

BACHELORARBEIT

Konzeption und Entwicklung eines Java-basierten Navigationssystems für GPS-fähige mobile Endgeräte

Fachhochschule Köln, Campus Gummersbach

Axel Freudiger

11038993

Zoya Ghoraishi Kaverdi

11038989

Erstprüfer: Prof. Dr. Lutz Köhler

Zweitprüfer: Dipl. Wirt.-Ing. Jens Piesk

Köln, 11. Juli 2007

Inhaltsverzeichnis

I. Einführung	8
1. Motivation	9
2. Ansatz und Zielsetzung	11
II. Grundlagen	12
3. Entwicklung für mobile Endgeräte	13
3.1. J2ME- Java 2 Micro Edition	13
3.2. CDC-Connected Device Configurations	14
3.3. CLDC-Connected, Limited Device Configuration	14
3.4. Unterschiede CDC-CLDC	15
3.5. MIDP-Mobile Information Device Profile	15
4. Lokalisierung mittels J2ME	16
4.1. Java Location API	17
4.2. Externer GPS-Empfänger	17
4.2.1. NMEA-Protokoll	17
5. Kommunikation geospezifischer Daten	19
5.1. GPX	19
5.2. KML	19
6. Web Services vs. Servlets	20
6.1. Web Services	20
6.2. Servlets	22
6.3. Fazit	23
7. Kartenmaterial	24
7.1. OpenStreetMaps	24
7.2. GoogleMaps	25
7.3. YahooMaps	26
7.4. Census 2000 Tiger	27

7.5. VMAP	27
7.6. Bluemarble	28
7.7. Fazit	28
8. Darstellung	29
8.1. SVG in Java	29
8.2. MIDLet Framework	30
8.3. Fazit	32
9. Software	33
9.1. Apache Tomcat	33
9.2. Xampp	33
10. ÖPNV - Öffentlicher Personennahverkehr	34
10.1. Deutsche Bahn	34
10.2. KVB	34
III. Vorgehensmodell	36
11. Analyse und Definition	40
11.1. Zielbestimmungen	41
11.1.1. Musskriterien	41
11.1.2. Wunschkriterien	42
11.1.3. Abgrenzungskriterien	42
11.2. Produkteinsatz	43
11.2.1. Anwendungsbereiche	43
11.2.2. Benutzeranforderungen	43
11.2.3. Betriebsbedingungen	43
11.3. Produktumgebung	44
11.3.1. Software	44
11.3.2. Hardware	44
11.3.3. Orgware	44
11.4. Produktfunktionen	45
11.4.1. Benutzerfunktionen	45
11.4.2. Hilfefunktionen	47
11.4.3. Standortfunktionen	47
11.4.4. Route-Funktionen	48
11.4.5. Suchfunktionen	49
11.4.6. Kartenfunktionen	50
11.4.7. System-Einstellungen	50
11.4.8. Administratorfunktionen	50
11.5. Produktdaten	52

11.6. Produktleistungen	54
11.7. Benutzungsoberfläche	55
11.7.1. Dialogstruktur	55
11.8. Qualitätszielbestimmungen	58
11.9. Globale Testszenarien und Testfälle	59
11.9.1. Benutzerfunktion	59
11.9.2. Hilfefunktion	60
11.9.3. Standortfunktionen	60
11.9.4. Route-Funktionen	61
11.9.5. Such-Funktionen	62
11.9.6. Karten-Funktionen	62
11.9.7. System-Einstellungen	62
11.9.8. Administrator-Funktionen	63
11.10. Entwicklungsumgebung	64
11.10.1. Software	64
11.10.2. Hardware	64
11.11. Ergänzungen	65
11.11.1. Barrierefreiheit	65
11.11.2. Automatisierung Routenfunktion	65
12. Entwurf	66
12.1. Zielgruppe	66
12.2. Szenarien	66
12.3. Teilprodukte identifizieren	68
12.3.1. Darstellung	68
12.3.2. Navigation	68
12.3.3. Routenverbindung mittels öffentlicher Verkehrsmittel	69
12.3.4. Übermittlung der GPS-Koordinaten	69
12.3.5. Favoriten managen	70
12.3.6. Points of Interest	70
12.3.7. Virtuelle Tour	71
12.4. Paper-Based-Prototype	71
12.4.1. Login	72
12.4.2. Help	74
12.4.3. Hauptseite	75
12.4.4. My Location	76
12.4.5. Favoriten	77
12.4.6. Points of Interest	78
12.4.7. Route	78
12.4.8. Virtual Tour	80
12.4.9. Search	81

12.4.10.Settings	82
12.4.11.Quit	82
12.4.12.get Anywhere	83
12.5. Probandenbefragung	85
12.6. Auswertung der Probandenbefragung	85
12.6.1. Ergebniss:	85
12.6.2. Fazit	88
13. Implementation	89
13.1. Systemarchitektur	89
13.1.1. Überblick	89
13.1.2. Dreischichtige Architektur	91
13.2. Client	92
13.2.1. MIDlet Framework	92
13.2.2. Google Maps Interface	95
13.3. Server	97
13.3.1. Das Zamspack-Servlet	97
13.3.2. MySQL-Datenbank	99
14. Test	100
IV. Fazit und Ausblick	101
V. Quellenangaben	102
VI. Anhang	104
15. Fragenkatalog	105
16. MySQL-Anweisungen der Datenbank	106
VII. Erklärung	109

Abbildungsverzeichnis

6.1. Funktionsweise: Webservice	21
6.2. Funktionsweise: Servlets	22
7.1. OpenSourceMap Karte für Köln	25
7.2. J2ME Map Beispiele	26
7.3. Yahoo MAPS API Kartenausschnitte	27
9.1. Serverkomponente: Apache Tomcat	33
9.2. Serverkomponente: Xampp Distribution	33
10.1. Vorgehensmodell: Wasserfallmodell	38
10.2. ISO 13407	39
11.1. Dialogstruktur der Startseite	55
11.2. Dialogstruktur der Hauptseite	56
11.3. Dialogstruktur der Konfigurationsseite	57
12.1. Prototyp Login-Screen	73
12.2. Prototyp Help-Screen	74
12.3. Prototyp Hauptmenü-Screen	75
12.4. Prototyp My Location-Screen	76
12.5. Prototyp Favoriten-Screen	77
12.6. Prototyp Point of Interest-Screen	78
12.7. Prototyp Route-Screen	79
12.8. Prototyp Virtual Tour-Screen	80
12.9. Prototyp Search-Screen	81
12.10 Prototyp Get Anywhere-Screen	84
13.1. Architektur: Interaktion der einzelnen Komponenten	90
13.2. Darstellung: Dreischichtige Architektur	92
13.3. ER-Diagramm der Datenbankstruktur	99

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
COO	Cell of Origin
CVM	Customer Virtual Machine
DIN	Deutsche Industrie Norm
GPS	Global Position System
GPX	GPS Exchange Format
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
ISO	Internationale Organisation für Normung
J2EE	Java 2 Platform, Enterprise Edition
J2ME	Java 2 Platform, Micro Edition
J2SE	Java 2 Platform, Standard Edition
JCP	Java Community Prozesses
JSP	JavaServer Pages
JSR	Java Specification Request
JVM	Java Virtual Machine
KML	Keyhole Markup Language
KVB	Kölner Verkehrs-Betriebe
MIDlet	Mobile Information Device Applet
MIDP	Mobile Information Device Profiles
MSA	Mobilen Service Architektur
NMEA	National Marine Electronics Association
PDA	Personal Digital Assistant
RMS	Record Management Store
ROM	Read-Only Memory
RPC	Remote Procedure Call
SMS	Short Message Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
TOA	Time of Arrival
UDDI	Universal Description, Discovery and Integration
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier

USB	Universal Serial Bus
VM	Virtual Machine
VPF	Vector Product Format
WGS84	World Geodetic System 1984
WLAN	Wireless Local Area Network
WSDL	Web Services Description Language
WTK	Wireless Toolkit
XML	Extensible Markup Language

Teil I.

Einführung

1. Motivation

In der heutigen kommunikationsorientierten Gesellschaft haben mobile Endgeräte eine hohe Verbreitung erlangt. Laut dem Statistischen Bundesamt Deutschland wurde im Jahr 2006 zum ersten Mal die 80-Prozent-Marke der Handy-Ausstattung im privaten Haushalt überschritten. Die meisten der inzwischen im Umlauf befindlichen mobilen Endgeräte bieten die Möglichkeit Java-Anwendungen auszuführen. Auch technisch Unbegabten gelingt es schnell und einfach diese Anwendungen per SMS oder Klick auf einer Internetseite zu installieren.

Auch die Übertragungstechniken für funkbasierte Geräte werden besser und die Bandbreite dieser immer größer. Somit erfreuen sich mehr und mehr Anwender der UMTS-Technik und schließen dementsprechende Handyverträge mit Datentransferpaketen ab. Auch die WLAN(*WiFi*)-Technik erfreut sich mehr Beliebtheit. Sogenannte HotSpots, welche einen Zugangsknoten eines solchen Netzes beschreiben, verteilen sich großzügig in deutschen Städten. Beispielsweise erstrecken sich über 200 HotSpots der FON-Community¹ allein in der Kölner Innenstadt.

Das Ziel dieser Arbeit ist, ein benutzerorientiertes Touristeninformationssystem zu entwerfen, dass eine Nutzung von Informationen öffentlicher Verkehrsmittel vorsieht, speziell für die KVB (Kölner Verkehrsbetriebe)². Die KVB bieten bereits eine Möglichkeit zur Fahrplanauskunft für mobile Geräte (Handyinfo³), welche kostenlos auf deren Server verfügbar ist.

Zum Zeitpunkt dieser Arbeit sind Navigationssysteme in Fahrzeugen aller Art fast serienmäßig integriert. Jedoch gibt es nur wenige solcher Systeme für Fußgänger bzw. Radfahrer. Des weiteren sind sie in jedem uns bekannten Fall unidirektional, also nur in einer Kommunikationsrichtung nutzbar.

Die Informationen dieser Systeme sind daher kritisch zu betrachten, da sie dem Benutzer oft unter Verträgen mit Drittanbietern bereitgestellt werden. Dies hat zur Folge, dass Benutzer des Systems durch den Anbieter beeinflusst werden in der Art und Menge der Informationen.

¹vgl. <http://www.fon.com/de/>

²<http://www.kvb-koeln.de>

³vgl. <http://www.kvb-koeln.de/german/fahrplan/java/java.html>

Um dieser Situation zu entgehen, konzipierten wir ein System mit bidirektionaler Kommunikation, um den Benutzern nicht nur die Möglichkeit zu bieten Daten zu empfangen, sondern auch zu erzeugen. Dadurch werden die Informationen des Systems nicht durch Gewinnmaximierung bestimmt.

2. Ansatz und Zielsetzung

In dieser Arbeit möchten wir ein Java-basiertes Navigationssystem für mobile Endgeräte konzipieren, das Benutzern nicht nur die Möglichkeit gibt, Informationen zu konsumieren, sondern auch für andere Benutzer bereitzustellen. Wichtige Teile dieser Arbeit werden als prototypische Implementierung entwickelt.

Um dem Benutzer eine Möglichkeit zu bieten möglichst effizient zu einem bestimmten Ziel zu gelangen, entwickelten wir eine "One-Click-Strategie". Damit bieten wir eine Variante, um die Route zu einem vom Nutzer festgelegten Standort mit so wenigen Klicks wie möglich unter optionaler Verwendung der öffentlichen Verkehrsmittel zu ermitteln.

Zu Beginn der Arbeit werden wir einige Grundlagen schaffen, die klären sollen, für welche Geräte diese Anwendung entworfen wird und was diese für technische Besonderheiten bietet.

Teil II.

Grundlagen

3. Entwicklung für mobile Endgeräte

Bei der Entwicklung von Anwendungen für mobile Endgeräte sind zwei verschiedenen Arten zur Entwicklung zu beachten:

- Java für J2ME (*plattformunabhängig*)
- native Anwendung für ein bestimmtes (*mobiles*) Betriebssystem

Für eine plattformabhängige Anwendung stehen sämtliche Ressourcen des Gerätes zur Verfügung, während für eine in Java geschriebene Anwendung nur ein beschränkter Teil der Ressourcen verfügbar ist.

Ein weiterer Unterschied besteht in der Portierbarkeit. Während in J2ME geschriebene Anwendungen auf allen Endgeräten laufen, die eine JVM(Java Virtual Machine) besitzen, laufen native Anwendungen nur auf den Geräten, für dessen Betriebssystem die Anwendung speziell entwickelt wurde.

Auf Grund der hohen Portierbarkeit der J2ME haben wir uns in dem Prototyp dieser Arbeit dafür entschieden.

3.1. J2ME- Java 2 Micro Edition

Zu den Hauptkomponenten von Java 2 Plattform, Micro Edition (J2ME-Plattform)¹ gehören CDC (Connected Device Configurations), CLDC (Connected Limited Device Configurations), MIDP (Mobile Information Device Profiles).

J2ME-Technologien stellen nur einen Teil der Java-Softwareproduktfamilie dar. Verwandte Java-Plattformen sind Java 2 Plattform, Standard Edition (J2SE-Plattform) und Java 2 Plattform, Enterprise Edition (J2EE-Plattform).

J2ME ist eine reduzierte Version von J2SE², die das Minimum an Unterstützung beinhaltet, das für mobile Endgeräte benötigt wird, wobei J2ME ein paar Elemente beinhaltet, welche nicht in J2SE enthalten sind. J2ME ist speziell für vernetzte, eingeschränkte Geräte gedacht.

¹vgl. <http://java.sun.com/javame/>

²vgl. <http://java.sun.com/javase/>

3.2. CDC-Connected Device Configurations

Die CDC (Connected Device Configuration) enthält die komplette Java 2 Plattform VM, die CVM (Customer Virtual Machine) und die minimal erforderlichen API Klassen für das System. Für Applikationen wird zusätzlich ein Profil benötigt. Neben dem "foundation profile", welches alle übrigen Klassen und APIs beinhaltet, müssen bei Bedarf noch andere Profile (z.B. für GUI) eingebunden werden.

CDC ist einer Superklasse von CLDC und ist somit auch für CLDC Geräte einsetzbar. Alle Geräte, welche Personal Java unterstützen, können leicht auf CDC portiert werden.

3.3. CLDC-Connected, Limited Device Configuration

Die CLDC (Connected, Limited Device Configuration) ist die Basis für die darauf aufbauenden Profile. Die Profile definieren eine zusätzliche Bibliothek an APIs und Eigenschaften, welche im Gegensatz zu den Konfigurationen für den vertikalen Markt bestimmt sind. Im Zusammenspiel von CLDC und den Profilen werden Applikationen entwickelt.

Das Ziel der CLDC ist die Unterstützung einer breiten Palette an Geräten, welche mit sehr beschränkten Ressourcen auskommen müssen. Als großer Vorteil der Java Technologie in portablen Geräten gilt das dynamische und sichere Verbreiten von interaktivem Inhalt und Anwendungen über verschiedenartige Netzwerke. Die CLDC setzt außer der minimalen Speicheranforderung keine besonderen Hardware-Anforderungen voraus.

Die Softwareanforderungen variieren genau so stark wie die Hardwareanforderungen. Einige Geräte werden über eine vollständig implementierte JVM verfügen, während andere Geräte mit einem beschränkten Softwareumfang nur die notwendigen Klassen benötigen.

3.4. Unterschiede CDC-CLDC

CDC

- vor allem auf Devices ausgerichtet mit grafischer Oberfläche
- Speicher: 2 -16 MB
 - min. 512K ROM
 - min. 256K ROM
- komplette Implementierung der JVM, CVM
- große Bandbreite für die Daten kommunikation
 - TCP/IP (verschiedene Verbindungstypen)
 - permanente Verbindung

CLDC

- minimale grafische Oberfläche
- Speicher: 160-512KB
 - min. 128KB ROM für JavaVM und CLDC-Libraries
 - min. 32KB for Java runtime and objects
- 16bit oder 32bit Prozessor
- beschränkte VM, z.B: KVM
- geringe Bandbreite für die Datenkommunikation
 - normalerweise kein TCP/IP
 - keine permanente Verbindung
 - 9600 bps
 - meist drahtlos
- ausgelegt für Geräte geringem Stromverbrauch (Batterie, Akku)
- Einsatz in Massenartikeln

3.5. MIDP-Mobile Information Device Profile

Das MIDP definiert grundlegende Funktionen für die Erstellung von Anwendungen. Anwendungen, die für das MIDP programmiert sind, werden "MIDlets" genannt. Das MIDP ist speziell für mobile Endgeräte mit kleinem Bildschirmen und beschränkten Eingabemöglichkeiten vorgesehen.

Definiert werden:

- Lebenszyklus: Jedes MIDlet muss von der Klasse MIDlet abgeleitet werden. Diese Klasse definiert den Lebenszyklus eines MIDlets durch die Methoden startApp(), destroyApp() und pauseApp().
- High-Level-API: Die High-Level-API besteht aus Klassen, die eine einfache Benutzerschnittstelle definieren.
- Low-Level-API: Die Low-Level-API besteht aus Klassen, die es ermöglichen, direkt auf die Eingaben des Benutzers zuzugreifen. Außerdem ist es möglich, auf dem Bildschirm des Gerätes zu malen.
- Record Management Store(RMS): Der RMS definiert eine einfache Möglichkeit zur Persistierung von Daten.

4. Lokalisierung mittels J2ME

Unter Lokalisierung versteht man die Bestimmung der eigenen Position in einem Koordinatensystem. Im Zusammenhang mit mobilen Geräten wird häufig ein ellipsoides, geografisches Koordinatensystem verwendet, mit welchem die Position in Längengrad (östlich oder westlich von Greenwich), Breitengrad (nördlich oder südlich des Äquators) und der Höhe über dem Meeresspiegel angegeben wird. Die Koordinaten beziehen sich dabei auf ein zugrundeliegendes Referenzsystem. Ein weit verbreitetes Referenzsystem ist das *World Geodetic System 1984* (WGS84).¹

- Die Position des mobilen Gerätes kann mit Hilfe von Winkeln und Entfernungen zu Referenzpunkten bestimmt werden. Eine typische Methode ist das satellitenbasierte GPS (Global Position System). Um die Position mit GPS zu bestimmen, müssen die Abstände zu drei Satelliten und eine Referenzzeit von einem vierten Satelliten bekannt sein. Die Position wird bei GPS anhand der Laufzeit eines Funksignals bestimmt (Time of Arrival, TOA). Mit GPS kann die Position auf ca. 15 Meter genau bestimmt werden. Mit Hilfe eines Korrektursignals, das von einer Basisstation ausgesendet wird, kann die Genauigkeit auf eins bis fünf Meter verbessert werden (Differential-GPS). Ein Nachteil des GPS-Verfahrens ist, dass es ohne Satellitenempfang (z. B. in Gebäuden) nicht funktioniert.
- Eine zweite bekannte Technologie ist die Positionsbestimmung auf der Basis von Zelleninformationen der GSM- und UMTS-Funknetze (Cell of Origin, COO). Das mobile Gerät kann feststellen, mit welcher Basisstation es verbunden ist. In städtischen Gebieten haben die Zellen einen Durchmesser von ca. 300 Metern, in ländlichen Gebieten jedoch über 20 Kilometer. Je größer der Durchmesser der Zelle, umso ungenauer ist diese Art der Positionsbestimmung. Der Vorteil dieser Methode ist jedoch der geringe Stromverbrauch und die Tatsache, dass die Lokalisierung auch in Gebäuden funktioniert.
- Für die Lokalisierung in Gebäuden werden Methoden verwendet, welche die Signalstärke von Sendern messen, deren Standort und Sendestärke bekannt ist. Da die Stärke des Signals mit zunehmender Entfernung vom Sender abnimmt, kann daraus die Distanz abgeschätzt werden. Als Funksignale werden entweder WLAN

¹vgl. http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html

oder Bluetooth verwendet. Ein Problem bei dieser Methode ist, dass die Signalstärke und somit auch die Distanzabschätzung von Hindernissen beeinflusst werden.

4.1. Java Location API

Das J2ME Location API definiert eine einheitliche und einfache Schnittstelle, über die Java-Anwendungen auf Lokalisierungsinformation zugreifen können.

Das J2ME Location API [JSR179] ist ein optionales Paket, welches im Rahmen des Java Community Prozesses (JCP) als Java Specification Request (JSR) 179 definiert worden ist. Dieses Paket kann mit unterschiedlichen J2ME-Profilen verwendet werden (z. B. Mobile Information Device Profile (MIDP) 1.0 oder MIDP 2.0). Als minimale Plattform wird die Connected Limited Device Configuration (CLDC) 1.1 verlangt, da das Location API mit Gleitpunktzahlen arbeitet. Das Location API ist auch ein bedingt zwingender Teil der "Mobilen Service Architektur" (MSA) [JSR248]. Die MSA legt fest, welche JSRs von einem MSA kompatiblen Gerät unterstützt werden müssen. Als Teil der MSA-Spezifikation muss das Location API von einem MSA-Gerät unterstützt werden, welches einen GPS-Empfänger enthält oder welches mit anderen Methoden Lokalisierungsinformation über ein API anbietet.

