Fachbereich Mathematik und Informatik
Institut für Informatik

Bachelorarbeit zur Erlangung des Grades
Bachelor of Science

# “iSightseeing – ein mobiler personen- und ortsbezogener Stadtführer”

Eingereicht bei Herrn Prof. Dr. Raúl Rojas

Zoran von der Heide (geb. Resanović)
Matr.-Nr. 2831465
zoran.r@fu-berlin.de

Abgabedatum: 03. Januar 2011

# Eigenständigkeitserklärung

Ich versichere, dass ich die Bachelorarbeit selbständig verfasst habe. Andere als die angegebenen Hilfsmittel und Quellen wurden nicht benutzt. Die Arbeit hat keiner anderen Prüfungsbehörde vorgelegen. Es ist mir bekannt, dass ich bei Verwendung von Inhalten aus dem Internet diese zu kennzeichnen habe und einen Ausdruck davon mit Datum sowie der Internet-Adresse (URL) als Anhang der Bachelorarbeit beizufügen habe.

Berlin, den 03.01.2011 Unterschrift

# Danksagung

Ich bedanke mich zunächst bei meinem Betreuer, Prof. Dr. Raúl Rojas, der mir das Thema der Bachelorarbeit bereitgestellt hat. Mein besonderer Dank richtet sich auch an meinen Kommilitonen und Betreuer Miao Wang für die tatkräftige Unterstützung und die vorgeschlagenen Verbesserungen.

Bedanken möchte ich mich im besonderen Maße bei dem Inhaber der Firma globalguides.net, Herrn Bernardus Steenbeek, für die Bereitstellung von Multimediadaten zur Einbindung in die prototypische Anwendung.

Des Weiteren möchte ich mich bei meiner Frau Livia für ihre Unterstützung und Geduld bedanken. Sie hat mir die Kraft und den Freiraum gegeben, die ich zum Anfertigen der Bachelorarbeit benötigt habe.

Zu guter Letzt geht mein Dank an meine Familie, die mich während des Studiums liebevoll unterstützt hat.

# Inhaltsverzeichnis


1 Einleitung ..... 1
1.1 Motivation für das Projekt – iSightseeing ..... 2
1.2 Zielsetzung der Arbeit ..... 4
1.3 Gliederung ..... 5
2 Technische und konzeptionelle Grundlagen ..... 7
2.1 Hardwareausstattung des iPhone 3GS und des iPhone 4 ..... 7
2.1.1 Überblick über die technischen Daten ..... 9
2.1.2 Multitouch-Display ..... 10
2.1.3 Zugang zum mobilen Internet ..... 11
2.1.4 Positionsbestimmung ..... 13
2.1.5 Multitasking mit iOS 4 ..... 15
2.2 Location Based City Guide ..... 16
3 Verwandte iPhone-Apps ..... 20
3.1 Begriffsdefinition ..... 20
3.1.1 Design ..... 20
3.1.2 Usability ..... 21
3.1.3 Content ..... 21
3.1.4 Personalization ..... 22
3.2 Ausgewählte Beispiele ..... 22
3.2.1 mTrip – Berlin-Reiseführer ..... 22
3.2.2 Navigaia – Berlin Multimedia Travel Guide ..... 27
3.2.3 City Walks – Berlin Map and Walking Tours ..... 32
3.3 Vergleichende Darstellung ..... 35
3.4 Kriterien für die Evaluierung ..... 36
3.4.1 Kriterienkatalog ..... 37
4 Anwendungsarchitektur ..... 39
4.1 Anforderungsanalyse ..... 39
4.1.1 Zielgruppe ..... 39
4.1.2 Funktionale Anforderungen ..... 41
4.1.3 Use Case ..... 41
4.2 Technical Design ..... 47
4.2.1 iOS-Architektur ..... 48
4.2.2 Cocoa Design Pattern ..... 49
4.2.3 Ausgewählte Aspekte der iOS-Architekturschichten ..... 51
4.3 Interface Design ..... 54
4.3.1 App-Icon – iSightseeing ..... 55
4.3.2 User Interface – iSightseeing ..... 56
4.4 Implementierung ..... 61
4.4.1 Datenmodell ..... 61

4.4.2 POI erstellen ..... 62
4.4.3 Tour erstellen ..... 63
4.4.4 Kartendarstellung mit Map Kit ..... 63
4.4.5 Local Notification ..... 63
4.4.6 Background Location ..... 63

5 Tests und Evaluation ..... 64
5.1 Testaufbau ..... 65
5.2 Qualitative Bewertung ..... 65

6 Fazit und Ausblick ..... 66
6.1 Offene Punkte ..... 66
6.2 Zukünftige Arbeiten ..... 67

7 Literaturverzeichnis ..... 68

8 Anhang ..... 73
8.1 Kriterienkatalog ..... 73
8.1.1 Analyse der mTrip-App ..... 73
8.1.2 Analyse der iSightseeing-App ..... 74
8.2 Diagramme ..... 75
8.3 Glossar ..... 76
8.4 Abbildungsverzeichnis ..... 78
8.5 Tabellenverzeichnis ..... 80

# 1 Einleitung

Die Einführung des iPhones von Apple im Jahr 2007 mit intuitivem Bedienungskonzept, hoher Datenübertragungsrate und gleichzeitiger Kopplung an günstige und transparente Datentarife hat die Mobilfunkbranche revolutioniert: Mobiles Internet wurde endlich für Endverbraucher intuitiv anwendbar und bezahlbar. Lästige Barrieren zum direkten Zugang ins Internet wie Portalseiten sind entfallen. Durch diese anwenderfreundliche Entwicklung dringt das Internet immer stärker in den Mobilfunkbereich ein. Aufgrund der flächendeckenden Verfügbarkeit drahtloser Netze via UMTS, EDGE oder 802.11 WLAN ändert sich der Schwerpunkt der mobilen Nutzung von reiner Telekommunikation hin zu multimedialen Anwendungen. Für den Boom der mobilen Internetnutzung per Smartphone sind verschiedene Faktoren ausschlaggebend. Dazu zählen die hohen Datenübertragungsraten, mobile Browser mit automatischer Skalierfunktion und das intuitive Bedienungskonzept über ein kapazitives Multi-Touch-Display. Dank der Ortungsmöglichkeiten der aktuellen Smartphones per integriertem GPS-Empfänger, Funkzelle oder WLAN wächst zudem die Interaktion des Mobilfunknutzers mit der räumlichen Umgebung durch Anwendung standortbezogener Dienste, sogenannter Location Based Services (LBS). Diese ermöglichen den Abruf ortsabhängiger Informationen oder Dienstleistungen.

Diese ständige Verfügbarkeit von positionsabhängigen Informationen macht die mobilen Endgeräte auch interessant für die Tourismusbranche, da das Internet sich zum wichtigsten Kommunikationsmedium entwickelt hat. Der Informationsbedarf kann darüber optimal bedient werden, da die Möglichkeit besteht, ortsbezogen sowohl Informationen beispielsweise über das Wetter oder Öffnungszeiten abzurufen als auch verschiedene Dienstleistungen in Anspruch zu nehmen, wie Taxiruf oder Reservierungen aller Art. Diese neuen Nutzungsmöglichkeiten der Smartphones beeinflussen auch den Konkurrenzkampf um die Präsentation touristischer Regionen, insbesondere von Städtereisen. Die Zahl der Anwender, die ihren ständigen „kleinen Begleiter“ als Ersatz für den klassischen Reise- oder Städteführer verwenden, wächst stetig an.

Im Verlauf der vorliegenden Bachelorarbeit wird speziell für das iPhone ein personen- und ortsbezogener Stadtführer für die Stadt Berlin konzipiert und entwickelt.

Die Entscheidung für die iOS-Plattform und somit gegen die Android-Plattform beruht maßgeblich auf der starken Fragmentierung des Android-Systems [29];[30].
Die Arbeit gliedert sich in einen praktischen und theoretischen Teil, das heißt in Entwicklung der nativen iPhone App „iSightseeing“ und einer dazugehörigen schriftlichen Analyse und Auswertung der Anwendung.
Die Grundlagen für die Motivation, Zielsetzung und Gliederung der Arbeit werden in den folgenden Unterpunkten gesondert dargestellt.

## 1.1 Motivation für das Projekt - iSightseeing

Das iPhone gilt aufgrund seiner intuitiven Bedienung als der populärste Vertreter unter den Smartphones. Die damit verbundene wachsende Nachfrage nach einer Möglichkeit zur Personalisierung des Smartphones durch individuelle Apps, sogenannte mobile Anwendungen, wurde durch die Einführung der Distributionsplattform App-Store bei gleichzeitiger Öffnung der iPhone-Plattform für Entwickler bedient. Diese Innovation der Software-Distribution ermöglichte einen rasanten Anstieg der Anzahl an Apps für iPhone OS[1] Geräte. Mehr als 250.000 Apps sind momentan laut Apple-Angaben über den App-Store verfügbar [23]. Laut einem Bericht von Netbiscuits wird das iPhone mit Abstand am häufigsten für den Besuch von mobilen Websites verwendet [25]. Auch in der Studie „Mobile Web Watch 2010“ wird bestätigt, dass die Häufigkeit der Nutzung des mobilen Internets durch das mobile Endgerät beeinflusst wird [22]. Die folgende Abbildung „Arten der Handynutzung“ wurde unverändert aus der eben genannten Studie übernommen. Sie dient hier zur Veranschaulichung der einzelnen Anwendungsgebiete von Mobilfunktelefonen (Telefonieren, SMS, MMS und mobiler Internetzugang). Der Vergleich erfolgt in Form eines Säulendiagrammes. Folgende Gruppen werden einander gegenübergestellt: Gesamt, iPhone, Android G1/G2, Smartphone mit Touchscreen, Smartphone ohne Touchscreen und Nicht-Smartphone. Aus der Darstellung geht hervor, daß die prozentuale Nutzung beim Telefonieren und beim Schreiben von SMS unabhängig ist von der Hardwareausstattung des mobilen Endgeräts. So nutzen circa 95 Prozent der Befragten das Handy zum Telefonieren. Auffällig beim Vergleich der Gruppen ist der gravierende Unterschied in der mobilen

[1] Bei der Einführung des iPads im Januar 2010 wurde das Betriebssystem in iOS umbenannt.

Internetnutzung. Trotz ähnlicher Ausstattung wie das iPhone wird das Android G1 von seinen Anwendern weit weniger für mobiles Surfen verwendet.

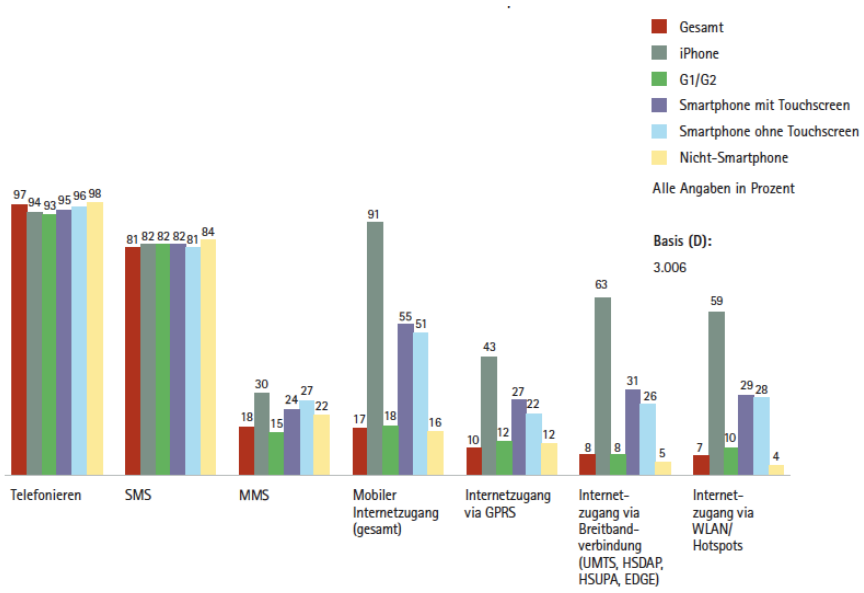


Abb. 1: Arten der Handynutzung

Aus dieser Studie geht weiter hervor, dass fast jeder fünfte Internetnutzer in Deutschland (circa 17 Prozent) mit seinem Mobiltelefon im Web surft [22, S. 4ff]. Es zeichnet sich zudem ein Trend ab, nach dem die Nutzung des mobilen Internets unterwegs, vor allem auf Reisen, steigt [22, S. 21].

Die Hardware-Ausstattung des Apple Smartphones begünstigt den Einsatz des mobilen Endgeräts als touristisches Informationssystem und Reiseführer. Dies macht es attraktiv für Anbieter von LBS. Zu den Vertretern von Apps für Lokalisierungsdienste gehören beispielsweise Gowalla und Foursquare [18]. Beide Anbieter betonen eher den spielerischen als den informativen Aspekt eines Städteführers [18]. Der Anwender kann durch die Nutzung der Ortungsdienste Freunde und interessante Orte oder Events in der Umgebung lokalisieren. Man kann durch den sogenannten „Check-In“, das heißt die öffentliche Freigabe der eigenen Position, Prämien sammeln und diese in Form von Vergünstigungen oder Vergütungen bei den teilnehmenden Werbepartnern einlösen [18]. Andere Anbieter folgen eher dem informativen Ansatz eines Stadtführers, wie die derzeit verfügbaren touristischen Audio Guides im App-Store. Sie bieten ortsabhängige Informationen und multimediale Inhalte zu Sehenswürdigkeiten der jeweiligen Stadt an. Der Nutzungsschwerpunkt dieser Anwendungen liegt somit auf dem Konsum der zur Verfügung gestellten Inhalte.

Für die Stadt Berlin sind über den App-Store verschiedene Stadtführer erhältlich. Im Anhang werden dazu Screenshots zur Verfügung gestellt, da die Anzahl der Apps keine Relevanz für die vorliegende Arbeit hat. Vielmehr liegt der Fokus der Betrachtung auf den Funktionsmerkmalen der Apps. Bei den derzeit zur Verfügung stehenden mobilen Guides ist ein manuelles Eingreifen zum Abspielen der Audio- und Videodateien erforderlich. Um dies zu veranschaulichen, werden beispielhaft Screenshots von zwei ausgewählten Anwendungen gesondert im Anhang präsentiert. Die Apps sind oft nicht intuitiv bedienbar, unübersichtlich gestaltet und wirken zudem häufig überladen. Die eben angeführten Mängel sowie die neuen technischen Möglichkeiten durch die Einführung von Multitasking [27] in iOS 4 sind Motivation genug, einen eigenen multimedialen Stadtführer für Berlin zu entwickeln. Die Anwendung iSightseeing richtet sich primär an ausländische Touristen, die Berlin besuchen, weswegen die Menüführung und die Informationen in der internationalen Sprache Englisch zur Verfügung gestellt werden.

## 1.2 Zielsetzung der Arbeit

Ziel der Arbeit ist die Realisierung einer intuitiv bedienbaren Anwendung, die dem mobilen Kontext [3] entspricht:

- kurze Interaktion des Anwenders mit dem iPhone (Microtasking)
- Abruf ortsabhängiger Informationen (Local)

Dafür müssen folgende Hauptkriterien erfüllt werden:

- Usability: verständliche Bedienerführung und leicht bedienbare Elemente
- Performance: schneller Abruf und Visualisierung von Informationen

Basierend auf den oben angeführten Kriterien, besteht die Intention der Arbeit darin, eine App zu realisieren, die automatisch auf Points of Interest (POI) hinweist, um dann ortsbezogene Informationen, bestehend aus Text und Mediendateien, bei Bedarf abzurufen oder zu visualisieren. Zu diesem Zweck sollen deshalb die neuen Multitasking Services, d.h. Background Location und Local Notifications, im Prototyp implementiert werden. Background Location ermöglicht eine kontinuierliche Positionsermittlung, während die App im Hintergrund läuft.

Neben den individuellen Touren bietet die App iSightseeing auch festgelegte Routen an, sogenannte Guided Tours. Diese sollen dem Anwender ungewöhnliche und erlebnisreiche Einblicke in die Geschichte und Gegenwart der Metropole liefern. Ziel ist eine multimediale Erlebnisführung durch die Hauptstadt im Taschenformat, anhand derer die Berliner Geschichte für den Benutzer erlebbar gemacht wird. Um dies zu realisieren, werden seltene historische Filmaufnahmen mit aktuellen Ansichten der Originalschauplätze bei ausgewählten Sehenswürdigkeiten kombiniert. Im Verlauf der Bachelorarbeit soll somit eine mögliche Lösung zur Konzeption eines mobilen personen- und ortsbezogenen Stadtführers für die iOS-Plattform inhaltlich, gestalterisch und programmiertechnisch erarbeitet werden.
Um die Unterschiede zwischen den vorhandenen Lösungen und dem Projekt sowie die mögliche Innovation der vorliegenden Arbeit zu verdeutlichen, ist es im Vorfeld notwendig, den aktuellen Stand der Technik zu beschreiben, bevor detailliert auf die Architektur des Systems eingegangen werden kann. Zum besseren Verständnis folgt nun eine grobe Übersicht über den Aufbau der Arbeit.

## 1.3 Gliederung

Die vorliegende Arbeit gliedert sich in sechs aufeinander aufbauende Kapitel. Nach dem einführenden Teil mit der Darstellung der Zielsetzung widmet sich das zweite Kapitel den Grundlagen der mobilen Endgeräte und der Location Based Services.
Auf Grund der unterschiedlichen technischen Ausstattung der verschiedenen iPod- und iPhone-Generationen werden dort nur ausgewählte Modelle kurz vorgestellt und miteinander verglichen. Des Weiteren wird in diesem Kapitel die Architektur eines Location Based City Guide definiert.
Im dritten Kapitel werden verwandte Lösungen für Stadtinformationssysteme für Apple Smartphones vorgestellt. Da für die Stadt Berlin zahlreiche Stadtführer existieren würde es den Rahmen der vorliegenden Arbeit sprengen, wenn alle näher thematisiert würden. Es erfolgt daher eine Reduzierung auf drei ausgewählte Beispiele, die aufgrund einer vorher festgelegten Bewertungsgrundlage analysiert und so anschließend miteinander verglichen werden können.
Das vierte Kapitel beschreibt den Lösungsansatz zur softwaregestützten Umsetzung der Anwendung. Dabei wird zunächst eine Anforderungsanalyse erstellt. Bevor

detailliert auf die Implementierung eingegangen werden kann, ist es jedoch erforderlich, Interface und Technical Design gesondert vorzustellen.
Die im Verlauf der Arbeit gewonnenen Ergebnisse werden im fünften Kapitel dargestellt und anschließend mit den vorhandenen Lösungen aufgrund des vorher definierten Kriterienkatalogs verglichen.
Am Ende der Arbeit werden im sechsten Kapitel die Ergebnisse kritisch reflektiert und mögliche Erweiterungen des Prototyps zur Diskussion gestellt. Abschließend erfolgt ein Ausblick auf zukünftige Arbeiten.

Nachdem damit ein kurzer Überblick zur Gliederung und zu inhaltlichen Schwerpunkten gegeben wurde, erfolgt eine kurze Erläuterung zu den Methoden der Arbeit.
Die vorhandene Fachliteratur zu iOS ist größtenteils veraltet, und da zum Zeitpunkt der schriftlichen Ausarbeitung der Arbeit nur wenige Fachbücher zum Thema iOS SDK 4 existieren, dient als Quelle für iOS 4-spezifische Funktionen ausschließlich die iOS Reference Library [28].
Grundlegende Kenntnisse über mobiles Internet und den Umgang mit Smartphones werden beim Leser der Arbeit vorausgesetzt, so dass keine Definitionen und Erläuterungen zum Basiswissen erfolgen. So werden innerhalb der vorliegenden Arbeit auch englische Fachbegriffe unreflektiert übernommen und nicht in die deutsche Sprache übersetzt, da dies internationale Fachtermini sind. Ebenso werden gängige Akronyme und Abkürzungen für Fachbegriffe verwendet, ohne dass diese direkt im Text erläutert werden. Sie werden gesondert im Glossar des Anhangs aufgeführt. Im Anhang finden sich ebenso die Verzeichnisse für die in der Arbeit verwendeten Tabellen und Abbildungen.

# 2 Technische und konzeptionelle Grundlagen

In diesem Kapitel werden zuerst die grundlegenden Eigenschaften des iPhones, insbesondere die des iPhone 3GS und des iPhone 4 mit Firmware 4.2, ohne Anspruch auf Vollständigkeit erläutert. Danach werden die konzeptionellen Grundlagen eines Location Based City Guide beschrieben.
Das iPad als ein Vertreter der Tablet-PC-Geräteklasse, wird nicht berücksichtigt, da dies den Rahmen der vorliegenden Arbeit sprengen würde.

## 2.1 Hardwareausstattung des iPhone 3GS und des iPhone 4

Das iPhone ist ein von Apple entwickeltes Smartphone, welches, wie bereits in der Einleitung erwähnt, 2007 auf den Markt gebracht wurde [26]. Die Innovation gegenüber den damals auf dem Markt befindlichen Smartphones bestand darin, dass es ein vergleichsweise großes Display mit hoher Auflösung besaß und es ausschließlich mit den Fingern per Multitouch-Gesten zu bedienen war. Die Integration eines modernen Web-Browsers ermöglichte es erstmals dem Mobilfunkbenutzer, das Internet auch unterwegs intuitiv und kostengünstig mit annähernder DSL-Geschwindigkeit zu benutzen. Mit diesem Designkonzept setzte Apple Standards für die Hardwareausstattung von Smartphones im Mobilfunkbereich. Bereits im Jahr 2008 sprang das iPhone OS auf den zweiten Platz des globalen Smartphone-Marktes [31]. Laut der Prognose des Analysten Gene Munster sollen allein im Jahr 2010 weltweit 36 Millionen iPhones verkauft werden [32]. Diese Prognose wurde für das Finanzjahr 2010 übertroffen, da weltweit bereits über 39 Millionen [26] iPhones verkauft worden sind.
Seit der ersten Markteinführung sind inzwischen vier Generationen von iPhones erschienen. Für die vorliegende Arbeit sind nur die beiden aktuellen Generationen, das iPhone 3GS und das iPhone 4, relevant. Gegenüber ihren Vorgängermodellen besitzen sie einen wesentlich größeren Arbeitsspeicher, welcher die Voraussetzung für Multitasking bildet. Die Unterstützung von Multitasking ist notwendig für die Funktionalität der App iSightseeing. Zum einen ist der Anwender nicht genötigt, permanent die Guide-Anwendung im Vordergrund zu halten und kann so parallel telefonieren oder Emails abrufen. Zum anderen kann auf die ressourcensparende Funkmastortung zurückgegriffen werden, die nur auf gravierende Positionsveränderungen reagiert und somit die Batterie weniger belastet als die GPS-Ortung.

Die beiden ausgewählten Modelle werden im Folgenden detailliert beschrieben. Zur besseren Übersicht erfolgt zuvor eine tabellarische Gegenüberstellung der beiden Modelle, um so Gemeinsamkeiten und Unterschiede herausarbeiten zu können. Die Vergleichstabelle iPhone 3GS/iPhone 4 wurde in Anlehnung an vorhandene Vergleiche von Apple [33], iPhone-Magazine.de und anderen Internet-Portalen eigenständig erstellt.

