

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL



INSTITUT
FÜR
SOFTWARE

EventAssistant

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2010

Autoren: Felix Egli und Michael Schnyder
Betreuer: Prof. Dr. Markus Stolze

Impressum

Kontakt

Felix Egli, felix.egli@hsr.ch

Michael Schnyder, michael.schnyder@hsr.ch

Copyright 2010

Felix Egli & Michael Schnyder

Druckdatum, 23.12.2010

Erklärung

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

Rapperswil, den 23.12.2010



Felix Egli



Michael Schnyder

Danksagung

Für die Unterstützung während der Bachelorarbeit möchten wir folgenden Personen einen besonderen Dank ausrichten:

Prof. Dr. Markus Stolze, für seine Unterstützung und sein wertvolles und konstruktives Feedback.

Remo Egli, für Paperprototyp Test und wertvollen Input.

Anita Hollenstein, für die Teilnahme Paperprototype-und Usability-Test und wertvollen Input.

Christoph Carl, für die Teilnahme am Usability Test und wertvollen Input.

Mischa Trecco super LaTeX Vorlage.

Yves Thrier, Sebastian Hunkeler für das Korrekturlesen.

Daniel Häfliger für die Erstellung des CD-Covers.

Abstract

Wer kennt es nicht? Gute Stimmung an einem Openair und plötzlich sind Fragen aktuell wie z.B.:

- Wer spielt als nächstes auf dieser Bühne?
- Wo befindet sich der Zeltplatz?

Solche Fragen können bis anhin nur über Infowände oder extra Festführer beantwortet werden. Mit EventAssistant, dem mobilen Begleiter für Grossevents (z.B. Openairs), haben wir für diese Probleme eine optimale Lösung zu entwickelt, indem die folgenden Features implementiert wurden:

- Unterstützung mehrerer Events
- Bestimmen des aktuellen Standorts auf einer vom Organisator individuell hinterlegten Karte
- Auf der Karte können Details zu georeferenzierten Punkten (z.B. einer Bühne) hinterlegt werden
- Diese georeferenzierten Punkte können auch einen zeitlichen Ablauf besitzen

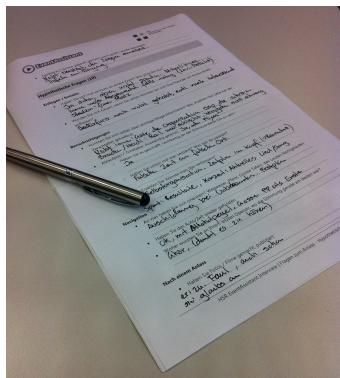
Jeder Veranstalter muss ab sofort nicht für jeden Event eine eigene Applikation entwickeln, sondern nur den Inhalt seines Events erfassen. Diese Inhalte werden auf einem zentralen Server verwaltet. Somit bleibt die Applikation auf dem Mobiltelefon bestehen und nur die Daten müssen aktualisiert werden. Bereits heruntergeladene Events sind auch ohne Internet verfügbar, werden aber nach Verfall und verfügbarer Internetverbindung aktualisiert. Die Applikation wurde für die Android-Plattform entwickelt. Die Kommunikation zwischen Server und Mobiltelefon erfolgt über JSON/HTTP. Auf dem Handy wurde das GUI komplett von den Daten entkoppelt. Dabei wurde die Architektur auf Basis der Best-Practices von Android implementiert.

Management Summary

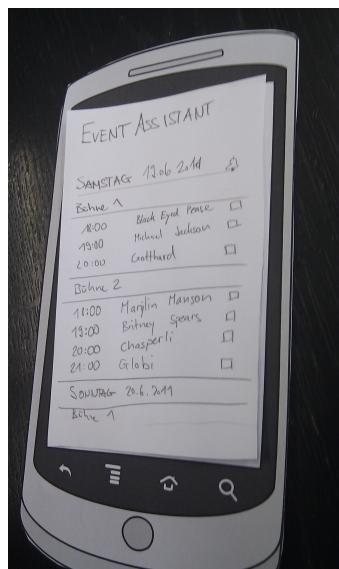
Wer kennt es nicht? Gute Stimmung an einem Openair und plötzlich sind Fragen aktuell wie z.B.:

- Wer spielt als nächstes auf dieser Bühne? Wo befindet sich der Zeltplatz?
- Wo befindet sich der Zeltplatz?

Solche Fragen konnten bis anhin nur über Infowände oder extra Festführer beantwortet werden. Dies ist natürlich immer mit grossem finanziellem Aufwand für den Veranstalter verbunden. Mit Benutzerinterviews konnten wir viele Anforderungen der Besucher erfassen und in Requirements transformieren. Diese Evaluation fand direkt mit Besuchern von Grossevents statt. Die abgeleiteten Features wurden in einen Papier-GUI-Prototyp umgewandelt und von Probanden getestet. Auch zum Schluss des Projektes haben wir die Applikation durch Probanden nochmals testen lassen.



(a) Feature-Evaluation mit Interviews



(b) GUI-Evaluation mit Papierprototyp



(c) Usabilitytest

Abbildung (1) Evaluation

Mit unserem Endprodukt, EventAssistant, dem mobilen Begleiter für Grossevents (z.B. Openairs), haben wir für die erfassten Requirements eine optimale Lösung entwickelt. Folgende Features wurden implementiert:

- Unterstützung mehrerer Events in der selben Applikation mit automatischer Weiterleitung zum aktuellen Event mittels GSM-Ortsbestimmung

- Bestimmen des aktuellen Standorts auf einer vom Organisator individuell hinterlegten Karte
- Auf der Karte können Details zu georeferenzierten Punkten (z.B. einer Bühne) abgefragt werden
- Diese georeferenzierten Punkte können auch einen Zeitplan beinhalten
- Allgemeine Infos über einen Event sind abrufbar
- Offline-Modus: Die Applikation ist auch ohne Handynetzt komplett funktionsfähig.

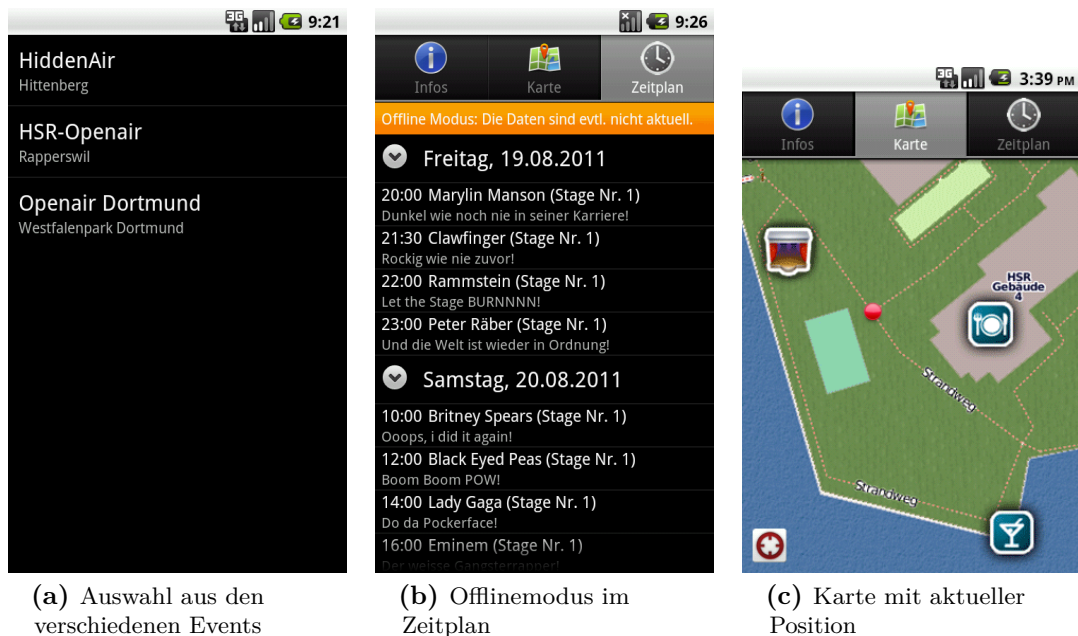


Abbildung (2) Applikation Screenshots

Jeder Veranstalter muss ab sofort nicht für jeden Event eine eigene Applikation entwickeln lassen. Er muss nur noch den Inhalt seines Events erfassen. Diese Inhalte werden auf einem zentralen Server verwaltet. Dieser Server besitzt eine RESTful Schnittstelle mit JSON und kann mit einer beliebigen Technologie implementiert werden. Die Referenzimplementation haben wir auf basis von einem JavaServlet erstellt.

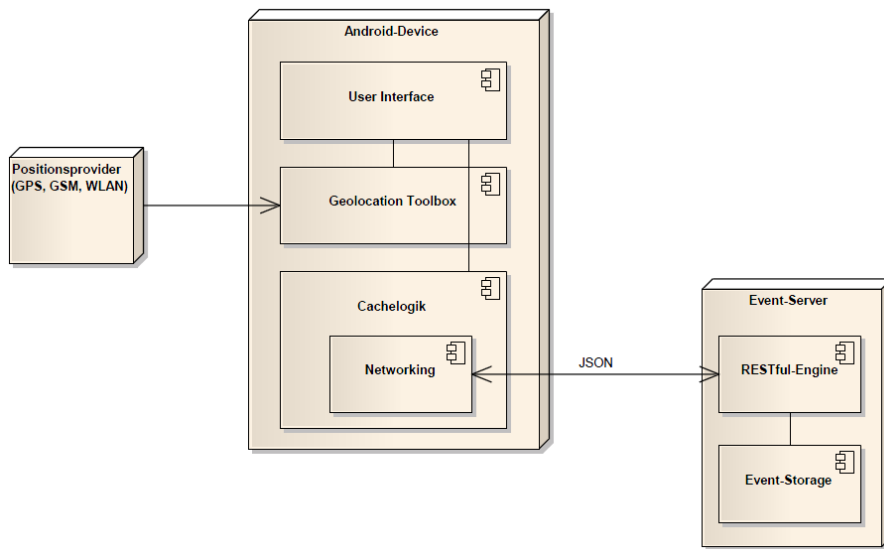


Abbildung (3) Schematische Darstellung

Die Applikation ist von den darin angezeigten Daten entkoppelt, dadurch bleibt die Applikation auf dem Client bestehen und nur die Daten müssen aktualisiert werden. Bereits heruntergeladene Events sind auch ohne Internet verfügbar, werden aber bei Verfall und verfügbarer Internetverbindung aktualisiert. Damit dies auch korrekt funktioniert sorgt eine CacheLogik die die Daten bei jedem Aufruf auf deren Verfall prüft. Damit am Event der Eventbesucher genau über GPS geortet werden kann, sorgt die Geolocation Toolbox. Diese beiden Teilbereiche werden vom User Interface angesprochen.

Die Applikation wurde für die Android-Plattform entwickelt. Bei der Implementation wurden speziell darauf geachtet Best-Practices in der Android-Entwicklung umzusetzen. Dies betraf besonders die Richtlinien über Ressourcenschonung, Performance und Usability.

Inhaltsverzeichnis

I	Technischer Bericht	1
1	Einführung	3
1.1	Problemstellung, Vision	3
1.1.1	Besucher	3
1.1.2	Veranstalter	4
1.1.3	Ausrichtung	4
1.2	Aufgabenstellung, Ziele	4
1.3	Umfeld der Arbeit	5
1.3.1	Rahmenbedingungen	5
1.3.1.1	Aufwand	5
1.3.1.2	Betreuung, Partner	5
1.3.1.3	Wichtige Termine	6
1.3.2	Infrastruktur	6
1.3.2.1	Hardware	6
1.3.2.2	Entwicklungsumgebung	7
1.3.2.3	Organisation	7
1.3.3	Glossar	7
1.3.4	Abkürzungsverzeichnis	7
1.3.5	Abgrenzungen	8
1.3.5.1	Ausrichtung	8
1.3.5.2	Administrations-Backend	8
1.3.5.3	Internationalisierung	8
1.3.5.4	Device Kompatibilität	8
1.4	Vorgehen, Aufbau der Arbeit	8
2	Stand der Technik	11
2.1	Bestehende Arbeiten	11
2.1.1	IndoorGuide4Android	11
2.1.2	IndoorGuide4Android II	11
2.2	Konkurrenzprodukte	12
2.2.1	Swisscom Openair Buddy	12
2.2.2	Festival Mob	13
2.2.3	Spezifische Mobile-Apps von Veranstaltern	14
2.2.4	Webseite der Veranstalter	14
2.2.5	Facebook-Places	15

3	Evaluation	17
3.1	Technologieentscheid Mobile Plattform	17
3.1.1	Bewertungskriterien	17
3.1.1.1	Aufwandschätzungen (je geringer der Aufwand, desto höher bewertet)	17
3.1.1.2	Potential	18
3.1.2	Bewertung	18
3.1.2.1	Funktionieren ohne Mobil/Datennetz	18
3.1.2.2	Karte anzeigen	19
3.1.2.3	Karte nach Norden ausrichten	19
3.1.2.4	Frameworks, GUI-Hilfen, Community	19
3.1.2.5	Lokalisierung mittels API und GPS	19
3.1.2.6	Strukturierter Lokaler Datenspeicher	19
3.1.2.7	Reminder Funktionen	20
3.1.2.8	Facebook Integration	20
3.1.2.9	Push Notifications	20
3.1.2.10	Verbreitungsgrad auf Mobiles	20
3.2	Map-Control	20
3.2.1	Variante „GoogleMapsAPI“	21
3.2.2	Variante „WebView (WebKit)“	21
3.2.3	Variante „ScrollView“	22
3.2.4	Variante „Eigenes Control“	22
3.3	Client-Server Kommunikation	22
3.3.1	Sockets	22
3.3.2	XML	23
3.3.3	JSON	23
3.3.4	Entscheid	23
3.4	Server-Architektur	23
3.4.1	Apache-Server mit PHP-Modul	23
3.4.2	Tomcat mit Servlet	24
3.4.3	ASP.NET-Server	24
3.4.4	Entscheid	24
3.5	Client-Architektur	24
3.5.1	Parsing	24
3.5.2	Zentraler Punkt	24
3.5.3	Cache	25
4	Umsetzungskonzept	27
4.1	Allgemein	27
4.2	Android	27
4.3	Daten und Protokolle	27
4.4	Organisation	28

5	Resultate	29
5.1	Zielerreichung	29
5.1.1	Funktionale Anforderungen	29
5.1.1.1	Erfüllte Use Cases	29
5.1.1.2	Optionale Use Cases	30
5.1.1.3	Abgegrenzte Use Cases	30
5.1.2	Nichtfunktionale Anforderungen	30
5.1.2.1	Zentrale Speicherung	30
5.1.2.2	Geo-Referenzierung	31
5.1.2.3	Performance	31
5.1.2.4	UI-Responsiveness	31
5.1.2.5	Akkuverbrauch / Systemdesign	31
5.1.3	Fazit	33
5.2	Ausblick	33
5.3	Persönliche Berichte	34
5.3.1	Felix Egli	34
5.3.1.1	Projektverlauf	34
5.3.1.2	Rückblick	35
5.3.1.3	Gelerntes	35
5.3.1.4	Fazit	35
5.3.2	Michael Schnyder	36
5.3.2.1	Projektverlauf	36
5.3.2.2	Rückblick	37
5.3.2.3	Gelerntes	37
5.3.2.4	Fazit	37
5.4	Lessons Learned	38
II	SW-Projektdokumentation	39
6	Anforderungsspezifikation	41
6.1	Personas	41
6.1.1	Peter Planer	41
6.1.2	Susi Sorglos	43
6.1.3	Behavioural Variables	44
6.2	Szenarios	44
6.2.1	Eventplanung am Tag zuvor	44
6.2.2	Preise und Tickets	44
6.2.3	Eventablauf	44
6.2.4	An MiniEvent erinnern lassen	45
6.2.5	Wiederfinden des Autos	45
6.2.6	Orientierung am Event	45

6.2.7	Fotos	45
6.2.8	Facebook	45
6.2.9	Verschiebung	46
6.2.10	Stimmung	46
6.3	Priorisierung der Szenarien	46
6.3.1	Must	46
6.3.2	Optional	46
6.3.3	Abgrenzung	47
6.4	Use Cases	48
6.4.1	Übersicht	48
6.4.2	Besucher	49
6.4.2.1	UC B01: Festivalablauf ansehen	49
6.4.2.2	UC B02: Karte ansehen	49
6.4.2.3	UC B03: Erinnerungen	49
6.4.2.4	UC B04: Freunde Finden	49
6.4.2.5	UC B05: Eigene Fotos veröffentlichen	50
6.4.2.6	UC B06: Einstellungen verwalten	50
6.4.3	Veranstalter	50
6.4.3.1	UC V01: Pushnachricht senden	50
6.4.3.2	UC V02: Festdaten Erfassen	50
6.5	Nichtfunktionale Anforderungen	50
6.5.1	Speicherung	50
6.5.2	Geo-Referenzierung	51
6.5.3	Client Cache	51
6.5.4	Performance	51
6.5.5	UI-Responsiveness	51
6.5.6	Akkuverbrauch / Ressourcen schonen	51
7	Analyse	53
7.1	Domain Model	54
7.2	Objektkatalog	55
7.2.1	Event	55
7.2.2	Map	55
7.2.3	PointOfInterest	55
7.2.4	TimedEvent	55
7.2.5	Eventorganisator	55
7.2.6	Visitor	55
7.2.7	Pushnachricht	55
7.2.8	Benachrichtigung	55
7.2.9	Foto	56

8	Design	57
8.1	Architektur	57
8.1.1	Komponenten	58
8.1.1.1	Positionsprovider	58
8.1.1.2	Android-Device	58
8.1.1.3	Event-Server	58
8.1.2	Android-Konzepte	58
8.2	Packagestruktur	59
8.2.1	Projekt „EventAssistant-Common“	59
8.2.2	Projekt „EventAssistant“	59
8.2.2.1	UI-Package	60
8.2.2.2	Provider-Package	60
8.2.2.3	Util-Package	60
8.2.2.4	Data-Package	60
8.2.3	Projekt „EventAssistant-Server“	61
8.2.4	Projekt „EventAssistant-Test“	61
8.3	Klassendiagramm	62
8.4	Datenbankschema	63
8.5	Schnittstellen und Formate	63
8.5.1	Schnittstelle zu ContentProvider	63
8.5.1.1	Parameter - Grundlagen	63
8.5.1.2	Ressourcen	64
8.5.1.3	Beispiel	65
8.5.2	Event-Assistant Server (REST-Schnittstelle)	65
8.5.2.1	Eventliste	66
8.5.2.2	Einzelner Event	67
8.5.2.3	Karte (Bild) für Event	67
8.5.2.4	Zeitplan für Event	68
8.5.2.5	Zeitplan für POI	69
8.5.2.6	Koordinaten Polygon	69
8.5.2.7	Alle POI's für Event	70
8.5.2.8	Spezifischer POI's von Event	71
8.5.3	Koordinatenformat	71
8.5.4	Datenformat für Karten	72
8.6	UI-Design	72
8.6.1	Allgemein	72
8.6.2	Organisation	72
8.6.3	Beschreibung der Screens	72
9	Implementation	75
9.1	Prototypen	75
9.1.1	BrowserMapPrototype	75

9.1.2	MapOverlayPrototype	75
9.1.3	ScrollViewPrototype	76
9.1.4	LocationGsmCell	76
9.1.5	CommunicationPrototype	76
9.1.6	JSONMarshaller	76
9.1.7	MessageArchitecture	76
9.1.8	ServerServlet	77
9.2	Implementations-Konzepte	78
9.2.1	Datenklassen Vererbung	78
9.2.1.1	Beispiel am Property „name“ eines Events	78
9.2.2	ContentValues und Cursors	79
9.2.3	Content Provider und Lokaler Cache	81
9.2.3.1	Ablauf - SSD	81
9.2.3.2	Offline-Notification	82
9.2.4	Kommunikationsprotokoll	82
9.2.5	Referenz-Server	83
9.3	Widgets und UI-Helper	86
9.3.1	MapView-Widget	86
9.3.1.1	Georeferenzierung	86
9.3.1.2	Positionsdarstellung	87
9.3.1.3	Interaktive Regionen	87
9.3.1.4	Fehlerbereinigte Events	88
9.3.2	ExpandableListCursorAdapter	88
9.3.2.1	XML-Markup für Gruppen- und Child-Einträge	89
9.3.2.2	OnViewCreatedCallbackHandler	89
9.3.2.3	ValueBinder für Gruppierung	90
9.3.2.4	ValueBinder für Einträge	90
9.3.2.5	Adapter Instanz	91
9.3.3	NetworkInfoView	91
9.3.4	I18nDialogHelper	91
10	Testing	93
10.1	Unit-Tests	93
10.1.1	Android	93
10.1.2	Datenklassen	94
10.1.3	Database-Fake-Test	95
10.1.4	Android-Testabdeckung	95
10.2	Systemtests	95
10.2.1	Systemtest vom 3.12.2010	96
10.2.2	Systemtest vom 19.12.2010	96

10.3 Usability Tests	96
10.3.1 Papierprototyp Nr. 1	96
10.3.1.1 Aussehen	96
10.3.1.2 Vorgehen	96
10.3.1.3 Resultat	97
10.3.1.4 Redesignentscheide	97
10.3.2 Papierprototyp Nr. 2	97
10.3.2.1 Aussehen	97
10.3.2.2 Vorgehen	98
10.3.2.3 Resultat	99
10.3.2.4 Redesignentscheide	100
10.3.2.5 Weiteres Vorgehen	100
10.3.3 Usability-Check Test vom 3.12.2010	100
10.3.3.1 Vorgehen	100
10.3.3.2 Redesignentscheide	100
10.3.4 Usability-End Test	101
10.3.4.1 Aussehen	101
10.3.4.2 Vorgehen	102
10.3.4.3 Resultat	103
10.3.4.4 Entscheide	103
11 Resultate und Weiterentwicklung	105
11.1 Resultate	105
11.2 Möglichkeiten der Weiterentwicklung	105
11.2.1 Feature Notifikation	105
11.2.2 Backend / Datenbank / Web-GUI	108
11.2.3 Optionale Features	108
12 Projektmanagement	109
12.1 Entwicklungsumgebung	109
12.2 Projektplan	109
12.2.1 Zeitplan	109
12.2.2 Iterationsplanung	109
12.2.3 Meilensteine	110
12.2.4 Meetings	110
12.3 Team und Verantwortlichkeiten	110
12.4 Risikomanagement	112
12.5 Prozessmodell	113
12.6 Q-Massnahmen	113

13	Projektmonitoring	115
13.1	Soll-Ist-Zeitvergleich	115
13.1.1	Haupttätigkeiten	115
13.1.2	Soll / Ist Vergleich	117
13.2	Codestatistik	118
13.2.1	Bemerkungen	119
13.3	Protokolle	119
13.4	Code-Reviews	119
14	Softwaredokumentation	121
14.1	Installation	121
14.1.1	Android-Client	121
14.1.2	Referenz-Server	121
14.1.3	Entwicklungsumgebung	122
14.2	Content Erfassung	122
14.2.1	Geokoordinaten	123
14.2.2	Geo-Referenzierung	123
	Literaturverzeichnis	125
	Glossar	127
	Abkürzungsverzeichnis	129
	Abbildungsverzeichnis	131
	Tabellenverzeichnis	133
	Anhang	133
A	Aufgabenstellung	135
B	Vereinbarung	141
C	Projektplan	143
D	Interviewfragebogen Freestyle	147
E	Interviewauswertung Freestyle	157
F	Interviewfragebogen Züspa	163
G	Interviewauswertung Züspa	167

H	Installationsanleitung LaTeX	171
I	Papierprototype Nr. 1	177
J	Android Coverage Report	181
K	Systemtest Vorlage	185
L	Systemtest Protokoll vom 19.12.2010	189
M	Sitzungsprotokolle	193
N	Codemetriken	205
	N.1 Codemeteriken Client	205
	N.2 Codemeteriken Server	214
	N.3 Codemeteriken Common	222
	N.4 Codemeteriken Test	230

Teil I

Technischer Bericht

1 Einführung

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	12.10.2010	Michael Schnyder	Dokument erstellt
0.2	13.10.2010	Michael Schnyder	Einführung erste Version
0.3	14.10.2010	Michael Schnyder	Einführung und Umfeld fertig-/erstellt
0.4	19.10.2010	Michael Schnyder	Umfeld fertiggestellt, Vorgehen erweitert
0.5	21.10.2010	Felix Egli	Papers referenziert
0.6	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

1.1 Problemstellung, Vision

Ortsbezogene Dienste oder „*Location Based Services*“ sind der grosse Renner auf Mobilfunkgeräten. Dienste wie Gowalla ¹, Foursquare ² und neuerdings Facebook Places³ sind vor allem in den USA stark verbreitet.[MP08][RCBH10][UB08] Auf solchen Plattformen wird anderen Benutzern mitgeteilt, in welchem Restaurant, Kino, Bar oder allgemein *Hotspot* man sich gerade befindet und wie man diesen bewertet. Andere Besucher erhalten einen auf Empfehlungen basierten Ortsplan. Der Betreiber dieses *Hotspots* kann häufig ein anwesende Besucher identifizieren und Sie mit Gutscheinen an sich binden.

Unser Kontext ist kleiner. Wir beschränken uns auf hilfreiche Funktionen innerhalb eines Grossanlasses wie z.B. einem Openair, einem grossen Stadtfest oder einer Sportveranstaltung.

1.1.1 Besucher

Die Orientierung an grossen Anlässen oder Festivals bereitet vielen Besuchern Probleme. Dank Festführern, Info-Plakaten und Übersichtskarten finden sich jedoch die meisten Besucher zurecht und oder haben sich der Situation angepasst. In diesem Punkt setzt unsere Idee an: Die allermeisten Besucher einer solchen Grossveranstaltung besitzen ein Internetfähiges Mobilfunkgerät der neuen Generation. Auf diesen würden sich diese Informationen wunderbar ortsbezogen darstellen lassen.

Interessante Informationen wären z.B. der Zeitplan für das Festival oder auch aktuelle News rund um den Anlass. Hintergrundinformationen (zu den einzelnen Bands beispielsweise) könnten ebenfalls über eine solche Applikation zugänglich gemacht werden.

1 <http://gowalla.com/>

2 <http://foursquare.com/>

3 <http://www.facebook.com/places/>

Die Integration von Sozialen Netzwerken würde neue interessante Funktionen für Besucher eines Grossanlasses ermöglichen (wie z.B. *Welche meiner Kontakte sind hier und wo sind Sie gerade auf dem Gelände? Wer kommt noch?*).

Mit der Verwendung des Bewegungssensors könnte herausgefunden werden, an welchem Ort des Geländes gerade am meisten Bewegung herrscht. Das wäre z.B. ein Indiz für gute Stimmung in einer Bar.

1.1.2 Veranstalter

Für den Veranstalter stehen genaue Bewegungsprofile der Besucher zur Verfügung. Dieser kann so Bewegungsströme besser erkennen und lenken. Verschiebungen im Zeitplan könnten ebenfalls sekundenschnell zu den Besuchern übermittelt werden.

Standbetreiber (z.B. der Betreiber einer Bar, welcher an jedem der grossen Openairs in der Schweiz eine Bar betreibt) hätten die Möglichkeit, jene Besucher zu belohnen, welche an jedem der Openairs mindestens 30min im Umkreis der Bar verbracht haben. Ein entsprechender Gutschein für ein Getränk würde auf das Mobilfunkgerät übermittelt wäre am letzten Openair der Saison einlösbar.

1.1.3 Ausrichtung

Als Teil dieser Arbeit werden die Bedürfnisse der Besucher erfasst, bzw. überprüft und danach priorisiert. Entsprechend der Priorisierung der Funktionen werden diese im Rahmen dieser Arbeit implementiert.

1.2 Aufgabenstellung, Ziele

Durch die zu entwickelnde Applikation erhält der Besucher von Grossanlässen die Möglichkeit, sich innerhalb von Grossanlässen (z.B. Openairs, Sportveranstaltungen, etc.) mittels Handy zu orientieren. Die Daten für die Navigation werden zentral gespeichert und je nach Standort des Benutzers die notwendigen Daten auf das Endgerät übermittelt. Mögliche Funktionen

- Gebäude / POI auf Karte darstellen und mit Informationen versehen
- Zeltsuche / eigenes Zelt wiederfinden
- Facebook-Integration / Wo sind meine Freunde am Openair?
- Direktes senden von Fotos an LED-Wand
- Wichtige Informationen per Push
- Sportresultate...

Als Teil der Arbeit müssen Umfang und gewünschte Funktionen anhand von Interviews bestimmt werden. Ziel ist die Erstellung eines lauffähigen Systems mit wertvollen Funktionen. Parallel zur Anforderungsanalyse soll auch eine Technologiestudie stattfinden in der unter anderem *HTML5* und *Android* untersucht wird.

1.3 Umfeld der Arbeit

1.3.1 Rahmenbedingungen

1.3.1.1 Aufwand

Die zwei Studenten Felix Egli und Michael Schnyder bearbeiten gemeinschaftlich die Aufgabenstellung und versuchen möglichst alle als muss definierten Ziele zu erreichen. Es wird mit einem Aufwand von 18h pro Person und Woche gerechnet. Diese Zahl ergibt sich aus der Rechnung: 8 ECTS-Punkte à 30h Aufwand pro ECTS-Punkt. Die Arbeit dauert knapp 14 Wochen. Das Wochenpensum entspricht demnach 240h / 14 Wochen \approx 18h/Woche. Lohnkosten sind über die gesamte Projektdauer keine vorgesehen.

1.3.1.2 Betreuung, Partner

Die Arbeit wird von Herrn Dr. Prof. Markus Stolze, Partner am Institut für Software IFS an der Hochschule Rapperswil, betreut. Bei Fragen rund um die Event-Organisation stehen dem Projektteam folgende Kontaktpersonen bei Swisscom zur Seite:

- Philipp Keller (Kommunikation)
- Peter Zimmerli (Projektmanager LiveEvents)

1.3.1.3 Wichtige Termine

<i>Datum</i>	<i>Termin</i>
16.09.2010	Pre-Kick-Off Meeting mit Betreuer
20.09.2010	Beginn der Studienarbeit
22.09.2010	Kick-Off Meeting mit Betreuer
20.12.2010	Abgabe Poster und Abstract
23.12.2010	Korrigiertes Abstract an Studiengangssekretariat
23.12.2010	Abgabe des Berichtes und CDs

Tabelle (1.1) Wichtige Termine

1.3.2 Infrastruktur

1.3.2.1 Hardware

<i>Gerät</i>	<i>Beschreibung</i>
2x Workstations	Windows 7 x64 Englisch, HSR-Domain
1x HTC Desire	Android Version 2.2, Zusammen mit der Snaplt-Gruppe
1x HTC Desire HD	Android Version 2.2, Privat von Michael Schnyder
1x Virtueller Server	Windows Server 2003 im HSR-Edu-Netz

Tabelle (1.2) Hardware

1.3.2.2 Entwicklungsumgebung

<i>Komponente</i>	<i>Beschreibung</i>	<i>Version</i>
IDE	Eclipse Java EE IDE for Web Developers	3.6 (Helios)
SDK	Java Platform, Enterprise Edition	6 SDK Update 1
SDK	Android-SDK,	Ver. r07 vom 23.09.2010
Eclipse-Plugin	Android Development Toolkit (Eclipse Feature)	0.9.9
Eclipse-Plugin	Subclipse (SVN-Client für Eclipse)	1.6.16
Tool	Tortoise SVN	1.6.10
Tool	Git (für Android Sourecode)	1.7.3.1
Tool	Enterprise Architect	8.0.863.13
Tool	Adobe Photoshop CS5 (Trial)	CS5
Tool	Paint.Net	3.5.6
Tool	Structure101 (Evaluation-Licence)	3.3 for Java
Dokumentation	MiKTeX	2.8
Dokumentation	Office 2010 Suite	10
Dokumentation	TeXnicCenter	2.0 Alpha 3 Build 1118

Tabelle (1.3) Entwicklungsumgebung

1.3.2.3 Organisation

<i>System</i>	<i>Beschreibung</i>
mite	Web-Basierte Zeiterfassung und Auswertung
trac	Wiki, Taskverwaltung und Sourcecode-Browser
Visual-SVN	SVN-Server (in Active Directory von Michael Schnyder integriert)

Tabelle (1.4) Organisations-Systeme

1.3.3 Glossar

Siehe Glossar auf Seite 127

1.3.4 Abkürzungsverzeichnis

Abkürzungen sind unterhalb der Seite ausgeschrieben. Siehe auch Abkürzungsverzeichnis auf Seite 129

1.3.5 Abgrenzungen

Nachfolgend Bereiche oder Funktionalitäten welche bei der Durchführung der Arbeit keine Relevanz besitzen. Somit werden die im folgenden genannten Funktionen explizit von der fertigen Arbeit nicht verlangt.

1.3.5.1 Ausrichtung

Der zu entwickelnde Prototyp richtet sich primär an einen bestimmten Festivaltyp. Dieser wurde vom Projektteam festgelegt und die Wahl fiel auf ein nicht näher definiertes grosses Openair.

1.3.5.2 Administrations-Backend

Die Arbeit verfolgt nicht das Ziel, eine Administrationsoberfläche zur Verwaltung der in der Applikation enthaltenen Daten zu erstellen.

1.3.5.3 Internationalisierung

Es wird nicht umgesetzt, dass die Daten in mehreren Sprachen zur Verfügung stehen. Bei korrekter Implementation sind Android-Applikationen bereits multilingual ausgelegt. Doch auch dieser Punkt ist nicht notwendigerweise zu erfüllen.

1.3.5.4 Device Kompatibilität

Die Mobilapplikation wird für eine Konfiguration erstellt und getestet. Die Mobilapplikation muss nicht Abwärtskompatibel sein. Der Betrieb der Applikation auf anderen Geräten ausser auf den unter (siehe *Hardware* auf Seite 6) aufgeführten Geräte ist nicht erforderlich.

1.4 Vorgehen, Aufbau der Arbeit

In einem ersten Schritt wurden vom Projektteam Funktionen zur Implementation gesammelt, welche das Team als besonders interessant erachtete. Dabei waren unter anderem die folgenden Kriterien ausschlaggebend:

- Technische Herausforderung
- Umsetzungskomplexität
- Relevanz für den Benutzer (aus eigener Erfahrung)

In einer zweiten Phase wurden diese Funktionen mit Interviews von realen Festivalbesuchern bestätigt. Siehe hierzu (siehe *Interviewfragebogen Freestyle* auf Seite 147) und (siehe *Interviewauswertung Züsä* auf Seite 167). Die Auswertung der Interviews ((siehe *Interviewfragebogen Freestyle* auf Seite 147) und (siehe *Interviewauswertung Züsä* auf Seite 167)) floss danach in die Priorisierung der Funktionen und in den Technologieentscheid (siehe *Technologieentscheid Mobile Plattform* auf Seite 17) ein.

Auf Basis der Hauptfunktionen wurde die Applikation entwickelt. Oft wurden zuerst Prototypen erstellt um sicherzustellen, dass die angedachte Funktion auch umgesetzt werden kann. Siehe dazu eine detaillierte Beschreibung aller Prototypen unter (siehe *Prototypen* auf Seite 75). Dies war auch zum Vorteil des Hauptprojekts, welches dadurch immer einen stabilen Stand hatte. Oft bestand die Arbeit des Projektteams mit Ausprobieren und Testen, da keines der Teammitglieder bisher Erfahrungen mit Android sammeln konnte. Dennoch war es dem Team wichtig, Prinzipien der Android-Entwicklung in das Projekt einfließen zu lassen.¹ Auf Serverseite wurde eine Referenzimplementation erstellt, welche noch fast beliebig erweitert werden kann. Diese dient aktuell im wesentlichen der Bereitstellung von Testdaten für die Applikation auf dem Android-Gerät.

Mit Benutzertests (Paperprototypen, Usability-Tests) wurde laufend überprüft, ob das entwickelte Userinterface den Bedürfnissen des Benutzers entspricht. Die aus den Tests erhaltenen Rückmeldungen sind direkt in die Weiterentwicklung eingeflossen.

Gegen Ende des Projektes hat sich das Projektteam entschieden auf den Einbau von Funktionen zu verzichten. Anstatt als optional klassifizierte Funktionen einzubauen, war es dem Team wichtiger, die implementierten Funktionen ausgiebig zu testen und abzuschließen.

¹ <http://developer.android.com/guide/practices/design/responsiveness.html>

2 Stand der Technik

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	03.11.2010	Michael Schnyder	Teildokument erstellt
0.2	15.11.2010	Michael Schnyder	Konkurrenzanalyse bis auf Webseite und Facebook Places erfasst
0.3	18.12.2010	Michael Schnyder	Konkurrenzanalyse fertiggestellt, bestehende Arbeiten erfasst
0.4	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

2.1 Bestehende Arbeiten

Die folgenden Arbeiten an der HSR befassen sich unter anderem mit der Lokalisierung von Android-Geräten und der Darstellung von orts-basierenden Zusatzinformationen. Es sind dies

2.1.1 IndoorGuide4Android

Im Rahmen dieser Bachelorarbeit wurde ein Indoor Informations- und Navigationssystem für die Android-Plattform entwickelt, welches seine Position mit Hilfe von Wireless LAN Signalen feststellt. Der IndoorGuide4Android stellt Informationen über sogenannte Points of Interest (POIs) in einer Augmented Reality Ansicht dar. [TO09]

2.1.2 IndoorGuide4Android II

Im Rahmen dieser Bachelor-Thesis wurde eine *AR*¹-Applikation für Android Smartphones erstellt, welche auch innerhalb eines Gebäudes verwendet werden kann. Anhand von Signalen, die von Wireless Access Points empfangen werden, kann der aktuelle Standort berechnet werden. Diese Technologie wird durch den IndoorWPS Community Server - einem weiteren Projekt der Forschungsgruppe - zur Verfügung gestellt. Dabei werden zum Schutz der Privatsphäre zu keinem Zeitpunkt Informationen wie die eigene Position dem Server mitgeteilt. Der IndoorGuide4Android kann dadurch *POI*² in einem Gebäude (z.B. Museum, Campus oder Einkaufszentrum) anzeigen. [CE10]

¹ *Augmented Reality*

² *Point of Interest*

2.2 Konkurrenzprodukte

Bei der Recherche für diese Arbeit ist aufgefallen, dass es vor allem Produkte gibt, welche entweder spezifisch für Android Geräte geschrieben sind, oder nur auf *iOS*¹ lauffähig sind.

2.2.1 Swisscom Openair Buddy

Der „Swisscom Buddy“ ist eine iPhone-Applikation, welche von Swisscom zur Openair Saison 2010 lanciert wurde. Die Applikation deckte die sechs grösseren Openairs der Schweiz ab. Dies sind: Festi'neuch Neuenburg, OpenAir St.Gallen, Estival Jazz Lugano, Open Air Frauenfeld, Gurtenfestival BernPaléo Festival Nyon. Die Applikation hat unter anderem die folgenden Funktionen²:

- **Festivalübersicht** Informationen zu den 6 Festivals, Karte, Spielzeiten mit Hintergrundinformationen zu den Bands, Essens- und Getränkestände mit Preisen
- **Concert Alert** Benachrichtigung wenn bevor die Lieblingsband zu spielen beginnt.
- **Zeltfinder** Das eigene(!) Zelt wiederfinden
- **Wetter** Aktuelle Wetterdaten zum Event
- **Chat & Mood** Chat via Facebook. Flirtfunktion mit allen anderen anwesenden Usern.

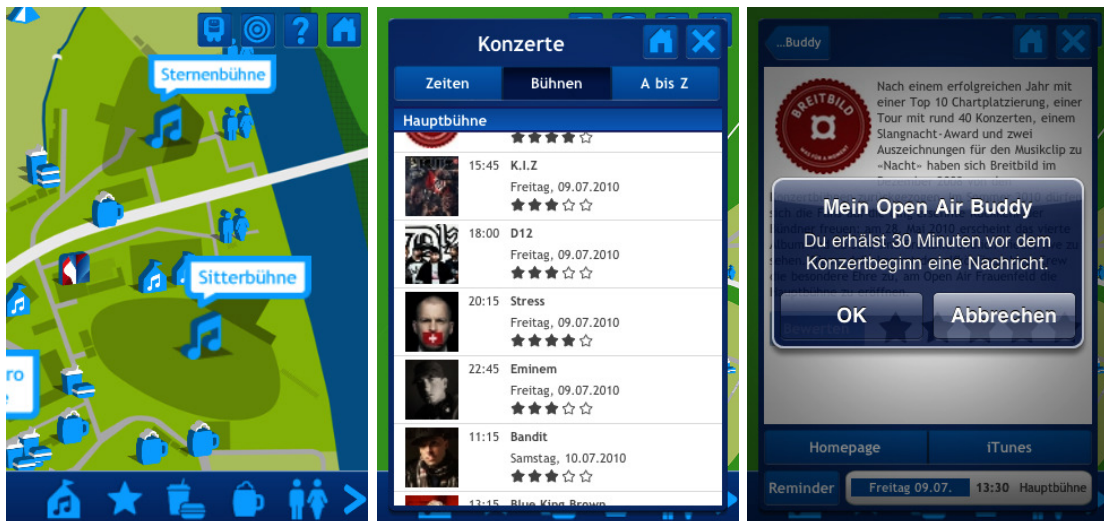


Abbildung (2.1) Screenshots Swisscom Openair Buddy

¹ *iPhone OS*

² <http://labs.swisscom.ch/en/content/swisscom-open-air-buddy>

2.2.2 Festival Mob

FestivalMob ist eine Software Plattform für Musik und Kulturveranstaltungen. Es können pro Event gebrandete Inhalte, Informationen zu Bands, Karten erfasst werden. Anbindung an Social Web ist ebenfalls vorhanden.

Der interessierte Veranstalter kauft bei FestivalMob seine angepasste Version des Clienten für iPhone und Android und kann diese dann über die jeweiligen Markets verteilen. Die Inhalte für die Devices (Karten, Spielpläne, etc.) werden auf einer Webseiten vom Veranstalter erfasst und auf die Geräte übertragen. FestivalMob stellt also dem Veranstalter eine Plattform für Mobiles bereit, welche die Funktionen des „FestivalGuides“ abdecken.