4.2. Externer GPS-Empfänger

Eine weitere Möglichkeit zur Positionsbestimmung für mobile Geräte durch GPS ist der Zugriff auf einen externen GPS-Empfänger über Bluetooth, Infrarot oder USB. Jedoch eignen sich Infrarot und USB nicht als Übertragungskanal. Infrarot hat eine zu geringe Reichweite und braucht für eine Verbindung Sichtkontakt. Da uns zum Zeitpunkt dieser Arbeit keine Handys mit USB-Anschluss bekannt sind, fällt diese Möglichkeit auch aus. Jedoch gibt es Entwicklungen einer Micro-USB-Schnittstelle, die in Zukunft in mobilen Geräten Verwendung finden könnte.

Für eine Verbindung über Bluetooth muss der entsprechende GPS-Empfänger diese Schnittstelle zur Verfügung stellen. Das Handy muß außer der Schnittstelle auch über das JSR82 Paket verfügen, um mittels JAVA mit der Schnittstelle zu kommunizieren.

4.2.1. NMEA-Protokoll

Das Standard-Protokoll zur Übertragung der Positionsdaten ist das NMEA-Protokoll, das von der NMEA ²(National Marine Electronics Association) definiert wurde.

Mit Hilfe der weitestgehend standardisierten NMEA-Daten gelingt es sehr leicht, die

²vgl. <http://www.nmea.org/>

Daten praktisch jedes GPS-Geräts mit einem Navigations- und Kartenprogramm auf dem PC, Laptop oder Handheld zu verwenden. Die GPS-Mäuse (GPS-Empfänger ohne Display, nur mit serieller Schnittstelle) kommunizieren ausschließlich auf diese Art mit ihrer Außenwelt.

Die Daten werden im ASCII-Format ³(American Standard Code for Information Interchange, Amerikanischer Standard Code für Informationsaustausch) übertragen.

Die Datensätze der unterschiedlichen Geräte können sehr viele verschiedene Informationen beinhalten, darunter Position, Geschwindigkeit, Richtung, Wassertiefe, Wassertemperatur, Wegpunkte, Windgeschwindigkeit usw. Wichtig ist, dass das Datenformat des GPS auf NMEA umgestellt wird.

Nachfolgend sind noch einige Datensätze im einzelnen aufgeschlüsselt:

Der GPRMC-Datensatz (RMC = recommended minimum sentence C, empfohlener Minimumdatensatz) ist eine Empfehlung für das Minimum, das ein GPS-Empfänger ausgeben soll.

\$GPRMC,191410,A,4735.5634,N,00739.3538,E,0.0,0.0,181102,0.4,E,A*19	
	Prüfsumme
	Modus (A,D,E,N,S)
	magnetische Deklination 0.4° E
	Datum: 18.11.02
	wahrer Kurs (ohne Bewegung 0)
	Geschwindigkeit über Grund (Knoten)
	007° 39.3538' östliche Länge
	47° 35.5634' nördliche Breite
	Empfängerwarnung, A = Daten OK, V = Warnung
	Uhrzeit der Positionsbestimmung: 19:14:10 UTC-Zeit

Das NMEA-Protokoll ist nicht nur für GPS-Empfänger entwickelt worden, deshalb enthalten die verschiedenen Sätze sowohl gerätespezifische Daten wie Wassertiefe, Windgeschwindigkeit und Wassertemperatur als auch allgemeine Informationen wie Bewegungsrichtung und die aktuelle Position.

Jedes Gerät, das das NMEA-Protokoll unterstützt, muss mindestens eine Satzart unterstützen - den RMC-Satz (Recommended Minimum Specific). Es gibt sehr viele weitere Sätze, auf die hier nicht weiter eingegangen wird, da sie zum einen geräte- bzw. herstellerabhängig sind und zum anderen in dieser Arbeit nicht genutzt werden.

³vgl. <http://www.torsten-horn.de/techdocs/ascii.htm>

5. Kommunikation geospezifischer Daten

5.1. GPX

Das GPS Exchange Format (kurz GPX) ist ein leicht-gewichtiges XML-basiertes Datenformat für den Austausch von GPS-Daten (Wegpunkte, Routen, Track Logs) zwischen Anwendungen und Web-Diensten im Internet. Zurzeit liegt das GPX-Format als XML-Schema in Version 1.1 vor. GPX ist ein offener Standard, der von jedem verwendet werden kann. Es wurde von der Firma Topografix entwickelt. Der Gebrauch ist aber nicht durch Gebühren oder Lizenzen beschränkt.

5.2. KML

Keyhole Markup Language, kurz KML, aktuelle Version 2.0, ist ein Austauschformat für die Client-Komponente des Programms Google Earth auf der Basis der Grammatik von XML 1.0. Es ist möglich, Punkte, Linien, Polygone und Bilder zu definieren. Zugleich lassen sich bestimmte Parameter spezifizieren, wie zum Beispiel Betrachtungswinkel und -entfernung.

Das Format KMZ ist eine datenkomprimierte KML-Datei im Format ZIP. Sie kann vom Client unmittelbar eingelesen werden.

6. Web Services vs. Servlets

6.1. Web Services

Bei Web Services handelt es sich um Software-Anwendungen, deren Schnittstellen als XML-Artefakte definiert, beschrieben und gefunden werden können. Diese sind mit einem URI eindeutig identifizierbar. Web Services sind nicht für menschliche Benutzer gedacht, sondern für Softwaresysteme, die automatisiert Daten austauschen und/oder Funktionen auf entfernten Rechnern aufrufen.

Web Services orientieren sich an der serviceorientierten Architektur (SOA). Sie vereinen daher verteilte und objektorientierte Programmierstandards und richten sich auf betriebswirtschaftliche Lösungen im Internet. Es lassen sich folgende Instanzen identifizieren:

- Service-Anbieter
- Service-Konsument
- Service-Verzeichnis (*Broker*)

Der Anbieter veröffentlicht in einem Verzeichnis die Beschreibung seiner Dienste. Der Konsument durchsucht das Verzeichnis und wählt den gewünschten Dienst aus. Nachdem eventuell weitere Protokolldetails ausgetauscht werden, findet die dynamische Anbindung des Konsumenten an den Anbieter statt. Der Konsument greift nun auf gewünschte Methoden zu.

Zur Kommunikation der einzelnen Komponenten dienen folgende Standards, die jeweils auf XML basieren:

- **UDDI** ist der Verzeichnisdienst zur Registrierung von Web Services, der das dynamische Finden durch den Konsumenten ermöglicht.
 - **WSDL** definiert eine plattform-, programmiersprachen- und protokollunabhängige XML-Spezifikation zur Beschreibung von Netzwerkdiensten (Web Services) zum Austausch von Nachrichten.
 - **SOAP (oder XML-RPC)** ist das Protokoll zur Kommunikation. Hier werden die eigentlichen Methoden des Web Service aufgerufen.
-

Die verwendeten plattformunabhängigen Standards sind in der Lage, entfernte Methodenaufrufe beliebiger Plattformen zu dekodieren und einer Anwendung weiterzuleiten. Auf diese Weise entsteht eine verteilte Architektur.

Die folgende Abbildung zeigt die wesentliche Funktionsweise von Web Services:

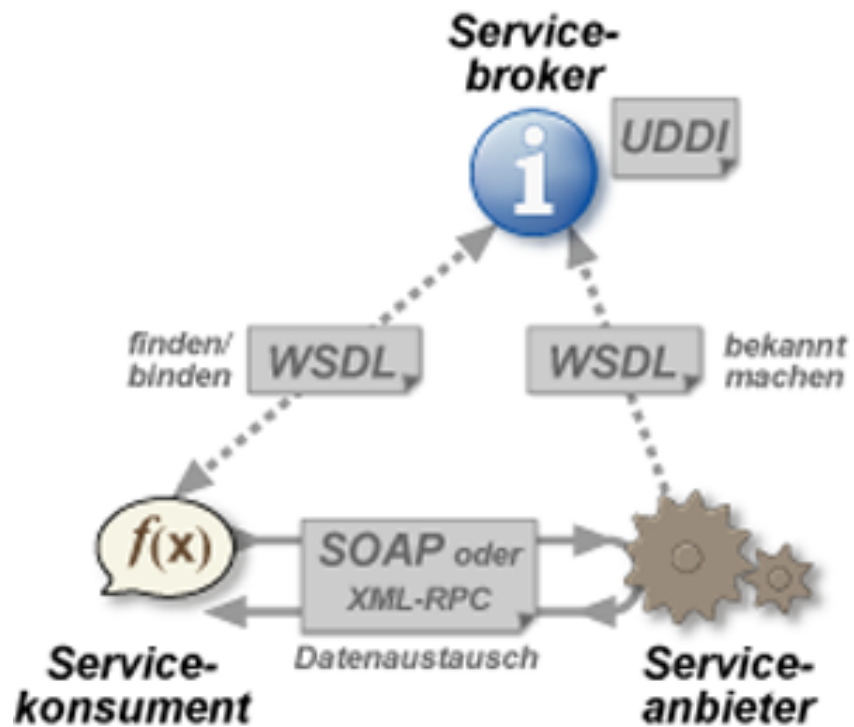


Abbildung 6.1.: Funktionsweise von Web Services

6.2. Servlets

Als Servlets bezeichnet man Instanzen von Java-Klassen, die in einer Web-Container-Umgebung (z.B. *Apache Tomcat*) verwendet werden und zur Laufzeit erzeugt und angesprochen werden. Diese Klassen müssen grundsätzlich die Schnittstelle *javax.servlet.Servlet* implementieren. Diese enthält zwei Methoden *doGet* und *doPost*, die überschrieben werden müssen, um die beiden wichtigsten HTTP-Methoden GET und POST verarbeiten zu können. Servlets sind Teil einer Architektur.

Die Metainformationen über das Servlet werden in einer XML-Datei namens "web.xml" hinterlegt, dem Deployment Descriptor. Diese XML-Datei wird zusammen mit der kompilierten Klasse (sowie ggf. weiteren benötigten Klassen) in eine einzige Archiv-Datei zusammengeführt (sogenanntes Web-Archiv), dieses wiederum wird dem Webcontainer über eine von ihm bereitgestellte Funktionalität zur Verfügung gestellt (sog. Deployment).

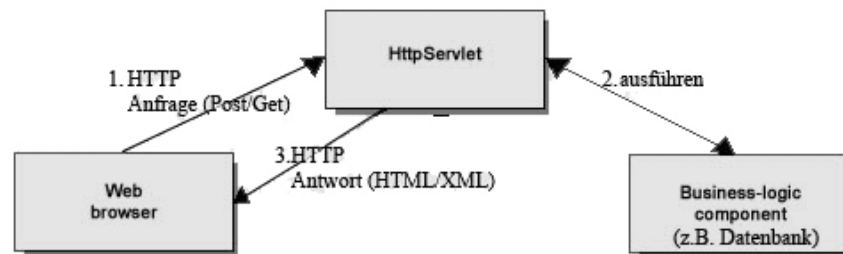


Abbildung 6.2.: Funktionsweise eines Servlets

1. Anfrage des Clients mit Übergabeparametern zur Verarbeitung
z.B. `http://server:80/?foo=bar` (*HTTP-GET*)
2. Verarbeitung der Paramter
z.B. `SELECT * FROM table WHERE foo=bar` (*MySQL*)
3. Antwort des Servers in Form von HTML oder XML Dokumenten
z.B.

```

<?xml version="1.0" encoding="UTF-8"?>
<Artikel>
    <foo>bar</foo>                // <-- WHERE foo=bar
    <eintrag>
        <stichwort>Senf</stichwort>
        <eintragstext>Senf ist gelb </eintragstext>
    </eintrag>
    <eintrag>
        <stichwort>Ketchup</stichwort>
  
```

```
<eintragstext>Ketchup ist rot</eintragstext>
</eintrag>
</Artikel>
```

6.3. Fazit

Durch die Verwendung von bereits bestehenden und weit verbreiteten Internet-Standards (HTTP, XML etc.) bei beiden Lösungen entstehen offene und dynamische Architekturen. Jedoch macht WSDL die Web Service Variante flexibler, die unabhängig von den verwendeten Plattformen, Programmiersprachen und Protokollen ist. Servlets dagegen sind reine Java-Programme, welche sich schnell lauffähig machen lassen. Um einen Web Service zu veröffentlichen ist es nötig, einiges mehr an Zeit zu investieren bis sämtliche Konfigurationen erfolgt sind.

Des weiteren brauchen Programmiersprachen, mit denen man Web Services einbinden will, spezielle Bibliotheken.

Auf Grund von fundierteren Vorkenntnissen in der Servlet-Programmierung und der Tatsache, dass Web Services mehr Zeit beanspruchen und das Servlets für den Funktionsumfang unseres Projektes ausreichen, entschieden wir uns für die klassische Servlet-Programmierung als Serverkomponente.

7. Kartenmaterial

Um eine Anwendung mit interaktiver Karte zu entwickeln ist es nötig Material zu verwenden, welches möglichst hochauflösend ist. Mittlerweile gibt es einige Anbieter, die diesem Kriterium genügen, jedoch einigten wir uns außerdem auf die Verwendung von kostenlosem Kartenmaterial zur Implementierung des Prototyps. Im Folgenden stellen wir nun einige Quellen vor, die Kartenmaterial liefern.

Die Kriterien, nach denen wir die Daten evaluiert haben, sind:

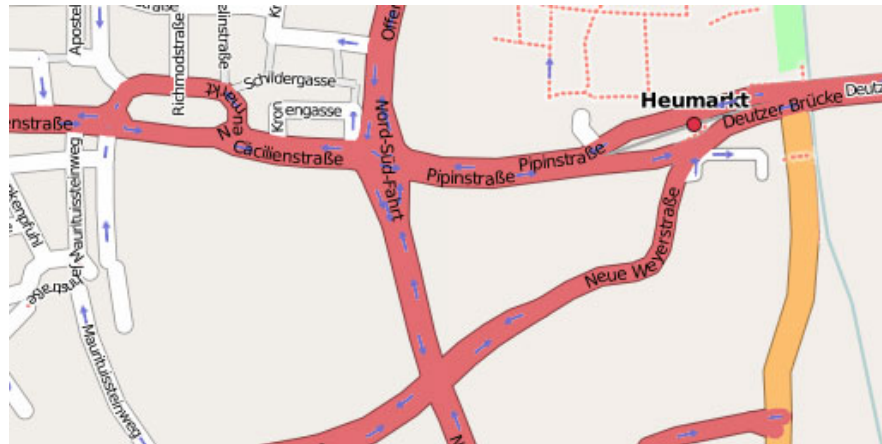
- **Auflösung**
Sind die Daten geeignet für eine optimale Darstellung auch auf hoher Zoomstufe?
- **Darstellbar in einem MIDlet**
Ist das Material auch von einem MIDlet aus verfügbar?
- **Kosten**
Sind die Daten kostenlos verfügbar?
- **Transfervolumen**
Wird die Datenmenge zur Darstellung möglichst gering gehalten?

7.1. OpenStreetMaps

"OpenStreetMap hat es sich zum Ziel gesetzt, freie geographische Daten wie zum Beispiel Straßenkarten zu erstellen und allen zur Verfügung zu stellen, die sie brauchen können. Dieses Projekt wurde begonnen, weil so ziemlich alle Karten, von denen man glaubt, dass sie von jedem für alle Zwecke und kostenlos verwendbar sind, tatsächlich aus technischen und rechtlichen Gründen leider nur eingeschränkt nutzbar sind. Dadurch wird verhindert, dass Leute diese Karten in kreativer und produktiver Weise einsetzen können. OpenStreetMap möchte diesem unglücklichen Zustand abhelfen."¹

¹Zitat von <http://wiki.openstreetmap.org>

Des Weiteren ist dieses Projekt noch in Entwicklung und zum Zeitpunkt dieser Arbeit sind für den Raum Köln noch wenig Daten verfügbar, wie die nachfolgende Abbildung zeigt.



7.2. GoogleMaps

Die Google Maps API⁴ lässt GoogleMaps-Content in Webseiten mit JavaScript integrieren. Da mobile Endgeräte, die dem MIDP2.0 Profil zugrunde liegen, über keine JavaScript-Funktion verfügen, lässt sich dieser Service nicht für unser Projekt einsetzen.

Das J2ME Map Paket von Landspurg⁵ ist ein Interface für Handys und PDAs, die mindestens auf MIDP2.0 und CLDC1.0/1.1 basieren. Zur Verwendung von GPS-Funktionen muß JSR179 und für Bluetooth JSR82 verfügbar sein. Dieses Paket ist frei verfügbar zur nicht-kommerziellen Verwendung. Über dieses Interface ist die komplette GoogleMaps Datenbank erreichbar und bietet eine sehr gute Auflösung für viele Bereiche der Erde, ob in der Satelliten- oder Kartenansicht. Daher lässt sich ein hoher Zoomfaktor erzielen

⁵ <http://j2memap.landspurg.net/>

mit sehr hoher Detaildichte. Außerdem ermöglicht es den Zugang zu RSS-Feeds um beispielsweise Points of Interest anzuzeigen, die kürzlich angelegt oder von einem bestimmten Benutzer stammen und ist erweiterbar mit eigenen, persönlichen Daten. Es verfügt über ein implementiertes GPS-Interface zur Ermittlung des eigenen Standortes.

Zum Datenaustausch unterstützt J2ME Map die GPX/KML Dateiformate, die in dieser Arbeit bereits näher beschrieben wurden.

Des weiteren bietet eine Implementierung dieses Pakets die Suche nach Fon-HotSpots⁶. Der Datenaustausch mit den Servern erfolgt im Vergleich mit den anderen Produkten sehr rasant und zuverlässig. Die nachfolgende Abbildung zeigt ein Suchergebnis des Begriffes "Sushi" im Raum San Francisco und die Freiheitsstatue in New York mit den dazugehörigen Koordinaten.

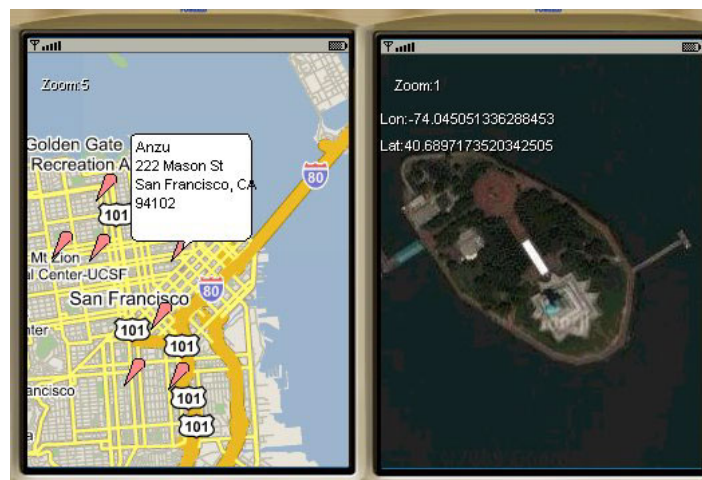


Abbildung 7.2.: J2ME Map Beispiele

7.3. YahooMaps

YahooMaps⁷ bietet vier verschiedene Varianten um Kartenmaterial verfügbar zu machen. Die ersten Drei: Flash API, AJAX API, Simple Maps API kommen nicht in Frage, da diese nicht in MIDlets implementierbar sind.

Map Image API

Mittels dieser vierten Variante lässt sich eine Kommunikation mittels HTTP-GET Anfragen für beliebige Punkte zu bestimmten Koordinaten erreichen. Als Antwort vom Server erhält man einen Link zu einem gerenderten Bitmap-Kartenausschnitt. Die Auflösung des Kartenmaterials ist für die meisten Teile der Erde sehr gut, jedoch lassen

⁶<http://www.fon.com/de>

⁷vgl. <http://maps.yahoo.com>

sich keine benachbarten Kartenausschnitte abfragen, ohne dass sich die Randteile überlappen. Folgende Abbildung zeigt zwei von der Yahoo Maps API abgefragte Kartenausschnitte⁸:



Abbildung 7.3.: Yahoo MAPS API Kartenausschnitte

7.4. Census 2000 Tiger

Census 2000 TIGER⁹ (Topologically Integrated Geographic Encoding and Referencing system) Daten sind geographische Daten des *U.S. Bureau of the Census*. Die Daten stehen kostenlos in verschiedenen standardisierten Daten zur Verfügung und bieten eine hohe Auflösung, jedoch leider nur für das Gebiet der USA.