### 2.1.1 Überblick über die technischen Daten

| Modell | iPhone 3GS | iPhone 4 |
|---|---|---|
| | | |
| Arbeitsspeicher | 256MB RAM | 512MB RAM |
| interner Flash-Speicher | 8GB[2], 16GB/32GB | 16GB/32GB |
| Prozessor | ARMv7 Cortex-A8 600MHz | Apple A4 1GHz |
| Display | 3,5 Zoll Multi-Touch-Widescreen-Display<br>320 x 480 Auflösung (Portrait)<br>163 ppi<br>Multi-Touch | 3,5 Zoll hochauflösendes Retina IPS-Display<br>640 x 960 Auflösung (Portrait)<br>326 ppi<br>Multi-Touch |
| Videotelefonie | | FaceTime |
| Kamera | 3 Megapixel<br>Autofokus<br>Fokussieren per Fingertipp | 5 Megapixel<br>Autofokus<br>Fokussieren per Fingertipp<br>LED-Blitz<br>Frontkamera mit VGA Auflösung<br>Rückwärtig belichteter Sensor |
| Videoaufnahme | VGA-Auflösung, 30 Bilder/s<br>Videofokus per Fingertipp während der Aufnahme | HD-Auflösung (720p), 30 Bilder/s<br>Videofokus per Fingertipp während der Aufnahme<br>LED-Licht |
| Mobilfunk | GSM/EDGE, UMTS/HSDPA (7,2 Mbit/s) | GSM/EDGE, UMTS/HSDPA (7,2 Mbit/s)/HSUPA (5,8 Mbit/s) |
| WLAN | 802.11b/g | 802.11b/g/n |
| Ortung | Assisted GPS<br>Digitaler Kompass<br>WLAN<br>Cell ID | Assisted GPS<br>Digitaler Kompass<br>WLAN<br>Cell ID |
| Sensoren | Beschleunigungssensor<br>Annäherungssensor<br>Umgebungslichtsensor | Beschleunigungssensor<br>Annäherungssensor<br>Umgebungslichtsensor<br>3-Achsen-Gyrosensor |
| SIM-Standard | SIM | Micro SIM |
| Batterielaufzeit | Sprechdauer: 5 Stunden (3G);<br>5 Stunden surfen (WLAN: 9),<br>10 Stunden Videos,<br>30 Stunden Musikhören,<br>Stand-by: 300 Stunden | Sprechdauer: 7 Stunden (3G);<br>6 Stunden surfen (WLAN: 10),<br>10 Stunden Videos,<br>40 Stunden Musikhören,<br>Stand-by: 300 Stunden |
| Größe | 62,1 x 115,2 x 12,3 mm | 58,6 x 115,2 x 9,3 mm |
| Gewicht | 135g | 137g |

Tabelle 1: Tabellarischer Vergleich von iPhone 3GS und iPhone 4

[2] Seit der Einführung des iPhone 4 ist nur die 8GB-Version des iPhone 3GS erhältlich

### 2.1.2 Multitouch-Display

Wie aus der oben angeführten Tabelle zu entnehmen ist, unterscheiden sich die ausgewählten Modelle in der Ausstattung des Displays.

Das iPhone 3GS ist mit einem 3,5“ Multi-Touch-Widescreen-Display ausgestattet. Der kapazitive Bildschirm besitzt eine Abdeckung aus optischem Glas, auf der eine geringe Spannung aufliegt. Durch das Berühren mit den Fingern ändert sich das Potential auf der Oberfläche und die daraus resultierenden Spannungen, so dass die Touchposition auf dem berührungsempfindlichen Display bestimmt werden kann.

Die so gewonnene Information ermöglicht die Steuerung des Gerätes. Die Bedienung kann ausschließlich per Finger oder leitfähigem Eingabestift erfolgen. Bei Handschuhen oder Handprothesen versagt die Technik mangels fehlender Leitfähigkeit. Dies führte in Südkorea zu kuriosen Lösungen. Im Winter benutzte man Würstchen zur Steuerung des Smartphones, um so die Handschuhe anbehalten zu können. Neben der intuitiven Bedienbarkeit per Multi-Touch-Gesten[3] beeindruckt das Display durch eine hohe Auflösung von 480 x 320 px bei 163 ppi. Das Display unterstützt zwei Modi, so dass der Benutzer je nach Bedarf mit einer einfachen Drehung des Geräts um 90° Grad zwischen Portrait- und Landscapemodus wechseln kann. Einige Entwickler belegen den Landscapemodus mit Zusatzfunktionen, wie zum Beispiel die von Apple mitgelieferte App „Rechner“. Im Landscapemodus wechselt die App von einem einfachen Taschenrechner zu einem Taschenrechner mit erweitertem Layout für wissenschaftliche Funktionen.

Das Nachfolgemodell iPhone 4 unterscheidet sich in der Bedienung nicht vom Vorgänger. Der gravierende Unterschied besteht in der Leistungsfähigkeit des auf IPS-Technologie basierendes Retina-Displays. Trotz gleicher Größe des Displays besitzt das iPhone 4 mit 960 x 640 px bei 326 ppi eine vierfach höhere Auflösung. Die Darstellungsqualität von Texten entspricht der von gedruckten Büchern. Text und Grafiken erscheinen gestochen scharf, so dass man kaum noch einzelne Pixel wahrnehmen kann. Das Display eignet sich aufgrund seiner hohen Auflösung auch für die Wiedergabe von Videos in HD-Qualität.

Zur bildlichen Veranschaulichung des eben angeführten Vorteils des Retina-Displays gegenüber dem Vorgängermodell dient die folgende Abbildung.

[3] Zu den Touchscreen-Gesten zählen: Tap (einfaches Tippen), Double Tap (doppeltes Fingertippen), Drag (Ziehen), Flick (Schubsen), Swipe and Slide (Schieben) und Pinch (Strecken und Stauchen); [3, S. 11 ff]

Abb. 2: Vergleich der Auflösung beim iPhone 3GS und beim iPhone 4,
Bildquelle: Apple, URL: http://www.apple.com/de/iphone/features/retina-display.html

### 2.1.3 Zugang zum mobilen Internet

Die grundlegenden Kriterien für die Akzeptanz der mobilen Internetnutzung per Smartphone sind die Geschwindigkeit der Internetverbindung und die ebenso schnelle und korrekte Darstellung von Internetseiten. Um dies zu gewährleisten, müssen gängige Internetstandards vom mobilen Web-Browser unterstützt werden. Mit der mobilen Version von Safari gelang es Apple, diesen Anforderungen gerecht zu werden, indem eine automatische Zoomfunktion integriert wurde, so dass jede Website ohne vorherige Anpassung, – ausgenommen Flash-Inhalte –, betrachtet werden kann.

Die erste Generation von iPhones unterstützte nur den Mobilfunkstandard GSM mit der Erweiterung EDGE, was einer maximalen Datenübertragungsrate von 384 kbit/s entspricht. Erst ab dem iPhone 3G wurde die Internetverbindung durch die Unterstützung des Mobilfunkstandards der dritten Generation UMTS/HSDPA auf 3,6 Mbit/s beschleunigt. Dessen Nachfolger, das iPhone 3GS, verdoppelte die Übertragungsgeschwindigkeit auf 7,2 Mbit/s. Die Erhöhung der Datenübertragungsrate führte zu einer verstärkten Nutzung des Smartphones für den Abruf von Emails und den Besuch von Internetseiten. Dieser Trend wurde zusätzlich durch native Social Network Apps wie Facebook unterstützt und führte so zum Hype ums iPhone. Um dem wachsenden Bedarf der Anwender gerecht zu werden, Fotos

und Videos mobil aufnehmen und sofort auf den gängigen Social Network-Plattformen veröffentlichen zu können, wurde beim iPhone 4 die Unterstützung des Mobilfunkstandards HSUPA integriert. Dieser ermöglicht eine Upload-Geschwindigkeit von bis zu 5,76 Mbit/s.

Zusammenfassend kann festgehalten werden, dass die Fähigkeiten der Smartphones auch das Nutzungsverhalten des Anwenders beeinflussen, so dass Dienste, die einen klaren Mehrwert für unterwegs haben, zu den gefragtesten Internetanwendungen gehören. Dies wird durch die folgende Abbildung aus der Studie „Mobile Web Watch 2010“ bestätigt.

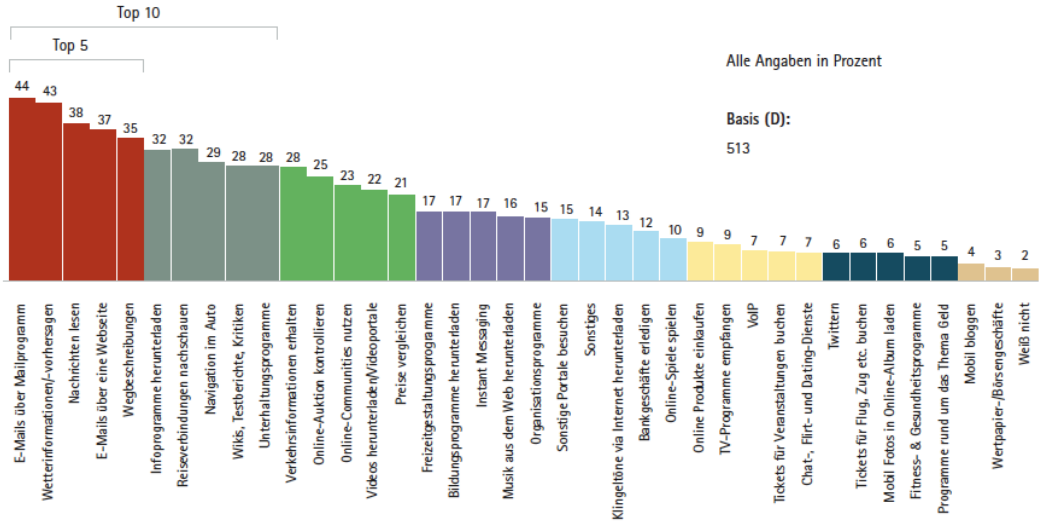


Abb. 3: Arten der mobilen Internetnutzung

Abbildung 3 zeigt die grafische Auswertung der Ergebnisse zu der Befragung „Arten der mobilen Internetnutzung“. Die Darstellung erfolgt in Form eines Säulendiagrammes mit absteigender Gewichtung, wobei jeweils fünf aufeinanderfolgende Punkte farblich zusammengefasst werden. Lediglich die letzte Gruppe besitzt nur drei Elemente. Die Top 5 sind hier rot gekennzeichnet und zeigen so, dass das mobile Internet am häufigsten zum Abrufen von Emails benutzt wird (zweimal unter den Top 5). Darauf folgen das Abrufen von Wetterinformationen, Nachrichten und Wegbeschreibungen.

Zusammenfassend kann hier fest gestellt werden, dass die Kommunikation per Mail und das Abrufen von Informationen den Schwerpunkt der mobilen Internetnutzung bilden.

### 2.1.4 Positionsbestimmung

Die Ortung bei Smartphones kann generell über drei verschiedene Möglichkeiten erfolgen. Diese sind GPS-, WLAN- und Mobilfunkortung. Die Voraussetzung dafür ist die Ausstattung des Smartphones mit GPS-Empfänger und WLAN-Modul. In Abhängigkeit von der Methode variiert die Präzision der Positionsermittlung. Um die Unterschiede und die damit verbundenen Vor- und Nachteile der drei Lokalisierungsverfahren zu verdeutlichen, folgt eine kurze Beschreibung der jeweils zugrundeliegenden Technik ohne Anspruch auf Vollständigkeit.

Die GPS-Ortung basiert auf den Empfang und der Auswertung von Mikrowellensignalen der 24 um die Erde kreisenden GPS-Satelliten. Für die dreidimensionale Positionsbestimmung per GPS ist die Distanzberechnung mittels Laufzeitmessung (Triangulation) zu mindestens drei Satelliten erforderlich, wie in der folgenden Abbildung 4 [35] dargestellt.

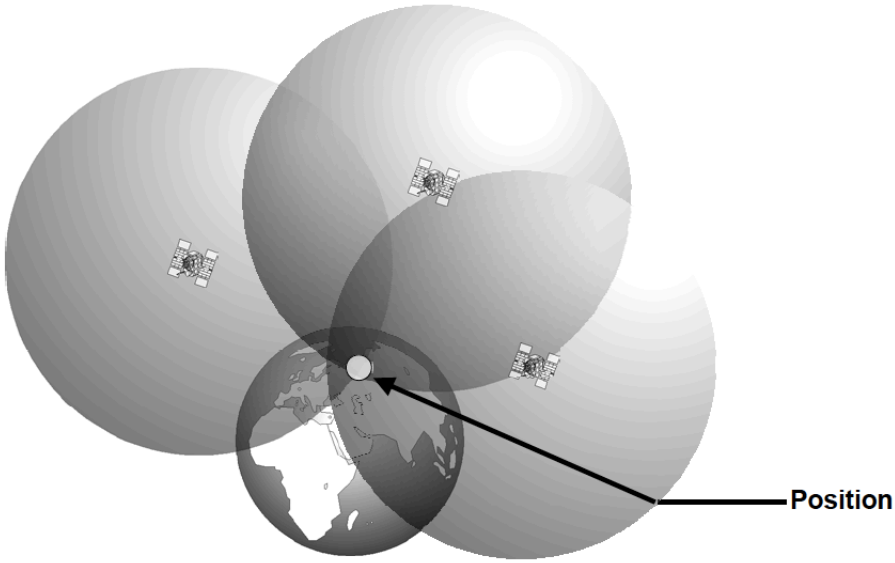


Abb. 4: GPS-Positionsbestimmung im Raum

Die Triangulation nutzt die geometrischen Eigenschaften von Dreiecken zur Positionsbestimmung und kann entweder über Angulation (Winkelmessung) oder Lateration (Distanzmessung) erfolgen [15, S. 17-19].

In der Praxis verwendet der GPS-Empfänger für die Ortung insgesamt vier Satelliten, da ein zusätzlicher Satellit für die Zeitsynchronisation benötigt wird. Ohne Zeitsynchronisation kommt es zu einer Abweichung bei der Positionsbestimmung.

Alle Positionsangaben beziehen sich auf das terrestrische Referenzsystem WGS 84[4]. Die GPS-gestützte Ortung gilt als präziseste Methode der drei Möglichkeiten zur Standortbestimmung. Diese Technik besitzt jedoch einen Nachteil: sie versagt in Gebäuden, da die hochenergetischen Radiosignale der GPS-Satelliten eine Sichtverbindung zum Empfänger benötigen und Wände schlecht durchdringen können. So ist die Dämpfung in Abhängigkeit zum Baumaterial unterschiedlich stark.

Für die Positionsbestimmung des mobilen Endgerätes per WLAN benötigt man vier Referenzpunkte (Access Points). Die Position wird anhand der Triangulation von WLAN-Netzwerken ermittelt. Dabei erfolgt die Entfernungsmessung anhand der Signalstärke der Access Points. Kennt das Smartphone den WLAN-Router, wird die ermittelte MAC-Adresse an eine weltweite Referenzdatenbank übertragen. Eine MAC-Adresse ist ebenso einzigartig wie ein Fingerabdruck, so dass anhand der MAC-Adresse Netzwerkgeräte eindeutig identifiziert werden können. Die Möglichkeit zur Manipulation der MAC-Adresse besteht zwar, wird jedoch hier im Text mangels Relevanz für die vorliegende Arbeit nicht näher erläutert. Die Datenbank enthält zu jeder ihr bekannten MAC-Adresse eine Positionsangabe, so dass die genaue Position des mobilen Endgerätes erfragt werden kann. Bis zum iPhone 3GS wurde dafür der Dienst Wi-Fi Positioning System vom Bostoner Startup SkyHook Wireless verwendet [11, S. 797 ff]. Ähnlich wie die Messautos von Google für Google Street View WLAN-Access Points kartographiert haben, hat dies SkyHook Jahre zuvor bereits unauffällig und völlig unbemerkt von der Öffentlichkeit getan [11, S. 798].

Da die WLAN-Ortung für das mobile Internet von entscheidener Rolle ist, haben Apple und Google begonnen eine eigene Datenbank mit WLAN-HotSpots aufzubauen. Neben dem iPhone 4 verwendet Apple seit Mac OS X – Snow Leopard auch Mac-Computer zum Aufbau der eigenen WLAN Access Point-Datenbank, ohne den Benutzer explizit darauf hinzuweisen. Unter „Systemeinstellungen – Sicherheit – Allgemein“ findet sich jetzt eine Option „Ortungsdienste deaktivieren“, die standardmäßig nicht markiert ist.

Die dritte Möglichkeit zur Standortbestimmung ist die sogenannte Funkmastortung (Cell Tower Positioning). Für die Ortsbestimmung über Triangulation von Funksignalen werden mindestens vier Basisstationen benötigt. Anhand der

[4] World Geodetic System 1984 ist das geodätische Referenzsystem für Positionsangaben auf der Erde und im erdnahen Weltraum

Signalstärke des gesendeten Funksignals wird dann die Position bestimmt. In einigen Fachbüchern [4, S. 485] wird auch von Funkzellenortung gesprochen (Cell ID oder Cell of Origin Locating).

Seit dem iPhone 3G gehört der GPS-Empfänger zur Hardwareausstattung, so dass alle drei genannten Möglichkeiten zur Ortung [7, S. 195 ff] zur Verfügung stehen. Das eingebaute GPS-Modul unterstützt A-GPS, das heißt mit Hilfe der Mobilfunkortung oder WPS kann die Position schneller genauer bestimmt werden. Die Genauigkeit der einzelnen Ortungsmethoden hängt stark von der Umgebung des Benutzers ab, da in Ballungszentren wie Großstädten der Radius einer Funkzelle wenige hundert Meter umfasst, während auf dem Land der Radius bis zu 30km betragen kann. Dies führt zu unscharfen Genauigkeitsangaben. So beträgt die Genauigkeit der Standortbestimmung innerhalb einer urbanen Region für die Lokalisierung per Mobilfunk je nach Netztechnologie ca. 30m – 50m (UMTS) beziehungsweise 100m – 500m (GSM), bei WLAN ca. 20m und bei GPS ca. 1 – 10m.

Aufgrund der Leistungsfähigkeit des eingebauten GPS-Empfängers des iPhones verzichten immer mehr Anwender auf ein separates Navigationsgerät im Auto und verwenden stattdessen das iPhone in Kombination mit einer Navigations-App.

### 2.1.5 Multitasking mit iOS 4

Das aktuelle Betriebssystem von Apples mobilen Endgeräten, iOS 4, beinhaltet mit Multitasking ein neues Funktionsmerkmal. Dieses ermöglicht Programmen bestimmte Aufgaben im Hintergrund zu bearbeiten, während andere Programme parallel verwendet werden können. Insgesamt stehen sieben verschiedene Multitasking-Dienste zur Verfügung [36]. Dazu zählen:

- Background audio
  ermöglicht der App Audiodateien kontinuierlich im Hintergrund abzuspielen.
- Voice over IP
  erlaubt dem Benutzer Telefonate per Internet zu führen bei gleichzeitigem Ausführen eines anderen Programms.
- Background location

Ortsänderungen können im Hintergrund verfolgt werden. Dafür wird die batterieschonende Funkmastortung verwendet, die nur auf gravierende Ortsänderungen reagiert.

- Push notifications
  Benachrichtigungsdienst für Erinnerungen und Hinweise aller Art.
  Für die Übermittlung einer Nachricht wird ein entfernter Server benötigt. Die Zustellung erfolgt auch bei ausgeschalteter App.
- Local notifications
  basiert auf Push Notifications, benötigt aber keinen Server.
- Task completion
  ermöglicht erstmalig Aufgaben zu Ende zu führen, selbst wenn die App zwischenzeitlich gewechselt wird. So können zum Beispiel Upload-Prozesse im Hintergrund fortgeführt und abgeschlossen werden.
- Fast app switching
  der aktuelle Stand eines Programms kann beim Wechsel der Anwendung gespeichert werden. Dies ermöglicht bei späterer Rückkehr zur App die nahtlose Fortsetzung. So können beispielsweise Spiele trotz Unterbrechung an gleicher Stelle fortgeführt werden.

Von den soeben kurz vorgestellten Multitasking-Diensten sind für die vorliegende Arbeit insbesondere „Background Location“, „Local Notifications“, „Task Completion“ und „Fast App Switching“ relevant. Diese sollen in der Anwendung iSightseeing implementiert werden, um so die Umsetzung des Entwurfskonzepts zu ermöglichen. Eine detaillierte Dokumentation über die Einbindung der API-Funktionen erfolgt im Kapitel Architektur.

## 2.2 Location Based City Guide

Zunächst erfolgt eine kurze Einführung in die Grundlagen der Thematik „Location Based Services“ (LBS), bevor näher auf die Konzeption eines ortsbezogenen Stadtführers eingegangen wird.
Der Begriff Location Based Service bezeichnet einen Dienst, der den Standort des Mobilfunkbenutzers berücksichtigt und ihn gleichzeitig über ein Netzwerk mit ortsabhängigen Informationen versorgt. An dieser Stelle muss darauf hingewiesen

werden, dass in der vorhandenen Fachliteratur eine Begriffsvielfalt vorherrscht und die Begriffe leider inkonsistent und zum Teil synonym verwendet werden. Einerseits werden die Begriffe Standort, Ort, Position und Location gleichgesetzt, andererseits erfolgt eine Unterscheidung zwischen Position (exakter Punkt im Raum) und Ort beziehungsweise Location (räumlicher Bereich) [19, Kapitel 2, Folie 5].
Nach einer Einteilung von Brimicombe (2002) resultieren Location Based Services aus einer Konvergenz der drei Bereiche Internet, Mobile Kommunikation und Geoinformationssysteme [19, Kapitel 2, Folie 21]. Es folgt eine schematische Darstellung aus dem Vorlesungsskript von Prof. Dr. Roth, die die Einordnung der Location Based Services sehr gut veranschaulicht.

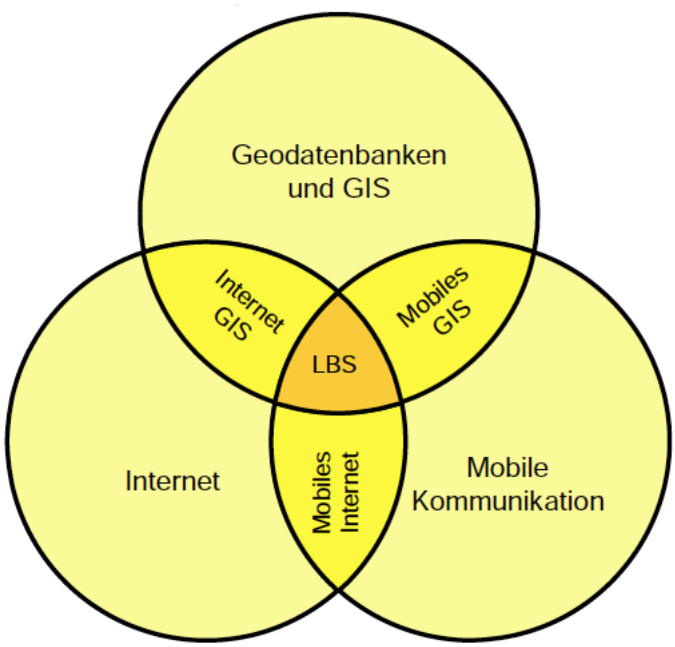


Abb. 5: Einordnung von LBS

Abbildung 5 zeigt, dass LBS die Schnittmenge der drei Bereiche Internet, Geodatenbanken und GIS sowie Mobile Kommunikation ist. Abhängig vom Zweck ihrer Anwendung kann man LBS in vier Gruppen einteilen [12]:

- sicherheitsbezogene LBS (location-safty)
  Notfall- und Notrufdienste, Pannenhilfe, Gebäude/Fahrzeug-Überwachung
- gebührenbezogene LBS (location-billing)
  Zoll- und Maut-Gebühren, Homezones
- informationsbezogene LBS (location-information)
  Wetter- und Verkehrsinformationen, Navigation, Routenplanung,
  City Guide, Gastronomieführer etc.