Dies sind:

1. **Festival Lineup**
2. **Personal Festival Schedule**
3. **Tent Refind**
4. **Augmented Reality** (Layar Browser)
5. **Festival News**

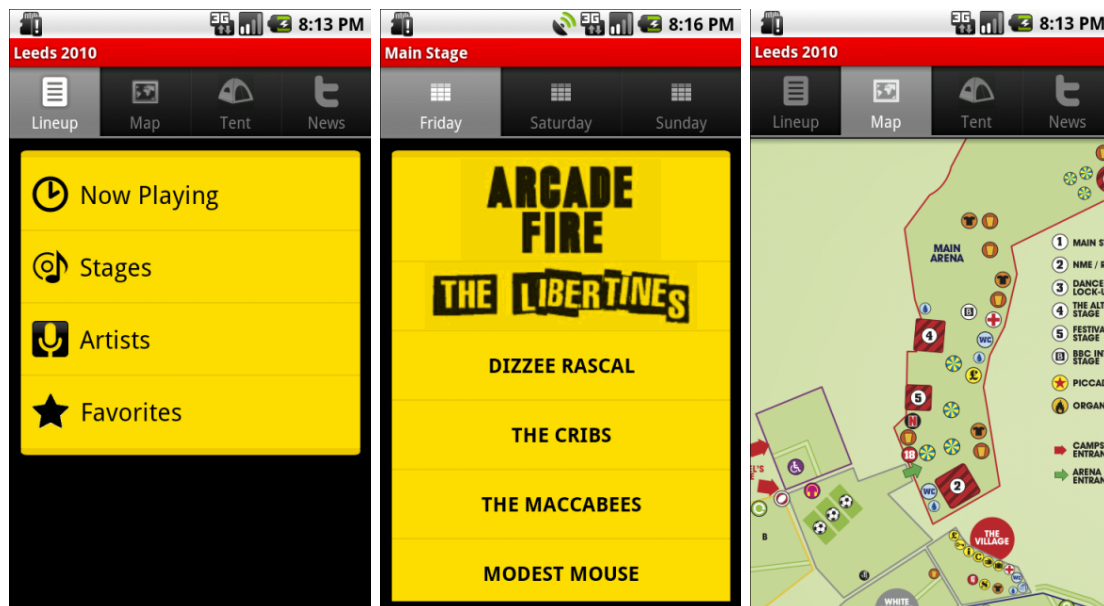


Abbildung (2.2) Screenshots FestivalMob

2.2.3 Spezifische Mobile-Apps von Veranstaltern

Es gibt diverse Veranstalter, welche für Ihre Festivals Applikationen (mehrheitlich für iPhone iOS) entwickelt haben. Je nach Grösse der Veranstaltung sind diese auch umfangreich. Beispiele

1. Wacken (iPhone App)
2. Soundlike Open Air Festival (iPhone App) ¹
3. Openair Laax (Android App) ²



Abbildung (2.3) Screenshots Festivalapps kleiner und grosser Festivals

2.2.4 Webseite der Veranstalter

Oftmals eignen sich die Webseiten der Veranstalter ideal zum auffinden von Informationen. Jedoch schränken zwei Fakten diese Möglichkeit für Mobilfunkteilnehmer wieder ein:

- Verbindungsprobleme
- Auftritt nicht für kleine Bildschirme geeignet oder nicht unterstütztes Format (Flash oder ähnliches)
- Aktuelle Informationen schlecht zugänglich oder nicht vorhanden, bzw. aktuell.

¹ <http://www.soundlike.at/info/iphone-app-festivalpilot/>

² <http://openairlax.ch/android>

- Geringer Mehrwert wenn nur Spielzeiten abgerufen werden können.

Unter Berücksichtigung dieser Fakten erscheint die Nutzung der Veranstaltungswebseiten plötzlich nicht mehr so attraktiv.

2.2.5 Facebook-Places

Seit August 2010 ist Facebook Places verfügbar. Die Zusatzfunktion erlaubt es seinen Benutzern unterwegs an Orten „einzuchecken“ und so anderen Benutzern mitzuteilen, wo man sich gerade befindet. Ebenso sollen sich in Zukunft interessante Orte in der Nähe direkt mit Facebook finden lassen - Oder auch wo sich die Freunde gerade befinden. Die Freunde werden jedoch nicht auf einer Karte dargestellt, sonder nur der „Hotspot“ an welchem sie sich gerade befinden. Die Places-Erweiterung funktioniert aktuell in den USA, Deutschland und der Schweiz. [Fac10]

3 Evaluation

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	12.10.2010	Michael Schnyder	Vorlage und Technologieentscheid für Mobiles erstellt
0.2	16.11.2010	Michael Schnyder	Entscheid für Map-Control eingefügt
0.3	17.11.2010	Felix Egli	Entscheide Client-Server Kommunikation / Server-Architektur / Client-Architektur
0.4	19.12.2010	Felix Egli	Entscheide Client-Server Kommunikation / Server-Architektur / Client-Architektur überarbeitet und abgeschlossen
0.5	23.10.2010	Michael Schnyder	Client-Architektur überarbeitet
0.6	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

3.1 Technologieentscheid Mobile Plattform

Zur Diskussion Standen die zwei Möglichkeiten Android und *HTML5*. *iOS* wurde aufgrund zu geringer Kenntnisse des Projektteams von Beginn her ausgeschlossen.

Die ermittelten Kriterien zum Technologieentscheid wurden anhand dem Superset der Szenarien erstellt und untereinander entsprechend gewichtet.

Weiter wurde der Verbreitungsgrad der Plattformen nur minimal berücksichtigt. Da sich dieser Indikator aktuell sehr stark verändert.

3.1.1 Bewertungskriterien

Folgende Bewertungskriterien wurden im Projektteam erarbeitet und gewichtet.

3.1.1.1 Aufwandschätzungen (je geringer der Aufwand, desto höher bewertet)

- Funktionieren ohne Mobil/Datennetz (10%): App auch ohne Datenverbindung bedienbar.
- Karte anzeigen (10%)
- Karte nach norden ausrichten (10%): Benötigt einen Kompass
- Frameworks, GUI-Hilfen, Community (10%)
- Lokalisierung mittels API und oder GPS (18.5%)
- Strukturierter lokaler Datenspeicher (7%)

- Reminder Funktionen (15%)
- Facebook Integration (8%)
- Push Notifications (8%)

3.1.1.2 Potential

- Verbreitungsgrad auf Mobiles (2%)

3.1.2 Bewertung

Jedes Kriterium wurde bei der Bewertung zwischen 1 (minimal) und 10 (maximal) bewertet. Je höher der Wert, desto besser ist die Plattform in der Disziplin oder je geringer ist der Implementierungsaufwand für die geforderte Funktion.

Technologieentscheid Tabelle

Einzeldisziplinen	Gewichtung	Android		HTML5		Bemerkungen
	in Prozenten	abs	gew	abs.	gew.	
Aufwandsschätzungen (höher -> besser)						
- Funktionieren ohne Mobil/Datennetz	10.00%	10	1	5	0.5	Karte nicht einfach speicherbar
- Karte anzeigen	18.50%	8	1.48	8	1.48	
- Karte nach norden ausrichten	3.00%	8	0.24	1	0.03	
- Frameworks, GUI-Hilfen, Community	10.00%	8	0.8	4	0.4	
- Lokalisierung mittels API und oder GPS	18.50%	8	1.48	6	1.11	
- Strukturierter lokaler Datenspeicher	7.00%	9	0.63	5	0.35	
- Reminder Funktionen	15.00%	10	1.5	3	0.45	
- Facebook Integration	8.00%	8	0.64	8	0.64	
- Push Notifications	8.00%	7	0.56	2	0.16	HTML: Nicht möglich wenn App geschlossen
Potential						
- Verbreitungsgrad auf Mobiles	2.00%	4	0.08	10	0.2	
Summe	100.00%	80	8.41	52	5.32	

Tabelle (3.1) Technologieentscheid Tabelle

3.1.2.1 Funktionieren ohne Mobil/Datennetz

Die Android-Plattform kann hier besser Punkten, da eine einmalig installierte Applikation zum Starten danach keine Internetverbindung mehr benötigt. Für HTML5 existiert jedoch eine Möglichkeit, den Browser anzuweisen, welche Dateien gecached¹² werden sollen und jedoch erscheinen dem Projektteam die Möglichkeiten in *HTML5* als zu wenig flexibel bzw. zu komplex zu konfigurieren und zu fehleranfällig. Ausserdem können auf diese Weise nur

1 <http://www.webreference.com/authoring/languages/html/HTML5-Application-Caching/>

2 <http://www.w3.org/TR/html5/offline.html>

Dateien gecached werden, welche sich auf dem gleichen Server befinden wie die Applikation selbst.

3.1.2.2 Karte anzeigen

Unentschieden. *HTML5* ist bestimmt besser geeignet um Bilder anzuzeigen, jedoch kann hier Android beim Event-Handling Punkten.

3.1.2.3 Karte nach Norden ausrichten

Hierfür wird der Kompass des Gerätes benötigt. Über *HTML5* existiert hier keine Schnittstelle, bzw. wird nur proprietär von einigen (Desktop!) Browsern unterstützt. Über Android kann nativ auf Accelerator und Kompass zugegriffen werden.

3.1.2.4 Frameworks, GUI-Hilfen, Community

Hierzu reicht eine einzige Abfrage in Google mit den Suchworten „Android Developer“ (ca. 60 Mio. Ergebnisse) und „HTML5 Developer“ (ca. 2 Mio. Ergebnisse) um diesen Punkt zu vergeben.

3.1.2.5 Lokalisierung mittels API und GPS

Hier liegen beide Plattformen etwa gleichauf. Android wie auch *HTML5* können den Benutzer unter Verwendung von Triangulation und/oder GPS orten. Einzig der Zugriff auf die WLAN-Verbindungsdaten ist via *HTML5* nicht möglich, welcher evtl. bei einem Indoor-Einsatz interessant wäre. Zur GPS-Nutzung von *HTML5* ¹

3.1.2.6 Strukturierter Lokaler Datenspeicher

HTML5 wie auch Android bieten Zugriff auf eine SQLite² Datenbank, mit dem Unterschied, dass bei Android dieser Zugang schon lange final abgesegnet ist. Bei *HTML5* hat der lokale Datenspeicher den Status „Editors Draft“ d.h. noch lange nicht final³. Zudem wird aktuell vorgeschlagen den Speicher auf 5MB zu beschränken, was für eine Applikation mit Bildern (Karte) doch sehr wenig ist⁴.

1 <http://dev.w3.org/geo/api/spec-source.html#high-accuracy>

2 <http://de.wikipedia.org/wiki/Sqlite>

3 <http://www.access-board.gov/sec508/refresh/teitac1st/W3C-WAI-20060928/Slides/slide12-0.htm>

4 <http://dev.w3.org/html5/webdatabase/#disk-space>

3.1.2.7 Reminder Funktionen

Es wurde vom Projektteam keine Möglichkeit gefunden, zeitgesteuert Nachrichten auf einem Mobile anzuzeigen, welche von einer *HTML5*-Applikation gesteuert wurden. Die *HTML5* Applikation könnte sich möglicherweise gar nicht mehr im Speicher befinden, bzw. ausgeführt werden. (Beispiel¹). W3C.org Editors draft from 13 October 2010². Unter Android existieren hier neben Push-Notifications (Android ≥ 2.2) auch die Möglichkeit, einen Background-Service zu erstellen, welcher Notifications an den User sendet. Diese werden dann in der Notification-Area angezeigt.

3.1.2.8 Facebook Integration

Facebook lässt sich via *HTML5* (Javascript) oder auch Andoird (Library direkt von Facebook) einbinden. Die potentiell benötigten Funktionen stehen in beiden Libraries zur Verfügung.

3.1.2.9 Push Notifications

Keine Möglichkeit in *HTML5*. In Android entweder über eine eigene Implementation (mittels Background-Service) oder neu ab Android ≥ 2.2 über den Push-Service von Google.

3.1.2.10 Verbreitungsgrad auf Mobiles

Gemäss diversen Umfragen sind hier *HTML5* fähige Devices (d.h. hauptsächlich *iOS* und neue Android Geräte) weit verbreitet. Jedoch werden die Verkäufe in den USA bereits jetzt von Android angeführt. In Europa wird Android wohl auch sehr bald Absatzmässig das iPhone überholen.³

3.2 Map-Control

Die Darstellung einer Karte stellt einen essentiellen Teil unserer Applikation dar. Wichtig zu wissen ist, dass wir nicht beabsichtigen die einzelnen Orte eines Anlasses (Bühne, Bars, etc.) aktiv auf die Karte zu zeichnen (die gestalterische Freiheit möchten wir komplett dem Designer/Veranstalter übergeben). Vielmehr möchten wir auf Taps (=Klicks) reagieren können, um Details zu dem gewählten Ort anzeigen zu können.

Allgemein gesehen soll die Karte die folgenden Funktionen abdecken:

1 http://webkrunk.com/demos/html5Notifications/html5_notification_test.html

2 <http://dev.w3.org/html5/eventsource/>

3 <http://assets.appcelerator.com.s3.amazonaws.com/docs/Appcelerator-IDC-Q4-Mobile-Developer-Survey-2010.pdf>

- **Eigene Karte** Es soll ein Bild als Karte verwendet werden. Damit ist die Auflösung genauer als bei Google-Maps¹ und der Designer/Veranstalter hat mehr Möglichkeiten das *CI*² des Anlasses in die Mobil-Applikation einzubringen.
- **Position(en) anzeigen** Es ist wichtig, dass der Benutzer sich auf der Karte darstellen lassen kann. Als Orientierungshilfe soll sich entweder die Karte drehen (d.h. immer nach Norden ausrichten) oder die Position des Benutzers soll so dargestellt werden, damit klar ist, in welche Richtung dieser schaut.
- **Interaktivität** Durch Tappen auf sogenannte aktive Regionen in der Karte (beispielsweise die Bühne, Verkaufsstände, etc.) kann der Benutzer zusätzliche Informationen zu dem Element auf der Karte abrufen.
- **Darstellungsoptionen** Der Benutzer soll die Möglichkeit haben, im Minimum die Karte so zu verschieben, dass das ganze Gelände sichtbar ist. Optional soll ihn dabei eine Zoom-Funktion unterstützen.

Um diese Anforderungen zu erfüllen, wurden die folgenden Controls/Möglichkeiten getestet. Wir haben uns auf Basis der unten ersichtlichen Ausführungen für die Verwendung der WebView als Map-Control entschieden.

3.2.1 Variante „GoogleMapsAPI“

Mit Google-Maps lassen sich sehr schnell Applikationen entwickeln, welche auf Karten von Google-Maps aufbauen. Es lassen sich auch zusätzliche Pins auf die Karte „stecken“, jedoch sind dann die Möglichkeiten zur Darstellung auf der Karte bereits erschöpft. Eine eigene Karte zu verwenden ist direkt nicht möglich. Dies funktioniert nur über Umwege (direktes Zeichnen mit dem Canvas Object). Diese Variante ist jedoch weder besonders schön noch performant. Ebenso lässt sich der Bereich, in welchem der Benutzer die Karte darstellen kann, nicht begrenzen. Dies wäre vor allem aus Usability-Gründen sinnvoll. Als weiteren Nachteil kann die zwingend notwendige Verwendung eines API-Keys gesehen werden. Dieser ist zwar gratis, jedoch in der Nutzung nicht unlimitiert (Genaue Zahlen fehlen seitens Google).

3.2.2 Variante „WebView (WebKit)“

Das Browser-Control lässt sich (über Umwege) dazu bewegen ein Bild anzuzeigen, welches nicht direkt aus dem Internet sondern vom Gerät selbst stammt. Zoom und Scroll sind im WebView-Control bereits integriert und müssen nur dazugeschaltet werden. Ebenso lässt sich der Minimale und Maximale-Zoom-Level festlegen. Auf die Karte lässt sich zeichnen

¹ <http://maps.google.ch/>

² *Corporate Identity*

(automatisch jedes Mal wenn das Bild neu geladen werden soll, oder manuell) und es kann auf die Taps (=Klicks) des Benutzers reagiert werden. Einziger Nachteil: Die Umrechnung von Koordinaten und Pixel und wieder zurück muss ausprogrammiert werden. Ebenso die Erkennung, ob der Benutzer innerhalb einer „Aktiven Fläche“ (Bühne, Bar, etc.) getappt hat, oder nicht.

3.2.3 Variante „ScrollView“

Die Scrollviews in Android können entweder einen zu grossen Inhalt vertikal oder horizontal scrollen - Nicht aber beides gleichzeitig. Durch diesen Umstand war schnell klar, dass unsere Anforderungen nicht erfüllt werden konnten. Es gibt jedoch eine Library, welche eine solche DuplexScrollView zu Verfügung stellen möchte. Diese ist jedoch fehlerbehaftet und funktioniert aktuell (16.11.2010) nicht unter Android 2.2 ¹.

3.2.4 Variante „Eigenes Control“

Diese Variante kombiniert alle Vorteile einer sauberen eigenständigen Lösungen mit den Nachteilen wie z.B. Komplexität, Aufwand, Speicherverbrauch, etc. Wir haben uns (zumindest im aktuellen Ausbaustadium) dagegen entschieden, ein Control für diesen Zweck selber zu entwickeln.

3.3 Client-Server Kommunikation

Das Ziele:

- Kommunikation so klein und simpel wie möglich halten
- kein eigens Protokoll implementieren
- in der ersten Phase Daten vom Server holen
- in der zweiten Phase Daten an den Server senden
- Client muss auch ohne Internet funktionieren

Zur Client-Server Kommunikation wurden folgende Varianten evaluiert:

3.3.1 Sockets

Dies wurde abgelehnt, da die Implementation der Kommunikation sehr aufwändig wäre und es für unsere Anforderungen einen ziemlich grossen Mehraufwand bedeuten würde.

¹ <http://code.google.com/p/android-pinch/>

3.3.2 XML

Die Kommunikation über XML ist sehr praktisch, da sich die Daten strukturiert austauschen lassen. Dennoch ist hier der Overhead der Notation ziemlich gross. Auch das Parsen von XML auf Android ist nicht ganz so einfach. Benötigt viel Code um das Dokument zu parsen.

3.3.3 JSON

Die Kommunikation über JSON ist optimal, da die Daten strukturiert sind aber keinen Notationsoverhead haben. Der Nachteil ist jedoch die Notation lässt sich nicht automatisch validieren. Das Parsen von JSON kann ziemlich kurz und knackig gehalten werden.

3.3.4 Entscheid

Die Entscheidung war sehr einfach zu treffen für uns da das Parsen sehr viel einfacher ist und der Overhead sehr klein von JSON, haben wir uns für JSON entschieden.¹

3.4 Server-Architektur

Ziele:

- Server muss schlank sein (d.h. Nicht sehr viel Zeit investiert werden, um den Serverteil zu implementieren, da dies nur ein Nebenprodukt in dieser Arbeit ist.)
- Der Servercode muss auf unterschiedlichen Plattformen laufen
- Server muss Bilder liefern können
- Server muss JSON Files (serialisierte Klassen) liefern

Zur Server-Architektur wurden folgende Varianten evaluiert:

3.4.1 Apache-Server mit PHP-Modul

Ein Apache-Server mit PHP würde den Anforderungen genüge tun. Nur müsste das URL-Rewriting aktiviert sein, damit mit schönen URL's gearbeitet werden kann.

¹ <http://ubikapps.net/?p=268>

3.4.2 Tomcat mit Servlet

Diese Technologie kann alle Anforderungen einhalten. Hier ist besonders einfach ein Servlet, dass ein doPost oder doGet abfängt und selbständig die URL parst und verarbeitet.

3.4.3 ASP.NET-Server

Diese Technologie kann alle Anforderungen einhalten. Ist jedoch zu umfangreich und würde eine neue IDE, Sprache usw. mit sich bringen.

3.4.4 Entscheid

Wir haben uns aus Kompatibilitäts-Gründen für die Variante „Tomcat mit Servlet“ entschieden. Denn die Requests können einfach abgefangen werden. Zudem kann das Server-Java-Projekt direkt in den selben Eclipse-Workspace integriert werden wie das Andorid-Projekt.

3.5 Client-Architektur

Die Client Architektur beinhaltet eine ganze Menge wichtiger Aspekte welche beachtet werden müssen.

- Parsing von Antworten auf dem Netzwerklayer für höherer Layer.
- Zentraler Punkt um Daten abzurufen
- Cache muss implementiert sein

3.5.1 Parsing

Dies war in Bezug auf die Ressourcenschonung wichtig. Die Daten werden vom Server als Json-Formatierte Strings zur Verfügung gestellt und müssen den oben liegenden Layern als Datenklassen vermittelt werden. Mit dem *MessageArchitecture* (siehe Seite 76) wurde hierzu ein Konzept entwickelt, bei welchem die alle Daten generisch in Datenklassen verpackt werden können. Ohne das auf dem Client für jede Entity eine separate Klasse erstellt werden musste.

3.5.2 Zentraler Punkt

Hier floss die Recherche im Zusammenhang mit ContentProvidern ein. Der Zentrale Punkt wurde durch die Implementierung eines eigenen Content Providers geschaffen, welcher nebst dem UI theoretisch auch in anderen Android-Applikationen verwendet werden

kann. Dieses Konzept wurde für Bilder (siehe *BrowserMapPrototype* auf Seite 75) und für Daten *MessageArchitecture* (siehe Seite 76) , *ContentValues* und *Cursors* (siehe Seite 79) mit je einem Prototypen getestet. Die hieraus gewonnen Erkenntnisse sind auch in die Beschreibung des Content Providers geflossen (siehe *Schnittstelle zu ContentProvider* auf Seite 63).

3.5.3 Cache

Hierfür wurde die Verwendung der SQLite als Cache für die Daten und der Lokale Datenspeicher für die Speicherung von Bildern getestet.

4 Umsetzungskonzept

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	18.12.2010	Michael Schnyder	Dokument und Umesetzungskonzept erstellt
0.2	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

4.1 Allgemein

Die Arbeit am Projektes beschränkte sich nicht auf die Erstellung eines lauffähigen Prototypen für die Android-Plattform. Vielmehr wurden nebst dem Android-Client Protokolle und Datenstrukturen definiert. Ebenso war es dem Projektteam wichtig, dass die Erfassung von Events mit möglichst wenig administrativem Aufwand möglich ist. Das führte dazu, dass für die Karte eine normale JPEG- oder PNG-Grafik verwendet werden kann, für welche Geo-Referenzierungsdaten bestimmt sind.

4.2 Android

Es wurden wenn immer möglich gängige Android-Konzepte wiederverwendet. Als Informationsquelle diente <http://developer.android.com>. Es war wichtig die Nichtfunktionalen Anforderungen betreffend Offline-Nutzung und Akkuschonung zu erfüllen.

4.3 Daten und Protokolle

An verschiedenen Stellen mussten Datenformate und auch Protokolle vom Projektteam spezifiziert werden. Dies sind insbesondere:

- **Kommunikation** Kommunikationsprotokoll mit EventAssistant-Server
- **Datenformat** Das damit verbundene Datenformat
- **Lokales Datenbankschema** Für den Offline-Modus

Siehe hierzu auch die spezifischen Kapitel (siehe *Schnittstellen und Formate* auf Seite 63) und (siehe *Datenbankschema* auf Seite 63)

4.4 Organisation

Mit der Definition der Datenformate auf Client und Server, wurde auch gleich definiert wie die Daten beschafft werden können. (siehe *Content Erfassung* auf Seite 122)

5 Resultate

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	14.11.2010	Michael Schnyder	Dokument und Struktur erstellt
0.2	18.11.2010	Michael Schnyder	Resultate erfasst
0.3	21.12.2010	Michael Schnyder	Nichtfunktionale Anforderungen Resultate, Lessons Learned
0.4	22.12.2010	Felix Egli	Persönlicher Bericht angepasst
0.5	23.10.2010	Michael Schnyder	Persönlicher Bericht
0.6	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

5.1 Zielerreichung

5.1.1 Funktionale Anforderungen

Auf Basis der ausgearbeiteten Szenarios wurden die gesetzten Ziele erreicht. (siehe *Szenarios* auf Seite 44). Durch die Einteilung dieser und den daraus resultierenden Use Cases in Must, Optional und Abgrenzung können die implementierten Funktionen wie folgt bewertet werden.

5.1.1.1 Erfüllte Use Cases

Die folgenden Use Cases wurden gemäss der Anforderungsspezifikation als Must-Use Cases kategorisiert und konnten wie nachfolgend ersichtlich umgesetzt werden:

Use Case	Erfüllungsgrad
Festivalablauf ansehen	erfüllt
Karte ansehen	erfüllt
Eigene Position bestimmen	erfüllt
Details zu POI anzeigen	erfüllt
Einstellungen verwalten	erfüllt
Erkennung des Events (keine Einteilung)	erfüllt
Offline/Online Information (keine Einteilung)	erfüllt

Tabelle (5.1) Zielerreichung Must-Use Cases

Wie in der Tabelle ersichtlich, konnten alle geforderten Use Cases implementiert werden. Zusätzlich wurden folgende Funktionalitäten umgesetzt, welche in der Anforderungsspezifikation nicht eindeutig definiert wurden:

- **Automatische Bestimmung des Events** Beim Starten der Applikation wird automatisch und falls möglich der Event geladen, an welchem sich der Benutzer befindet. Das Auswählen aus der Liste der vorhandenen Events entfällt hierbei. Steht aber trotzdem bei Bedarf zu Verfügung.
- **Offline/Online Information** Der Benutzer erhält ein Feedback über die Aktualität der angezeigten Daten. Bei einem Netunterbruch sind diese ggf. nicht mehr aktuell.

5.1.1.2 Optionale Use Cases

Die folgenden Use Cases wurden gemäss der Anforderungsspezifikation als Optionale-UseCases definiert und konnten wie nachfolgend ersichtlich umgesetzt werden:

Use Case	Erfüllungsgrad
An MiniEvent erinnern lassen	teilweise umgesetzt
Freunde finden	nicht umgesetzt
Pushnachricht senden	nicht umgesetzt

Tabelle (5.2) Zielerreichung Optionale-UseCases

Anmerkung: Der Use Case „An MiniEvent erinnern lassen“ wurde auf UI-Seite komplett umgesetzt. Jedoch stellte sich die Umesetzung mit Notifikationen und einem Background-Service als komplexer heraus als ursprünglich angenommen. Die bereits erstellte UI-Sicht wurde daher wieder verworfen.

5.1.1.3 Abgegrenzte Use Cases

Es wurden keine der abgegrenzten Use Cases umgesetzt

5.1.2 Nichtfunktionale Anforderungen

Die Nichtfunktionalen Anforderungen sind unter der SW-Projektdokumentation (siehe *Nichtfunktionale Anforderungen* auf Seite 50) erfasst und sind an dieser Stelle nicht genauer beschrieben.

Nachfolgend einige Erläuterungen zu den oben genannten erreichten Resultaten.

5.1.2.1 Zentrale Speicherung

Die Referenzimplementation des EventAssistant-Servers besitzt eine Architektur, mit welcher sehr einfach reale Datenquellen, (z.b. eine relationale Datenbank) angebunden werden können. Dies wurde über eine Schnittstelle als Interface implementiert.

Anforderung	Erfüllungsgrad
1) Zentrale Speicherung	Umgesetzt mit Interface für Datenbankanbindung
2) Geo-Referenzierung	Umgesetzt
3) Client Cache	Umgesetzt mit Verfallsdatum
4) Performance	Umgesetzt und mit System-Tests getestet
5) UI-Responsiveness	Umgesetzt und mit Usability-Tests getestet
6) Akkuverbrauch / Systemdesign	Umgesetzt (siehe Erläuterungen)

Tabelle (5.3) Zielerreichung Nichtfunktionale Anforderungen

5.1.2.2 Geo-Referenzierung

Umgesetzt mittels zwei Referenzpunkten auf einer beliebigen Grafik. Die Ermittlung dieser Daten ist als Prozess beschrieben (siehe *Content Erfassung* auf Seite 122).

5.1.2.3 Performance

Wurde verschiedentlich von den Entwicklern bei unterschiedlichen Entwicklungsständen getestet. Ist auch als Systemtest (siehe *Systemtest Protokoll vom 19.12.2010* auf Seite 189) vorhanden.

5.1.2.4 UI-Responsiveness

Konnte zum einen von den Entwicklern selber, als auch während Usability-Tests geprüft werden. Bei länger dauernden Aktionen (z.B. Nachladen von Daten über das Internet) wird jeweils ein Progress-Dialog angezeigt. Je nach Situation kann das Laden von Daten auch abgebrochen werden.

5.1.2.5 Akkuverbrauch / Systemdesign

Dies war sicher ein Knackpunkt während der Entwicklung. Alle unnötigen Threads werden nach Gebrauch wieder sauber beendet und auch der GPS-Provider wird beim verlassen der Applikation deaktiviert (Akkuverbrauch). Nachfolgend ein Screenshot über die laufenden Prozesse und Threads nach 10 maligem Starten unserer Applikation. Hauptaugenmerk gilt hier den Threads, welche sich im Status „timed-wait“ befinden. Diese Threads sind möglicherweise sehr aktiv (siehe Spalte „utime“ und „stime“) und können Akkulaufzeit und Geschwindigkeit negativ beeinflussen.

ID	Tid	Status	utime	stime	Name
1	4695	wait	547	83	main
*2	4696	vmwait	25	5	HeapWorker
*3	4697	vmwait	0	0	Signal Catcher
*4	4698	running	6	9	JDWP
*5	4699	vmwait	110	15	Compiler
6	4700	native	0	0	Binder Thread #1
7	4701	native	0	0	Binder Thread #2
8	4723	wait	0	18	AsyncTask #4
9	4704	native	0	0	Binder Thread #3
10	4705	wait	0	42	AsyncTask #1
11	4706	wait	0	22	AsyncTask #2
12	4707	wait	0	35	AsyncTask #3
13	4783	wait	0	38	AsyncTask #5
15	4791	timed-wait	16	2	WebViewWorkerThread
16	4787	wait	9	5	WebViewCoreThread
17	4788	timed-wait	0	0	CookieSyncManager
18	5057	timed-wait	0	1	Thread-38
19	4923	native	0	0	Binder Thread #4

Abbildung (5.1) Screenshot Laufende Threads

Threadname	Zweck	Auslöser
Threads 1-5	System	Systembedingt
Binder Thread #1-4	System	Systembedingt
AsynTask #1-5	Background Taskqueue (von Android verwaltet, max. 5)	EA
WebViewWorkerThread	Teil der MapView, bzw. WebView	WebView
WebViewCoreThread	Teil der MapView, bzw. WebView	WebView
CookieSyncManager	Teil der MapView, bzw. WebView	WebView
Thread-38	Periodische Überprüfung Internetverbindung	EA

Tabelle (5.4) Laufende Threads Erläuterung

Anmerkungen: Die BinderThreads werden von Android für die Applikation erstellt, sobald Layouts als XML in Views gerendert werden. Diese Threads werden dann auch wieder von Android abgeräumt. AsyncTasks befinden sich ebenfalls in einem Pool, welcher von Android verwaltet wird. Es sind nie mehr als fünf AsyncTasks als Threads am laufen. Im ersten Augenblick erscheint hier der Thread „Thread-38“ als „Zombie“-Thread. Dies ist jedoch der Thread, welcher für das Ein- und Ausschalten des Offline-Modus-Indikators zuständig ist. Wird die Applikation geschlossen, werden dieser Thread und auch die „Web*-Threads beendet.

5.1.3 Fazit

Mit der Arbeit wurde somit ein System geschaffen, welches sich in der aktuellen Entwicklungsstufe die Grundbedürfnisse eines Benutzers abdeckt. Darüber hinaus definiert die Arbeit das Format und die Daten, welche für die Verwendung der Mobil-Applikation an einem beliebigen Event ermöglicht.

5.2 Ausblick

Befindet sich in einem separatem Kapitel, welches auch die Ausbaumöglichkeiten darstellt. (siehe *Möglichkeiten der Weiterentwicklung* auf Seite 105) Auch weitere Arbeiten stehen an, die nach dem Final Usability Test (siehe *Usability-End Test* auf Seite 101) angegangen werden müssen.

5.3 Persönliche Berichte

5.3.1 Felix Egli

5.3.1.1 Projektverlauf

Als wir unser Projekt eingegeben haben, wussten wir selbst nicht so genau was uns da erwarten wird. Wir haben im Voraus mit der Swisscom ein Meeting abgehalten, um ihre mögliche Anforderungen an ein solches Projekt zu erfahren. Eines der Hauptprobleme der Swisscom war, dass sie für jedes OpenAir eine neue Version ihrer mobilen Applikation erzeugen und von Apple verifizieren lassen, damit sie auf dem Appstore verfügbar ist. Zudem ist die Contententerfassung der verschiedenen OpenAirs sehr aufwändig. Ihre Meinung war auch, dass es kaum möglich ist eine Applikation zu entwerfen, welche für mehrere Events gleich aussehen würde.

Dies war unsere Herausforderung. Eine Applikation zu erschaffen, welche für mehrere Events verwendbar ist und ihren Inhalt selber aktualisieren kann. Also starteten wir voller Elan in unsere Studienarbeit.

Die Features die wir implementieren wollten, waren nicht einfach aus der Luft gegriffen. Denn diese stützen wir auf Interviews die wir mit Besuchern an Grossanlässen durchgeführt haben. Dies war eine grosse Herausforderung, denn ich werde auch nicht gerne an solchen Anlässen angesprochen. Wir bereiteten uns gut vor auf die folgenden Interviews. Unser Fragenkatalog war viel zu lang und wir wussten nicht die Befragten darauf reagieren würden. Ich war erstaunt wie die Befragten einwilligten als wir erzählten, dass wir Studenten sind und für unsere Studienarbeit recherchierten. Dennoch mussten wir nach der Erstbefragung unser Fragebogen mächtig kürzen. Nach den zwei Interviewtagen am Freestyle.ch und an der Züsipa fühlten wir uns beflügelt die richtigen Features für unsere Applikation zu entwickeln.

Wir legten die Features anhand der Antworten des Interviews fest. Wir erstellten einen Papierprototypen um auch die Bedienung unseres zukünftigen *GUI*¹, benutzerfreundlich zu gestalten. Schon bald ging es um die technische Umsetzung, wir erstellten etliche Prototypen auf Android. Denn hier fehlte es uns mächtig an Erfahrung, denn die wenigen Mikrobeispiele die wir bis anhin auf Android erstellt haben waren eher bedürftig. Dies war eine sehr grosse Herausforderung für uns. Eine Cachelogik, Remotezugriff auf einen entfernten Server, Datenhaltung und *GUI*-Komponenten zu implementieren und das alles auf einem System, dass wir bis dahin noch nicht gekannt haben. Die Herausforderung war enorm. Glücklicherweise haben wir die Features unserer Applikation als (Must / Optional / Abgrenzung) kategorisiert. Denn wir merkten schnell, dass wir nur die Must-Features implementieren konnten. Auch dies gelang uns Schlussendlich nur mit Überstunden zu leisten. Was uns auch gelang. Zum Schluss war die grosse Dokumentation angesagt. Denn

1 *Graphical User Interface*

durch das viele intensive Programmieren rutsche die Dokumentation etwas ins Hintertreffen. Dazu kam noch, dass wir erstmals mit \LaTeX gearbeitet haben und dies einen zusätzlichen Aufwand bedeutete.

5.3.1.2 Rückblick

Rückblickend muss ich sagen, dass die Interviews sehr Wertvoll waren und bin froh, dass wir diese auch durchgeführt haben. Leider haben wir uns übernommen mit der Abklärung der Features und der „*Android-Referenz*“ implementation. Auch die Dokumentation musste Anhand der Kriterienliste sehr genau ausgearbeitet werden. Darum waren die letzten 14 Wochen eine sehr intensive Zeit. Vorallem die letzte Woche. =)

5.3.1.3 Gelerntes

An meisten habe ich bestimmt mit der Projektplanung gelernt, wenn mit einer meist unbekannten Technologie gearbeitet wird. Auch das die Umsetzung einer Mobilapplikation noch einiges mehr Tücken mit sich bringt (Unberechenbares Betriebssystem, keine genaue GPS-Daten, aufwändiges Testing usw.). Dies macht die Planung noch schwieriger. Dafür konnte ich sehr viel über Android lernen. Von allgemeinen Konzepten bis hin zur GUI-Konzepten. Alles war dabei. Die Dokumentation ohne einen WYSIWYG-Editor zu erfassen war nicht leicht, aber auch hier konnten wir einige Lehren daraus ziehen. Auch das Meeting bei der Swisscom war für die Anforderungserfassung sehr lehrreich. Auch in zwischenmenschlichen Beziehungen habe ich bestimmt einiges dazugelernt.

5.3.1.4 Fazit

Zusammengefasst muss ich sagen: „Es war eine sehr schöne, aber harte Arbeit!“ Sehr vielseitig und anspruchsvoll. Wir konnten unser selbst eingebrachtes Thema umsetzen, was natürlich immer mit zusätzlichem Anspron verbunden ist. Mehr Zeit wäre in unserem Fall sicher sehr hilfreich gewesen um die noch etwas eckigen Kanten unseres Projektes abzurunden.

5.3.2 Michael Schnyder

5.3.2.1 Projektverlauf

Das Projekt war von Beginn an spannend. In einem sehr interessanten Meeting mit teils ehemaligen Arbeitskollegen von mir konnten wir mehr über die vorhandene Lösung der Swisscom (Openair Buddy) erfahren. Gleichzeitig konnten wir auch unsere Idee und unser generisches Konzept vorstellen. Die Resonanz war durchaus positiv, nur den Umfang unserer Arbeit schätzten sie anders ein als wir. Sie durften Recht behalten. Doch genau diese Einschätzung motivierte uns, gründlich voller Elan die Arbeit zu beginnen.

Anstatt direkt mit der Implementation zu beginnen, prüften wir als erstes unsere Annahmen über nützliche Funktionen für unsere Applikation. Viel Zeit floss in die Ausarbeitung und die Bewertung der durchgeführten Interviews am freestyle.ch und an der Züspa. Gerade die ersten Interviews hatten einen viel zu grossen Umfang. Dieser wurde für die zweite Interview-Runde angepasst. Die Rückmeldungen aus den Interviews war nebst den erhaltenen Erkenntnissen durchaus positiv. Schwierig war es, geeignete Personen zu finden, anzusprechen und mit ungebrochener Motivation „auszufragen“. Die Interviews halfen uns sehr, belegbare Szenarien und Use Cases auszuarbeiten. An diesen haben wir uns während dem Projektverlauf auch immer wieder orientiert. So kam auch nie das Gefühl auf, Funktionen zu implementieren, welche nicht benötigt würden. Nebst den Funktionen für die Besucher eines Grossanlasses, war es für uns auch immer wichtig, den Veranstalter als Stakeholder nicht aus den Augen zu verlieren. Schliesslich sollte das System auch für den Veranstalter sinnvoll einsetzbar sein. Hier konnte ich auf meine Erfahrung bei der Organisation von Grossanlässen zurückgreifen.

Wir haben uns intensiv mit der Architektur der Applikation und den benötigten Abläufen, Formaten und Protokollen beschäftigt. Als Grundlage diente uns die Developer-Webseite von Android, welche viele Informationen zu nützlichen Konzepten und Best Practices enthält. Leider decken diese Informationen nur immer einen kleinen Teil einer Applikation ab. Wir waren deshalb trotzdem gefordert diesen Aspekten in einem grösseren Kontext Beachtung zu schenken. Vieles war neu, vieles war anders. Android ist nicht Java, Android ist Java-Sourcecode in einer Umgebung, welche der Entwickler nicht beeinflussen kann. Intensiv habe ich mich mit dem Activity-Lifecycle¹, der sauberen Trennung zwischen UI und Daten, der Entwicklung von eigenen Widgets (User-Controls) und Threading² auseinandergesetzt. Zu Beginn der Entwicklung war die *AndroidManifest.xml*-Datei³ ein Dauerthema. Fehlt hier eine Information oder ist diese am falschen Ort erfasst, funktioniert die ganze Applikation nicht. Die Fehlersuche war jeweils sehr umständlich.

Doch es macht Spass für Android zu entwickeln, sobald die anfänglichen Probleme überwunden sind. Sie Lernkurve ist steil. Meine Motivation, möglichst viele der Best-

1 <http://developer.android.com/reference/android/app/Activity.html>

2 <http://d.android.com/guide/practices/design/seamlessness.html>

3 <http://d.android.com/guide/topics/manifest/manifest-intro.html>

Practices umzusetzen war darum bis zum Ende des Projekts ungebrochen. So kann ich mir auch die teilweise massiven Überstunden erklären und auch akzeptieren.

Überstunden gab es auch bei der Dokumentation. Es unglaublich aufwändig, aus vier verschiedenen Dokumenten die Anforderungen an die Arbeit abzuleiten. Natürlich sollte auch der Gesamtkontext erhalten bleiben. Es hat auch etwas schönes, die Dokumentation: Erst mit diesen Zeilen wir mir bewusst, an wie viele Dinge wir bei der Entwicklung beachtet und wie intensiv wir uns mit der Lösungsfindung auseinandergesetzt haben. Das manifestiert sich auch Umfang das Anhangs.

5.3.2.2 Rückblick

Ich blicke ein wenig ermüdet auf das Projekt zurück. Die letzten Wochen waren doch sehr intensiv und mit wenig Freizeit verbunden. Im Team haben wir uns gut ergänzt: Felix Egli konnte mich öfters mit pragmatischen Lösungen von einem übertriebenen Ansatz abhalten, umgekehrt habe ich eher den Sinn für die korrekte Umsetzung innerhalb der Android-Konzeptidee. Die Kommunikation war gut. Probleme konnten jederzeit angesprochen werden und führten immer zu einer einvernehmlichen Lösung. Die Entwicklung für die Android-Plattform war sehr spannend und auch herausfordernd. Dieses Projekt bleibt mir in guter Erinnerung, auch wenn der Endspurt selbst für meine Verhältnisse zu hektisch war.

5.3.2.3 Gelerntes

Intensiv habe ich mich mit Android auseinander gesetzt. Hier habe ich auch am meisten profitiert. Ebenfalls waren die Ideen und Konzepte hinter REST und des damit verbundenen Json-Datenformats. Die Usability-Tests, welche immer wieder während dem Projekt durchgeführt wurden erlebte ich positiv. Auch wenn das Feedback teilweise ernüchternd war, motivierte es, trotzdem die bemängelten Probleme zu beheben. Ich habe auch gelernt die Rückmeldungen aus diesen Tests ernst, aber nicht persönlich zu werten. Das war nicht immer einfach. Speziell stolz bin ich auf unsere Dokumentation mit \LaTeX . Obwohl es ein Risiko war, auf ein komplett neues Tool für die Dokumentation zu setzen hat es sich gelohnt. Hier durften wir glücklicherweise immer auf die spontane Unterstützung von anderen Teams zählen.

5.3.2.4 Fazit

Die Arbeit war toll. Unsere Probleme hinsichtlich Software oder auch in anderen Bereich konnten gelöst werden. Der Testlauf für die Bachelorarbeit ist geglückt und ich schaue gespannt auf die kommende Arbeit. Bestimmt wird es noch Änderungen in der Art der Projektplanung geben, hier haben wir beide noch Potential.

Mit dem Resultat der Arbeit bin ich zufrieden. Selbstverständlich gibt es immer noch Kanten und Ecken, welche noch geschliffen werden können. Doch irgendwann ist Schluss. Punkt.

5.4 Lessons Learned

Zusammenzug aus den oben genannten Problemen und Lösungen.