7.5. VMAP

VMAP (Vector Map)¹⁰ ist ein weltweites vektor-basiertes topographisch-kartographisches Informationssystem der NATO, das von der NGA, der militär-geographischen Behörde der USA, herausgegeben wird. Die Karte gibt es in verschiedenen Genauigkeitsstufen:

- VMAP Level 0 entspricht etwa dem Maßstab 1:1.000.000 und ist frei verfügbar,
- VMAP Level 1 basiert auf dem Maßstab 1:250.000 und ist in weiten Teilen für militärische Zwecke reserviert.

Die VMAP-0-Daten sind in vier Gebiete (Nordamerika, Südamerika und Afrika, Europa und nördliches Asien, Südostasien und Australien) unterteilt. Jedes Datenpaket ist

⁸ vgl. Bachelorarbeit von Tim Hausmann, FH Köln, Campus Gammersbach

⁹ <http://www.census.gov/geo/www/tiger/>

¹⁰ <http://en.giswiki.org/wiki/VMAP>

gezippt gut 200 MB groß. Die VMAP ist im VPF (Vector Product Format), einer Gliederung in Verzeichnisse und Binärdateien, abgelegt. Das VPF ist im military standard MIL-STD-2407¹¹ beschrieben.

7.6. Bluemarble

Die NASA bietet unter dem Namen Bluemarble¹² kostenlose Luftbilder der Erde an, die aus dem Weltraum geschossen wurden. Jeden Monat gibt es neue Bilder für die gesamte Erde. Die Bilder werden mittels Plate Carée Projektion¹³ abgebildet. Die Daten sind im JPEG-Format und PNG-Format verfügbar. Jedoch bieten die Bilder nur geringe Auflösungen an:

- 8km/Pixel
- 2km/Pixel
- 0,5km/Pixel

7.7. Fazit

Wir haben einige Datenquellen, die zur Darstellung interaktiver Karten genutzt werden können. Dabei haben wir unter anderem die Bachelorarbeit von Tim Hausmann (*FH-Köln*) als Referenz benutzt, der das Material unter gleichen Aspekten evaluiert hat. OpenStreetMaps und Yahoo Maps bieten neben GoogleMaps das beste Kartenmaterial, lassen sich allerdings aus oben genannten Gründen für unser Projekt nicht verwenden. Die Daten von Census 2000 Tiger sind nur für die USA verfügbar und somit auch nicht einsetzbar. Die Auflösungen der Daten von VMAP0 und Bluemarble sind nicht ausreichend, um in hoher Zoomstufe zu navigieren und scheiden somit auch aus.

Somit bleibt lediglich GoogleMaps übrig, das durch die Schnittstelle von Thomas Landsburg sehr gutes Kartenmaterial für J2ME-basierende Anwendungen bereitstellt.

¹¹vgl. http://earth-info.nga.mil/publications/specs/printed/2407/2407_VPF.pdf

¹²vgl. <http://earthobservatory.nasa.gov/Newsroom/BlueMarble/>

¹³mathematische Korrektur, dass Abstände der Breitengrade gleich sind

8. Darstellung

Es gibt verschiedene Varianten, grafischen Content in einem MIDlet darzustellen. Da wir lediglich Kartenmaterial und eine Menüstruktur zur Navigation verwenden, ist es naheliegend, sämtlichen Inhalt als Vektorgrafiken darzustellen. Dies hat zum einen den Vorteil, dass durch das Einsetzen XML-basierter Kommunikationstechniken die Übertragungsrate für benötigte Inhalte geringer bleibt. Des weiteren sollte die Darstellung portierbarer sein, da die dargestellten Module jeweils skaliert werden können, um an verschiedene Displaygrößen angepasst zu werden. Dies ist in der Praxis dennoch problematisch, da verschiedene Handys nicht nur verschiedene Displays besitzen, sondern auch verschieden JSR-Pakete nutzen. Um skalare Formen darzustellen, wird mindestens das MID Profile 2.0 und außerdem das JSR226 (*Scalable 2D Vector Graphics API*) Paket benötigt.

8.1. SVG in Java

SVG (*Scalable Vector Graphics*) ist ein Standard zur Beschreibung zweidimensionaler Vektorgrafiken in der XML-Syntax. Da sich mobile Geräte hinsichtlich der Prozessorleistung, dem Arbeitsspeicher und der Farbtiefe unterscheiden, war es nötig, zwei verschiedene mobile Profile zu erstellen: SVG Tiny (SVGT) für Einstiegsmodelle und Smartphones und SVG Basic (SVGB)¹ für PDAs und andere Spitzenmodelle.

Einige Eigenschaften von SVG werden nicht in beiden mobilen Profilen unterstützt, beispielsweise Filtereffekte (Schattenwurf, Beleuchtung etc.): Im Profil SVG Basic wird nur ein Teil der Filtereffekte unterstützt, im Profil SVG Tiny sogar kein einziger. Speziell für Java-fähige mobile Geräte existiert eine Entwicklung, die lediglich auf J2ME CLDC 1.0 basiert:

TinyLine SVG

TinyLine² lässt sich als "Standalone"-SVG-Viewer auf den meisten gängigen Handys installieren, um SVG-Dateien betrachten zu können.

¹<http://www.w3.org/TR/2003/REC-SVGMobile-20030114>

²<http://www.tinyline.com/svg>

Außerdem lässt sich das TinyLine Toolkit in jedes J2ME-Projekt einbetten, um die Funktionen angepasst zu nutzen.

Das TinyLine API besteht aus folgenden zwei Paketen:

- `com.tinyline.svg`
- `com.tinyline.tiny2d`

Beide sind streng portierbar, da sie auf einem minimalen Set an CLDC 1.0 Klassen aufbauen.

Inzwischen existiert eine Web Service Entwicklung "SVG Tiny generation"³, um mittels SVG Tiny Kartenmaterial zu generieren.

Dieser "SVG generator service" ist eine Java-Entwicklung, basierend auf dem BATIK⁴ Toolkit der Apache Software Foundation und repräsentiert eine Sequenz von Serviceanfragen.

BitFlash

BitFlash⁵ ist eine kanadische Firma, die sich zur Aufgabe machte, SVG Tiny Implementierungen auf kommerzieller Basis zu konzipieren. BitFlash SVGT läuft auf Symbian-Systemen, sowie auf anderen geschlossenen Plattformen. Der BitFlash SVGT Viewer ist proprietär und wird auf marktführenden mobilen Geräten, wie dem Sharp V601SH mitgeliefert.

8.2. MIDLet Framework

Das MIDLet-Framework von LMR⁶ ist ein Grundgerüst für mobile Anwendungen. Es wird bereits in Mabber⁷, einem Instant Messenger für Handys, eingesetzt. Durch das Einsetzen des Frameworks wird ein hoher Wiedererkennungswert der GUI-Objekte garantiert.

Teil des Frameworks ist ein Preprozessor, durch den zum Einen die korrekte Darstellung und Lauffähigkeit auf verschiedenen Handys garantiert und zum Zweiten die Möglichkeit, Reverse Engineering⁸ durchzuführen, gering gehalten wird.

Der in C geschriebene Preprozessor wird in verschiedenen Java-Klassen eingebettet und

³vgl. <http://www.svgopen.org/2004/papers/WebServicesForSVGTinyMapsOnMobile/>

⁴<http://xmlgraphics.apache.org/batik>

⁵<http://www.bitflash.com>

⁶<http://www.lmr.khm.de>

⁷vgl. <http://www.mabber.de/>

⁸Bezeichnet den Vorgang, aus einem bestehenden, fertigen System Strukturen, Zustände und Verhaltensweisen zu extrahieren.

beim Kompilieren dieser gestartet. Der Preprozessor verwendet diverse *.inc*-Dateien, welche Parameter für Auflösung, Koordinaten, Sprache usw. verschiedener Handys beinhalten.

In der *gfx.inc*-Datei wird zu jedem einzelnen Objekt der Oberfläche das dazugehörige Bild zugewiesen.

Beispielsweise benutzt das DropDown-Objekt ein Bild (*siehe rechts*), das in dem *res_S40_128x128*-Verzeichnis gespeichert ist. Dieses wird wie folgt in der *gfx.inc*-Datei verwendet.



```
#define DROPDOWN "/selbxbtn.png"
```

Durch die Datei *gfx_S40_128x128.inc*, die von *gfx.inc* verwendet wird, werden die Koordinaten des jeweiligen Objektes auf der Oberfläche definiert.

```
#define DROP_DOWN_X    10
#define DROP_DOWN_Y    50
#define DROP_DOWN_WIDTH 97
#define DROP_DOWN_HEIGHT 10
```

Die Datei *language_en_EN.inc* beschreibt die Bezeichnungen der einzelnen Oberflächenobjekte, welche wiederum von *language.inc* verwendet wird. Dadurch lässt sich das System um beliebige Sprachmodule erweitern. In dem Beispiel DropDown werden die textbasierten Ausgaben, die als Auswahlmöglichkeit existieren, wie folgt beschrieben:

```
#define DROP_DOWN_PRESETS "My Location, HotSpots, Address"
```

In der Datei *global.inc* werden sämtliche Zustände des Programms als ganze Zahlen kodiert, um das fertige System möglichst klein zu halten und die Rückgewinnung des Quellcodes (*Reverse Engineering*) zu unterbinden.

Dies soll folgendes Beispiel verdeutlichen:

```
#define SCREEN_HotSpotNew 24
#define SCREEN_HotSpotEddit 25
#define SCREEN_HotSpotDelete 26
#define SCREEN_FriendsNew 29
#define SCREEN_SPLASH 0
#define INIT 1
#define GUI_TOGGLE_MAINMENU1 101
#define GUI_TOGGLE_MAINMENU2 102
#define USE_VIBRATION 2
#define SOUND_VOL 3
```


In der Datei *platform.inc* werden für verschiedene Handys unterschiedliche Maße für die Größe des Telefons festgelegt.

Für das Beispiel der Handy-Serie **Nokia S40 (Auflösung: 128x128)** würde die Initialisierung wie folgt lauten:

```
#ifdef S40_128x128
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 128
#define SCREEN_WIDTH_ALL 128
#define SCREEN_HEIGHT_ALL 128
#endif
```

8.3. Fazit

Abschließend bleibt über die Darstellung auf mobilen Geräten zu sagen, dass es grundsätzlich kompliziert ist, grafische Inhalte auf möglichst vielen Geräten einheitlich darzustellen. Darüber hinaus befindet sich die Entwicklung der SVG-Technik noch in den Anfängen. Da die betreuende Firma dieser Arbeit - LMR bereits eine Variante entwickelt hat, die eine breite Masse von verschiedenen Mobilfunkgeräten erreicht, entschieden wir uns, auf diesem Framework aufzubauen.

9. Software

In diesem Kapitel stellen wir die verwendete Software vor. Für das gesamte Projekt wurden ausschließlich frei verfügbare Anwendungen verwendet.

9.1. Apache Tomcat



Abbildung 9.1.: Apache Tomcat

Apache Tomcat¹ ist ein in Java geschriebener Servlet-Container der Apache Software Foundation, der mithilfe des JSP-Compilers Jasper auch JavaServer Pages in Servlets übersetzen und ausführen kann.

In diesem Projekt wird Tomcat eingesetzt, um eine Serveranwendung, die als Servlet implementiert ist, zu beherbergen.

9.2. Xampp



Abbildung 9.2.: Xampp

Xampp² ist eine Distribution von Apache, MySQL, PHP und Perl. Wir setzen dieses Projekt besonders wegen der effizienten Handhabung von MySQL-Datenbanken ein. Mittels des im Paket enthaltenen PHPMyAdmin lassen sich Tabellen der Datenbank und deren Attribute einfach und schnell konfigurieren.

¹vgl. <http://tomcat.apache.org>

²vgl. <http://www.apachefriends.org>

10. ÖPNV - Öffentlicher Personennahverkehr

Eine wesentliche Funktion unseres mobilen Navigationssystems stellt die Integration von Informationen öffentlicher Verkehrsmittel dar. Dies gilt als wichtiger, zielgruppenorientierter Punkt, da man als Tourist in einer fremden Stadt meist nicht über Informationen des lokalen Schienennetzes verfügt. Auch die als Bestandteil der Arbeit durchgeführte Probandenbefragung ergab, dass Personen die sich neu in einer Stadt befinden, sich zielgerichtet zum Streckenplan des jeweiligen Schienenbetreibers begeben.

10.1. Deutsche Bahn

Eine optimale Informationsbreite stellt die Datenbank der Deutschen Bahn¹ dar, die neben deutschlandweiten, auch über regionale Datenbestände verfügt.

Leider ergaben unsere Recherchen und Anfragen keine erfolgreiche Korrespondenz.

10.2. KVB

Die Kölner Verkehrs-Betriebe AG² ist eines der größten öffentlichen Verkehrsunternehmen der Bundesrepublik Deutschland. Mit Stadtbahnen und Bussen bietet sie ein leistungsstarkes Angebot im Stadt- und Regionalverkehr.

Da die KVB mittels "Handyinfo" bereits eine Option bietet, mobile Anfragen an die Fahrplanauskunft zu stellen, eruierten wir verschiedene Möglichkeiten zur Kontaktaufnahme.

Der Herausgeber dieser Software ist die "Sparkasse KölnBonn", welche uns an die ProsystemsIT GmbH, Bereich Anwendungsentwicklung vermittelte. Der Ansprechpartner

¹<http://www.db.de>

²vgl. <http://www.kvb-koeln.de>

wies jedoch die Zuständigkeit zur Sparkasse zurück. Die Zuständige der Sparkasse ermittelte den zuständigen Chefentwickler der damals entwickelten Software des Unternehmens der Sparkassen-Finanzgruppe Rheinlandmobil³.

Nach Schilderung des Anliegens, eine Schnittstelle zur KVB in ein Navigationssystem zu implementieren erfuhren wir, dass die Serverkomponente von "Handyinfo" lediglich ein Servlet ist, dass eine HTTP-POST Anfrage an die Webseite "<http://www.kvb-koeln.de>" richtet und entsprechend die Antwort so parst⁴, dass das Resultat im Handy-Client darstellbar ist.

Der zuständige Chefentwickler bot uns an, Rücksprache mit zuständigen Personen der KVB zu führen, um dieses Servlet zur Verwendung frei zu geben. Bis zum Ende dieser Arbeit kam leider keine Zusammenarbeit zustande und somit kann diese Option nicht im Prototyp implementiert werden.

³<http://www.rheinlandmobil.de>

⁴Ein Parser ist ein Computerprogramm, das für die Zerlegung und Umwandlung einer beliebigen Eingabe in ein für die Weiterverarbeitung brauchbares Format zuständig ist

Teil III.

Vorgehensmodell

Die Entwicklung eines Informationssystems bzw. Anwendungssystems ist ein komplexer Prozess. Für die Bewältigung von Projekten werden Vorgehensmodelle zu Hilfe genommen, die festlegen, in welchen Schritten vorzugehen ist. Generell beschreibt jedes Vorgehensmodell die Folge aller Aktivitäten, die zur Durchführung eines Projekts erforderlich sind. Vorgehensmodelle speziell für die Systementwicklung von Informations- bzw. Anwendungssystemen geben an, wie die Prinzipien, Methoden, Verfahren und Werkzeuge der Software-Entwicklung einzusetzen sind.

Die Modelle können in klassische und agile Vorgehensweisen unterschieden werden. Der klassische Ansatz wird vor allem bei großen Projekten eingesetzt, bei denen während der Entwicklung wenige Veränderungen bei den Anforderungen und in der Technik auftreten. In den Mittelpunkt rückt hier vor allem die Dokumentation des Systems. Generell kann der Ansatz als eher kostspielig angesehen werden. Das Gegenstück dazu bildet die agile Vorgehensweise, die eher für kleinere Projekte bei Teamarbeit eingesetzt wird.

Die Entwicklung muss flexibel auf Veränderungen eingehen können. Anstatt sich an der Dokumentationen zu orientieren, rückt die Benutzerfreundlichkeit des Systems in das Zentrum der Betrachtung. Die gewonnene Struktur des klassischen Modells geht bei der agilen Vorgehensweise verloren und die Aufbau wirkt eher chaotisch.

Ein Beispiel für die klassische Methode ist das Wasserfallmodell, welches in diesem Projekt Anwendung findet.

Das Wasserfallmodell

Das Wasserfallmodell ist ein bewährtes Modell zur strukturierten Software-Entwicklung. Es nähert sich systematisch und sequentiell an den Prozess der Softwareentwicklung an, indem der Prozess in mehreren Stufen oder Phasen unterteilt wird. Jede einzelne Ebene lässt sich gemäß den Vorstellungen des Auftraggebers und der zuvor erstellten Spezifikation validieren, überprüfen und testen. Diese Phasen werden in ihrer richtigen Reihenfolge und in ihrer vollen Breite abgearbeitet. Das sequentielle Vorgehen bei der Abarbeitung der Phasen wird auf jeder Stufe mit einem Meilenstein abgeschlossen. Erst wenn der Meilenstein erreicht wurde, kann mit der nächsten Phase begonnen werden. Wie bei dem klassischen Ansatz beschrieben, orientiert sich auch das klassische Wasserfallmodell an Dokumenten und es kann als stufenweise bezeichnet werden. Gegründet wurde das klassische Wasserfallmodell auf dem Stufenmodell von Herbert D. Benington. Folgende Abbildung zeigt den Ablauf eines Wasserfallmodells:

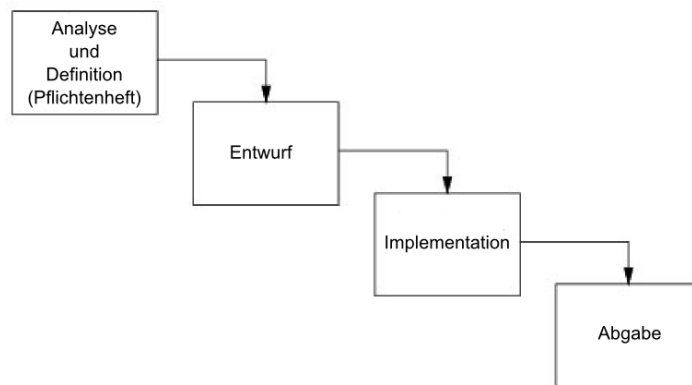


Abbildung 10.1.: Das klassische Wasserfallmodell

Mit dem linearen Phasenmodell eng verwoben ist ein weiteres grundlegendes Problem der Software-Entwicklung: die *unzureichende Benutzerbeteiligung*. Zwar werden die Anforderungen an die zu entwickelnde Software häufig noch unter Beteiligung der Anwender definiert, dann kommt die Beteiligung aber ins Stocken und wird erst nach Abschluss der Programmierarbeiten wieder aufgenommen. Geeignete Methoden der Anwenderbeteiligung fehlen zumeist völlig.

Wenn die Anwenderbeteiligung nicht funktioniert, fällt die Verantwortung für die Software-Ergonomie den Software-Entwicklern zu, die allein auf Grund ihres fehlenden Fachbezugs diese Aufgabe nicht erfüllen können.

Was ist wichtig, damit es besser klappt?

Wichtige Ansatzpunkte, wie es besser zu machen ist, finden sich in der Norm DIN EN ISO 13407. Sie beschreibt einen benutzerorientierten Prozess der Software-Entwicklung, der auf das Ziel der Gebrauchstauglichkeit (usability) ausgerichtet ist.

Auf Grund des unterstellten Zusammenhangs zwischen Produkt- und Prozessqualität soll die DIN EN ISO 13407 über eine Verbesserung der Prozessqualität einen entscheidenden Beitrag zur Verbesserung der Produktqualität im Sinne von Gebrauchstauglichkeit leisten.

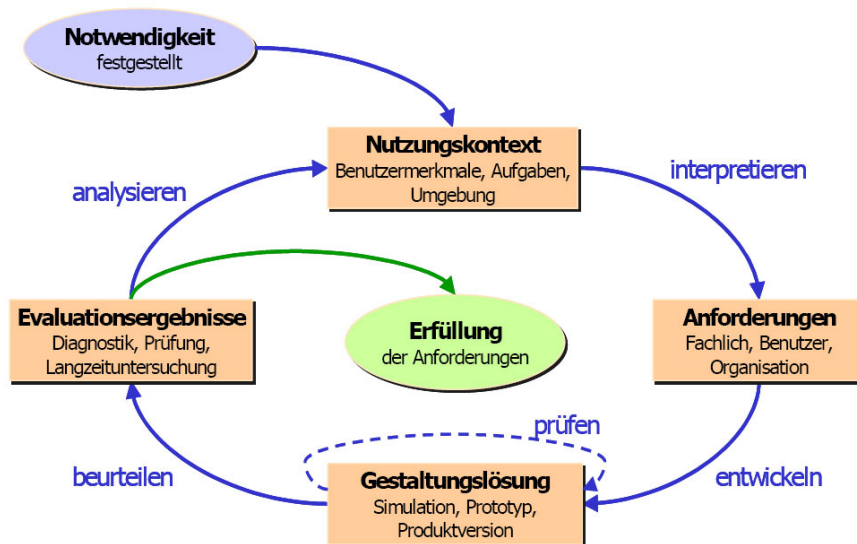


Abbildung 10.2.: ISO 13407

Um bei der Konzeption unseres Projektes eine möglichst hohe Gebrauchstauglichkeit zu erzielen, erweiterten wir das klassische Wasserfallmodell um den benutzerzentrierten Ansatz der ISO 13407.