- positionsübermittelnde LBS (location-tracking / location-positioning)
  Personen- und Güterortung, Flottenmanagement

Neben der Einteilung nach dem Anwendungszweck kann zusätzlich eine Unterteilung nach der Art der Informationsbereitstellung erfolgen. Wird die Information automatisch zugeführt, spricht man von Push-Service. Der Pull-Service dagegen bezeichnet Dienste, wo der Nutzer explizit die Daten anfordern muss.
Es folgt eine schematische Darstellung der Autoren Steiniger, Neun und Alistair, zur Funktionsweise eines LBS; sie wurde aus ihrem Vorlesungsskript zum Thema „Foundations of LBS“ entnommen [21, S. 36].

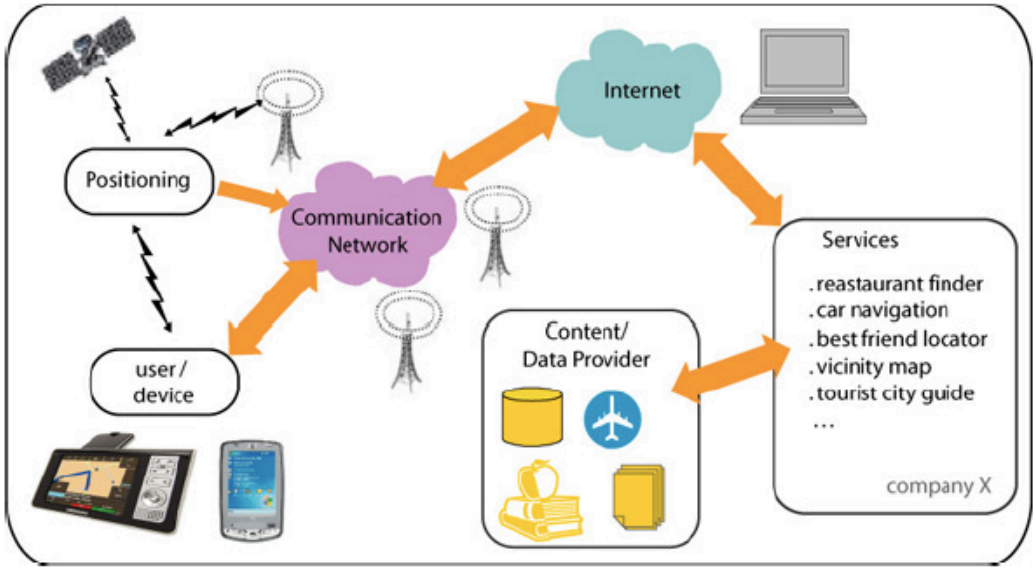


Abb. 6: LBS-Komponenten und Informationsfluss

Dargestellt werden sowohl die benötigten Komponenten eines LBS (mobiles Endgerät, Positionsbestimmung, Service Provider, Internet, Dienstanbieter, Content Provider) als auch der damit verbundene Informationsfluss. Wie aus der Skizze zu entnehmen ist, erhält der Mobilfunknutzer auf Anfrage Informationen zu den ortsabhängigen Diensten. Wird beispielsweise nach Restaurants in der näheren Umgebung gesucht, erfolgt die Positionsübermittlung über das mobile Internet an einen Dienstanbieter für ortsabhängige Informationen. Dieser ruft in seiner Datenbank Informationen zu der übermittelten Position ab und sendet das Ergebnis seiner Suchabfrage via Internet zurück an den Mobilfunkanwender. Der Benutzer kann die erhaltenen Geoinformationen zudem mit Hilfe eines Geoinformationsdienstes (GIS) auf seinem mobilen Endgerät visualisieren.

Wie soeben dargestellt, bieten Location Based Services eine Fülle an vielfältigen Informationen. Bei der Entwicklung einer Location Aware App gilt es daher, diese zu filtern und sie für den Benutzer über ein geeignetes User-Interface aufzubereiten. Dabei sollte berücksichtigt werden, dass dem Anwender ein hohes Maß an Individualisierbarkeit gewährt wird. Dies bedeutet, dass der Nutzer selbst wählen kann welche Dienste er auf seinem mobilen Endgerät sehen möchte. Durch diese gezielte Vorauswahl der zu präsentierenden Informationen erhöht sich die Usability der mobilen Anwendung und somit der Mehrwert des mobilen Informationssystems für den Nutzer. Diese zwei Kriterien – Präsentation von personalisierten ortsabhängigen Informationen und geringe Notwendigkeit zur Interaktion des Touristen mit dem mobilen Endgerät – bilden die Basis für einen Location Based City Guide. Dieser GPS-gestützte mobile Touristenführer ermöglicht sowohl ein Navigieren als auch das Abrufen und Visualisieren von ortsbezogenen Informationen. Dem Besucher einer Stadt sollte durch Kombination von Push- und Pull-Diensten der Konsum touristischer Inhalte ermöglicht werden. So besteht die Möglichkeit, den Touristen durch Hinweise auf Sehenswürdigkeiten oder Orte vom Interesse (OVI), sogenannte Points Of Interest (POI), aufmerksam zu machen (Push-Mechanismus) und ihm so die Stadt näher zu bringen. Im Gegensatz dazu kann der Tourist durch die gezielte Informationsanforderung (Pull-Mechanismus) zu einem POI, detailliert Auskunft erhalten, verschiedene multimediale Inhalte sichten und bei Bedarf über zusätzliche POI in der Umgebung informiert werden.
Bei der prototypischen Implementierung eines mobilen Touristenführers sollten daher folgende Punkte berücksichtigt werden:

- persönliches Benutzerprofil (Sprache, Alter, Interessen)
- Ausstattung des Smartphones (Hard- und Software)
- Netzanbindung (Bandbreite, Verfügbarkeit)
- zeitlicher Kontext der Benutzung (Tag, Nacht, Feiertage)
- örtlicher Kontext der Benutzung (Standort des Smartphones)

Die eben aufgeführten Kriterien dienen auch als Grundlage für die Umsetzung von iSightseeing. Eine detaillierte Beschreibung des Designs und der Implementierung der mobilen Anwendung erfolgt im Kapitel Architektur.

# 3 Verwandte iPhone-Apps

Dieses Kapitel stellt drei verwandte iPhone Apps in der Kategorie „Berlin-Stadtführer“ vor. Die ausgewählten Beispiele werden zunächst einzeln beschrieben, wobei die Punkte Design, Usability, Content und Personalization im Vordergrund der Betrachtung stehen. Aufgrund des hohen Aufwands wäre es zu komplex, die enorme Datenmenge für eine komplette Routenbetrachtung samt Inhalten als Vergleichsgrundlage für die mobilen Anwendungen zu verwenden, so dass auch hier eine Reduzierung stattfinden muss. Das Brandenburger Tor als eines der berühmtesten Wahrzeichen der Stadt Berlin eignet sich daher hervorragend als Vergleichsgrundlage. Um die Funktionalität der jeweiligen App zu überprüfen, wird einerseits die Präsentation der Sehenswürdigkeit und andererseits die Routenplanung untersucht. Anhand der so gewonnenen Erkenntnisse wird dann ein Kriterienkatalog erarbeitet. Dieser dient als Basis für einen Vergleich der Anwendungen. Gleichzeitig werden die Ergebnisse der Untersuchung auch als Bewertungsgrundlage für die Evaluation der Smartphone-Anwendung iSightseeing herangezogen.

## 3.1 Begriffsdefinition

Bevor die ausgewählten Beispiele einzeln vorgestellt werden können, erscheint es im Vorfeld notwendig, die Bewertungskriterien und die damit zusammenhängenden Begriffe zu definieren. Hierbei muss darauf hingewiesen werden, dass es sich um keine allgemeinen, aus Fachbüchern übernommenen Definitionen handelt. Sie basieren auf einem eigenständigen Ansatz und erheben nicht den Anspruch auf Vollständigkeit.

### 3.1.1 Design

Für die vorliegende Arbeit wird das Design der iPhone-Apps hinsichtlich folgender Punkte betrachtet:

- Gestaltung (Layout, Farbkomposition, Datenpräsentation)
- Individualität
- Wiedererkennungswert

Die Untersuchung umfasst sowohl das App-Icon und als auch das User-Interface.

### 3.1.2 Usability

Die Usability ist generell eine eher subjektive Wahrnehmung und daher schwer zu bewerten. Ausschlaggebend für die Einstufung der Bedienbarkeit ist letztendlich die positive User Experience. Um eine möglichst objektive Bewertung vornehmen zu können, ist es notwendig, vorher verschiedene Merkmale festzulegen. Für die Untersuchung der Usability werden daher folgende Kriterien herangezogen:

- Benutzerführung (klare Strukturierung nach funktionalen Bereichen)
- intuitive Bedienbarkeit per Touchscreen (keine Anleitung notwendig)
- fingergerechte Bedienelemente (Buttongröße)

Bei der Usability-Betrachtung der Apps bezieht sich die Bewertung lediglich auf das User-Interface.

### 3.1.3 Content

Der Fokus der Betrachtung liegt hier auf der Präsentationsform und dem Format der Inhalte. So bieten nicht alle in der vorliegenden Arbeit vorgestellten Anwendungen Audio- und Videodateien zu den POIs an. Zu der Qualität der einzelnen Fotos wird keine bewertende Aussage getroffen, da dies vom jeweiligen Geschmack des Betrachters abhängt. Dieser neutrale Ansatz kann jedoch nicht für die Betrachtung der Textinhalte gelten, da der Informationsgehalt nach folgenden Kriterien untersucht werden sollte:

- Lesbarkeit (Schreibstil und roter Faden)
- Verständlichkeit (einfache Wortwahl oder Überladung mit Fachbegriffen)
- Übersichtlichkeit (gruppierte Informationen oder Fließtext)

Bei der vorliegenden Arbeit wird jedoch keine Betrachtung der Textinhalte mit anschließender Bewertung erfolgen, da dies fachfremde Qualifikation erfordern würde. Untersucht wird stattdessen die visuelle Lesbarkeit des Textes anhand folgender technischer Aspekte:

- Übersichtlichkeit der Informationspräsentation (Anordnung)
- Kontrastverhältnis zwischen Schrift- und Hintergrundfarbe
- Schriftgröße, Zeilenabstand, Schriftart
- Zoomfunktion für den Textbereich oder den Text selbst
- alternative Textwiedergabe in audiovisueller Form

### 3.1.4 Personalization

Dieses Kriterium soll helfen, die ausgewählten mobilen Anwendungen, daraufhin zu überprüfen, ob und inwieweit sie dem Benutzer die Möglichkeit zur einer individuellen Anpassung bieten. Dabei sollte untersucht werden, ob folgende Optionen zur Disposition stehen:

- individuelle Routen planen
- POIs löschen, hinzufügen oder verschieben
- POI-Kategorien ein- und ausblenden
- eine Favoritenliste erstellen
- POIs bewerten
- Anbindung an Social Networks

## 3.2 Ausgewählte Beispiele

### 3.2.1 mTrip – Berlin-Reiseführer

Laut eigenen Angaben auf seiner Homepage bietet der kanadische Anbieter mTrip Travel Guides seit dem 21.06.2010 interaktive Reiseführer für iPhone und iPod Touch an [37]. Das Angebot umfasst derzeit dreizehn Metropolen Europas, sechs Metropolen Nordamerikas und vier Metropolen Asiens [37]. Der mobile Reiseführer wird in den folgenden fünf Sprachen angeboten: Deutsch, Englisch, Französisch, Italienisch und Spanisch [38]. Für den europäischen Markt wurde eine Kooperation mit Europas führendem Reiseverleger Falk Content & Internet Solutions abgeschlossen, der die multimedialen Inhalte für die App zur Verfügung stellt. Die Firma mTrip Travel Guide bewirbt den eigenen Reiseführer auf ihrer Homepage als „einzigartigen und interaktiven Reiseführer mit exklusiven Eigenschaften“ [37].
Der Erwerb der mTrip App kostet pro Stadt 4,99 EUR [38]. Da die App keine Internetverbindung für den Betrieb benötigt, fallen keine zusätzlichen Kosten wie Roaming-Gebühren für ausländische Touristen an. Jedes Update ist gratis.

#### 3.2.1.1 Design

Die iPhone-App mTrip sticht durch ein gelungenes Farbkonzept hervor. Die dominierenden Farben sind Schwarz, Grün und Weiß. Das Interfacedesign besteht

hauptsächlich aus Farbverläufen, um einen 3D-Look zu erzeugen (siehe Abbildung 7; Screenshot 1, 2 und 3).

Abb. 7: Screenshots User Interface – mTrip Travel Guide

Trotz der Fülle der angezeigten Informationen wirkt die App gut strukturiert und übersichtlich. So werden die Sehenswürdigkeiten anschaulich in einer Listendarstellung präsentiert (siehe Abbildung 7, Screenshot 1). Der zweite Screenshot zeigt die Detailansicht zum ausgewählten POI. Der dritte Screenshot präsentiert die Kartenansicht.

Das User Interface oder „Human Interface“, wie Apple es nennt, besitzt einen hohen Wiedererkennungswert und ein hohes Maß an Individualität, da beim Design nicht vorgegebene Standardelemente des UIKits verwendet wurden. Stattdessen sind eigene Bedienelemente kreiert worden wie beispielsweise die TabBar oder der Slider. Dieses Designkonzept findet sich auch im App Icon wieder. Die jeweilige Stadt wird immer im oberen Bereich des Icons mit einem eigenen 3-Letter-Code angegeben, ansonsten bleibt das Icon unverändert.

Abb. 8: App Icon – mTrip Travel Guide

Aufgrund des schlichten und klaren Designs des Icons besteht ein hoher Wiedererkennungswert für die App. Die Funktion der App ist daraus für den Anwender jedoch nicht ersichtlich. Die Abkürzung BER ist kein allgemein gültiger

3-Letter-Code für Berlin, so dass nicht automatisch eine Assoziation mit der Stadt Berlin entsteht.
Um einen direkten Vergleich mit den anderen Anbietern zu ermöglichen, werden die Icons der ausgewählten Apps in der Tabelle 2 einander gegenüber gestellt.

#### 3.2.1.2 Usability

Für die funktionale Strukturierung der App wird das Konzept des Tab Bar Controllers verwendet. Hierbei sollte jeder Button auf der Tab Bar eine andere Funktion der App repräsentieren. Wie aus der Abbildung 7 ersichtlich, hält sich mTrip an diese Vorgaben. Die Navigation gliedert sich in Route, Reiseführer, Navigation, Senden und Mehr. Hinter Mehr verstecken sich noch weitere Funktionen der App, die hier jedoch keine Erwähnung finden, da sie nicht relevant sind. Grundsätzlich wird die Tab Bar auf fünf Bedienelemente begrenzt; diese Vorgabe basiert auf der Klasse UITabBarController des UIKit frameworks von Apple.
Der Anwender kann sich die einzelnen Stationen einer Tour über den Button Route anschauen. Abhängig vom An- und Abreisedatum ändert sich die vorgeschlagene Tour. Diese Benutzerführung ist aufgrund der angebotenen Funktionsvielfalt etwas verwirrend und entspricht somit nicht einer intuitiven Bedienung. Der Anwender benötigt eine gewisse Zeit, um sich mit den Optionen der einzelnen Funktionen vertraut zu machen. Die Icons der TabBar lassen zudem nicht klar auf ihre Funktion schließen. Die Größe der Bedienelemente entspricht den Richtlinien der Apple Human Interface Guidelines. Die Highlights der mTrip App sind die integrierte Turn-by-Turn-Navigation (siehe Abbildung 9; Screenshot 1) zu Fuß und per U-Bahn sowie die Möglichkeit, Postkarten mit Fotos als Motivvorlage per Email zu versenden (siehe Abbildung 9; Screenshot 2). Die Fotos können entweder aus der vorhandenen mTrip-Datenbank stammen oder aus dem Bildarchiv des iPhones. Zusätzlich kann direkt auf die Kamerafunktion des iPhones zugegriffen werden.

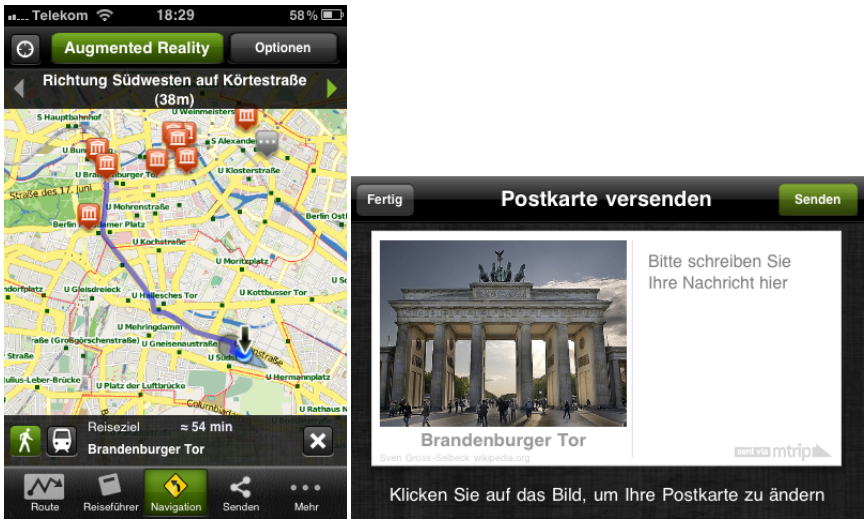


Abb. 9: Screenshots Turn-by-Turn-Navigation und Postkarte - mTrip Travel Guide

### 3.2.1.3 Content

Die Präsentation der Inhalte erscheint aufgeräumt und übersichtlich. Die Texte sind aufgrund des guten Kontrastverhältnisses, der Schriftgröße und der Schriftart gut leserlich. Eine Zoomfunktion für die in der Textform enthaltenen Informationen zu den verschiedenen Sehenswürdigkeiten oder anderen POIs existiert nicht. Bis auf Fotos werden keine audiovisuellen Inhalte zur Verfügung gestellt.

Abb. 10: Screenshot Fernsehturm am Alex – mTrip Travel Guide

Obwohl im Vorfeld klar formuliert wurde, dass keine Bewertung des Fotomaterials erfolgen wird, muß hier jedoch eine Ausnahme gemacht werden, da die Mängel so gravierend sind. Wie in der Abbildung 7 (erster Screenshot) zu erkennen ist, fehlen in der Routenplanübersicht teilweise die Thumbnails der Sehenswürdigkeiten. Des

Weiteren gibt es irreführende Fotos zu Sehenswürdigkeiten, wie zum Beispiel das Foto zum Fernsehturm, wo Teile des Reichstagsgebäudes im Bildvordergrund zu sehen sind und so das Bild dominieren, während der Fernsehturm im Hintergrund kaum wahrgenommen wird (siehe Abbildung 10). Bei einem ortsunkundigen Touristen können solche Mängel bei der mobilen Stadtführung für Verwirrung sorgen.

#### 3.2.1.4 Personalization

Die iPhone App mTrip Travel Guide bietet alle unter Personalization aufgeführten Merkmale an. So besteht die Möglichkeit, die Tour den individuellen Bedürfnissen je nach Interessengebiet durch Löschen, Verschieben oder Hinzufügen von POIs anzupassen. Über die Option Reiseführer kann man zusätzliche POIs aus anderen Kategorien wie beispielsweise Shopping oder Restaurants zur Tour hinzufügen. Ebenso kann der Anwender seine Position lokalisieren und diese auf einer Karte anzeigen lassen. Durch die Auswahl der verfügbaren POI-Kategorien und die Angabe eines relativen Radius können zusätzlich POIs der näheren Umgebung auf der Karte angezeigt werden. Zudem besteht für den Benutzer die Möglichkeit, eine Favoritenliste zu erstellen und die POIs zu bewerten. Diese Bewertung steht auch anderen Benutzern der iPhone App zur Verfügung und hat somit einen Community-Charakter. Des Weiteren können neben Bewertungen auch eigene POIs inklusive Fotos erstellt und anschließend zur Tour hinzugefügt werden.

Die personalisierten Touren und Postkarten können zusätzlich per Facebook-Anbindung Freunden mitgeteilt werden.

Abb. 11: Screenshots Personalization – mTrip Travel Guide

### 3.2.2 Navigaia – Berlin Multimedia Travel Guide

Das französische Reiseunternehmen Navigaia vertreibt über sein Tochterunternehmen PocketVox multimediale Guides für überwiegend europäische Städte. Diese werden in den Sprachen Deutsch, Englisch und Französisch angeboten, wobei jede Sprache separat erworben werden muss. Der Preis pro App liegt bei 5,99 EUR [39]. Der Content Provider ist eine weitere Tochterfirma, Intermèdes Le Loisir Culturel, die sich auf die Konzeption und den Vertrieb von kulturellen Reisen spezialisiert hat. Laut eigenen Angaben auf der iTunes-Produktseite ist der Navigaia Travel Guide der „einzige Multimedia-Reiseführer für Berlin“ [39]. Der mobile Guide bietet Informationen zu ca. 450 Orten an. Außerdem stellt er circa 65 Videos, drei Audioführungen und eine interaktive Karte zur Verfügung.
Dem Benutzer entstehen durch die Anwendung keine zusätzliche Kosten, da keine Internetverbindung benötigt wird. Die jährlichen Updates sind kostenlos.

#### 3.2.2.1 Design

Das Design des Xcode-Application-Templates wird beibehalten, so dass auch das Farbkonzept zum größten Teil übernommen worden ist. Da die Standardelemente des UIKits ohne farbliche Abänderung verwendet werden, besteht kein wirklicher Wiedererkennungswert für das User Interface.

Abb. 12: Screenshots User Interface – Navigaia Travel Guide

Nur beim Startscreen (siehe Abbildung 12; Screenshot 1) ist die Anwendung anhand des Logos zu erkennen. Die dominierenden Farben sind hier die Standardfarben des UIKits, das heißt grau, blau, weiß und schwarz. In der Listenansicht der

Sehenswürdigkeiten (siehe Abbildung 12; Screenshot 2) werden keine Vorschaubilder angezeigt.