- **Android-Konzepte:** Eine unendliche Geschichte. Hier gibt es sicher für jede Applikation auf Android noch Verbesserungspotential. Ein guter Einstiegspunkt ist immer die Developer-Webseite von Android.
- **Android:** Ganz allgemein die Android-Umgebung mit Ihren Tücken und Möglichkeiten.
- **Unit Testing unter Android:** Ist einfach solange das Projekt unter Test von keinen zusätzlichen Projekten abhängig ist. Damit das Testing auch mit einem zusätzlichen Projekt funktioniert, muss die Referenz im zu testenden Projekt als exportierbar gewählt werden¹.
- **GPS Debugging mit der DDMS-Konsole:** Funktioniert nicht². GPS-Daten werden falsch gerundet. Das Feature ist unbrauchbar. GPS-Anwendungen können dadurch nur im Freien (z.T. bei eisiger Kälte) getestet werden.
- **L^AT_EX:** Speziell ohne WYSIWYG-Editor L^AT_EX in einem so grossem Dokument. Natürlich so vieles dass eine Aufzählung unmöglich wird.
- **Requirements Engineering:** Requirements Engineering auf eine eher ausgedehnte Weise, welche aber zwingend nötig war.
- **Planung:** Umgang mit der Planung bei einer neuen Technologie. Genügend Zeitbuffer einplanen.

¹ <http://blog.js-development.com/2010/06/android-instrumentation-test.html>

² <http://code.google.com/p/android/issues/detail?id=11264>

Teil II

SW-Projektdokumentation

6 Anforderungsspezifikation

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	06.10.2010	Felix Egli	Dokumentvorlage erstellt und erste Szenarios erfasst v1
0.2	07.10.2010	Felix Egli	Personas erfasst v1
0.3	10.10.2010	Michael Schnyder	Personas überarbeitet, Szenarios überarbeitet und erweitert
0.4	14.10.2010	Michael Schnyder	Szenarios gemäss Meeting-Feedback überarbeitet
0.5	18.10.2010	Michael Schnyder	Dokumentstruktur angepasst, UseCases und Nicht-funktionale Anforderungen
1.0	20.10.2010	Felix Egli, Michael Schnyder	Dokument auf Reviewstand gebracht.
1.1	16.12.2010	Felix Egli	Dokument Layout und kleinere Textliche anpassungen vorgenommen.
1.2	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

6.1 Personas

Diese wurden aus den Interviews (Freestyle und Züspa) abgeleitet. (siehe *Interviewauswertung Freestyle* auf Seite 157) (siehe *Interviewauswertung Züspa* auf Seite 167)

6.1.1 Peter Planer



Abbildung (6.1) Peter Planer

Name: Peter Planer
Alter: 24 Jahre
Freundin seit 3 Jahre
Beruf: Wirtschaftstudent
Hobbies: Ausgang, Kollegen, Musik
gute Computerkenntnisse
ordentliche Person
muss immer das neueste Handy besitzen

Tabelle (6.1) Profil Peter Planer

Seine dreijährige Lehre als Kaufmännischer Angestellter auf der Bank hat Peter Planer vor drei Jahr abgeschlossen. Seine Arbeit hat ihm immer sehr gut gefallen. Schon in der Lehre durfte er nach wenigen Monaten einige kleinere Projekte durchführen. Diese hat er mit grosse Passion und Präzission gemeistert.

Peter studiert aktuell Wirtschaft an der Fachhochschule St. Gallen. Er sieht das Studium als Investition in die Zukunft und möchte gerne einmal eine Führungsrolle übernehmen. Dennoch empfindet er die Inhalte im Studium etwas sehr theoretisch.

Peter Planer verdient sich mit Studentenjobs ein wenig Kleingeld und legt es sorgsam auf die hohe Kante. In der Sommerzeit wird er das Geld brauchen, denn er ist ein leidenschaftlicher Openair-Besucher. Er ist ein sehr gut organisierter und zielgerichteter junger Mensch, er versucht bei einem Openairbesuch alles so gut wie möglich im Voraus zu planen. *„Es gibt nichts schöneres als zu wissen was, wann, wo abgeht.“*

Um sein geplantes Leben zu vereinfachen, ist Peter ein fleissiger Smartphoneuser. *„Ohne mein Smartphone gehe ich nicht mehr aus dem Haus.“*

Dadurch ist sein technisches Verständnis sehr gut. Eine neue App die er sich auf sein Smartphone runterlädt ist schnell verstanden und optimal in Benutzung. Dennoch kämpft er hin und wieder mit einem zu grossem Funktionsumfang an einer Applikation. *„Manchmal sehe ich nur noch Buttons und weiss gar nicht mehr was ich überhaupt wollte.“*

6.1.2 Susi Sorglos



Abbildung (6.2) Susi Sorglos

Name:	Susi Sorglos
Alter:	18 Jahre single
Beruf:	Coiffure Lehrtochter
Hobbies:	Ausgang, Kollegen, Mode mässige Computerkentisse Chaotin kann iPhone Facebook nicht genug kriegen

Tabelle (6.2) Profil Susi Sorglos

Susi Sorglos konnte schon von kleinauf nie stillsitzen. Sie ist eine sehr energiegeladene Person und pausenlos auf Achse. Mit ihren 18 Jahren konnte sie ihr Leben schon in vollen Zügen geniessen. Jedes Wochenende besucht sie die wildesten Partys in der Region.

Im aktuellen Lebensabschnitt ist ihr die Lehre nicht so wichtig. So nimmt Sie es jeweils am Montagmorgen auch nicht so genau mit der Pünktlichkeit. Dennoch ist sie in der Berufsschule mit ihren Noten im Mittelfeld dabei. „*Ja id Schuel mueni eifach, da chani nüüt mache!*”

An den wilden Partyorgien dürfen ihre Kolleginnen auf keinen Fall fehlen. Sie sprechen sich meist am Abend zuvor auf Facebook ab, welche Party sie als nächstes besuchen wollen. Dabeisein ist alles, je mehr Freunde desto besser.

„*Letschi hani annere Party hani mindestens d’Helfti vo mine Facebookkollege troffe! Isch so geil xi!*”

Den Nachhauseweg findet sie, je nach Zustand mit der Hilfe ihres aktuellen iPhones.
„Amel am morgue nachem Usgang bini amel scho froh weni für de Fahrplan hei nur es paar Klicks bruche!“

Susi Sorglos shoppt gerne. Vorallem Ende Monat, wenn Sie wieder Geld auf Ihrem Konto hat. Meist ist danach ein grossen Teil ihres Lohns aufgebraucht und sie lebt den Rest des Monats auf Sparflamme. *„Es isch für mich scho immer recht schwierig xi mit Gäld umzgah!“*

6.1.3 Behavioural Variables

Aus Zeitgründen verzichtet (abgesprochen M. Stolze - 13. Oktober 2010).

6.2 Szenarios

6.2.1 Eventplanung am Tag zuvor

Peter Planer kann am Donnerstagabend nicht schlafen, denn er hat für das kommende grosse HSR-Openair am Wochenende die Planung noch nicht vorgenommen, wann er welche Band sehen möchte. Er zückt also sein Handy und startet EventAssistant. Mit wenigen Klicks kann er das Programm aufrufen und so schon seine Favoriten finden. Bei den „Fantastischen Vier“ und bei „Pausenbrot“ möchte er sich zusätzlich 15min vor Beginn benachrichtigen lassen. Dazu wählt er die entsprechende Aufführung aus und speichert seine Erinnerungen im EventAssistant. Peter ist froh, dass er seine Erinnerungen ohne aufzustehen erfassen konnte und schläft beruhigt ein.

6.2.2 Preise und Tickets

Ganz unverhofft erreicht Peter Planer ein Telefon seines Arbeitskollegen, welche nun doch noch am Wochenende ans Openair mitkommen möchte. Leider hat Peter nicht mit seinem Kollegen gerechnet und nun fehlt ein Ticket für die ganze Gruppe. Durch den EventAssistant kann Peter jedoch herausfinden, dass es noch Tickets gibt und lässt sich eines zur Abholung am Eingang reservieren. Den Preis sendet er per SMS an seinen Kollegen, welcher dann das Geld mitnehmen muss.

6.2.3 Eventablauf

Peter Planer isst eine Bratwurst. Da fragt ihn sein Sitznachbar wann Britney Spears ihren Auftritt hat. Peter Planer ist nicht Fan von Britney Spears, möchte aber seinem Sitznachbar helfen. Hierzu zieht er sein Mobiltelefon, startet den EventAssistant und

besorgt sich über wenige Klicks die Information, dass Britney Spears um 19:00 auf der Bühne 2 ihren Auftritt hat.

6.2.4 An MiniEvent erinnern lassen

Peter Planer sitzt an der Bar des Festgeländes eines Openairs, plötzlich vibriert in seiner Hosentasche sein Handy. Er schaut kurz auf sein Handy und wird erinnert, dass in 15 Minuten der Gig seiner Lieblingssängerin Pink! beginnt.

6.2.5 Wiederfinden des Autos

Peter Planer kommt sehr früh am Freitagabend zum Openair-Festgelände und stellt sein Auto in Mitte des sehr grossen Parkplatzes ab. Der EventAssistant teil ihm mit, dass er nun die Möglichkeit hat, seine Position zu markieren, so dass er später sein Auto wieder findet. So markiert er mit wenigen Klicks den aktuellen Standort auf der Karte.

Als er am Sonntagabend müde zum Parkplatz schlendert, steht er vor dem grossen Parkplatz und hat keine Ahnung mehr wo sein Auto steht. Er startet EventAssiant welcher ihn zu seinem Auto navigiert.

6.2.6 Orientierung am Event

Susi Sorglos ist in Mitten des Gig's von Gotthard. Sie muss dringend auf die Toilette. Nun ist aber alles von Leuten umstellt und Sisi kann sich an nichts orientieren. Sie zückt ihr Handy und startet EventAssistant und sucht den schnellsten Weg zur nächsten Toilette. Es ist sehr dringend.

Susi Sorglos sucht während dem Anlass die nächste Bus-/Bahnhaltestelle. Sie startet EventAssistant und sucht auf der Map die nächste Haltestelle und wird hingeführt.

6.2.7 Fotos

Susi Sorglos teil sich gerne mit. Gerade möchte sie ihre aktuell geschossenen Fotos mit den Eventteilnehmer teilen und hofft dass sie mit einem der Fotos am Wettbewerb gewinnt. Sie startet EventAssist und wählt die gemachten Fotos aus und überträgt sie in die allgemeine Gallery des Events.

6.2.8 Facebook

Susi Sorglos möchte erfahren ihre Kollegin Hanna Herzig, welche sie seit Jahren nicht mehr gesehen hat sich auch an dem aktuellen Anlass befindet. Sie startet EventAssist und verbindet ihn mit ihrem Facebookaccount und sieht, ob sich die Kollegin an diesem Event aufhält.

Peter Planer hat die Facebook Funktion auf dem EventAssistant schon seit Beginn des Events aktiviert und genießt aktuell das Openair-Geschehen. Da erhält er eine Nachricht auf seinem Handy und der EventAssistant öffnet sich. Auf der Karte wird die Position seines langjähriger Schulfreundes angezeigt, welcher gerade das Openair-Gelände betreten hat.

6.2.9 Verschiebung

Kurzzeitig wird am Event eine Band verschoben. Peter Planer wird sofort auf seinem Handy notifiziert, ohne dass er explizit die App startet.

6.2.10 Stimmung

Susi Sorglos und Ihre Kolleginnen sind müde. Sie möchten aber noch genau ein Party-Zelt besuchen. Können sich aber nicht entscheiden. Susi startet EventAssistant und sieht, dass im „Party-Tempel“ am meisten Männer sind und die Besucher sich dort am meisten bewegen. Der Weg zum „Party-Tempel“ ist nicht weit und EventAssistant führt sie des Weges.

6.3 Priorisierung der Szenarien

Kontext der Applikations wird in dem must und optimalen Bereich an einem **Openair** liegen.

6.3.1 Must

- Eventablauf (Auflistung der Bands wenn wer wo spielt)
- Orientierung am Event (Über eine Karte)

6.3.2 Optional

- Eventplanung am Tag zuvor (Erinnerungen gesetzt)
- Eventerinnerung (Reminder für bestimmte Bands)
- Facebook (ohne Positionierung, sie befindet sich an diesem Event)
- Verschiebung (Pushnotifikation)

6.3.3 Abgrenzung

- Preise und Tickets
- Wiederfinden des Autos
- Fotos
- Stimmung

6.4 Use Cases

6.4.1 Übersicht

Nachfolgend eine Übersicht über die maximal implementierten Use Cases. Diese unterteilen sich in die Kategorien Besucher und Veranstalter. Dies sind in die Kategorien **Must**, **Optional** und **Abgrenzung** eingeteilt. Diese Einteilung entspricht der Einteilung aus den Szenarien.

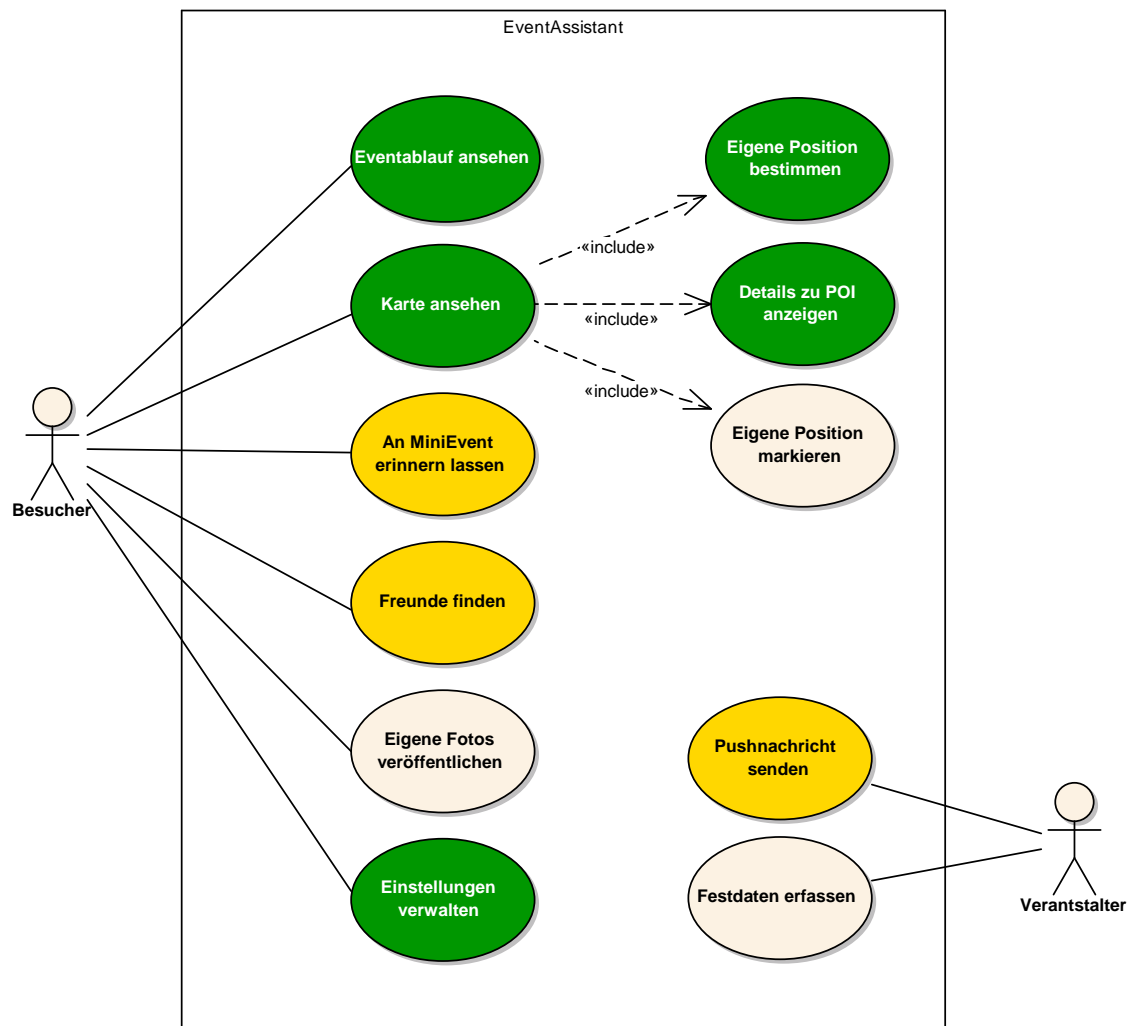


Abbildung (6.3) Use Cases

6.4.2 Besucher

6.4.2.1 UC B01: Festivalablauf ansehen

Beschreibt den Ablauf, bis ein Benutzer das Programm von einem gewählten Event anzeigen kann.

1. Der Benutzer startet Event Assistant
2.
 - a) Der aktuelle (positionenabhängig) Event wird geladen
 - b) Der Benutzer erhält eine Auswahl der Events, welche gerade stattfinden und wählt einen aus
3. Der Benutzer sieht die gewünschten Informationen auf seinem Bildschirm

6.4.2.2 UC B02: Karte ansehen

Dieser Use-Case ist die Grundannahme für die zwei folgenden.

Eine Position bestimmen Der Anwender kann seine eigene Position auf der Karte bestimmen. Hierzu muss GPS eingeschaltet und aktiviert sein.

Eine Position markieren Der Anwender markiert seine Position oder löscht eine bereits getätigte Positionsmarkierung.

Details zu POI anzeigen Wichtige Orte auf dem Festgelände können Zusatzinformationen enthalten, welche der Benutzer per Klick anzeigen kann.

6.4.2.3 UC B03: Erinnerungen

Beinhaltet die folgenden Use-Cases.

Erinnern erfassen Per Klick auf einen Timed-Event kann eine Erinnerung generiert werden, welche lokal gespeichert wird. Dies wird vom System bestätigt. Wenn die Threshold für die Erinnerung erreicht ist, wird der Benutzer notifiziert.

6.4.2.4 UC B04: Freunde Finden

Dieses Use Case kann auf zwei Arten angestoßen werden:

Freunde aktiv suchen

1. Die Applikation wird gestartet und UC01 ausgeführt.
2. Der Benutzer lässt sich eine Liste der Freunde anzeigen
3. Aus der Liste lassen sich die Freunde kontaktieren.

Benachrichtigung Der Benutzer wird informiert, dass einer seiner Facebook-Freunde das Gelände betreten hat.

6.4.2.5 UC B05: Eigene Fotos veröffentlichen

Wird nicht implementiert und ist aus diesem Grund nicht genauer beschrieben.

6.4.2.6 UC B06: Einstellungen verwalten

1. Die Applikation wird gestartet.
2. Der Benutzer gelangt durch einen Einstellungs-Option in die Einstellungs-Liste
3. Diverse Einstellungen stehen zur Verfügung.
4. Das Speichern der Einstellungen wird quittiert.

6.4.3 Veranstalter

Die folgenden Use Cases werden nicht implementiert und sind deshalb nicht genauer beschrieben.

6.4.3.1 UC V01: Pushnachricht senden

Dieses Use Case wird nicht implementiert und ist deshalb nicht genauer beschrieben.

6.4.3.2 UC V02: Festdaten Erfassen

Dieses Use Case wird nicht implementiert und ist deshalb nicht genauer beschrieben.

6.5 Nichtfunktionale Anforderungen

6.5.1 Speicherung

Die Eventdaten müssen an einem zentralen Punkt verwaltet werden können. Diese Schnittstelle muss so gewählt werden, dass diese möglichst einfach austauschbar ist.

6.5.2 Geo-Referenzierung

Kartendaten sollen nicht aus einzelnen Grafiken bestehen, welche auf den Client gerendert werden. Vielmehr soll eine einzelne Grafik „aktive Regionen“ (Hotspots) besitzen, mit welchen der Benutzer durch geeignete Interaktionsmöglichkeiten interagieren kann. Dies macht das Erstellen einer Karte einfacher. Es müssen nicht einzelne Grafiken (Icons für POI, etc.) für aktive Regionen erstellt und mit Geo-Informationen referenziert werden. Die Icons können direkt in die Karte integriert werden.

6.5.3 Client Cache

Es muss sichergestellt sein, dass der Client auch dann funktioniert, wenn keine Verbindung zum Netzwerk besteht. Der Client soll eine Art Cache implementieren.

6.5.4 Performance

Die Wartezeiten auf Client und Serverseite sollten sich im angemessenen Rahmen aufhalten.

6.5.5 UI-Responsiveness

Bei längeren Wartezeiten auf dem UI muss entsprechend reagiert werden.

6.5.6 Akkuverbrauch / Ressourcen schonen

Keine unnötigen Threads im Hintergrund aktiv, oder teure Berechnungen auf dem Android-Client.

7 Analyse

Dokumenthistory

Rev.	Datum	Wer	Änderung
1.0	12.10.2010	Felix Egli	Dokument erstellt und die zwei Prototypen dokumentiert
1.0	20.10.2010	Felix Egli, Michael Schnyder	Dokument auf Reviewstand gebracht.
1.1	16.12.2010	Felix Egli	Objektkatalog erstellt und DomainModel auf aktuellsten Stand gebarcht.
1.2	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

7.1 Domain Model

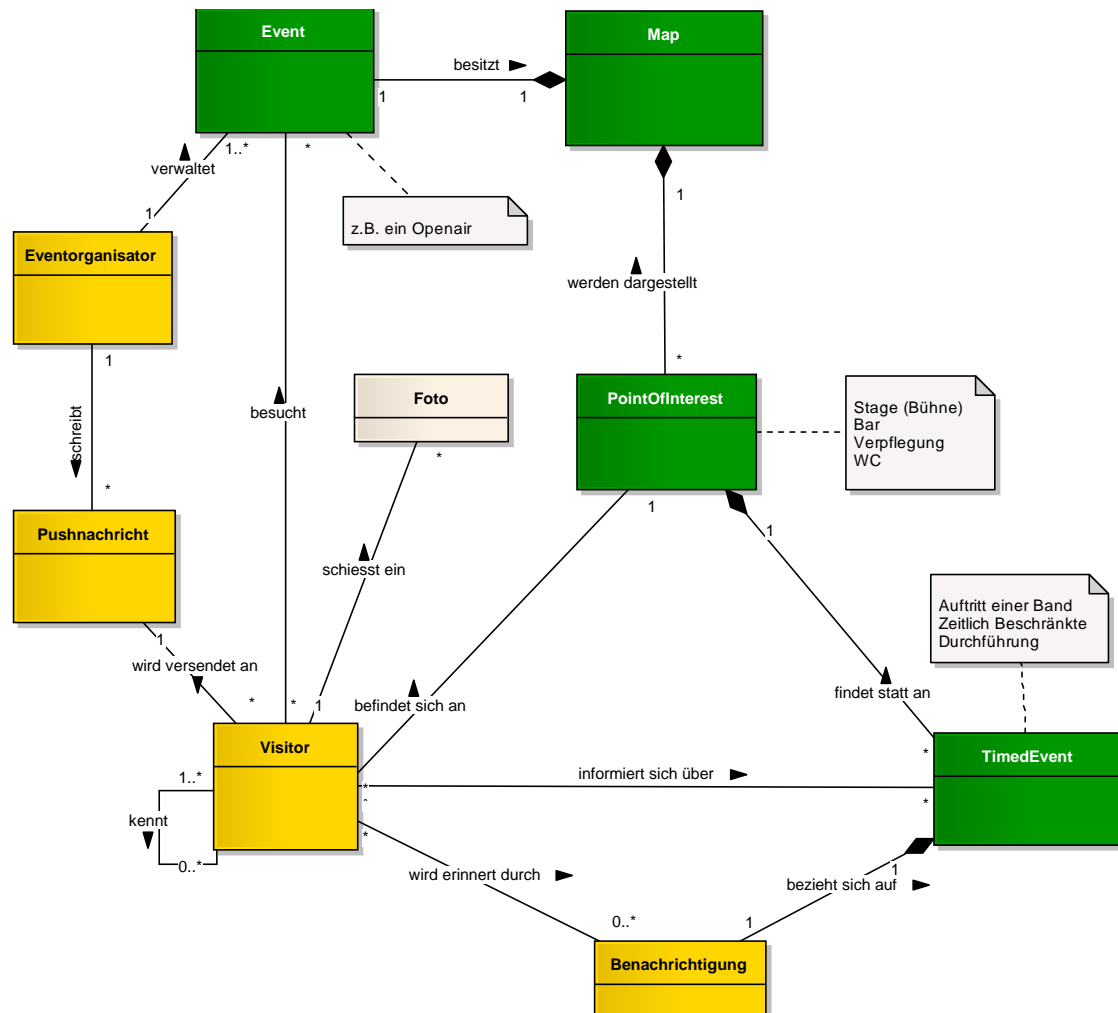


Abbildung (7.1) Problem Domain

Die oben dargestellten Konzepte im Rahmen des Domain Modells sind aus den UseCases und den Szenarien abgeleitet. Wie auch die Use-Cases sind diese in die Kategorien **Must**, **Optional** und **Abgrenzung** eingeteilt. Diese Einteilung entspricht wiederum der Einteilung aus den Szenarien.

7.2 Objektkatalog

7.2.1 Event

Must Dies repräsentiert den Event an dem man sich befindet, oder der ausgewählt werden kann.

7.2.2 Map

Must Jeder Event besitzt eine Karte auf der wichtige Punkte eingezeichnet sind.

7.2.3 PointOfInterest

Must Dies repräsentiert eine Bühne eine Bar oder sonst einen wichtigen Punkt, welcher sich am Event auf der entsprechenden Karte befindet.

7.2.4 TimedEvent

Must Der Timedevent ist zum Beispiel einen Gig einer Band. Welcher an einem Point of Interest stattfindet zu einer bestimmten Zeit.

7.2.5 Eventorganisator

Optional Der Eventorganisator verwaltet sein Event, ist verantwortlich, dass die Karte und die Point of Interests und das TimedEvent erfasst sind.

7.2.6 Visitor

Optional Dies ist der Besucher welcher den Event besucht, dieser kann dann auch via Pushnachricht notifiziert werden.

7.2.7 Pushnachricht

Optional Der Eventorganisator sendet eine Pushnachricht an einen Besucher des Events.

7.2.8 Benachrichtigung

Optional Der Besucher kann sich für einen Timed Event notifizieren lassen. Und wird somit eine gewisse Zeit vor dem Timed Event darüber informiert.

7.2.9 Foto

Abgrenzung Dies die Fotos die der Eventbesucher machen kann und veröffentlichen.

8 Design

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	12.10.2010	Michael Schnyder	Dokument erstellt
0.2	16.12.2010	Felix Egli	Packagediagramm und Klassendiagramme erstellt
0.3	18.12.2010	Felix Egli	Packagediagramm, Klassendiagramme, Architekturübersichtsdiagramm erweitert und übersichtlicher gemacht
0.4	19.12.2010	Michael Schnyder	Architektur-Overview hinzugefügt.
0.5	19.12.2010	Michael Schnyder	Content Provider fertig beschrieben
0.6	19.12.2010	Michael Schnyder	REST-Schnittstelle (Server) beschrieben
0.6	20.12.2010	Michael Schnyder	UI-Screens und Organisation beschrieben
0.7	21.12.2010	Michael Schnyder	Datenbankschema beschrieben (short), Formate fertiggestellt
0.8	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

8.1 Architektur

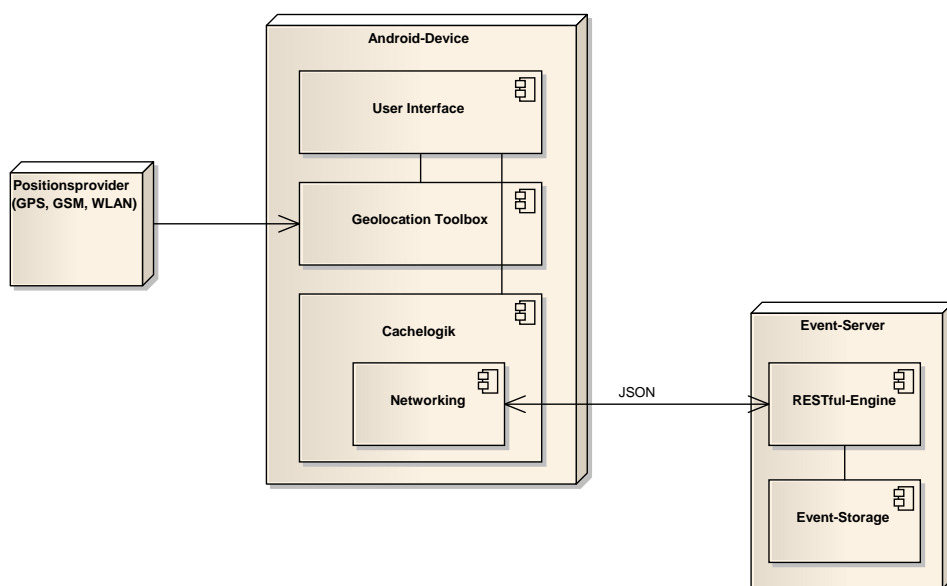


Abbildung (8.1) Architektur Overview

Obenstehendes Diagramm zeigt die Verteilung von EventAssistant auf den verschiedenen Geräten. Nachfolgend eine Übersicht über die beteiligten Komponenten, welche danach in

der Packagestruktur weiter beschrieben werden.

8.1.1 Komponenten

8.1.1.1 Positionsprovider

Die Positionsdaten werden zum grössten Teil via *GPS*¹ empfangen. Nebst *GPS* wird auch Triangulation für *GSM*² und *WLAN*³ verwendet um schnell eine Position des Benutzers zu erhalten.

8.1.1.2 Android-Device

Die Applikation auf dem Android-Device besteht aus den folgenden Hauptkomponenten:

- **User Interface:** Umfasst alle sichtbaren Teile und Hilfskonstrukte
- **Geolocation Toolbox:** Stellt Hilfsmethoden für Geo-Berechnungen zur Verfügung.
- **Cache-Logik:** Verwaltet den Lokalen Cache und greift über **Networking** auf den Event-Server zu, sofern erforderlich.

8.1.1.3 Event-Server

Der Event-Server dient als Referenzimplementation um auf den Client sinnvolle Daten zu erhalten und die Kommunikation und Datenformate zu testen. Die Architektur ist so gewählt, dass eine reale Datenquelle mit kleinem Aufwand implementiert werden kann. Die Anbindung an eine Datenbank ist somit möglich.

8.1.2 Android-Konzepte

Obwohl die Android-Architektur dem Projektteam fremd war, wurden wenn immer möglich empfohlene Konzepte von Android verwendet. Dazu gehören z.B. die Verwendung von ContentProvidern⁴, die saubere Trennung zwischen UI-Code und Applikationslogik oder die Wiederverwendung von UI-Segmenten mittels include-Befehl⁵. Viele dieser spezifischen Konzepte sind unter Implementation (siehe *Implementations-Konzepte* auf Seite 78) genauer beschrieben.

1 *Global Positioning System*

2 *Global System for Mobile Communications*

3 *Wireless Local Area Network*

4 <http://developer.android.com/guide/topics/providers/content-providers.html>

5 <http://developer.android.com/resources/articles/layout-tricks-reuse.html>

8.2 Packagestruktur

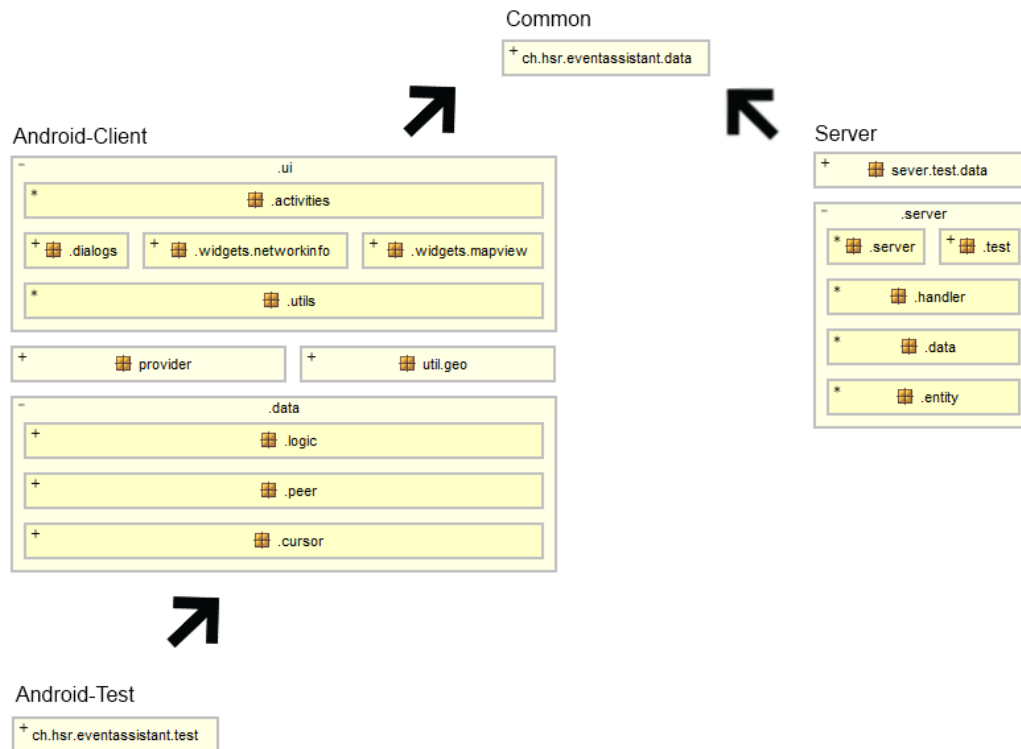


Abbildung (8.2) Package Overview

Alle Pakete beginnen mit dem Prefix *ch.hsr.eventassistant*.

8.2.1 Projekt „EventAssistant-Common“

In diesem Projekt befinden sich alle Klassen welche auf dem Client und auf dem Server verwendet werden. Dazu gehören die Datenklassen mit den Properties, die der Client wie auch der Server Verwendung finden.

8.2.2 Projekt „EventAssistant“

Der ganze Android Client ist in vier Hauptpakete unterteilt. Diese repräsentieren die verschiedenen Schichten.

- das .ui Package für das User Interface

- das .provider Package für den ContentProvider¹
- das .util Package für die Utilitys
- das .data Package für das Datenhandling

8.2.2.1 UI-Package

Das UI Packet setzt sich, wie der Name schon sagt, mit dem User-Interface auseinander. Hierzu gehören folgende Subpakete:

Packagename	Zweck
.activities	Hier befinden sich alle Android-Activities die verwendet werden auf dem Device.
.dialogs	Hier befinden sich alle Dialoge die im Programm vorkommen.
.widgets	Hier befindet sich die modulare Implementation von Offlinemodus und dem MapControl.
.utils	Hier befinden sich diverse kleine Helfer die das UI verwendet.

Tabelle (8.1) UI-Packete

8.2.2.2 Provider-Package

Im Provider-Package befindet sich der EventAssistant-ContentProvider, welcher vom Android ContentProvider² abgeleitet ist. Dieser stellt unsere Daten anderen Applikationen, wie auch unserer UI-Schicht zur Verfügung.

8.2.2.3 Util-Package

Hier befindet sich nur das Subpaket *geo*. Darin sind allgemeine Helfer für die Georeferenzierung vorhanden und die Umrechnung von Koordinaten.

8.2.2.4 Data-Package

Das data-Packet ist in folgende Unterpakete unterteilt.

¹ <http://developer.android.com/guide/topics/providers/content-providers.html>

² <http://developer.android.com/guide/topics/providers/content-providers.html>

Packagename	Zweck
.cursor	Hier drin befinden sich der speziell erstellte CursorAdapter, welcher Listen von ContentValues ¹ in einen Cursor umwandeln kann.
.logic	Darin befindet sich die ganze Logik wie die Daten gecached werden sollen. Auch das Verhalten was zu tun ist, wenn kein Netz verfügbar ist, ist hier definiert.
.peer	Hier sind die verschiedenen Peers abgelegt, welche zum Beispiel auf die interne Datenbank zugreifen oder über das Internet auf den Remoteserver Daten laden.

Tabelle (8.2) Data-Packete

8.2.3 Projekt „EventAssistant-Server“

Hier befinden sich alle Klassen die es benötigt um den Server zu betreiben. Diese sind in folgende Packete aufgeteilt.

Packagename	Zweck
.server	In diesem Paket befindet sich das Servlet, welches die HTTP-Requests handelt.
.server.data	In diesem Paket befindet sich die Datasource welche die Daten zur Verfügung stellt.
.server.entity	In diesem Paket befinden sich alle Datenobjekte, welche für das JSON Marshalling annotiert sind.
.server.handler	Darin sind alle Handler welche auf die von ihnen registrierte URL matchen.
.server.test	Hier befinden sich alle JUnit Testfälle welche alle Entitäten auf ihre Korrektheit und Vollständigkeit prüft.
.server.test.data	Hier befinden sich alle JUnit Testfälle welche die Datasource testen.

Tabelle (8.3) Server-Packete

8.2.4 Projekt „EventAssistant-Test“

Hier befinden sich alle Android Testfälle, welche alle androidspezifischen Funktionalitäten testen, welche einen Android-Kontext² verwenden.

² <http://developer.android.com/reference/android/content/Context.html>

8.3 Klassendiagramm

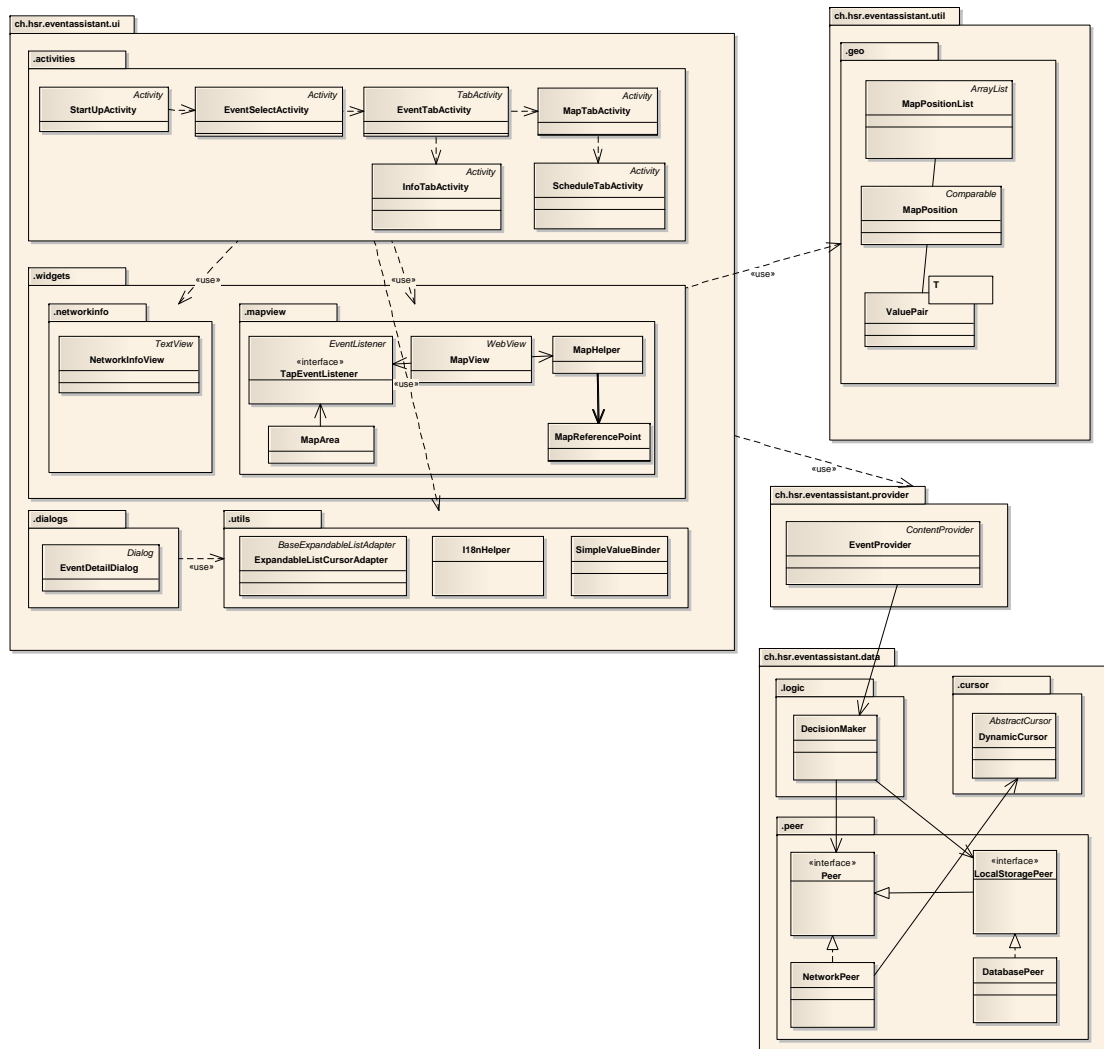


Abbildung (8.3) Klassen Overview

In diesem Diagramm wird das grobe Zusammenspiel der Packages und Klassen gezeigt. Hier sind schon einige Konzepte erkennbar. (siehe *Implementations-Konzepte* auf Seite 78)

8.4 Datenbankschema

Auf dem Android-Client wird eine SQLite-Datenbank eingesetzt. Diese wird ausschliesslich vom Cache-Mechanismus verwendet. Die Datenbank ist analog den Datenklassen aufgebaut und wird deshalb nur oberflächlich beschrieben. Folgenden Konventionen wurden für die Erstellung der benötigten Tabellen auf Basis der Datenklassen verwendet (siehe *Datenklassen Übersicht* auf Seite 65).

Betrifft	Konvertierung/Datenformat	Beispiel
Entitätsname	lower-case	Event wird zu event
Integer-Werte	keine	integer not null
String-Werte	keine	string not null
DateTime-Werte	keine	string not null

Tabelle (8.4) Datenbankschema Konventionen

8.5 Schnittstellen und Formate

8.5.1 Schnittstelle zu ContentProvider

Alle Abfragen zum ContentProvider [AB10] beginnen immer mit derselben URI: `content://ch.hsr.eventassistant.provider.EventProvider/event` und werden von Android über eine ManagedQuery verwaltet. (siehe *Content Provider und Lokaler Cache* auf Seite 81). Als Rückgabe erhält der Aufrufer jeweils einen Cursor, mit welchem er durch die Ergebnismenge navigieren kann. Falls ein spezifischer Eintrag via Id angefordert wurde, befindet sich dieser ebenfalls in einer Liste, einfach an erster Position. Die Liste hat hierbei die Länge eins. Dabei unterscheiden sich die einzelnen Funktionen für den Datenabruf, in dem Namen der Ressource und von Parametern, welche der URI angehängt werden.

8.5.1.1 Parameter - Grundlagen

Als Grundlage dient jeweils die Unterscheidung zwischen einer Liste (evtl. mit Kriterien in Form von Parametern) mit Einträgen oder genau einem Element, welches durch seine Id eindeutig referenzierbar ist. Nachfolgendes Beispiel beschreibt genau dieses Vorgehen beim Abruf von Auftritten für den ganzen Event (Parameterlos), eines spezifischen Auftritts (Zahl als Index ohne Bezeichner) oder der Liste der Auftritte an einem bestimmten POI (benannter Fremd-Parameter):

- **Liste aller Auftritte** (Parameterlos):

```
content://ch.hsr.(...).EventProvider/event/2/schedule
```

- **Auftritt mit Id 2 (Index):**

`content://ch.hsr.(...).EventProvider/event/2/schedule/2`

- **Auftritte am POI mit Id 5 (benannter Fremd-Parameter):**

`content://ch.hsr.(...).EventProvider/event/2/schedule/?poi=5`

Da sich im aktuellen Entwicklungsstand alle Abfragen auf eine spezifischen Event beziehen ist nach dem *FQDN*¹ auch das erste Segment „/event“ vorgegeben. Dahinter befinden sich die verschiedenen Ressourcen wie folgt beschrieben.