11. Analyse und Definition

Bei Softwareprojekten, die im Auftrag eines Auftraggebers entstehen sollen, wird die *vom Auftraggeber festgelegte Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines Auftrages* in einer Anforderungsspezifikation (auch *Lastenheft*) beschrieben.

Das Lastenheft beschreibt in der Regel also, was und wofür etwas gemacht werden soll.

Da im Rahmen dieser Arbeit kein Arbeitgeber im eigentlichen Sinn existiert, entfällt dieser Schritt.

Das Pflichtenheft

Die Aufgabe des Pflichtenheftes ist es, zu beschreiben, was die zu entwickelnde Software für den Anwender leisten soll.

11.1. Zielbestimmungen

Welche Musskriterien, Wunschkriterien, Abgrenzungskriterien sind erforderlich?

Das System stellt einen Dienst für mobile Endgeräte dar, der das Benutzen von einem Navigationssystem für *Fußgänger*, *Autofahrer* und *Fußgänger in Verbindung mit öffentlichen Verkehrsmitteln* ermöglicht.

11.1.1. Musskriterien

- Der Benutzer-Account
 - Der Benutzer kann sich selbst am System registrieren.
 - Der Benutzer kann sich am System anmelden und vom System abmelden.
 - Der Benutzer kann seine Kennung anfordern.
 - Der Benutzer kann seine persönlichen Daten einsehen und ändern, sowie deren Sichtbarkeit beeinflussen.
 - Der Benutzer kann sein persönliches Profil einsehen, sowie seinen aktuellen Standort.
 - Der Benutzer kann das Profil anderer Benutzer einsehen, soweit dieses als sichtbar eingestellt ist.
 - Der Benutzer kann verschiedene Points of Interest ansehen, anlegen und ändern.
 - Der Benutzer kann seine Benutzungseinstellung nach eigenem Bedarf konfigurieren.
 - Der Benutzer kann sich eine einfache Route berechnen und anzeigen lassen.
 - Die Benutzer können mittels des Systems Informationen austauschen.
 - Der Benutzer kann sich eine Tour von mehreren Points of Interest erstellen und anzeigen lassen.
 - Der Benutzer kann nach einem Point of Interest oder eine Adresse im System suchen.

- Der Benutzer kann verschiedene Points of Interest als seine Favoriten speichern.
- Das System
 - Das System kann am PC oder am Handy benutzt werden.
 - Das System muss mit dem Internet in Verbindung sein.
 - Das System beinhaltet einen Routenberechner.
 - Das System beinhaltet verschiedene Points of Interest.
 - Das System hat eine Suchfunktion.
 - Das System beinhaltet eine Help-Seite, die jeder Zeit für die Benutzer verfügbar ist.
- Der Administrator
 - Der Administrator konfiguriert die Betriebsparameter des Systems.
 - Der Administrator sichert die Datenbank.

11.1.2. Wunschkriterien

- Der Benutzer kann nach der Abmeldung die zuletzt angesehenen Points of Interest auf der Karte für die nächste Anmeldung beibehalten.
- Für den Benutzer soll die Auswahl der Sprache möglich sein.

11.1.3. Abgrenzungskriterien

- Das System muss unbedingt eine Verbindung ins Internet haben.
- Das System eignet sich nur für Java-fähige Handys.
- Die Benutzung des System ist erst sinnvoll, wenn mehrere Benutzer das System benutzen und damit viele Points of Interest zur Verfügung stehen.

11.2. Produkteinsatz

Welche Anwendungsbereiche (Zweck), Zielgruppen (Wer mit welchen Qualifikationen), Betriebsbedingungen (Betriebszeit, Aufsicht)?

11.2.1. Anwendungsbereiche

Touristen verwenden diesen Dienst als Navigationshilfe mittels der oben angegebenen Kriterien. Diese Plattform soll dem Einzelnen eine bessere Kommunikation in einer fremden Umgebung ermöglichen, um schneller und effizienter zum Ziel zu gelangen.

11.2.2. Benutzeranforderungen

Es werden Basiskenntnisse in Internetnutzung vorausgesetzt. Ebenso sollte der Einzelne mit dem Umgang von mobilen Endgeräten vertraut sein.

11.2.3. Betriebsbedingungen

Diese Software wird möglichst plattformunabhängig entwickelt. Jedoch bedient sie sich folgender Spezifikationen, die das System unterstützen muss:

- MIDP 2.0 mit Java Bluetooth API (JSR-82)
- CLDC 1.1
- eine Verbindung zum Internet mittels GPRS (langsam) oder UMTS (schnell)

11.3. Produktumgebung

Welche Software, Hardware und Orgware wird benötigt?

Das Produkt ist konzipiert für die Verwendung auf Java-basierten portablen Endgeräten, sofern folgende Produktumgebung vorhanden ist:

11.3.1. Software

- Client
 - **MIDP 2.0**-Profil der J2ME
 - **CLDC 1.1**-Profil zur Fließkommaberechnung
- Server
 - **MySQL**-Datenbank
 - **Apache**-Server
 - **Servlet** zur KVB-Anfragenübersetzung
 - **Map**-Server *GoogleMaps, Ask.com, Yahoo.com, VirtualEarth*

11.3.2. Hardware

- Client
 - Anbindung zum Internet
- Server
 - Internetfähige Server
 - Rechner, der die Ansprüche der o.g. Server-Software erfüllt
 - Ausreichend Rechen- und Festplattenkapazität

11.3.3. Orgware

- Gewährleistung der permanenten Internetanbindung
- Administrator muss den Internetdienst starten und die Betriebsparameter konfigurieren

11.4. Produktfunktionen

Was leistet das Produkt aus Benutzersicht?

11.4.1. Benutzerfunktionen

Benutzer-Kennung

Ein im System registrierter Benutzer kann das System erst nutzen, wenn er angemeldet ist.

/F0010/ *Registrieren*: Ein beliebiger Internet-Benutzer kann sich über die Start- bzw. Login-Seite des Systems schnell und bequem registrieren. Zum Registrieren sind mindestens folgende Angaben erforderlich:

- gewünschte **Kennung**
 - gewünschter **Benutzername**
 - gewünschtes **Passwort**
- private **eMail-Adresse**

Die Registrierung ist erfolgreich, wenn der *Benutzername* und die *eMail-Adresse* innerhalb des Systems jeweils eindeutig sind. Die *eMail-Adresse* wird auf ihre Gültigkeit geprüft.

Mit dem erfolgreichen Abschließen des Registrierungsvorgangs ist der neue Benutzer am System angemeldet, zudem erhält der Benutzer automatisch via *eMail* vom System seine aktuelle Kennung.

/F0020/ *Anmelden*: Ein bereits registrierter Benutzer kann sich über die Start- bzw. Login-Seite des Systems schnell und bequem anmelden (*login*). Dazu ist seine Kennung erforderlich:

- sein **Benutzername**
- sein **Passwort**

Alternativ zum *Benutzernamen* kann der Benutzer seine *eMail-Adresse* angeben.

/F0030/ *Abmelden*: Der angemeldete Benutzer kann sich jeder Zeit wieder vom System **abmelden** (*logout*).

/F0040/ *Kennung anfordern*: Falls ein bereits registrierter Benutzer seine Kennung oder sein **Passwort vergessen** haben sollte, so kann er seine korrekte Kennung über die Start- bzw. Login-Seite des Systems anfordern. Dem Benutzer wird unter Angabe

- seines **Benutzernamens** oder
- seiner **eMail-Adresse**

seine vollständige Kennung automatisch via *eMail* vom System zugesendet.

/F0050/ Passwort ändern: Der angemeldete Benutzer kann das Passwort seiner Kennung ändern. Das neue Passwort muss zweimal angegeben werden, wobei sich diese Angaben nicht unterscheiden dürfen. Nach erfolgreicher Änderung des Passwortes erhält der Benutzer automatisch via *eMail* vom System seine aktuelle Kennung.

Der Benutzer kann seinen *Benutzernamen* nicht ändern.

Im Folgenden sei der Benutzer stets am System angemeldet.

Persönliche Daten

Der Benutzer verfügt über persönliche Daten (*siehe /D0110/*), die er frei gestalten kann.

/F0110/ Anzeige der eigenen, persönlichen Daten: Der Benutzer kann sich seine persönlichen Daten vom System **vollständig anzeigen** lassen.

/F0120/ Ändern der eigenen, persönlichen Daten: Der Benutzer kann seine persönlichen Daten aktualisieren bzw. **ändern**.

/F0130/ Sichtbarkeit der eigenen, persönlichen Daten: Der Benutzer kann jeden einzelnen Eintrag seiner persönlichen Daten für die Spielgemeinschaft auf **sichtbar** bzw. **unsichtbar** setzen.

/F0140/ Anzeige der persönlichen Daten anderer Benutzer: Der Benutzer kann sich von anderen Benutzern die persönlichen Daten anzeigen lassen; dabei können auf unsichtbar gesetzte Einträge nicht gesehen werden.

Im Gegensatz zu */F0110/* kann der Benutzer seine eigenen, persönlichen Daten auch auf diese Weise anzeigen lassen.

Persönliche Konfiguration

Der Benutzer kann persönliche Einstellungen *siehe /F12??/* bezüglich der Art der Verwendung vornehmen.

/F0210/ Anzeige der persönlichen Konfiguration: Der Benutzer kann sich alle einstellbaren Werte seiner persönlichen Konfiguration **anzeigen** lassen.

/F0220/ *Ändern der persönlichen Konfiguration*: Der Benutzer kann alle einstellbaren Werte seiner persönlichen Konfiguration **ändern** oder die voreingestellte Konfiguration wiederherstellen.

/F0230/ *Speichern der persönlichen Konfiguration*: Der Benutzer kann seine persönliche Konfiguration **speichern**.

/F0240/ *Löschen der persönlichen Konfiguration*: Der Benutzer kann bereits gesicherte Konfigurationen entfernen.

11.4.2. Hilfefunktionen

Dem angemeldeten Benutzer wird eine Hilfe geboten, um die Funktionen des Systems zu erklären.

/F0310/ *Hilfe bei Erstbenutzung*: Der angemeldete Benutzer bekommt bei erstmaliger Anmeldung eine Navigationshilfe angezeigt.

/F0320/ *Hilfe zum System*: Der angemeldete Benutzer erhält allgemeine Hinweise zur Systembenutzung.

11.4.3. Standortfunktionen

persönliche GPS-Funktionen

/F0410/ *aktuellen Standort anzeigen*: Dem Benutzer wird der aktuelle GPS-Standort auf der Karte angezeigt.

/F0420/ *aktuellen Standort übermitteln*: Der Benutzer kann einem anderen angemeldeten Benutzer seine eigenen GPS-Daten übermitteln.

/F0430/ *aktuellen Standort als Point of Interest hinzufügen*: Der Benutzer kann mit seinen aktuellen GPS-Koordinaten einen neuen Point of Interest anlegen.

Favoriten-Funktionen

/F0510/ *Favoriten hinzufügen*: Der Benutzer kann bestehende Points of Interest zu seiner Favoriten-Liste hinzufügen.

/F0520/ *Favoriten bearbeiten*: Der Benutzer kann bestehende Favoriten bearbeiten.
siehe /F0620/

/F0530/ *Favoriten löschen*: Der Benutzer kann bestehende Points of Interest aus seiner Favoriten-Liste entfernen.

/F0540/ *Favoriten anzeigen*: Der Benutzer kann bestehende Points of Interest aus seiner Favoriten-Liste auf der Karte anzeigen.

Point of Interest-Funktionen

/F0610/ *Point of Interest erstellen*: Der angemeldete Benutzer kann mittels seiner GPS-Koordinaten oder einer Adresse einen neuen Point of Interest anlegen. Diesem Point of Interest kann man Informationen hinzufügen, wie *Name*, *Telefonnummer*, *allgemeine Informationen*.

Ihm wird eine Auswahl an Kategorien aufgelistet zu dem er eine angeben **muss**. Des weiteren kann ein Point of Interest als *private*, *public*, *friends* gespeichert werden.

Es besteht die Möglichkeit ein Foto hinzuzufügen.

/F0620/ *Point of Interest bearbeiten*: Der angemeldete Benutzer kann bereits existierende Points of Interest bearbeiten.

/F0630/ *Point of Interest löschen*: Der angemeldete Benutzer kann selbst erstellte Points of Interest löschen.

/F0640/ *Point of Interest anzeigen*: Der Benutzer kann bestehende Points of Interest auf der Karte anzeigen lassen.

Friends-Funktionen

Ein **Friend** wird im System als Point of Interest-Objekt betrachtet.

/F0710/ *anderen Benutzer zu Friends hinzufügen*: Mittels des Benutzernamens eines anderen Nutzers des Systems, kann dieser zur *Friends*-Liste hinzugefügt werden.

/F0720/ *anderen Benutzer entfernen*: Ein Benutzer, der als *Friend* geführt wird, kann aus der *Friends*-Liste entfernt werden.

11.4.4. Route-Funktionen

einfache Route

/F0810/ *Route berechnen*: Der Benutzer kann eine Route berechnen lassen, von einem *Favoriten*, einem *Point of Interest*, einer *Adresse* oder einem aktuellen *GPS-Standpunkt* zu einem *Favoriten*, einem *Point of Interest* oder einer *Adresse* und bekommt diese auf der Karte dargestellt.

Die Route wird während der Bewegung sekundlich neu berechnet.

/F0820/ *Öffentliche Verkehrsmittel*: Der Benutzer hat die Möglichkeit die Benutzung von öffentlichen Verkehrsmitteln in seine Routenberechnung einzubeziehen. Die Haltestellen werden dem Benutzer als dynamische Points of Interest angezeigt. In der Beschreibung dieser speziellen Points of Interest stehen Informationen, wie *Linie, Abfahrtszeit, Richtung* und *Kosten*.

/F0830/ *Dauer*: Der Benutzer erhält Informationen über die geschätzte Dauer der Route.

/F0840/ *Entfernung*: Der Benutzer erhält Informationen über die Entfernung vom Start- bis zum Zielpunkt.

/F0850/ *Richtungsfunktion*: Der Benutzer bekommt die Richtung der Luftlinie von seinem aktuellen Standpunkt zum Ziel als *Pfeil* dargestellt.

Tour-Funktion

/F0910/ *Tour erstellen*: Der Benutzer kann aus verschiedenen existierenden *Favoriten, Points of Interest* oder *Adressen* eine Tour zusammenstellen.

/F0920/ *Tour bearbeiten*: Der Benutzer kann einer existierenden Tour *Favoriten, Points of Interest* oder *Adressen* hinzufügen, oder aber auch entfernen. Des weiteren lassen sich Positionen einzelner Punkte in der Reihenfolge verschieben.

/F0930/ *Tour löschen*: Der Benutzer kann Touren löschen.

/F0940/ *Tour anzeigen*: Der Benutzer kann sich eine existierende Tour auf der Karte anzeigen lassen.

11.4.5. Suchfunktionen

/F1010/ *Adresse suchen*: Der Benutzer kann nach einer Adresse suchen und bekommt diese auf der Karte angezeigt.

/F1020/ *Point of Interest suchen*: Der Benutzer kann nach einem Begriff suchen und angeben, ob dieser Point of Interest in der näheren Umgebung der *GPS-Position*, einem *Kartenausschnitt*, einem *Favoriten, Point of Interest* oder einer *Adresse* sein soll.

11.4.6. Kartenfunktionen

/F1110/ *Scrollfunktion*: Der Benutzer kann sich mittels der Navigationstasten frei auf der Karte bewegen.

/F1120/ *Kurzinfo-Funktion*: Scrollt der Benutzer auf einen Point of Interest, wird ihm eine Kurzinformation bezüglich dieses Points of Interest angezeigt.

/F1130/ *Informationsfunktion*: Klickt der Benutzer auf einen Point of Interest, so wird ihm eine Informationsseite mit *Bild*, *Telefonnummer* und weiteren Informationen angezeigt.

/F1140/ *Anruffunktion*: Ist dem Point of Interest eine Telefonnummer zugewiesen, so kann der Benutzer diese direkt anrufen.

/F1150/ *Kartenmodus-Funktion*: Der Benutzer kann die Ansicht zwischen Satelliten- und Kartenansicht wechseln.

11.4.7. System-Einstellungen

Im System können mehrere Standardeinstellungen gespeichert werden.

/F1210/ *Art der Fortbewegung*: Der Benutzer kann festlegen, ob er *Fußgänger*, *Radfahrer* oder *Autofahrer* ist.

/F1220/ *Angezeigte Point of Interest-Kategorien*: Der Benutzer kann die dem System zur Verfügung gestellten Kategorien für Points of Interest festlegen.

/F1230/ *Anzeige des Datenvolumens*: Der Benutzer kann sich die Menge der übertragenen Daten anzeigen lassen.

11.4.8. Administratorfunktionen

Der Administrator verfügt über alle Benutzerfunktionen, und kann darüberhinaus die Eigenschaften des Systems konfigurieren. Zudem kann der Administrator Benutzer aus dem System verbannen.

Systemverwaltung

/F1310/ *öffentliche Points of Interest*: Der angemeldete Administrator kann alle öffentlichen Points of Interest bearbeiten und löschen.

Benutzerverwaltung

/F1410/ Verbannung eines Benutzers: Der angemeldete Administrator kann Benutzer vom System entfernen.

Dabei werden die als *private* geführten Points of Interest dieses Benutzers gelöscht. Die eMail-Adresse des Benutzers wird für erneutes Anmelden gesperrt.

11.5. Produktdaten

Was speichert das Produkt (langfristig) aus Benutzersicht?

Jeder Punkt */D????/* stellt im Prinzip einen Datensatz dar.

Benutzerspezifische Daten:

/D0110/ persönliche Daten: Alle Informationen zu einem Benutzer

- **BenutzerID** (*eindeutig*)
- **Benutzername** (*eindeutig*)
- **Passwort** (*verschlüsselt*)
- **eMail-Adresse** (*eindeutig, gültig*)
- **Kurzinfo**
- **Vorname**
- **Nachname**
- **Telefonnummer**
- **Straße**
- **Hausnummer**
- **Ort**
- **Land**
- **Informationen**
- **Bild**

Hilfebezogene Daten:

/D0310/ Hilfedaten: Daten bezüglich der Hilfeseite

- **Erstbenutzung** (*true, beim ersten Start*)

Standortbezogene Daten:

/D0510/ Favoritendaten: Favoriten sind Points of Interest, die zusätzlich in einer eigenen Tabelle registriert werden.

- **Point of InterestID** (*eindeutig*)

/D0610/ Point of Interest-Daten: Daten bezüglich Hot-Spot-Funktionen.

- **Point of InterestID** (*eindeutig*)

- **Koordinaten** (*Längen- und Breitengrad*)
- **Sichtbarkeit**
- **Kategorie**
- **Kurzinfo**
- **Name**
- **Straße**
- **Hausnummer**
- **Ort**
- **Land**
- **Informationen**
- **Bild**
- **Telefonnummer**

/D0710/ Friends-Daten: Friends sind Objekte, die wie Points of Interest behandelt werden.

- **FriendBenutzerID** (*eindeutig*)

Routenbezogene Daten:

/D0910/ Tour-Daten: Touren lassen sich dauerhaft speichern.

- **Name**
- **Start-Point of InterestID**
- **Ziel-Point of InterestID**
- **Vector-List** *weitere Routenstationen(Points of Interest)*

Systemeinstellungen:

/D1210/ Systemdaten: Der Benutzer kann Einstellungen zur Navigation und Anzeige speichern.

- **FortbewegungsmittelID** (*Fußgänger/Radfahrer/Autofahrer*)
- **Point of Interest-Kategorien Liste**(*Auswahl bestimmter Kategorien*)
- **Datenvolumenanzeige** (*Boolean*)

11.6. Produktleistungen

Welche zeit- und umfangsbezogenen Anforderungen gibt es?

/L200/ *Akkumulation:* Bei fehlererzeugenden Eingaben erhält der Benutzer als Fehlermeldung eine Auflistung aller eingegebenen Fehler.

/L210/ *Toleranz:* Bei fehlererzeugenden Eingaben muss der Benutzer die Möglichkeit haben, eine Korrektur der Eingabedaten vorzunehmen, ohne bereits vorhandene richtige Eingaben wiederholen zu müssen.