Abb. 13: ausgewählte App Icons zum Vergleich - Navigaia Travel Guide

Das Navigaia App-Icon besteht für jede Stadt aus einem stadtspezifischen Symbol (berühmte Sehenswürdigkeit oder Assoziation) und einer individuell wechselnden Hintergrundfarbe (siehe Abbildung 13). Das einzig Beständige ist der NAVIGAIA-Schriftzug im unteren Teil des Icons. Für die Stadt Berlin ist die Hintergrundfarbe rot und als Symbol dient das Brandenburger Tor. Paris und New York lassen sich leicht anhand der Popularität ihrer jeweiligen Sehenswürdigkeiten erkennen, während dies bei der Identifikation von Venedig und München aufgrund ihrer Events, wie des Karnevals oder des Oktoberfestes, nicht zwangsläufig gegeben ist.
So besteht beispielsweise bei der App für die Stadt München die Gefahr, dass durch den Bierkrug die App eher mit einem generellen Gastronomieführer verwechselt wird.
Ohne den Schriftzug NAVIGAIA bestünde kein Wiedererkennungswert der App-Icons, da auch andere mobile Travel Guides die berühmteste Sehenswürdigkeit einer Stadt als Designelement des Icons verwenden.

#### 3.2.2.2 Usability

Die Navigaia-App wirkt in ihrem Erscheinungsbild strukturiert und aufgeräumt. Das einzig Verwirrende bei der Benutzerführung ist der häufige Wechsel des Navigationskonzepts (siehe Abbildung 12, Screenshot 2). Die TabBar bietet keine funktionale Unterteilung der App, sondern dient hier lediglich zum Auswählen des Sortierfilters. Dies hätte man genauso gut über eine Toolbar oder einen Scopebereich erreichen können.
Die Kartenansicht bietet versteckte Funktionen an, mit denen man die Rubrik der POIs wählen kann. In Abhängigkeit von der gewählten Rubrik ändert sich die Kartendarstellung (siehe Abbildung 14, Screenshot 3) und zusätzlich die Filteroption

(siehe Abbildung 14, Screenshot 2). So kann der Anwender die angezeigten POIs nach Kategorie, Bewertung oder Preis filtern.

Abb. 14: Screenshot Kartenansicht – Navigaia Travel Guide

Für die Bedienung werden die Standardelemente des UIKits verwendet, so dass sie den Richtlinien der Apple Human Interface Guidelines entspricht.

#### 3.2.2.3 Content

Die App präsentiert die Inhalte übersichtlich und strukturiert. Die Lesbarkeit der Texte ist aufgrund des guten Kontrastverhältnisses zwischen Schriftfarbe und weißem Hintergrund gegeben. Die Textinhalte der Sehenswürdigkeiten erscheinen wegen ihrer kleinen Schriftgröße und des geringen Zeilenabstandes gedrungen und sind somit nur aus der Nähe lesbar. Leider wird keine Textzoomfunktion angeboten, um dieses Manko zu beheben. Als ausgleichender Ersatz wird dafür bei einigen Sehenswürdigkeiten ein Video angeboten, in dem der Textinhalt von einem Sprecher wortgenau wiedergegeben wird. Zusätzlich bietet Navigaia drei ausgewählte Audioführungen durch die Stadt an. Bei diesen Führungen handelt es sich um festgelegte Routen mit jeweils unterschiedlichen Startpunkten (siehe Abbildung 15) und Verläufen (siehe Abbildung 16). Die Reihenfolge der Wegpunkte ist festgelegt.

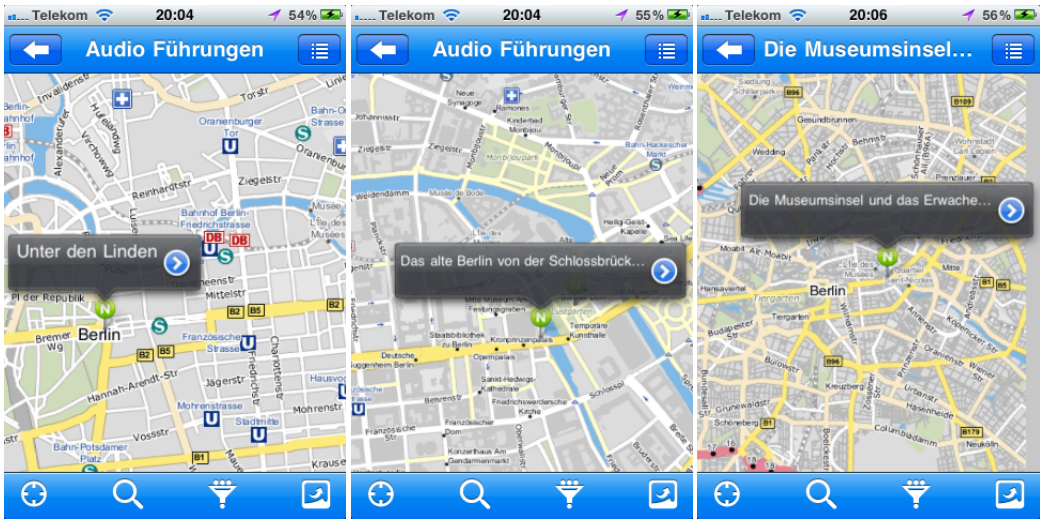


Abb. 15: Screenshots Startpunkte der Touren – Navigaia Travel Guide

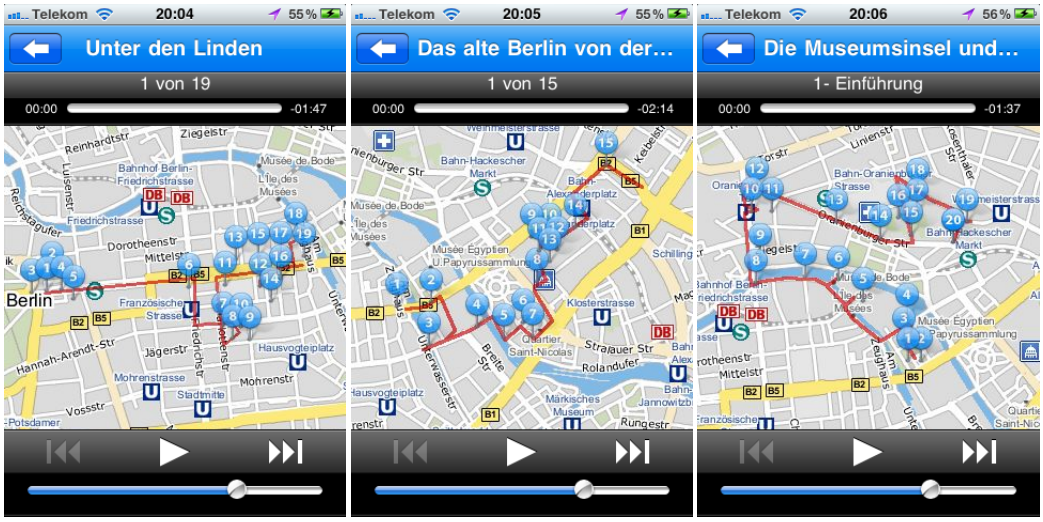


Abb. 16: Screenshots Verlauf der Touren – Navigaia Travel Guide

Alle drei Touren beziehen sich lediglich auf Sehenswürdigkeiten in Berlin-Mitte. Zur Auswahl stehen Touren mit den Themen „Unter den Linden“, „Das alte Berlin von der Schlossbrücke zum Alexanderplatz“ und „Die Museumsinsel und das historische Berlin“. An dieser Stelle muss darauf hingewiesen werden, dass die Abbildung 15 Screenshot 3 einen abweichenden Titel „Die Museumsinsel und das Erwachen“ vom gesprochenen Text enthält.

Der Schwerpunkt der Audioführungen liegt auf dem historischen Kern Berlins. Daher stellte sich die Frage, ob prinzipiell auch ehemals West Berliner Sehenswürdigkeiten in der Datenbank enthalten sind. Bei einer Stichprobe zur Überprüfung dieser Fragestellung wurde mit Erstaunen festgestellt, dass die Kaiser-Wilhelm-Gedächtnis-Kirche in der Auflistung der Sehenswürdigkeiten fehlt.

#### 3.2.2.4 Personalization

Wie unter Content dargestellt, bietet die Navigaia App keine Möglichkeit zur individuellen Routenplanung. Der Anwender kann nur zwischen den drei festgelegten Routen wählen. Die Personalisierung der App beschränkt sich auf die Auswahl der anzuzeigenden POIs in der Kartenansicht. So bietet die mobile Anwendung je nach Kategorie verschiedene Filteroptionen an, wie aus den Screenshots der folgenden Abbildung zu entnehmen ist.

Abb. 17: Screenshots Filteroptionen – Navigaia Travel Guide

Über die Detailansicht der Sehenswürdigkeiten besteht die Möglichkeit, eine Favoritenliste anzulegen. Diese Liste kann nach der Distanz zum aktuellen Standort des Anwenders sortiert und auf der Karte als Stecknadeln angezeigt werden. Sie enthält aber keine Routendarstellung. Die Karte basiert auf Kartenmaterial von NAVTEQ und bietet außer der Positionsbestimmung keine Navigationshilfe.

Eine Option zum Hinzufügen eigener POIs ist nicht vorhanden. Überdies existiert weder eine Funktion zum Bewerten der Sehenswürdigkeiten noch eine Anbindung an Soziale Netzwerke.

### 3.2.3 City Walks – Berlin Map and Walking Tours

Im Gegensatz zu den anderen vorgestellten Apps arbeitet der Hersteller GPSmyCity.com nicht mit einem Reiseverlag als Content Provider zusammen. Getreu Web 2.0 setzt man hier auf Community Building und Crowdsourcing über das eigene Portal. Das Monetarisierungskonzept basiert auf der prozentualen Gewinnbeteiligung von sogenannten Hobbyautoren am Verkaufserlös der App. Jedes Portalmitglied, das Lust und Freude daran hat, eine Tour zusammenzustellen, kann diese einreichen. Für die Abnahme eines Tourvorschlags gelten folgende Voraussetzungen [40]:

- sehr gute Englischkenntnisse in Wort und Schrift
- mindestens sechs monatiger Aufenthalt vor Ort
- Erstellung von Fotoaufnahmen und Beschreibungen zu den Sehenswürdigkeiten
- Angabe der Anschrift und der GPS-Koordinaten zu den Sehenswürdigkeiten
- Anfertigung eines Audioberichts

Über das Portal von GPSmyCity.com werden momentan mehr als 2.000 Walking Tours für über 180 Städte weltweit angeboten [41]. Basierend auf dem Community-Konzept können so für eine Stadt mehrere verschiedene Guides zur Verfügung stehen, die jedoch alle das gleiche App-Icon besitzen. Für die Stadt Berlin gibt es derzeit 4 verschiedene Angebote. Die Preisgestaltung hängt dabei vom Inhaltsangebot ab. So beträgt der Kaufpreis für eine ausgewählte Audiotour 2,99 EUR. Das teuerste Angebot mit 4,99 EUR bietet zwar elf verschiedene Touren an, diese beinhalten jedoch nur Bild- und Textmaterial. Die Angebote, die Audioführungen anbieten, sind teilweise anhand eines Lautsprechersymbols im App-Icon gekennzeichnet.

An dieser Stelle muss darauf hingewiesen werden, dass trotz des Konzepts der Einbindung von Hobbyautoren bei der vorliegenden App überwiegend Bild- und Textmaterial aus der freien Enzyklopädie wikipedia stammen. Dies wurde leider erst nach dem Erwerb dieser City Walks App festgestellt.

OpenStreetMap dient als Kartenmaterial für die Kartenansicht der Touren.

### 3.2.3.1 Design

Das Design des User Interface ist schlicht und funktional gestaltet. Die Farbkomposition ist durch die Abweichung von den Standardfarben individuell. Die Farbverläufe im Hintergrund dienen zum optischen Trennen der Tabellenzellen (siehe Abbildung 18; Screenshot 1). Ein Wiedererkennungswert des UI-Designs liegt trotz individueller Farbgestaltung nicht vor.

Abb. 18: Screenshots User Interface – City Walks

Das City Walks App-Icon wird von einer roten Silhouette auf blauem Grund dominiert. Sie belegt ca. ein Drittel des Icons. Dargestellt wird ein Mann in Bewegung, der etwas in seinen ausgestreckten Händen hält (siehe Abbildung 19). Der Schriftzug City Walks sticht aufgrund des Farbkontrastes hervor.

Für Audioführungen kann zusätzlich ein Lautsprechersymbol in der oberen rechten Bildecke eingeblendet werden (Abbildung 19). Dies wird jedoch nicht durchweg eingehalten.

Das Design des User Interfaces spiegelt sich nicht im App-Icon wieder. So besteht keine direkte visuelle Verknüpfung der beiden Komponenten miteinander. Eine Assoziation mit der Stadt Berlin ist nicht gegeben, da weder ein Hinweis in symbolischer noch in schriftlicher Form vorliegt.

Abb. 19: App-Icon – City Walks

### 3.2.3.2 Usability

Eine intuitive Bedienung des mobilen Stadtführers ist aufgrund der Verwendung der bekannten Navigationskonzepte des UIKits gegeben. Die TabBar trennt die App sauber nach Funktionalität, wie es in den Richtlinien der Apple Human Interface Guidelines vorgeschrieben ist. Eine Einarbeitungszeit ist so nicht notwendig. Die Bedienelemente sind Touchscreen-gerecht (siehe Abbildung 18).

### 3.2.3.3 Content

Wie bereits in der Einleitung zu der App City Walks erwähnt, wird bei dem hier vorgestellten Beispiel größtenteils Bild- und Textmaterial der freien Enzyklopädie wikipedia verwendet (siehe Abbildung 20). Die Darstellung der Inhalte ist übersichtlich und gut strukturiert. Die Lesbarkeit ist aufgrund des Farbkontrastes (Schrift/Hintergrund) gegeben. Eine Textzoomfunktion ist nicht integriert.

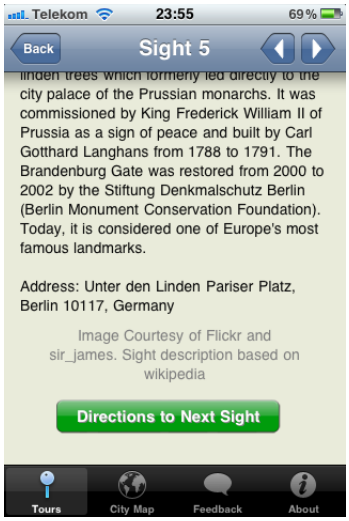


Abb. 20: Screenshot wikipedia Textmaterial – City Walks

### 3.2.3.4 Personalization

Die mobile Applikation bietet keinerlei Möglichkeit, die Stadtführung zu personalisieren. Weder kann man die Reihenfolge der Wegpunkte verändern noch eigene POIs hinzufügen. Ebenso wenig besteht die Option, zusätzliche Informationen neben den Sehenswürdigkeiten abzurufen. So enthält die App beispielsweise keine ortsbasierten Angaben zu Restaurants, Bars oder dergleichen in der näheren

Umgebung. Es wird lediglich eine Tour mit der Bezeichnung „Berlin's Top Nightclubs“ angeboten.

## 3.3 Vergleichende Darstellung

Die bisherige Untersuchung erfolgte anhand der vorher formulierten Kriterienpunkte. Dabei wurde jedes Merkmal einzeln betrachtet. Die so gewonnenen Ergebnisse werden nachfolgend in tabellarischer Form präsentiert. Dabei werden die einzelnen Kriterien von Design, Usability, Content und Personalization miteinander verglichen. Die Überprüfung, ob die aufgestellten Kriterien erfüllt worden sind, erfolgt mit Hilfe einer vereinfachten Notation (+, 0, -). Für die Teilwertungen werden nur die Pluspunkte berücksichtigt. Die Gesamtwertung setzt sich dann aus den vier Teilwertungen zusammen.

| Analyse | mTrip | Navigaia | City Walks |
|---|---|---|---|
| | | | |
| **Design** | | | |
| Gestaltung | + | 0 | 0 |
| Individualität | + | + | - |
| Wiedererkennungswert | + | - | - |
| **Teilwertung** | 3 | 1 | 0 |
| **Usability** | | | |
| Benutzerführung | + | + | + |
| Bedienbarkeit | 0 | 0 | + |
| Bedienelemente | + | + | + |
| **Teilwertung** | 2 | 2 | 3 |
| **Content** | | | |
| Übersichtlichkeit | + | + | + |
| Lesbarkeit | + | + | + |
| Text-Zoom | - | - | - |
| Audio-/Videoinhalte | - | + | - |
| **Teilwertung** | 2 | 3 | 2 |
| **Personalization** | | | |
| Routenplanung | + | 0 | 0 |
| POI-Bearbeitung | + | - | - |
| POI-Kategorieauswahl | + | + | - |
| Favoritenliste | + | + | - |
| Bewertung / Share | + | - | - |
| **Teilwertung** | 5 | 2 | 0 |
| **Gesamtwertung** | 12 | 8 | 5 |

Tabelle 2: Tabellarischer Vergleich der ausgewählten Apps

Die Gesamtauswertung verdeutlicht die zum Teil gravierenden Unterschiede bei der Möglichkeit, den mobilen Stadtführer zu personalisieren. Von den ausgewählten Apps bietet nur mTrip zahlreiche Funktionen zur individuellen Routenplanung an.

## 3.4 Kriterien für die Evaluierung

Aus der Analyse ist deutlich geworden, welche Merkmale im Kontext einer positiven User Experience für die Konzeption eines mobilen Stadtführers wichtig sind. Die so gewonnenen Erkenntnisse fließen in die prototypische Umsetzung von iSightseeing mit ein. Dabei liegt ein besonderer Schwerpunkt in der Implementierung von Funktionen, die eine Personalisierung beziehungsweise Individualisierung der Anwendung ermöglichen, insbesondere eine personalisierte Routenplanung. Der Anwender sollte die Möglichkeit erhalten, durch die Auswahl von POIs eine Tour eigenständig nach seinen individuellen Bedürfnissen zusammenstellen zu können. Dafür ist eine Option zum Hinzufügen oder Löschen von POIs notwendig. Die Reihenfolge der Wegpunkte sollte zudem veränderbar sein, beispielsweise durch eine manuelle Anordnung der POIs oder eine Sortierung nach Distanz beziehungsweise Alphabet. Zudem könnte die Anbindung an Soziale Netzwerke die Individualisierung der App zusätzlich fördern.

Weitere wichtige Punkte für die Evaluierung sind Abruf, Präsentation und Visualisierung von Informationen. Zu Letzterem zählen sowohl die Darstellung der Tourroute auf einer digitalen Karte als auch das Design des User Interface. Dies beinhaltet eine intuitive Bedienbarkeit der mobilen Applikation.

Anhand der erarbeiteten Kriterien kann allerdings keine äquivalente Analyse der iSightseeing-App erfolgen, da es schwierig ist, die Anwendung des „Testsiegers“ mit dem Prototyp direkt zu vergleichen. Beide Anwendungen befinden sich in ganz unterschiedlichen Entwicklungsstadien, weswegen keine Grundlage für eine Benotung gegeben ist. Zudem steckt der Prototyp noch in der Testphase, so dass nicht alle Funktionalitäten getestet und somit bewertet werden können. Anstelle eines direkten Vergleiches werden die oben definierten positiven Eigenschaften eines mobilen Stadtführers als Grundlage für die Entwicklung eines eigenen Konzeptes herangezogen.

Für einen besseren Überblick wird im Anschluss ein eigener Kriterienkatalog in tabellarischer Form konzipiert. Untersucht werden dabei die bereits genannten

Aspekte – Design, Usability, Content und Personalization – anhand von spezifischen Fragestellungen.

### 3.4.1 Kriterienkatalog

| **Name** | Name der mobilen Anwendung |
|---|---|
| **Anwendungsbereich** | Persönliche Führung und/oder generelles Touristeninformationssystem? |
| **Design** | |
| Icon | Wie ist das Icon gestaltet?<br>Besteht ein Wiedererkennungswert?<br>Geht die Funktionalität aus dem Icon hervor?<br>Findet sich das Design des User Interface im Icon wieder? |
| User Interface | Besteht eine sinnvolle Aufteilung und Gliederung der Benutzeroberfläche?<br>Besteht ein Wiedererkennungswert?<br>Basiert die Gestaltung auf UIKit-Standardbedienelementen? |
| **Usability** | |
| Navigationskonzept | Welches View Controller-Konzept liegt zugrunde?<br>Wird eine konsequente Benutzerführung beibehalten? |
| Bedienbarkeit | Geht die Funktionalität intuitiv aus der Benutzerführung hervor?<br>Benötigt die Anwendung eine Internetverbindung? |
| Bedienelemente | Wie groß sind die Bedienelemente?<br>Ist ihre Bedienung fingergerecht? |
| Routenfunktionalität | Ist eine Routensuche möglich?<br>Wie wird die Route dargestellt?<br>Werden verschiedene Fortbewegungsmöglichkeiten für die Routenberechnung angeboten?<br>Spaziergänge?<br>Fahrradtouren?<br>Öffentliche Verkehrsmittel?<br>PKW? |
| Kartenfunktionalität | Welche Perspektive für die Kartendarstellung liegt vor?<br>Werden Ebenen angezeigt?<br>Besteht eine Interaktivität für Objekte?<br>Kann die Anzeige von Objekten gefiltert werden?<br>Auf welchem Kartenmaterial basiert die Anwendung? |
| Tourenfunktionalität | Sind Touren möglich?<br>Gibt es vordefinierte Touren?<br>Können zusätzlich Touren individuell zusammengestellt werden?<br>Können POIs ausgewählt werden? Wie?<br>Können POIs nach Kategorien ausgesucht werden? Wie? |
| **Content** | |
| Informationsangebot | Welche Informationen werden zur Verfügung gestellt?<br>Wie sind die Informationen aufbereitet?<br>Ist die Darstellung von Text und Bildern benutzerfreundlich?<br>Besteht eine Skalierbarkeit für Text oder Bildschirmausschnitte?<br>Sind die angebotenen Informationen aktuell?<br>Werden Zusatzinformationen zu POIs bereitgestellt? |
| Quelle | Wird der Inhalt redaktionell gepflegt?<br>Stammt der Inhalt aus Wikipedia? |
| Sprachenangebot | Welche Sprachen werden angeboten? |
| Multimediadaten | Werden Audioinhalte integriert? |

| | |
|---|---|
| | Sind Videoinhalte eingebunden? |
| **Personalization** | |
| Routenplanung | Wie können Touren personalisiert werden?<br>Können eigene POIs erstellt werden? Wie?<br>Werden diese auf der Karte angezeigt?<br>Ist der Startpunkt einer Tour frei wählbar? |
| Zielgruppe | Gibt es eine spezielle Zielgruppe? Welche?<br>Können Informationen nach Altersgruppen gefiltert werden?<br>Können Informationen nach Interessengebieten gefiltert werden? |
| Favoritenliste | Ist eine Favoritenliste von POIs erstellbar? Wie? |
| Bewertung/Share | Können POIs bewertet werden?<br>Besteht eine Anbindung an Soziale Netzwerke? |

Tabelle 3: Kriterienkatalog

Basierend auf den Fragestellungen des eben vorgestellten Kriterienkatalogs wird der mobile Stadtführer iSightseeing entwickelt. In dem folgenden Kapitel zur Anwendungsarchitektur werden dazu die notwendigen technischen Grundlagen der App erläutert.

