerksamkeit und eine Person schickte die Reservierung sogar ab, ohne den Termin auf den gewünschten Zeitpunkt zu ändern. Dieses Missgeschick bemerkte sie aber rasch selbst.

Als Lösung wird der Termin nun nicht mehr vorausgewählt, und der Benutzer muss in jedem Fall die Auswahl selbst treffen. Dieser zusätzliche Schritt kann dadurch gerechtfertigt werden, dass Reservierungen typischerweise nicht für den nächstmöglichen, sondern für einen späteren Zeitpunkt erfolgen.

Weiters wurde die Anregung geäußert, dass nicht nur die Anzahl der gewählten Sitzplätze, sondern auch die Anzahl der gewählten Tische angezeigt werden soll, sofern mehr als ein Tisch selektiert wurde. Diese Anregung wurde umgesetzt (Abbildung 5.9).

Datumswahl

Abbildung 6.6 zeigt die in der Testversion eingesetzte Ansicht zur Datumswahl. Damit wurde sowohl für eine Reservierung als auch für eine Bestellung der gewünschte Zeitpunkt ausgewählt.

Mehrere Testpersonen wurden dadurch verwirrt, dass ein gewünschtes Datum nicht zur Auswahl stand und vermuteten anfangs einen Fehler im Programm. Allerdings resultierte das daher, dass der Betrieb an diesem Tag nicht geöffnet hatte. Zur besseren Orientierung wurde deswegen entschieden, die Öffnungszeiten des jeweiligen Betriebs im freien Raum über dem Element zur Wahl des Zeitpunkts anzuzeigen.

Weiters wurde die Anregung umgesetzt, die Wochentage bzw. „heute“ und „morgen“ bei den auswählbaren Daten anzuzeigen. Abbildung 5.15 zeigt die überarbeitete Ansicht.

Check-In: Bestellung

Anfänglich für Verwirrung sorgte bei allen Teilnehmern die Unterscheidung zwischen dem Warenkorb (Abbildung 5.19) und der Ansicht der gesamten Bestellung (Abbildung 5.21), sobald in einen Betrieb eingechekkt wurde.

Zuerst werden Artikel ausgewählt und in den Warenkorb gelegt. Danach werden diese verbindlich bestellt und wechseln vom Warenkorb in die Übersicht aller im Rahmen dieses Besuchs bestellten Artikel.

Dies war für die Testpersonen nicht sofort ersichtlich, und sie wunderten sich, wohin ihre Artikel nach dem Abschicken der Bestellung verschwunden waren. Da diese beiden Ansichten in ihrem Inhalt ähnlich sind, sind sie auch optisch nicht klar voneinander zu unterscheiden.

Dieses Problem wurde versucht dadurch zu beheben, indem nach dem Absenden der Bestellung nun automatisch von der Ansicht des Warenkorbs zu der Ansicht aller bestellten Artikel gewechselt wird.

6.3.2 Konzept

Da nur fünf Personen befragt wurden, kann keine direkte Gegenüberstellung mit den Zahlen der zuvor durchgeführten soziologischen Studie (Kapitel 4) erfolgen.

Die erste Frage zum Konzept beschäftigte sich damit, an welchen Features des Systems die Testpersonen im realen Einsatz Interesse hätten. Zur Entdeckung von neuen Gastronomiebetrieben gaben alle Befragten eine positive Tendenz an. Nur eine Person benutzt bereits andere Services und müsste das daher von der Qualität der Empfehlungen des Systems abhängig machen. Über spezielle Angebote, wie z. B. Mittagsmenüs, würden sich alle Befragten mithilfe eines derartigen Systems informieren.

Drei Personen würden Tische reservieren, die übrigen zwei reservieren allgemein nie in einem Gastronomiebetrieb.

Alle Befragten würden Speisen zur Lieferung bestellen, allerdings nur eine zur Abholung. Die anderen vier gaben an, generell kein Interesse an diesem Service zu haben.

Die Reaktionen zum Dine-In waren gemischt. Zwei Personen würden es ohne Vorbehalte verwenden und eine weitere hätte zwar Interesse, aber anfänglich Bedenken, ob das System auch tatsächlich zuverlässig ist. Die anderen zwei Personen haben kein Interesse an dem Konzept, wobei eine als Begründung anführte, dass die Dauer des Besuchs für sie selten eine Rolle spielt.

Auf die Frage, ob direkt im Restaurant mittels Check-In bestellt werden würde, waren sich die Personen unschlüssig. Jeweils eine Person meinte, das auf jeden bzw. auf keinen Fall verwenden zu wollen. Die anderen drei platzierten ihre Antwort in der Mitte der Skala, jedoch zweimal mit positiver Tendenz.

Zwei der Befragten würden auf jeden Fall damit im Restaurant bezahlen wollen und eine weitere Person vergab ebenfalls eine „2“ auf der Skala von „1“ (Ja) bis „6“ (Nein). Die anderen Antworten fielen mit „4“ bzw. „6“ auf die negative Seite.

Den Einsatz von automatisch angewandten Bonussystemen würden alle Teilnehmer begrüßen. Reviews vergeben würden vier der Befragten, wobei zwei explizit angegeben haben, dies nur tun zu wollen, wenn das Erlebnis besonders positiv oder besonders negativ gewesen sei. Die übrige Person hat allgemein kein Interesse an derartigen Features.

Als bevorzugten Weg der Bezahlung gaben zwei Personen *PayPal* und zwei Personen ein eigenes *E-Wallet* an. Eine Person würde dieses über *PayPal* aufladen wollen und die andere über eine Überweisung oder eine Gutschein-karte. Die übrige Person würde gerne direkt im System via Überweisung bezahlen wollen, da sie nicht im Besitz einer Kreditkarte ist.

Die Frage, ob ein Restaurant eher besucht werden würde, wenn es ein System nach diesem Konzept einsetzen würde, bejahten vier Personen. Die

übrige Person hat angegeben, dass sie das nur tun würde, wenn die Gesamtdauer bei dem konkreten Besuch ein wichtiger Faktor wäre. Alle meinten aber, dass der wichtigste Faktor die Art der angebotenen Speisen ist und ihre Antwort somit im Vergleich mit einem dahingehend gleichwertigen Restaurant zu verstehen ist.

Grundsätzlich kann somit gesagt werden, dass die Meinung der Testpersonen gegenüber einem derartigen System durchaus positiv ist, wobei natürlich nicht jede Person an jedem Feature Interesse hat.

Kapitel 7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein Konzept erstellt, das den Benutzer mittels eines Smartphones als Konsument die Geschäftsprozesse eines Gastronomiebetriebs wesentlich mitsteuern lässt.