11.7. Benutzungsoberfläche

11.7.1. Dialogstruktur

Im Folgenden wird die grobe Dialogstruktur einer fehlerfreien bzw. konfliktfreien Benutzung des Systems gezeigt. Fehlereingaben haben in der Regel einen Rücksprung auf die Ausgangsseite mit einer akkumulierten Fehlermeldung zur Folge.

Startseite

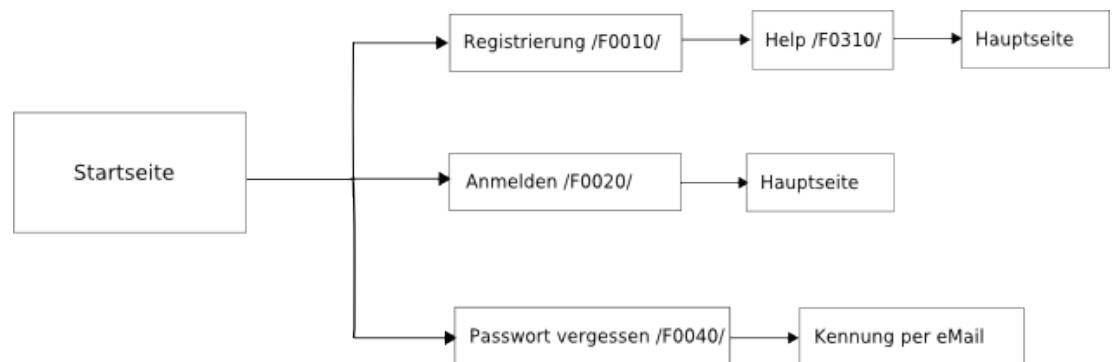


Abbildung 11.1.: Dialogstruktur der Startseite

Menü

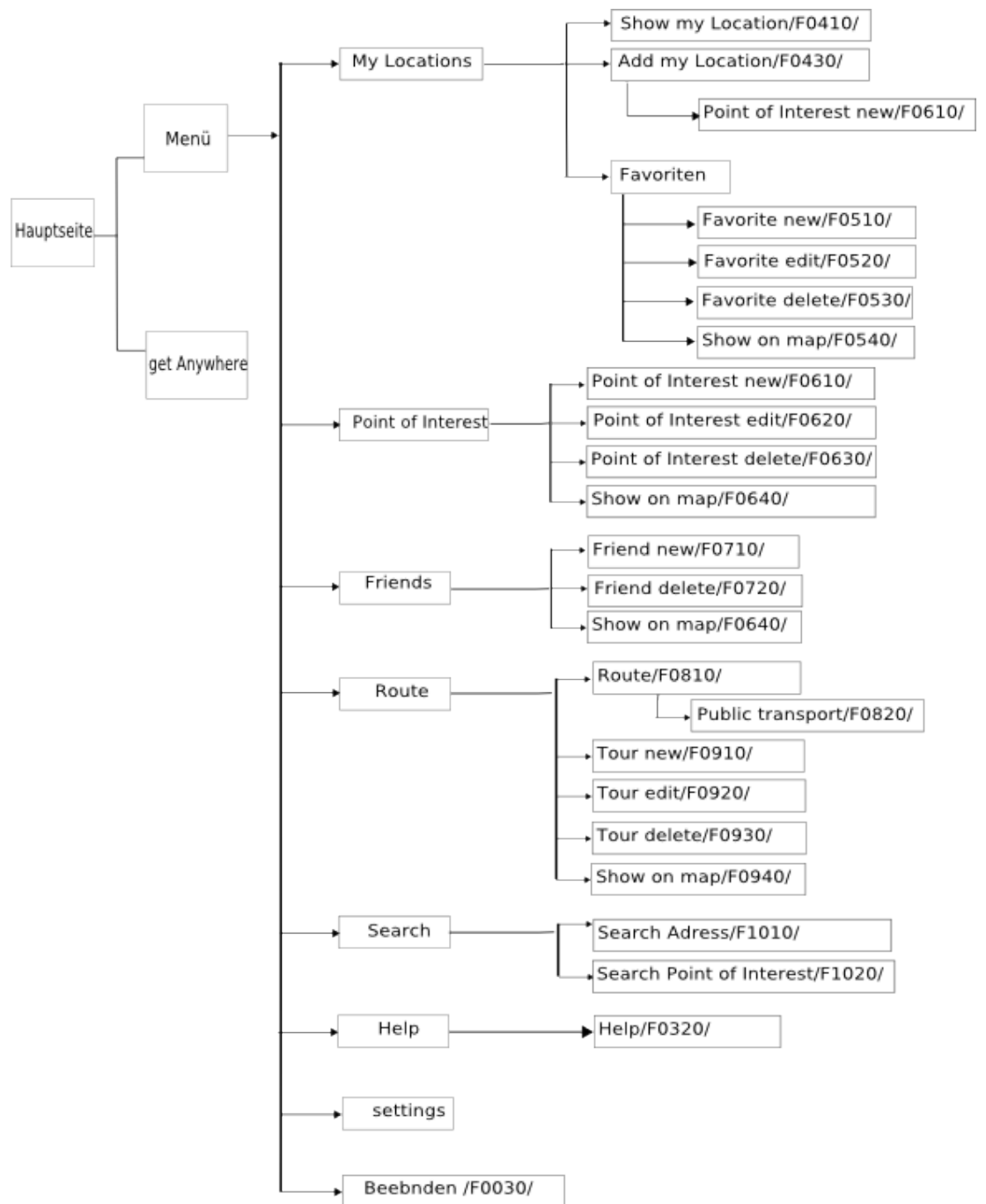


Abbildung 11.2.: Dialogstruktur der Hauptseite

Settings

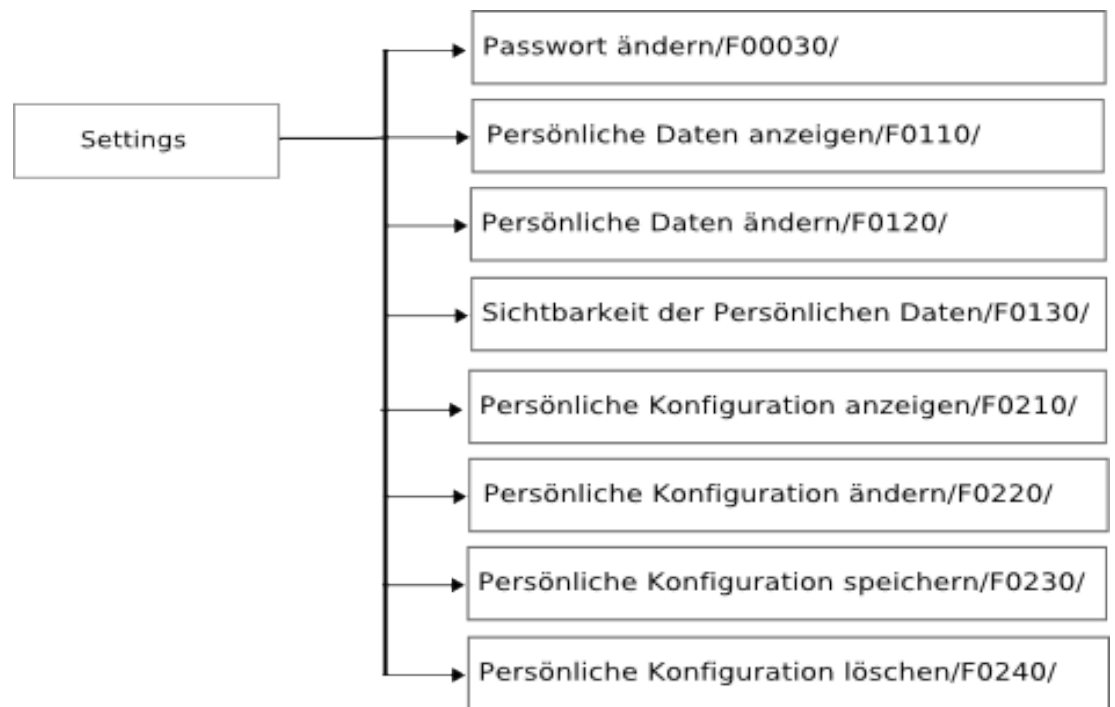


Abbildung 11.3.: Dialogstruktur der Konfigurationsseite

11.8. Qualitätszielbestimmungen

Auf welche Qualitätsanforderungen (Zuverlässigkeit, Robustheit, Benutzungsfreundlichkeit, Effizienz, ...) wird besonderen Wert gelegt?

	sehr wichtig	wichtig	weniger wichtig	unwichtig
<i>Robustheit</i>	X			
<i>Zuverlässigkeit</i>	X			
<i>Korrektheit</i>	X			
<i>Benutzungsfreundlichkeit</i>	X			
<i>Mobilität</i>	X			
<i>Effizienz</i>		X		
<i>Portierbarkeit</i>			X	
<i>Kompatibilität</i>			X	

11.9. Globale Testszenarien und Testfälle

Was sind typische Szenarien, die das Produkt erfüllen muss?

Jede Produktfunktion */F????/* wird anhand von konkreten Testfällen */T????/* überprüft.

11.9.1. Benutzerfunktion

Benutzer-Kennung

/T0010/ Registrieren: Herr Tim Testmann registriert sich mit dem gewünschten Benutzernamen *testmann* und dem Passwort *testtest* und der eMail-Adresse *tim@testmann.de* am System.

/T0020/ Anmelden: Tim Testmann meldet sich unter Benutzung seines Benutzernamens und Passwortes am System an.

/T0030/ Abmelden: Tim Testmann meldet sich vom System ab, sofern er das System beendet.

/T0040/ Kennung anfordern: Beate Betamuster hat ihre Kennung vergessen und fordert unter Angabe ihrer eMail-Adresse oder ihres Benutzernamens ihre Kennung an.

/T0050/ Passwort ändern: Beate Betamuster ändert ihr Passwort *betabeta* in *testbeta* ab. Sie gibt das Passwort zweimal an und erhält eine Bestätigungs-eMail.

Persönliche Daten

/T0110/ Anzeige der eigenen persönlichen Daten: Tim lässt sich sein Profil anzeigen.

/T0120/ Ändern der eigenen persönlichen Daten: Tim ändert seine Telefonnummer in seinem eigenen Profil.

/T0130/ Ändern der eigenen Sichtbarkeit: Tim ändert seinen Status auf sichtbar, damit die andere Benutzer ihn sehen können.

/T0140/ Anzeige der persönlichen Daten anderer Benutzer: Tim lässt sich die persönlichen Daten von Beate anzeigen.

Persönliche Konfiguration

/T0210/ Anzeige der persönlichen Konfiguration: Beate lässt sich ihre eigene Konfiguration anzeigen.

/T0220/ Ändern der persönlichen Konfiguration: Beate ändert die Art ihrer Internetverbindung.

/T0230/ Speichern der persönlichen Konfiguration: Beate speichert ihre eigene Konfiguration.

/T0240/ Löschen der persönlichen Konfiguration: Beate löscht ihre Konfiguration und damit sind die Standardwerte wiederhergestellt.

11.9.2. Hilfefunktion

/T0310/ Anzeige der Hilfe nach der Erstbenutzung: Tim benutzt zum ersten Mal das System und die Hilfe-Seite öffnet sich nach der Anmeldung.

/T0320/ Anzeige der Hilfe im System: Tim sucht Tipps zur Benutzung des Systems.

11.9.3. Standortfunktionen

Persönliche GPS-Funktionen

/T0410/ aktuellen Standpunkt anzeigen: Tim fordert die Anzeige seines aktuellen Standorts auf der Karte.

/T0420/ aktuellen Standpunkt übermitteln: Tim will seinen Standort an Beate übermitteln.

/T0430/ aktuellen Standpunkt als Point of Interest hinzufügen: Tim will seinen aktuellen Standort als einen Point of Interest speichern.

Favoriten-Funktionen

/T0510/ Favoriten hinzufügen: Beate fügt den bestehenden Point of Interest "*Arbeit*" zu ihrer Favoriten-Liste hinzu.

/T0520/ Favoriten bearbeiten: Beate ändert die Telefonnummer eines Favoriten.

/T0530/ Favoriten löschen: Beate entfernt den Favoriten "*Arbeit*" von der Favoriten-Liste.

/T0540/ Anzeige von Favoriten auf der Karte: Beate lässt sich einen Favoriten auf der Karte anzeigen.

Point of Interest-Funktionen

/T0610/ *Point of Interest erstellen*: Tim steht vor dem DOM und erstellt diesen Standort als einen neuen Point of Interest und macht noch ein Foto zu dem Point of Interest. Er gibt auch die Telefonnummer des DOMs ein.

/T0620/ *einen bestehenden Point of Interest bearbeiten*: Tim ändert das Foto von dem Point of Interest "HOME".

/T0630/ *Einen Point of Interest löschen*: Tim löscht den bestehenden Point of Interest "Arbeit".

/T0640/ *Points of Interest auf der Karte anzeigen*: Tim lässt sich die Points of Interest *Friends* auf der Karte anzeigen.

Friends-Funktionen

/T0710/ *einen anderen Benutzer hinzufügen*: Beate fügt den bestehenden Point of Interest "Tim" mittels seines Benutzernamens zu ihrer *Friends*-Liste hinzu.

/T0720/ *anderen Benutzer entfernen*: Beate entfernt den Point of Interest "Tim" von ihrer *Friends*-Liste.

11.9.4. Route-Funktionen

Einfache Route

/T0810/ *Route berechnen*: Beate lässt sich die Route von ihrem aktuellen Standort, zum DOM berechnen. Die Route wird auf der Karte angezeigt.

/T0820/ *öffentliche Verkehrsmittel*: Beate benutzt die Bahn für die Routenberechnung von der Arbeit zu ihrem Freund.

/T0830/ *Dauer*: Tim kriegt die Dauer der Route angezeigt.

/T0840/ *Entfernung*: Tim kriegt die Länge der Route angezeigt.

/T1850 *Richtungsfunktion*: Tim sieht den Pfeil auf der Karte, der zum Ziel zeigt, und orientiert sich dementsprechend.

Tour-Funktionen

/T0910/ *Tour erstellen*: Beate erstellt sich von verschiedenen Points of Interest eine Tour.

/T0920/ *Tour bearbeiten*: Beate bearbeitet die Sightseeing-Tour und entfernt den Point of Interest *DOM* und fügt dafür die Adresse *Aachener Straße 77* hinzu, wobei sie die Adresse als den ersten Punkt in der Tour ordnet.

/T0930/ *Tour löschen*: Beate löscht eine bestehende Tour.

/T0940/ *Tour anzeigen*: Beate lässt sich eine bestehende Tour anzeigen.

11.9.5. Such-Funktionen

/T1010 *Adresse suchen*: Tim tippt die gesuchte Adresse *Luxemburger Straße 8* ein.

/T1020 *Point of Interest suchen*: Tim sucht nach einem *Subway* in der Nähe von seiner *Arbeit* und bekommt ihn auf der Karte angezeigt.

11.9.6. Karten-Funktionen

/T1110 *Scrollen auf der Karte*: Tim scrollt mit Hilfe der Navigationstasten auf der *Karte* hin und her.

/T1120 *Kurzinfo-Funktion*: Tim scrollt mit Hilfe der Navigationstasten auf den angezeigten Point of Interest *DOM* und bekommt ein kleines Infofenster zu sehen.

/T1130 *Informationsfunktion*: Beate klickt auf den angezeigten Point of Interest *DOM* auf der Karte und landet damit auf der dazugehörigen Informationsseite.

/T1140 *Anruf-Funktion*: Beate sieht auf der Informationsseite vom DOM die Telefonnummer und ruft diese direkt an.

/T150 *Kartenmodus-Funktion*: Beate wechselt den Modus auf *Satelliten*.

11.9.7. System-Einstellungen

/T1210 *Art der Fortbewegung*: Beate stellt die Art der Fortbewegung auf *Fußgänger* ein.

/T1220 *Angezeigte Point of Interest-Kategorien*: Beate wählt die Kategorien *Shops* und *Cafés* als ihre Point of Interest-Kategorien aus.

/T1230 *Anzeige des Datenvolumens*: Beate schaltet die Anzeige der Menge der übertragenen Daten aus.

11.9.8. Administrator-Funktionen

Systemverwaltung

/T1310 *öffentliche Point of Interest*: Der Administrator löscht den Point of Interest *Subway in Aggerstraße*, weil der nicht mehr existiert.

Benutzerverwaltung

/T1410 *Verbannung des Benutzers*: Der Administrator wirft den Benutzer *Jürgen* raus und blockiert seine eMail-Adresse, weil er gegen die Regeln verstossen hat.

11.10. Entwicklungsumgebung

Welche Software, Hardware und Orgware wird zur Entwicklung benötigt?

Es wird darauf geachtet, dass alle Entwicklungstools kostenlos (*Freeware*) sind.

11.10.1. Software

- Plattform
 - Xampp (*MySQL*)
 - Apache
 - J2ME
- Tools
 - Eclipse
 - WTK
 - Nokia Prototype SDK 4.0 for Java(TM) ME
 - PHPMyAdmin
 - ERwin
 - TortoiseSVN
 - MIDlet-Framework(*LMR*)

11.10.2. Hardware

- Java-fähiges mobiles Endgerät
- GPS-Mouse

11.11. Ergänzungen

Spezielle, noch nicht abgedeckte Anforderungen.

11.11.1. Barrierefreiheit

Um ein möglichst barrierefreies System zu entwickeln, soll speziell zusätzlich zur reinen Routenfunktion eine

- Sprachausgabe
- textbasierte Ausgabe

entwickelt werden.

11.11.2. Automatisierung Routenfunktion

Routen mit mehreren Points of Interest können automatisch so angeordnet werden, dass zwischen den Punkten eine möglichst kurze Strecke vorliegt.

12. Entwurf

12.1. Zielgruppe

Die Software soll sich an Menschen richten, die viel unterwegs sind und dadurch einen hohen Bedarf an Navigationsmöglichkeiten haben. Durch die Konzeption einer Schnittstelle zu öffentlichen Verkehrsmittelanbietern erweitert sich die Zielgruppe natürlich auch auf Menschen, die sich mittels Bus und Bahn in ihrer eigenen Stadt fortbewegen.

12.2. Szenarien

Ein zentrales Instrument zur systematischen Auseinandersetzung mit der Zukunft sind Szenarien. Sie beschreiben mögliche Situationen und ermöglichen die Handhabung komplexer Entscheidungssituationen. Szenarien werden heute vor allem in der strategischen Planung und Früherkennung eingesetzt.

Ein Szenario ist somit eine allgemeinverständliche Beschreibung einer möglichen Situation in der Zukunft, die auf einem komplexen Netz von Einflussfaktoren beruht. Ein Szenario kann darüber hinaus die Darstellung einer Entwicklung enthalten, die aus der Gegenwart zu dieser Situation führt. Die Gesamtheit alternativer Szenarien beschreibt den Zukunftsraum.

Folgende Szenarien spiegeln die Anwendungsmöglichkeiten des Systems wieder:

Thomas J., Geschäftsreisender, 32 Jahre

Herr J. ist auf Geschäftsreise in Ulm. Nach Ankunft auf dem Bahnhof möchte er in das Maritim Hotel. Um sich zu erkundigen, wo das Hotel liegt, bedient er sich der "SOFTWARENAME" und öffnet das Programm. Er sieht sich zunächst als roten Punkt auf einer Karte, nahe dem Bahnhof. Er öffnet das Kontextmenü und geht zum Punkt "Adresse finden". Er gibt die Adresse des Hotels ein und stellt fest, dass die automatisch errechnete Route vom aktuellen Standort 3 km entfernt ist und entschließt sich, lieber ein Taxi zu benutzen. Das Kontextmenü enthält einen weiteren Punkt "Unternehmen suchen". Diesen aktiviert er und es öffnet sich

ein Untermenü mit verschiedenen Kategorien, wie Taxen, Sehenswürdigkeiten, Restaurants, usw. Er wählt Taxen aus und bekommt auf der Karte sämtliche Taxistände in der näheren Umgebung angezeigt. Er wählt ... aus und ruft ihn auf. Es wird ein kleines Fenster angezeigt mit Kurzinformationen und der Telefonnummer. Es gibt einen weiteren Punkt, Taxen zum Standort rufen. Diesen wählt Herr J. aus und bekommt eine Meldung zur Bestätigung der Aktion. Es werden die GPS-Informationen des Standorts von Herrn Thomas J. auf den Client des Taxiunternehmens übermittelt. Die Informationen werden dann in diesem Fall auf der Karte jenes Clients angezeigt. Ein Taxifahrer wird zum Bahnhof geschickt.