8.5.1.2 Ressourcen

Für die Abfrage stehen die nachfolgenden Ressourcen zur Verfügung:

Ressource	Beispiel-URI	Datenklasse
Eventliste	.../event	EventListItem
Einzelner Event	.../event/2	Event
Karte (Bild) für Event	.../event/2/map	JPEG-Bild
Karteninformationen für Event	.../event/2/map/info	Event
Zeitplan für Event	.../event/2/schedule	TimedEvent
Zeitplan für POI	.../event/2/schedule/?poi=5	TimedEvent
Koordinaten Polygon	.../event/2/area/12	Position
Alle POI's für Event	.../event/2/poi	PointOfInterest
Spezifischer POI	.../event/2/poi/5	PointOfInterest

Tabelle (8.5) Ressourcenbeschreibung

Wie Eingangs bereits erwähnt, kann der (von Android (siehe *ContentValues* und *Cursors* auf Seite 79)) erhaltene Cursor mit den Spaltennamen des entsprechenden Datentyps abgefragt werden. Dabei sind sowohl Listen als auch einzelne Einträge als Liste behandelt, d.h. wenn nur ein spezifischer Eintrag abgerufen wurde, befindet sich dieser an der ersten Position beim Cursor.

Es gilt zu beachten, dass bei der Liste der Events nicht die gleichen Daten zur Verfügung stehen als bei einem spezifischen Event. Grund hierfür ist die Grösse der Antwort, welche vom Server zurückgeliefert werden muss. Durch weglassen unwichtiger Informationen konnte die Grösse reduziert werden.

Einen Spezialfall stellt hier die Karte dar, welche direkt als JPEG zurückgegeben wird. Die Karte kann dadurch direkt im MapView-Control (oder einem beliebigen anderen Widget, welches Bilder anzeigen kann) angezeigt werden.

¹ *Eindeutiger Domain Name (engl. Fully Qualified Domain Name)*

Die Daten, bzw. die verfügbaren Spalten entsprechen den static-Fields der folgenden Klassen und befinden sich im Package `ch.hsr.eventassistant.data` des Common-Projekts (siehe *Projekt „EventAssistant-Common“* auf Seite 59).

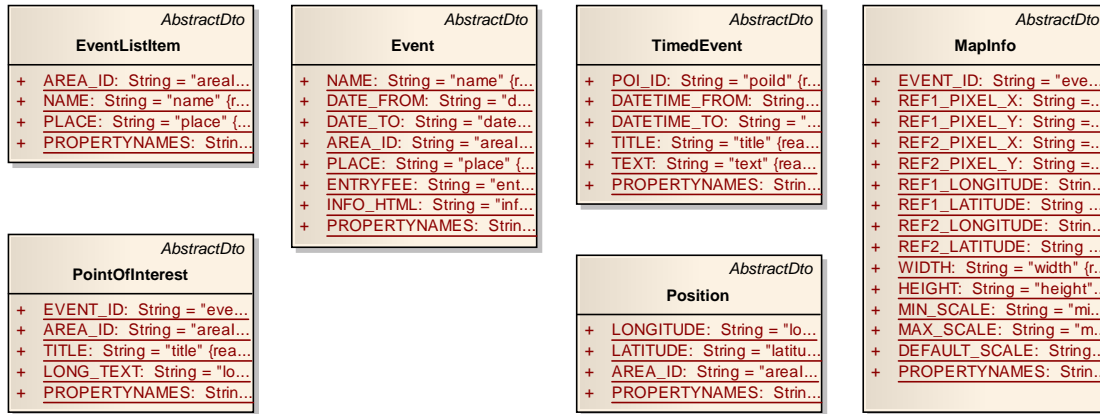


Abbildung (8.4) Datenklassen Übersicht

8.5.1.3 Beispiel

Mit dem folgenden Beispiel wird die Liste der Events geladen und auf der Console ausgegeben. Durch die Definition der Spaltennamen als static fields in den „pseudo“ Datenklassen (siehe *Datenklassen Vererbung* auf Seite 78) können diese als Zugriffsparameter verwendet werden. Es wird damit ausgeschlossen, dass z.B. Spaltennamen falsch geschrieben werden:

```

1  ...
   Uri allEvents = "content://.../event";
   listCursor = managedQuery(allEvents, null, null, null, null);

5  if(listCursor.moveToFirst() {
      do {
         String columnName = EventListItem.NAME;
         int columnIndex = listCursor.getColumnIndex(columnName);
         System.out.println("Eventname: " + listCursor.getString(columnIndex));
10  } while (listCursor.moveToNext());
   }
   ...

```

8.5.2 Event-Assistant Server (REST-Schnittstelle)

Der Event-Assistant Server dient als Referenzimplementation (siehe *Referenz-Server* auf Seite 83) für das Kommunikationsformat zwischen Client und Server. Analog den oben

genannten Zugriffen auf den ContentProvider der Applikation existiert je auch ein direkter Zugriff auf die entsprechenden Daten auf dem Server. Diese Kommunikation wird mit JSON über HTTP gehandhabt. ContentType im Answerheader ist je nach Anwendungsfall `text/json` oder `image/png`. Als Format wurde JSON gewählt, da es sich sehr gut lesen lässt, jedoch nicht so einen grossen Overhead wie XML oder gar SOAP besitzt. Als Basisadresse gilt hier <http://eventassistant.dyndns.org/EventAssistant-Server/event>. Hierbei ist zu beachten, dass sowohl der Domain-Namen, wie auch das erste Segment des Pfades auf dem Server frei gewählt werden kann. Die Struktur ist erst ab und mit „/event“ vorgegeben. Identisch zum Schema beim ContentProvider wird auch hier wieder zwischen Listen, exakten Elementen und gefilterten Listen unterschieden.

Datenformate und Konvertierung Das Protokoll unterstützt die gängigen Datenformate. Diese sind wie folgt definiert:

Java-Datentyp	Konvertierung	Beispiel
String	Nein	"aBCde12345"
Boolean	Ja	"1" / "0" oder "true" / "false"
Integer, int	Nein	10
Double	Nein	4.25
Date (mit Sekunden)	Ja	"19.12.2010 22:51:00"
Date (ohne Sekunden)	Ja	"19.12.2010 22:51"
Grafiken	Ja	JPEG oder PNG

Tabelle (8.6) Datenformate und Konvertierung

Die Namen der Felder müssen hierbei identisch mit den auf dem Client definierten Spalten sein. (siehe *Datenklassen Vererbung* auf Seite 78). Die Spezifikation definiert keine Sortierung bei Listen, diese ist Sache des Clients.

Die nachfolgenden Abfragen sind auf Serverseite zu implementieren:

8.5.2.1 Eventliste

Alle Events werden zurückgegeben.

Pattern: `.* /event [/]? $`

Beispiel-URL: `http://... /event`

Antwort

```
1 [
  {
    "_id" : 1,
```



```

5      "areaId"   : 1,
      "expires"  : "19.12.2010 22:02:58",
      "name"     : "HiddenAir",
      "place"    : "Hittenberg"
    },
    {
10     "_id"      : 2,
      "areaId"   : 2,
      "expires"  : "19.12.2010 22:02:58",
      "name"     : "HSR-Openair",
      "place"    : "Rapperswil"
15    },
    {
      "_id"      : 3,
      "areaId"   : 3,
      "expires"  : "19.12.2010 22:02:58",
20     "name"     : "Openair Dortmund",
      "place"    : "Westfalenpark Dortmund"
    }
  ]

```

8.5.2.2 Einzelner Event

Nur gewählter Event wird zurückgegeben.

Pattern: `.*event/([1-9][0-9]*)[/]?$`

Parameter: (1) = EventId

Beispiel-URL: `http://.../event/2`

Antwort

```

1  {
      "_id"      : 2,
      "areaId"   : 2,
      "dateFrom" : "17.05.2011 20:00",
5   "dateTo"     : "19.05.2011 21:30",
      "entryfee" : "Samstag: 50.- CHF\nSonntag: 65.- CHF",
      "expires"  : "20.12.2010 00:38:02",
      "infoHtml" : "Das HSR-Openair (...) werden k&ouml;nne.",
      "name"     : "HSR-Openair",
10     "place"    : "Rapperswil"
    }

```

8.5.2.3 Karte (Bild) für Event

Rückgabe der Karte als JPEG- oder PNG-Datei

Pattern: `.*event/([1-9][0-9]*)/map[/]?$`

Parameter: (1) = EventId

Beispiel-URL: <http://.../event/2/map>

Antwort

```
1  ### Grafik im JPEG- oder PNG-Format ###
```

8.5.2.4 Zeitplan für Event

Alle Spielzeiten werden zurückgegeben.

Pattern: `.*event/([1-9][0-9]*)/schedule[/]?s`

Parameter: (1) = EventId

Beispiel-URL: <http://.../event/2/schedule>

Antwort

```
1  [
    {
        "_id":1,
        "datetimeFrom":"19.08.2011 17:00",
    5  "datetimeTo":"19.08.2011 19:00",
        "expires":"19.12.2010 22:10:04",
        "poiId":"1",
        "text":"Fetteflossbaraction mit der Flossbarband!!!",
        "title":"Flossbar-Band"
    10  },
    {
        "_id":2,
        "datetimeFrom":"19.08.2011 19:30",
        "datetimeTo":"19.08.2011 21:30",
    15  "expires":"19.12.2010 22:10:04",
        "poiId":"1",
        "text":"Das Rockspektakel aus dem Oberland!!!",
        "title":"Cudvain"
    20  },
    {
        "_id":18,
        "datetimeFrom":"20.08.2011 18:00",
        "datetimeTo":"20.08.2011 20:00",
    25  "expires":"19.12.2010 22:10:04",
        "poiId":"2",
        "text":"Die Technologie!",
        "title":"Milow"
    }
  ]
```

8.5.2.5 Zeitplan für POI

Alle Spielzeiten eines POI's werden zurückgegeben.

Pattern: `.*event/([1-9][0-9]*)/schedule/\?poi=([1-9][0-9]*)$`

Parameter: (1) = EventId, (2) = PoiId

Beispiel-URL: `http://.../event/2/schedule/?pid=2`

Antwort

```

1  [
    {
      "_id"           : 1,
      "datetimeFrom"  : "19.08.2011 17:00",
      "datetimeTo"    : "19.08.2011 19:00",
      "expires"       : "19.12.2010 22:10:04",
      "poiId"         : 1,
      "text"          : "Fetteflossbaraction mit der Flossbarband!!!",
      "title"         : "Flossbar-Band"
    },
    {
      "_id"           : 2,
      "datetimeFrom"  : "19.08.2011 19:30",
      "datetimeTo"    : "19.08.2011 21:30",
      "expires"       : "19.12.2010 22:10:04",
      "poiId"         : 1,
      "text"          : "Das Rockspektakel aus dem Oberland!!!",
      "title"         : "Cudvain"
    }
  ]

```

8.5.2.6 Koordinaten Polygon

Eine Liste von Koordinaten für einen Bereich auf der Karte wird zurückgegeben

Pattern: `.*event/([1-9][0-9]*)/area/([1-9][0-9]*)$`

Parameter: (1) = EventId, (2) = AreaId

Beispiel-URL: `http://.../event/2/area/2`

Antwort

```

1  [
    {
      "_id"           : 1,
      "areaId"        : 2,
      "expires"       : "20.12.2010 00:46:02",
      "latitude"      : "47.22500",
      "longitude"     : "8.81445"
    },
    {

```

```

10  " _id"      : 2,
    "areaId"   : 2,
    "expires"  : "20.12.2010 00:46:02",
    "latitude" : "47.22444",
    "longitude" : "8.82007"
15  },
    {
    " _id"      : 3,
    "areaId"   : 2,
    "expires"  : "20.12.2010 00:46:02",
    "latitude" : "47.22164",
    "longitude" : "8.81895"
20  },
    {
    " _id"      : 4,
    "areaId"   : 2,
    "expires"  : "20.12.2010 00:46:02",
    "latitude" : "47.22213",
    "longitude" : "8.81443"
25  }
    ]
30

```

8.5.2.7 Alle POI's für Event

Eine Liste von allen POI's für einen Event

Pattern: `.*event/([1-9][0-9]*)/poi[/]?$`

Parameter: (1) = EventId

Beispiel-URL: `http://.../event/2/poi`

Antwort

```

1  [-
    {
    " _id"      : 1,
    "areaId"   : 5,
    "eventId"  : 2,
    "expires"  : "19.12.2010 22:03:09",
    "longText" : "Die Flossbar ist die beste und lustigste Bar am (...).",
    "title"    : "Flossbar"
    },
10  {
    " _id"      : 2,
    "areaId"   : 6,
    "eventId"  : 2,
    "expires"  : "19.12.2010 22:03:09",
    "longText" : "Die Stage Nr. 1 ist die gr&ouml;sste und (...)",
    "title"    : "Stage Nr. 1"
15  },
    ]

```

```

20  {
    "_id"      : 3,
    "areaId"   : 7,
    "eventId"  : 2,
    "expires"  : "19.12.2010 22:03:09",
    "longText" : "Der beste Würstlistand des (...)",
    "title"    : "Würstlistand"
25  },
    {
        "_id"      : 4,
        "areaId"   : 8,
        "eventId"  : 2,
        "expires"  : "19.12.2010 22:03:09",
        "longText" : "Hier treffen sich nur die wichtigen (...)",
        "title"    : "VIP-Lounge"
30  }
  ]

```

8.5.2.8 Spezifischer POI's von Event

Liefert einen spezifischen POI eines Events

Pattern: `.* /event/ ([1-9] [0-9]*) /poi/ ([1-9] [0-9]*) $`

Parameter: (1) = EventId, (2) = PoiId

Beispiel-URL: `http://.../event/2/poi/`

Antwort

```

1  {
    "_id"      : 2,
    "areaId"   : 6,
    "eventId"  : 2,
5   "expires"  : "19.12.2010 22:03:09",
    "longText" : "Die Stage Nr. 1 ist die grösste und (...)",
    "title"    : "Stage Nr. 1"
  }

```

8.5.3 Koordinatenformat

Alle Koordinaten werden im als Lat und Long-Werte erfasst (siehe DIN 13.312). Hierbei sind die Werte für Long und Lat als Dezimalzahlen dargestellt. Die Koordinaten (der HSR) 47° 13'24.1104" und 8° 49'1.956" (In Grad, Minuten und Sekunden) entspricht dabei den Werten 47.223364, 8.817210 (in Dezimalgrad). Wir verwenden die Werte in Dezimalgrad.

8.5.4 Datenformat für Karten

Als Karte können beliebige Bilder im PNG oder JPG Format verwendet werden. Hierbei gilt es jedoch die folgenden Kriterien zu erfüllen:

- Kleiner als 1 MB, Muss ja via Mobilfunk übertragen werden
- Genug hohe Auflösung (min 800x480 Pixel)

8.6 UI-Design

8.6.1 Allgemein

Das UI-Design orientiert sich an den Design Guidelines von Android [and10], welche auf developer.android.com ersichtlich sind.

8.6.2 Organisation

Die verschiedenen Android-UI-Elemente sind entweder direkt als Java-Files im entsprechenden Package, oder als xml oder Bild im /res-Ordner gespeichert. Siehe <http://developer.android.com/guide/topics/resources/index.html>. Ein Dialog oder eine Activity kann beim Initialisieren das Layout aus einem XML-Markup-File laden. Somit ist für jede Activity auch ein entsprechendes XML-Markup-File definiert.

Typ	Speicherort	Bemerkung
Activity	Package <code>ch.hsr.eventassistant.ui.activity</code>	—
Dialog	Package <code>ch.hsr.eventassistant.ui.dialog</code>	—
Widgets	Package <code>ch.hsr.eventassistant.ui.widgets</code>	—
XML-Markup	Ordner: <code>/res/layout</code>	Je nach Auflösung in Ordnern mit Suffix (-hdpi, -ldpi, -mdpi)
Bilder	Ordner: <code>/res/drawable</code>	Je nach Auflösung in Ordnern mit Suffix (-hdpi, -ldpi, -mdpi)
Texte/Strings	Ordner: <code>/res/values</code>	Kann Internationalisiert werden mit Suffixen (-de, -en, etc.)

Tabelle (8.7) Organisation UI-Elemente

8.6.3 Beschreibung der Screens

Nachfolgend alle wichtigen Screens der Applikation, mit der jeweiligen Referenz zur Activity-Klasse oder dem XML, welches den Screen beschreibt.

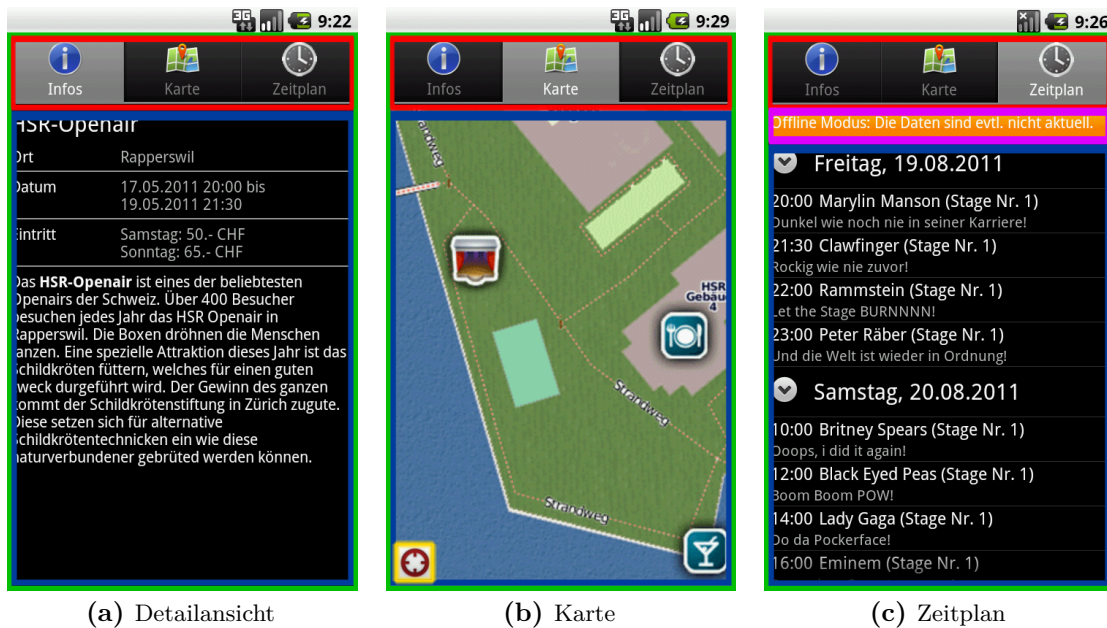


Abbildung (8.5) Screenshots Hauptscreens

Screen	Element	Definition in	XML-Markup
Allgemein	Hauptscreen	EventTabActivity	act_event_main.xml
	Tabs	EventTabActivity	(via Code)
	Notifikation	NetworkInfoView	inc_networkinfo.xml
	Tab-Container	EventTabActivity	act_event_main.xml
(a) Detailsansicht	Tab-Content	InfoTabActivity	tab_event_info.xml
(b) Karte	Tab-Content	MapTabActivity	tab_event_map.xml
	Map-Widget	MapView (Widget)	keines
	CenterMeButton	(ToggleButton)	keines
(c) Zeitplan	Tab-Content	MapView (Widget)	tab_event_schedule.xml
	Liste	ExtendableListView	(siehe Dialog)

Tabelle (8.8) Definitionen Hauptscreens

Die einzelnen Tabs sind hierbei als einzelne Activities realisiert, welche in den Tab-Container eingebunden werden.

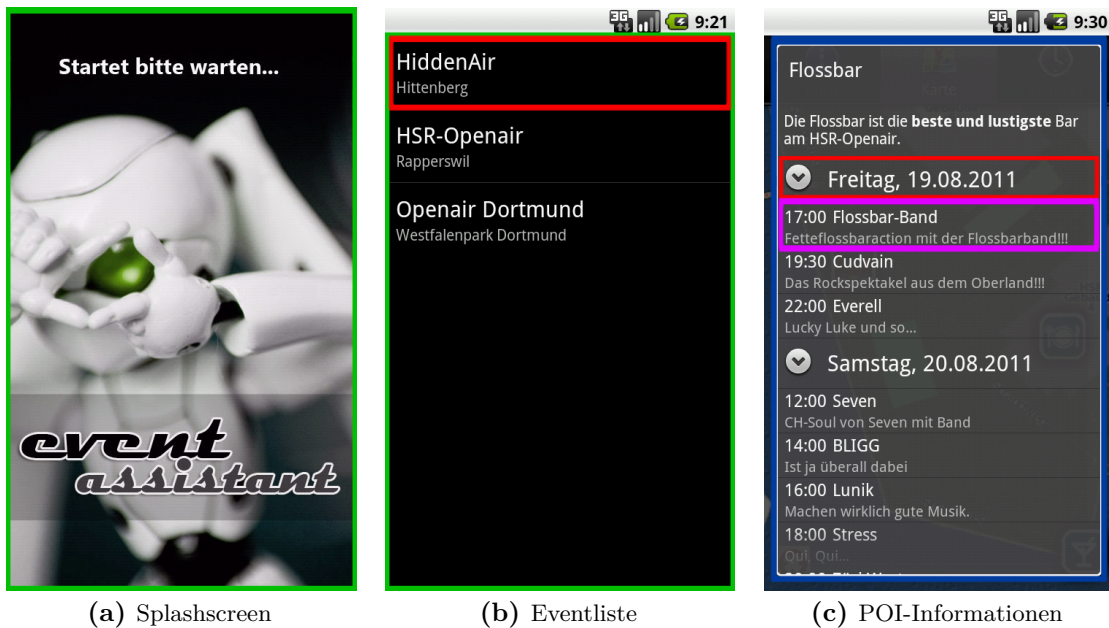


Abbildung (8.6) Screenshots Nebenscreens

Screen	Element	Definition in	XML-Markup
(a) Splashscreen	Splashscreen	StartupActivity	(Bild via Code)
(b) Eventliste	Hauptscreen	EventSelectActivity	atc_event_select.xml
	ListView	EventSelectActivity	atc_event_select.xml
	Eintrag	—	item_event_select.xml
(c) POI-Informationen	Dialog	EventDetailDialog	keines
	ListGroup	—	item_reminder_group.xml
	ListChild	—	item_reminder_child.xml

Tabelle (8.9) Definitionen Nebenscreens

Der POI-Informationen-Dialog ist unterteilt in einen Textbereich und einen Listenbereich. Der Listenbereich wird nur eingeblendet wenn für den POI auch „TimedEvents“ zur Verfügung stehen.

9 Implementation

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	12.12.2010	Felix Egli	Dokumentvorlage erstellt und erste Testfälle erfasst
0.2	18.12.2010	Felix Egli	Prototypen erfasst
0.3	18.12.2010	Felix Egli	Konzept Content Provider und Lokaler Cache erfasst
0.4	19.12.2010	Felix Egli	Konzept Content Provider und Lokaler Cache erweitert und ContentValues und Cursors erfasst
0.5	19.12.2010	Felix Egli	Referenz Server erfasst
0.8	20.12.2010	Michael Schnyder	Konzept für Datenklassen geschrieben und Prototypen erweitert
0.9	21.12.2010	Michael Schnyder	Widgets fertiggestellt, Tests in eigenes Kapitel verschoben
1.0	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

9.1 Prototypen

Nachfolgend werden alle Implementationsprototypen aufgelistet. GUI-Prototypen sind im Kapitel Usability Tests zu finden. (siehe *Usability Tests* auf Seite 96)

9.1.1 BrowserMapPrototype

Dieser Prototyp diente zur Evaluation des Karten-Controls. Es wurde getestet, welche Funktionalitäten das WebView-Widget besitzt und wie gut diese erweitert werden können. Ebenso wurde in diesem Projekt ein ContentProvider verwendet, um im WebView-Widget ein Bild aus dem lokalen Datenspeicher anzeigen zu können. (siehe *Variante „WebView (WebKit)“* auf Seite 21)

9.1.2 MapOverlayPrototype

Dieser Prototyp ist ebenfalls im Zusammenhang mit der Evaluation des Karten-Controls entstanden und diente zum Testen der Google Maps API, welche auf jedem Android Device zur Verfügung steht. Ziel war es, festzustellen ob sich dieses Widget zur Darstellung einer eigenen Karte eignet. Das Widget ist jedoch ungeeignet, da nur einzelne Punkte auf die Karte gezeichnet werden können. Sobald ganze Flächen oder Grafiken auf das Control gezeichnet werden, verhält es sich ungewohnt. Beispielsweise werden Grafiken erst nach dem Beenden eines Zoomvorgangs neu gezeichnet, womit dem Benutzer während dem Zoomen keine homogen zoomende Fläche präsentiert werden kann. (siehe *Variante „GoogleMapsAPI“* auf Seite 21)

9.1.3 ScrollViewPrototype

Diente zum Überprüfen, ob ein Bild mit Scrollviews gescrollt werden kann und hätte dann als Karten-Control verwendet werden können. Jedoch stellte sich mit diesem Prototyp heraus, dass unter Android das Scrollen nur in jeweils eine Richtung funktioniert. Auch das Zooming hätte noch separat implementiert werden müssen. Es existieren zu beiden Problemen Ansätze im Internet, jedoch scheiden dies hinsichtlich Kompatibilität und Art der Umsetzung aus.

9.1.4 LocationGsmCell

Dieser Prototyp fand vor Entwicklung und auch während der Entwicklung Anwendung. Vor der Entwicklung war das Ziel, festzustellen wie genau die GSM-Triangulation arbeitet. Zu diesem Zweck wurden alle vom Android-Betriebssystem erhaltenen Positionsangaben auf einer Google-Maps Karte eingezeichnet. Es zeigte sich, dass die Lokalisierung ziemlich genau arbeitete (Abweichung von 50-100m im HSR-Umfeld). Mit dieser Genauigkeit können wir in den meisten Fällen den Benutzer korrekt lokalisieren und den passenden Event direkt anzeigen. Während der Entwicklung konnten wir mit diesem Prototypen feststellen wie genau die Positionierung auf dem HSR-Gelände funktioniert und die Area für die Erkennung des HSR-Openairs noch etwas vergrössern.

9.1.5 CommunicationPrototype

Dieser Prototyp war dazu da, um die Kommunikation mit dem Server auszutesten. Hier wurde ein einfacher HTTP-Request an den Server geschickt und anschliessend das Element zu parsen. Dies hat vorzüglich geklappt und wurde auch dementsprechend umgesetzt.

9.1.6 JSONMarshaller

Hier wurde der *jsonmarshaller* getestet. Die Frage war hier, ob dieser in der Lage ist, die ganzen Objekte ohne grossen zusätzlichen Aufwand zu marshallen, das heisst in die JSON-Notation zu bringen. Das Umgekehrte (aus String -> Objekt) sollte auch auf der Androidseite möglich sein. Was aber fehlschlug, da diese Library über Reflection arbeitete. Denn auf der Dalvic VM in Android ist der Bytecode unterschiedlich und somit nicht zu gebrauchen. Hier wurde dann wie im CommunicationPrototype von Hand geparkt. Aber auf der Serverseite wird jedoch mit dem Marshaller gearbeitet.

9.1.7 MessageArchitecture

Hier wurde das Offlinekonzept ausprobiert welches genauer unter *Content Provider und Lokaler Cache* (siehe Seite 81) beschrieben ist. Was wir aus dem Prototyp gelernt haben und folglich nicht so umgesetzt haben:

Nach dem ansprechen des Content Providers leitet dieser, die Anfrage weiter an den DecisionMaker. Nun haben wir im Prototyp hier mit Java Objekten gearbeitet. Dies ist sehr Übersichtlich, jedoch das parsen abfüllen usw. ist sehr aufwändig. Somit haben wir uns für Cursors und ContentValues entschieden.

9.1.8 ServerServlet

Hier haben hier ein einfaches Java-Servlet genommen. Dann haben wir die URL mit dem es aufgerufen worden ist geparkt und entsprechend eine Java-Methode aufgerufen. Dies wurde so umgesetzt um eine möglichst einfache Serverimplementation zu gewährleisten.

9.2 Implementations-Konzepte

9.2.1 Datenklassen Vererbung

Es war uns wichtig, Datenklassen (DTO) auf allen Layern wiederverwenden zu können. Dazu haben wir eine Vererbungshierarchie entwickelt, welche es uns erlaubt, gemeinsame Basisklassen auf dem Client wie auch auf dem Server zu verwenden. (siehe *Server Klassendiagramm* auf Seite 85). Die Vererbungshierarchie verteilt sich dabei über 3 Stufen.

Stufe	Zusätzliche Fähigkeit	Bemerkung	Beispielklasse
1	_id-Feld ist definiert	_id wird bei allen Datenklassen vorausgesetzt	AbstractDto
2	Propertynamen sind definiert	Nur auf Client verwendet	Event
3	Felder sind definiert und mit Annotations bestückt für JSON-Marshallers	Nur auf Server verwendet	EventEntity

Tabelle (9.1) Objekthierarchie

9.2.1.1 Beispiel am Property „name“ eines Events

Stufe **2** (auf dem Client)

```

1 package ch.hsr.eventassistant.data;

public class Event extends AbstractDto {

5     public static final String NAME = "name";
    public static final String DATE_FROM = "dateFrom";
    ...

10    public static final String[] PROPERTYNAMES = { _ID, NAME,
        DATE_FROM, ... }

}
```

Stufe **3** (auf dem Server)

```

1 package ch.hsr.eventassistant.server.entity;

import ch.hsr.eventassistant.data.*;
...

5 @Entity
public class EventEntity extends Event {
```

```
10      @Value
      public int _id;

      @Value
      public String name;

15      @Value
      public String dateFrom;

      ...

20  }
```

- **Der Server** verwendet das Feld „name“ um Daten von der Datenbank im Objekt (Stufe **3**) zu speichern. Der JSON-Marshall (welcher das Objekt in JSON umsetzt), verwendet den Namen dieses Feldes (via Reflection und Annotations zur Kennzeichnung) um den Ausgabestring zu erzeugen.
- **Der Client** verwendet das Array PROPERTYNAMES mit den Einträgen aus den static-fields um die Werte der Spalten aus dem JSON-String auszulesen.
- **Das UI** (auf dem Client) verwendet die Datenklasse auf Stufe **2** um Kenntnis über die vorhandenen Spalten im Cursor zu erhalten. (siehe *ContentValues und Cursors* auf Seite 79)

Durch diese Aufteilung ergibt sich die Möglichkeit, bei der „fertig ausgebildeten Klasse“, d.h. Stufe **3** sehr einfach Tests zu erstellen. Diese Stellen dann sicher, dass alle Felder auch mit deren Namen als Feld erfasst sind und zudem auch die entsprechenden Properties vorhanden sind. (siehe *Datenklassen* auf Seite 94)

9.2.2 ContentValues und Cursors

Für dieses Kapitel wird vorausgesetzt, dass der Leser Kenntnisse über die Funktionsweise des Android ContentProvider's ¹, Android ContentValues² und Android Cursors³ besitzt.

Als vereinfachtes Beispiel ContentValues gleich normale HashMap und Cursor gleich Tabelle.

¹ <http://developer.android.com/guide/topics/providers/content-providers.html>

² <http://developer.android.com/reference/android/content/ContentValues.html>

³ <http://developer.android.com/reference/android/database/Cursor.html>

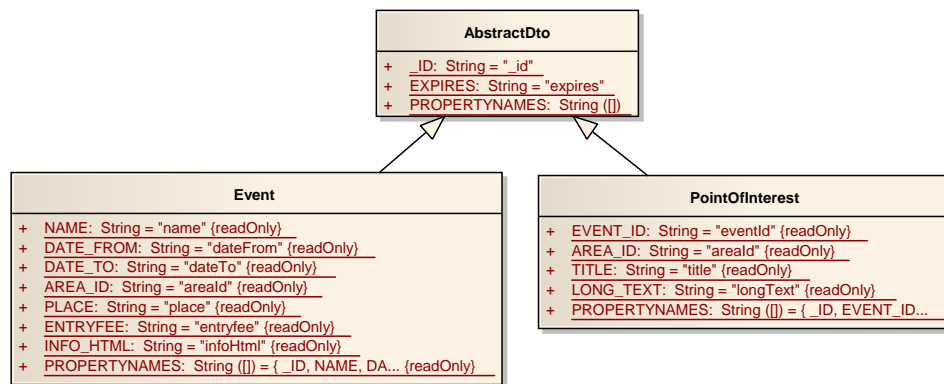


Abbildung (9.1) Beispiel für ContentValues generisch

Da wir die Funktionsaufrufe generisch erledigen wollten und nicht von Hand-Objekte abfüllen (Reflection funktioniert nur begrenzt), haben wir uns für die Verwendung von ContentValues¹ und Cursors² entschieden. Diese werden in Android für Datenbankabfragen, ContentProvider und deren Rückgabewerte verwendet. So kann ganz einfach beim einfügen der ContentValues über die z.B. *Event.PROPERTYNAMES* iteriert werden und alle Felder abgefüllt werden.

```

1  // Parsing the JSON Object to ContentValues
   ContentValues values = new ContentValues();
   for (String property : Event.PROPERTYNAMES) {
       values.put(property, jsonObject.getString(property));
5  }

```

Um diese dynamischen Daten anschliessend in einen Cursor zu verwandeln, haben wir den DynamicCursor erstellt, welcher von AbstractCursor abgeleitet ist und als Cursor fungieren kann. Dieser wird Schlussendlich auch an das GUI über den ContentProvider weitergegeben.

```

1  ...
   String jo = getRemoteJSONString(getServerUrl() + eventUrl);
   ArrayList<ContentValues> eventlist =
       parseToContentValuesList(jo, EventListItem.PROPERTYNAMES);
5  return new DynamicCursor(eventlist);
   ...

```

1 <http://developer.android.com/reference/android/content/ContentValues.html>

2 <http://developer.android.com/reference/android/database/Cursor.html>

9.2.3 Content Provider und Lokaler Cache

Für dieses Kapitel wird vorausgesetzt, dass der Leser Kenntnisse über die Funktionsweise des Android ContentProvider's¹ besitzt.

Zum groben Verständnis des *SSD*² bitte *Klassendiagramm* (siehe Seite 62) beachten.

9.2.3.1 Ablauf - SSD

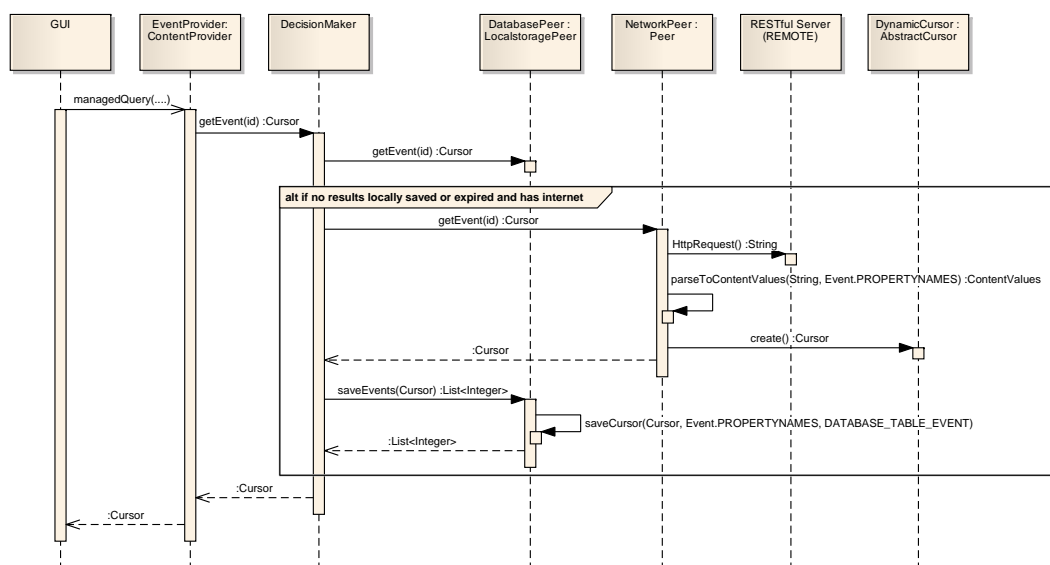


Abbildung (9.2) Cache Algorithmus

Erläuterung des Ablaufs Im EventProvider wird beim Aufruf der Methode *query* überprüft, ob die aufgerufene URI mit einer der bereits registrierten matcht. Falls dies der Fall ist fragt er die übermittelte Id ab und leitet sie an die Methode *getEvent(id)* weiter.

Der DecisionMaker fragt als erstes seine Instanz der LocalstoragePeer (in diesem Fall bei der Erstellung der LocalstoragePeer wurde eine DatabasePeer übergeben) an ob der angefragte Event bereits vorhanden ist.

¹ <http://developer.android.com/guide/topics/providers/content-providers.html>

² *System Sequenz Diagramm*

Fall - Daten nicht vorhanden

Falls der DatabasePeer keine Daten im Cursor zurückliefert und Verbindung zum Internet besteht, holt der DecisionMaker sich die aktuellen Daten über die NetworkPeer. Diese ruft einen HTTP-Request auf der gegebenen URL auf. In unserem Beispiel <http://eventassistant.dyndns.org/EventAssistant-Server/event/1> und parst die empfangenen Daten in eine Liste von ContentValues. Diese Liste mit den ContentValues kann mit dem DynamicCursor als Cursor zurückgegeben werden. Anschliessend wird der zurückgegebene Cursor in der LocalStoragePeer gespeichert und an den EventProvider geliefert, welcher auch einen Cursor zurückgeben muss.

Fall - vorhanden aber Daten expired

Falls hier die Internetverbindung vorhanden ist, wird hier weiter verfahren wie oben *Fall - Daten nicht vorhanden*

Fall - vorhanden und aktuell

Wenn die Daten nicht expired sind, werden einfach der Cursor der aus der Datenbank gelesen ist zurückgeschrieben.

Überprüfen ob expired

Da im AbstractDTO alle das Feld *ID* und *EXPIRED* besitzen, kann generisch in jedem Cursor auf das Feld *EXPIRED* prüfen ob dies verfallen ist. Dieses Feld wird vor dem *jsonmarshaller* auf dem Server gesetzt und übermittelt.

9.2.3.2 Offline-Notification

Der Benutzer kann bei diesem Entscheidungsprozess nicht einbezogen werden, da die entsprechenden Routinen vom UI losgelöst sind. Dazu wurde das NetworkInfoView-Widget (siehe *NetworkInfoView* auf Seite 91) entwickelt.

9.2.4 Kommunikationsprotokoll

Das Kommunikationsprotokoll und Formate (siehe *Event-Assistant Server (REST-Schnittstelle)* auf Seite 65) sowie der Zugriff auf den ContentProvider (siehe *Schnittstelle zu Content-Provider* auf Seite 63) ist im vorherigen Kapitel beschrieben. (siehe *Schnittstellen und Formate* auf Seite 63)

9.2.5 Referenz-Server

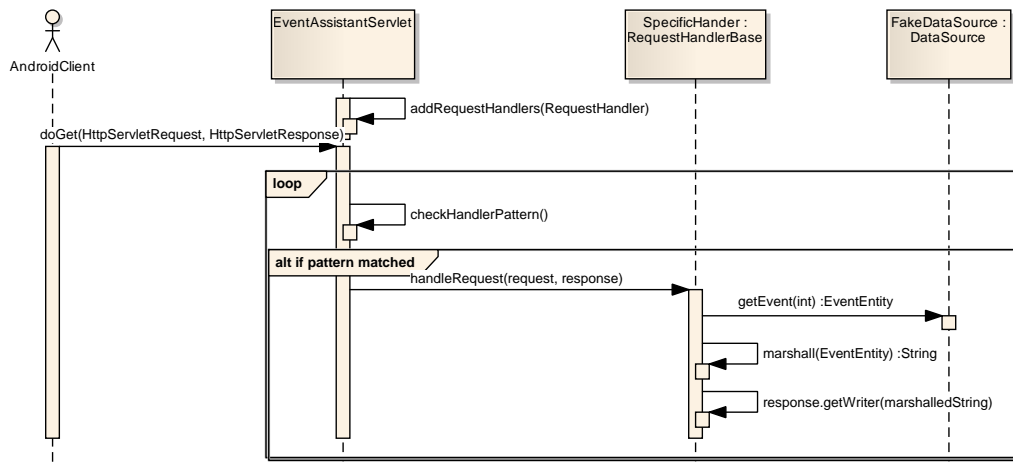


Abbildung (9.3) Server SSD

Hier handelt es sich um ein einfaches Servlet in einem Tomcatserver welches die `doGet`-Methode abfängt und seine eigene `RequestHandler` anhängt. Diese sind von der `RequestHandlerBase` abgeleitet. Diese `RequestHandler` registrieren sich am Servlet über eines oder mehrere `Regex-Patterns`. Das Servlet übernimmt das Matching der eingehenden Requests auf die einzelnen Handler und leitet die Abrufe an die Handler weiter, sofern möglich.



Abbildung (9.4) Server 404

Die RequestHandlerBase verfügt über die DataSource, in unserem Fall ist das die FakeDatabase. Die FakeDatabase ist sehr einfach gehalten, sie erzeugt einfach die Entitäten im Code. Anschliessend kann über die in der DataSource vorgegebenen Methoden die Anfrage gestartet werden.

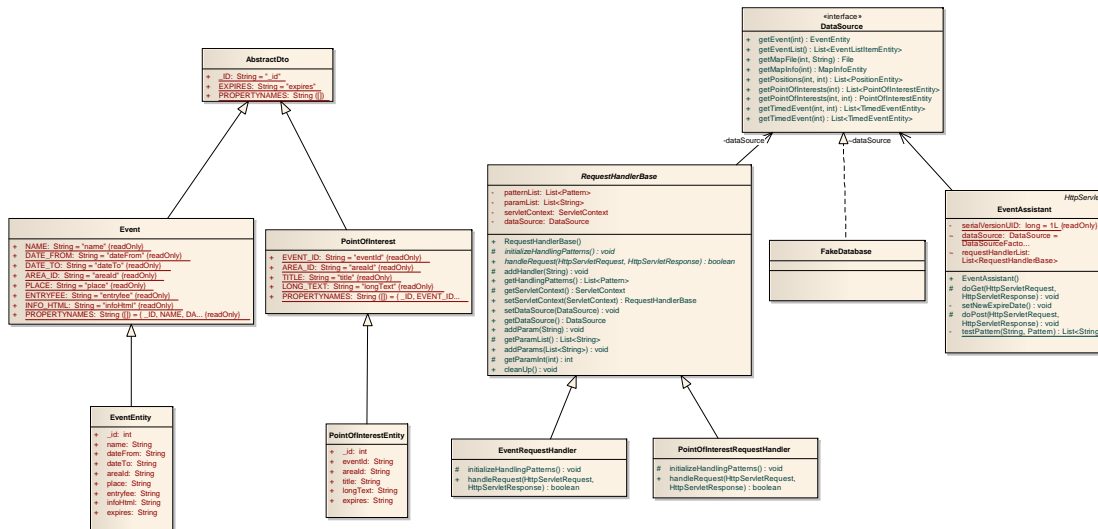


Abbildung (9.5) Server Klassendiagramm

Auf der Abbildung ist gut zu erkennen, dass die Entity's von den im Common erfassten Klassen abgeleitet werden. Diese Entities sind nötig, um das Marshalling durchzuführen. Dies wird durch Annotieren der Attribute erreicht. Wie man nun sieht, ist es ein leichtes das Interface `DataSource` zu implementieren und anstatt eine `FakeDatabase`, eine richtige Datenbank dahinter zu hängen.