Die damit angestrebte Effizienz hinsichtlich der Dauer des Aufenthalts in einem Betrieb konnte somit erzielt werden, da der Konsument nicht mehr von der Verfügbarkeit des Servicepersonals abhängig ist. Durch zusätzliche Features, wie z. B. Reviews, und der dadurch verbesserten Möglichkeit neue Betriebe zu entdecken, steigt der Mehrwert für den Benutzer weiter.

Das Interesse an einem praktischen Einsatz wurde mit zwei Studien festgestellt und fiel klar zu Gunsten des veranschlagten Konzepts aus. Auch der funktional umfangreiche Prototyp stieß auf positive Reaktionen bei den Testpersonen.

Der nächste Schritt nach der in dieser Arbeit erfolgten Entwicklung von Konzept und Prototyp wäre die weitere Entwicklung des Produkts mit dem Ziel der Markteinführung. Dazu sind aber etliche Entscheidungen notwendig.

Ein wesentlicher Punkt ist die Auswahl der Methodik zum Check-In. In Abschnitt 3.4 wurden dazu verschiedene Alternativen präsentiert. Zum Zeitpunkt der Entwicklung muss ermittelt werden, welche davon die beste Option ist bzw. ob neue technologische Entwicklungen weitere Alternativen geschaffen haben. Zum Beispiel würde eine Lösung, die eine Kommunikation mittels *NFC* einsetzt, eine gute Option darstellen. Dazu müsste *Apple* aber diese Technologie in ihre zukünftigen *iOS*-Geräte integrieren.

Die Schnittstelle zu den verschiedenen POS-Systemen ist ebenfalls ein zentraler Punkt, der für ein finales Produkt gelöst werden muss. Eine Evaluierung der eingesetzten POS-Systeme am Zielmarkt ist erforderlich, um festzustellen mit welchem Aufwand eine Entwicklung von Schnittstellen für all die verschiedenen Systeme verbunden ist bzw. ob dies überhaupt alle ermöglichen.

Als Alternative zur Implementierung einer Schnittstelle für alle gewünschten POS-Systeme kann überlegt werden, ein eigenes zu entwickeln. Da der Entwicklungsaufwand als hoch einzustufen ist, kann auch die Option in Betracht gezogen werden, eine Kooperation mit dem Hersteller eines bestehenden POS-Systems einzugehen. Beides hätte den Vorteil, dass die Kommunikation mit dem POS-System problemlos integriert werden kann. Dazu müssten die Gastronomen aber bereit sein, ihr bestehendes System auszuwechseln.

Als optimal wird sich ein Mittelweg erweisen. Einerseits sollte für populäre POS-Systeme, die es ermöglichen mit ihnen zu kommunizieren, eine Schnittstelle entwickelt werden. Andererseits sollte die Möglichkeit eines Gesamtpaketes aus dem Produkt samt POS-System angeboten werden.

Auch über die Möglichkeiten zur Bezahlung, die in Abschnitt 3.3 besprochen wurden, muss entschieden werden. Speziell im Zusammenhang mit der Angabe beim Personal bezahlen zu wollen, und den dadurch entstehenden Möglichkeiten des Missbrauchs, muss eine Lösung gefunden werden.

Darüber hinaus muss entschieden werden, mit welchem Umfang das Produkt veröffentlicht werden soll. Es wird von den finanziellen Mitteln, dem Team und dem Zeitplan abhängen, ob beispielsweise den Konsumenten zu Beginn nur die Smartphone Clients zur Verfügung gestellt werden, oder ob die ganze in Abschnitt 3.2 vorgestellte Produktpalette gemeinsam veröffentlicht wird. Speziell interessant wäre die Entwicklung eines robusten Tablets, das für Kunden ohne Smartphones als Alternative am Tisch platziert werden kann.

Parallel zur Produktentwicklung muss ein Vertriebsteam aufgebaut werden, das Gastronomiebetriebe als Kunden gewinnt. Im Idealfall sollte zu Beginn ein Partner gefunden werden, mit dem eng zusammengearbeitet wird und in dessen Betrieb das Produkt ausführlich getestet werden kann. So kann umfangreiche Erfahrung gewonnen werden, um bei einer Markteinführung möglichst viele Probleme bereits eliminiert zu haben.

Optimal wäre eine Markteinführung konzentriert auf eine Stadt oder einen Stadtteil, um dort eine hohe Konzentration von Betrieben zu erreichen. Dadurch steigt für den Benutzer der Mehrwert durch die Verwendung des Systems, da für viele Betriebe in seiner Umgebung dieser Service und somit auch Informationen über die einzelnen Betriebe verfügbar sind.

Das Potential dieses Konzepts wird als groß bewertet, da kein vergleichbares System am Markt existiert und die Reaktionen in den durchgeführten Studien positiv ausgefallen sind. Außerdem liegt das Geschäftsmodell auf der Hand, da pro Transaktion verrechnet werden kann und somit ab der ersten Bestellung, die über das System getätigt wird, Umsatz generiert wird.

Smartphones sind auf dem Weg, mehr und mehr Abläufe des täglichen Lebens zu beeinflussen. Entweder man ist Beobachter dieser Entwicklung oder man beteiligt sich an diesem Prozess und gestaltet die Zukunft aktiv mit. Mit dieser Arbeit wurde versucht, Letzteres zu tun.

Anhang A

Fragenkatalog der soziologischen Studie

Technologie-Einsatz

1. Welches Betriebssystem ist auf Ihrem Smartphone im Einsatz?

- *Android*
- *BlackBerry OS*
- *iOS (iPhone)*
- *Symbian (Nokia)*
- *webOS*
- *Windows Phone 7*
- Sonstiges

2. Sind Sie mit dem Betriebssystem zufrieden?

- Ja
- Nein, ich plane einen Wechsel auf:

3. Besitzen Sie ein Tablet?

- Ja, ein *iPad*
- Ja, und zwar:
- Nein, aber ich plane eine Anschaffung in den nächsten sechs Monaten
- Nein

4. Wenn ja: Tragen Sie es mit sich herum und haben dieses während eines Restaurantbesuchs typischerweise bei sich?
- Ja, zu jeder Zeit
 - Ja, aber nur zu Mittag
 - Ja, aber nur am Abend
 - Nein

Nutzung der Gastronomie

1. Wie versorgen Sie sich typischerweise zu Mittag an einem Wochentag in der Arbeit bzw. am Studienplatz? (Mehrfachnennung möglich)
- Kantine
 - Stamm-Restaurant
 - Verschiedene Restaurants
 - Lieferdienst
 - Supermarkt
 - Mitgebrachte Speisen von Zuhause
 - Frische Zubereitung von Speisen in einer Küche am Arbeitsplatz
 - Sonstiges:
2. Wie oft nehmen Sie pro Monat im Durchschnitt das Angebot folgender Gastronomiebetriebe in Anspruch?