Marc H., in Ausbildung, 22 Jahre

Marc besucht seinen Kumpel Stefan, in seiner neuen Wohnung. Stefan muss kurz aus dem Haus um sich seinen neuen Schrank zu holen, den er bei seinem Umzug bestellt hatte und bittet Marc, sich ums Essen zu kümmern bis er zurückkommt. Marc will sich zwei Pizzen liefern lassen, dafür braucht er aber die Telefonnummer von den Pizzerien in der Nähe. Er entscheidet sich, die Nummern im Programm zu suchen, das er auf dem Handy hat. Er öffnet das Programm "SOFTWARENAME" auf dem Handy und meldet sich an. Er geht unter dem "Menu" auf "Search" und gibt als Begriff Pizzeria an und klickt bei "near by" auf "GPS". Dann bekommt er die Pizzerien (die auch Nutzer dieser Programm-Community sind) auf der Karte angezeigt. Er scrollt mit den Tasten auf die, die am nächsten liegt und betätigt die "Anruf-Funktion", und ruft direkt bei der Pizzeria an. Nach der Bestellung wird Marc aufgefordert die Adresse zu nennen. Er kennt aber die genaue Adresse von Stefans Wohnung nicht. Aber zum Glück hat er ja dieses Programm und kann seine Koordinaten einem anderen Benutzer übermitteln. Er legt auf, um dann nochmal auf das ICON für diese Pizzeria zu klicken und betätigt die Funktion "send my GPS". Damit bekommt der andere Benutzer, in diesem Fall der Ladenbesitzer, die Koordinaten von Marc und damit auch seine Adresse auf dem PC dargestellt.

12.3. Teilprodukte identifizieren

12.3.1. Darstellung

Ziele

- Allgemeine Anzeige eines Kartenausschnittes
- Darstellung von eigenen Koordinaten als Punkt in der Karte
- Darstellung der Standorte anderer Community-Mitglieder
- Anzeige der Route
- Anzeige der Routenrichtung (Pfeil)
- Optionale Anzeige des Datentransfers
- Anzeige von Routendauer und -entfernung

Alternativen

- Die erste Alternative ist das *Package von Landspurg* ¹ in Kombination mit dem MIDlet Framework. Bei dieser Alternative wird der größte Teil der Ziele abgedeckt.
- Die zweite Alternative ist *SVG*, der ein Standard zur Beschreibung zweidimensionaler Vektorgrafiken ist.
Mit dieser Technik kann man *OpenstreetMaps (OSM)*-Material darstellen. OSM ist ein Projekt, das vektorbasierte Kartendaten und eine entsprechende Plattform für die Visualisierung dieser Daten zur Verfügung stellt.
- Die dritte Alternative ist die Nutzung des TCG-Systems, das von Herrn Tim Hausmann entwickelt wurde.

12.3.2. Navigation

Ziele

- Routenberechnung (dynamische Neuberechnung)
- Auswahl, ob zu Fuß, mit Auto oder mit Fahrrad
- grafische, textbasierte und sprachliche Ausgabe

¹vgl. <http://j2memap.landspurg.net/>

Alternativen

- Die erste Alternative ist *Package von Landspurg*, die zur Routenberechnung dient.
- Die zweite Alternative ist die Implementierung anderer Navigationsysteme, um die zusätzliche Funktionen wie die Nutzung der öffentlichen Verkehrsmittel, als textbasierte und sprachliche Ausgabe zu ermöglichen.

12.3.3. Routenverbindung mittels öffentlicher Verkehrsmittel

Ziele

- Berechnung von Routen in Verbindung mit öffentlichen Verkehrsmittel für Fußgänger und Radfahrer
- Anzeige der naheliegenden Haltestellen

Alternativen

- *DB* (Deutsche Bahn) ist die erste Alternative, die aber um einen Algorithmus für den Fußweg erweitert werden soll.
- *KVB* (Kölner Verkehrs-Betriebe) ist die zweite Alternative, die allerdings auch um den Algorithmus für den Fußweg zu erweitern ist.
- Die dritte Alternative ist die Nutzung der DB oder KVB ohne Fußwegberechnung.

12.3.4. Übermittlung der GPS-Koordinaten

Ziele

- eigene GPS-Koordinaten übermitteln an ein anderes Community-Mitglied (*send*)
- empfangen von GPS-Koordinaten anderer Community-Mitglieder (*receive*)
- Sichtbarkeit der eigenen GPS-Informationen für andere Community-Mitglieder (*boolean*)

Alternativen

- Die Benutzung der Community, die von der Firma *LMR* erstellt wurde.
- Die Entwicklung eines eigenen Community-basierten Systems.

12.3.5. Favoriten managen

Ziele

- Hinzufügen von:
 - Points of Interest in die Favoriten-Liste
 - Adressen, die von dem Benutzer auf der Adress-Seite eingegeben werden
 - GPS-Koordinaten, die den aktuellen Standort des Benutzers entsprechen
- Entfernen der Favoriten aus der Favoriten-Liste
- Bearbeiten einzelner Favoriten
- Kategorisierungen der Favoriten

Alternativen

- Die erste Alternative ist die Möglichkeit, die Favoriten lokal auf dem Handy zu speichern und zur Aktualisierung die Daten aus dem Internet herunterzuladen.
- Die zweite Alternative ist, die Favoriten nur lokal auf dem Handy zu speichern.
- Die dritte ist, online auf die Favoriten zugreifen.

12.3.6. Points of Interest

Ziele

- Filterung verschiedener Points of Interest und Favoriten
- die Möglichkeit einen anderen ausgewählten Benutzer durch die Anruffunktion zu erreichen
- zu jedem Point of Interest besteht die Möglichkeit ein Bild hinzuzufügen
- der Benutzer kann zu jedem einzelnen Point of Interest seine persönliche Bewertung abgeben
- jeder Benutzer kann Kommentare zu den Points of Interest hinzufügen und jeder Zeit abrufen
- zu jedem einzelnen Point of Interest existiert ein Informationsfeld (Öffnungszeiten, etc.)

Alternativen

- Die erste Alternative ist eine eigene Datenbank.
- Die zweite Alternative ist die Nutzung des Systems von Herr Tim Hausmann.
- Die dritte Alternative ist eine Kombination von Flickr und Tagzania.

12.3.7. Virtuelle Tour

Ziele

- Zusammenstellung von mehreren Points of Interest zu einer Gesamt-Tour
- Sortierung der Reihenfolge der Points of Interest in einer Tour manuell oder automatisch

Alternativen

- Die virtuelle Tour zu programmieren.

12.4. Paper-Based-Prototype

Ein Prototyp ist ein für jeweilige Zwecke funktionsfähiges Versuchsmodell eines geplanten Produktes. Es kann dabei nur rein äußerlich oder auch technisch dem Endprodukt entsprechen.

Paper-Based-Prototyping ist eine schnelle und effiziente Methode, um User-Interfaces ohne Programmieraufwand zu modellieren und mit echten Nutzern zu testen. Sie wird verwendet, um ein direktes Feedback von Usern über die Software betreffend des Layouts und der bereitgestellten bzw. notwendigen Funktionalitäten zu erhalten. Das Ziel des Paper-Based-Prototyping ist es, Zeit und Geld zu sparen, da Probleme (Layout/-Funktionalität) schon gelöst werden können, bevor die Implementationsphase begonnen hat. Das direkte Feedback des Users kann dazu verwendet werden, um im Entwicklungsprozess bereits am Anfang die richtigen Schwerpunkte setzen zu können.

12.4.1. Login

Softkeys:²

- LOGIN
- QUIT

Nach dem Start des Programms bekommt der Benutzer die Möglichkeit sich anzumelden, bzw. für den Fall, dass er zum ersten mal mit dem Programm arbeitet, sich zu registrieren.

Zum Einloggen braucht der Benutzer einen individuell wählbaren Benutzernamen und ein Passwort.

Zur Registrierung bekommt der Benutzer eine automatische eMail an die eingegebene eMail-Adresse. Dies ist auch genauso von einem PC mit Internetverbindung möglich.

Für die Verbindung zum Internet ist eine Standardeinstellung vorhanden. Falls der Benutzer sich eine andere Konfiguration wünscht, kann er sich durch Benutzereinstellung eine neue anlegen. Dabei sind auch weitere Einstellungen für Internetverbindung möglich.

Zur optischen Trennung von zwei funktionell verschiedenen Bereichen(*Login und Internetverbindung*), sind zwei differenziert wahrnehmbare Darstellungen vorhanden. Dadurch wird dem Benutzer ermöglicht, diese leichter auseinander zu halten.

² Als Softkeys bezeichnet man bei Mobiltelefonen die beiden Tasten rechts und links unter dem Display



Abbildung 12.1.: Login

12.4.2. Help

Softkeys:

- OK
- QUIT

Die Help-Seite wird angezeigt, nachdem sich der Benutzer registriert hat. Bei jedem weiteren Login wird die Darstellung dieser Seite unterbunden, ist jedoch jeder Zeit unter dem Hauptmenü abrufbar.

Auf dieser Seite werden Informationen angezeigt, die dem Benutzer helfen, in dem Programm zu navigieren. Hierbei werden allgemeine Programminformationen und die verschiedenen Funktionlitäten der Telefontasten in Textform beschrieben.



Abbildung 12.2.: Help

12.4.3. Hauptseite

Softkeys:

- MENU
- GET ANYWHERE

Auf der Hauptseite werden zwei verschiedene Softkeys mit verschiedenen Funktionalitäten, jeweils links *MENU* und rechts *get Anywhere*, positioniert.

Der Softkey, der links angelegt wird, ist standardgemäß für die Hauptnavigation in dem Programm gedacht. Um die Perzeption des Benutzers zu unterstützen, wird wie in vielen Systemen das Navigationsmenü links unten orientiert.

Der Softkey *get Anywhere* beinhaltet eine Extra-Funktion, die nicht im Menü vorhanden ist. Darunter verbirgt sich die One-Click-Strategie, die dem Benutzer ermöglicht mit wenigen Klicks eine Route von der aktuellen Position zu einem vorher festgelegten Point of Interest berechnen zu lassen. Diese Funktion wird im nächsten Absatz ausführlicher beschrieben. Im Folgenden werden die Punkte des *Menüinhalts* näher erläutert.



Abbildung 12.3.: Hauptmenü

12.4.4. My Location

My Location beinhaltet GPS-bezogene Funktionalitäten, die dazu dienen, den aktuellen Standort des Benutzers auf der Karte zu repräsentieren.



Abbildung 12.4.: My Location

- My Location-Untermenü

- *show my location*
"show my Location" zentriert den aktuellen Standpunkt des Benutzers auf der Karte.
- *send my location*
"send my Location" ist eine Funktion, die das Übermitteln der GPS-Koordinaten an einen anderen Benutzer ermöglicht.
- *add my location*
Hier kann der Benutzer seinen GPS-Standort als einen Point of Interest in die Liste hinzufügen.
- *change my status*
"change my status" dient zur Auswahl der Sichtbarkeit für die Point of Interest-Kategorie Freunde.

12.4.5. Favoriten

Die Points of Interest, die für den Benutzer eine höhere Priorität besitzen, können unter Favoriten gespeichert werden, damit der User schneller und einfacher auf diese zugreifen kann. Diese ausgewählten Points of Interest werden in einer Liste "Favorites" hinzugefügt.

Softkeys:

- MORE
- BACK

Für die Verwaltung der Favoriten gibt es einen Manager, mit dessen Hilfe eine einheitliche Konfigurationsstruktur eingehalten werden kann. Die angelegten Favoriten werden aufgelistet.

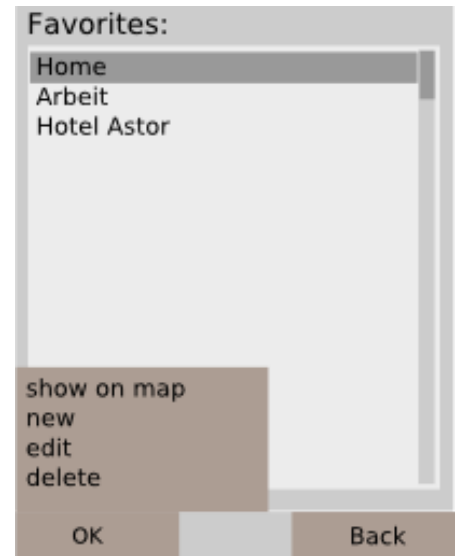


Abbildung 12.5.: Favoriten

- Favoriten-Untermenü

- *show on map*
"show on map" ermöglicht die Anzeige der ausgewählten Favoriten auf der Karte.
- *new*
Unter "new" kann der Benutzer einen Punkt aus der Point of Interest-Liste in die Favoriten-Liste übertragen.
- *edit*
Unter "edit" hat der Benutzer die Möglichkeit, einzelne Points of Interest, die als Favoriten gespeichert sind, zu bearbeiten. Dies ist jedoch auch unter "edit" bei Point of Interest möglich.
- *delete*
Unter "delete" lassen sich einzelne Favoriten aus der Favoriten-Liste löschen.

12.4.6. Points of Interest

Hier werden sämtliche Points of Interest in mehreren Kategorien für verschiedene Verwendungen aufgelistet. Unter jeder Kategorie sind die dazugehörigen Points of Interest gespeichert.

Softkeys:

- MORE
- BACK

Bei den Points of Interest existiert derselbe Manager wie bei Favoriten, der die Verwaltung der Points of Interest übernimmt.

Alle Kategorien sind hier aufgelistet, jedoch ist eine Auswahlmöglichkeit der Kategorien und/oder einzelner Points of Interest zur Weiterprozessierung vorhanden.

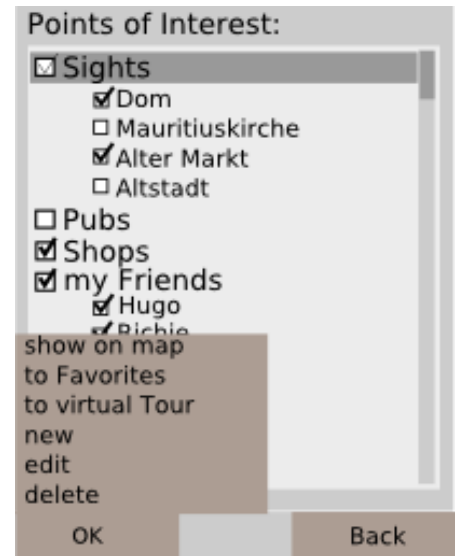


Abbildung 12.6.: Point of Interest

-Point of Interest-Untermenü

- *show on map*
"show on map" ermöglicht die Anzeige der ausgewählten Points of Interest auf der Karte.
- *new*
Unter "new" kann der Benutzer einen neuen Point of Interest in einer bestimmten Kategorie hinzufügen. Das kann sowohl der aktuellen Standpunkt des Benutzers als auch eine Adresse sein. Der Benutzer hat die Möglichkeit zu jedem einzelnen Point of Interest ein Foto und einige Informationen hinzuzufügen.
- *edit*
Unter "edit" hat der Benutzer die Möglichkeit einzelne Points of Interest zu bearbeiten.
- *delete*
Unter "delete" kann der Benutzer einzelne Points of Interest löschen.

12.4.7. Route

Unter "Route" im Hauptmenü existiert ein Untermenü mit den Auswahlmöglichkeiten: "mit Auto", "zu Fuß" oder "mit Fahrrad". Die Reihenfolge der Punkte bestimmt sich nach der Häufigkeit der Benutzung im Untermenü. Dabei unterscheiden sich die Alternativen vor allem hinsichtlich der Nutzung öffentlicher Verkehrsmittel. Des Weiteren werden bei der Option "mit Fahrrad" Radwege in die Routenplanung einbezogen.

Softkeys:

- OK
- BACK

Abbildung 12.7.: Route

Nach der Auswahl der Fortbewegungsart erfolgt die Berechnung der Route zwischen zwei Punkten.

Bei der Darstellung dieser Seite entschieden wir uns aus Gründen des Platzmangels und der Übersichtlichkeit für ein Drop-Down-Menü, da jeweils bei Start und Ziel mehrere Optionen zur Auswahl stehen und das Auflisten von allen Möglichkeiten ("Favoriten", "Points of Interest" und "Adressen") viel Platz benötigt. Beim Start ist außerdem die Auswahl vom aktuellen GPS-Standpunkt des Benutzers möglich.

Falls sich der Benutzer für einen Favoriten oder einen Point of Interest entscheidet, wird ihm eine Seite dargestellt, in der die Favoriten bzw. die Points of Interest aufgelistet werden. Falls der Benutzer aber eine Adresse als Start oder Ziel eingeben möchte, wird eine andere Seite aufgerufen, die dem Benutzer das Eintippen der Adresse ermöglicht. Unter jedem Drop-Down-Menü befindet sich ein Textfeld, in dem die Adresse des gewählten Start- bzw. Zielpunktes dargestellt wird. Dies dient zur Verifikation.

Im weiteren Verlauf der Seite ist die Auswahlmöglichkeit der Benutzung öffentlicher Verkehrsmittel für Fußgänger und Fahrradfahrer vorhanden.

12.4.8. Virtual Tour

Eine "Virtual Tour" ist eine Verkettung von mehreren Routen zwischen verschiedenen Punkten, die zu einer Gesamt-Tour zusammengestellt werden können.

Softkeys:

- MORE
- BACK

Hier kommt wieder der Manager zur Verwaltung der Touren zum Einsatz.



Abbildung 12.8.: Virtual Tour

- Virtual Tour-Untermenü

- *show on map*
"show on map" dient zur Anzeige der ausgewählten Tour auf der Karte.
- *new*
"new" ermöglicht die Zusammenstellung einer neuen Tour.
- *edit*
Unter "edit" hat der Benutzer die Möglichkeit einzelne Touren zu bearbeiten.
- *delete*
Unter "delete" kann der Benutzer einzelne Touren löschen.

12.4.9. Search

Die Option "Search" ist eine allgemeine Suche nach Points of Interest.

Softkeys:

- OK
- BACK

Es existiert ein Eingabefeld für Suchkriterien. Dazu hat der Benutzer die Möglichkeit die Suchumgebung zu bestimmen. Für die Suchumgebung gibt es fünf verschiedene Optionen:

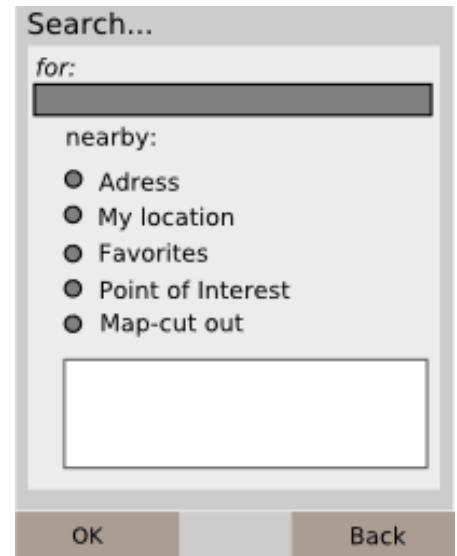


Abbildung 12.9.: Search

- *Adresse*
Unter "Adresse" kann der Benutzer eine Adresse bestimmen, die zur Ergänzung des Radius der Umgebung des zu suchenden Point of Interest dient. Bei der Auswahl der Option "Adresse" wird ein neues Fenster mit Textfeldern zur Eingabe der Adresse aufgerufen.
- *My Location*
Bei der Option "My Location" wird der gesuchte Point of Interest in der Nähe der aktuellen GPS-Position gesucht.
- *Favorites*
Bei der Option "Favorites" kann der Benutzer den Standort eines der Favoriten als Suchumgebung bestimmen.
- *Point of Interest*
Bei der Option "Point of Interest" kann der Benutzer den Standort eines der Points of Interest als Suchumgebung bestimmen.
- *Map-cut out* (Kartenausschnitt)
Bei der Option "Map-cut out" kann der Benutzer einen beliebigen aktuellen Kartenausschnitt als Suchumgebung bestimmen.

Nach der Auswahl einer der Optionen wird die Adresse des ausgesuchten Punktes als Suchumgebung in einem Textfeld angezeigt.

12.4.10. Settings

Im System können mehrere Standardeinstellungen gespeichert werden.

- Angezeigte Point of Interest-Kategorien:
Der Benutzer kann die dem System zur Verfügung gestellten Kategorien für Points of Interest festlegen.
- Anzeige des Datenvolumens:
Der Benutzer kann sich die Menge der übertragenen Daten anzeigen lassen.

Der Benutzer verfügt über persönliche Daten, die er frei gestalten kann.

- Anzeige der eigenen, persönlichen Daten:
Der Benutzer kann sich seine persönlichen Daten vom System **vollständig anzeigen** lassen.
- Ändern der eigenen, persönlichen Daten:
Der Benutzer kann seine persönlichen Daten aktualisieren bzw. **ändern**.
- Sichtbarkeit der eigenen, persönlichen Daten:
Der Benutzer kann jeden einzelnen Eintrag seiner persönlichen Daten auf **sichtbar** bzw. **unsichtbar** setzen.
- Anzeige der persönlichen Daten anderer Benutzer:
Der Benutzer kann sich von anderen Benutzern die persönlichen Daten anzeigen lassen, dabei können auf unsichtbar gesetzte Einträge nicht gesehen werden.