9.3 Widgets und UI-Helper

Die folgenden Widgets und UI-Helper wurden speziell für diese Applikation entwickelt. Die Wiederverwendbarkeit ist dadurch gegeben, dass die Widgets in ihrem jeweiligen Package von keinen anderen UI-Packages abhängig sind, ausser generellen Hilfsklassen.

9.3.1 MapView-Widget

Das MapWidget ist für die Interaktion des Benutzers mit der Karte und den drauf enthaltenen Elementen zuständig. Diese Klasse leitet von der WebView-Klasse ¹ ab und erweitert diese um die folgenden Funktionen:

- Georeferenzierung
- Positionsdarstellung
- Interaktive Regionen (POI)
- Fehlerbereinigte/Zusätzliche Events

9.3.1.1 Georeferenzierung

Die von der WebView darstellbaren Bilder lassen sich mit dieser Erweiterung mit zwei Positionsangaben versehen. Dadurch kann jeder Pixel eines Bildes einer Positionsangabe zugeordnet werden und umgekehrt. Die Umrechnung zwischen Positionsangaben und Pixeln wird von der MapHelper-Klasse erledigt und diese wiederum wird nur von der MapView verwendet. Für die Initialisierung der Grafik als Karte sind die folgenden Zeilen Code notwendig:

```
1  ...  
   // get MapView-Widget from View  
   mapView = (MapView) findViewById(R.id.mapview);  
  
5  // first reference point is at the top left border  
   Point point1 = new Point(0, 0);  
   MapPosition loc1 = new MapPosition(47.22416, 8.81448);  
  
   // second reference point is at the bottom right border  
10  MapPosition loc2 = new MapPosition(47.222097, 8.81931);  
   Point point2 = new Point(641, 1000);  
  
   // initialize the mapView  
   mapView.setReferencePoints(new MapReferencePoint(loc1, point1),
```

1 <http://developer.android.com/reference/android/webkit/WebView.html>

```
15      new MapReferencePoint(loc2, point2));  
  
    mapView.setMapWidth(1000);  
    mapView.setMapHeight(641);  
    ...  
20 }
```

Hierbei wird die Klasse `MapReferencePoint` verwendet, welches jeweils eine Positionsangabe und Pixel-Paar zusammenfasst. Die Georeferenzierung von Bildern birgt allein noch keinen Zusatznutzen, doch diese wird von den zwei folgenden Funktionen intensiv verwendet.

9.3.1.2 Positionsdarstellung

Das Widget kann eine einzelne Positionsangabe auf der Karte mit einem roten Punkt darstellen (Methode `setCurrentLocation(MapPosition pos)`). Falls gewünscht, kann dieser mit der Methode `scrollToLocation(MapPosition pos)` auf dem Bild zentriert werden. Die Karte wird dabei jedoch nicht über die Ränder hinaus verschoben. Es kann bei grosser Zoomstufe daher schon sein, dass der Punkt nicht zentriert werden kann.

9.3.1.3 Interaktive Regionen

Einer Karte innerhalb einer `MapView` können beliebig viele interaktive Bereich hinzugefügt werden. Aktuell werden diese für POI verwendet. Eine interaktiver Bereich besteht hierbei aus einer Liste von mindestens drei Positionsangaben und einer Reihenfolge. Aus diesen Informationen wird ein Polygon erzeugt und verwendet um bei Tap-Events zu erkennen, ob der Benutzer in einen interaktiven Bereich getapt hat. Diese ganze Funktionalität ist in der Klasse `MapArea` gekapselt. Der `OnTap`-Handler wird mit dem `TapEventListener`-Interface realisiert.

```
1      ...  
    // get MapView-Widget from View  
    mapView = (MapView)findViewById(R.id.mapview);  
  
5      MapArea ma = new MapArea();  
  
    // add positions clockwise, starting top left  
    ma.addMapPosition(1, new MapPosition(47.222514, 8.816188);  
    ma.addMapPosition(1, new MapPosition(47.222514, 8.816800);  
10   ma.addMapPosition(1, new MapPosition(47.222135, 8.816800);  
    ma.addMapPosition(1, new MapPosition(47.222135, 8.816188);  
  
    ma.setOnTapListener(new TapEventListener() {  
        // do some fancy stuff when event is raised  
15        System.out.println("Here we are!");  
    });
```

```
20 // Add the interactive Area to the mapview  
    mapView.addMapArea (ma) ;  
    ...
```

9.3.1.4 Fehlerbereinigte Events

Der `OnTapEvent` existiert auf dem `WebView` nicht. Für die Weiterleitung an eine interaktive Region war es deshalb notwendig diese Funktionalität selber nachzubauen. Für das UI war es zudem wichtig sich an einem verlässlichen `PageComplete`-Event anhängen zu können. Nur so konnte während der ganzen Ladezeit ein `ProgressDialog` angezeigt werden. Registrieren für die zwei zusätzlichen Events kann man sich mit den folgenden Methoden:

- `setOnTapListener(TapEventListener listener)` Für den Tap-Event
- `setPageLoadedListener(OnPageLoadedEventListener listener)` Für Event, wenn die Seite/Bild komplett geladen und angezeigt wird.

9.3.2 ExpandableListCursorAdapter

Der `ExpandableListCursorAdapter` stellt Werte aus einem `Cursor` in einer `Expandable List` dar. Dazu leitet er von der abstrakten Klasse `BaseExpandableListAdapter`¹ ab und implementiert die dafür vorgesehenen Methoden [SYH10]. Um mit diesem Adapter eine `ExpandableList` zu erstellen, werden die folgenden Komponenten benötigt:

- `ExpandableList` (in XML deklariert)
- `Cursor`
- XML-Markup für Gruppen-Eintrag (als Android-Ressource)
- XML-Markup für Child-Eintrag (als Android-Ressource)
- `OnViewCreatedCallbackHandler`
- `ValueBinder` für Gruppierung
- `ValueBinder` für Einträge
- Adapter-Instanz (in Code)

Nachfolgend genauere Informationen zu den oben genannten Punkten.

1 <http://developer.android.com/reference/android/widget/BaseExpandableListAdapter.html>

9.3.2.1 XML-Markup für Gruppen- und Child-Einträge

Es muss definiert werden, wie die Gruppen- und Child-Einträge aus sehen sollen. Dies geschieht in XML, auf welches zur Runtime mittels Android-RessourceId zugegriffen werden kann. Beispiel eines Gruppen-Eintrag-XMLs:

```
1  <?xml version="1.0" encoding="utf-8"?>
   <LinearLayout android:id="@+id/LinearLayout01"
       android:layout_width="fill_parent" android:layout_height="45dip"
       xmlns:android="http://schemas.android.com/apk/res/android" >
5
   <TextView xmlns:android="http://schemas.android.com/apk/res/android"
       android:id="@+id/title"
       android:layout_width="fill_parent"
       android:layout_height="45dip"
10      android:textSize="20sp"
       android:textColor="#FFF"
       android:gravity="left|center_vertical"
       android:paddingLeft="45dip"
       />
15 </LinearLayout>
```

Zu beachten ist hier speziell die Zeile 7: Dort wird Android angewiesen, auch dem TextView eine Id zuzuweisen. Dadurch kann nach dem Rendern wiederum auf dieses Element zugegriffen werden, wenn die Id (in diesem Fall "title") bekannt ist.

9.3.2.2 OnViewCreatedCallbackHandler

Der ExpandableListCursorAdapter erzeugt nach dem Zuweisen eines Cursors automatisch die Gruppen- und Child-Einträge. Je Gruppe und je Eintrag für eine View auf Basis der angegebene XML-Markups erzeugt. Mit diesem Callback-Handler wird dem Entwickler die Möglichkeit gegeben, nach dem Erstellen der Views aus dem XML einmalig per Code auf diese zuzugreifen. Dies kann dazu verwendet werden, EventHandler an Einträge der Views wie z.B. Buttons anzuhängen. Beispiel (Annahme: Auf der View für die Child Einträge existiert ein Button welcher die Id „myButton“¹ besitzt):

```
1  ...
   ExpandableListCursorAdapter adapter = ExpandableListCursorAdapter(...);
   adapter.setOnViewCreatedCallback(new OnViewCreatedCallbackHandler() {
5
       public void created(int layoutId, View view, ViewGroup parent,
           ContentValues data) {
```

¹ <http://developer.android.com/reference/android/content/res/Resources.html>

```

10         Button but = view.findViewById(R.id.myButton);
            but.setOnClickListener(...);
    }
};

```

9.3.2.3 ValueBinder für Gruppierung

Der `ExpandableListCursorAdapter` gruppiert die Einträge nicht automatisch. Auch kann dem Adapter kein Feld übergeben werden, nach welchem die Gruppierung der Daten durchgeführt wird. Vielmehr erwartet der Adapter eine Anweisung, welche Spalte des Cursors mit welchem `ValueConverter` konvertiert werden muss. Erst dieser konvertierte Wert wird verwendet um die Daten zu Gruppieren. Das ergibt den Vorteil, dass die Gruppierungslogik komplett vom Adapter getrennt ist und vom Entwickler selber implementiert werden kann. Folgender Ausschnitt beschreibt die Definition eines `SimpleValueBinder` (implementiert das Interface `ExpandableListCursorAdapter.ValueBinder`) welcher für die Bestimmung der Gruppen verwendet wird.

```

1    ...
    ValueBinder vb = new SimpleValueBinder(
        TimedEvent.DATETIME_FROM, // Source Column
        ExpandableListCursorAdapter.dateTimeToShortConverter
5    );
    ...

```

Dieser `ValueBinder` definiert das Binding der Spalte „`TimedEvent.DATETIME_FROM`“ (siehe *Datenklassen Vererbung* auf Seite 78) und (siehe *ContentValues und Cursors* auf Seite 79) mit dem `ValueConverter` „`ExpandableListCursorAdapter.dateTimeToShortConverter`“. Der `ExpandableListCursorAdapter` kann mit Hilfe dieser Definition bei jedem Eintrag die Spalte „`DATETIME_FROM`“ mit dem „`dateTimeToShortConverter`“ konvertieren und so die Gruppierung bestimmen.

9.3.2.4 ValueBinder für Einträge

Funktioniert analog der `ValueBinder`-Definition für die Gruppierung. Jedoch wird hier zusätzlich noch angegeben in welches Widget in der Gruppen-oder Child-View der Wert der Spalte geschrieben werden muss. Beispiel (wir möchten wiederum den Wert der Spalte „`TimedEvent.DATETIME_FROM`“ mit dem „`ExpandableListCursorAdapter.dateTimeToTimeConverter`“ konvertieren und danach als Text dem `TextView` mit der Id „`titletime`“ zuweisen):

```

1    ...

```



```
ValueBinder vb = new SimpleValueBinder(  
    TimedEvent.DATETIME_FROM, // Source Column  
    R.id.titletime,           // Target Widget Id  
    ExpandableListCursorAdapter.dateTimeToTimeConverter  
);  
...
```

9.3.2.5 Adapter Instanz

Alle oben genannten Einstellungen (mit Ausnahme des `OnViewCreatedCallbackHandler`) können direkt im Konstruktor übergeben werden. Folgende Argumente werden benötigt:

- **Context context** Your Context
- **Cursor c** Cursor for your Data
- **int groupLayout** Identifier to the Layout for the Group Item
- **int itemLayout** Identifier to the Layout for the Child Item
- **ValueBinder groupByColumnBinder** Defines a Binding for the Groupnames
- **ValueBinder[] groupBinder** Array of Bindings for the group-item
- **ValueBinder[] childBinder** Array of Bindings for the child-item

9.3.3 NetworkInfoView

Um dem Benutzer einen Anhaltspunkt über die Aktualität der angezeigten Daten zu geben, wurde dieses Widget entwickelt. Dieses basiert auf einer `TextView` und kann den Internet-Verbindungsstatus anzeigen. Alle 10 Sekunden prüft das Widget selbständig, ob eine Internetverbindung besteht und ruft bei einer Änderung entsprechende Callbacks auf. Auf die Callbacks können sich wiederum einzelne Views registrieren und bei Bedarf das Widget ein- oder ausblenden. Bei eingeblendetem Zustand ist jeweils der Text aktiv, welcher über den aktiven Offline Modus informiert. (siehe *Screenshots Hauptscreens* auf Seite 73)

9.3.4 I18nDialogHelper

Dies ist eine Klasse, welche die Anzeige eines Dialogs ermöglicht, bei welchem der Titel und die Nachricht direkt aus dem Ressource-File geladen wird. Die Handhabung des Dialogs ist danach identisch eines jenen, welcher direkt erzeugt wurde.

10 Testing

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	19.12.2010	Felix Egli	Unittests erfasst, Usabilitytest erfasst
0.2	20.12.2010	Felix Egli	Kapitel Test überarbeitet und ergänzt
0.3	21.12.2010	Michael Schnyder	Tests in eigenes Kapitel verschoben
0.4	22.12.2010	Felix Egli	Unittests Screenshots eingefügt
0.5	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

10.1 Unit-Tests

10.1.1 Android

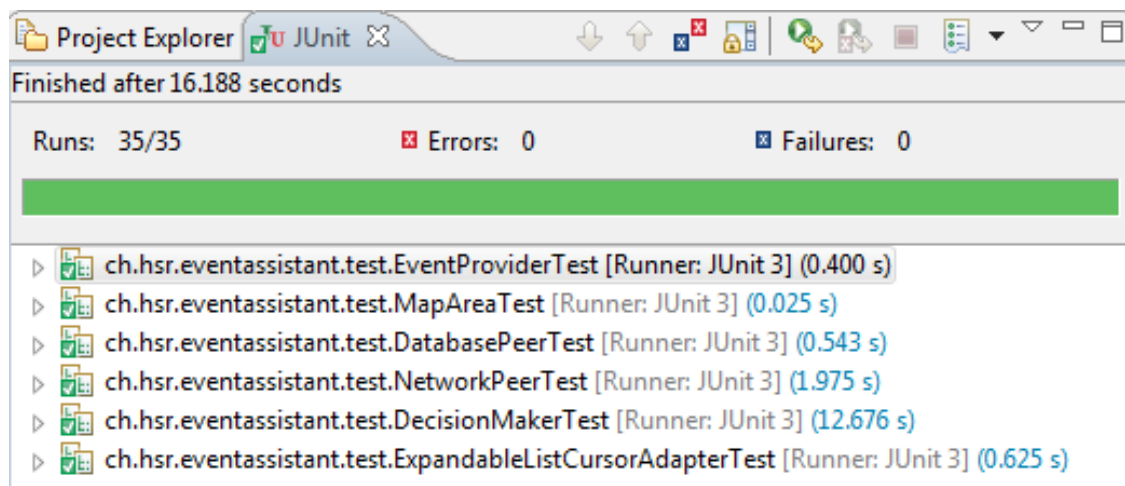


Abbildung (10.1) Unittests ausgeführt auf Android-Testprojekt

Alle Android Testfälle sind im Eclipse-Projekt EventAssistant-Test untergebracht. Dies ist ein spezielles von Android vorgegebenes Projekt, welches auf das zu instrumentierende Projekt verlinkt werden muss. Die folgenden Tests sind abhängig von einem Android-Context. Damit dieser verfügbar ist, muss von den über die vorgegebenen Testklassen von Android (ProviderTestCase2, AndroidTestCase usw.) abgeleitet werden. Folgende Tests werden genauer erläutert.

DatabasePeerTest In diesem Test wird eine Instanz des DatabasePeers getestet. Haupt-

sächlich die CRUD Anwendungen.

NetworkPeerTest In diesem Test wird eine Instanz des NetworkPeers getestet. Hier wird direkt auf den Remoteserver zugegriffen, dies dient der Prüfung ob Daten empfangen wurden und ob sich der Server auf dem aktuellen Stand befindet.

DecisionMakerTest Hier wird der DecisionMaker getestet, ob er die Online/Offline Logik korrekt implementiert hat.

EventProviderTest Hier wird der EventProvider getestet, ob er auf die entsprechenden URL's die korrekten Objekte liefert. (siehe *Content Provider und Lokaler Cache* auf Seite 81)

ExpandableListCursorAdapterTest Dieser testet, ob die Daten vom Cursor korrekt in Gruppen und Untereinträge aufgeteilt werden.

MapAreaTest Hier wird der Algorithmus der MapArea getestet. Ob sich ein Objekt innerhalb oder ausserhalb der Area befindet.

10.1.2 Datenklassen

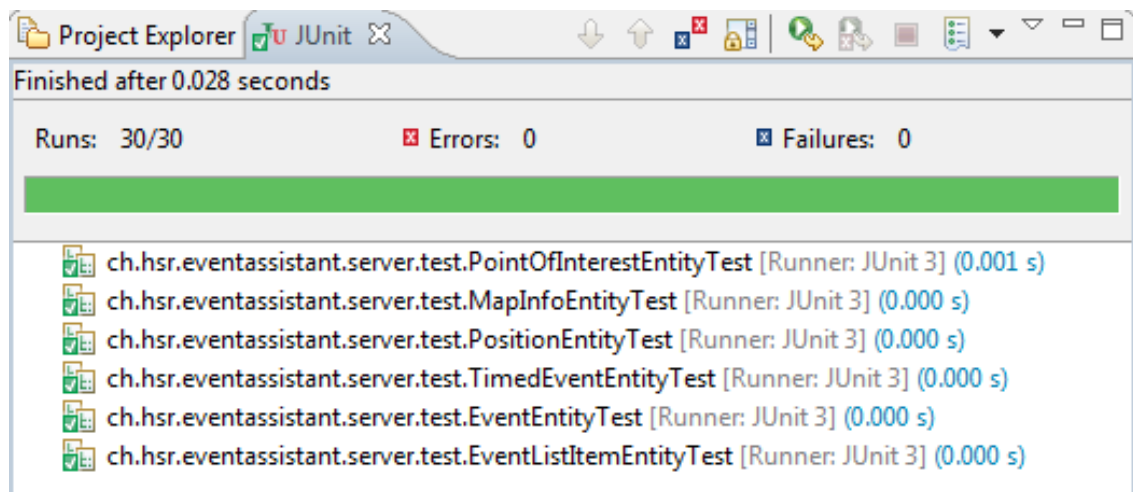


Abbildung (10.2) Unittests ausgeführt der Datenklassen

Im Package *ch.hsr.eventassistant.server.test* (Serverprojekt) sind die generischen Tests welche die Attribute in der Entitys gegenüber der statischen Properties testet. Zudem wird getestet ob alle statischen Properties gegenüber den Entity-Attributen vorhanden sind.

Dies kann ganz einfach durch den generischen-EntityTester erzwungen werden, welcher

folgendermassen erzeugt wird `EntityTester<EventEntity>`.

10.1.3 Database-Fake-Test

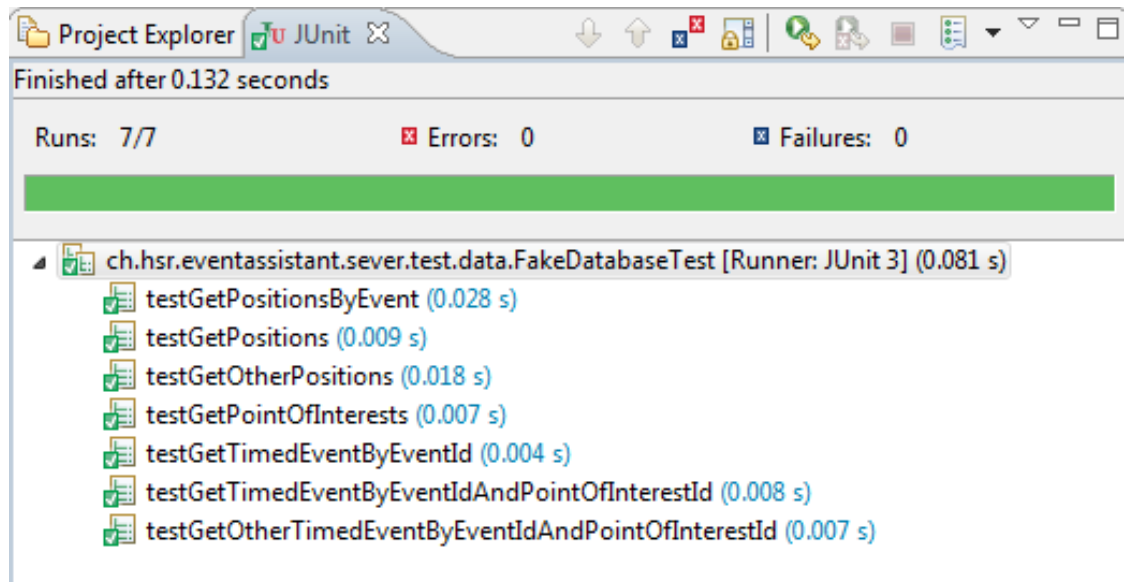


Abbildung (10.3) Unittests ausgeführt der FakeDatabase

Im Package `ch.hsr.eventassistant.sever.test.data` (Serverprojekt) befindet sich der `FakeDatabaseTest`, welcher testet, ob die Datenbanklogik korrekt implementiert ist. Der Vorteil, Indirektion über HTTP fällt weg.

10.1.4 Android-Testabdeckung

Wir haben versucht die Android Testabdeckung zu machen, was uns leider nicht gelang. Es sind nur etwa drei bis vier Google-Mailinglists über dieses Thema aktiv. Nach etwa einem Tag Research haben wir das Unterfangen aufgegeben. (siehe *Android Coverage Report* auf Seite 181)

10.2 Systemtests

Systemtests werden anhand der Systemtest Vorlage ausgefüllt. Diese Vorlage, welche auch die Testfälle enthält wird laufend aktualisiert. (siehe *Systemtest Vorlage* auf Seite 185) Zur besseren Übersichtlichkeit müsste die Vorlage noch Kategorisiert und den Use Cases zugeordnet werden. Auf welche des kleinen Projektumfanges verzichtet wurde.

10.2.1 Systemtest vom 3.12.2010

Hier wurde leider kein Protokoll geführt, dieser wurde während der Entwicklung durchgeführt. Die Testfälle orientieren sich an den Testfällen vom 19.12.2010. Am selben Tag wurde auch der Usability-Check Test durchgeführt. (siehe *Spezifischer POI's von Event* auf Seite 100)

10.2.2 Systemtest vom 19.12.2010

Systemtest Protokoll vom 19.12.2010 (siehe Seite 189) Es wurde zudem nur mit dem mobilen Handynetzt getestet. Hier gibt es noch einige Erläuterungen zu machen.

1. Alle Testfälle wurden erfüllt. Trotzdem können natürlich Probleme auftreten.
2. Die Lokalisierung über *GSM* ist teilweise sehr ungenau. Dies hat zur Folge, dass nicht immer automatisch der korrekte Event vorselektiert werden kann und der Benutzer aus der Liste den richtigen Event selbst auswählen muss.
3. Offlinemodus wird meist erst verzögert aufgehoben. Dies ist aber von Gerät zu Gerät unterschiedlich.
4. Die Verbindung über *WLAN* kann plötzlich unterbrochen werden und es wird Offlinemodus (obwohl Handynetzt verfügbar ist) angezeigt. Beim neuverbinden an das *WLAN* wird diese Meldung dann aufgehoben. Dies liegt wahrscheinlich am HSR-WLAN, welches Android-Geräte öfters Probleme bereitet.

10.3 Usability Tests

10.3.1 Papierprototyp Nr. 1

10.3.1.1 Aussehen

Die Skizzen des Papierprototyp sind hier ersichtlich (siehe *Papierprototype Nr. 1* auf Seite 177).

10.3.1.2 Vorgehen

Es wurden einige grobe Skizzen vom einem mobilen *GUI* angefertigt, wie es aussehen könnte. Anschliessend wurden es in einer gemeinsamen Sitzung (m2, fx) die Vor-/Nachteile einer solchen Lösung diskutiert.

10.3.1.3 Resultat

Punkte die festgestellt wurden, auf was geachtet werden muss. Welche im nächsten Prototyp ausgearbeitet wird.

- Wann wird die Menütaste wie verwendet.
- Wie werden die Settings eingebaut.
- Wie sollen die Items auf der Karte sichtbar sein.

10.3.1.4 Redesignentscheide

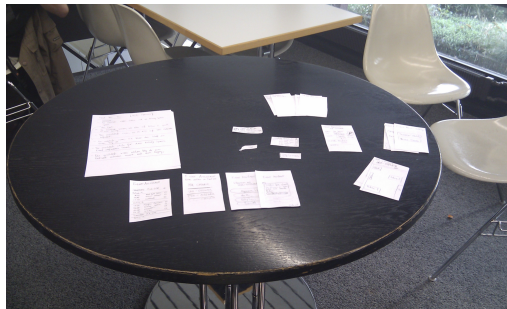
Sind direkt in Papierprotoyp Nr. 2 eingeflossen.

10.3.2 Papierprotoyp Nr. 2

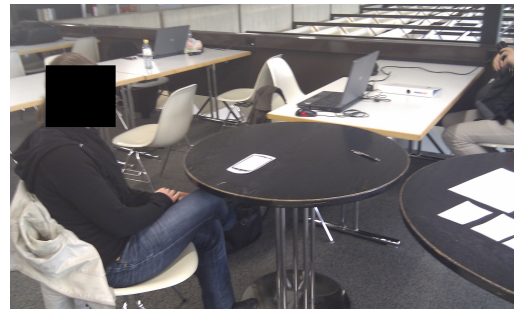
10.3.2.1 Aussehen



Abbildung (10.4) Papierhandy

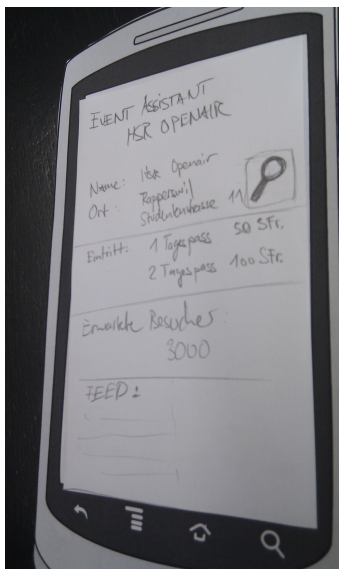


(a) Menuauslage

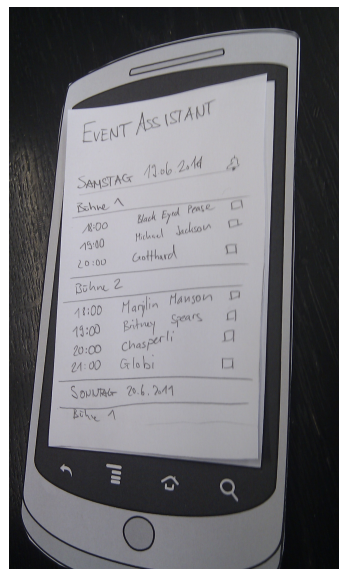


(b) Testperson

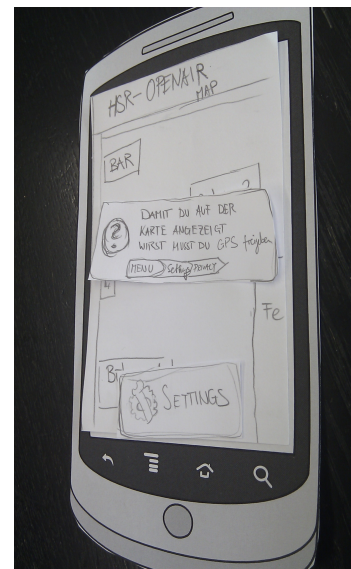
Abbildung (10.5) Testumgebung



(a) Eventinfo



(b) Minievent



(c) Karte

Abbildung (10.6) Beispielscreens

Testvideo: Es wurde zusätzlich noch ein Video aufgezeichnet, welches sich auf der beiliegenden CD befindet.

10.3.2.2 Vorgehen

Es wurden verschiedene Benutzer aus verschiedenen Handynutzergruppen ausgewählt.

- iPhone Benutzer (2 Tester)
- Android Benutzer (1 Tester)

- Standard Mobiltelefon Benutzer (1 Tester)

Diese wurden alle mit folgenden sechs Aufgaben konfrontiert.

- Du bist zuhause und möchtest wissen, wann Britney Spears am Wochenende am HSR-Openair ihren Auftritt hat.
- Du bist zuhause und möchtest wissen, wie viel Geld du für einen Eintagespass bezahlen musst.
- Ab jetzt befindest dich auf dem Festgelände und möchtest nun direkt wissen wer Heute (Samstag) und eventuell auch Morgen Sonntag auf der Bühne 2 spielt.
- Nun möchtest du wissen wo dass du dich auf dem Gelände befindest.
- Nun möchtest du dich für den Gig von Britney Spears am Samstag abend notifizieren lassen, so dass du 15 Minuten vor dem Event daran erinnert wirst.
- Nun möchtest du wissen welchen Weg du vom aktuellen Standort zum Ausgang nehmen musst.

Diese Aufgaben mussten die Testpersonen an einem Papierhandy entsprechend lösen.

10.3.2.3 Resultat

Allgemein ist zu sagen, dass die Benutzer sehr gut mit der erstellten Lösung zurecht kamen. Sie fanden sich im gegebenen Menü schnell zurecht und konnten die Aufgaben lösen.

Fragen, offene Punkte:

- Veranstaltungskarte in Karte umbenennen
- Das ständige wechseln zurück ins Aufzählmenu mühsam (-> Register einsetzen)
- Wo wurde die Notifikation gespeichert? Wo kann diese abgerufen werden?
- Wo auf der Karte ist Norden? In welche Richtung schaue ich? Kompass einbauen!
- Das Glocke auf der MiniEventliste war verwirrend. Der Nutzen war erst nach dem ersten Klick ersichtlich.
- Was passiert wenn ich auf die Notifikation klicke, also zur Aktion oben in der Liste
- Info beim ersten Start anzeigen seien Teilweise mühsam.
- Zoomen wäre schön auf der Karte
- Auf die Frage wo stehe ich würde Handy nicht verwendet, denn man sei selbst genügend gut um anhand von Gebäuden den eigenen Standort zu bestimmen.

10.3.2.4 Redesignentscheide

- Tabs einbauen, vereinfacht die Navigation
- Notifikation anhand Häckchen einbelden.
- Kompass oder Drehbarer Mann einbauen
- Tooltip auf Glocke
- Wenn Notifikation gesetzt wird, erscheint eine kurze Statusmeldung, dass die Notifikation für den Event gespeichert wurde.
- Zoomen für v.2 vorgemerkt.
- GPS automatisch per Ja/Nein Dialog beim Start der Karte aktivieren falls gewünscht und verfügbar.

10.3.2.5 Weiteres Vorgehen

Es wird beim nächsten Mal an einem technischen interaktiven Prototypen getestet.

10.3.3 Usability-Check Test vom 3.12.2010

Ziel Erkennen und Ausmerzen der letzten kleinen grafischen Unschönheiten.

10.3.3.1 Vorgehen

Drei verschiedenen Testpersonen wurde wie beim Paperprotoyp Nr. 2 die lauffähige Version auf dem HTC Desire mit Android 2.2 zum Test gegeben. Alternativer Test wurde mit dem Desire HD mit Android 2.2 getätigt. Unterschiede zudem einzelnen Mobiltelefonen waren kaum auszumachen.

10.3.3.2 Redesignentscheide

- Zu lange Responsiveness war grösser 1s -> AsyncTask erstellen
- Offline Modus -> Wenn kein Netz verfügbar, dies grafisch darstellen
- Zu komplizierte Einstellungen -> nur Frage GPS einstellen oder nicht, wenn ausgeschaltet
- Scrollbereich beim Zeitplan nicht intuitiv ersichtlich (zusätzliche Infos weiter unten, wie weiss das der Benutzer)
- Automatische Zentrierung ausschalten (Nervt, wenn auf Karte navigiert werden muss)

- Event-Startscreen hat Event-Datum nicht. -> Datum einfügen

10.3.4 Usability-End Test

10.3.4.1 Aussehen



(a) Test zu Hause



(b) Test zu Hause 2

Abbildung (10.7) Tests zu Hause



(a) Test am Openair



(b) Testtelefon

Abbildung (10.8) Testumgebung und Personen

10.3.4.2 Vorgehen

Es wurden verschiedene Benutzer aus verschiedenen Handynutzergruppen ausgewählt.

- iPhone Benutzer (1 Tester)
- Standard Mobiltelefon Benutzer (1 Tester)

Diese wurden alle mit folgenden fünf Aufgaben konfrontiert. Der Test fand teilweise im Freien statt.

1. Du bist zu Hause und möchtest wissen, wann Britney Spears am Wochenende am HSR-Openair ihren Auftritt hat.
2. Du bist zu Hause und möchtest wissen, wie viel Geld du für einen Eintagespass bezahlen musst.
3. Ab jetzt befindest dich auf dem Festgelände und möchtest nun direkt wissen wer Heute (Samstag) auf der „Stage Nr. 1“ spielt.
4. Nun möchtest du wissen wo du dich auf dem Gelände befindest.

5. Nun möchtest du wissen welchen Weg du vom aktuellen Standort zur Flossbar nehmen musst.

Diese Aufgaben mussten die Probanden anschliessend mit unserem HTC Desire mit Android 2.2 lösen.

10.3.4.3 Resultat

Grundsätzlich haben alle Probanden mit der Applikation die geschilderten Probleme lösen können. Folgende Punkte wurden jedoch angemerkt:

1. Im Zeitplan-Tab wurde vermutet, dass auf die einzelnen Schedule-Einträge geklickt werden kann. Erwarten wurden mehr Infos zu dem entsprechenden Punkt.
2. Es war mühsam, dass man bei dem Aufruf des Karten-Tap automatisch bei der GPS-Aktivierung zentriert wurde. Dies kann für das Suchen auf der Karte hinderlich wirken.
3. Die verschiedenen Touchpunkte werden nicht bei jeder Fingerposition (Click) erkannt. Teilweise mehrere Klicks nötig.
4. Bei der GPS-Aktivierung in den Einstellungen wussten die nicht Android Nutzer nicht, wie man zurück in das Program kommt (Zurück-Button).
5. Der Zoom über die Pinchaktion wurde nicht verwendet. Hier wurden hauptsächlich die Plus und Minus-Tasten verwendet.
6. Auf der Karte werden die Punkte ohne Beschriftung auf der Karte nicht erkannt.

10.3.4.4 Entscheide

Folgende Verbesserungen beziehen sich auf die Resultate.

Anmerknungsnummer	Weiteres Vorgehen
1	Hier wird nichts verändert, da keine weitere Informationen zu den Punkten vorhanden sind.
2	Dies muss umgebaut werden, dass auch wenn GPS aktiviert ist nicht gleich die aktuelle Position zentriert wird. Sonst kann sich nur über die Zoomfunktion einen Überblick geschaffen werden. Dennoch könnte dies über den Knopf im unteren Bereich ausgeschaltet werden.
3	Dies scheint je nach Gerät unterschiedlich zu sein. Auf dem Desire HD tritt dies weniger auf.
4	Dies ist leider nicht anders zu Implementieren, Android spezifisch.
5	Hier irritieren eventuell die +/- Buttons, funktionalität bleibt.
6	Die Beschriftung muss dringend auf der Karte vermerkt werden!

Tabelle (10.1) Final Usabilitytests-Entscheide

Diese Entscheide sind **NICHT** umgesetzt, diese sind in den nächsten Schritten geplant.

11 Resultate und Weiterentwicklung

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	01.12.2010	Michael Schnyder	Dokumentvorlage erstellt
0.2	11.12.2010	Felix Egli	Evaluation Feature Notifikation angefangen
0.3	20.12.2010	Felix Egli	Evaluation Feature Notifikation abgeschlossen
0.4	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

11.1 Resultate

Wurde bereits im technischen Bericht (siehe *Zielerreichung* auf Seite 29) umfassend beschrieben.

11.2 Möglichkeiten der Weiterentwicklung

11.2.1 Feature Notifikation

Das Feature wurde aus Zeitgründen nicht mehr implementiert, aber dafür erweitert evaluiert. Hierfür wurde ein Android-Service¹ angedacht, welcher immer im Hintergrund läuft. Dieser läuft auch wenn die Applikation nicht gestartet ist.

Dieser Service muss in der Lage sein mit unserem EventAssistant-Programm zu kommunizieren. Dies geschieht über eine von Android zur Verfügung gestelltes Interface. Dieses wird durch ein AIDL-File² erzeugt. Das kann direkt in der Eclipse-Umgebung erstellt werden. Danach wird im generated-Folder direkt das .java-File erzeugt.

In unserem Fall muss über das Interface zwei Dinge erledigt werden können: Notifikation hinzufügen und Notifikation entfernen.

INotificationService.aidl

```
1 package ch.hsr.eventassistant.service;

interface INotificationService {
    boolean addNotification(in int id, in String notification, in int date);
5    boolean deleteNotification(in int id);
}
```

1 <http://developer.android.com/reference/android/app/Service.html>

2 <http://developer.android.com/guide/developing/tools/aidl.html>


```
}
```

Nun eigenen Service erzeugen welcher von der Android-Service-Klasse ableitet.

EventAssistantNotificationService.java

```
1 package ch.hsr.eventassistant.service;

import android.app.Service;
import android.content.Intent;
5 import android.os.IBinder;
import android.util.Log;

public class EventAssistantNotificationService extends Service {

10     private static final Object notificationHandlerLock = new Object();
    private static NotificationHandler notificationHandler = null;

    @Override
    public void onCreate() {
15         synchronized (notificationHandlerLock) {
            if (notificationHandler == null) {
                notificationHandler = new NotificationHandler
                    (getApplicationContext(), true);
            }
20         }
        super.onCreate();
    }

    @Override
25     public IBinder onBind(Intent intent) {
        if (INotificationService.class.getName()
            .equals(intent.getAction())) {
            return notificationHandler.asBinder();
        }
30         return null;
    }
}
```

Nun kann ein eigener Service erstellt werden welcher dieses generierte Interface implementiert, um den Handler zur Verfügung zu stellen.

NotificationHandler.java

```
1 package ch.hsr.eventassistant.service;

import android.content.Context;
```



```
5  import android.os.IBinder;
import android.os.RemoteException;

public class NotificationHandler extends INotificationService.Stub {

    10      public NotificationHandler(Context applicationContext, boolean b) {
        ...
    }

    public IBinder asBinder() {
    15      return this;
    }

    public boolean addNotification(int id, String notification, int date)
        throws RemoteException {
    20      ...
    }

    public boolean deleteNotification(int id) throws RemoteException {
    25      ...
    }
}
```

Zum Schluss muss der Service noch im `AndroidManifest.xml`¹ des Android Projektes eingetragen werden.

AndroidManifest.xml

```
1  <application>
    ...
    <service android:enabled="true" android:name=
5      ".service.EventAssistantNotificationService" />
    ...
</application>
```

Eine erste Version des GUI's wurde auch hierfür schon entwickelt. Auch der UI-Paperprototyp wurde anhand dieses Features entwickelt. Dieses Feature hätte noch ca. einen Aufwand von zwei Wochen bis es implementiert ist.

¹ [urlhttp://developer.android.com/guide/topics/manifest/manifest-intro.html](http://developer.android.com/guide/topics/manifest/manifest-intro.html)

11.2.2 Backend / Datenbank / Web-GUI

Die wichtigste Erweiterung wäre auf der Serverseite. Als erstes müsste hier eine richtige Datenbank als Adapter gebaut werden, welche das Interface *DataSource* implementiert. Hier würde eine einfache MySQL-Datenbank den Anforderungen entsprechen. Der Aufwand für eine solche Anpassung würde ca. zwei Tage in Anspruch nehmen.

Ein weiterer Schritt wäre ein Web-GUI, auf welchem die Daten komfortabel in die Datenbank eingepflegt werden könnten. Wenn dies erledigt wurde, kann die Applikation offiziell freigegeben werden. Als Technologie zur Darstellung eine simple Webseite, denn die Eingaben können einfach über Formulare gehandhabt werden. Wenn man für die Vermessung der Karte und das Setzen der Referenzpunkte eine etwas grafisch ansprechende Methode machen will (z.B. mit on-Screen Punkte setzen) kann dies sicher ziemlich aufwändig werden. Der Aufwand für eine solche Arbeit ohne komfortable Referenzpunkte setzen wäre auch ca. zwei Tage. Mit Fancy Referenzierung ca. ein bis zwei Wochen.

11.2.3 Optionale Features

Freunde finden

Bestimmt eines der in die Zukunft gerichteten Features[MP08][RCBH10][UB08], wie auch schon in Einführung erwähnt. (siehe *Problemstellung*, *Vision* auf Seite 3) Hier steht die Facebook Integration im Vordergrund, hier sind auf dem Server wie auch auf dem Client Änderungen nötig. Zudem müsste es möglich sein, Daten vom Client zum Server zu übertragen. Dies ist keine Hexerei, sondern wäre mit einfachen POST aufrufen möglich.

Dennoch müssten noch einige genauere Abklärungen für dieses Feature gemacht werden. Darum ist der Aufwand für das Feature auch nicht genau abzuschätzen, würde aber bestimmt mehrere Wochen benötigen.

Push-Notifications

Hier wurden noch keine konkreten Abklärungen gemacht. Doch es müssten auf Client wie auf Serverseite Anpassungen vorgenommen werden, Implementationszeitraum mehrere Wochen.

12 Projektmanagement

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	23.09.2010	Felix Egli	Dokumentvorlage erstellt
0.2	30.09.2010	Felix Egli	Q-Massnahmen erstellt
0.3	20.12.2010	Felix Egli	Q-Massnahmen und weiters überarbeitet
0.4	21.12.2010	Felix Egli	Formatierungen überarbeitet
0.5	22.12.2010	Felix Egli	Projektplan verlinkt
0.6	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

12.1 Entwicklungsumgebung

Wurde bereits beschrieben (siehe *Entwicklungsumgebung* auf Seite 7) und (siehe *Entwicklungsumgebung* auf Seite 122).