# 4 Anwendungsarchitektur

Im Kapitel „Verwandte iPhone-Apps“ wurden drei ausgewählte mobile City Guides für die Stadt Berlin vorgestellt und analysiert. Die so gewonnenen Erkenntnisse über die Eigenschaften eines mobilen Stadtführers dienen als Anforderungskatalog für die Umsetzung von iSightseeing. Bevor jedoch auf die Grundarchitektur der prototypischen iPhone-App eingegangen werden kann, ist es notwendig, im Vorfeld sowohl Zielgruppe als auch funktionale Anforderungen mit Hilfe einer Anforderungsanalyse zu ermitteln. Für die Analyse und das Design des Systems wird UML 2.3 als Modellierungssprache verwendet.

## 4.1 Anforderungsanalyse

Innerhalb der Anforderungsanalyse werden die Parameter für die Zielgruppe und die funktionalen Anforderungen definiert. Die Bestimmung einer Zielgruppe ist dabei Voraussetzung für die Erstellung von Anwendungsfällen (Use Case). Mit Hilfe dieser Anwendungsszenarien soll die Interaktion des Nutzers (Akteur) mit der Anwendung (System) beschrieben werden. Anhand der so entstehenden Use Case-Diagramme soll die Struktur der Anwendung modelliert werden. Ziel dieser bedarfsorientierten Softwareentwicklung ist die Erfüllung der individuellen Anforderungen des Benutzers durch die implementierten Funktionen.

### 4.1.1 Zielgruppe

Die hier entwickelte Anwendung iSightseeing richtet sich primär an ausländische Berlintouristen, die sich in einer für sie fremden Stadt befinden und touristische Informationen abrufen wollen. Dabei kann die mobile Anwendung als Städteführer und mobiles Touristeninformationssystem dienen.

Prinzipiell können Touristen grob in zwei Kategorien eingeteilt werden. Individualtouristen planen meistens ihre An- und Abreise und den Aufenthalt vor Ort in Eigenregie. Gruppentouristen hingegen überlassen die gesamte Reiseplanung einem Reiseveranstalter. In der vorliegenden Bachelorarbeit wird der Schwerpunkt auf Individualtouristen gelegt. Innerhalb dieser Gruppe können die Benutzer zusätzlich nach folgenden Kriterien unterschieden werden:

- Alter
- Interesse
- Ortskenntnis (ortskundig oder ortsunkundig)
- Erfahrung im Umgang mit mobilen Endgeräten

Basierend auf diesen benutzerspezifischen Unterschieden können die Anforderungen an Inhalte und Funktionen des mobilen Stadtführers bestimmt werden. An dieser Stelle muss allerdings darauf hingewiesen werden, dass sich durch die beliebige Kombination der einzelnen Kriterien eine Vielzahl von verschiedenen Benutzergruppen definieren lässt. Für die vorliegende Arbeit kann jedoch nicht jede mögliche Benutzergruppe berücksichtigt werden, da dies den Rahmen erheblich sprengen würde.
Es erfolgt stattdessen eine Festlegung auf eine selbstdefinierte Gruppe, in der Alter, Interessen, Ortskenntnis und Erfahrung im Umgang mit mobilen Endgeräten stark variieren können. Weitere Merkmale wie Bildung, Herkunft, Familienstand oder Geschlecht werden dabei nicht berücksichtigt. Dieser Ausschluss führt zu folgender zusammengefasster Benutzergruppe.

| Benutzergruppe | Alter | Interesse | Ortskenntnis | Erfahrung im Umgang mit mobilen Endgeräten |
|---|---|---|---|---|
| Junge Menschen, Menschen mittleren Alters, Rentner | 18 – 69 | Unterhaltung, Information | keine – mittel | gering – hoch |

Tabelle 4: Benutzergruppe „Ausländischer Berlintourist“

Die Benutzergruppe „Ausländischer Berlintourist“ umfasst somit eine Zielgruppe mit einer weit gefassten Altersstruktur, die von der Volljährigkeit bis zum Rentenalter reicht. Für diesen eben definierten Typ von Individualtouristen kann aufgrund seines Alters von einer gewissen Kaufkraft und einer hinreichenden körperlichen Mobilität ausgegangen werden. Bei dieser Benutzergruppe kann zusätzlich eine eher geringe oder nicht vorhandene Ortskenntnis vorausgesetzt werden. Die Erfahrung im Umgang mit mobilen Endgeräten kann herkunfts-, bildungs- oder einkommensspezifisch stark variieren, so dass die Anwendung in ihrer Bedienung einfach gehalten werden sollte. Ein optimales System würde zudem den Nutzungskontext und die Benutzeranforderungen berücksichtigen.

Die möglichen Interaktionen des Benutzers (Akteur) mit der mobilen Anwendung (System) werden gesondert bei den funktionalen Anforderungen beschrieben.

### 4.1.2 Funktionale Anforderungen

Mit Hilfe der funktionalen Anforderungen sollen die möglichen Interaktionen des Systems mit der Umgebung (Akteur) aufgezeigt werden.
Für die Implementierung des Prototyps ist es sinnvoll, die Aufgabenstellung in Produzenten- und Konsumentensicht aufzuteilen. Dafür werden Use Case-Diagramme erstellt und voneinander gesondert betrachtet. Dabei muss das System für jede Aufgabenstellung die Anwendungsfälle erfüllen. So beinhaltet die Produzentensicht das Erstellen neuer POIs und deren Annotation von Multimediadaten. Die Konsumentensicht enthält dagegen das Abrufen ortsabhängiger POIs und ihre Darstellung auf einer digitalen 2D-Karte.
Eigentlich müssen neben den funktionalen Anforderungen analog auch die nichtfunktionalen Anforderungen des Systems untersucht werden. Dies kann jedoch aufgrund des damit verbundenen zeitlichen Aufwands nicht innerhalb der vorliegenden Arbeit erfolgen. So werden lediglich ausgewählte Aspekte im Bereich Interface Design untersucht.

### 4.1.3 Use Case

Bevor explizit auf die beiden eben genannten Aufgabenstellungen des Systems näher eingegangen werden kann, erscheint es notwendig, die Begriffe Use Case und System Use Case zu definieren, da aus UML- und Softwareentwicklungssicht der Systemanwendungsfall die normale Form eines Anwendungsfalles ist [42, S. 37].

Use Case:
*„Ein Anwendungsfall beschreibt anhand eines zusammenhängenden Arbeitsablaufs die Interaktionen mit einem (geschäftlichen oder technischen) System. Ein Anwendungsfall wird stets durch einen Akteur initiiert und führt gewöhnlich zu einem für die Akteure wahrnehmbaren Ergebnis.“* [42, S. 29]

System Use Case:

*„Ein Systemanwendungsfall ist ein Anwendungsfall, der speziell das für außen stehende Akteure (Benutzer oder Nachbarsysteme) wahrnehmbare Verhalten eines (Hard-/Software-)Systems beschreibt.“* [42, S.37]

#### 4.1.3.1 Anwendungsfälle aus Produzentensicht

Die vom System zu erfüllenden Anwendungsfälle aus Produzentensicht werden in Abbildung 21 angezeigt. Der Produzent kann entweder der Content Provider sein, der POIs über die integrierte Datenbank zur Verfügung stellt, oder der Nutzer selbst, indem er neue POIs innerhalb der Anwendung erzeugt.

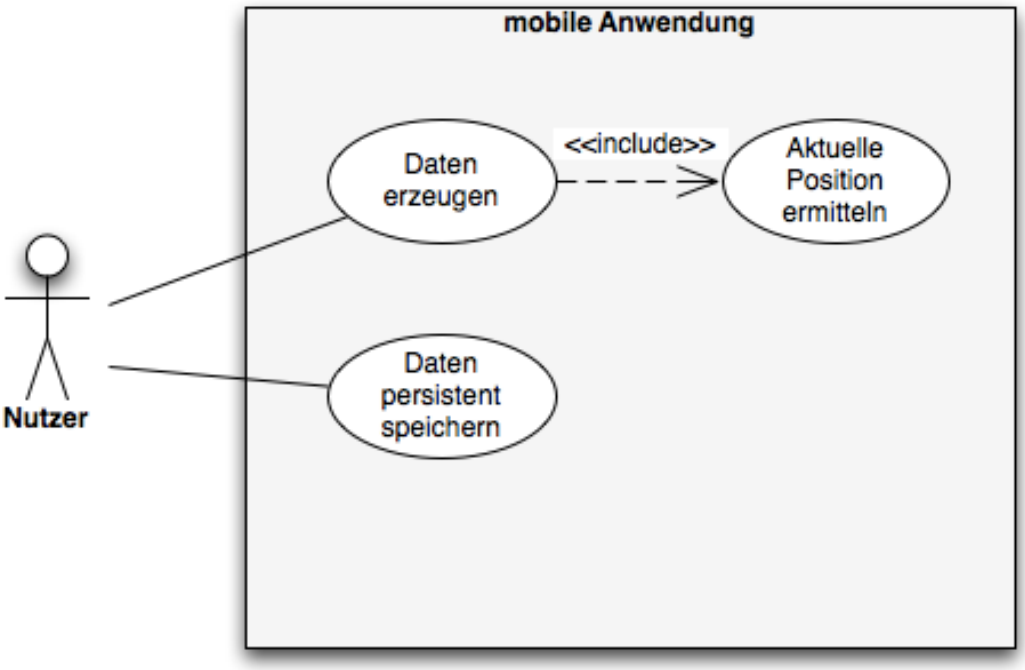


Abbildung 21: Anwendungsfalldiagramm – Produzentensicht

Die im Use Case-Diagramm „Produzentensicht“ enthaltenen Anwendungsfälle werden jeweils einzeln in Form von Anwendungsfallbeschreibungen aufgeführt. Sie enthalten Angaben über Aufgabe, Akteur, Auslöser sowie Vor- und Nachbedingung des jeweiligen Anwendungsfalls.

Um zu gewährleisten, dass die hier erzeugten Daten eindeutig dem aktuellen Standort zugeordnet werden können, muss im Vorfeld die Position ermittelt werden. Ohne Georeferenzierung (Geotagging) der Daten können keine ortsabhängige Informationen angezeigt werden.

| Name des Use Case | Daten erzeugen |
|---|---|
| Aufgabe | Zu einem POI sollen Textinformationen und Multimediadaten erzeugt werden. Um POIs nach Interesse filtern zu können, beinhalten die Textinformationen neben Namen und Beschreibung zusätzlich eine Kategorie. |
| Akteur | Nutzer |
| Auslöser | Nutzer mittels mobiler Anwendung |
| Vorbedingung | 1. Das mobile Endgerät muss mit einer Kamera ausgestattet sein.<br>2. Die Ansteuerung der Kamera wird von der Laufzeitumgebung unterstützt.<br>3. Die vorhandene Speicherkapazität muss ausreichend groß für die Aufnahme von Videos und Fotos sein. |
| Nachbedingung | Die beschreibenden Daten für einen POI wurden erstellt und stehen der mobilen Anwendung zur weiteren Verarbeitung zur Verfügung. |

Tabelle 5: Anwendungsfallbeschreibung „Daten erzeugen“

Durch die Positionsbestimmung steht der Anwendung dann ein vollständiger POI zur Verfügung. So beinhaltet ein vollständiger POI die Anwendungsszenarien „Daten erzeugen“ und „Aktuelle Position ermitteln“.

| Name des Use Case | Aktuelle Position ermitteln |
|---|---|
| Aufgabe | Ermittlung des eigenen Standorts und Geokodierung (Geotagging) der erzeugten Daten. |
| Akteur | Nutzer |
| Auslöser | Nutzer mittels mobiler Anwendung |
| Vorbedingung | 1. Erfolgreiche Ausführung des Use Case „Daten erzeugen“.<br>2. Ein Lokalisierungsverfahren muss der Anwendung ermöglichen die aktuelle Position zu bestimmen. |
| Nachbedingung | Der aktuelle Standort wurde bestimmt. Einem POI wurden Multimediadaten hinzugefügt und stehen der mobilen Anwendung zur Verfügung. |

Tabelle 6: Anwendungsfallbeschreibung „Aktuelle Position ermitteln“

Um sicherzustellten, dass beim Beenden der mobilen Anwendung keine Daten verloren gehen, müssen diese persistent gespeichert werden. Nur so können die erzeugten Daten zu einem späteren Zeitpunkt wieder abgerufen werden.

| Name des Use Case | Daten persistent speichern |
|---|---|
| Aufgabe | Ein zuvor erstellter POI wird lokal in einer SQLite-Datenbank persistent gespeichert. |
| Akteur | Nutzer |
| Auslöser | Nutzer mittels mobiler Anwendung |
| Vorbedingung | 1. Eine erfolgreiche Ausführung des Use Case „Aktuelle Position ermitteln“<br>2. Speicher und Zugriffsrechte müssen gegeben sein. |
| Nachbedingung | Die POI-Daten wurden lokal in der Datenbank gespeichert. Nach dem Speichern wird dem Nutzer eine Erfolgsmeldung auf dem Display angezeigt. |

Tabelle 7: Anwendungsfallbeschreibung „Daten persistent speichern“

Die Verwendung einer SQLite-Datenbank zur persistenten Datenspeicherung ermöglicht eine performantere Datenverwaltung als das Speichern in einer Property List auf dem lokalen Dateisystem der mobilen Anwendung. So liefert die Datenbank für Suchabfragen schneller Ergebnisse zurück und unterstützt zudem Sortiermechanismen. Die Performance bleibt selbst bei stark anwachsender Datenmenge weitestgehend stabil.

#### 4.1.3.2 Anwendungsfälle aus Konsumentensicht

In der Abbildung 22 werden die Anwendungsfälle aus Konsumentensicht dargestellt. Wie aus dem Diagramm zu entnehmen ist, sind die zwei beteiligten Akteure der Nutzer und ein Map Server.

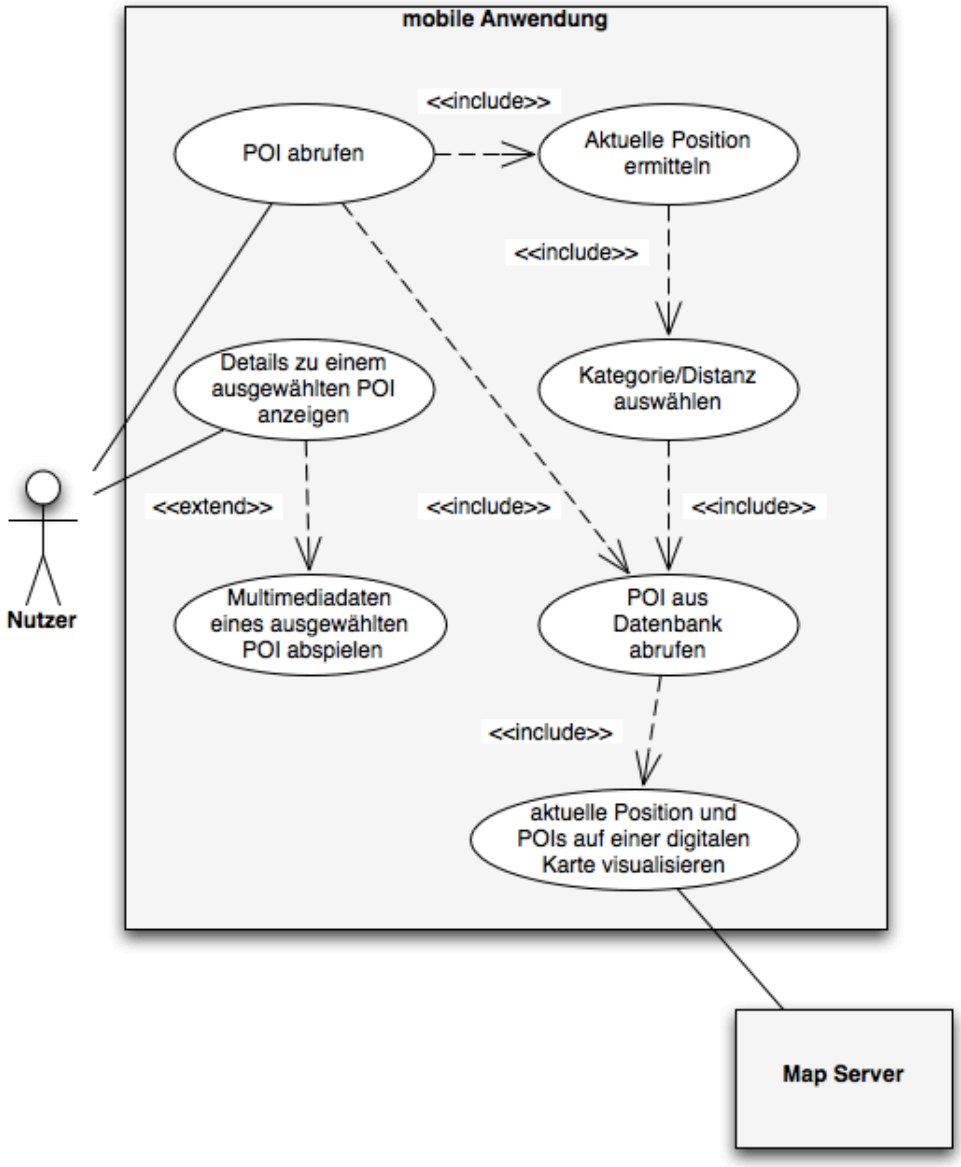


Abbildung 22: Anwendungsfalldiagramm – Konsumentensicht

Der Nutzer als Konsument möchte auf die bereitgestellten Informationen eines POI zugreifen können. Der Map Server stellt für die Visualisierung der POIs auf einer digitalen Karte das Kartenmaterial zur Verfügung. Der vom Nutzer initiierte Anwendungsfall „POI abrufen“ beinhaltet die auf der Abbildung 22 mit <<include>> gekennzeichneten Use Cases. Diese werden, wie bereits in der Produzentensicht, gesondert als Anwendungsfallbeschreibung aufgeführt. Der strukturelle Aufbau der Anwendungsfallbeschreibung ist analog zur Produzentensicht.
Damit der Nutzer ortsabhängig hinterlegte Informationen zu POIs abrufen kann, muss im Vorfeld seine aktuelle Position bestimmt werden. Da hier die gleiche Anforderung wie beim Erzeugen eines POI vorliegt, wird auf die Spezifikation „Aktuelle Position ermitteln“ verwiesen, um so eine unnötige Wiederholung zu vermeiden.

| **Name des Use Case** | Kategorie/Distanz auswählen |
|---|---|
| **Aufgabe** | Anzeigen der möglichen Kategorien und Distanzen zum Filtern der POIs. |
| **Akteur** | Nutzer |
| **Auslöser** | Nutzer mittels mobiler Anwendung |
| **Vorbedingung** | Erfolgreiche Ausführung des Use Case „Aktuelle Position ermitteln“ |
| **Nachbedingung** | Die ausgewählte Kategorie und Distanz stehen der mobilen Anwendung zur Verfügung. |

Tabelle 8: Anwendungsfallbeschreibung „Kategorie/Distanz auswählen“

Der Nutzer kann sich entweder sämtliche in der Datenbank verfügbaren POIs anzeigen lassen oder in gefilterter Form je nach Interesse (Kategorie) und Entfernung. So können durch die Filterfunktion nur die gewünschten und in unmittelbarer Umgebung befindlichen POIs angezeigt werden.

| **Name des Use Case** | POI aus Datenbank abrufen |
|---|---|
| **Aufgabe** | Der mobilen Anwendung werden anhand der ausgewählten Kategorie, der Distanz und der eigenen Position die von der lokalen Datenbank bereitgestellten POIs zur Verfügung gestellt. |
| **Akteur** | Nutzer |
| **Auslöser** | Nutzer mittels mobiler Anwendung |
| **Vorbedingung** | 1. Erfolgreiche Ausführung des Use Case „Kategorie/Distanz auswählen“<br>2. Filterung der Daten nach Kategorien<br>3. Umkreisberechnung für Distanzangaben und angegebene Position |
| **Nachbedingung** | Die gefilterten POIs stehen der mobilen Anwendung zur weiteren Bearbeitung zur Verfügung. |

Tabelle 9: Anwendungsfallbeschreibung „POI aus Datenbank abrufen“

Die aktuelle Position des Nutzers soll zur besseren Orientierung auf einer digitalen Karte angezeigt werden. Dabei werden der zuvor ermittelte Standort des Nutzers sowie die abgerufenen POIs mit Symbolen (Pins) auf der Karte markiert und dargestellt.

| Name des Use Case | aktuelle Position und POIs auf einer digitalen Karte visualisieren |
|---|---|
| Aufgabe | Zur Orientierung sollen auf einer digitalen Karte die aktuelle Position und die abgerufenen POIs angezeigt werden. |
| Akteur | Nutzer |
| Auslöser | Nutzer mittels mobiler Anwendung |
| Vorbedingung | 1. Erfolgreiche Ausführung des Use Case „POI aus Datenbank abrufen“<br>2. Der Datenbankzugriff muss vorhanden sein. |
| Nachbedingung | Dem Nutzer werden mittels Kartendarstellung die aktuelle Position und die zuvor abgerufenen POIs angezeigt. |

Tabelle 10: Anwendungsfallbeschreibung „Aktuelle Position und POIs auf einer digitalen Karte visualisieren“

Die Kartendarstellung mit den markierten POIs sollte es dem Nutzer ermöglichen, direkt Detailinformationen zu jedem einzelnen POI abzurufen. Dazu soll bei Berührung der Markierung (Pin) der Titel des POI eingeblendet werden.

| Name des Use Case | Details zu einem ausgewählten POI anzeigen |
|---|---|
| Aufgabe | Namen, Beschreibung und Bild eines ausgewählten POI anzeigen |
| Akteur | Nutzer |
| Auslöser | Nutzer mittels mobiler Anwendung |
| Vorbedingung | 1. Erfolgreiche Ausführung des Use Case „Aktuelle Position und POIs auf einer digitalen Karte visualisieren“<br>2. Mindestens ein POI muss vorhanden und aus der Liste heraus ausgewählt worden sein |
| Nachbedingung | Die Detailinformationen zu dem POI wurden dem Nutzer angezeigt. |

Tabelle 11: Anwendungsfallbeschreibung „Details zu einem ausgewählten POI anzeigen“

Bei Interesse kann der Nutzer dann auf die bereitgestellten Multimediainhalte zu dem ausgewählten POI zugreifen.

| Name des Use Case | Multimediadaten zu einem ausgewählten POI abspielen |
|---|---|
| Aufgabe | Abspielen des vom Produzenten erzeugten Videos und der Audiodateien zu einem POI |
| Akteur | Nutzer |
| Auslöser | Nutzer mittels mobiler Anwendung |
| Vorbedingung | 1. Erfolgreiche Ausführung des Use Case „Details zu einem ausgewählten POI anzeigen“<br>2. Das Abspielen der Multimediainhalte muss vom Smartphone unterstützt werden. |
| Nachbedingung | Die von der Datenbank zur Verfügung gestellten Videos und Audiodateien eines ausgewählten POI wurde dem Nutzer angezeigt bzw. vorgespielt. |

Tabelle 12: Anwendungsfallbeschreibung „Multimediadaten eines ausgewählten POI abspielen“

Die so beschriebenen Anwendungsfalldiagramme sollen beispielhaft aufzeigen, welche funktionalen Anforderungen für einen mobilen Stadtführer erforderlich sind. Die Analyse der funktionalen Anforderungen erhebt dabei keinen Anspruch auf Vollständigkeit.