	0 - 1 x	2 - 4 x	5 - 7 x	8 - 10 x	> 10 x
Restaurant (zu Mittag)					
Restaurant (am Abend)					
Kantine					
Fast food Kette					
Lieferdienst					
Kaffeehaus					
Club/Bar					

3. Welche Faktoren halten Sie öfters davon ab, das Angebot von Gastronomiebetrieben zu nutzen? (Mehrfachnennung möglich)
- Der Aufenthalt im Lokal kostet zu viel Zeit.
 - Die Lieferung dauert zu lange.
 - Der Mindestbetrag zur kostenlosen Lieferung wird nicht erreicht.
 - Es kostet gesamt zu viel Geld.
 - Das Lokal ist zu voll.
 - Online-Bestellung ist nicht möglich.
 - Sonstiges:
4. Was ist Ihre bevorzugte Zahlungsart in einem Gastronomiebetrieb?
- Bargeld
 - Bankeinzug
 - Quick
 - Kreditkarte
 - Sonstiges:

Auswahl des Lokals

1. Wie werden Sie i.d.R. auf ein Lokal aufmerksam? (Mehrfachnennung möglich)
- Werbung/Postwurfsendung
 - Empfehlung von Bekannten
 - Zufällige Entdeckung
 - Online:
 - Banner
 - E-Mail Werbung
 - Location-based Services (z. B. *Google Places*, *Qype*)
 - Gutschein Services (z. B. *DailyDeal*, *Groupon*)
 - Bestelldienste (z. B. *NetKellner.at*)
 - Anderen Dienst:
 - Sonstigen Weg:

2. Wie wichtig sind Ihnen folgende Faktoren bei der Auswahl eines Lokals?

	unwichtig	wenig wichtig	wichtig	sehr wichtig
Eindruck von außen				
Angebotene Speisen				
Preisklasse				
Spezielle Angebote wie Mittagmenüs				
Kritik von Bekannten				
Online-Reviews				
Fotos des Lokals				
Fotos der Speisen				
Bezahlungsmöglichkeiten				
Gastgarten vorhanden				

3. Reservieren Sie Sitzplätze in Gastronomiebetrieben?

- Ja, mehr als 6 Mal pro Monat
- Ja, 5 bis 6 Mal pro Monat
- Ja, 3 bis 4 Mal pro Monat
- Ja, 1 bis 2 Mal pro Monat
- Ja, aber selten
- Nein, niemals

4. Informieren Sie sich im Vorfeld über spezielle Angebote wie z. B. Mittagmenüs?

- Ja, über: (Mehrfachnennung möglich)
 - Aushänge des Betriebs
 - Website des Betriebs
 - Telefonanruf
 - Betrieb faxt das Angebot in das Büro des Arbeitsplatzes
 - Sonstiges:
- Ich würde mich gerne (mehr) informieren, kenne aber keinen effizienten Weg
- Nein, kein Interesse

5. Bestellen Sie Speisen zur Lieferung bzw. Abholung telefonisch?

- Ja, mehr als 6 Mal pro Monat
- Ja, 5 bis 6 Mal pro Monat

- Ja, 3 bis 4 Mal pro Monat
 - Ja, 1 bis 2 Mal pro Monat
 - Ja, aber selten
 - Nein, niemals
6. Nehmen Sie das Angebot von Online-Bestellsystemen für die Lieferung bzw. Abholung von Speisen in Anspruch?
- Ja, über: (Mehrfachnennung möglich)
 - *NetKellner.at*
 - *Lieferservice.at*
 - *willessen.at/mjam.net*
 - Sonstiges:
 - Nein; ich würde gerne, kenne aber keine Systeme
 - Nein; ich würde gerne, habe aber kein Vertrauen in mir bekannte Systeme
 - Nein, kein Interesse
7. Wenn ja, wie oft pro Monat?
- über 6 Mal
 - 5 bis 6 Mal
 - 3 bis 4 Mal
 - 1 bis 2 Mal
 - Seltener

Zukünftige Möglichkeiten

Skala von 1 (Nein, keinesfalls) bis 4 (Ja, unbedingt)

1. Würden Sie Platz-Reservierungen online bzw. über eine App durchführen wollen?
2. Würden Sie Essen im Vorhinein bestellen um es später im Betrieb sofort konsumieren zu können?

	1	2	3	4
Restaurant				
Kantine				
Fast food Kette				

3. Wenn ja, wären Sie bereit dieses auch schon bei der Bestellung über einen sicheren Weg zu bezahlen?
4. Würden Sie Bestellungen direkt im Betrieb über ihr Smartphone/Tablet abgeben wollen?

	1	2	3	4
Restaurant				
Kantine				
Fast food Kette				
Kaffeehaus				
Club/Bar				

5. Würden Sie die Rechnung von ihrem Smartphone/Tablet über ein sicheres System bezahlen wollen? Bei Aufenthalt im Lokal vom Sitzplatz aus, bei Lieferung direkt bei der Bestellung.

	1	2	3	4
Restaurant				
Kantine				
Fast food Kette				
Lieferdienst				
Kaffeehaus				
Club/Bar				

6. Würden Sie integrierte Bonussysteme, wie z. B., dass jede 10. Speise einer Art gratis ist, nutzen?

Zukünftige Möglichkeiten (II)

1. Würden Sie Zahlungsinformationen wie eine Kreditkarte vor Bezahlung im System hinterlegen?
 - Skala von 1 (Nein, keinesfalls) bis 4 (Ja, unbedingt)
2. Wenn nein, was hält sie davon ab?
 - Ich bezahle aus Prinzip nicht über Online-Systeme.
 - Ich vertraue nur Diensten, die ich bereits kenne, wie z. B. *PayPal*.
 - Ich müsste zuerst eine positive Erfahrung von Bekannten oder aus den Medien hören.
 - Sonstiges:

3. Würden Sie Trinkgeld geben, wenn Sie über ein mobiles System bezahlen?
 - Ja, denselben Betrag wie bei normaler Bestellung
 - Ja, sogar mehr
 - Ja, aber weniger
 - Nein

4. Würden Sie eine Bewertung, evtl. inklusive Bilder der Speisen, nach getätigtem Besuch mobil abgeben wollen?
 - Ja, immer
 - Ja, aber nur bei Zufriedenheit
 - Ja, aber nur bei Unzufriedenheit
 - Nein

5. Würden Sie Community-Features wie Freundschaftsanfragen, private Nachrichten etc. in einem Restaurant-System benützen?
 - Skala von 1 (Nein, keinesfalls) bis 4 (Ja, unbedingt)

Angaben zur Person

1. Geschlecht
 - männlich
 - weiblich

2. Welcher Altersgruppe gehören Sie an?
 - Unter 16
 - 16 - 18
 - 19 - 25
 - 26 - 35
 - 36 - 50
 - über 50

Anhang B

Usability Test

B.1 Aufgaben

Task 1	Beschreibung
Aufgabe	Tischreservierung
Vorbedingung	Benutzer ist bereits automatisch am System angemeldet.
Aufgabe Endzustand	Tischreservierung erfolgreich hinzugefügt.
Formulierung für die Testperson	Reservieren Sie im Restaurant „Figlmüller Wollzeile“ Plätze für min. 6 Personen für morgen Abend, beliebige Uhrzeit.