12.2 Projektplan

12.2.1 Zeitplan

(siehe *Projektplan* auf Seite 143)

12.2.2 Iterationsplanung

<i>Iteration</i>	<i>Beschreibung</i>
Inception	Projektantrag, Vision, Dokumentvorlagen, Projektplan (Zeitplan, Risiko Management, Codierungsrichtlinien, Qualitätsmerkmale), Konfigurationsverwaltung, Interviews
Elaboration 1	Funktionale und nichtfunktionale Anforderungen (Use Cases im brief Format, Supplementary Specification), Domain Model, Technologieabklärngen, Paperprototyp, auswerten Interviews
Elaboration 2	100% Use Cases im fully dressed Format (zumindest der nichtoptionalen Features), Architekturprototyp, Paperprototypetests
Construction 1	Prototyp der Applikation und der einzelnen Teilaufgaben, UI Paperprototyp
Construction 2	Zusammenführen der Prototypen & Testen
Construction 3	Ausbau Prototyp zu weiteren Funktionalität
Transition	Usabilitytest, Installationsanleitung, Schlusspräsentation, YouTube Video

Tabelle (12.1) Iterationsplanung

12.2.3 Meilensteine

(siehe *Projektplan* auf Seite 143)

12.2.4 Meetings

Was	Wann	Wo
Weeklymeeting - Arbeitsdefinitionen, Todo für kommende Woche	Mittwoch's um 13.10	IFS

Tabelle (12.2) Meetings

12.3 Team und Verantwortlichkeiten

Das Team besteht aus zwei Personen. Michael Schnyder (m2) und Felix Egli (fx).

Bereich/Dokument	Verantwortlich
Projektplan Excel	m2
Zeiterfassung	m2
Q-Massnahmen	fx
LaTex	fx
Konfigurationsmanagement	m2
Projektautomation	fx
Interviews	fx
Szenarios	fx
Domain Model	m2
Technologieabklärungen	beide
Caching	fx
GUI-Implementation	m2
GUI-Prototypen	fx
Tests	beide

Tabelle (12.3) Verantwortlichkeiten

12.4 Risikomanagement

12.51 fx, m2 Autor: See SVN Revisor:		Risikobeschreibung	max. Schaden [h]	Eintretswahrscheinlichkeit	gewichteter Schaden [h]	Indikator	Massnahmen zur Vermeidung/Vermindeung	Aufwand Massnahmen	Vorgehen bei Eintreffen
R1	Fehlenschätzung des Aufwandes	Zeitplan wird nicht eingehalten, da der Aufwand von einzelnen Arbeitspaketen falsch eingeschätzt wurde	60	1.00 %	0.60	Jede Iteration weicht Ist/Soll immer mehr ab	kritische Arbeitspakete frühzeitig bearbeiten		nicht alle geplanten Funktionen und Features umsetzen, Mehrarbeit
R1.4	Unbrauchbares System	System bzw. Funktionen des Systems werden nicht genutzt werden	504	0	0	Paperprototype Funktionen werden nicht benötigt	Interviews, Paperprototype erstellen	15.00	
R4.1	Technologie	Technologieknow-how ist zu niedrig um spezifisch arbeiten zu können.	60	1.00 %	0.60		CodeReview mit Anderen Android-Gruppen	10.00	
R1.7	Infrastrukturausfall	Wiki, Trac, SVN fallen aus	476	1.00 %	4.76	HW/SW-Teilausfälle	Backup	5.00	
R1.7	Usability	Software kann nicht benutzt werden	150	1.00 %	1.50	Bedenfehler bei Testen	Prototypen, Usability Tests (ganz zum Schluss)	13.00	
R4.2	Technologie	Einarbeitung benötigt zuviel Zeit	25	1.00 %	0.25		Diskussion mit anderen Android-Fachverständigen	10.00	
R2	Interviews sind nicht aussagekräftig	Es können zu wenige Interviews mit vernünftigen Teilnehmern gemacht werden.	30	0.00 %		Testlauf Interviews funktioniert nicht optimal	Interviews testen	7.00	Weitere Interviews machen
R9	App: Architekturprobleme	Architektur stimmt nicht mehr mit Anforderungen überein	35	1.00 %	0.35	Anforderungen konsultieren	Laufend mit Anforderungen überprüfen	5.00	
R1.2	App: Zugriffprobleme Client-Server	Zugriff funktioniert nur wenig sicher, zu komplizierte Aufrufe	35	1.00 %	0.35	Laufend überprüfen	Architekturprototyp	15.00	
R1.5	Technologie	Technologie ist (noch) nicht brauchbar	150	1.00 %	1.50	Funktionen sind buggy, wenig Hilfestellung im Netz	Technologiestudium (theoretisch) benötigte Funktionen ableiten	8.00	
R5	Streit im Team	Konflikte im Team	100	0		Unruhe im Team, schlechte Umgangssprache	konstruktive Kritik und Kommunikation innerhalb des Teams pflegen		
R6	Datenverlust	Daten gehen auf Server oder Lokal verloren	50	0		HW-Teilausfälle	HW-Monitoring/ Backup		
R1.1	Testing	HW-Geräte lassen sich nicht geeignet testen, testen in Real-Life entfallen, weil keine Events.	50	0.00 %		Testevents lassen sich nicht finden	Frühzeitige Abbildungen	5.00	Imaginärer Event an HSR
R1.6	Technologie	HW-Geräte sind nicht in gefordertem Umfang erhältlich	50	0.00 %		Lieferschwierigkeiten, Organisationsprobleme iPS	Geräte werden private selber gekauft	5.00	
R1.0	Dokumentversionsverwaltung Probleme	Änderungen werden durch Check-in überschrieben, nicht gemerged werden	8	0.2	1.6		Abklären wer an diesem Dokument arbeitet, commit Kommentare setzen, locked Check-out, verwendung von LaTeX		
R8	Paper-Prototype nicht brauchbar	Der PPT kann nicht für die Applikation verwendet werden, und muss neu erstellt werden	20	0.05	1		Szenarios genau beachten		
R1.3	Partnerrolle	Partner würden Corporate Identity voraussetzen	100	0.00 %	0.00		Partner wurde eliminiert		
Summe			1903	33.00 %	12.51				

Abbildung (12.1) Risikoliste

12.5 Prozessmodell

Der Organisation des Projekts liegt der *RUP*¹ zugrunde. Ebenso wird die Vorlage von Herr Prof. Dr. Stefan Keller [Kel08] bezüglich der Dokumentation angewendet. Ergänzend dazu wird die Kriterienliste BA/SA von Herr Prof. Dr. Markus Stolze [Sto10] verwendet und die Bestimmungen zur Durchführung von SA und BA an der HSR.

12.6 Q-Massnahmen

Folgende Q-Massnahmen werden durchgeführt. Alle in der Dokumentation protokolliert.

Massnahme	Umsetzung
Codereview	Dreimal im Projekt werden Codereviews gemacht um gegenseitig Erfahrungen auszutauschen.
Codeanalysetool	Structure 101 wird eingesetzt um Codemetriken und zykläre Abhängigkeiten usw. zu testen.
Risikomanagement	Jede Woche wird die Risikoliste neu bewertet und aktualisiert.
Unittests	Für die wichtigsten Komponenten werden Unittests geschrieben.
Systemtests	Zweimal werden Systemtests durchgeführt.
Papierprototypen	Damit unser Interface benutzerfreundlich wird, werden Papierprototypen erstellt.
Interviews	Damit die Anforderungen für unser Projekt nicht aus der Luft gegriffen werden, werden Interviews durchgeführt.
Szenarios	Damit die Anforderungen auch testbar sind, werden Szenarios beschrieben.
Zeiterfassung	Damit die Zeit nicht aus dem Ruder-Läuft wird das Zeiterfassungstool mite ² verwendet.
Usabilitytests	Mehrmals während des Projekts werden Usabilitytests durchgeführt.

Tabelle (12.4) Q-Massnahmen

¹ Rational Unified Process

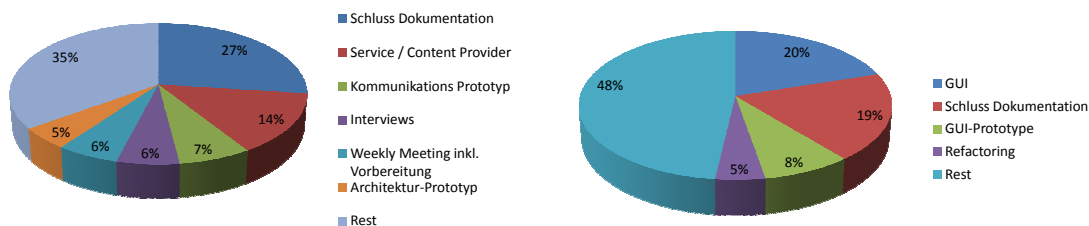
13 Projektmonitoring

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	15.12.2010	Michael Schnyder	Dokumentvorlage erstellt
0.2	20.12.2010	Felix Egli	Sitzungsprotokolle verlinkt
0.3	21.12.2010	Felix Egli	Codestatistik erstellt
0.4	21.12.2010	Michael Schnyder	Haupttätigkeiten, Soll/Ist. Vergleiche
0.5	22.12.2010	Felix Egli	Codereviews referenziert
0.6	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

13.1 Soll-Ist-Zeitvergleich

13.1.1 Haupttätigkeiten



(a) Auswertung Felix Egli

(b) Auswertung Michael Schnyder

Abbildung (13.1) Pie-Charts Haupttätigkeiten

Aus den obigen Diagramme wird gut ersichtlich, dass die Arbeitsteilung funktioniert hat. Ebenso ist klar ersichtlich, dass die Dokumentation (ohne Szenarien, Interviews, Use Cases und Personas) einen sehr grossen Aufwand darstellt. Vergl. auch die folgende Grafik.

Grösste Arbeitspakete Beide

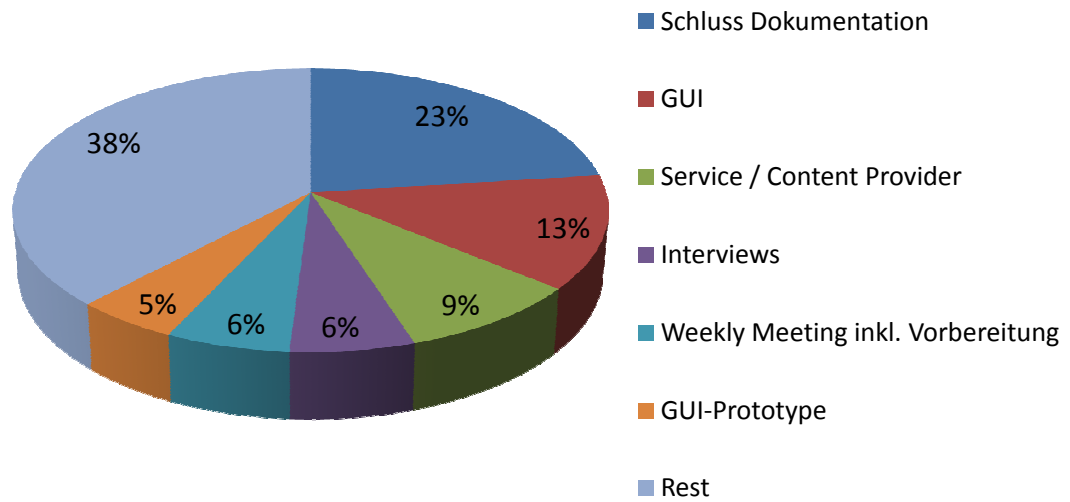


Abbildung (13.2) Pie-Charts Haupttätigkeiten Beide

13.1.2 Soll / Ist Vergleich

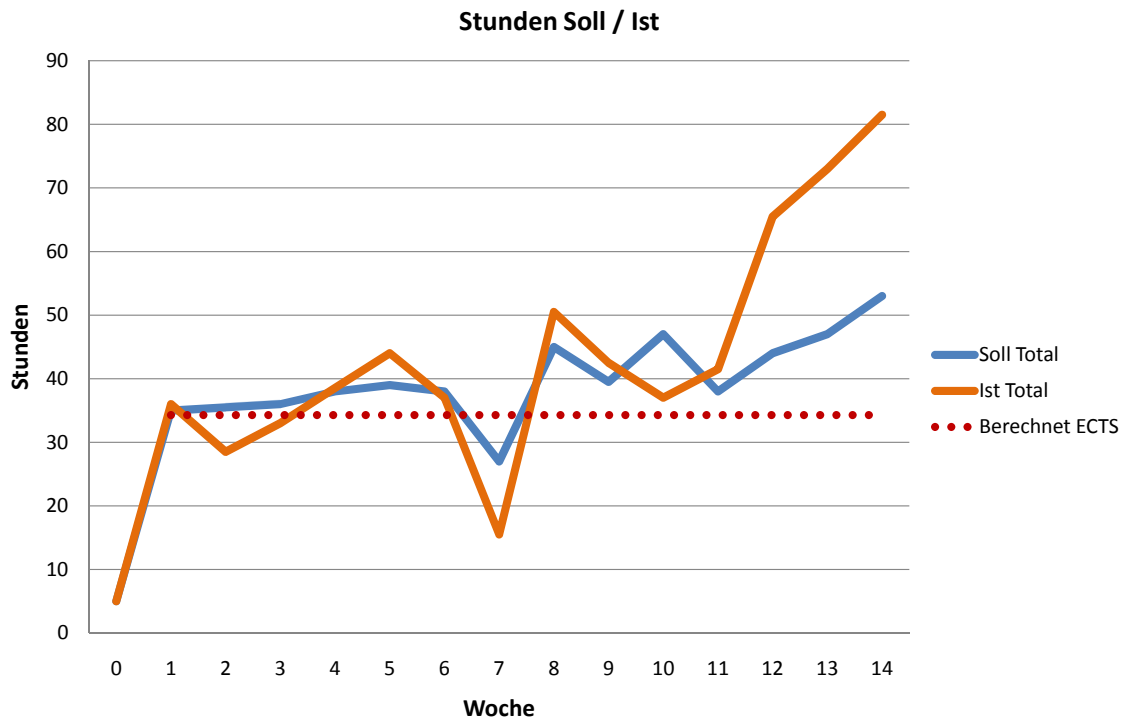


Abbildung (13.3) Verlauf Soll/Ist Stunden

Das Projekt begann bereits mit Vorbereitungen in der Woche 0. Danach wurde versucht, die geforderten 240h/Person aufgeteilt auf 15 Wochen einzuhalten. Wir haben jeweils mit ca. 34.3 Stunden gerechnet. (Siehe rote, punktierte Linie). Es fällt auf, dass die Ist-Stunden ab Woche 11 sehr stark von den geplanten Wochen abweichen. Hauptgrund hierfür ist das zu optimistische Einschätzen der Aufwendungen für die Dokumentation. In Woche 7 war die Arbeitsleistung durch die Abwesenheit von Felix Egli (Militärdienst) vermindert. Dies konnte jedoch bis zu Woche 11 wieder kompensiert werden.

13.2 Codestatistik

Folgende Codemetriken wurden mit dem *Structure101 for Java* erstellt.

EventAssistant-Server

Beschrieb	Anzahl
Packages	6
Classes	30
NI (Number of bytecode Instructions)	3,657
LOC (Non Comment Non Blank Lines Of Code)	approx. 1,573

Tabelle (13.1) Server Codemetriken

EventAssistant-Client

Beschrieb	Anzahl
Packages	11
Classes	28
NI (Number of bytecode Instructions)	8,673
LOC (Non Comment Non Blank Lines Of Code)	approx. 3,729

Tabelle (13.2) Client-Codemetriken

EventAssistant-Common

Beschrieb	Anzahl
Packages	1
Classes	9
NI (Number of bytecode Instructions)	379
LOC (Non Comment Non Blank Lines Of Code)	approx. 163

Tabelle (13.3) Common Codemetriken

EventAssistant-Test

Beschrieb	Anzahl
Packages	1
Classes	8
NI (Number of bytecode Instructions)	1,433
LOC (Non Comment Non Blank Lines Of Code)	approx. 616

Tabelle (13.4) Test Codemetriken

13.2.1 Bemerkungen

In keinem unserer Projekte wurde eine zirkuläre Abhängigkeit gefunden. Auch keine übergrossen Klassen anhand der hinterlegten Metriken von Structure101. Lediglich einige Weisenklassen (Orphans) gibt es. Diese sind aber hauptsächlich in den Test-Projekten vorhanden. Aber auch auf dem Client Projekt, da sind es vom Android-Plugin generierte Ressourcen. Die genauen Auswertungen sind unter Anhang aufgelistet vorhanden.

(siehe *Codemetriken* auf Seite 205)

13.3 Protokolle

(siehe *Sitzungsprotokolle* auf Seite 193)

13.4 Code-Reviews

Codereviews wurden jeweils an den Meilensteinen in der Construction Phase durchgeführt worden. (siehe *Projektplan* auf Seite 143)

14 Softwaredokumentation

Dokumenthistory

Rev.	Datum	Wer	Änderung
0.1	15.12.2010	Michael Schnyder	Dokumentvorlage erstellt
0.2	21.12.2010	Felix Egli	Referenz-Server erfasst
0.3	22.12.2010	Michael Schnyder	Content Erfassung beschrieben
0.4	22.12.2010	Michael Schnyder	Installation für Client und Entwicklungsumgebung beschrieben
0.5	23.10.2010	F. Egli & M. Schnyder	Review und Korrektur

14.1 Installation

14.1.1 Android-Client

Der Android-Client befindet sich bereits auf dem Leihgerät der HSR. Eine Neuinstallation kann wie folgt durchgeführt werden.

1. **Unbekannte Quellen:** Das Installieren aus Unsicheren Quellen Zulassen. Menue „Einstellungen“ » „Applikationen“ » „Unbekannte Quellen“ einschalten.
2. **Download:** .apk-File auf Gerät kopieren (CD) oder downloaden (Wiki-Seite)
3. **Installation:** .apk-File öffnen und installieren.

Die Applikation wurde auf folgenden Plattformen einem funktionalen Test unterzogen.

Gerät	Android Version	Auflösung	Prozessor	Status
HTC Desire	2.2	480 x 800 Pixel (WVGA), 4.3"	1 GHz	Kompatibel
HTC Desire HD	2.2	480 x 800 Pixel (WVGA), 3.7"	1 GHz	Kompatibel
HTC Legend	2.2	320 X 480 Pixel HVGA, 3.2"	600 MHz	Kompatibel

Tabelle (14.1) Gerätekompatibilität

14.1.2 Referenz-Server

Als Referenzserver wurde ein Apache Tomcat v7¹ verwendet. Standartinstallation durchgeführt, unser Server-Projekt als .war-File exportiert und im Tomcat *webapps*-Folder deployt.

¹ <http://tomcat.apache.org/download-70.cgi>

Dieser läuft zur Zeit auf <http://eventassistant.dyndns.org/EventAssistant-Server/> und ist über die Formate in *Schnittstellen und Formate* (siehe Seite 63) beschrieben.

14.1.3 Entwicklungsumgebung

Die nachfolgenden Schritte müssen durchgeführt werden, um das Projekt innerhalb einer Android-Entwicklungsumgebung zu laden

1. **Eclipse:** Für die Entwicklung wird eine Eclipse-Umgebung vorausgesetzt.
2. **Android SDK:** Download auf der Android-Webseite¹
3. **Installation:** Analog den Anweisungen auf der Android-Webseite²
4. **Import:** Die benötigten Projekte befinden sich auf der CD unter *Source*. Dies ist bereits ein Eclipse-Workspace.
5. **Debugging:** Für das Debugging auf einem Android-Gerät muss den Treiber separat auf dem PC installiert werden. Siehe Android-Webseite³ für Windows-Installationen.

Die genauen Versionsinformationen sind im Technischen Bericht aufgelistet (siehe *Entwicklungsumgebung* auf Seite 7).

14.2 Content Erfassung

Die Bereitstellung von Daten für den EventAssistant ist denkbar einfach. Folgende Informationen werden für einen Event benötigt:

- **Eventinformationen** Eventnamen, Ort, Datum, Eintritt, Zusatztext. Siehe hierzu auch die Screenshots der Applikation (siehe *Subfigure 10 10.6a* auf Seite 98)
- **Event-Position** Bereich in welchem der Event statt findet (Liste von Geokoordinaten als Polygon). Die Applikation kann mit dieser Information direkt den aktuellen Anlass anzeigen
- **Karte** Normales Bild mit Icons für die POI und zwei Referenzpunkten für die Geo-Referenzierung
- **Points of Interest** Einzelne Punkte auf der Karte mit Zusatzinformationen und Kartenposition (Liste von Geokoordinaten als Polygon)
- **Zeitpläne** Können für jeden POI erfasst werden. Verwendete Daten: Datum/Zeit, Titel, Beschreibung. (siehe *Screenshots Nebenscreens* auf Seite 74)

¹ <http://developer.android.com/sdk/index.html>

² <http://developer.android.com/sdk/installing.html>

³ <http://developer.android.com/sdk/win-usb.html>

14.2.1 Geokoordinaten

Als Geokordinaten werden Koordinaten im lat-/long-Format verwendet (siehe *Schnittstellen und Formate* auf Seite 63). Diese können sehr einfach über Online-Karten-Tools ermittelt werden. Wir haben für die Erfassung der Daten Google-Maps¹ verwendet. Hier gibt es das Addon „LatLng-Markierung“ von Marceki C. welches unter dem Menüpunkt „Neu!“ in der Titelleiste rechts eingeschaltet werden kann. Mit diesem Addon können mit der rechten Maustaste Markierungen auf der Karte gesetzt werden, welche die exakten Koordinaten anzeigen.

14.2.2 Geo-Referenzierung

Wird benötigt, damit die Position des Besuchers auf der Karte berechnet werden kann. Die Position des Besuchers wird mit einem roten Punkt dargestellt.

¹ <http://maps.google.ch>

Literaturverzeichnis

- [AB10] ARNO BECKER, Marcus Pant: *Android 2: Grundlagen und Programmierung* (2010)
- [and10] Android User Interface Guidelines (2010), URL http://developer.android.com/guide/practices/ui_guidelines/index.html, [Letzter Aufruf: 10.11.2010]
- [CE10] CHRISTOPH EGGER, Adrian Geiter: *IndoorGuide4Android2*, Dissertation, HSR Hochschule für Technik Rapperswil (2010)
- [Fac10] FACEBOOK: Places: Who. What. When. And now Where. (2010), URL <http://www.facebook.com/places/>, [Aufruf: 23.09.2010]
- [Kel08] KELLER, Prof. Dr. Stefan: Tipps für die Strukturierung und Planung von Studien-, Diplom- und Bachelorarbeiten (2008)
- [MP08] MARTIN PRESS, Roberto A. Flores, Daniel Goodwin: PALASS: A Portable Application for a Location-Aware Social System. *ACM* (2008), URL <http://portal.acm.org/citation.cfm?id=1484422.1484522&coll=DL&dl=GUIDE&CFID=2403832&CFTOKEN=13983720>
- [RCBH10] ROST, Mattias; CRAMER, Henriette; BELLONI, Nicolas und HOLMQUIST, Lars Erik: Geolocation in the Mobile Web Browser. *ACM* (2010), URL <http://delivery.acm.org/10.1145/1870000/1864468/p423-rost.pdf?key1=1864468&key2=6957492921&coll=DL&dl=ACM&CFID=2403832&CFTOKEN=13983720>
- [Sto10] STOLZE, Prof. Dr. Markus: Bewertungskriterien SA / BA (2010)
- [SYH10] SAYED Y. HASHIMI, Dave MacLean, Satya Komatineni: *Pro Android 2* (2010)
- [TO09] TATJANA OSSIPOVA, Michel Reiter: *IndoorGuide4Android*, Dissertation, HSR Hochschule für Technik Rapperswil (2009)
- [UB08] UDANA BANDARA, Pulathisi Bandara: Tagciti: A Practical Approach for Location-Aware and Socially-Relevant Information Creation and Discovery for Mobile Users. *IEEE* (2008), URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04726030&tag=1>

Glossar

Android	Von der OpenHaedset-Alianz geschaffenes mobiles Betriebssystem
Fully Qualified Domain Name	Ist die Bezeichnung für einen Ausdruck welcher als Domain Namen eindeutig ist. Beispiel: www.hsr.ch ist eindeutig, da er als Top-Level Domain der Domain ch. angehört
HTML5	Format für Interaktive Anwendungen im Webbrowser. Basiert auf HTML4, jedoch mit vielen Erweiterungen, welche bisher in Javascript gemacht werden mussten. Befindet sich noch in der Entwicklung. Die Unterstützung ist je nach Browserhersteller unterschiedlich
JSON	JavaScript Object Notation, welches hauptsächlich für leichtgewichtige Kommunikation verwendet wird. Fast, Lightweight, Easy to Use and Type Safe JSON
jsonmarshaller	marshalling library for Java (http://code.google.com/p/jsonmarshaller/)
Structure101 for Java	Codemetrikanalyse Tool für Java mehr unter http://www.headwaysoftware.com/products/structure101/

Abkürzungsverzeichnis

AR	Augmented Reality
CI	Corporate Identity
EA	EventAssistant
FQDN	Eindeutiger Domain Name (engl. Fully Qualified Domain Name)
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
IDE	Integrierte Entwicklungsumgebung (engl. integrated development environment)
iOS	iPhone OS
POI	Point of Interest
RUP	Rational Unified Process
SSD	System Sequenz Diagramm
SVN	Subversion
WLAN	Wireless Local Area Network

Abbildungsverzeichnis

1	Evaluation	XI
2	Applikation Screenshots	XII
3	Schematische Darstellung	XIII
2.1	Screenshots Swisscom Openair Buddy	12
2.2	Screenshots FestivalMob	13
2.3	Screenshots Festivalapps kleiner und grosser Festivals	14
5.1	Screenshot Laufende Threads	32
6.1	Peter Planer	41
6.2	Susi Sorglos	43
6.3	Use Cases	48
7.1	Problem Domain	54
8.1	Architektur Overview	57
8.2	Package Overview	59
8.3	Klassen Overview	62
8.4	Datenklassen Übersicht	65
8.5	Screenshots Hauptscreens	73
8.6	Screenshots Nebenscreens	74
9.1	Beispiel für ContentValues generisch	80
9.2	Cache Algorithmus	81
9.3	Server SSD	83
9.4	Server 404	84
9.5	Server Klassendiagramm	85
10.1	Unittests ausgeführt auf Android-Testprojekt	93
10.2	Unittests ausgeführt der Datenklassen	94
10.3	Unittests ausgeführt der FakeDatabase	95
10.4	Papierhandy	97
10.5	Testumgebung	98
10.6	Beispielscreens	98
10.7	Tests zu Hause	101
10.8	Testumgebung und Personen	102
12.1	Risikoliste	112
13.1	Pie-Charts Haupttätigkeiten	115
13.2	Pie-Charts Haupttätigkeiten Beide	116

13.3 Verlauf Soll/Ist Stunden	117
---	-----

Tabellenverzeichnis

1.1	Wichtige Termine	6
1.2	Hardware	6
1.3	Entwicklungsumgebung	7
1.4	Organisations-Systeme	7
3.1	Technologieentscheid Tabelle	18
5.1	Zielerreichung Must-Use Cases	29
5.2	Zielerreichung Optionale-UseCases	30
5.3	Zielerreichung Nichtfunktionale Anforderungen	31
5.4	Laufende Threads Erläuterung	32
6.1	Profil Peter Planer	42
6.2	Profil Susi Sorglos	43
8.1	UI-Packete	60
8.2	Data-Packete	61
8.3	Server-Packete	61
8.4	Datenbankschema Konventionen	63
8.5	Ressourcenbeschrieb	64
8.6	Datenformate und Konvertierung	66
8.7	Organisation UI-Elemente	72
8.8	Definitionen Hauptscreens	73
8.9	Definitionen Nebenscreens	74
9.1	Objekthierarchie	78
10.1	Final Usabilitytests-Entscheide	104
12.1	Iterationsplanung	109
12.2	Meetings	110
12.3	Verantwortlichkeiten	110
12.4	Q-Massnahmen	113
13.1	Server Codemetriken	118
13.2	Client-Codemetriken	118
13.3	Common Codemetriken	118
13.4	Test Codemetriken	119
14.1	Gerätekompatibilität	121

Anhang A

Aufgabenstellung

Aufgabenstellung SA Michael Schnyder & Felix Egli

EventAssistant

0. Auftraggeber und Betreuer

Diese Studienarbeit hat keinen direkten Auftraggeber. Als interimistischer Auftraggeber fungiert das Institut für Software.

Ansprechpartner Auftraggeber:
(NA)

Betreuer HSR:
Prof. Dr. Markus Stolze, Institut für Software mstolze@hsr.ch

1. Ausgangslage

Organisationen wie die Swisscom stellen vermehrt spezielle Mobil-Anwendungen für Grössenveranstaltungen wie Open-Air Konzerte, Turnfeste oder Skirennen zur Verfügung. Aktuell hat die Swisscom eine Basisanwendung für iPhone entwickelt, aber entsprechender Support für Android Telefone fehlt.

Diese Lücke soll durch die zu entwickelnde Anwendung „EventAssistant“ geschlossen werden

2. Ziele der Arbeit

Durch die zu entwickelnde Applikation erhält der Besucher von Grossveranstaltungen die Möglichkeit, sich innerhalb von Grossveranstaltungen (z.B. Openairs, Sportveranstaltungen, etc.) mittels Handy zu orientieren. Die Daten für die Navigation werden zentral gespeichert und je nach Standort des Benutzers die notwendigen Daten auf das Endgerät übermittelt.

Mögliche Funktionen

- Gebäude / POI auf Karte darstellen und mit Informationen (tbd) versehen
- Zeltsuche / eigenes Zelt wiederfinden
- Facebook-Integration / Wo sind meine Freunde am Openair?
- Direktes senden von Fotos an LED-Wand
- Wichtige Informationen per Push
- Sportresultate...
- etc.

Als Teil der Arbeit müssen Umfang und gewünschte Funktionen anhand von Interviews bestimmt werden.

Ziel ist die Erstellung eines lauffähigen Systems mit wertvollen Funktionen. Parallel zur Anforderungsanalyse sollte auch eine Technologiestudie stattfinden in der unter anderem HTML 5 und Android untersucht wird.

3. Zur Durchführung

Mit dem HSR-Betreuer finden in der Regel wöchentliche Besprechungen statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen.

Alle Besprechungen sind von den Studenten mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das dem Betreuer E-Mail zugestellt wird. Zudem wird eine Projektseite ausgehend von Prof. Stolzes internen SA 2010 Wiki gemacht:

<http://sinv0002.hsr.ch/MarkusStolze/wiki.cgi?StolzeArbeitenHS2010>

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsergebnisse in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsergebnisse erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt aufgrund der am Abgabetermin abgelieferten Dokumentation. Die Evaluation erfolgt aufgrund des separat abgegebenen Kriterienkatalogs in Übereinstimmung mit den Kriterien zur SA Beurteilung. Es sollten hierbei auch die Hinweise aus dem abgegebenen Dokument „Tipps für die Strukturierung und Planung von Studien-, Diplom- und Bachelorarbeiten“ beachtet werden.

4. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen. Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Die Dokumentation ist vollständig auf CD/DVD in 2 Exemplaren abzugeben. Zudem ist eine kurze Projektergebnisdokumentation im Wiki von Prof. Stolze zu erstellen dies muss einen Link auf einen YouTube Video enthalten welche das Resultat der Arbeit dokumentiert.

5. Termine

Siehe auch Terminplan auf

<https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html>

20.09.10 Beginn der Studienarbeit, Ausgabe der Aufgabenstellung durch die Betreuer.

20.12.10 Die Studierenden senden folgende Dokumente der Arbeit per Email zur Prüfung an ihre Betreuer:

- Abstract/Kurzfassung
- A0-Poster

Vorlagen stehen unter den allgemeinen Infos Diplom-, Bachelor- und Studienarbeiten zur Verfügung.

- 23.12.10 Die Studierenden senden das vom Betreuer abgenommene und freigegebene Abstract/Kurzfassung als Word-Dokument an das Studiengangsekretariat (cfurrer(at)hsr.ch).
23.12.10 Abgabe des Berichtes und CDs an den Betreuer bis 17.00 Uhr.

6. Beurteilung

Eine erfolgreiche SA zählt 8 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von ca. 25 bis 30 Stunden budgetiert. Entsprechend sollten ca. 240h Arbeit für die Studienarbeit aufgewendet werden. Dies entspricht ungefähr 17h pro Woche (auf 14 Wochen) und damit ca. 2 Tage Arbeit pro Woche.

Für die Beurteilung sind der HSR-Betreuer verantwortlich unter Einbezug des Feedbacks des Auftraggebers (welches in diesem Fall entfällt)

Die Bewertung der Arbeit erfolgt entsprechend der verteilten Kriterienliste

Rapperswil, den 20. September 2010



Prof. Dr. Markus Stolze
Institut für Software
Hochschule für Technik Rapperswil

Anhang B

Vereinbarung



Vereinbarung

1. Gegenstand der Vereinbarung

Mit dieser Vereinbarung werden die Rechte über die Verwendung und die Weiterentwicklung der Ergebnisse der Studienarbeit EventAssistant von Felix Egli und Michael Schnyder unter der Betreuung von Dr. Prof. Markus Stolze geregelt.

2. Urheberrecht

Die Urheberrechte stehen der Studentin / dem Student zu.

3. Verwendung

Die Ergebnisse der Arbeit dürfen sowohl von der Studentin / dem Student und der HSR nach Abschluss der Arbeit verwendet und weiter entwickelt werden. Die Überführung in ein kommerzielles Produkt durch die HSR wird untersagt.

Beilage/n:

keine

Rapperswil, den 22.12.2010


.....
Die Studentin/der Student

Rapperswil, den 22.12.2010


.....
Der Betreuer / die Betreuerin der Studienarbeit

Anhang C

Projektplan

Seite 2 von 2

Anhang D

Interviewfragebogen Freestyle

Interviewbogen EventAssistant

Soll ein Raster für das Interview liefern, ebenso gewisse Leitfragen welche wir uns durch das Interview leiten sollen.

Einleitung

- Vertraulichkeit hinweisen
- Intro wieso dieses Interview
- Fokussieren auf Episoden welche weniger als 3 Wochen alt sind
- Identifizieren: Ziele, Probleme, Werkzeuge (auch Notizzettel sind Werkzeuge), Strategien („Tricks“)
- Hinter „Tech“ Intentionen und Ziele schauen
- Interpretationen können falsch sein (=> testen)

Allgemeines Profil

- Alter (schätzen):
- Beruf:
- Wie oft werden Events besucht (In der Art des aktuellen Events, was für andere im letzten Jahr?):
.....
- Wie viel Geld geben Sie für Anlässe dieser Art aus?
- Technische Kenntnisse usw.
 - Wie oft benutzen Sie Ihren Computer/ Wie schätzen Sie ihre Kenntnisse ein?
- Haben Sie einen PC/Laptop (und wie häufig nutzen Sie diesen)?
- Haben Sie einen E-Mail Account (und wie häufig nutzen Sie diesen)?
- Haben Sie ein Facebook Account (und wie häufig nutzen sie diesen)?

Handyprofil / Nutzung

Fragen zum Telefon und der Nutzung

- Zeigen Sie Bitte uns mal Ihr Telefon
Modell und OS:
- Sind Sie zufrieden? Kaufen Sie bald ein neues?
- Welches Mobile möchten Sie sich als nächstes kaufen?
.....
.....
- Haben Sie ein Datenabo? / Ist mobiles Internet ein grosses Thema
.....

- Lieblingsapps (ohne Spiele), welche Apps sind auf dem Startscreen?

.....
.....

- Was war die letzte App die Sie installiert hatten? Wie haben sie von der gehört?

.....
.....

- War die Installation der Applikation schwierig (warum)

.....
.....

Konkrete Fälle

Kontext: Vorbereitung für diesen Anlass

Datum:

Name des Anlasses:

Allgemeine Fragen:

- Wie haben Sie von dem Event gehört?

.....
.....
.....

- Kommen Sie mit Freunden?

.....
.....

- Haben sie etwas eingekauft für den Event?

.....
.....

- Essen /Gadgets usw?

.....
.....

- Was haben Sie für ein Budget für den Event

.....
.....

Organisation / Vorbereitung

- Kennen Sie das Programm?

.....

- Von wo?

.....

- Wieso nicht?

.....
.....

- Festführer? (Schon gekauft, benutzt wenn gratis, würde benutzen)

.....
.....

- War das ein Vorteil?

.....

- Wie teuer für ein Festführer auf Papier?

.....

- Haben Sie eine Kamera (Foto/Video/Handy) mitgenommen?

.....

Tickets

- Wie haben Sie die Tickets für den Event gekauft?
.....
- War das Organisieren des/der Tickets schwierig (warum)?
.....
.....
- Haben Sie schon mal ein elektronisches Ticket gekauft (z.B. SBB; Flug) Was sind Ihre Erfahrungen?
.....

Kollegen / Freunde

- Haben Sie sich abgesprochen? (Wie? Mail, Telefon, SMS, FB-Chat?)
.....
.....
- Haben Sie etwas verabredet falls Sie sich verlieren?
.....
- Kennen Sie alle Personen in Ihrer Gruppe hier?
.....
- Würden Sie auch gehen, wenn Sie nicht alle kennen würde / Keinen kennen würden?
.....
 - Wie würden Sie sich verabreden wollen?
.....

Anreise

- ÖV:
 - Würden sie die Bushaltestelle wiederfinden?
.....
 - Fahrplan ausgedruckt? Oder schon vorher über in und Rückfahrt informiert?
.....
.....
- Auto / Velo
 - Parkplatzsuche und auch wieder finden am Schluss?
.....
- Alternative?
.....

Kontext: Letzter Anlass

Datum:

Name des Anlasses:

Benachrichtigungen

- Woher wussten Sie, dass es am Stand XYZ eine Happy-Hour gibt?
.....
.....
 - Würden Sie eine entsprechende Benachrichtigung (gratis/nicht gratis) abonnieren?
gratis:
kostenpflichtig:

Freunde / Soziales

- Haben Sie sich mal verloren an einem Event? Wie ging das aus?
.....
.....
.....
.....
.....
- Würden Sie sich freuen, wenn Sie sehen könnten wo sich Ihre Freunde am Anlass direkt aufhalten?
.....
.....
 - Sind Sie schonmal irgendwo aufgewacht und wussten nicht wo sie sind / Ihre Freunde wussten auch nicht wo Sie sind?":
.....
 - Würden Sie wollen, dass all Ihre Freunde (z.B. Facebook Freunde) am Event sehen, *dass* sie sich am Event aufhalten?
.....
.....
 - Würden Sie wollen, dass all Ihre Freunde (z.B. Facebook Freunde) am Event sehen, *wo* sie sich am Event aufhalten?
.....
.....

Informationen

- Fühlten Sie sich vom Veranstalter gut über das Geschehen am Anlass informiert?
.....
.....
- Konnten Sie sich selber über wichtige Dinge Informieren? Wie, erzählen Sie!
.....
.....
.....
- Haben Sie die Ausschilderungen genutzt? Wieso nicht?
.....
.....

.....

.....

- Ablaufplan / Timetable

- Auswendig gelernt?:
- Haben Sie es geschafft, rechtzeitig an der Bühne/anderer Ort zu sein?

- Haben Sie schon mal was verpasst? Was war da passiert?

- Erzählen Sie: Könnte man das verbessern?

- Haben Sie Fotos gemacht (und anderen gezeigt)?

.....

.....

Navigation

- An was haben Sie sich orientiert? Wegweiser, Pfeile, Grosse Tafeln, den Leuten Nachgelaufen

- Haben Sie das Auto wieder gefunden

- Haben Sie Ihr Zelt wieder gefunden

- Woher wussten Sie an Ihrem letzten Openair, wo die Stimmung gerade am besten war?

.....

.....

Konsumation

- T-Shirt, Essen, Getränke

Kontext: Nach dem Anlass

Datum:

Name des Anlasses:

- Haben Sie Fotos / Filme gemacht, publiziert

.....
.....

- Bekanntschaften wieder gefunden? (Wo, Facebook, Telefon-Nr, Mail-Adresse)

.....
.....

- Wo? FB, Tel-Nr.

- Unbekannte Personen ausfindig machen

.....
.....

- Austausch über den Anlass?

.....
.....
.....

- Haben Sie Fotos zum Event sich angeschaut?

- Selbergemachte

.....
.....

- Von Kollegen (auf FB z.B.)

.....
.....

- Im Internet (Portale)

.....
.....

Kontext: Der schlimmste Anlass, Warum?

Im vergangenen Jahr in dieser Art!

Was könnte besser gewesen sein?

.....

.....

.....

.....

.....

Kontext: Der Beste Anlass, Warum?

Im vergangenen Jahr in dieser Art!

Warum? Mal abgesehen von der Stimmung/Kollegen und den Bands/Darstellungen/etc.

.....

.....

.....

.....

.....

Nachbearbeitung / Nachfragen

Personalien erfassen um nach dem aktuellen Anlass noch weitere Informationen erhalten zu können.

.....

.....

.....

.....

.....

Notizen

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Anhang E

Interviewauswertung Freestyle

Interviewauswertung Freestyle

Interview Ort:	Zürich vor Landiwiese (Haupteingang Freestyle.ch)
Datum:	Sonntag, 26. September 2010
Vorgehen:	Es wurden nur Personen angesprochen, die sich ausserhalb des eintrittspflichtigen Geländes aufgehalten haben.
Zielgruppe:	Es wurden gezielt Personen gesucht, die Handys mit sich tragen und diese auch regelmässig nutzen.

Allgemeines Profil

Die **Befragten** waren schätzungsweise alle in einem Range von **16 – 30 Jahren**. Diese hielten sich oft in **2-5er Gruppen** auf. Diese besuchen in der Regel **1 - 5 Events im Jahr**. Die einen nur 1-2 und die anderen 3-5 Event pro Jahr.

Die Selbsteinschätzung betreffend **Computerkenntnissen** ergab bei allen Werte von gut **bis sehr gut**.

Einige geben von **pro Event** (konkretes Beispiel Openair) **bis zu 300 Franken** aus.

Eine ältere Person (**Familienvater**) konnten wir dennoch interviewen, welcher den Event nur seines **Sohnes** zuliebe besuchte. Dieser Vater besass aber auch **sehr gute Computerkenntnisse**.

Handyprofil / Nutzung

Bei den aktuellen **Handys** wurden hauptsächlich **IPhone** oder ein **Sony Ericsson** genannt. Aber auch einige **sehr alte Nokias** waren auszumachen. HTC, Samsung, Windows Mobile und LG waren nur **selten anzutreffen**. Einige meinten allerdings, dass sie in Kürze ein neues Handy anschaffen würden, welches zumindest einen Touchscreen besitzen würde.

Die Leute die ein **Touch-Handy** besitzen hatten auch ein **Datenabo**, welches ihnen **regelmässige Nutzung des Internets** ohne zusätzliche Kosten ermöglicht. Hier gab es aber **vereinzelt Kommentare**, dass die **500 MB im Monat viel zu wenig** wären.

Die **HauptApps** die verwendet wurden waren **SBB, Facebook, Xtra-Zone, News** (20min, TA usw.) und natürlich **viele Spiele**.

Vorbereitung auf Freestyle.ch

Die meisten **besuchten den Anlass** schon aus **Tradition**, andere sind durch Plakate, Zeitung oder über **Kollegen** auf den Anlass aufmerksam geworden.