Neben den bereits spezifizierten Funktionen der App könnten noch weitere Anforderungen definiert werden, so beispielsweise für Touren und Routen. Eine Tour besteht dabei aus mehreren POIs und den dazwischen liegenden Routen [16, S. 53]. Auf einer digitalen Karte wird eine Route als Kette von Wegpunkten dargestellt [16, S. 53]. Mit Hilfe einer Routenfunktion kann sich der Nutzer den kürzesten Weg zwischen Start- und Zielpunkt anzeigen lassen. Das dazu gehörende Anwendungsfalldiagramm „Route lokalisieren“ wird gesondert im Anhang angeführt. Ebenso befindet sich im Anhang das Anwendungsfalldiagramm „Tour lokalisieren“. Es beschreibt die notwendigen Interaktionen des Nutzers mit der mobilen Anwendung (System) zum Erstellen oder Auswählen einer Tour.

## 4.2 Technical Design

Die in iOS enthaltenen Technologien (Features) und Frameworks werden von Apple in verschiedene Schichten unterteilt [7, S.19]. Zur Veranschaulichung der iOS-Architektur dient folgende Abbildung:

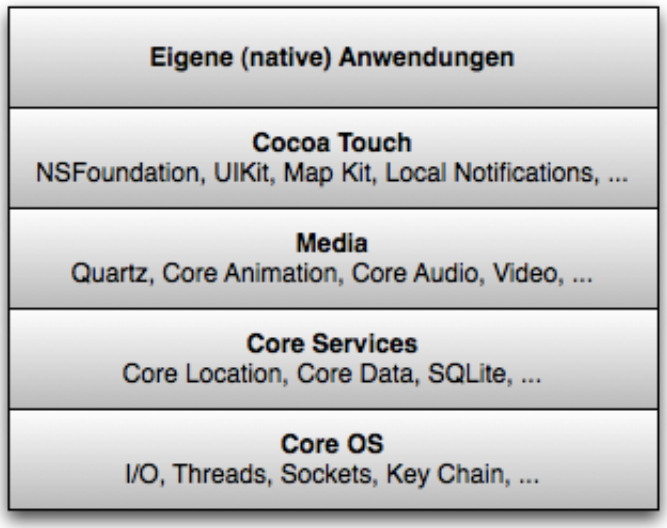


Abbildung 23: iOS-Architektur

Die Grundlage für alle Anwendungen bilden die Low-Level-Dienste der unteren Schichten. Die oberen Schichten enthalten dagegen objektorientierte Abstraktionen

für die Basisschichten. Obwohl dem Entwickler sämtliche Schichten zugänglich sind, empfiehlt Apple zunächst die oberen Schichten mit ihren gut getesteten Komfortfunktionen zu verwenden und somit einer Top-Down-Methode zu folgen [7, S. 20]. Im Folgenden werden die einzelnen Schichten kurz vorgestellt, wobei hier nur ausgewählte, für die prototypische Anwendung relevante Features und Frameworks beschrieben werden.

### 4.2.1 iOS-Architektur

Grundsätzlich besitzt jede App nur ein Fenster und läuft in ihrer eigenen Sandbox-Umgebung. Die Architektur einer iOS-Anwendung basiert auf den in Abbildung 23 dargestellten Schichten. So enthält die oberste Schicht immer die eigene Anwendung. Die darunterliegende Schicht Cocoa Touch beinhaltet die wichtigsten für die Entwicklung benötigten Funktionalitäten [7, S. 20]. Im Einzelnen sind dies folgende Technologien und Frameworks, die hier im Anschluss in tabellarischer Form kurz genannt werden.

| Technologien | Frameworks |
|---|---|
| Multitasking | Address Book UI Framework |
| Printing | Event Kit UI Framework |
| Data Protection | Game Kit Framework |
| Apple Push Notification Service | iAd Framework |
| Local Notifications | Map Kit Framework |
| Gesture Recognizers | Message UI Framework |
| File-Sharing Support | UIKit Framework |
| Peer-to-Peer Services | |
| Standard System View Controllers | |
| External Display Support | |

Tabelle 13: Technologien und Frameworks der Cocoa Touch-Schicht [43]

Von Relevanz für die prototypische Umsetzung sind dabei die Technologien Multitasking, Local Notifications, Gesture Recognizers und Standard System View Controllers. Die eben aufgezählten Features werden durch Einbinden des UIKit Framework zur Verfügung gestellt. Das UIKit Framework ist insbesondere für die grafische Darstellung der Applikation zuständig. Dies beinhaltet die Aspekte User Interface, Multi-Touch-Verhalten, Screen-Rotation, View Controller, Events sowie Gesten [7, S. 21]. Von besonderer Bedeutung für die Implementierung der mobilen Anwendung sind dabei User Interface und View Controller, so dass diese nach der

Beschreibung der einzelnen Schichten der iOS-Architektur gesondert vorgestellt werden.

Für eine gewünschte Kartendarstellung des aktuellen Standorts und der POIs innerhalb der mobilen Anwendung iSightseeing ist zusätzlich die Einbindung des Map Kit Framework notwendig.

Nach der Cocoa-Touch-Schicht folgt die Media-Schicht. Sie bietet Grafik-, Audio- und Videotechnologien an [7, S. 22]. Die damit verbundenen Technologien und Frameworks dienen zwar als Komponenten der Anwendung iSightseeing und sind für die Darstellung der multimedialen Inhalte verantwortlich, werden aber im Verlauf der Arbeit nicht näher besprochen, da der Fokus eher auf der Erstellung von POIs und Touren liegt.

Die darunterliegende Core-Services-Schicht beinhaltet grundlegende Systemservices. Für die vorliegende Arbeit sind insbesondere die beiden Frameworks Core Data und Core Location von Bedeutung. Sie werden gesondert unter dem Punkt „Ausgewählte Aspekte der iOS-Architekturschichten“ vorgestellt.

In der untersten Schicht Core OS finden sich Kernel, dazugehörige Treiber und grundlegende UNIX-Interfaces des Betriebssystems [7, S. 26]. Auf diese Schicht wird in der vorliegenden Arbeit nicht näher eingegangen, da für die Implementierung der Applikation kein direkter Zugriff auf sie erfolgen muss.

Bevor im Einzelnen auf die ausgewählten Aspekte der iOS-Architekturschichten eingegangen werden kann, erscheint es im Vorfeld sinnvoll, notwendige Entwurfsmuster kurz vorzustellen.

### 4.2.2 Cocoa Design Pattern

Entwurfsmuster (Design Patterns) als Lösungsschablonen für wiederkehrende Entwurfsprobleme in der Softwarearchitektur gelten als wichtige Bausteine für erfolgreiche Softwareprojekte. Ihre Verwendung kann bei der Lösung komplexer Probleme helfen. So ist das in Cocoa Touch enthaltene UIKit Framework speziell für die Verwendung einiger Patterns konzipiert [7, S. 27]. Die wichtigsten damit verbundenen Entwurfsmuster werden hier kurz vorgestellt.

#### 4.2.2.1 Model View Controller

Das Model View Design Pattern ist ein Entwurfsmuster für die Softwareentwicklung. Der Grundgedanke besteht darin, die Präsentationsschicht und Anwendungslogik einer Software zu trennen, indem der Code in verschiedene funktionale Bereiche aufgeteilt wird [7, S. 27]. So soll eine einfache Wart- und Erweiterbarkeit der Anwendung gewährleistet werden. Die folgende Abbildung zeigt die Zusammenhänge des MVC-Pattern. Sie wurde aus dem Cocoa Fundamentals Guide der Mac OS X Reference Library unverändert übernommen.

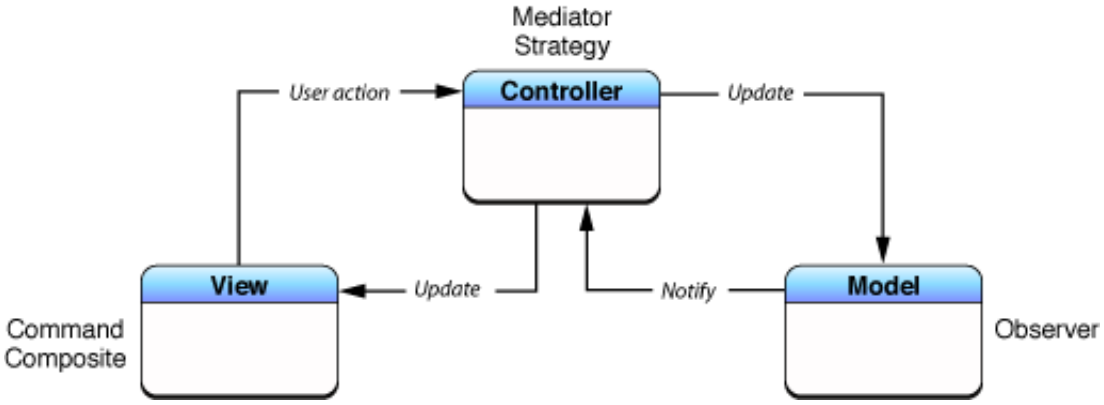


Abbildung 24: MVC-Pattern [44]

Wie in der Abbildung 24 dargestellt, wird die Anwendung in die Bereiche Model, View und Controller aufgeteilt. Dabei wird das zugrunde liegende Datenmodell durch das Model repräsentiert. Es soll die Funktionalität und die Daten der Anwendung kapseln. Für die Visualisierung der Daten und die Übernahme der Nutzereingaben beispielsweise über Buttons oder Textfelder ist der View zuständig. Eine direkte Kommunikation von Model und View ist nicht gegeben, stattdessen kommunizieren sie über den Controller. So liefert der Controller dem View die Daten aus dem Model und ändert die Daten im Model als Reaktion auf entsprechende Benutzeraktionen im View [7, S. 27]. Eine Verbindung von View und Controller kann bei einer iPhone-App über Outlets und Actions erfolgen.

#### 4.2.2.2 Delegation

Delegation ist ein Mechanismus, der, analog zur Vererbung, die Erweiterung der Funktionalität einer Klasse ermöglicht und somit als Alternative zur Erstellung von Subklassen betrachtet werden kann [7, S. 28]. Die hier zuzufügende Funktionalität

wird von einer weiteren Klasse, dem Delegate, realisiert. So erhält die zu erweiternde Klasse eine Instanzvariable für das Delegate und leitet Anfragen für die neue Funktionalität an das Delegate weiter [7, S. 28]. Innerhalb der iOS-Entwicklung ist das „ApplicationDelegate“, das Delegate des Applikation-Objektes, für die gesamte Applikationssteuerung verantwortlich [7, S. 28].

#### 4.2.2.3 Target Action

Das Target Action Pattern bezeichnet die Verbindung des Bedienelements mit der beim Betätigen des Bedienelements aufzurufenden Methode. So wird das Target als Objekt beim Auftreten des Ereignisses informiert und erhält die Aktion als Nachricht gesendet [7, S. 28].

### 4.2.3 Ausgewählte Aspekte der iOS-Architekturschichten

Wie einleitend erwähnt, werden hier kurz ausgewählte Aspekte der iOS-Architekturschichten vorgestellt, die für die Implementierung der Anwendung iSightseeing relevant sind.

#### 4.2.3.1 View Controller

View Controller bilden die Brücke zwischen View und Model, so dass sie Views mit dem Rest der Anwendung verbinden [7, S. 127]. Dabei lassen sich Views per Drag & Drop im Interface Builder erzeugen. Das Datenmodell kann ähnlich einfach mit Core Data erstellt werden, so dass aus dem MVC-Modell nur noch der Controller programmiert werden muss. Der Controller hat Zugriff auf das Model und informiert den View in welcher Form die Daten dargestellt werden sollen. Zusätzlich ist der View Controller zuständig für das Ausführen von Aktionen, die beim Klicken eines Bedienelements ausgelöst werden.

Das UIKit unterscheidet drei grundlegende Arten von View Controllern. Diese sind Custom View Controller, Container View Controller sowie Modal View Controller. Zu den Custom View Controllern gehören Table View Controller, die für die Darstellung von Listen (einspaltige Tabellen) bestimmt sind. Die Bildung von Subklassen ist für diesen Typ erlaubt. Für die hierarchische Strukturierung einer

Anwendung verwendet man entweder den Navigation Controller oder den Tab Bar Controller, die beide Vertreter des Container View Controller-Typs sind. Eine Erweiterung durch Subklassen ist hier nicht erlaubt. Sowohl Navigation Controller als auch Tab Bar Controller können mehrere Views oder View Controller kontrollieren [7, S. 173]. Der Modal View Controller bezeichnet View Controller, die animiert eingeblendet werden.

Für die Implementierung sind zwar alle genannten View Controller relevant, jedoch ist der Tab Bar Controller insbesondere für die Umsetzung des Navigationskonzeptes entscheidend. Zur Veranschaulichung des Aufbaus eines Tab Bar Controller dient die folgende Abbildung, die unreflektiert aus dem View Controller Programming Guide for iOS aus der iOS Reference Library übernommen wurde:

Abbildung 25: Aufbau des Tab Bar Controller [28]

Wie aus der Abbildung 25 zu entnehmen ist, setzt sich die Anzeige (View) des Tab Bar Controller aus Tab Bar und Content-Bereich zusammen. Die Tab Bar Items dienen dabei zur funktionalen Gliederung der Anwendung.

#### 4.2.3.2 Map Kit

Das Map Kit Framework ermöglicht den bequemen Umgang mit Kartendarstellungen, indem es ein Interface zum Einbinden von Maps in Windows und Views anbietet. Dabei unterstützt es die Darstellung von Annotations (Markierungen von POIs mit Kommentaren) und des aktuellen Aufenthaltsortes [7,

S. 207]. Zusätzlich besteht eine Funktion zum Reverse Geocoding, welche die Adresse zur einer bestimmten Koordinate ermittelt. Das Map Kit verwendet zur Kartendarstellung die Mercator-Projektion, eine winkelkorrigierte Darstellung der Erdoberfläche.

#### 4.2.3.3 Core Data

Das Persistenz-Framework Core Data ermöglicht den vereinfachten objektorientierten Zugriff auf die relationale Datenbank SQLite, welche für die lokale Datenspeicherung verwendet werden kann [7, S. 113]. Das Repository kümmert sich dabei sowohl um das Abspeichern des Objektgraphen als auch um die Bereitstellung der benötigten Datenobjekte. Im MVC-Pattern stellt Core Data die Model-Schicht dar.

| Core Data | SQLite | Objective-C |
|---|---|---|
| Entität | Tabelle | Klasse |
| Attribut | Spalte | Eigenschaft dieser Klasse |

Tabelle 14: Umwandlung des relationalen in ein objektorientiertes Datenbankmodell

Die Tabelle 14 soll verdeutlichen, dass mit Hilfe von Core Data die Abbildung des relationalen Datenbankmodells in einem objektorientierten Modell und umgekehrt möglich ist. So entspricht die Entität in Core Data einer Tabelle in SQLite sowie das Attribut einer Spalte [7, S. 115]. Daraus lassen sich dann Klassen und Eigenschaft dieser Klasse in Objective-C ableiten beziehungsweise darstellen.

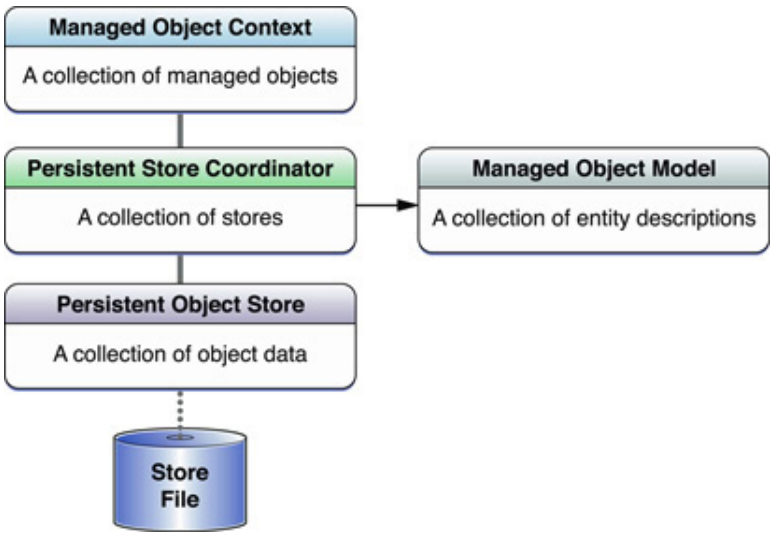


Abbildung 26: Aufbau der Zugriffsschicht Core Data Stack

Abbildung 26 zeigt den schematischen Aufbau der Zugriffsschicht für die Persistenz, den Core Data Stack. Er umfasst die Klassen NSManagedObjectContext, NSManagedObjectModel und NSPersistentStoreCoordinator [7, S. 119]. Dabei repräsentiert Managed Object Model das Datenmodell. Der Persistent Store Coordinator ist zuständig für die Verbindung zwischen Model und Datenbank, also für die Speicherung der Daten. Die oberste Schicht, Managed Object Context, stellt die Verbindung zwischen der Persistenzschicht und dem Anwendungscode dar [7, S. 122].

#### 4.2.3.4 Core Location

Mit Hilfe des Core Location Framework lassen sich sowohl der aktuelle Standort als auch die Bewegungsrichtung ermitteln. Die Positionsbestimmung ist, wie bereits in Kapitel 2 erläutert, über die drei verschiedenen Wege GPS, Mobilfunk- und WLAN-Ortung möglich [7, S. 195]. Dabei ist die Positionsermittlung unabhängig von der Location-Ermittlung des Map Kits. Der Anwender muss erst der Nutzung der Location Services durch eine Applikation zustimmen. Die Core Location API als Schnittstelle für die Ortsbestimmung besteht im wesentlichen aus den drei Klassen `CLLocationManager`, `CLLocationManagerDelegate` und `CLLocation` [7, S. 196].

## 4.3 Interface Design

Im Folgenden wird die Gestaltung des User Interface für einen mobilen Stadtführer beschrieben. Der Begriff User Interface bezeichnet die grafische Benutzeroberfläche, die dem Benutzer die Interaktion mit der mobilen Anwendung über grafische Symbole erlaubt. Für die Gestaltung des User Interface für mobile iOS-Endgeräte stellt Apple eine Richtlinie, die Human Interface Guidelines, zur Verfügung. Sie dient zur Orientierung, wie iOS-Applikation gestaltet werden sollten. So werden Empfehlungen für die Gestaltung des Icons und des User Interface der Anwendung gegeben. Basierend auf diesen Richtlinien wurde sowohl das App-Icon als auch das User Interface des zu implementierenden Prototypen gestaltet. Diese werden nun im Einzelnen vorgestellt.

### 4.3.1 App-Icon – iSightseeing

Für die Gestaltung des Icons wurde am 16. Juni 2010 ein Designcontest auf der australischen Design-Plattform „99designs“ gestartet. Auf dieser Plattform werden Aufträge an Grafikdesigner weltweit über Designwettbewerbe vergeben. Sie basieren auf dem Prinzip des Crowdsourcing, bei dem der Auftraggeber im Vorfeld bestimmen muss, für welche Kategorie er einen Designentwurf benötigt, und die Anforderungen in Form eines Briefings definieren muss. Für jede Designkategorie existieren Preisvorschläge, die eine Mindestanzahl an teilnehmenden Grafikern garantieren; sie sind aber nicht bindend.
Für den öffentlichen Designwettbewerb mit dem Titel „iPhone App Icon Design - iSightseeing“ wurden folgende Anforderungen gestellt:

| | |
|---|---|
| **Brief Overview** | This iPhone app is a location based multimedia city guide for Berlin (capital of germany). You can play Audio and/or Video information for every point of interest. The symbol should have a close connetion to an app like "Path Tracker". It should provide rather an illustration than a picture of a Berlin landmark. |
| **Target Audience** | Tourists; Berlin visitors |
| **Requirements** | Illustrator .ai format (vector graphic) or Photoshop .psd, 57 x 57 px, 163 dpi or higher; keep layers in final handout |

Insgesamt wurden 42 verschiedene Designvorschläge für den ausgeschriebenen Designcontest eingereicht. Der Wettbewerb wurde über Soziale Netzwerke kommuniziert, so dass die Meinung Dritter in die Entscheidung für die Designauswahl mit eingeflossen ist. Auf die Gestaltung des Icons wurde mit Hilfe von Bewertungen und Kommentaren zu den eingereichten Entwürfen Einfluss genommen. Ausschlaggebend für die Wahl des Wettbewerbssiegers war das ansprechende Design und die vorhandene Assoziation mit einem mobilen multimedialen Stadtführer. Das Icon ist universell für mobile Stadtführer gestaltet und wäre somit auch für andere Städte verwendbar. Es folgt eine bildliche Darstellung des freigestellten App-Icons. Zusätzlich ist ein Screenshot des iPhone-Homescreen angefügt, um so die Wirkung des Icons in direkter Nachbarschaft mit anderen Icons zu demonstrieren.

Abbildung 27: App-Icon – iSightseeing

Der Hintergrund des Icons besteht aus einer vereinfachten zweidimensionalen Kartendarstellung. Der rote Pin als Markierung für ein POI dominiert das Icon. Durch Verwendung der allgemein gültigen Symbole für Audio (Kopfhörer) und Video (Filmstreifen) wird auf die multimediale Funktionalität der iPhone-Anwendung hingewiesen. In der Darstellung des Homescreen ist gut zu erkennen, dass der rote Pin in der 4. Reihe von oben markant auffällt und somit die Aufmerksamkeit des Betrachters auf sich ziehen kann.

### 4.3.2 User Interface – iSightseeing

Im Folgenden werden Design und Konzeption des User Interface des mobilen Stadtführers vorgestellt. Dabei wird das Design des User Interface maßgeblich durch die gegebene Displaygröße des iPhone bestimmt [11, S. 213]. Für die vorliegende Arbeit wird das Design des User Interface für den Portraitmodus des iPhone 3GS (480x320px) optimiert. Die verbleibende Höhe für den Content-Bereich ergibt sich nach Abzug der Statusleiste (20px), der Navigation Bar (44px) und der Tab Bar (48px) [11, S. 213 - 215]. So stehen lediglich 368x320px für die Gestaltung des Content-Bereichs zur Verfügung.

Zum Verständnis des User Interface erscheint es notwendig, die Funktionalitäten der Anwendung zu beschreiben, um so das dahinterliegende Navigationskonzept

verstehen zu können. Der grafische Aspekt spielt zwar bei der prototypischen Umsetzung eine untergeordnete, für das Endprodukt jedoch eine sehr wichtige Rolle, trotzdem wird er in diesem Zusammenhang nicht berücksichtigt. Wesentlicher erscheint eine chronologische Darstellung der einzelnen funktionalen Bereiche Places, My Tour, Search, Guided Tour und Map.

Beim ersten Start einer iPhone-App besteht die Möglichkeit, einen Startscreen einblenden zu lassen. Dieser sollte eigentlich nach den Richtlinien der Human Interface Guidelines den Hintergrund der Anwendung anzeigen, um damit dem Benutzer die Wartezeit beim Startvorgang der App kürzer erscheinen zu lassen. Solange sich die Anwendung jedoch im Speicher befindet, wird beim erneuten Zugriff auf die Anwendung der Startscreen nicht mehr anzeigt.