Task 2	Beschreibung
Aufgabe	Angebot
Vorbedingung	Benutzer ist bereits automatisch am System angemeldet.
Aufgabe Endzustand	Erfolgreiche Bestellung durchgeführt.
Formulierung für die Testperson	Suchen Sie ein Tages- bzw. Wochenangebot Ihrer Wahl in einem beliebigen Restaurant, und bestellen Sie zwei Stück zum Dine-In für morgen, 13:00 Uhr. Bezahlen Sie mittels <i>paybox</i> .

Task 3	Beschreibung
Aufgabe	Check-In
Vorbedingung	Benutzer ist bereits automatisch am System angemeldet.
Aufgabe Endzustand	Abgeschlossene Bestellung via Check-In.
Formulierung für die Testperson	<p data-bbox="655 656 1190 801">3.1 Wählen Sie das Lokal „<i>Pizzeria Vesuvio da Mario</i>“, und führen Sie einen Check-In durch. Das System geht nun davon aus, dass Sie sich im Betrieb aufhalten.</p> <p data-bbox="655 835 1190 1014">3.2 Bestellen Sie folgende Artikel: - Pizza „<i>Al Pollo</i>“ mit zwei Zusatz-Zutaten ihrer Wahl - „<i>Misto Grill</i>“ mit Pommes - Zwei Getränke Ihrer Wahl</p> <p data-bbox="655 1048 1190 1160">Sie erhalten nun innerhalb von 20 Sekunden eine Benachrichtigung, dass diese Artikel zur Abholung verfügbar sind.</p> <p data-bbox="655 1193 1190 1261">3.3 Bestellen Sie anschließend zwei Desserts Ihrer Wahl.</p> <p data-bbox="655 1305 1190 1487">3.4 Nach der erneuten Bestätigung, dass diese zur Abholung bereit sind, wollen Sie bezahlen. Führen Sie dazu den Check-Out durch, und bezahlen Sie die Bestellung via <i>paybox</i>.</p>

Task 4	Beschreibung
Aufgabe	Review zur Bestellung
Vorbedingung	Erfolgreiche Durchführung von Task 3
Aufgabe Endzu- stand	Neues Review für Bestellung von Task 3 vergeben.
Formulierung für die Testperson	<p>Bewerten Sie die soeben durchgeführte Bestellung. Sie haben dazu vom System eine Benachrichtigung erhalten, dass dies nun möglich ist.</p> <p>Fügen Sie zu mindestens einem Posten ein Foto hinzu, das Sie zum Zweck dieses Tests neu machen. Vergeben Sie Texte und Bewertungen frei und übermitteln Sie anschließend das Review.</p>
Task 5	Beschreibung
Aufgabe	Fragen
Vorbedingung	Benutzer ist bereits automatisch am System angemeldet.
Aufgabe Endzu- stand	Korrekte Beantwortung der Fragen
Formulierung für die Testperson	<p>5.1 Ist das Restaurant „<i>Pizzeria Vesuvio da Mario</i>“ im Allgemeinen gut bewertet?</p> <p>5.2 Bietet Restaurant „<i>Figlmüller Wollzeile</i>“ Artikel zur Lieferung an?</p> <p>5.3 Führt „<i>Pizzeria Angelina</i>“ Pizza Tonno?</p> <p>5.4 Wie viele verschiedene Restaurants führen Scholle?</p> <p>5.5 Bonusprogramm: Wie viele Pizzas müssen bei „<i>Pizzeria Vesuvio da Mario</i>“ noch bestellt werden, um eine Pizza kostenlos zu erhalten?</p>

B.2 Fragebogen

Zum Prototyp

Die Beantwortung der Fragen 1 bis 4 erfolgte jeweils mittels einer Skala von 1 („sehr große Zustimmung“) bis 6 („sehr geringe Zustimmung“) inkl. einem Feld für zusätzliche Anmerkungen.

1. Allgemein bin ich zufrieden mit der Benutzbarkeit des Systems:
2. Es war leicht die benötigten Informationen zu finden:
3. Die angebotenen Informationen waren einfach zu verstehen:
4. Das System hat alle Funktionen und Möglichkeiten geboten, die ich erwartet habe:
5. Nennen Sie bis zu drei Dinge, die Ihnen besonders gut gefallen haben:
6. Nennen Sie bis zu drei Dinge, die Ihnen überhaupt nicht gefallen haben:
7. Besondere Probleme, die bei der Benutzung der Plattform entstanden sind:

Zum Konzept

1. An der Benutzung folgender Features des Systems hätte ich Interesse:
Beantwortung mittels einer Skala von 1 („Ja, sehr“) bis 6 („Nein, keinesfalls“)
 - Entdeckung von neuen Restaurants
 - Information über Angebote (Mittagsmenüs etc.)
 - Reservierung von Plätzen
 - Bestellung zur Lieferung
 - Bestellung zur Abholung
 - Bestellung zum Dine-In
 - Bestellung im Restaurant (Check-In)
 - Bezahlung im Restaurant (Check-In)
 - Benutzung von Bonussystemen (z. B. jede 10. Pizza gratis)
 - Vergabe eines Review nach einer Bestellung

2. Wie würden Sie bevorzugt über ein derartiges System bezahlen wollen?

- Kreditkarte
- Pre-paid (E-Wallet)
- *paybox*
- Regulär beim Personal
- Sonstiges:

3. Würden Sie bei Verfügbarkeit eines derartigen Systems das Service eines Restaurants öfters in Anspruch nehmen?

- Ja
- Nein

Angaben zur Person

1. Geschlecht:

2. Alter:

3. Smartphone Modell:

Literaturverzeichnis

Literatur

- [1] David M. Brann und Beth C. Kulick. „Simulation of restaurant operations using the Restaurant Modeling Studio“. In: *Simulation Conference, 2002. Proceedings of the Winter*. Bd. 2. Dez. 2002, S. 1448–1453.
- [2] Steve Burbeck. *Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*. Englisch. Techn. Ber. 1987, 1992.
- [3] Ching-Su Chang, Che-Chen Kung und Tan-Hsu Tan. „Development and implementation of an e-restaurant for customer-centric service using WLAN and RFID technologies“. In: *Machine Learning and Cybernetics, 2008 International Conference on*. Bd. 6. Juli 2008, S. 3230–3235.
- [4] Soon Nyeen Cheong, Wei Wing Chiew und Wen Jiun Yap. „Design and development of Multi-touchable E-restaurant Management System“. In: *Science and Social Research (CSSR), 2010 International Conference on*. Dez. 2010, S. 680–685.
- [5] Sara A. Curin u. a. „Reducing service time at a busy fast food restaurant on campus“. In: *Simulation Conference, 2005 Proceedings of the Winter*. Dez. 2005, 8 pp.
- [6] Giuseppe Di Fabrizio u. a. „Have2eat: a Restaurant Finder with Review Summarization for Mobile Phones“. In: *Coling 2010: Demonstration Volume*. Beijing, Aug. 2010, S. 17–20.
- [7] Song-Ching Fan, An-Shin Shia und Jason C. Hung. „A study on the Restaurant industry implementing E-Commercialization-Taking Dragon Food Company as an example“. In: *Ubi-media Computing (U-Media), 2010 3rd IEEE International Conference on*. Juli 2010, S. 243–248.

- [8] Xinyang Feng, Jianjing Shen und Ying Fan. „REST: An alternative to RPC for Web services architecture“. In: *Future Information Networks, 2009. ICFIN 2009. First International Conference on*. Okt. 2009, S. 7–10.
- [9] Roy Thomas Fiedling. „Architectural Styles and the Design of Network-based Software Architectures“. Diss. Irvine: University of California, 2000. (Besucht am 15.01.2012).
- [10] Sun Guiling und QingQing Song. „Design of the Restaurant Self-Service Ordering System Based on ZigBee Technology“. In: *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*. Sep. 2010, S. 1–4.
- [11] Takeru Inoue u. a. „Key Roles of Session State: Not against REST Architectural Style“. In: *Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual*. Juli 2010, S. 171–178.
- [12] Tabacu Iulia-Maria, Horia Ciocarlie und Maria Ciocarlie. „Best practices in iPhone programming: Model-view-controller architecture; Carousel component development“. In: *EUROCON - International Conference on Computer as a Tool (EUROCON), 2011 IEEE*. Apr. 2011, S. 1–4.
- [13] Tzou Jyh-Hwa und L. Su Kuo. „The development of the restaurant service mobile robot with a Laser positioning system“. In: *Control Conference, 2008. CCC 2008. 27th Chinese*. Juli 2008, S. 662–666.
- [14] N.T. Kuo u. a. „The impact of service quality, customer satisfaction and loyalty in the restaurant industry: Moderating effect of perceived value“. In: *Quality and Reliability (ICQR), 2011 IEEE International Conference on*. Sep. 2011, S. 551–555.
- [15] Mishul Prasad, E. Scornavacca und H. Lehmann. „Using wireless personal digital assistants in a restaurant: impact and perceived benefits“. In: *Mobile Business, 2005. ICMB 2005. International Conference on*. Juli 2005, S. 69–74.
- [16] Trygve Reenskaug. *Thing-Model-View-Editor: an Example from a planning system*. Englisch. Techn. Ber. 1979.
- [17] Fernando Sanchez-Vilas u. a. „Applying Multicriteria Algorithms to Restaurant Recommendation“. In: *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*. Aug. 2011, S. 87–91.
- [18] Horng-Lin Shieh und Yi-Chun Liao. „RFID restaurant POS system“. In: *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*. Bd. 2. Juli 2011, S. 528–531.

- [19] Takeshi Shimmura, Takeshi Takenaka und Motoyuki Akamatsu. „Real-Time Process Management System in a Restaurant by Sharing Food Order Information“. In: *Soft Computing and Pattern Recognition, 2009. SOCPAR '09. International Conference of*. Dez. 2009, S. 703–706.
- [20] Tan-Hsu Tan, Ching-Su Chang und Yung-Fu Chen. „Developing an Intelligent e-Restaurant With a Menu Recommender for Customer-Centric Service“. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* PP.99 (2011), S. 1–13.
- [21] YongChai Tan u. a. „Automated Food Ordering System with Interactive User Interface approach“. In: *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*. Juni 2010, S. 482–485.
- [22] Liu Tao, Li Lu-Sheng und Ye Jun-Yiao. „Research and design on the handy terminal dishes-ordering system for catering industry“. In: *Business Management and Electronic Information (BMEI), 2011 International Conference on*. Bd. 2. Mai 2011, S. 226–228.
- [23] Hung-Wen Tung und Von-Wun Soo. „A personalized restaurant recommender agent for mobile e-service“. In: *e-Technology, e-Commerce and e-Service, 2004. EEE '04. 2004 IEEE International Conference on*. März 2004, S. 259–262.
- [24] Roy Want. „iPhone: Smarter Than the Average Phone“. In: *Pervasive Computing, IEEE* 9.3 (Juli 2010), S. 6–9.
- [25] Zhen-fang Wu und Yi-jie Li. „Research of Web Services based on GIS in the mobile catering service“. In: *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*. Juli 2010, S. 747–750.

Online-Quellen

- [26] A1. *Android Handys - Android Smartphones - Android Apps*. 2011. URL: <http://www.a1.net/handys-telefonie/android> (besucht am 01.12.2011).
- [27] All About Symbian: Blandford, R. *Ovi Store passes 9 million downloads a day*. Englisch. 2011. URL: http://www.allaboutsymbian.com/news/item/13193_Ovi_Store_passes_9_million_dow.php (besucht am 01.12.2011).
- [28] Android Developers Blog. *10 Billion Android Market Downloads and Counting*. Englisch. 2011. URL: <http://android-developers.blogspot.com/2011/12/10-billion-android-market-downloads-and.html> (besucht am 31.12.2011).