Wir haben **niemanden** getroffen, der **alleine an diesem Anlass** unterwegs war. Nur welche die **alleine auf Andere gewartet haben**.

Auf die Frage **ob** sie den **Ablauf des Events kennen**, meinten die Meisten **nein**, denn dies sei auch nicht so wichtig, oder **einer** in der Gruppe weiss so einen **groben Ablauf**.

Personen die sich für das **Programm interessierten**, haben sich hauptsächlich über ein **Plakat** oder ein **Flyer** informiert. Besucher die genaueres wissen wollten, haben sich mittels **Internet** oder der **iPhone App** informiert. Ansonsten war in jeder Gruppe immer einer dabei der in etwa wusste was wann wo lief.

Als Andenken an den Anlass machten die meisten mit dem **Handy Fotos**. Denn viele haben ihre **Digitalkamera zuhause vergessen**, mit dem Kommentar: „Die **Auflösung** ihres **Handys** sei **ziemlich gut**.“.

Die meisten haben ihr **Ticket** über den **SBB-Schalter** oder **Online** über **Starticket** gekauft. **Elektronische Tickets** haben die meisten Befragten schon einmal gekauft. SBB ist für Zugreisende ein Vorteil, da es nur da Kombi (Zugreise und Ticket)-Angebote gibt.

Verabredet haben sich die Meisten über **Telefon** oder **Facebook**.

Falls sie sich am **Anlass verlaufen**, würden sie dies mit **telefonieren** lösen.

Die meisten wären bereit auch mit **Fremden** einen Anlass besuchen, sofern **sie ein bis zwei Leute in der Gruppe kennen** würden.

Die meisten Befragten sind mit den **ÖV angereist** und **finden** die **Haltestelle ohne Probleme** wieder. Den **Fahrplan** sei ihnen bereits etwas **geläufig**, oder es fahren so **oft** welche, dass es bestimmt **keine Probleme** gäbe.

Letzter Anlass

Hier wurden häufig folgende Anlässe genannt:

- Freestyle.ch (Samstag)
- Zürifest
- Openair Frauenfeld

Über spezielle und aktuelle Ereignisse wurde am **Freestyle** über **Speaker** informiert. Am **Frauenfelder sprach sich herum** wie wann wo die Post abgeht. Und am **Zürifest** war man komplett auf sich alleine gestellt. Hier half nur sich **frühzeitig** über das **Internet** zu **informieren**. Die **Befragten** wären aber **erfreut** darüber wenn man per **Pushnachricht** darüber **informiert** werden würde.

Wenn **jemand verloren** ging an einem Event, haben sie bei einem **Platz abgemacht** oder einfach **telefoniert**.

Das Interesse an einer **Statusmeldung** auf z.B. Facebook, wer von Ihren **Freunden** sich **direkt** am **Event befindet**, wurde mit gemischten Gefühlen aufgenommen. Einige fänden dies **eine gute Idee**, andere eher nicht, es wurden auch **Bedenken bezüglich Privatsphäre** ausgesprochen.

Bei der Information an einem Anlass wurden alle **genügend** durch den **Veranstalter informiert**. Dennoch haben einige schon gewisse **Gig's** einiger **Band verpasst**. Einige meinten sogar, hier könne das **Handy** bestimmt **helfen**.

Fotos wurden fast von jedem **Befragten** schon **geschossen** an einem Anlass.

Das **Zelt** oder ein **sonstiger Punkt** wurde **immer gefunden**, dies **dauerte** aber je nach **Zustand** der Person etwas **länger**, oder **weniger lang**.

Nach dem Anlass

Hier wurden die **Aussagen** meist **nicht zu konkreten Anlässen** gemacht, sondern eher allg. zu verschiedenen. Dennoch tauchte auch hier **Zürifest** des Öfteren auf.

Filme und Fotos wurden, **wenn überhaupt auf Facebook** publiziert. Doch die meisten behalten die Fotos einfach für sich oder zeigen sie dem Kollegen auf dem Handy.

Von **Veranstaltern**, werden **sehr selten die Portale auf Fotos** abgeklappert. Höchstens werden die **Fotos des Kollegen** über **Facebook** abgerufen.

Über den Anlass wird höchstens noch **kommende Woche** ein wenig **diskutiert**, ansonsten wird sich **kaum ausgetauscht**.

Schlechtestes Event

Hier wurde darauf vermerkt, dass die **meisten Events** an die als **Besucher** gehen **gut wären**, ansonsten wären sie nicht an dem Entsprechenden Anlass.

Doch eine Gruppe hat sich darüber geäußert, dass ein Anlass **zu gross** gewesen wäre und dadurch **unübersichtlich gewesen** wäre. Hier wäre eine **bessere Ausschilderung** oder etwas Ähnliches sehr hilfreich gewesen.

Andere Negativpunkte waren

- Kindergarten
- Schlechte Musik

Bester Event

Zürifest, alte Kollegen getroffen

Freestyle

- Interesse
- Gute Atmosphäre

Frauenfeld

- Gute Bands

Smelt

- Kein Kindergarten
- Gute Atmosphäre

Bemerkungen

Interessant war, dass wir doch auf zwei Kleidermarken Promoter getroffen sind, die nur am Freestyle waren um ihr Produkt zu vermarkten. Denn es ist ein grosser Markt, denn fast jeder kauft ein Erinnerungsstück an einen solchen Anlass.

Alle Befragten waren sehr freundlich und waren bereit an unserem Interview teilzunehmen.

Anhang F

Interviewfragebogen Züspa

Interviewbogen EventAssistant

Einleitung

- Vertraulichkeit hinweisen
- Intro wieso dieses Interview
- Fokussieren auf Episoden welche weniger als 3 Wochen alt sind
- Identifizieren: Ziele, Probleme, Werkzeuge (auch Notizzettel sind Werkzeuge), Strategien („Tricks“)
- Hinter „Tech“ Intentionen und Ziele schauen
- Interpretationen können falsch sein (=> testen)

Fragen zum Anlass

Allgemeine Fragen (6)

- Woher kennen Sie den Anlass?
.....
.....
- Wie sind Sie angereist / finden Sie wieder zurück?
.....
- Kommen Sie mit Freunden?
.....
- Wie haben Sie sich verabredet?
.....
.....
- Haben Sie etwas verabredet falls Sie sich verlieren?
.....
- Haben Sie eine Kamera (Foto/Video/Handy) mitgenommen?
.....

Organisation (2)

- Kennen Sie das Programm?
.....
 - Von wo?
.....
 - Wieso nicht?
.....
.....

- Festführer? (Schon gekauft, benutzt wenn gratis, würde benutzen)

.....

.....

Hypothetische Fragen (10)

Kollegen / Soziales

- Haben Sie sich mal verloren an einem Event? Wie ging das aus?

.....

.....

.....

- Würden Sie sich freuen, wenn Sie sehen könnten wo sich Ihre Freunde am Anlass direkt aufhalten?

.....

.....

Benachrichtigungen

- Konnten Sie sich selber über wichtige Dinge Informieren? Wie, erzählen Sie! (Wie hat der Veranstalter geholfen))

.....

.....

- Ablaufplan / Timetable: Auswendig gelernt?

- Haben Sie es geschafft, rechtzeitig an der Bühne/anderer Ort zu sein?

.....

.....

- Haben Sie schon mal was verpasst? Was war da passiert?

.....

.....

- Erzählen Sie: Könnte man das verbessern?

.....

.....

Navigation

- An was haben Sie sich orientiert? Wegweiser, Pfeile, Grosse Tafeln, den Leuten Nachgelaufen

.....

.....

- Haben Sie das Auto/Zelt wieder gefunden

.....

.....

- Woher wussten Sie an Ihrem letzten Openair, wo die Stimmung gerade am besten war?

.....

.....

Nach einem Anlass

- Haben Sie Fotos / Filme gemacht, publiziert

.....

.....

- Bekanntschaften wieder gefunden? (Wo, Facebook, Telefon-Nr, Mail-Adresse)

.....

.....

- Haben Sie Fotos zum Event sich angeschaut?

- Selbergemachte
- Von Kollegen (auf FB z.B.)
- Im Internet (Portale)

Der schlimmste/beste Anlass?

Im vergangenen Jahr in dieser Art!?

.....

.....

.....

Mobile Support

Welches nützliche Feature könnten Sie auf dem Handy vorstellen?

.....

.....

.....

.....

.....

Nachbearbeitung / Nachfragen

Personalien erfassen um nach dem aktuellen Anlass noch weitere Informationen erhalten zu können.

.....

.....

.....

Handyprofil / Nutzung

Fragen zum Telefon und der Nutzung

- Zeigen Sie uns bitte mal Ihr Telefon
Modell und OS:
- Sind Sie zufrieden? Kaufen Sie bald ein neues?
- Haben Sie ein Datenabo? / Ist mobiles Internet ein grosses Thema
.....
- Lieblingsapps (ohne Spiele) - (welche Apps sind auf dem Startscreen)?
.....
.....
- Was war die letzte App die Sie installiert hatten? Wie haben sie von der gehört?
.....
.....

Anhang G

Interviewauswertung Züspa

Interviewauswertung Züspa

Interview Ort:	Zürich vor Messehalle (Haupteingang Messe)
Datum:	Samstag, 2. Oktober 2010
Vorgehen:	Es wurden nur Personen angesprochen, die sich ausserhalb des eintrittspflichtigen Geländes aufgehalten haben.
Zielgruppe:	Es wurden gezielt Personen gesucht, die Handys mit sich tragen und diese auch regelmässig nutzen.

Allgemeine Fragen

Die **Befragten** waren schätzungsweise alle im Altersrange von **20 – 35 Jahren**. Diese hielten sich oft in **2-4er Gruppen** auf. Diese besuchen in der Regel **2 - 3 Events im Jahr**.

Die Züspa entpuppte sich als ein sehr **traditioneller Event**, denn die interviewten **Besucher** besuchten die Züspa schon **mehrere Male**.

Die **Anreise** beschränkte sich auf **Auto** oder **ÖV**, die Anzahl war ausgeglichen.

Verabredet haben sich alle per **Telefon**.

Im Falle eines **Orientierungsverlustes**, würden sich die **Personen** per Telefon **kontaktieren**.

Auch hier wurden **nicht bewusst Fotokameras** mitgenommen, **Handykameras** reichten völlig aus.

Organisation

Der **Ablauf** des **Events** war den meisten nicht so wichtig, denn die Züspa ist eher ein Anlass an dem man durch die **Stände schlendert**.

Aber wenn Sie den **Ablauf** etwas **genauer wissen** wollten hätten sie das **Internet konsultiert** oder den **Messeführer**.

Hypothetische Fragen

Wenn sich die Leute schon mal **verloren** hatten, liessen sie **sich ausrufen**, haben einen **fixen Treffpunkt** ausgemacht oder **telefonierten** einfach.

Auf die Frage, ob es **interessant wäre Kollegen direkt** auf dem **Handy** sehen würden, welche auch an dem **Anlass** sind. Meinten sie könnte noch interessant sein.

Informiert über einen **Anlass** hat man sich im **Voraus**, wenn es wichtig war, ansonsten nahm man es gerade, wie es kam.

Ihnen ist es dennoch schon **passiert**, dass sie eine **Gig**, oder eine **Aktivität verpasst** haben, weil sie länger **hocken** geblieben sind, oder einfach den **falschen Ort** aufgesucht haben. Um dies zu **verhindern** würden die Interviewten, sich einen **Wecker stellen**.

Die **Orientierung** an **Anlässen** geschah hauptsächlich über **Schilder** oder **Festpläne**.

Ihre **Zelte** konnten alle wieder **finden**, dies ist zwar **sehr abhängig von Alkoholpegel oder Anzahl parkenden Autos**.

Die **besten Bands** und die **beste Stimmung** konnten sie immer anhand des **Lärmpegels** ausmachen.

Wenn die **Interviewten** nicht gerade zu faul waren um **Fotos zu schiessen** haben sie diese aber nur auf der **eigenen Festplatte** gespeichert.

Auf **Internetportalen** wurden wenn von einem ein **Portrait geschossen** wurde, eventuell nach dem **Portrait Ausschau** gehalten.

Bester Anlass

Als beste **Anlässe** wurden das **Coldplaykonzert**, das **Churerfest** und das **Zürifest** genannt. Da diese **gemütlich, gut ausgeschildert**, in der Nähe von **Bahnhöfen** waren. **Zu viele Leute** wurden meist als **negativ** empfunden.

Mobile Support

Auf direkte Befragung was sie sich in einer solchen **mobilen Applikation** wünschen würden kam die Antwort:

- Punkt setzen, dahin Navigieren
- In welchem Parkhausstock ein freier Parkplatz ist
- Einen Reminder, welche an wichtige Events erinnert
- Wie weit die nächste Tram/Bus/Zugstation entfernt ist

Handyprofil

ALLE befragten **Besucher** hatten ein **iPhone** oder waren kurz **davor sich eines zu kaufen**. Sie sind auch sehr zufrieden damit.

Die **Daten** die in ihrem **Abonnement** dabei sind **reichen** normalerweise aus.

Als die **meistverwendeten** Apps wurden: **SBB-Fahrplan, Einkaufshelfer, Newsservice, Search** und **Spiele** genannt.

Bemerkungen

Wir haben zusätzlich noch ein **Interview** mit der **Infostandfrau** durchgeführt. Sie hat uns mitgeteilt, dass an einem **Event** wie die **Züspa** die Fragen zu **75%** von **älteren Leuten** stammen die einen **bestimmten Stand** suchen. Zu **25%** wird nach der **Toilette** oder einem **ähnlichen Ort** gesucht.

Aber an einer **Fachmesse** werden aber **hauptsächlich keine Fragen** gestellt, die Leute wären denn da schon so gut **im Voraus informiert**. Auch ist der Informationsstand bei Fachmessen meist nicht besetzt, da die Leute sehr gut Informiert sind.

Anhang H

Installationsanleitung LaTeX

Installationsanleitung LaTeX

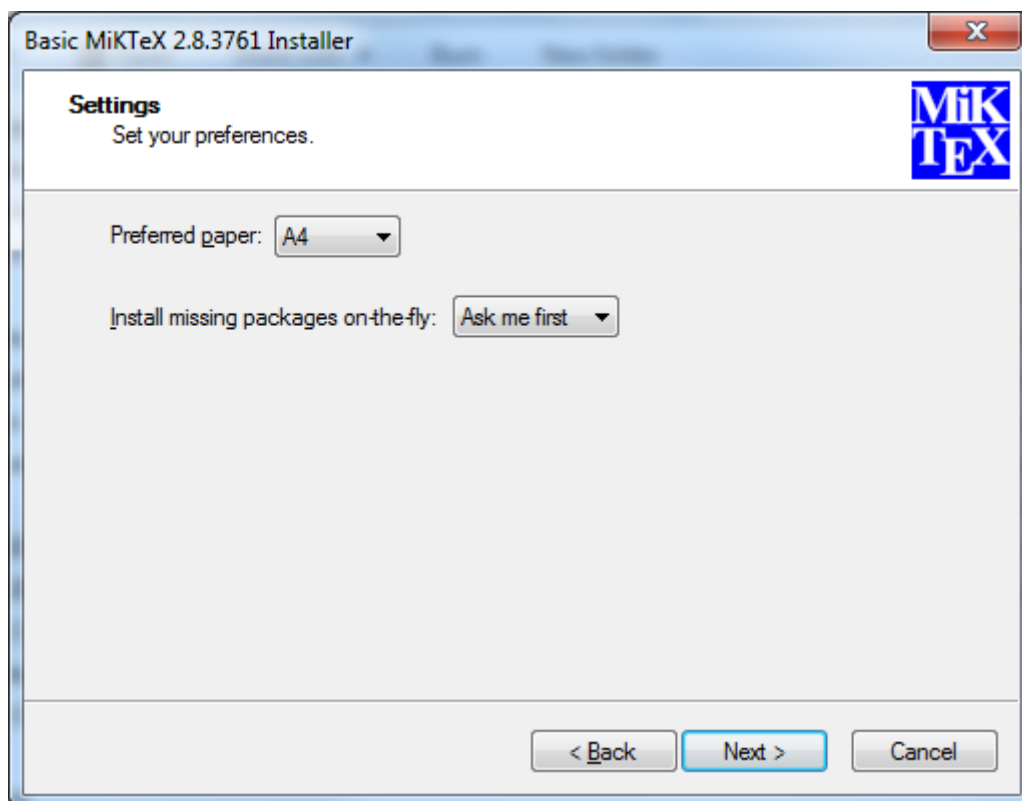
Installation MiKTeX

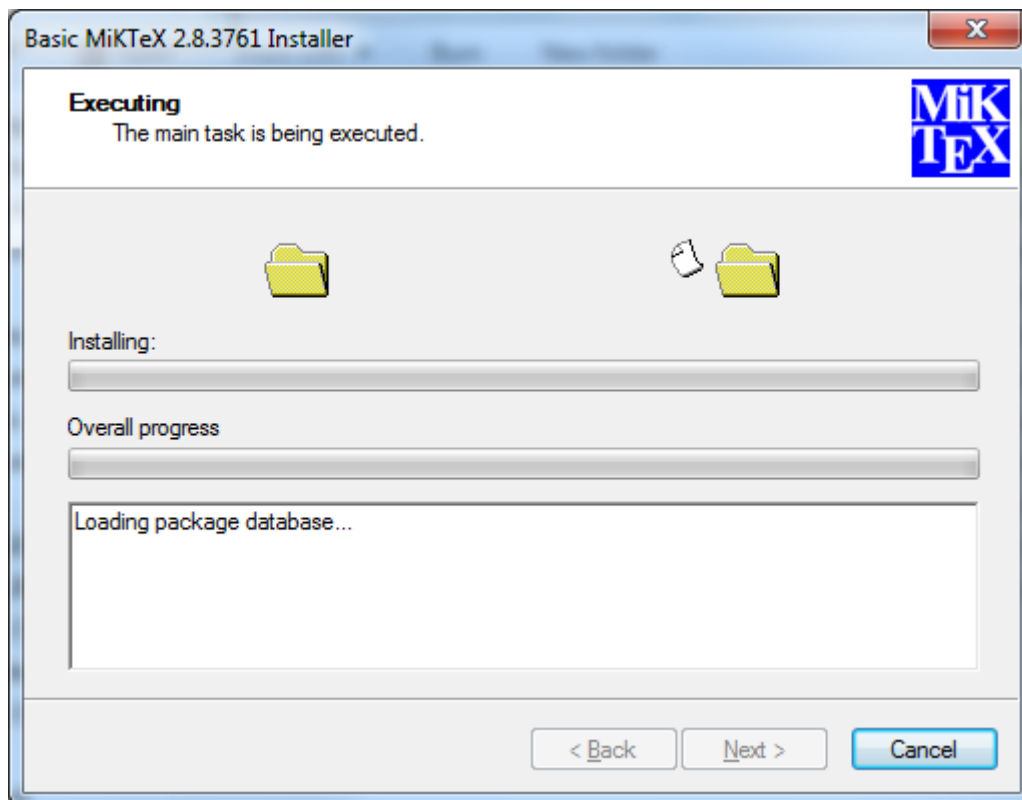
Dieses Packet enthält alle Informationen damit das .tex-Dokument in verschiedenen Schritten in ein PDF umgewandelt werden kann. Stellt die Notwendigen Library's und zusätzlichen Fancy-Addons zur Verfügung.

Downloadlink:

<http://miktex.org/2.8/setup>

Installieren wie folgt





Installation TEXnicCenter

Dies ist eine grafische IDE um das arbeiten mit dem Textdokument zu erleichtern.

Downloadlink

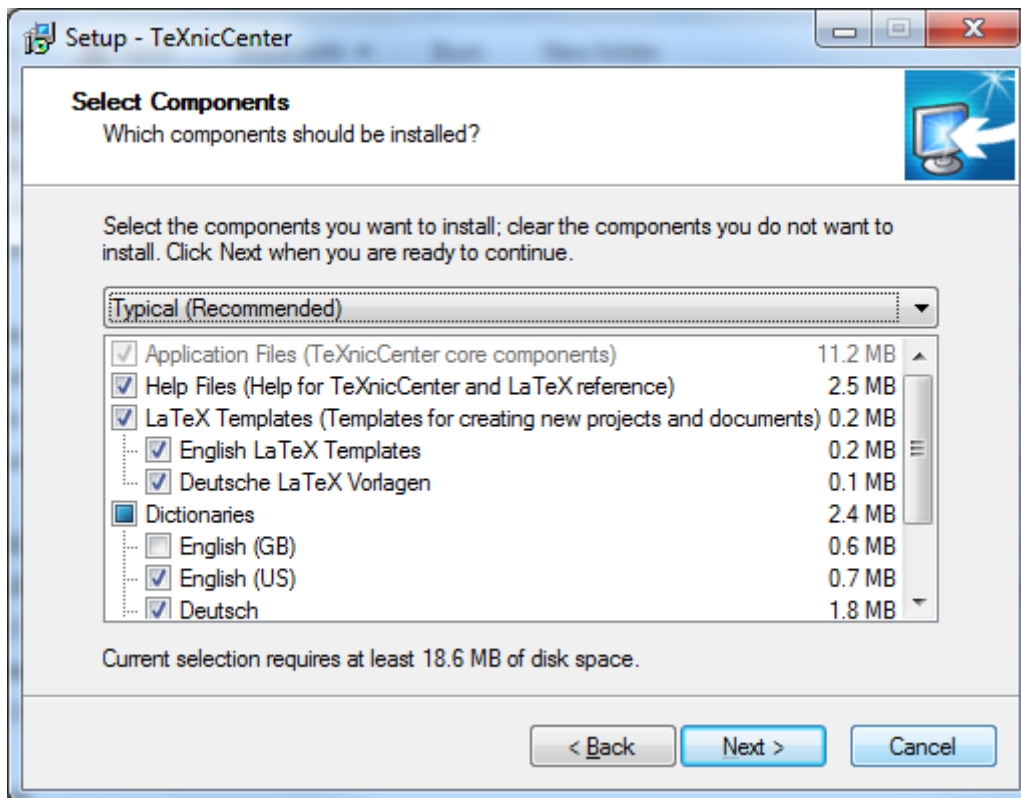
<http://www.texniccenter.org/resources/downloads/29>

Auswählen

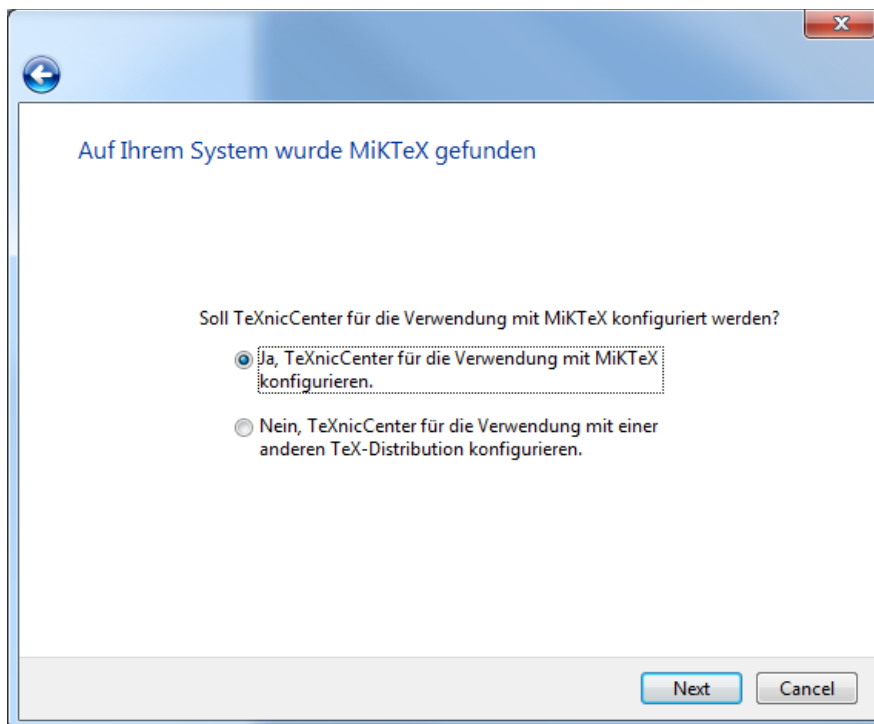
TeXnicCenter Alpha Installer

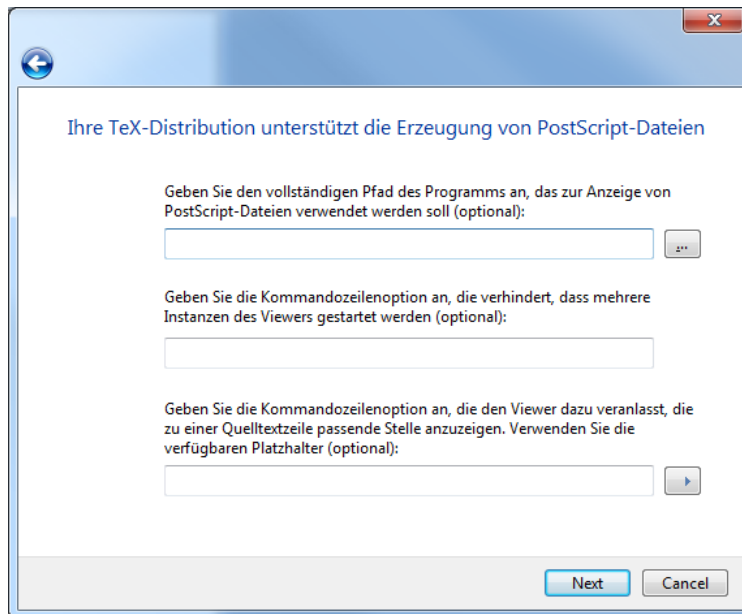
Direktdownload

http://sourceforge.net/projects/texniccenter/files/TeXnicCenter%20Alpha/TXCSetup_2Alpha3.exe/download

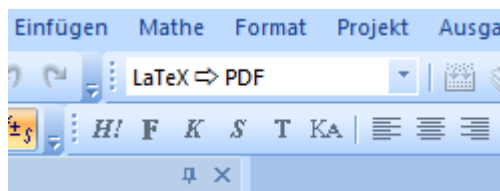
Installieren wie folgt:**Starten und Anwenden**

Nun ist das TeXnicCenter bereit zum Start. Beim ersten Start wird nachgefragt ob nun MiKTeX verwendet werden soll.

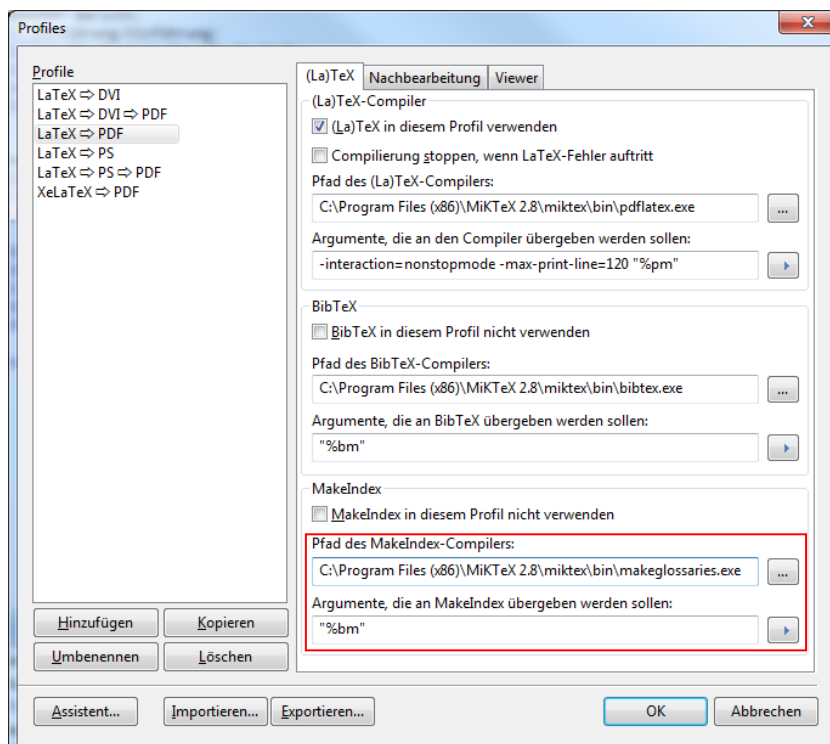




Nun nur noch auswählen, dass ein PDF erzeugt werden soll und alles ist gut!



Ausgabeprofil anpassen (**ALT + F7**)



Für den Glossar muss noch ActivePerl installiert werden.

Download unter:

<http://www.activestate.com/activeperl/downloads>

Version 5.12.2.1203 für Windows (64-bit, x64)

Installieren & Viel Spass mit LaTeX!

Anhang I

Papierprototype Nr. 1

Start screen

→ Touchscreen
→ Menubutton

Menu
→ Settings
→ Push Konfigurationen
→ Location 7x pro
→ FB



V2

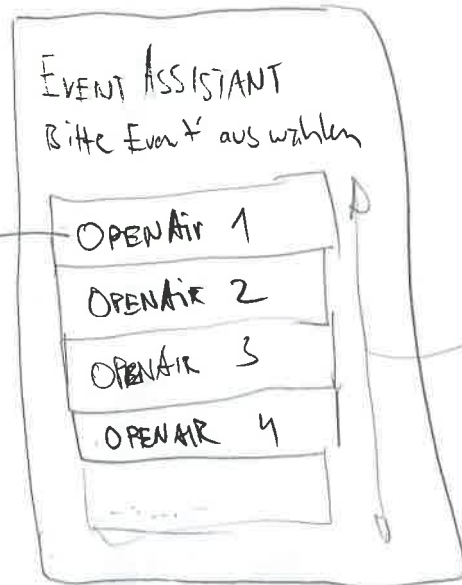
Form fields and buttons:

- Label: "Name"
- Input field: "Hans FACEBOOK" (with a checkmark to its right)
- Label: "Password"
- Input field: "x x x x x"
- Buttons: "Anmelden" and "Seite"

Am Event



Nicht am Event



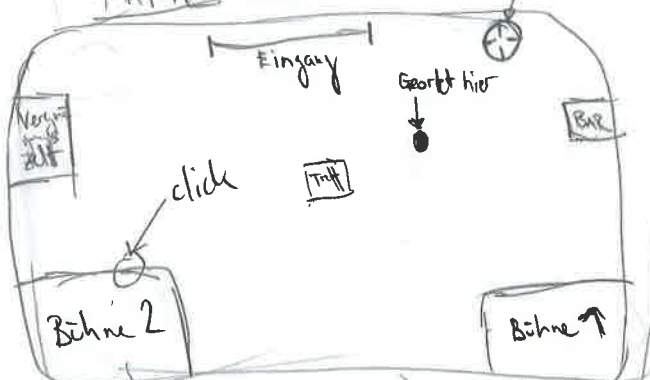
Scrollable

Freunde Anzeige welche am Event sind.

Versucht über FB zu orten!

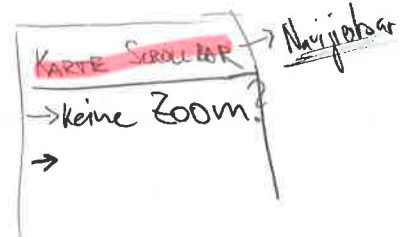
KARTE

Querschnitt
Prüfbar?



Schedule Bühne 2

Zeit	Musik	...
19:00	Rocked E	
20:00	Budda	



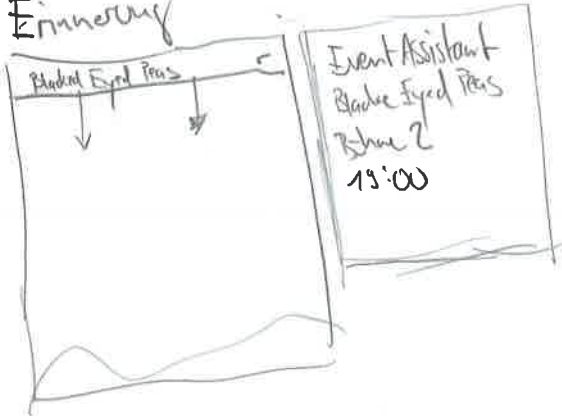
mit Open Air

Android != iPhone

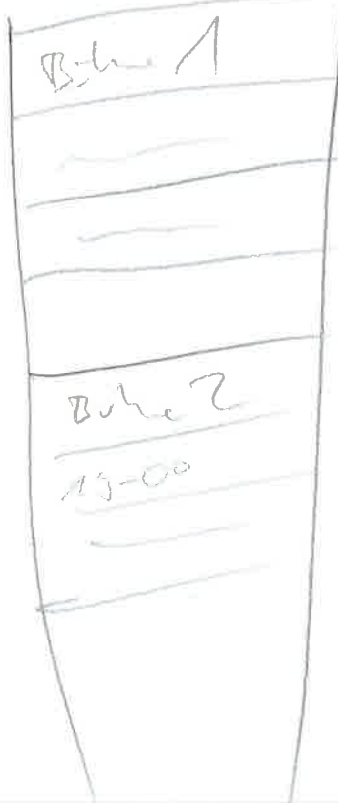
→ Menu taste
Wo Einbauen?

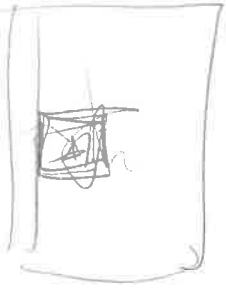
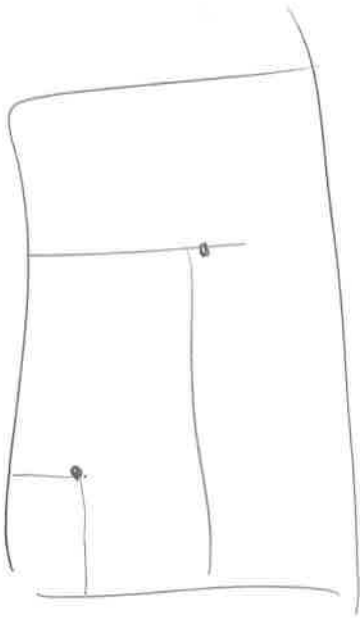


Erinnerung



Push





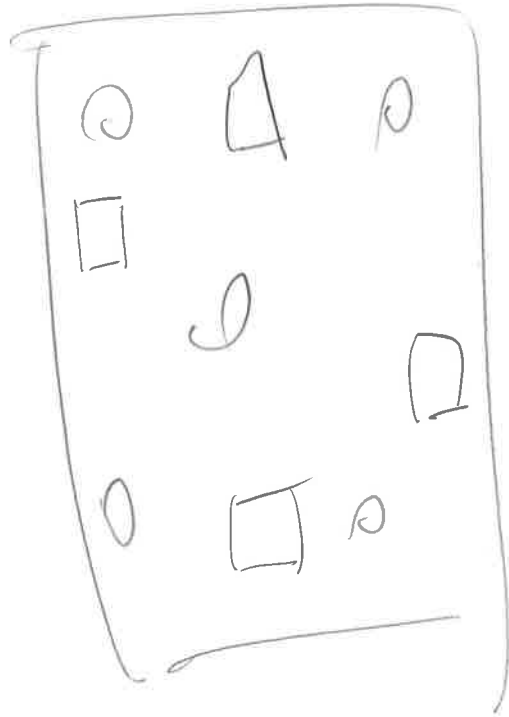
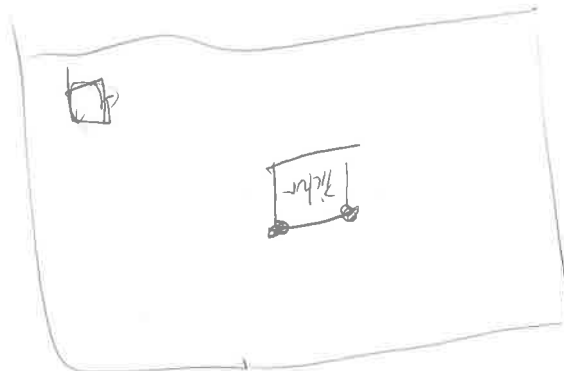
PO1

2 Varianten
von Karte

Freie Gestaltung

- PO1 Koordinaten
ex. Long/Lat

PO1



Anhang J

Android Coverage Report

Android Coverage Report

Felix Egli 11.12.2010 Version 1

Coverage-Tool

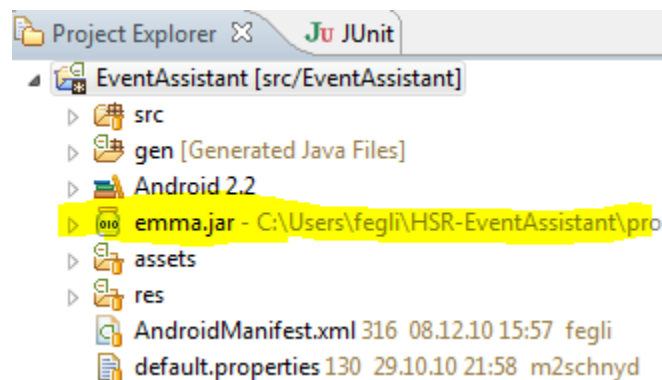
Emma (<http://emma.sourceforge.net/intro.html>)

Das Emma Code-Coverage Tool kann die Tests von Android instrumentieren und einbinden. Dies wird von der Android ADB (Android Debugger Bridge) alphanumatisch unterstützt.

Wir haben versucht das Ganze für unser Projekt einzubauen, was auch teilweise funktioniert hat.

Vorgehensweise:

1. Als erstes muss in das Projekt, das getestet werden soll das emma.jar eingebunden werden.
 - a. Dies befindet sich im Ordner %ANDROID_SDK%/tools/lib/emma.jar



2. Das Testprojekt, wie auch das Hauptprojekt müssen auf den Emulator (Mobile) installiert werden. Dies wird am einfachsten erreicht, wenn man einfach das Testprojekt laufen lässt um die Tests auszuführen.

```
Android
[2010-12-11 15:42:57 - EventAssistant-Test] -----
[2010-12-11 15:42:57 - EventAssistant-Test] Android Launch!
[2010-12-11 15:42:57 - EventAssistant-Test] adb is running normally.
[2010-12-11 15:42:57 - EventAssistant-Test] Performing android.test.InstrumentationTestRunner JUnit launch
[2010-12-11 15:42:57 - EventAssistant-Test] Automatic Target Mode: using existing emulator 'emulator-5554' running compatible AVD 'test'
[2010-12-11 15:42:57 - EventAssistant-Test] Uploading EventAssistant-Test.apk onto device 'emulator-5554'
[2010-12-11 15:42:57 - EventAssistant-Test] Installing EventAssistant-Test.apk...
[2010-12-11 15:43:00 - EventAssistant-Test] Success!
[2010-12-11 15:43:00 - EventAssistant-Test] Project dependency found, installing: EventAssistant
[2010-12-11 15:43:00 - EventAssistant] Uploading EventAssistant.apk onto device 'emulator-5554'
[2010-12-11 15:43:01 - EventAssistant] Installing EventAssistant.apk...
[2010-12-11 15:43:05 - EventAssistant] Success!
[2010-12-11 15:43:05 - EventAssistant-Test] Launching instrumentation android.test.InstrumentationTestRunner on device emulator-5554
[2010-12-11 15:43:06 - EventAssistant-Test] Collecting test information
[2010-12-11 15:43:14 - EventAssistant-Test] Sending test information to Eclipse
[2010-12-11 15:43:14 - EventAssistant-Test] Running tests...
[2010-12-11 15:43:31 - EventAssistant-Test] Test run complete
```

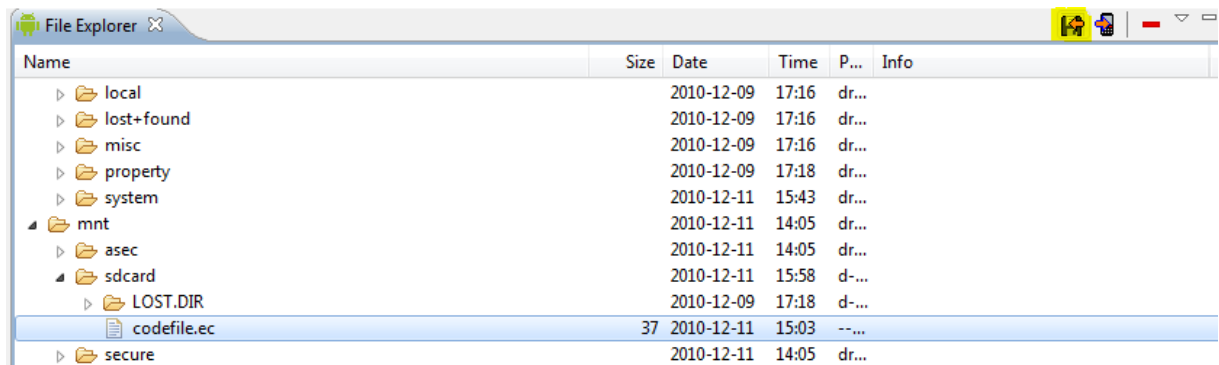
3. Über die DDMS kann nachgeschaut werden, ob die Files richtig installiert wurden.

Name	Size	Date	Time	P...	Info
data		2010-12-09	17:17	dr...	
app		2010-12-11	15:46	dr...	
ch.hsr.eventassistant-1.apk	1174816	2010-12-11	15:43	-r...	ch.hsr.eventassistant
ch.hsr.eventassistant.test-1.apk	21102	2010-12-11	15:42	-r...	ch.hsr.eventassistant.test
app-private		2010-12-09	17:16	dr...	

4. Nun kann über die ADB-Shell das Test-Instrumentieren und Codecoverage-Generation gestartet werden. Dies geschieht mit folgendem Befehl.

```
adb shell am instrument -w -e coverage true -e coverageFile  
/sdcard/codefile.ec  
ch.hsr.eventassistant.test/android.test.InstrumentationTestRunner
```

5. Anschliessend ist das Coveragefile im DDMS-Fileexplorer ersichtlich. Dieses kann anschliessend darüber auf die lokale Maschine heruntergeladen werden.



6. Nun kann es über das emma.jar in ein HTML Output umgewandelt werden. Dies geschieht über folgenden Befehl:

```
java -cp emma.jar emma report -r html -in coverage.ec
```

Probleme

Unser generiertes Coveragefile „codefile.ec“ war gerade mal 37 Bytes gross. Auch nach dem Ausführen auf der lokalen Maschine mit dem emma.jar Reportgenerator ergab, dass anscheinend keine Metadaten im ec-File vorhanden waren.

Conclusion

Die Dokumentation über dieses Flag, hält sich sehr in Grenzen. Es gibt nur erwähnt, dass es ein solches Flag gibt.

Auch die Community in den Googlegroups spricht nur von „Bastelbeispielen“ welche alle soweit funktionieren wie oben beschrieben. Die meisten Leute melden ähnliche Probleme, aber ohne konkrete Antwort.