Für den Prototypen wird ein selbsterstellter Startscreen verwendet mit der Absicht, dem Anwender einen visuellen Hinweis auf den enthaltenen Content der iPhone-Anwendung zu geben (siehe Abbildung 28).

Abbildung 28: Startscreen – iSightseeing

Wie in der oberen Abbildung zu sehen ist, beinhaltet der Startscreen die wesentlichen Elemente des App-Icons (Pin, Video- und Audiomarker). Um einen Bezug zur Stadt Berlin zu erzeugen, wurden der Umriss der Stadt sowie die Silhouetten der wichtigsten Sehenswürdigkeiten (Siegessäule, Reichstag, Fernsehturm und Brandenburger Tor) in den Hintergrund integriert.

Nach dem Start der Applikation erscheint grundsätzlich der Root View des Tab Bar Controller und das erste Tab Bar Item „Places“ wird dabei farblich unterlegt und

somit hervorgehoben. Zur Veranschaulichung der funktionalen Gliederung der Applikation folgt die Darstellung der implementierten Tab Bar:

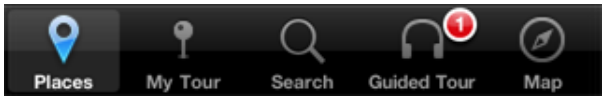


Abbildung 29: Tab Bar – iSightseeing

Der Root View des Tab Bar Controller enthält die Listendarstellung aller in der Datenbank enthaltenen POIs (siehe Abbildung 30, Screenshot 1).

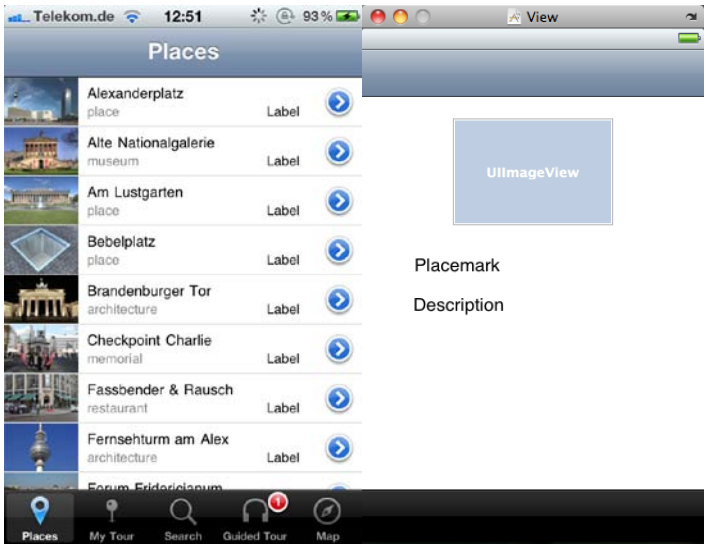


Abbildung 30: Screenshots – Places

Die Darstellung der Tabellenzelle enthält verschiedene Angaben zum POI, das heißt Name, Kategorie und Entfernung (hier noch Label) vom aktuellen Standort des Benutzers. Über den jeweiligen Disclosure-Button gelangt man zur Detailansicht des POI. Diese enthält neben dem Foto und dem Namen des POI dazugehörige Textinformationen (siehe Abbildung 30, Screenshot 2, Interface Builder .xib Datei). Das Hinzufügen beziehungsweise Bearbeiten eines POI sollte über die Buttons „+“ und „Edit“ möglich sein. Dabei soll die Ansicht zum nächsten View (Geotagging) wechseln. Hier soll es möglich sein, Angaben zum POI zu machen und die Georeferenzierung der Daten vorzunehmen. Zudem wird die Funktion angeboten, Fotos über die integrierte Kamera des iPhone aufzunehmen oder wahlweise aus der eigenen Library auszuwählen, um sie so dem POI hinzufügen zu können.

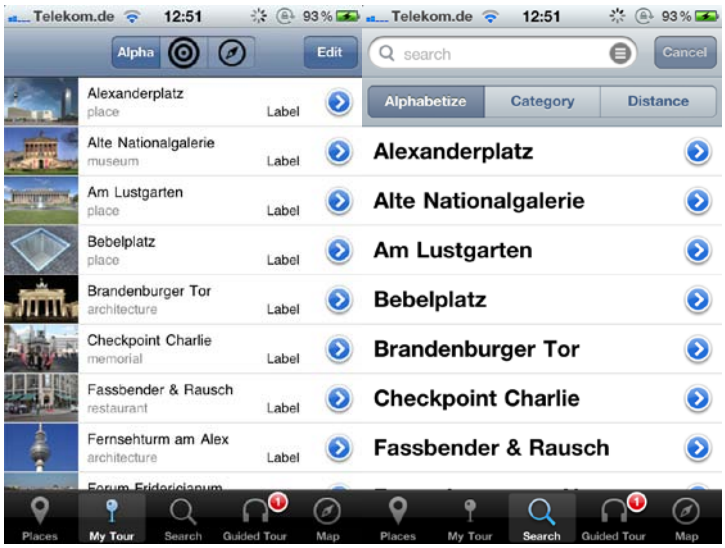


Abbildung 31: Screenshots – My Tour und Search

Abbildung 31 zeigt die Funktionalitäten „My Tour“ und „Search“ an. Wie aus dem ersten Screenshot zu entnehmen ist, enthält die Ansicht von My Tour zunächst alle der in der Datenbank enthaltenen POIs. Über die Edit-Ansicht können die gewünschten POI ausgewählt und manuell angeordnet werden. Die Liste kann entweder alphabetisch oder nach Distanz zum aktuellen Standort über die jeweiligen Buttons sortiert werden (Abbildung 31, Screenshot 1). Hier führt der Disclosure-Button ebenfalls zur Detailansicht des POI (Abbildung 30, Screenshot 2). Über das Map-Symbol in der Navigation Bar gelangt man zur Kartendarstellung von My Tour. Diese enthält alle ausgewählten POIs. Der Search View ermöglicht eine schnelle und vereinfachte Suche nach POIs (Abbildung 31, Screenshot 2). Das Suchergebnis kann dabei nach den Suchfiltern Alphabet, Kategorie oder Entfernung sortiert werden.

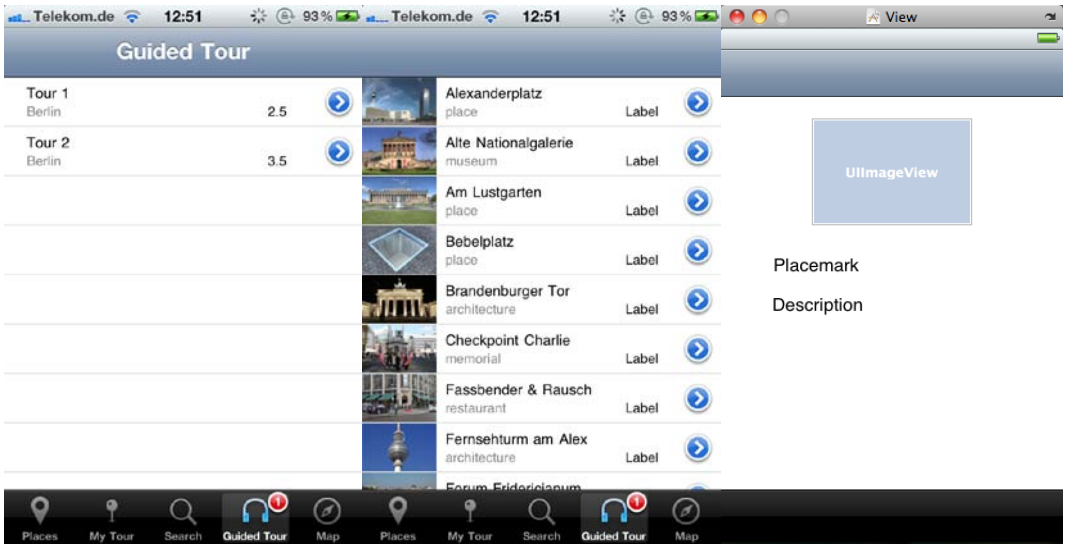


Abbildung 32: Screenshots – Guided Tour

In Abbildung 32 wird die Funktionalität „Guided Tour“ dargestellt. Der erste View enthält eine Liste der angebotenen geführten Touren. Die Tabellenzelle enthält Angaben über den Namen und die Dauer einer Tour sowie eine kurze Beschreibung der Strecke. Der Disclosure-Button führt zur POI-Listenansicht der vordefinierten und nicht veränderbaren Tour (siehe Abbildung 32, Screenshot 2). Damit nicht unnötig die Batterie belastet wird, soll ein Start-Button integriert werden, der das Location-Tracking startet und gleichzeitig die Genauigkeit der Ortung und den Distanzfilter festlegt. Die Detailansicht eines POI der Guided Tour enthält im Gegensatz zur normalen Detailansicht eines POI zusätzliche Multimediainhalte in Form von Video- und Audiodateien.
Die verschiedenen Kartendarstellungen der Anwendung werden in der Abbildung 33 verdeutlicht.

Abbildung 33: Screenshots – Map

Vorgesehen ist eine Kartenansicht, die alle vorhandenen POIs anzeigt. Wahlweise sollte aber auch die eigene aktuelle Position und die eventuell in der näheren Umgebung befindlichen POIs angezeigt werden. Ebenso soll eine Routendarstellung der jeweiligen Tour möglich sein. Der zweite Screenshot der Abbildung 33 ist die Kartendarstellung für einen einzelnen POI aus der Guided Tour. Der letzte Screenshot zeigt im Vergleich einen einzelnen POI aus My Tour an. Der Unterschied besteht in der Belegung der Funktionalität des noch zu integrierenden Disclosure-Buttons der Pin-Annotation. Dieser soll bei der Guided Tour direkt zum Video führen, während bei My Tour die Detailansicht aufgerufen werden soll.

## 4.4 Implementierung

In den folgenden Abschnitten wird auf die prototypische Implementierung eingegangen. Basis der Implementierung ist die persistente Speicherung der Daten in einer SQLite-Datenbank, so dass das zugrundeliegende Datenmodell im Vorfeld zunächst vorgestellt werden muss. Da der Fokus dieser Arbeit in der Realisierung eines mobilen Stadtführers liegt, werden die Umsetzungen zum Erstellen eines POI und zum Zusammenstellen einer Tour präsentiert. Von Bedeutung ist auch die Funktion zur Generierung von interaktiven Kartenausschnitten. Diese enthalten die Darstellung von POIs und des aktuellen Benutzerstandorts. Die genannten Umsetzungen werden in der vorliegenden Arbeit werden nicht in Form von Codefragmenten dokumentiert, sondern bei der Präsentation vorgestellt.

### 4.4.1 Datenmodell

Der Implementierung liegt das in Abbildung 33 dargestellte Datenmodell zugrunde. Es weicht vom erwarteten Datenmodell einer relationalen Datenbank ab, da die sonst übliche Normalisierung beim Datenbankentwurf für Core Data maximal bis zur 2. Normalform angewendet werden sollte [45, S. 96]. Beim Modellieren müsste weiterhin berücksichtigt werden, dass ein NSFetchedResultsController jeweils nur eine Entität abfragen kann [45, S. 189], so dass 1:1 Beziehungen von Entitäten daraufhin überprüft werden sollten, ob die Attribute nicht auch in der primären Entität untergebracht werden könnten. Im Core Data Programmierung Guide empfiehlt Apple für jede Relation eine Inverse-Relation zu definieren, um so die Konsistenz der Objektgraphen zu wahren [7, S. 116]. Es empfiehlt sich, das Datenmodell dahingehend zu optimieren, dass für binäre Daten Core Data Faults verwendet werden können [45, S. 93 - 95]. Diese Technik ermöglicht es Objekte nur bei Bedarf, das heißt beim Zugriff, in den Speicher zu laden. Bei dieser Technik gelten daher folgende Regeln für den Entwurf der Datenbank:

- Small Binary Data < 100kB (Icons, Thumbnails) sollten als Property in der korrespondierenden Entität mit dem Attribute-Typ Transformable abgelegt werden
- Medium Binary Data > 100kB < 1MB (Bilder, kleine Audiodateien) sollten als eigenständige Entität definiert werden

- Large Binary Data > 1MB (große Bilder, Audio- und Videodateien) sollten als Pfad in der primären Entität angelegt werden

Diese Anforderungen führten schließlich zum vorliegenden Datenmodell.

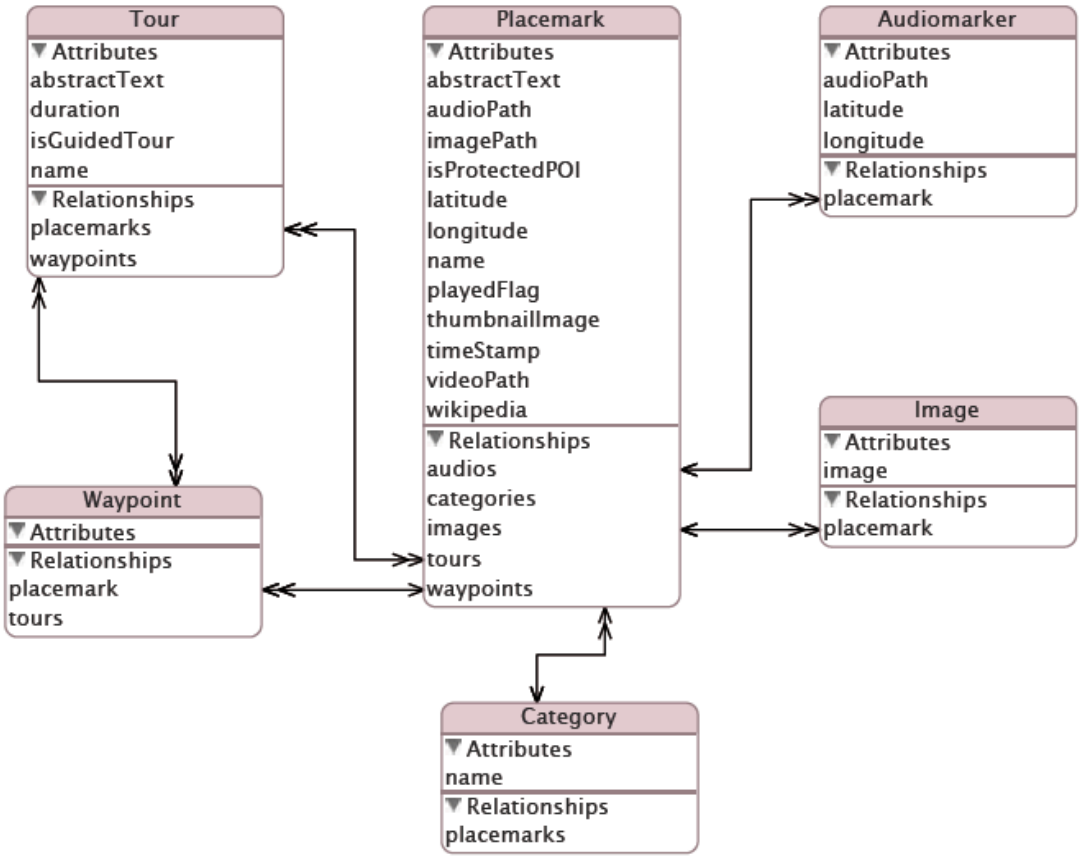


Abbildung 34: Datenmodell – iSightseeing

Wie aus Abbildung 34 zu entnehmen ist, enthält die Entität Placemark alle Informationen zu einem POI. Tour und Placemark stehen in einer n:n Beziehung, da einerseits verschiedene Touren gleiche POIs enthalten können und andererseits ein POI zu mehreren Touren gehören kann.

### 4.4.2 POI erstellen

Beim Erstellen eines POI wird automatisch die aktuelle Position ermittelt und als Koordinate (Latitude und Longitude) in der Datenbank gespeichert. Der Nutzer kann den POI mit weiteren Daten komplettieren.

### 4.4.3 Tour erstellen

Der Nutzer kann die Reihenfolge der POIs manuell anordnen und durch Auswahl einen POI hinzufügen oder entfernen. Dies gilt nur für den Bereich My Tour, da eine Guided Tour unveränderbar sein soll.

### 4.4.4 Kartendarstellung mit Map Kit

Ein POI wird als Piktogramm (Pin) dargestellt und anhand seiner geografischen Koordinaten als interaktives Element auf die Karte gelegt (Overlay). Eine farbliche Unterscheidung der Pins für unterschiedliche POI-Kategorien ist nicht vorgesehen. Über den Disclosure-Button der Pin-Annotation ist der Zugriff auf die Detailansicht (My Tour) beziehungsweise auf das Video (Guided Tour) möglich.

### 4.4.5 Local Notification

Local Notifications werden als Alert auf dem Display angezeigt, sobald sich ein POI in der näheren Umgebung befindet.

### 4.4.6 Background Location

Für den Background Location Service wird die Lokalisierung von GPS auf Funkmastortung umgestellt. Dabei reagiert der Dienst nur auf signifikante Ortsänderungen (circa 500m). Der mobile Stadtführer wurde jedoch speziell für Spaziergänge konzipiert, die dicht aufeinanderfolgende POIs enthalten können, so dass sich die Frage stellt, ob und inwieweit sich dieser Dienst für diese Nutzung eignet. Diese Fragestellung muss noch weiter verfolgt werden.

# 5 Tests und Evaluation

Ursprünglich war vorgesehen, die Bedienbarkeit des Prototypens mit Hilfe verschiedener Probanden aus der zuvor definierten Zielgruppe ausländischer Berlintouristen zu testen. Dieser Ansatz erwies sich in der Umsetzung als problematisch, da verschiedene Faktoren dies blockierten. Das zuvor angedachte Testszenario sollte eigentlich mit einer ausländischen Reisegruppe der Firma globalguides.net durchgeführt werden. Da Herr Steenbeek über die entsprechenden mobilen Endgeräte verfügt und als Stadtführer die notwendigen Kenntnisse besitzt die Korrektheit der Daten zu überprüfen. Das vorgesehene Testszenario konnte jedoch nicht realisiert werden, da der Firmeninhaber aus persönlichen Gründen sämtliche Touren storniert hat, so dass nach einer alternativen Lösung gesucht werden musste.

Die Idee unbekannte ausländische Berlintouristen für den Testlauf zu gewinnen, musste wegen verschiedener Faktoren wieder fallen gelassen werden. Zum einen bestand im November 2010 eine akute Sicherheitswarnung für öffentliche Plätze und Menschenansammlungen in Berlin, so dass ein generelles Misstrauen gegenüber Fremden vorherrschte. Zum anderen war aufgrund der aktuellen Witterungsverhältnisse[5] niemand bereit, unter diesen erschwerten Bedingungen als Proband zur Verfügung zu stehen. Zusätzliche Hindernisse waren technische Anforderungen für den Test. Entweder müsste ein Testgerät zur Verfügung gestellt oder ein Mittel gefunden werden, um die Applikation vor Ort auf das iPhone zu übertragen. Beide Varianten scheiden aus Sicherheitsbedenken aus. Der Zeitfaktor drängte zu einer Lösung, und so wurden zwei Probanden aus dem beruflichen Umfeld gefunden, die zum einen Besitzer eines iPhone und zum anderen als freiberufliche Stadtrundführer tätig sind. Aufgrund ihrer Tätigkeit besitzen sie die notwendigen Kenntnisse, um die Inhalte und ihre Darstellung innerhalb der Applikation auf Korrektheit zu überprüfen. Die Probanden unterscheiden sich in ihrer Erfahrung im Umgang mit dem iPhone und verfügen zudem über verschiedene Gerätegenerationen. So stehen ein iPhone 3GS und iPhone 4 als Testgeräte zur Verfügung.

[5] Im Dezember herrschte in Berlin Schneechaos mit arktischer Kälte

## 5.1 Testaufbau

Im Vordergrund der Untersuchung steht die Überprüfung folgender Funktionalitäten:

- POI erstellen
- Tour erstellen
- multimediale Guided Tour erproben

So hatte jeder Proband die beiden Aufgaben, eigenständig ohne Hilfestellung einen neuen POI zu erstellen und diesen in My Tour zu integrieren. Die dritte Aufgabe bestand darin, die Wiedergabefunktion der Guided Tour zu prüfen.

## 5.2 Qualitative Bewertung

Eine qualitative Auswertung des Usability-Tests des Prototypen kann unter diesen Umständen hier nicht erfolgen, da sowohl die Anzahl der teilnehmenden Probanden als auch der Testaufbau selbst dem Anspruch einer wissenschaftlichen Untersuchung nicht gerecht werden. Beim Testen der Anwendung sind verschiedene Fehler aufgetreten, so dass die Testläufe abgebrochen werden mussten. Dennoch werden die im Verlauf der durchgeführten Testläufe gewonnenen Erkenntnisse nicht komplett verworfen, sondern fließen in das Fazit und den Ausblick der vorliegenden Arbeit mit ein.

# 6 Fazit und Ausblick

Die prototypische Umsetzung des mobilen Stadtführers hat wesentlich mehr Zeit in Anspruch genommen als erwartet, so dass nicht alle zu Beginn der Arbeit formulierten Funktionalitäten umgesetzt werden konnten. Die Gründe dafür sind vielseitig. So erwies sich beispielsweise die Umstellung vom vertrauten relationalen Datenbankmodell auf das Core Data Datenmodell schwieriger als angenommen. Insbesondere ein Zusammenspiel von Tab Bar Controller, Navigation Controller, Core Data und NSFetchedResultsControllerDelegate zu erreichen bereitete mehr Probleme, als im Vorfeld absehbar war. Die Verwendung des Delegate verursachte permanent Abstürze beim Programmstart. Zusätzlich sind beim Entwickeln der Applikation Probleme durch den Wechsel der Entwicklungsumgebung von iOS SDK 4.1 auf Version 4.2 entstanden, so dass die zentrale main.xib Datei zu Gänze beschädigt war und das ganze Projekt neu angelegt werden musste. Dieses Problem (Assertion Failure) beim Öffnen der main.xib mit dem Interface Builder hat sich seit der Umstellung mehrfach wiederholt und die Ursache konnte trotz intensiver Recherche im Internet nicht gefunden und somit behoben werden. Für die Erstellung des User Interface mit dem Interface Builder war es notwendig, auf einem zweiten Rechner die Vorgängerversion 4.1 des iOS SDK zu installieren, da mit dieser Entwicklungsumgebung die main.xib sich problemlos bearbeitet lies.
Aus heutiger Sicht ist die Entscheidung für Core Data und gegen die SQLite API, trotz der genannten Schwierigkeiten, richtig gewesen, da durch die Verwendung von Core Data der Programmieraufwand für den Zugriff auf das Datenmodell entfällt.
Neben den genannten technischen Problemen kam noch die zeitliche Beschränkung zur Fertigstellung der Arbeit hinzu, so dass in vielen Bereichen Abstriche gemacht werden mussten. So konnte der angedachte Usability-Test nicht wie geplant durchgeführt werden. Er stellt somit einen offenen Punkt der Arbeit dar, der zu einem späteren Zeitpunkt nachgeholt werden könnte.