- [29] Android Handys. *Android Handys im Vergleich*. 2011. URL: <http://www.androidhandys.com/android-handy-vergleich/> (besucht am 01.12.2011).
- [30] Apple Inc. *A billion thanks. 25 times over*. Englisch. 2012. URL: <http://www.apple.com/itunes/25-billion-app-countdown/> (besucht am 10.03.2012).
- [31] Apple Inc. *App Store Review Guidelines*. Englisch. 2012. URL: <https://developer.apple.com/appstore/resources/approval/guidelines.html> (besucht am 15.01.2012).
- [32] Apple Inc. *iPhone 4S - Technical Information*. Englisch. 2011. URL: <http://www.apple.com/iphone/specs.html> (besucht am 01.12.2011).
- [33] Apple Inc. *NSJSONSerialization Class Reference*. Englisch. 2011. URL: http://developer.apple.com/library/ios/#documentation/Foundation/Reference/NSJSONSerialization_Class/Reference/Reference.html (besucht am 15.01.2012).
- [34] Apple Inc. *Reachability*. Englisch. 2010. URL: <http://developer.apple.com/library/ios/#samplecode/Reachability/Introduction/Intro.html> (besucht am 15.01.2012).
- [35] Apple Inc. *View Controller Programming Guide for iOS*. Englisch. 2012. URL: <http://developer.apple.com/library/ios/#featuredarticles/ViewControllerPGforiPhoneOS/Introduction/Introduction.html> (besucht am 15.01.2012).
- [36] Austrian Payments Council. *Die neue SEPA-Lastschrift*. 2011. URL: http://www.austrianpaymentscouncil.at/9796_DE.662CE41b3b28e660d0a57981b7cb655aebc01880 (besucht am 01.02.2012).
- [37] BiteHunter. *BiteHunter - Search and Dine. Real Time*. Englisch. 2011. URL: <http://www.bitehunter.com/> (besucht am 15.12.2011).
- [38] J. Bonto. *JSON Libraries for iOS Comparison*. Englisch. 2011. URL: <http://www.bonto.ch/blog/2011/12/08/json-libraries-for-ios-comparison-updated/> (besucht am 15.01.2012).
- [39] Business Insider: Rosoff, M. *This iPhone App Knows What You Like - Before You Ask It A Single Question*. Englisch. 2011. URL: <http://www.businessinsider.com/this-iphone-app-knows-what-you-like-before-you-ask-it-a-single-question-2011-7> (besucht am 15.12.2011).
- [40] Business Wire. *Amazon Web Services Achieves Level 1 Payment Card Industry (PCI) Compliance*. Englisch. 2010. URL: <http://www.businesswire.com/news/home/20101207005794/en/Amazon-Web-Services-Achieves-Level-1-Payment> (besucht am 01.02.2012).

- [41] Business Wire. *Starbucks Mobile Transactions Exceed 26 Million Within First Year*. Englisch. 2011. URL: <http://www.businesswire.com/news/home/20111206005582/en/Starbucks-Mobile-Transactions-Exceed-26-Million-Year> (besucht am 01.03.2012).
- [42] Christian. *McDonalds: Fast-Food Kette testet Computerterminals für Kunden*. 2010. URL: <http://www.vip-chicks.de/mcdonalds-fast-food-kette-testet-computerterminals-fuer-kunden-7216.html> (besucht am 01.08.2011).
- [43] Conceptic. *eMenu*. Englisch. 2011. URL: <http://www.emenu-international.com> (besucht am 01.03.2012).
- [44] B. Copsey. *ASIHTTPRequest documentation*. Englisch. 2011. URL: <http://allseeing-i.com/ASIHTTPRequest/> (besucht am 15.01.2012).
- [45] Daily Maverick. *BBX, BlackBerry's latest purported saviour*. Englisch. 2011. URL: <http://dailymaverick.co.za/article/2011-10-21-bbx-blackberrys-latest-purported-saviour> (besucht am 01.12.2011).
- [46] P. Demery. *Costs of data breach far overshadow cost of PCI compliance, study finds*. Englisch. 2008. URL: <http://www.internetretailer.com/2008/01/03/costs-of-data-breach-far-overshadow-cost-of-pci-compliance-stud> (besucht am 01.02.2012).
- [47] E la Carte. *Order Play Pay | E la Carte | Restaurant Tablets*. Englisch. 2012. URL: <http://www.elacarte.com> (besucht am 01.03.2012).
- [48] E la Carte. *Presto Effect*. Englisch. 2012. URL: <http://www.elacarte.com/presto> (besucht am 01.03.2012).
- [49] Element Payment Services. *PCI Compliance Levels - Determining Your PCI DSS Level*. Englisch. 2012. URL: <http://www.elementps.com/merchants/pci-dss/compliance-level/> (besucht am 01.02.2012).
- [50] J. Engelhart. *JSONKit*. Englisch. 2012. URL: <https://github.com/johnzang/JSONKit> (besucht am 15.01.2012).
- [51] S. Erickson. *FTSWAbstractSingleton*. Englisch. 2008. URL: <http://www.cocoadev.com/index.pl?FTSWAbstractSingleton> (besucht am 15.01.2012).
- [52] Flurry: Newark-French, C. *Mobile App Inventory Hungry Enough to Eat Internet Display Ad Spend*. Englisch. 2011. URL: <http://blog.flurry.com/bid/71285/Mobile-App-Inventory-Hungry-Enough-to-Eat-Internet-Display-Ad-Spend> (besucht am 01.12.2011).
- [53] Fresh Intermedia. *OrderingApps - Custom branded iPhone, iPad and Android mobile ordering applications for restaurants*. Englisch. 2011. URL: <http://www.orderingapps.com> (besucht am 15.12.2011).

- [54] Gartner. *Gartner Executive Programs Survey of CIOs in India Identifies Cloud Computing as the Top Technology Priority for CIOs in 2011*. Englisch. 2011. URL: <http://www.gartner.com/it/page.jsp?id=1574514> (besucht am 01.12.2011).
- [55] Gartner. *Gartner Says Apple Will Have a Free Run in Tablet Market Holiday Season as Competitors Continue to Lag*. Englisch. 2011. URL: <http://www.gartner.com/it/page.jsp?id=1800514> (besucht am 01.12.2011).
- [56] Gartner. *Gartner Says Worldwide Smartphone Sales Grew 16 Per Cent in Second Quarter of 2008*. Englisch. 2008. URL: <http://www.gartner.com/it/page.jsp?id=754112> (besucht am 01.12.2011).
- [57] Gartner. *Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth*. Englisch. 2012. URL: <http://www.gartner.com/it/page.jsp?id=1924314> (besucht am 01.03.2012).
- [58] Google. *Galaxy Nexus - Tech Specs*. Englisch. 2011. URL: <http://www.google.com/nexus/#/tech-specs> (besucht am 01.12.2011).
- [59] Google. *Google Wallet - make your phone your wallet*. Englisch. 2011. URL: <http://www.google.com/wallet> (besucht am 01.12.2011).
- [60] Google Think Insights. *The Mobile Movement*. Englisch. 2011. URL: http://www.gstatic.com/ads/research/en/2011_TheMobileMovement.pdf (besucht am 01.12.2011).
- [61] GrubHub. *Food Delivery | Restaurant Takeout | Order Food Online | GrubHub*. Englisch. 2011. URL: <http://www.grubhub.com> (besucht am 15.12.2011).
- [62] HP. *HP to Contribute webOS to Open Source*. Englisch. 2011. URL: <http://www.hp.com/hpinfo/newsroom/press/2011/111209xa.html> (besucht am 15.12.2011).
- [63] HTC Smartphones. *HTC Wildfire S - Technische Daten*. 2011. URL: <http://www.htc.com/de/smartphones/htc-wildfire-s/#specs> (besucht am 01.12.2011).
- [64] ISTQB (International Software Testing Qualifications Board). *Certified Tester - Foundation Level Syllabus*. Englisch. 2011. URL: http://istqb.org/download/attachments/5439583/ISTQB_Foundation+Level+Syllabus_2011.pdf?version=1&modificationDate=131712618000 (besucht am 01.02.2012).
- [65] Lieferheld. *Lieferservice | Essen bestellen | Pizza online - www.lieferheld.de*. 2012. URL: <http://www.lieferheld.de/> (besucht am 01.03.2012).
- [66] Lieferservice.at. *Lieferservice.at - Pizza bestellen - Essen bestellen*. 2012. URL: <http://www.lieferservice.at> (besucht am 01.03.2012).