12.4.11. Quit

Die Funktion "Quit" dient zum Beenden des Programms. Dabei werden die zuletzt auf dem Display angezeigten Points of Interest gespeichert.

12.4.12. get Anywhere

"get Anywhere" ist eine spezielle One-Click-Strategy für die Suchfunktion, die dem Benutzer hilft, möglichst schnell und einfach zum Ziel zu gelangen. Bei dieser besonderen Suche wird immer der aktuelle GPS-Standpunkt des Benutzers als Startposition festgelegt.

Softkeys:

- OK
- BACK

-get Anywhere-Menü

- 3 Top-Favoriten
Hier werden im Menü als erstes die drei meistbenutzten Favoriten aufgelistet, damit der Benutzer schnell auf einen der Favoriten zugreifen kann.
- Other Favorites
Unter "Other Favorites" kann sich der Benutzer weitere Favoriten aus der Favoriten-Liste auswählen.
- Type address
Unter "Type address" hat der Benutzer die Möglichkeit eine Adresse als Ziel der Route einzutippen. Bei dieser Option wird wieder die Adress-Seite mit Adressfeldern aufgerufen.
- Settings
Hierbei kann der Benutzer seine persönliche Einstellungen, wie z.B. die Nutzung der öffentlichen Verkehrsmittel oder die Art seiner Fortbewegung, als Standard speichern.



Abbildung 12.10.: Get Anywhere

12.5. Probandenbefragung

In unserer Arbeit haben wir einen großen Wert darauf gelegt, den User als zentralen Gegenstand der Entwicklung mit einzubeziehen. Dies garantiert eine höhere Gebrauchstauglichkeit, um unter einem bestimmten Benutzungskontext Ziele effektiv, effizient und zufriedenstellend zu erreichen.

Um den entwickelten Prototyp zu evaluieren, entwarfen wir einen Fragebogen, der sich im Anhang befindet.

In die Auswahl der befragten Benutzer haben wir vorrangig Menschen aufgenommen, die:

- Touristen,
- zwischen 14 und 70 und
- Handybesitzer

waren.

12.6. Auswertung der Probandenbefragung

Wir haben uns für zehn Probanden entsprechend der Zielgruppendefinition entschieden. Die Probanden kamen zum größten Teil mit dem Prototypen, den wir ihnen vorgestellt hatten, zurecht und konnten den Sinn der Produkt-Funktionen nachvollziehen.

12.6.1. Ergebnis:

get Anywhere

Das Problem bei "get Anywhere" war der Begriff, der für die meisten Probanden unverständlich war, wobei nur einer von zehn Befragten auf Anhieb den Sinn verstand. Aber nach der Erklärung der Funktion war der Sinn nachvollziehbar. Wir stellten fest, daß wir einen *neuen Begriff* für die Funktion finden müssen.

Um dieses Problem zu lösen, erörterten wir neue Begriffe, die wir erneut mit Benutzern testeten und entschieden uns für ein neues dynamisches Label. Der Benutzer legt einen Point of Interest als Standard für diese Funktion fest, dessen Name als Label erscheint. Zum Beispiel wird der Point of Interest "HOME" als "get HOME" für die Bezeichnung der Funktion verwendet.

My Location

Bei dieser Option war die Unverständlichkeit des Begriffes erneut problematisch. Viele Probanden assoziierten dabei die Funktionalität der Favoriten.

Favorites

Der Sinn der Funktion war einfach zu verstehen aber die Positionierung war leicht kompliziert. Die Befragten haben bei den favoritenbezogenen Fragen die Lösung direkt bei "My Location" gesucht.

Bei der Option "Favorites" haben die Probanden eine Möglichkeit zum Setzen vom Ziel- oder Startpunkt der Routenberechnung gewünscht.

Da uns nach der Befragung eine Zusammenführung der Funktionen "My Location" und "Favorites" als sinnvoller erschien, haben wir uns entschieden *Favoriten* als ein Teil von *My Location* zu setzen und das als "My Locations" zu bezeichnen. Dies steht für die persönliche Favoriten und auch für eigene GPS-Koordinaten.

Points of Interest

Das Label der Funktion war den Befragten nicht eindeutig genug, daher haben wir uns für den Begriff "*HotSpots*" entschieden, der nach erneuter Befragung verständlicher ist. Bevor wir eine Erklärung über die Fülle des Informationsgehaltes von Points of Interest abgegeben haben, konnten die Teilnehmer nicht richtig einschätzen, was sich hinter einem Point of Interest verbirgt.

Den Benutzern waren die Funktionen "new" und "edit" im Point of Interest-Menü nicht auseinander zu halten. Probanden waren teils überfordert, was sie benutzen sollen. Nach der erneuten Rücksprache mit den Befragten haben wir uns für eine ausführliche Beschriftung von Funktionsnamen entschieden. Der Funktionsname "new" wurde demzufolge umgeändert in "new HotSpot" und "edit" in "edit HotSpot".

Bei Point of Interest-New sollte das Hinzufügen eines Bildes "add Picture" mit ins Hauptfenster. Dies ist benutzerfreundlicher als ein neues Fenster aufgehen zu lassen.

Die Probanden haben eine Möglichkeit zum Setzen vom Ziel- oder Startpunkt der Routenberechnung gewünscht.

Route

Die Funktionalität der Route war leicht verständlich.

Die gewünschten Optionen von den Teilnehmern waren:

- Die Entfernung und die Dauer der Route sollte anzeigbar sein, und auch nach Wunsch ausblendbar.
- Die Befragten haben sich die Möglichkeit der Berechnung der Route zu einem beliebigen Punkt auf der Karte gewünscht.
- Die Auswahl von kürzester oder schnellster Strecke sollte möglich sein.

Nach der Vorführung des Prototypen zeigten lediglich wenige Probanden Interesse an einer gezeichneten Route. Es gab den Vorschlag eines Probanden die Richtung zum Ziel zu kennzeichnen. Wir eruierten eine Idee, die diesen Vorschlag aufgreift. Ein Richtungspfeil, der zum Ziel zeigt und sich der Bewegung des Benutzers dynamisch anpasst. Diese Idee fanden die Probanden wesentlich sinnvoller.

Virtual Tour

Die Trennung von Virtual Tour und Route war für die Probanden nicht ganz klar.

Nach der Befragung erwies sich die Idee als sinnvoller, beide Funktionen zusammenzufassen.

Da Die Option "Virtual Tour" als ein Extra-Menü-Punkt nicht mehr existiert, werden die Namen der Funktionen "new" und "edit" in "new Route" und "edit Route" umgeändert. Der Funktionssinn erab sich nach erneuter Befragung verständlicher.

Search

Die Funktion zeigte sich als verständlich und gut nutzbar, jedoch fehlte die Suche nach einer Adresse. Die Suchfunktion wird um diese Option erweitert.

12.6.2. Fazit

Nach den Ergebnissen der Probandenbefragung werden folgende Punkte überarbeitet:

- Das Label "get Anywhere" in einem dynamischen Label nach der Auswahl eines HotSpots in "get *HotSpot-Name*" umbenennen.
- Die Funktion "Favorites" und "My Location" zu einer Option "My Locations" zusammenfassen, die die GPS-Koordinaten und die persönlichen Favoriten beinhaltet.
- Die Funktionen "*add Favorite as target*" und "*add Favorite as start*" als neue Optionen in Favoriten-Menü hinzufügen.
- Das Label "Point of Interest" in "HotSpots" umändern.
- Die Funktionen "*add HotSpots as target*" und "*add HotSpots as start*" als neue Optionen in HotSpots-Menü hinzufügen.
- Die Funktionen "new" und "edit" in "new HotSpot" und "edit HotSpot" umbenennen.
- Die Anzeige der Entfernung (*Luftlinie bei Fußgängern*) vom Start- zum Zielpunkt einer Route.
- Die Anzeige der Dauer einer Route für Autofahrer.
- Die Routenberechnung zu einem beliebigen Punkt auf der Karte.
- Die Auswahlmöglichkeit der kürzesten oder schnellsten Strecke bei der Routenberechnung.
- Anzeige der Zielrichtung durch einen Pfeil, der sich dynamisch mit der Bewegung des Benutzers ausrichtet.
- Die Funktion "Virtual Tour" wird mit der Funktion "Route" zusammengefasst.
- Die Funktionen "new" und "edit" bei "Virtual Tour" werden in "new Route" und "edit Route" umbenannt.
- Die Funktion "search" wird um die Option *search address* erweitert.

Im Prototyp implementierte Funktionen wurden unter diesen Überarbeitungen entwickelt.

13. Implementation

13.1. Systemarchitektur

13.1.1. Überblick

Bevor wir auf das implementierte Architekturmuster eingehen, möchten wir zunächst einen Überblick über den Ablauf der verwendeten Anwendungen geben.

Zum einen existieren die im Rahmen dieses Projektes entstandenen Anwendungen. Dabei stellt das MIDlet, die zentrale Benutzerinteraktion dar. Das MIDlet lokalisiert mittels des NMEA-Protokolls die aktuelle Position der GPS-Mouse. Anfragen des Benutzers werden teils mittels HTTP-POST- und teils mittels HTTP-GET-Methoden an das Zamspack-Servlet übermittelt. Dieses analysiert die Anfragen und leitet sie an andere Methoden weiter. Werden Anfragen zur Suche bestimmter HotSpots gestellt oder Daten übermittelt, die zum Erstellen eines neuen HotSpots dienen, bedient sich das Servlet der Datenbank. Wurde eine Anfrage erfolgreich bearbeitet, werden die Daten an das Servlet zurückgegeben, welches wiederum ein XML-Dokument generiert und an das MIDlet zurückschickt.

Des Weiteren existieren Anwendungen von Drittanbietern, die wir verwenden:

Um Kartenmaterial darzustellen, werden Anfragen an das im MIDlet implementierte J2MEMap-Interface übertragen, welches wiederum mit dem GoogleMaps-Server kommuniziert.

Leider ließ sich aus bereits oben genannten Gründen die Implementierung zum KVB-Datenbestand nicht verwirklichen. Aus diesem Grund wird auch nicht näher auf die Kommunikation zum Servlet der KVB eingegangen.

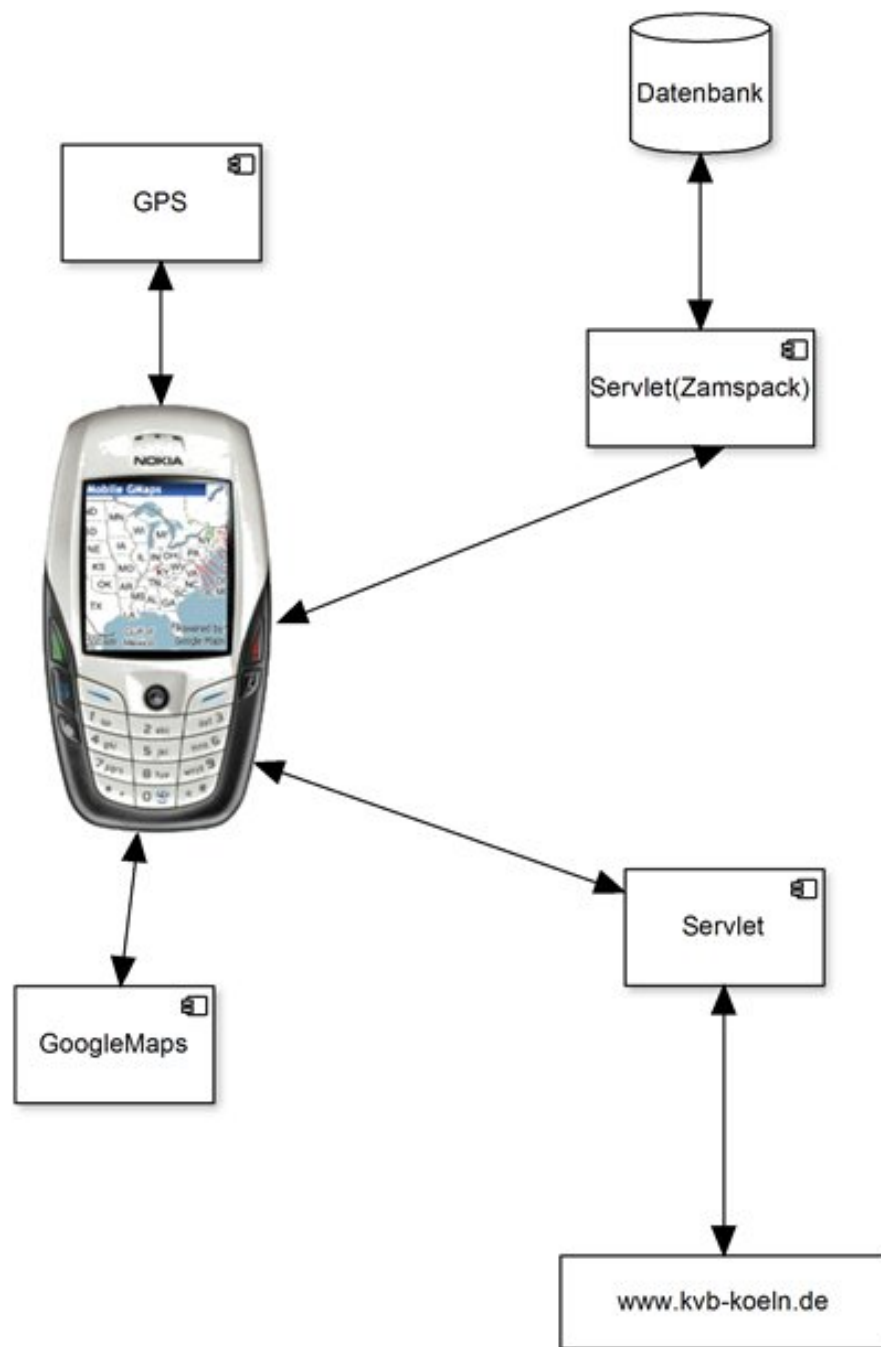


Abbildung 13.1.: Interaktion der einzelnen Komponenten

13.1.2. Dreischichtige Architektur

Unser System wurde als dreischichtige Architektur implementiert und fällt somit unter die Klasse der Webanwendungen.

Die dreischichtige Architektur (englisch *three tier architecture*) ist eine Client-Server-Architektur, die softwareseitig drei Schichten hat. Im Gegensatz zur zweischichtigen Architektur (englisch *two tier architecture*), bei der die Rechenkapazität weitestgehend auf die Client-Rechner ausgelagert wird, um den Server zu entlasten, existiert bei der dreischichtigen Architektur noch eine zusätzliche Schicht, die Logikschicht, die die Datenverarbeitung vornimmt.¹

Bei dieser Architektur werden die Schichten wie folgt aufgeteilt:

- **Präsentationsschicht** (*presentation layer*)

Diese Schicht ist für die Repräsentation der Daten, Benutzereingaben und die Benutzerschnittstelle verantwortlich. Die Clients dieser Schicht werden auch *Thin-Clients* genannt, da sie außer den oben genannten Funktionen keine weiteren Aufgaben besitzen.

In unserem System werden die Aufgaben dieser Schicht vom MIDlet übernommen.

- **Logikschicht** (*application logic layer*)

Sie stellt sozusagen die Programmintelligenz dar. In dieser Schicht befinden sich alle Verarbeitungsmechanismen, die zur Darstellung benötigt werden. Diese Schicht entspricht dem in unserem System verwendeten Servlet.

- **Datenschicht** (*resource management layer*)

Die Datenschicht repräsentiert den eigentlichen Datenbestand. Sie enthält die Datenbank und ist verantwortlich für das Speichern und Laden von Daten. In unserem System existieren zwei Datenschichten, zum einen die Datenbank auf der sich Informationen über HotSpots und Benutzer befinden und des Weiteren die Datenbank von GoogleMaps, auf denen sich das Kartenmaterial befindet.

¹Zitat von: http://de.wikipedia.org/wiki/Dreischichtige_Architektur

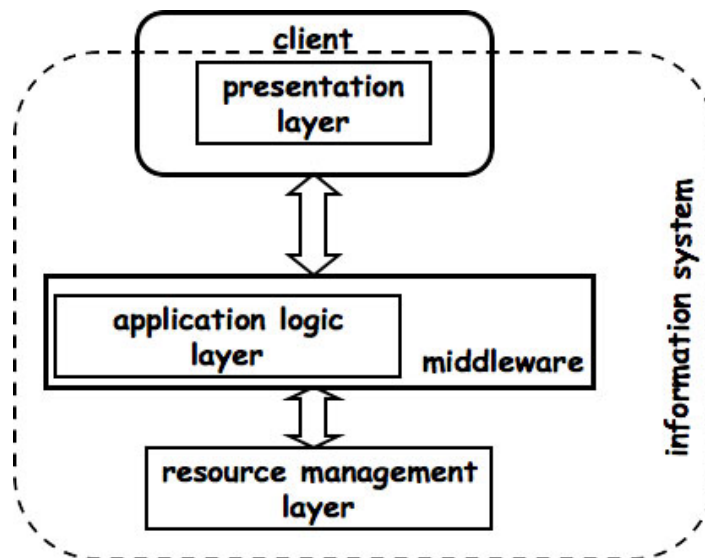


Abbildung 13.2.: Dreischichtige Architektur

13.2. ausgewählte Funktionen

Da wir im Rahmen dieser Arbeit nicht alle Funktionen implementieren können, haben wir eine Auswahl getroffen, die sich auf folgende Funktionen beschränkt:

/F0410/ *aktuellen Standort anzeigen:*

/F0430/ *aktuellen Standort als Point of Interest (HotSpot) hinzufügen:* anlegen.

/F0610/ *Point of Interest (HotSpot) erstellen:*

/F0640/ *Point of Interest (HotSpot) anzeigen:*

/F0850/ *Richtungsfunktion:*

/F1020/ *Point of Interest (HotSpot) suchen:*

/F1110/ *Scrollfunktion:*

/F1120/ *Kurzinfo-Funktion:*

/F1130/ *Informationsfunktion:*

/F1150/ *Kartenmodus-Funktion:*

/F1220/ *Angezeigte Point of Interest-Kategorien:*

/F1230/ *Anzeige des Datenvolumens:*

13.3. Client

13.3.1. MIDlet Framework

In dem MIDlet Framework wird die gesamte Oberfläche zur Menünavigation erstellt. In der in Java geschriebenen Datei *AppController* werden die einzelnen Objekte der Oberfläche erstellt. Dafür greift der *AppController* auf die in C geschriebenen *.inc*-Dateien zu. Das eigentliche Erzeugen von Darstellungsobjekten auf der Oberfläche wird mittels dieses Controllers ermöglicht.

Im folgenden Beispiel wird die grafische Implementierung der HotSpots-Oberfläche näher beschrieben.

Listing 13.1: Screenhotspot Schema

```

1 case SCREEN_HotSpots:
2     clearScreen();
3
4
5     resetSoftkeys();
6     m_nSoftkeyStates[APP_KEY_SOFT_L] = SCREEN_MORE3;
7     m_nSoftkeyStates[APP_KEY_SOFT_R] = SCREEN_MAIN;
8     setSoftkeyLabels(GENERAL_OPTIONS, GENERAL_BACK);
9
10    GUIObjList list1 = new GUIObjList(
11        LIST1_ENTRIES,
12        LIST1_X,
13        LIST1_Y,
14        SCREEN_WIDTH - (2*LIST1_X),
15        LIST1_DIST,
16        4,
17        0,
18        GUIObjMenue.SCROLL_STOP,
19        STRING_DELIMITER_DEFAULT,
20        GUIObjList.CHECKBOX,
21        GUIObject.OBJ_NOT_MOVEABLE);
22
23    list1.setScrollAutomatically(true);
24
25    addGUIObject(list1);
26
27    m_focusManager.addElement(list1);
28
29    m_focusManager.setFocus(list1);
30
31    break;

```

In den Zeilen sechs bis acht werden die Softkeys mit zwei Funktionen belegt, wobei beim Klicken der rechten Taste die Hauptseite und beim Klicken der linken Taste das Untermenü (*SCREEN_MORE3* (wird im Verlauf dieses Abschnittes beschrieben)) angezeigt wird. Ab Zeile zehn wird die Liste erzeugt, die die HotSpot-Kategorien beinhaltet, mit verschiedenen Parametern wie Größe, Farbe und Koordinaten, die aus verschiedenen *.inc*-Dateien entnommen werden und der Anzahl der Bestandteile.

Um einzelne HotSpots gemäß den Produktfunktionen (siehe */F06??/*) bearbeiten zu können wird im folgenden das *SCREEN_MORE3*-Untermenü näher erläutert.