## 6.1 Offene Punkte

Im Folgenden werden Erweiterungen beschrieben, die in den Prototypen aufgrund des Zeitmangels nicht implementiert werden konnten. Dazu zählt etwa die wikipedia-Anbindung per API für die ausführliche Beschreibung einzelner POIs. Die angestrebte Integration eines Bewertungsportals für POIs konnte ebenfalls nicht

umgesetzt werden. Dies gilt auch für die Anbindung sozialer Netzwerke. Die Probleme bei der Darstellung der POIs sowohl in der Detail- als auch in der Kartenansicht müssen noch behoben werden. Die Schwierigkeit besteht darin die Detailansichten per pushViewController des Navigation Controller fehlerfrei aufzurufen.

## 6.2 Zukünftige Arbeiten

Der vorliegende Prototyp beinhaltet in seinem gegenwärtigen Entwicklungsstand nur die Darstellung von POIs, so dass eine Erweiterung zur Darstellung von Ereignissen (Events) denkbar wäre. Dazu müssten XML Feeds aggregiert abgerufen und mit Hilfe eines XML-Parsers aufbereitet werden. Eine vorherige Anpassung des Datenmodells wäre dazu notwendig. Ein zusätzlicher Mehrwert könnte auch durch Funktionalitäten zum Im- und Export von Touren im XML oder KML-Format erzielt werden. Der Personenbezug könnte eine stärkere Betonung erfahren, indem Funktionalitäten zur Filterung der angezeigten Ereignisse und POIs nach Alter und Interesse integriert würden. Dadurch könnte der mobile Stadtführer zielgruppengerecht POIs und Events anzeigen. Neue Touren und Routen würden sich zusätzlich durch eine Berücksichtigung weiterer Fortbewegungsmittel (Fahrrad, PKW oder öffentliche Verkehrsmittel) ergeben. Interessant dafür erscheint die Verwendung von Cloudmade anstelle von Google Maps, da das auf OpenStreetMap basierende CloudMade-Framework generell verschiedene Fortbewegungsmittel sowie Forward Geocoding unterstützt.

Des Weiteren könnte die Anwendung um die Funktionalität zur Verwaltung einer Favoritenliste erweitert werden. Dies würde eine Anpassung des Datenmodells voraussetzen.

# 7 Literaturverzeichnis

## Bücher


[1] Ali, M.: iPhone SDK Programming. Chichester (UK): Wiley, 2009

[2] Bennett, G.; Peterson, S.; Zarra, Marcus S.: iPhone Cool Projects. Apress, New York (USA): Springer-Verlag, 2009

[3] Clark, J.: Tapworthy: Designing Great iPhone Apps. Sebastapol (USA): O'Reilly Media, 2010

[4] Dudney, B.; Adamson, C.: Entwickeln mit dem iPhone SDK. Köln: O'Reilly Verlag, 2010

[5] Hughes, J.: iPhone and iPad Apps Marketing: Secrets to Selling Your iPhone and iPad Apps. Que Publishing, 2010

[6] Kochan, Stephen G.: Programming in Objective-C 2.0. 2nd Ed. Addison-Wesley Verlag, 2009

[7] Dr. Koller, D.: iPhone-Apps entwickeln. Poing: Franzis Verlag GmbH, 2010

[8] Dr. Lewis, R.: iPhone and iPad Apps for Absolute Beginners. Apress, New York (USA): Springer-Verlag, 2010

[9] Mark, D.; LaMarche, J.: Beginning iPhone Development: Exploring the iPhone SDK. Apress, New York (USA): Springer-Verlag, 2009

[42] Oestereich, B.: Die UML-Kurzreferenz 2.3 für die Praxis. 5. Auflage, München: Oldenbourg Verlag, 2009

[10] Pilone, D.; Pilone, T.: Head First iPhone Development. Sebastapol (USA): O'Reilly Media, 2010

[11] Sadun, E.: Das große iPhone Entwicklerbuch. Addison-Wesley Verlag, 2010

[12] Schiller, J.; Voisard, A: Location Based Services. San Francisco (USA): Morgan Kaufmann Publishers, 2004

[13] Stäuble, M.: Programmieren fürs iPhone. Heidelberg: dpunkt.verlag, 2009

[14] Zdziarski, J.: iPhone Open Application Development. Sebastapol (USA): O'Reilly Media, 2008

[45] Zarra, M.: Core Data: Apple's API for Persisting Data on Mac OS X. Dallas (USA): The Pragmatic Bookshelf, 2009

## Papers

[15] Teker, U.: Realisierung und Evaluation eines Indoor Lokalisierungssystems mittels WLAN. Diplomarbeit, Universität Bremen, 2005

[16] Halbach, C.: Konzeption und Entwicklung eines mobilen Stadtführers. Diplomarbeit, Fachhochschule für Technik und Wirtschaft Berlin, 2004

## Zeitschriften

[17] Stäuble, M.: Entwicklung eines Location Based Service. In: Mac Developer (2010), Heft 1/2010, S. 27 - 33

[18] Bihr, P.; Schwarzmann, I.: Hier bin ich!: Wie uns Smartphones und Geodienste helfen, die Umwelt intensiver zu erfahren. In: t3n (2010), Heft 22, S. 40 - 42

## Vorlesungsskripte

[19] Prof. Dr. Roth, J.
Ortsbezogene Anwendungen und Dienste WS10/11
URL: http://www.informatik.fh-nuernberg.de/professors/roth/WS1011/Ortsbezug/OrtsbezogeneAnwendungenUndDienste_WS1011.pdf

[20] Prof. Dr.-Ing. Schiller, J.
Mobilkommunikation SS2005
FU Berlin, Institut für Informatik, AG Technische Informatik

[21] Steiniger, S.; Neun, M.; Alistair, E.
Foundations of LBS
CartouCHe - Lecture Notes on LBS, ETH Zürich, 2008
URL: http://www.e-cartouche.ch/content_reg/cartouche/LBSbasics/en/text/LBSbasics.pdf

## Internet

[22] Studie „Mobile Web Watch 2010: Durchbruch auf Raten – mobiles Internet im deutschsprachigen Raum“ [Stand 2010]
accenture
URL: http://www.accenture.com/NR/rdonlyres/7931CF1F-3B82-4310-A3F4-ECB5C25491A5/0/Accenture_Mobile_Web_Watch_2010.pdf

[23] Apple iPhone [Stand Oktober 2010]
Apple Inc.
URL: http://www.apple.com/de/iphone/apps-for-iphone/#heroOverview

[24] What’s New in iOS 4 [Stand November 2010]
Apple
URL: http://developer.apple.com/technologies/ios/whats-new.html

[25] Goebel, M.
iPhone ist das wichtigste Internet-Handy [Stand 18. Mai 2010]
www.areamobile.de
URL: http://www.areamobile.de/news/15350-website-statistik-iphone-ist-das-wichtigste-internet-handy

[26] Apple iPhone [Stand 30. Oktober 2010]
wikipedia
URL: http://de.wikipedia.org/wiki/Apple_iPhone

[27] Apple iPhone 4 Multitasking [Stand November 2010]
Apple
URL: http://www.apple.com/de/iphone/features/multitasking.html

[28] Apple iOS Reference Library [Stand November 2010]
Apple
URL: http://developer.apple.com/library/ios/navigation/index.html

[29] Platform Versions [Stand Dezember 2010]
Android Developers
URL: http://developer.android.com/resources/dashboard/platform-versions.html

[30] Android Fragmentierung - hat Steve Jobs Recht? [Stand 21. Oktober 2010]
Fabien Röhlinger
URL: http://www.androidpit.de/de/android/blog/393292/Android-Fragmentierung-hat-Steve-Jobs-Recht

[31] globaler Smartphone-Markt [Stand November 2010]
fscklog
URL: http://www.fscklog.com/2008/11/iphone-os-springt-auf-platz-2-im-globalen-smartphone-markt.html

[32] Apple verkauft 2010 weltweit 36 Millionen iPhones [Stand November 2010]
ZDNet
URL: http://www.zdnet.de/news/mobile_wirtschaft_analyst_ apple_verkauft_ 2010_weltweit_36_millionen_iphones_story-39002365-41525271-1.htm

[33] Vergleichstabelle iPhone 3GS / iPhone 4 [Stand November 2010]
Apple
URL: http://www.apple.com/de/iphone/compare-iphones/

[34] Touchscreen-Trick: Bockwürstchen erspart kalte Finger [Stand November 2010]
Focus Online
URL: http://www.focus.de/digital/videos/touchscreen-trick-bockwuerstchen-erspart-kalte-finger_vid_15584.html

[35] GPS Grundlagen [Stand 2003]
u-blox AG
URL: http://www.u-blox.com (Online nicht mehr verfügbar)

[36] What’s New in iOS 4 [Stand November 2010]

Apple
URL: http://developer.apple.com/technologies/ios/whats-new.html

[37] mTrip Homepage [Stand November 2010]
mTrip
URL: http://www.mtrip.de/

[38] mTrip Berlin Reiseführer [Stand November 2010]
Apple iTunes
URL: http://itunes.apple.com/de/app/berlin-reisefuhrer-mtrip/id377135805?mt=8

[39] Navigaia [Stand November 2010]
iTunes Produktseite
URL: http://itunes.apple.com/de/app/berlin-multimedia-travel- guide/
id340954734?mt=8

[40] Become A Tour Author [Stand November 2010]
GPSmyCity.com
URL: http://www.gpsmycity.com/author.html

[41] About Us [Stand November 2010]
GPSmyCity.com
URL: http://www.gpsmycity.com/about-us.html

[43] iOS Technology Overview [Stand Dezember 2010]
Apple Inc.
URL: http://developer.apple.com/library/ios/#documentation/Miscellaneous/
Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//
apple_ref/doc/uid/TP40007898-CH1-SW1

[44] Cocoa Fundamentals Guide
Apple Inc.
URL: http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/
CocoaFundamentals/CocoaDesignPatterns/CocoaDesignPatterns.html#//apple_ref/
doc/uid/TP40002974-CH6-SW6

# 8 Anhang

## 8.1 Kriterienkatalog

### 8.1.1 Analyse der mTrip-App

| **Name** | mTrip |
|---|---|
| **Anwendungsbereich** | Persönliche Navigation und mobiler Stadtführer |
| **Design** | |
| Icon | funktionales Design<br>ein Wiedererkennungswert ist gegeben<br>die Funktionalität geht aus dem Icon nicht hervor<br>das Design des User Interface spiegelt sich im Icon wieder |
| User Interface | klare Aufteilung und Gliederung der Benutzeroberfläche<br>ein Wiedererkennungswert ist gegeben<br>Nein |
| **Usability** | |
| Navigationskonzept | Tab Bar Controller<br>Ja |
| Bedienbarkeit | Ja<br>Nein |
| Bedienelemente | fingergerechte Gestaltung |
| Routenfunktionalität | Ja<br>als Linie<br>Ja<br>Ja<br>nicht vorhanden<br>Ja<br>nicht vorhanden |
| Kartenfunktionalität | 2D<br>Ja<br>Ja<br>Ja<br>OpenStreetMap |
| Tourenfunktionalität | Ja<br>Ja<br>Ja<br>Ja; über das An- und Abreisedatum<br>Ja; über Tab Bar Item Reiseführer |
| **Content** | |
| Informationsangebot | Beschreibungen zu POIs und Kurzinformation (Adresse und Öffnungszeiten); Auswahl über Piktogramme in der Kartenansicht |
| Quelle | Ja; Detailbeschreibung durch wikipedia |
| Sprachenangebot | Deutsch, Englisch, Französisch, Spanisch und Italienisch |
| Multimediadaten | Nein |
| **Personalization** | |
| Routenplanung | Hinzufügen des POI über Toolbar „Routenplan“<br>Aufnahme per Kamera oder Auswahl eines Fotos aus dem Album<br>Ja<br>Ja |
| Zielgruppe | Berlintouristen allgemein<br>Nein<br>Ja |
| Favoritenliste | Verwaltung der Favoritenliste von POIs über Toolbar |

| | |
|---|---|
| Bewertung/Share | Ja<br>nicht vorhanden |

### 8.1.2 Analyse der iSightseeing-App

| | |
|---|---|
| **Name** | iSighseeing |
| **Anwendungsbereich** | Persönliche Navigation und mobiler Stadtführer |
| **Design** | |
| Icon | gezielte Verwendung von allgemeingültigen Symbolen<br>Ja<br>Ja<br>Nein |
| User Interface | Prototyp; Design noch veränderbar |
| **Usability** | |
| Navigationskonzept | Tab Bar Controller<br>Ja |
| Bedienbarkeit | Ja<br>Nein |
| Bedienelemente | fingergerechte Gestaltung |
| Routenfunktionalität | nicht vorhanden |
| Kartenfunktionalität | interaktive Piktogramme auf 2D-Karte |
| Tourenfunktionalität | nicht vorhanden |
| **Content** | |
| Informationsangebot | Beschreibungen zu POIs und Kurzinformation; Auswahl über Piktogramme in der Kartenansicht |
| Quelle | globalguides.net |
| Sprachenangebot | Englisch |
| Multimediadaten | Ja |
| **Personalization** | |
| Routenplanung | Ja; My Tour<br>Ja<br>Ja<br>Ja; My Tour |
| Zielgruppe | ausländische Berlintouristen<br>Nein<br>Ja |
| Favoritenliste | Nein |
| Bewertung/Share | nicht vorhanden |

## 8.2 Diagramme

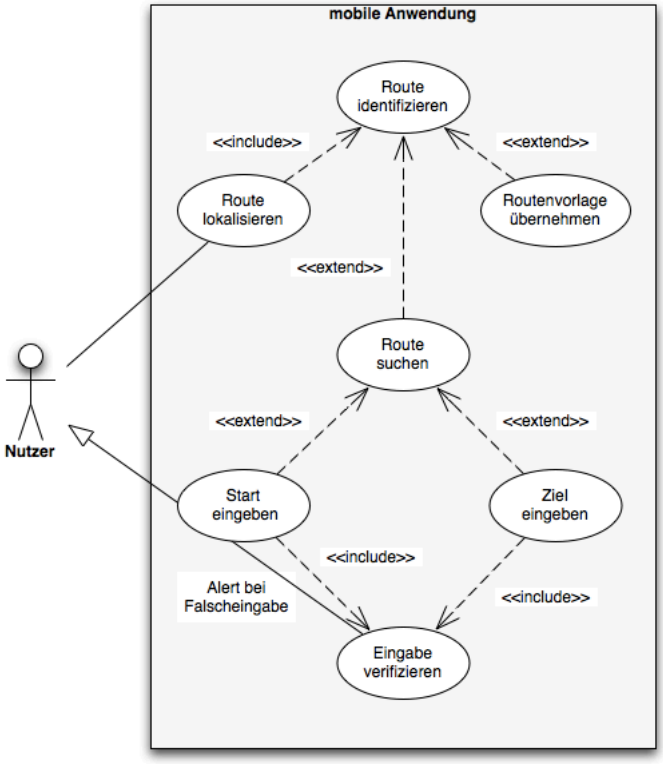


Abbildung Anhang 1: Anwendungsfalldiagramm – „Route lokalisieren“

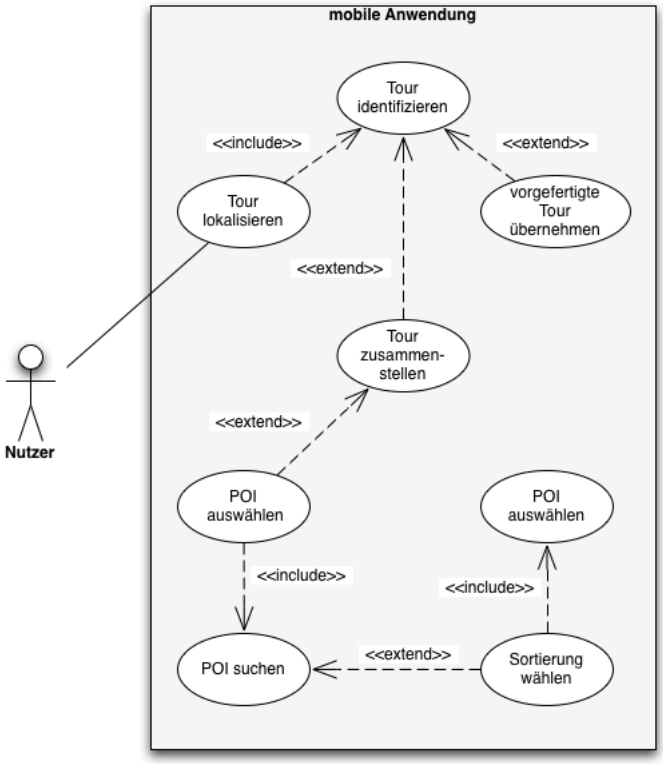


Abbildung Anhang 2: Anwendungsfalldiagramm – „Tour lokalisieren“

## 8.3 Glossar

| | |
|---|---|
| A-GPS | Assisted GPS, Hilfsinformationen werden zur schnelleren Positionsbestimmung durch Mobilfunkortung hinzugezogen |
| API | Application Programming Interface (Programmierschnittstelle) |
| APP | mobile Smartphone-Anwendung |
| Delegation | Weiterleiten von Verantwortung an andere Klassen. Umgeht das Erstellen von Subklassen. |
| EDGE | Enhanced Data Rates for GSM Evolution, Erweiterung des GSM-Standards |
| Early Adopter | bezeichnet einen Menschen, der die neuesten technischen Errungenschaften verwendet |
| Entwurfsmuster | (engl. design patterns) sind bewährte Lösungsschablonen für wiederkehrende Entwurfsprobleme in Softwarearchitektur und Softwareentwicklung. |
| GIS | Geoinformationssysteme |
| GPS | Global Positioning System |
| GSM | Global System for Mobile Communications, Mobilfunkstandard 2G |
| HSDPA | High Speed Downlink Packet Access, Erweiterung von UMTS |
| HSUPA | High Speed Uplink Packet Access, Erweiterung von UMTS |
| IDE | Integrated Development Environ von Apple im Jahr 2007, mit intuitivem Bedienungskonzept, hoher Datenübertragungsrate und gleichzeitige Kopplung an günstige und transparente Datentarife, hat die Mobilfunkbranche revolutioniert: Mobiles Internet wurde endlich für Endverbraucher intuitiv anwendbar und bezahlbar. Lästige Barrieren zum direkten Zugang ins Internet, wie Portalseiten, sind entfallen |
| MVC Pattern | Model View Controller Pattern. Trennung von Code verschiedener Ebenen. |
| NMEA | National Marine Electronics Association |
| OVI | Orte von Interesse |
| POI | Point Of Interest |
| SDK | Software Development Kit |
| Smartphone | Bezeichnung für Mobiltelefone mit vollständiger virtueller oder physischer Tastatur und eigenem Betriebssystem. Der Funktionsumfang ist durch Zusatzapplikation erweiterbar. |
| Target Action | Verknüpfung zwischen View und Controller |
| Thumbnail | Vorschaubild |
| Touchscreen | berührungsempfindliches Display |
| Triangulation | Vermessung eines Punkts im Dreieck |
| UIKit | grafische Benutzeroberfläche des iPhones und iPads |
| UMTS | Universal Mobile Telecommunications System, Mobilfunkstandard 3G |
| Web | Synonym für Internet |

| | |
|---|---|
| WGS 84 | World Geodetic System von 1984 ist das geodätische Referenzsystem für Positionsangaben auf der Erde und im erdnahen Weltraum |
| WLAN | Wireless Local Area Network |
| WPS | Wi-Fi Positioning System, der Anbieter Skyhook besitzt eine weltweite Datenbank von WLAN Access Points |
| XML | Extensible Markup Language |

## 8.4 Abbildungsverzeichnis

Abbildung 1: Arten der Handynutzung
Abbildung 2: Vergleich der Auflösung beim iPhone 3GS und beim iPhone 4
Abbildung 3: Arten der mobilen Internetnutzung
Abbildung 4: GPS-Positionsbestimmung im Raum
Abbildung 5: Einordnung von LBS
Abbildung 6: LBS-Komponenten und Informationsfluss
Abbildung 7: Screenshots User Interface – mTrip Travel Guide
Abbildung 8: App Icon – mTrip Travel Guide
Abbildung 9: Screenshots Turn-by-Turn-Navigation und Postkarte – mTrip Travel Guide
Abbildung 10: Screenshot Fernsehturm am Alex – mTrip Travel Guide
Abbildung 11: Screenshots Personalization – mTrip Travel Guide
Abbildung 12: Screenshots User Interface – Navigaia Travel Guide
Abbildung 13: ausgewählte App Icons zum Vergleich – Navigaia Travel Guide
Abbildung 14: Screenshot Kartenansicht – Navigaia Travel Guide
Abbildung 15: Screenshots Startpunkte der Touren – Navigaia Travel Guide
Abbildung 16: Screenshots Verlauf der Touren – Navigaia Travel Guide
Abbildung 17: Screenshots Filteroptionen – Navigaia Travel Guide
Abbildung 18: Screenshots User Interface – City Walks
Abbildung 19: App-Icon – City Walks
Abbildung 20: Screenshot wikipedia Textmaterial – City Walks
Abbildung 21: Anwendungsfalldiagramm – Produzentensicht
Abbildung 22: Anwendungsfalldiagramm – Konsumentensicht
Abbildung 23: iOS-Architektur
Abbildung 24: MVC-Pattern
Abbildung 25: Aufbau des Tab Bar Controller
Abbildung 26: Aufbau der Zugriffsschicht Core Data Stack
Abbildung 27: App-Icon – iSightseeing
Abbildung 28: Startscreen – iSightseeing
Abbildung 29: Tab Bar – iSightseeing
Abbildung 30: Screenshots – Places
Abbildung 31: Screenshots – My Tour und Search
Abbildung 32: Screenshots – Guided Tour

Abbildung 33: Screenshots – Map

Abbildung 34: Datenmodell – iSightseeing

## 8.5 Tabellenverzeichnis

Tabelle 1: Tabellarischer Vergleich von iPhone 3GS und iPhone 4

Tabelle 2: Tabellarischer Vergleich der ausgewählten Apps

Tabelle 3: Kriterienkatalog

Tabelle 4: Benutzergruppe „Ausländischer Berlintourist“

Tabelle 5: Anwendungsfallbeschreibung „Daten erzeugen“

Tabelle 6: Anwendungsfallbeschreibung „Aktuelle Position ermitteln“

Tabelle 7: Anwendungsfallbeschreibung „Daten persistent speichern“

Tabelle 8: Anwendungsfallbeschreibung „Kategorie/Distanz auswählen“

Tabelle 9 Anwendungsfallbeschreibung „POI aus Datenbank abrufen“

Tabelle 10: Anwendungsfallbeschreibung „aktuelle Position und POIs auf einer digitalen Karte visualisieren“

Tabelle 11: Anwendungsfallbeschreibung „Details zu einem ausgewählten POI anzeigen“

Tabelle 12: Anwendungsfallbeschreibung „Multimediadaten eines ausgewählten POI abspielen“

Tabelle 13: Technologien und Frameworks der Cocoa-Touch-Schicht

Tabelle 14: Umwandlung des relationalen in ein objektorientiertes Datenbankmodell