- [67] I. Lunden. *Only 8.5% Of Internet Usage Comes Via Mobile*. Englisch. 2012. URL: <http://paidcontent.org/article/419-only-8.5-of-internet-usage-comes-via-mobiles-double-last-years-figure> (besucht am 15.02.2012).
- [68] McDonald's Österreich. *Facebook Status*. 2011. URL: <http://www.facebook.com/McDonaldsAustria/posts/309483362408224> (besucht am 15.12.2011).
- [69] MenuDrive. *Mobile Food Ordering System: Restaurant Marketing Features*. Englisch. 2011. URL: <http://www.menudrive.com/mobile-food-ordering.php> (besucht am 15.12.2011).
- [70] MenuPages. *MenuPages*. Englisch. 2011. URL: <http://www.menupages.com> (besucht am 15.12.2011).
- [71] Mjam. *Pizzaservice - Lieferservice - Essen bestellen | www.mjam.net*. 2012. URL: <http://www.mjam.net> (besucht am 01.03.2012).
- [72] Mobile Marketing Association Austria (MMA). *Mobile Communications Report 2011*. 2011. URL: http://austria.mmaglobal.com/sites/default/files/downloads/mma_communications_report_material_0.zip (besucht am 01.12.2011).
- [73] NetKellner.at. *Pizza, Pasta, Sushi Lieferservice - Speisen Online Bestellen - NetKellner.at*. Englisch. 2012. URL: <http://www.netkellner.at> (besucht am 01.08.2011).
- [74] Nielsen. *More US Consumers Choosing Smartphones as Apple Closes the Gap on Android*. Englisch. 2012. URL: <http://blog.nielsen.com/nielsenwire/consumer/more-us-consumers-choosing-smartphones-as-apple-closes-the-gap-on-android/> (besucht am 01.02.2012).
- [75] J. Nielsen. *Why You Only Need to Test with 5 Users*. Englisch. 2000. URL: <http://www.useit.com/alertbox/20000319.html> (besucht am 15.02.2012).
- [76] Nokia Developer News. *Nokia's Ovi Store tops 6 million daily downloads*. Englisch. 2011. URL: <http://www.developer.nokia.com/Community/Blogs/blog/nokia-developer-news/2011/06/23/nokia-s-ovi-store-tops-6-million-daily-downloads> (besucht am 01.12.2011).
- [77] NYTimes: Markoff, J. *The iPad in Your Hand: As Fast as a Supercomputer of Yore*. Englisch. 2011. URL: <http://bits.blogs.nytimes.com/2011/05/09/the-ipad-in-your-hand-as-fast-as-a-supercomputer-of-yore/> (besucht am 01.12.2011).
- [78] Official Google Blog: McClendon, B. *A new frontier for Google Maps: mapping the indoors*. Englisch. 2011. URL: <http://googleblog.blogspot.com/2011/11/new-frontier-for-google-maps-mapping.html> (besucht am 31.12.2011).

- [79] OpenTable. *About OpenTable*. Englisch. 2011. URL: <http://www.opentable.com/info/aboutus.aspx> (besucht am 30.11.2011).
- [80] OpenTable. *OpenTable - Österreich Reservations*. 2011. URL: <http://www.opentable.de/start.aspx?m=284> (besucht am 30.11.2011).
- [81] OpenTable. *OpenTable Mobile*. Englisch. 2011. URL: <http://www.opentable.com/mobile> (besucht am 30.11.2011).
- [82] Oxford Dictionaries Pro. *smartphone*. Englisch. 2011. URL: <http://oxforddictionaries.com/definition/smartphone> (besucht am 01.12.2011).
- [83] Oxford Dictionaries Pro. *tablet*. Englisch. 2011. URL: http://oxforddictionaries.com/definition/tablet?q=tablet+PC#tablet__6 (besucht am 01.12.2011).
- [84] Paybox. *paybox - Zahl's mit dem Handy*. 2011. URL: <https://www.paybox.at/> (besucht am 15.12.2011).
- [85] Paybox. *So bezahlen Sie mit paybox*. 2012. URL: <https://www.paybox.at/content/6192/Privat/Bezahlen/Wie-bezahlen> (besucht am 15.12.2011).
- [86] paysafecard.com. *Entdecke die Prepaid-Lösung für das Internet*. 2012. URL: <http://www.paysafecard.com> (besucht am 01.02.2012).
- [87] PCI DSS Compliance Blog. *Cost of PCI Compliance*. Englisch. 2009. URL: http://blog.elementps.com/element_payment_solutions/2009/02/pci-compliance-costs.html (besucht am 01.02.2012).
- [88] PCI Guru. *Merchant Beware – New Mobile Payment Solution Out in the Wild*. Englisch. 2011. URL: <http://pciguru.wordpress.com/2011/12/06/merchant-beware-new-mobile-payment-solution-out-in-the-wild/> (besucht am 01.02.2012).
- [89] PCI Security Standards Council. *A Perfect Fit: Understanding the Interrelationship of the PCI Standards*. Englisch. 2008. URL: https://www.pcisecuritystandards.org/pdfs/pci_ssc_0908_webinar_presentation.pdf (besucht am 01.02.2012).
- [90] PCI Security Standards Council. *Payment Card Industry Security Standards*. Englisch. 2010. URL: <https://www.pcisecuritystandards.org/documents/PCI%20SSC%20-%20Overview.pdf> (besucht am 01.02.2012).
- [91] PCI Security Standards Council. *PCI Security Standards Council update on PA-DSS and mobile payment acceptance applications*. Englisch. 2011. URL: https://www.pcisecuritystandards.org/documents/statement_110624_pcissc.pdf (besucht am 01.02.2012).