Listing 13.2: Menüdarstellung Schema

```

1 case SCREEN_MORE3:
2     resetSoftkeys();
3     m_nSoftkeyStates[APP_KEY_SOFT_L] = GUI_TOGGLE_MAINMENU3;
4     m_nSoftkeyStates[APP_KEY_SOFT_R] = SCREEN_MAIN;
5     setSoftkeyLabels(GENERAL_OPTIONS, GENERAL_BACK);
6
7
8     m_allMenus[MENU_MAIN3] = new GUIObjMenue(
9         MAIN_MENU3_ENTRIES,
10        null,
11        MAIN_MENU_X,
12        MAIN_MENU_Y,
13        MAIN_MENU_DIST,
14        4,
15        0,
16        GUIObjMenue.O_VERTICAL,
17        Graphics.BOTTOM|Graphics.TOP,
18        0xFBD8C6,
19        GUIObjMenue.SCROLL_STOP,
20        STRING_DELIMITER_DEFAULT);
21
22     m_allMenus[MENU_MAIN3].setScreenTransitions(new int[]{
23         SCREEN_HotSpotNew, SCREEN_HotSpotEddit, SCREEN_HotSpotDelete,
24         SCREEN_Show});
25     m_allMenus[MENU_MAIN3].setVisible(false);
26     break;

```

In der dritten und vierten Zeile werden die Softkeys mit Funktionen belegt. Dem Softkey auf der linken Seite des Displays wird *GUI_TOGGLE_MAINMENU3* zugewiesen und dem auf der rechten Seite des Displays *SCREEN_MAIN*. Beim Klicken des linken Softkey bekommt der Benutzer das Menü und beim Klicken des rechten Softkey die Hauptseite angezeigt.

In der fünften Zeile werden die Namen der Labels festgelegt, die allerdings aus der *language_en_EN.inc*-Datei herausgesucht werden. In diesem Beispiel sind die Keys mit den Labels *ENERAL_OPTIONS* und *GENERAL_BACK* versehen, die jeweils in der

language_en_EN.inc-Datei für "MORE" und "BACK" stehen. In der achten Zeile des Codes wird das Untermenü mit dem namen *MENU_MAIN3* erzeugt, das ein neues Objekt vom Typ *GUIObjMenue* ist. Die Parameter, die hier übergeben werden, entstammen aus den verschiedenen *.inc*-Dateien:

- *MAIN_MENU3_ENTRIES*
Dieser Parameter vom Typ *String*, der in der neunten Zeile übergeben wird, wird aus der *language_en_EN.inc*-Datei herausgelesen. Diese enthält die Namen: "new HotSpot", "edit HotSpot", "delete HotSpot" und "show HotSpots".
- *MAIN_MENU_X*, *MAIN_MENU_Y* und *MAIN_MENU_DIST*
Die drei Parameter aus der elften, zwölften und 13. Zeile sind vom Typ *int* und werden jeweils aus der *gfx_S40_128x128.inc*-Datei entnommen. Diese Parameter werden für die Bestimmung der Koordinaten und der Größe des Menüobjekts benötigt.
- Die in der 14. Zeile stehende Zahl entspricht der Anzahl der Untermenüeinträge.
- *GUIObjMenue.O_VERTICAL*
Dieser Parameter ist vom Typ *int* und beschreibt die vertikale Darstellung dieses Menüs.
- *0xFBD8C6*
Dieser hexadezimale Parameter stellt die Grundfarbe des Menüs dar.

In der 22. Zeile werden dem Menü die *SWITCH CASE*-Bezeichnungen übergeben, welche bei Betätigung des jeweiligen Falles aufgerufen werden sollen.

13.3.2. Google Maps Interface

Diese Schnittstelle zum Server von Google Maps wurde von dem Franzosen Thomas Landspurg entwickelt. Es stellt grundlegende Klassen und Methoden bereit, die zur Kommunikation mit Google Maps nötig sind.

Dieses orientiert sich nach Koordinaten, die als östliche Länge (*Longitude*) und nördliche Breite (*Latitude*) kodiert werden und lädt 3x3 benachbarte Kartenausschnitte vom GoogleMaps-Server herunter. Das Hauptinterface stellt das MapCanvas dar, dessen Instanz die benachbarten Kartenausschnitte auf dem Display darstellt.

```
mapCanvas c=new mapCanvas();
```

Durch den Constructor *oneLoc(float lon, float lat)* lassen sich einzelne Punkte anhand ihrer Koordinaten darstellen. Danach lässt sich die Karte um den Punkt (oder auch mehrere) zentrieren.


```
c.listPlaces.addElement(new oneLoc(getLoc("Dom",lat),getLoc("Dom",lon));
c.recenterMapAroundSel();
```

Um die aktuelle Position darstellen zu können, schrieben wir eine Funktion zum Ermitteln der GPS-Daten. Diese Funktion (*getGPS()*) verwendet zunächst die aus *discoverBlueTooth()* ermittelte Adresse der GPS-Mouse und öffnet eine *StreamConnection*. Der dabei ausgelesene String des NMEA-Protokolls wird an die *output*-Variable übergeben, welche wiederum zur Weiterverarbeitung zurückgegeben wird.

Listing 13.3: die *getGPS()* Methode

```
1 private String getGPS() {
2     try
3     {
4         String url = this.discoverBlueTooth();
5         javax.microedition.io.StreamConnection connection = (
6             StreamConnection) Connector.open(url, Connector.READ);
7         java.io.InputStreamReader reader = new InputStreamReader(
8             connection.openInputStream());
9         String output = new String();
10        int input;
11        while ((input = reader.read()) != 13)
12            output += (char) input;
13        output = output.substring(1, output.length() - 1);
14        System.out.println(output);
15        return output;
16    }
17    catch (IOException exc){exc.printStackTrace(); }
18 }
```

Der zurückgegebene String sieht dann beispielsweise so aus:

```
GPRMC,184846.841,A,5057.4999,N,0653.5999,E,0.00,349.08,310107,, ,A*6
```

13.4. Server

13.4.1. Das Zamspack-Servlet

Das Zamspack-Servlet ist einfach gehalten. Anfragen kommen als HTTP-POST teils auch HTTP-GET an, dabei wird jeweils das *HttpServletRequest*- und *HttpServletResponse*-Objekt übergeben. Nach Empfang der Anfrage wird zuerst die enthaltene Nachricht geparkt. Nachrichten, die falsch formatiert sind, werden nicht weiter bearbeitet. Nun wird die Art der Anfrage festgestellt und die entsprechende Funktion des Servlets aufgerufen.

Listing 13.4: die doPost() Methode des Servlets

```

1 public void doPost(HttpServletRequest request, HttpServletResponse
   response)
2 {
3     ...
4     operation = request.getParameter("op");
5     if(operation.equals("get")) xmlDoc.getP0I(request, response);
6     else if(operation.equals("put")) xmlDoc.putP0I(request, response);
7     else xmlDoc.sendError(response);
8     ...
9 }
```

Diese ermittelt die nötigen Daten und führt Änderungen in der Datenbank aus. Im letzten Schritt wird die Antwort als XML erzeugt und zurückgesendet.

Folgende XML-Struktur zeigt eine mögliche Antwort des Servlets:

```

<?xml version="1.0"?>
  <items>
    <item>
      <title>Mongos</title>
      <description>leckerer Restaurant</description>
      <geo:lat>50.9402</geo:lat>
      <geo:long>6.9723</geo:long>
      <icon>http://www.gm.fh-koeln.de/~mi350/navihand/food.png</icon>
      <tel>0221 989 38 10</tel>
    </item>
    <item>
      <title>Sorento</title>
      <description>italienische und persische Kueche</description>
      <geo:lat>50.9215</geo:lat>
      <geo:long>6.9336</geo:long>
      <icon>http://www.gm.fh-koeln.de/~mi350/navihand/food.png</icon>
```

```
      <tel>null</tel>
    </item>
  </items>
```

13.4.2. MySQL-Datenbank

Das folgende Diagramm zeigt den Aufbau der Datenbank. Die dazugehörigen MySQL-Anweisungen sind im Anhang zu finden.

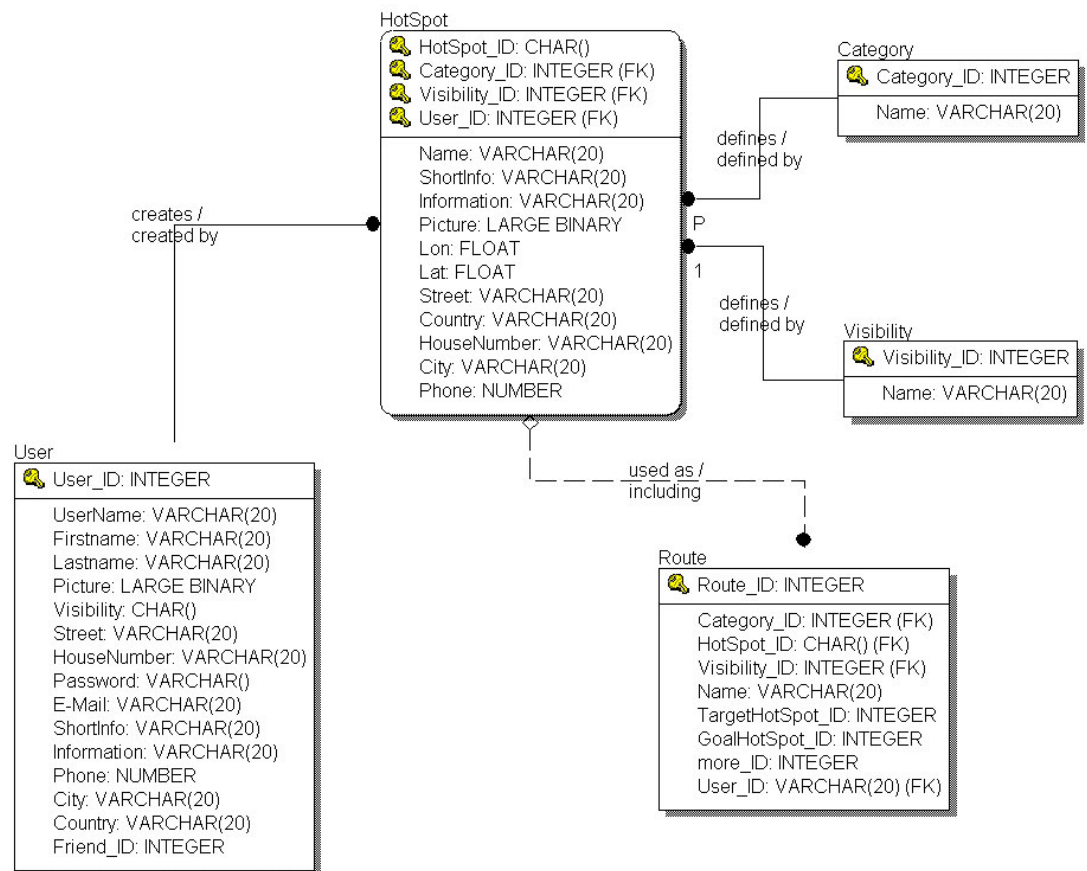


Abbildung 13.3.: ER-Diagramm der Datenbankstruktur

Teil IV.

Fazit

Die gestellte Aufgabe, ein Java-basiertes Navigationssystem für GPS-fähige mobile Endgeräte zu konzipieren wurde gründlich bearbeitet und erfolgreich abgeschlossen. Wir modellierten ein einsatzfähiges mobiles System, welches ausführlich mit potenziellen Endbenutzern evaluiert wurde. Als Hauptbestandteil galt es, eine hohe Usability zu erreichen, um den konzeptuellen Teil der Arbeit zur Entwicklung eines solchen Navigationssystem in anderen Projekten erneut einsetzen zu können. Dazu wurde eine Probandenbefragung durchgeführt, die wichtige Ergebnisse lieferte, die aus Sicht von Entwicklern nicht bedacht wurden.

Es wurden sämtliche Teilfunktionalitäten ausgehend von den Zielbestimmungen detailliert untersucht und im Pflichtenheft beschrieben. Die Funktion, eine Anbindung an den Datenbestand eines ÖPNV herzustellen, ist aus bereits genannten Gründen nicht erfolgt.

Die Implementierung des Prototypen wurde auch weitestgehend erfolgreich bearbeitet. Weitestgehend deshalb, weil sich die einzelnen Teilbereiche des J2MEMap-Interfaces und des MIDlet-Frameworks bis dato nicht kombinieren ließen. Somit sind die Funktionen die für den Prototypen ausgewählt wurden zwar verfügbar, jedoch nicht über die GUI erreichbar.

Die Vorteile der gewählten Architektur, die Trennung von Präsentation, Anwendungslogik und Datenhaltung, wurden konsequent umgesetzt und bieten für die Weiterführung des Projekts die nötige Freiheit, um Komponenten auszutauschen oder neue Komponenten hinzuzufügen.

Teil V.

Quellenangaben

Bundesamt für Statistik: 80-Prozent-Marke bei der Handy-Ausstattung überschritten

(URL: http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Presse/pm/zdw/2007/PD07__019__p002.psml)

Classic Waterfall Model in Software Engineering

(URL: <http://www.medien.ifl.lmu.de/lehre/ws0607/mmi1/essays/Anette-Grimm/>)

Entwicklung mobiler Anwendungssysteme

von Alban Schmidt (Autor) Verlag: Vdm Verlag Dr. Müller; Auflage: 1 (Januar 2007)

Erfahrungen mit dem Location API (JSR 179)

(URL: <http://www.gruntz.ch/papers/LocationAPI.pdf>)

J2ME - CDC/CLDC

http://www.ifl.unizh.ch/riedl/lectures/Vortrag_CLDC_Teil1.pdf

Java Server und Servlets. Portierbare Web-Applikationen effizient entwickeln

von Schreiber Roßbach (Autor) Verlag: Addison-Wesley; Auflage: 2. Aufl. (15. Juli 2000)

Methoden der Usability Evaluation. Wissenschaftliche Grundlagen und praktische Anwendung

von Florian Sarodnick (Autor), Henning Brau (Autor) Verlag: Huber, Bern; Auflage: 1 (Mai 2006)

Mobile SVG

(URL: <http://www.forteach.net/xmls/multimedia/33399.html>)

Scalable Vector Graphics

(URL: <http://www.scale-a-vector.de/mobil.htm>)

Scalable Vector Graphics (SVG) Tiny 1.2 Specification

(URL: <http://www.w3.org/TR/SVGMobile12/>)

Software Ergonomie umsetzen - nicht ohne den Anwender

(URL: http://www.sovt.de/PDF/SW-Ergonomie_umsetzen.pdf)

SUN: JSR 118: Mobile Information Device Profile 2, SUN Microsystems Inc.

(URL: <http://jcp.org/aboutJava/communityprocess/mrel/jsr118i>)

SUN: JSR-139 Connected Limited Device Configuration 1.1, SUN Microsystems Inc.
(URL: <http://jcp.org/aboutJava/communityprocess/final/jsr139i>)

SUN: JSR-179 Location API for J2ME, SUN Microsystems Inc.
(URL: <http://jcp.org/aboutJava/communityprocess/mrel/jsr179i>)

Vorgehensmodelle zum Softwareentwicklungsprozess
(URL: <http://www.torsten-horn.de/techdocs/sw-dev-process.htm>)

Teil VI.

Anhang

14. Fragenkatalog

Probandenbefragung zum Prototyp

Fragen zum Probanden:

Name:

Alter:

eMail-Adresse:

Wohnort:

Beruf:

Benutzungsverhalten vom Handy:

Handyvertrag:

UMTS:

Grund des Aufenthalts:

Aufgaben:

1. Finden Sie eine schnelle Möglichkeit, den Weg nach Hause zu berechnen.
2. Erstellen Sie Ihren aktuellen Standort als neuen "Point of Interest".
3. Ermitteln Sie den Weg von hier zu einem Subway-Restaurant in der Nähe der Universität.
4. Fügen Sie den Point of Interest "Dom" zu einer virtuellen Tour hinzu und zeigen diese an.
5. Schauen Sie, ob sich Ihr Freund Richie in Ihrer Nähe befindet.

Abschlussfragen:

Unter welchen Umständen können Sie sich vorstellen das Programm zu benutzen?

Was hätten Sie sich gewünscht?

Hätten Sie noch Ideen?

Fanden Sie es unübersichtlich?

Welche Funktion könnte für Sie die meist benutzte sein?

Sonstiges:

15. MySQL-Anweisungen der Datenbank

Listing 15.1: MySQL Schema

```

1  --
2  -- Tabellenstruktur für Tabelle 'category'
3  --
4
5  CREATE TABLE 'category' (
6      'Category_ID' int(1) NOT NULL auto_increment,
7      'Name' varchar(20) collate latin1_general_ci NOT NULL,
8      'icon' varchar(100) collate latin1_general_ci default NULL,
9      PRIMARY KEY ('Category_ID')
10 ) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci
    AUTO_INCREMENT=3 ;
11
12  -- -----
13
14  --
15  -- Tabellenstruktur für Tabelle 'hotspot'
16  --
17
18  CREATE TABLE 'hotspot' (
19      'HotSpot_ID' int(11) NOT NULL auto_increment,
20      'Category_ID' int(11) NOT NULL,
21      'Visibility_ID' int(11) NOT NULL,
22      'User_ID' varchar(20) collate latin1_general_ci NOT NULL,
23      'title' varchar(20) collate latin1_general_ci NOT NULL,
24      'description' varchar(100) collate latin1_general_ci default NULL,
25      'photoURL' varchar(100) collate latin1_general_ci default NULL,
26      'lon' float NOT NULL,
27      'lat' float NOT NULL,
28      'Street' varchar(20) collate latin1_general_ci default NULL,
29      'op' varchar(20) collate latin1_general_ci default NULL,
30      'HouseNumber' varchar(20) collate latin1_general_ci default NULL,
31      'City' varchar(20) collate latin1_general_ci default NULL,
32      'tel' char(15) collate latin1_general_ci default NULL,
33      PRIMARY KEY ('HotSpot_ID','Category_ID','Visibility_ID','User_ID'),
34      KEY 'User_ID' ('User_ID'),
35      KEY 'Visibility_ID' ('Visibility_ID'),
36      KEY 'Category_ID' ('Category_ID')

```

```

37 ) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci
    AUTO_INCREMENT=40 ;
38
39 -- -----
40
41 --
42 -- Tabellenstruktur für Tabelle 'route'
43 --
44
45 CREATE TABLE 'route' (
46   'Route_ID' int(11) NOT NULL,
47   'Category_ID' int(11) NOT NULL,
48   'HotSpot_ID' char(1) collate latin1_general_ci NOT NULL,
49   'Visibility_ID' int(11) NOT NULL,
50   'Name' varchar(20) collate latin1_general_ci default NULL,
51   'StartHotSpot_ID' int(11) NOT NULL,
52   'GoalHotSpot_ID' int(11) NOT NULL,
53   'more_ID' int(11) default NULL,
54   'User_ID' varchar(20) collate latin1_general_ci default NULL,
55   PRIMARY KEY ('Route_ID'),
56   KEY 'HotSpot_ID' ('HotSpot_ID','Category_ID','Visibility_ID','User_ID')
57 ) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
58
59 -- -----
60
61 --
62 -- Tabellenstruktur für Tabelle 'user'
63 --
64
65 CREATE TABLE 'user' (
66   'User_ID' int(11) NOT NULL,
67   'UserName' varchar(20) collate latin1_general_ci NOT NULL,
68   'Firstname' varchar(20) collate latin1_general_ci NOT NULL,
69   'Lastname' varchar(20) collate latin1_general_ci NOT NULL,
70   'Picture' blob,
71   'Visibility' char(1) collate latin1_general_ci NOT NULL,
72   'Street' varchar(20) collate latin1_general_ci default NULL,
73   'HouseNumber' varchar(20) collate latin1_general_ci default NULL,
74   'Password' varchar(30) collate latin1_general_ci NOT NULL,
75   'E-Mail' varchar(20) collate latin1_general_ci NOT NULL,
76   'ShortInfo' varchar(20) collate latin1_general_ci default NULL,
77   'Information' varchar(20) collate latin1_general_ci default NULL,
78   'Phone' char(1) collate latin1_general_ci default NULL,
79   'City' varchar(20) collate latin1_general_ci default NULL,
80   'Country' varchar(20) collate latin1_general_ci default NULL,
81   'Friend_ID' int(11) default NULL,
82   PRIMARY KEY ('User_ID')
83 ) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
84

```

```
85  -- -----
86
87  --
88  -- Tabellenstruktur für Tabelle 'visibility'
89  --
90
91  CREATE TABLE 'visibility' (
92      'Visibility_ID' int(11) NOT NULL auto_increment,
93      'Name' varchar(20) collate latin1_general_ci NOT NULL,
94      PRIMARY KEY ('Visibility_ID')
95  ) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci
      AUTO_INCREMENT=3 ;
```

Teil VII.
Erklärung

Wir versichern hiermit, die von uns vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen worden sind, haben wir als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die wir für die Arbeit benutzt haben, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Zoya Ghoraishi Kaverdi

Axel Freudiger