- [92] PCI Security Standards Council. *PCI Security Standards Council Updates Market on Mobile Position*. Englisch. 2011. URL: https://www.pcisecuritystandards.org/pdfs/pci_pr_110624.pdf (besucht am 01.02.2012).
- [93] PCMag.com: Yin, S. *Android Malware Surges Nearly Five-Fold Since July*. Englisch. 2011. URL: <http://www.pcmag.com/article2/0,2817,2396505,00.asp> (besucht am 01.12.2011).
- [94] Pressefach der hobex AG. *Kontaktlos Zahlen: Start in Österreich*. Englisch. 2011. URL: <http://www.pressefach.info/hobex/> (besucht am 01.12.2011).
- [95] Qype. *Qype - Entdecken. Empfehlen*. 2011. URL: <http://www.qype.com> (besucht am 15.12.2011).
- [96] Friedrich Schneider. *The Shadow Economy in Europe 2011*. Englisch. 2011. URL: <http://www.visaeurope.com/idoc.ashx?docid=4d53b726-cd71-4ba5-a50b-735d11ca4075&version=-1> (besucht am 01.02.2012).
- [97] Seamless. *Food Delivery | Order Food Online Restaurants and Take Out | Seamless*. Englisch. 2011. URL: <http://www.seamless.com/> (besucht am 15.12.2011).
- [98] Snapfinger. *About Snapfinger*. Englisch. 2011. URL: <http://snapfinger.com/Info> (besucht am 15.12.2011).
- [99] Starbucks Coffee Company. *Mobile Applications*. Englisch. 2011. URL: <http://www.starbucks.com/coffeehouse/mobile-apps> (besucht am 01.03.2012).
- [100] Strowman, S. *Best Practices in Survey Design*. Englisch. 2004. URL: http://my.simmons.edu/services/technology/ptrc/pdf/survey_design.pdf (besucht am 01.08.2011).
- [101] STUZZA. *eps Online-Überweisung*. 2012. URL: http://www.stuzza.at/1109_DE.61A32C51747b3904dfa3250139a5203ba6a340e0 (besucht am 01.02.2012).
- [102] SurveyMonkey. *Smart Survey Design*. Englisch. URL: <http://s3.amazonaws.com/SurveyMonkeyFiles/SmartSurvey.pdf> (besucht am 01.08.2011).
- [103] Tabbedout. *Tabbedout - Locations*. Englisch. 2011. URL: <http://www.tabbedout.com/locations> (besucht am 15.12.2011).
- [104] TechCrunch: Kincaid, J. *Ordr.in Raises Seed Funding From Google Ventures For Its Food API*. Englisch. 2011. URL: <http://techcrunch.com/2011/10/18/ordr-in-raises-seed-funding-from-google-ventures-for-its-food-api/> (besucht am 15.12.2011).

- [105] TechCrunch: Perez, S. *Google Acquires Clever Sense, Creator Of Local Recommendations App Alfred*. Englisch. 2011. URL: <http://techcrunch.com/2011/12/13/google-acquires-clever-sense-creator-of-local-recommendations-app-alfred/> (besucht am 13.12.2011).
- [106] TechCrunch: Perez, S. *You Can't Ignore The Web: OpenTable Launches New HTML5 Website*. 2011. URL: <http://techcrunch.com/2011/11/09/you-cant-ignore-the-web-opentable-launches-new-html5-website/> (besucht am 30.11.2011).
- [107] Telekom Deutschland. *Konzern-Zwischenbericht: 1. Januar bis 30. September 2011*. 2011. URL: http://www.download-telekom.de/dt/StaticPage/10/95/20/zwischenbericht_q3_2011.PDF_1095204.pdf (besucht am 01.12.2011).
- [108] The Official BlackBerry Blog. *BlackBerry App World: Great Stats, Awesome Apps!* Englisch. 2011. URL: <http://blogs.blackberry.com/2011/07/blackberry-app-world-success/> (besucht am 14.12.2011).
- [109] The Official Nokia Blog. *Open Letter from CEO Stephen Elop, Nokia and CEO Steve Ballmer, Microsoft*. Englisch. 2011. URL: <http://conversations.nokia.com/2011/02/11/open-letter-from-ceo-stephen-elop-nokia-and-ceo-steve-ballmer-microsoft/> (besucht am 01.12.2011).
- [110] Twitter: BlackBerry. *30 Nov 2010*. Englisch. 2010. URL: <https://twitter.com/#!/BlackBerry/status/9618916905062400> (besucht am 01.12.2011).
- [111] Twitter Developers. *REST API Resources*. Englisch. 2012. URL: <https://dev.twitter.com/docs/api> (besucht am 15.01.2012).
- [112] Vision Mobile. *Mobile Platforms: The Clash of Ecosystems*. Englisch. 2011. URL: http://www.visionmobile.com/rsc/researchreports/Vision-Mobile-Clash-of-Ecosystems_v1.pdf (besucht am 01.12.2011).
- [113] Windows Phone Applist. *Windows Phone Marketplace Statistics*. Englisch. 2011. URL: <http://www.windowsphoneapplist.com/en/stats/> (besucht am 31.12.2011).
- [114] WKO - Fachverband Hotellerie. *Kreditkarten-Gebühren und Zahlungssysteme*. 2009. URL: http://portal.wko.at/wk/dok_detail_file.wk?angid=1&docid=1071915&stid=486103&dstid=316 (besucht am 15.01.2012).
- [115] World Wide Web Consortium (W3C). *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. Englisch. 2007. URL: <http://www.w3.org/TR/soap12-part1/> (besucht am 15.01.2012).
- [116] World Wide Web Consortium (W3C). *Web Services Architecture: What is a Web service?* Englisch. 2004. URL: <http://www.w3.org/TR/ws-arch/#whatis> (besucht am 15.12.2011).

- [117] World Wide Web Consortium (W3C). *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. Englisch. 2007. URL: <http://www.w3.org/TR/wsdl20/> (besucht am 15.01.2012).
- [118] Yelp. *Yelp*. Englisch. 2011. URL: <http://www.yelp.com> (besucht am 15.12.2011).
- [119] ÖIAT (Österreichische Institut für angewandte Telekommunikation). *ÖIAT-Studie: Elektronische Zahlungsmittel viel besser als ihr Ruf*. 2009. URL: http://www.oiat.at/index.php?id=433&tx_ttnews%5Btt_news%5D=94&cHash=3e55b0ee265cf2bc8ecd91a27cd3f79c (besucht am 15.12.2011).
- [120] ÖNB - Österreichische Nationalbank. *Single Euro Payments Area (SEPA)*. 2012. URL: <http://www.oenb.at/de/zahlungsverkehr/Zahlungsverkehrsstrategie/sepa/sepa.jsp> (besucht am 01.02.2012).