Quellen:

http://groups.google.com/group/android-developers/browse_thread/thread/43cf8a8ca5662f85

<http://osdir.com/ml/AndroidDevelopers/2009-08/msg02689.html>

<http://developer.android.com/reference/android/test/InstrumentationTestRunner.html>

<http://emma.sourceforge.net/intro.html>

Anhang K

Systemtest Vorlage

Systemtest

Systemtest Testfälle Version: 4

Tester: _____

Testdatum: _____

Testgerät: _____

Android-Version: _____

Es wird nur mit dem HSR-Openair gearbeitet, da hier die Testdaten vollständig erfasst sind.

ID	Aktion	Erfolgreich?	Bemerkung
1	EventAssistant lässt sich über das Symbol starten.		
2	Applikation stürzt nicht ab wenn keine Daten vorhanden sind und sie ohne Internet (WLAN & GPRS) gestartet wird		
3	Applikation stürzt nicht ab wenn keine Daten vorhanden sind und sie mit Internet (WLAN & GPRS) gestartet wird		
4	Man befindet sich an einem Event. Der Event wird automatisch erkannt und der Event gestartet		
5	Wenn 4 erfüllt ist mit dem Back-Button in die Liste der Events		
6	Alle Tabs lassen sich aufrufen und Daten werden angezeigt.		
7	Bei plötzlicher Unterbrechung der Verbindung stürzt die Applikation nicht ab.		Lässt sich schwer gezielt testen.
8	Wenn 6 erfüllt ist und die Applikation geschlossen wird und alle Internetverbindungen getrennt werden. Müssen die Daten vom letzten Mal trotzdem angezeigt werden.		
9	Wenn 8 erfüllt muss der Offlinemodus angezeigt werden. (Oranger Balken)		
10	Wenn 9 erfüllt, Applikation mit Zurückbutton verlassen und Internet einstellen und Applikation über Icon wieder starten und Oranger Balken müsste nach spätestens 15 Sekunden verschwinden.		Kann Device und Empfangsabhängig sein.
11	Splashscreen wird kurz angezeigt beim Start.		
12	Nach dem Laden des Info-Tabs müssen Ort, Datum und Eintritt mit der Beschreibung erscheinen.		

13	Der Tab „Karte“ wird geladen. Karte wird angezeigt.		
14	Wenn 13 erfolgreich. Mit dem Klick auf die Stage Nr. 1 wird ein Beschreibungstext und die Schedules von Freitag dem 19.8.2011 und Samstag 20.8.2011.		
15	Wenn 13 erfolgreich. Mit dem Klick auf die Flossbar unten in der Mitte öffnet sich der der Flossbar Dialog mit den Schedules vom 19.8.2011 und 20.8.2011.		
16	Wenn 13 erfolgreich. Und GPS aktiviert ist wird man im Freien nach mindestens 20 Sekunden erkannt.		
17	Beim Klick auf dem Tab-Zeitplan wird der komplette Eventschedule geladen.		
18	Wenn 17 erfolgreich. Es kann auf die Tage ein und ausgeklappt werden?		
19	Wenn GPS eingeschaltet ist auf dem „Karten“-Tab. Kann über den zentrieren Button (links unten) die eigene Position ein und ausgeschaltet werden.		
20	Wenn 19 erfolgreich. Bewegungen auf dem Gelände Punkt zentriert, ausgeschaltet nicht. Die Touchpunkte (POI's Icons) sind genügend gross und sie werden auch mit Wurstfingern erkannt.		
21	Wenn 20 erfolgreich. Rote GPS-Punkt +/- 5m genau auf der Karte.		
22	Wenn 21 erfolgreich. App verlassen über Home und GPS deaktivieren und zur Karte zurück.		Der Rote Punkt wird noch angezeigt. Aber bewegt sich nicht mehr.
23	Zurückbutton Stresstest. ➔ Viele Male während dem laden der einzelnen Seiten immer wieder den Zurückbutton gehämmert. Die Applikation sollte nicht abstürzen.		
24	Wenn GPS eingeschaltet ist man sich nicht am Event befindet und der Karten-Tab offen hat, wird man nicht irgendwo ausserhalb der Karte zentriert.		

Anhang L

Systemtest Protokoll vom 19.12.2010

Systemtest

Tester: Felix Egli

Testdatum: 19.12.2010

Testgerät: Desire (HSA)
Android 2.2

Es wird nur mit dem HSR-Openair gearbeitet, da hier die Testdaten vollständig erfasst sind.

ID	Aktion	Erfolgreich?	Bemerkung
1	EventAssistant lässt sich über das Symbol starten.	✓	
2	Applikation stürzt nicht ab wenn keine Daten vorhanden sind und sie ohne Internet (WLAN & GPRS) gestartet wird	✓	Meldung erscheint keine Daten gefunden
3	Applikation stürzt nicht ab wenn keine Daten vorhanden sind und sie mit Internet (WLAN & GPRS) gestartet wird	✓	
4	Man befindet sich an einem Event. Der Event wird automatisch erkannt und der Event gestartet	1. Start nicht 2. Start ja	Kann nicht beeinflusst werden
5	Wenn 4 erfüllt ist mit dem Back-Button in die Liste der Events	✓	
6	Alle Tabs lassen sich aufrufen und Daten werden angezeigt. <i>Position angezeigt</i>	✓	
7	Bei plötzlicher Unterbrechung der Verbindung stürzt die Applikation nicht ab.		Lässt sich schwer gezielt testen. <i>gute Verbindung</i>
8	Wenn 6 erfüllt ist und die Applikation geschlossen wird und alle Internetverbindungen getrennt werden. Müssen die Daten vom letzten Mal trotzdem angezeigt werden.	✓	<i>navigations top</i>
9	Wenn 8 erfüllt muss der Offlinemodus angezeigt werden. (Oranger Balken)	✓	<i>Perfekt</i>
10	Wenn 9 erfüllt, Applikation mit Zurückbutton verlassen und Internet einstellen und Applikation über Icon wieder starten und Oranger Balken müsste nach spätestens 15 Sekunden verschwinden.	✓	Kann Device und Empfangsabhängig sein. <i>Perfekt</i>
11	Splashscreen wird kurz angezeigt beim Start.	✓	
12	Nach dem Laden des Info-Tabs müssen Ort, Datum und Eintritt mit der Beschreibung erscheinen.	✓	
13	Der Tab „Karte“ wird geladen. Karte wird angezeigt. <i>Zoom bar, scrollen x+y</i>	✓	
14	Wenn 13 erfolgreich. Mit dem Klick auf die Stage		

	Nr. 1 wird ein Beschreibungstext und die Schedules von Freitag dem 19.8.2011 und Samstag 20.8.2011.	✓	
15	Wenn 13 erfolgreich. Mit dem Klick auf die Flossbar unten in der Mitte öffnet sich der der Flossbar Dialog mit den Schedules vom 19.8.2011 und 20.8.2011.	✓	
16	Wenn 13 erfolgreich. Und GPS aktiviert ist wird man im Freien nach mindestens 20 Sekunden erkannt.	✓	
17	Beim Klick auf dem Tab-Zeitplan wird der komplette Eventschedule geladen.	✓	
18	Wenn 17 erfolgreich. Es kann auf die Tage ein und ausgeklappt werden?	✓	
19	Automatisch zentrieren ausgeschaltet + eingeschaltet	✓	sehr gut
20	Beamer auf dem Gelände Punkt zentriert, ausgeschaltet nicht	✓	sehr gut
21	Punkt $\pm 5m$ genau GPS	✓	ca. auf 5m genau
22	deaktivieren + in App zurück → Meldung ab	✓	letzter Punkt wird nach angezeigt Punkt
23	Zurückbutton Stressiert → viele male während laden immer zurückbutton gehämmert nicht abgestört	✓	

Anhang M

Sitzungsprotokolle

Weekly Meeting Wo14

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2), F. Egli (fx)

Zeit/Datum 13:10 / 22. Dezember 2010

Ort Rapperswil

Traktanden

- Protokoll -> ok
- Abstract / Management Summary -> MS für ihn
- Nutzungsrechte auf CD Kopieren anstatt licences.txt
- Aufgabenstellung unterschreiben
- Ausführbarkeit / Installationsanleitung
 - Nur Grob verweisen. Ansonsten nachfragen von SZE
- .tex-File müssen nicht auf die CD
- Inhaltsverzeichnis kurzes Review
- Video Reviewn
 - Offlinebar Annotieren mit YouTube?
 - Summary der Vorteile am Schluss des Videos
 - Mit Tags versehen auf Youtube
- Persönliche Berichte (mit Pfeffer und Zitrone)
- Swisscom in Kenntniss setzen
- Treffen im 5. Januar 2011 10:00 an HSR
- Abnahme: Inhaltsverzeichnis Bericht
- Abnahme: Review Draft Wiki Page & Video

Ausblick

- BA
 - Kinect
 - Meeting 5. Januar 15.00 Technorama

Weekly Meeting Wo13

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2), F. Egli (fx)

Zeit/Datum 13:10 / 15. Dezember 2010

Ort Rapperswil

Traktanden

- Protokoll -> ok
- Abstract:
 - "Internet Technologie" anstatt "Mobil Technologie"
 - Anderes Vorgehen zum Schreiben:
 - Was ist das Problem?
 - Warum ist es ein Problem?
 - Wie wurde es gelöst?
 - Positive Seiteneffekte

- Zuerst Bedürfnisse (Einleitungssatz)
 - zb. auch was ist ein Grossevent?
 - "Auf einem Openair sich zu verlaufen ist äusserst unangenehm..."
 - Handy Applikation für Veranstalter zu teuer, etc.
- Interessante Aspekte
 - Saubere Trennung zwischen UI, Net und Datenhaltung
 - Georeferenzierung
 - Caching / Update Mechanismus
 - Arbeitsprozess für Datenerfassung definiert
- Warum sollte es andere Informatiker interessieren?
- Kommentar zu Plakat, richtung ok oder mehr Text? Für wen ist das Plakat?
 - Ist Hauptprobe für BA.
 - Werbetafel für Entfernung, Detailtafel für Interessierte zum erklären
 - Interessierte Aspekte
 - Vorschlag: Grobes Problem (links), Technical/Learned? (rechts)
 - Nur Bullets, keine ganzen Sätze
- Auf Hauptfeatures konzentriert, Offline Modus (stabil!) -> 2 Tage gebraucht
 - ok
- TestAbdeckung? research, Emma, -> erfolg in grenzen gehalten, nicht so einfach möglich
 - Funktiniert nicht. Nicht verbreitet. Bis zu bestimmten Punkt gebracht, jedoch kein Report. Gemäss Mailingliste Android sind Probleme bekannt, jedoch keine Lösungen. Wird entsprechend dokumentiert.
- Sonstige Metriken vorhanden
 - keine Zirkuläre Abhängigkeiten
 - ok
- Videodrehbuch
 - Storyline soweit ganz gut -> jedoch zum Umfangreich
 - Inhalt wie auch in Abstract
 - Vorschlag: Erster Screen People am Openair (mit Interessanten Aspekten) -> Demo -> Abschluss Slide
- Demo Applikation
 - Ausgelassen, aus zeitlichen Gründen
- Kriterienliste Fragen
 - ACM, IEEE und Buch zitieren. (www.acm.org)
- Mgmt Summary
 - Ähnlich wie Anstract, einfach mehr "ausgehämmert". Gut 2-3 Seiten, 4 ist zuviel.
 - Problem ausgeweitet / Referenz zu Umfrage / ACM-Paper (Location Based Services, etc.)
 - Bilder von Hauptscreens
 - Content Management vorgesehen
 - Technisches (vielleicht 4 Punkte)

Laufend

- Endspurtplanung

Ausblick

- BA
 - Kinect
 - Muss noch durch SZE abgeklärt werden (hohe Priorität)
 - Treiber stehen zur Verfügung (nicht von MS, sondern von Hersteller direkt). Eignung nicht abgeklärt
 - Themen für Technorama:
 - Physik-Versuch (unterschiedliche Gravitationskräfte, etc.)
 - Orchester / Piano Conductor mit Bewegungserkennung und audativem Feedback

Weekly Meeting Wo12

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2), F. Egli (fx)

Zeit/Datum 13:10 / 8. Dezember 2010

Ort Rapperswil

Traktanden

- Protokoll -> OK
- Updatestrategie am laufen -> Für alle Fälle umgesetzt, kleine Bugs gefixt
 - Kleine Abhängigkeiten die berücksichtigt werden mussten ausgemerzt oder dabei
 - Layoutanpassungen -> Erwiesen länger gebraucht als gedacht?
 - Gemacht ->
 - Wie wird das Scrollen angezeigt? Eventuell Darstellen? Dunkler werden?
 - Glocken gross genug? -> Vergleich mit Settings -> den Checkbuttons
 - Android Styleguides Milimetergrösse angegeben
 - Verschiedene Touchpunkte direkt in der Nähe oder nicht.
 - Knappe grösse (Dicke-Finger)
 - ExpendableList? erklärt!
 - Und Anzeige der Wer spielt. Anhand Interesse.
- Interessiert mehr Info? -> Doch nicht?
- Was wenn keine Verbindung!
 - Keine spezielle Anzeige
 - Einfach expired
 - Leere Liste der Events oder die alten Events -> Als Notification
 - Sehen im Offlinemodus??
- Zeiten nachgetragen (letzte Woche)
- HAUPTFEATURES READY -> jetzt wird erweitert mit Push, voll dabei, Service macht einige Mühe.
 - Dokumentation
 - Wecker stellen / für Notifikation
 - Als Advanced Feature deklariert aber noch nicht umgesetzt ausser GUI.
 - 2 Versionen mit GUI
- WEITERENTWICKLUNG DOKUMENTIEREN!! WIE WO? Expendable Friends.
 - GPS funktioniert bei Android nicht, wenns schneit.
 - Bei schön 10-20 Meter an der HSR
- Spezielle Lessons Learned werden in Dokumentation erwähnt
 - Dokumentieren
- Im Video besonders Stolz anweisen. Technische Stolzheit / Update oder nicht?
 - Netz kein Netz / Aktualisiert

Offen

- Testcoverage wäre schön (wird noch abgeklärt)
- Sonstige Metriken
- Kriterien Stolze
 - Endspurtplanung

Ausblick

- BA
 - Anschlussarbeit, Kinect für Technorama wird abgeklärt M. Stolze

Weekly Meeting Wo11

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2), F. Egli (fx)

Zeit/Datum 13:10 / 1. Dezember 2010

Ort Rapperswil

Traktanden

- Protokoll -> Wird per e-Mail nachgeliefert
- Updatestrategie am laufen -> DecisionMaker? erklärt und als gute Lösung empfunden
- Testdaten ausgebaut -> Damit schlaue Gui-Anzeigen erscheinen.
- Layoutanpassungen -> Fast abgeschlossen
- Zeiten nachgetragen -> OK
- HAUPTFEATURES READY (BISSCHEN LAYOUT NOCH) -> Gut -> lieber gut getestet als Failen
- Testcoverage wäre schön wenn, ansonsten begründen

Ausblick

- BA
 - Anschlussarbeit, Kinect für Technorama wird abgeklärt M. Stolze
-

Weekly Meeting Wo10

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2), F. Egli (fx)

Zeit/Datum 13:20 / 24. November 2010

Ort Rapperswil

Traktanden

- Protokoll -> OK
- Testkonzept
 - Viel Zeit in anspruch genommen (dafür sauber) -> Besprochen, sehr nützlich
- Architektur (in groben Zügen)
 - Datenbank auf Serverseite vonnöten? -> Nein, FakeDatabase?, die mit Objekten arbeitet.
 - Handler-Konzept -> Erklärt und als TipTop? empfunden
- Code Review erledigt
 - > Gut
- Build Server -> Nicht nötig
- Dokumentation aktueller Stand
 - Ein Designentscheid noch nicht dokumentiert -> Aussage
 - Nächster Punkt SAD -> Aussage
- Zeiten geplant und nachgetragen
- Wikiseite, Lessons Learned auflisten

Ausblick

- BA
 - Besprochen & auf nächste Woche verschoben
-

Weekly Meeting Wo9

Ausgefallen wegen IDay's

Weekly Meeting Wo8

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2), F. Egli (fx)

Zeit/Datum 17:05 / 10. November 2010

Traktanden

- Protokoll -> Abgenommen
- Client/Server?-Architektur (Content Providers, Cache, Sql, Filesystem, UI)
 - Aktueller Stand informiert
 - Architektur implementiert
 - Auf Dev-Server installieren, so dass immer konstant ansprechen kann
- Aktuelles Client-UI
 - Nochmals demonstriert
 - Liste der Events durch GPS (Auswahl, der schwarzen Liste)
- Dokumentation aktueller Stand
 - Noch nicht weiter fortgeschritten
- Sourcecode aus <http://d.android.com> (Beispiele, alle referenzieren, wenn ja, wo?)
 - Wurde so abgenommen am Meeting
- UnitTests? müssen eingefügt werden (UnitTest?)
 - Grob und grobes JavaDoc?
 - Emulator Demo das nächste mal mit den UnitTest?
- Lizenz für Enterprise Architect -> OK funktioniert
- BA
 - ScrumTable? aussicht
 - Comparis -> GoogleTV / Made4TV
 - Microsoft Surface (Technorama)
 - BA für Technorama (HSR Logo)
 - Idee noch offen
 - Migros/Coop? Dating

Ausblick

- BA
-

Weekly Meeting Wo7

Allgemeines

Anwesend M. Stolze, M. Schnyder (m2)

Entschuldigt F. Egli (fx), Militärdienst

Zeit/Datum 13:10 / 3. November 2010

Traktanden

- Protokoll -> ok
- Anpassungen Domain Model (Abnahme)
 - Multiplizität bei Foto fehlt noch
 - Abgenommen!
- Client-Architektur (Content Providers, Cache, Sql, Filesystem, UI)
 - Wahrscheinlich nur noch ein Content Provider notwendig. Funktionen können zusammengefasst werden
- Aktuelles Client-UI
 - Tabs sind in Queransicht nicht optimal
 - Titlebar evtl. weglassen. Doch wo Zeit anzeigen?
 - Erklärung wie und wo der Auswahldialog für die Events aufgerufen werden kann
- Dokumentation aktueller Stand
 - Evaluationen genau so machen (Prototypen, verschiedene Architekturen)
 - Dokument senden, damit Detailgrad abgeklärt werden kann
- Sourcecode aus <http://d.android.com> (Beispiele, alle referenzieren, wenn ja, wo?)
 - Sofern aus <http://d.android.com> oder sonstigen offiziellen Quellen von Google: Einmal global Referenzieren (gilt v.a für Code-Snippets)
 - Sofern ganze Klasse übernommen werden, dies klar in der Klasse vermerken. (Egal ob von Google oder sonstwo)
- Lizenz für Enterprise Architect?
 - Abklärung mit Peter Bühler

Ausblick

- noch nichts
-

Weekly Meeting Wo6

Allgemeines

Anwesend M. Stolze, F. Egli (fx), M. Schnyder (m2)

Zeit/Datum 13:10 / 27. Oktober 2010

Traktanden

- Protokoll abgenommen
- Paperprototyp (Menubutton erwähnen 1&2)
- UseCases? gezeigt, farblich
- Domainmodel die Richtungen genauer beschrieben
 - Domainmodel nur grüne UseCases?
 - OK (Abgenommen) für grünen Teil
 - Bei optionalen muss Domainmodel angepasst werden *Unschönheit? Schon anfang optionalen einbauen in DomainModel?
- Kommunikation prototyp
 - problematik hinterher
 - laden schwieriger / mehr zeitaufwendig
 - verschiedene ansätze / SERVERLETS!!!!
 - formate nicht technologie/serverimplementierung "egal"
 - security keine anforderung (ev. dann mit Facebook) https zB
- Seebühne Prototyp anhand von 2 koordinaten und pixel gezeigt

- Militär -> Zeit wird zum Schluss sowieso nachgeholt
- Meeting kann auch nur einer kommen, wenn effizienter
- Gegenleser wie ist da der Stand?
 - BA dort ein Muss!
 - Studienleitung wer bekommt ein Gegenleser würfel!

Ausblick

- Terminliches
 - fx in Wo7 weg
 - Meeting Wo7
 - " Gegenleser?
-

Weekly Meeting Wo5

Allgemeines

Anwesend M. Stolze, F. Egli (fx), M. Schnyder (m2)

Zeit/Datum 10:10 / 21. Oktober 2010

Traktanden

- Abnahme d. Szenarios + Personas
 - P. P. "ist" Bratwurst
 - Abgenommen!
 - Vision usw. Abnahme
 - Abgenommen!
 - Domain-Modell (Abnahme)
 - Richtungen nicht klar der Beschriftungen
 - Besucher nicht an einem POI also auch 0..1
 - Adressart Visitor und Push beschreiben
 - UseCades?
 - Fotos nicht oben
 - Iterations UseCases?
 - Volle Vision
 - Agenda Minievent
 - Wenn nötig lieber ein SSD erstellen
 - Paperprototyp wie Dokumentieren, Vorschlag
 - Sehr gut
 - Handy Abo
 - Handy ohne Abo
 - Anderer Ort nicht Realitätsnahe
 - Sunrise Prepay
 - Handy Demo
 - Overlay (Karte)
 - Koordination Abklärungen andere Teams Geräte
 - Anfang der Sitzung vom letzten mal abnehmen lassen
-

Weekly Meeting Wo4

Allgemeines

Anwesend M. Stolze, F. Egli (fx), M. Schnyder (m2)

Zeit/Datum 10:10 / 13. Oktober 2010

Traktanden

- Stand der Arbeiten
 - Es ist viel gelaufen in der Letzten Zeit
- Behavioural Variables (nötig? / validierung?)
 - Die aktuellen reichen gut, müssen nicht noch verbessert werden. Validierung nicht notwendig, da auf Basis der Inretviews
- Desire HD zu spät (weitere Vorgehen)
 - Erledigt: HTC Desire wurde geliefert und wird nun mit der anderen Gruppe verwendet.
- Personas / Szenarios abnehmen
 - V0 ist abgenommen, Einzelne Anpassungen wie folgt:
 1. Klar machen, dass fertig ist. / Warum macht er das auf dem Handy. / Ein Muss das die Bedienung einfach ist
 2. Motivation ist unklar.
 3. Woher kennt er diese Funktion? / Evtl. Information wenn er das Gelände nähert.
 4. Motivation?
 5. Konkret sein. Wieso erhält PP die Nachricht? Allgemein: konkreter sein. Nicht "eine Band" sondern konkret
- SuperSets? abnehmen (Hauptfeatures)
 - Sind abgenommen. heissen aber nun "Szenario Auswahl"
- Technologieabklärung (Stand informieren)
 - Wie genau Dokumentieren? Mit Beispiele zu jedem Punkt? Es so genau richtig gemacht. Abgenommen. Kommt vorne hin. Details richtig.
- Paper Prototypen besprechen, Menü Funktion Android -> Einschätzung
 - Input: Direkt mit PPT testen ob Menu funktioniert oder nicht
 - Evtl. Intro oder Tipps in Applikation einbauen (m2)
- Use-Cases aktuelle Stufe
 - Noch nicht besprochen. Nicht viel Zeit verwenden.
- Vision, Ziele-Dokument (richtige Richtung?)
 - m2 richtig verstanden, weiter in diese Richtung
- Android Buch
 - <http://apress.com/book/view/1430232765> (Pro Android Web Apps: Developing HTML5, JavaScript?, CSS, and Chrome OS Web Apps)
- Sonstiges
 - NFA gefunden: Server muss Datenverwaltung unterstützen. So dass später mal ein GUI drübergeklatscht werden kann.
 - Zusätzliche Stakeholder nicht vergessen (Veranstalter, Standbetreiber, Infopersonal)

Weekly Meeting Wo3

Allgemeines

Anwesend M. Stolze, F. Egli (fx), M. Schnyder (m2)

Zeit/Datum 13:10 / 6. Oktober 2010

Traktanden

- Alle Interviews fertig durchgeführt.
 - Eindrücke erzählt
 - Standbesitzer auf arbeit bezogen
 - Durchsagen? Grosse verlosung?
 - Cupon zum wiederfinden, Cupon einlösen für günstiger
 - Cupon Büechli
 - Sunrise
 - Interviews ausgewertet
 - Mehr Szenarios (zwei Personas sollten reichen)
 - Weiteres vorgehen
 - Szenarios
 - Personas
 - Technische Abklärungen
 - Desire normal kaufen
 - Facebook-Locations freigeschaltet
 - Facebook integrations (Züspa)
 - Live Demo von m2
-

Weekly Meeting Wo2

Anwesend M. Stolze, F. Egli (fx), M. Schnyder (m2)

Zeit/Datum 14:00 / 30. September 2010

Traktanden

- Erledigte Arbeiten / Tasks
 - siehe [ToDoList](#)
- Aufwendungen Stunden
 - Information betreffend mite (und wieso nicht direkt in trac abrechnen: Zu Starr, Plugin ungetestet/nicht funktionsfähig)
- Risikoliste Quercheck
 - Einige Zusätzliche Risiken identifiziert und besprochen
- Projektplan Quercheck
 - Ok, keine Anmerkungen
- Taskliste Anwendung / Zugang Wiki ok?: Gezeigt und ok, auch Login funktioniert.
- Interview Eindrücke
 - Interessante Erfahrungen
 - Umfangreicher Fragenkatalog
 - Wird für Züspa noch angepasst
- Zeitplan / Arbeitspakete überprüfen / Eindruck
 - Zeitplanung jeweils vor nächster Iteration für 1 Iteration ok
 - Fully dressed [UseCases?](#) weglassen, lieber gute Szenarios machen
- Stand Androidhardware
 - Motorola Droid und Nexus One möglich auszuleihen. Wichtig für Tests auf älteren HW-Plattformen (Nexus One kann Android 2.2.1)
 - Evtl. HTC DesireHD angeschafft, Android Tablet wäre auch gute Testmöglichkeit

Projekt Ausblick

- Züspa
 - Fragenkatalog muss vereinfacht werden

Spezifische Fragen an sze

- Milestone 9 von seinem Bewertungsbogen: Super-Set User Stories????
 - Liste von allen Möglichen Features welche dann eingeschränkt wird um zu implementieren (analog unserer Must, Can and Stage2 Liste)

Todo

- Fragebogen für Sohnbefragung SZE per Mail
 - Mitte Projekt: Visionsdokument sollte stehen
-

Weekly Meeting Wo1

Allgemeines

Anwesend M. Stolze, F. Egli, M. Schnyder

Zeit/Datum 13:10 / 22. September 2010

Geklärte Punkte

- Umstellung des Meetings vom 13.10.2010 verschoben 14.10.2010 10:10
- Umstellung des Meetings vom 29.09.2010 verschoben 30.09.2010 14:00
- Link auf unser Wiki möglich von HSR Wiki
- Bildschirme grössere möglich? Ersetzt wenn SA fertig. Nicht Begehrlichkeiten schaffen. Privat abklären.
- Virtueller Server wurde bestellt.
- Android Hardware wurde angefragt, vier Android-Geräte vorhanden, zwei könnten angeboten werden HTC G1 und Motorola Milestone angeboten werden.
 - Desire HD austatten wäre von Vorteil wird abgeklärt ob möglich (Tageweise ausleihen).
- Ein komplettes Dokument mit Seitenzahl durchnummeriert inkl. höhere Tiefe? JA
- Events (zwei sollten reichen)
 - Freestyle
 - Züspa/Olma?
- Interview gelesen, wann Zeit? Wird per Email versandt, wenn grosse Änderungen Skypen.
- Youtubevideo ab Scrum Beispiel. Geführte Demo!
- Kapitel Implementation nicht sehr viel Sinn. (Einzeihler, siehe JavaDoc?)
- Dry in Doku, lieber Referenzen (Mitte Semester, nochmals Inhaltsverzeichnis durcharbeiten)
- Dokumentation am Schluss muss er es laufen lassen kann (Installationsanleitung usw.)
- Aufgabenstellung
 - Typofehler
 - Nicht sehr spezifisch,
 - Erweiterung der Aufgabenstellung
 - kein Partner sondern anhand Interviews mit den wertvollen Funktionen
 - Technologie Evaluation HTML5/Android Java untersuchen
- Usabilitytest wird verwendet um Ausblick zu formulieren
 - Für diese Benutzergruppe ist es benutzbar
 - Diese Probleme sind vorhanden und mit der und der Änderungen wäre es möglich
 - Situationsabhängig (MAINFEATURES müssen gut sein)
- Bericht & 2 CD das eine Dokument!!
- Highlevel Projektplan ok!
- Gewählte SE-Vorgehen wird RUP sein
- Stundenplanung muss genau sein, vorhanden sein
- LaTeX Verwendung
- Probleme mit Freestyle Rücksprache

Offene Todos

- Wikilink für Stolze
- Stolze klärt Desire ab
- Schnyder klärt Tickets ab
- Interview Fragen vorbereiten

- Zeiten in Projektplanung jeweils für nächste Sitzung vorbereiten

Anhang N

Codemetriken

N.1 Codemeteriken Client

Project Summary: CLIENT

Created: 21. Dezember 2010 13:22 by [Structure101](#)
Project file: C:\Users\fegli\HSR-EventAssistant\docs\14_Umsetzungskonzept\structure101 metrics.java.hsp
Classpath: C:\Users\fegli\HSR-EventAssistant\src\EventAssistant\bin
Granularity: "Overview" - no inner classes, methods or fields (fat classes and methods not detected)
Externals: "Hide" - referenced items are not included
Excludes: None
Transformations: None
Hierarchy: "Package"
Notes and tips: On

- | | |
|-------------|---|
| Tip: | <ul style="list-style-type: none">This summary has been generated with "Show notes and tips" enabled - these are shown with a yellow background, like this one. You can switch them off via the drop-down menu in the Overview perspective (or with the -notips switch from the command line).The Collaboration perspective is great for tracking down your dependency on third party libraries. Set "Show externals" in the project properties and then select the library item in the right-hand tree - anything that doesn't use it will be grayed out in the left-hand tree. |
|-------------|---|

Size

Jars (and/or classpath directories):	1	NI (Number of bytecode Instructions):	8,673
Packages (that contain classes):	11	LOC (Non Comment Non Blank Lines Of Code):	~3,729
Classes (outer):	28		
Classes (all):	28		
Classes (external):	0		

- | | |
|--------------|---|
| Note: | <ul style="list-style-type: none">Though less commonly used than LOC, NI is an excellent, style-independent, relative measure of code size.The value for LOC is estimated from the NI. |
|--------------|---|

Flat Tangles

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
Leaf package	11	0	0	0	0%
Jar	1	n/a	n/a	n/a	n/a
Outer class	28	0	0	0	0%

Note:	<ul style="list-style-type: none">• "Flat" tangles are detected and measured from the flat dependency graphs of all items at a specific level. These are independent of the "Design" levels which are created by the higher-level packaging and are measured and reported in the "Excessive Structural Complexity (XS)" section below.• Tangles at the Jar or Package-level are highly disruptive.• The bigger the tangle, the more problems it is likely to cause.• Larger tangles at the class level are likely to cross package boundaries and so may also indicate a problem.• The degree is calculated so as to give a higher value for larger tangles - that is, one large tangle will give a higher degree than the same number of items involved in several smaller tangles.
Tip:	<ul style="list-style-type: none">• The "Leaf package" and "Outer class" tangles can be directly viewed and analyzed in the Slice perspective.• To view and analyze jar-level tangles, select the Jar hierarchy in the Composition perspective.

Architecture

#Diagrams: 1
#Violations: 0
Violation frequency: 0%

Note:	<ul style="list-style-type: none">• The architecture is defined in the form of a number of architecture diagrams that define layering and composition, and map to the physical code.• Architecture diagrams are not automatically created - that must be done by users (e.g. you).• The "#Violations" value is the number of class-to-class dependencies if the project granularity is "Overview" (see the top of this report). If the granularity is "Detail" then it is the number of method/field/class dependencies.• The "Violation frequency" is the number of violations as a percentage of all the dependencies in the model.
Tip:	<ul style="list-style-type: none">• You can initialize new diagrams wherever you see the "new architectural diagram" button - in the Composition perspective, the Slice perspective and the Architecture perspective.

- Create architecture diagrams from dependency diagrams that most closely reflect the intended architecture.
- You can also create architecture diagrams from scratch in the Architecture perspective.
- Once you have created architectural diagrams, you can edit them in the Architecture perspective.
- If you publish Architecture diagrams to a repository, the Structure101 IDE plug-ins will warn developers if they make code changes that are inconsistent with them, plus they will be browsable via the Structure101 web application.

Excessive Structural Complexity (XS)

Cumulative XS: 0

Average XS: 0%

Note:	<ul style="list-style-type: none"> • The XS Configuration is a set of thresholds for "Fat" and "Tangles" at different levels of composition. • Fat indicates "too much stuff" in one place and is measurable at every level of composition - Design, Package, Class and Method. • Cyclomatic Complexity is used as a measure of Fat at the method level - the number of dependencies in the child dependency graph is used at the other levels. • For the purposes of XS, tangles are only measured at the design level, i.e. cyclic dependencies between packages. • Both Fat and Tangles are measured as the percentage by which the threshold is exceeded. This percentage is multiplied by the size of the item so that the Average XS can be read as a percentage of the code-base. • Read more about the concepts behind XS.
Tip:	<ul style="list-style-type: none"> • Use the Measurement perspective to obtain detailed information on the XS in your project. • Change the metrics and thresholds used to calculate XS by putting the XS Configuration into "Edit mode" from the drop down menu, bottom-left, in the Measurement perspective. Save the changes as a new named configuration. • If you have a very large project and find the metrics calculations too time-consuming, you can exclude this section of the report via the Options menu.

XS breakout by metric (and scope)

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Tangled (design)	0	0 of 5	0%	0%
Fat (design)	120	0 of 5	0%	0%
Fat (leaf package)	120	0 of 11	0%	0%
Fat (class)	120	0 of 28	0%	0%
Fat (method)	15	0 of 0	0%	0%

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Total				100%

Note:	<ul style="list-style-type: none"> The Metric (and scope) column contains all the metrics that are enabled in the selected XS Configuration (named in the previous section). The "Threshold" is taken from the selected XS Configuration The "#Offenders" indicates the number of items that exceed the threshold. The "Offenders (%)" indicates the number of items that exceed the threshold as a percentage of the total number of items at that level. The "XS contribution" indicates the degree to which design tangles and fat at each level contribute to the total amount of XS in the project.
Tip:	<ul style="list-style-type: none"> Use the Measurement perspective to obtain detailed information on the XS in your project.

Tangled (design)

No items exceed the threshold for Tangled at the design level.

Note:	<ul style="list-style-type: none"> The "design" level items are high-level packages (i.e. packages that contain sub-packages). Tangles are only measured at this level. The value is calculated as the number of dependencies that go bottom-up as a percentage of all the dependencies on the child graph for the item.
Tip:	<ul style="list-style-type: none"> To view and analyze these tangles, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu. The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Tangled (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (design)

No items exceed the threshold for Fat at the design level.

Note:	<ul style="list-style-type: none">• The "design" level items are high-level packages (i.e. packages that contain sub-packages).• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the sub-packages of the measured package.
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (leaf package)

No items exceed the threshold for Fat at the leaf package level.

Note:	<ul style="list-style-type: none">• The "leaf package" level items are packages that contain classes.• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the classes in the measured package
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (leaf package) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (class)

No items exceed the threshold for Fat at the class level.

--	--

Note:	<ul style="list-style-type: none">• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the methods and fields in the class.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (class) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (method)

No items exceed the threshold for Fat at the method level.

Note:	<ul style="list-style-type: none">• The value is calculated as the Cyclomatic Complexity (McCabe's metric) of the method.• The value is calculated from the bytecode and so may differ from the value that would be calculated from the corresponding source code in some (rare) cases.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the class that contains the method in any perspective.• Click on the "Show source" button on the main toolbar to view the source.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (method) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

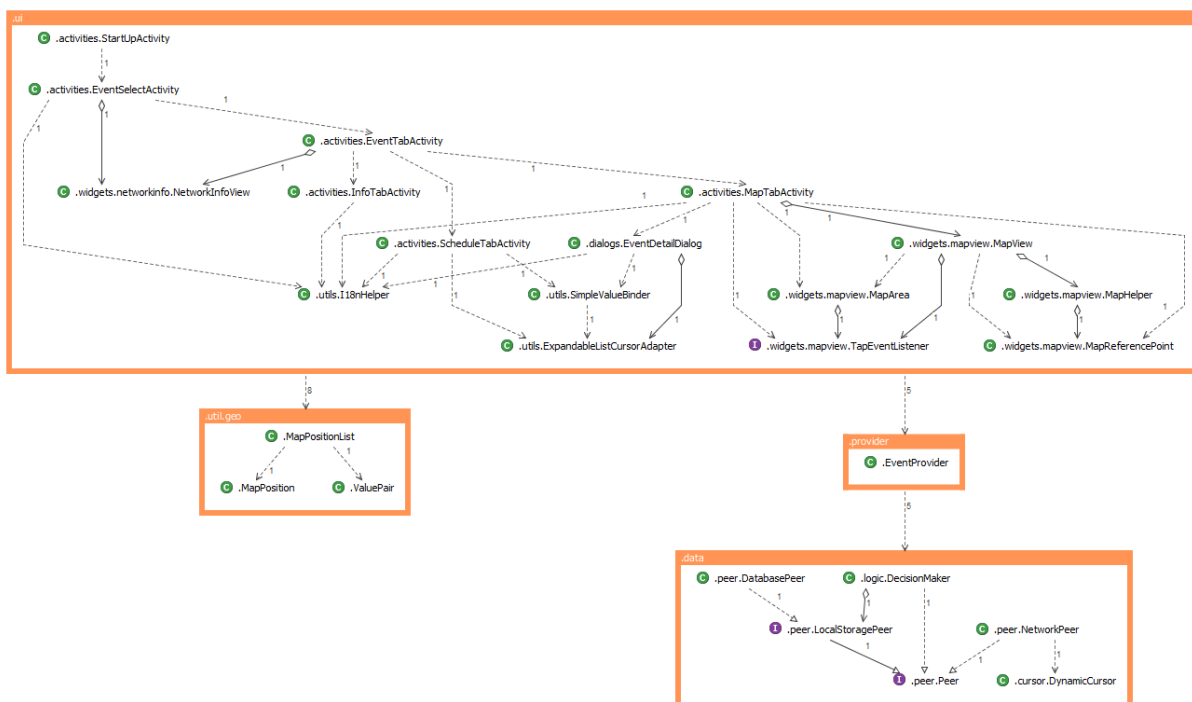
Items with highest XS

None

Note:	<ul style="list-style-type: none">• XS is related to the value and the size of the item so that bigger items (usually higher in the hierarchy) score higher. Since excessive structural complexity on bigger items (e.g. packages) will tend to disrupt development activities more than on smaller items (e.g.
--------------	---

	<div>methods), you should consider fixing these first.</div> <ul style="list-style-type: none">Values in red exceed threshold.
Tip:	<ul style="list-style-type: none">The complete list is available in the Measurement perspective - make sure you are at the root of the "Drill-down" in the top-left, and do not select any items in the pie.You can integrate the generation of this summary into your build process. See help for details.

Classtructure



N.2 Codemeteriken Server

Project Summary: structure101 server

Created: 21. Dezember 2010 13:23 by [Structure101](#)
Project file: C:\Users\fegli\HSR-EventAssistant\docs\14_Umsetzungskonzept\structure101 server.java.hsp
Classpath: C:\Users\fegli\HSR-EventAssistant\src\EventAssistant-Server\build\classes
Granularity: "Overview" - no inner classes, methods or fields (fat classes and methods not detected)
Externals: "Hide" - referenced items are not included
Excludes: None
Transformations: None
Hierarchy: "Package"
Notes and tips: On

Tip:	<ul style="list-style-type: none">• This summary has been generated with "Show notes and tips" enabled - these are shown with a yellow background, like this one. You can switch them off via the drop-down menu in the Overview perspective (or with the -notips switch from the command line).• The Collaboration perspective is great for tracking down your dependency on third party libraries. Set "Show externals" in the project properties and then select the library item in the right-hand tree - anything that doesn't use it will be grayed out in the left-hand tree.
-------------	---

Size

Jars (and/or classpath directories):	1	NI (Number of bytecode Instructions):	3,754
Packages (that contain classes):	6	LOC (Non Comment Non Blank Lines Of Code):	~1,614
Classes (outer):	27		
Classes (all):	27		
Classes (external):	0		

Note:	<ul style="list-style-type: none">• Though less commonly used than LOC, NI is an excellent, style-independent, relative measure of code size.• The value for LOC is estimated from the NI.
--------------	---

Flat Tangles

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
Leaf package	6	0	0	0	0%
Jar	1	n/a	n/a	n/a	n/a
Outer class	27	0	0	0	0%

Note:	<ul style="list-style-type: none">• "Flat" tangles are detected and measured from the flat dependency graphs of all items at a specific level. These are independent of the "Design" levels which are created by the higher-level packaging and are measured and reported in the "Excessive Structural Complexity (XS)" section below.• Tangles at the Jar or Package-level are highly disruptive.• The bigger the tangle, the more problems it is likely to cause.• Larger tangles at the class level are likely to cross package boundaries and so may also indicate a problem.• The degree is calculated so as to give a higher value for larger tangles - that is, one large tangle will give a higher degree than the same number of items involved in several smaller tangles.
Tip:	<ul style="list-style-type: none">• The "Leaf package" and "Outer class" tangles can be directly viewed and analyzed in the Slice perspective.• To view and analyze jar-level tangles, select the Jar hierarchy in the Composition perspective.

Architecture

#Diagrams: 1
#Violations: 0
Violation frequency: 0%

Note:	<ul style="list-style-type: none">• The architecture is defined in the form of a number of architecture diagrams that define layering and composition, and map to the physical code.• Architecture diagrams are not automatically created - that must be done by users (e.g. you).• The "#Violations" value is the number of class-to-class dependencies if the project granularity is "Overview" (see the top of this report). If the granularity is "Detail" then it is the number of method/field/class dependencies.• The "Violation frequency" is the number of violations as a percentage of all the dependencies in the model.
Tip:	<ul style="list-style-type: none">• You can initialize new diagrams wherever you see the "new architectural diagram" button - in the Composition perspective, the Slice perspective and the Architecture perspective.

- Create architecture diagrams from dependency diagrams that most closely reflect the intended architecture.
- You can also create architecture diagrams from scratch in the Architecture perspective.
- Once you have created architectural diagrams, you can edit them in the Architecture perspective.
- If you publish Architecture diagrams to a repository, the Structure101 IDE plug-ins will warn developers if they make code changes that are inconsistent with them, plus they will be browsable via the Structure101 web application.

Excessive Structural Complexity (XS)

Cumulative XS: 0

Average XS: 0%

Note:	<ul style="list-style-type: none"> • The XS Configuration is a set of thresholds for "Fat" and "Tangles" at different levels of composition. • Fat indicates "too much stuff" in one place and is measurable at every level of composition - Design, Package, Class and Method. • Cyclomatic Complexity is used as a measure of Fat at the method level - the number of dependencies in the child dependency graph is used at the other levels. • For the purposes of XS, tangles are only measured at the design level, i.e. cyclic dependencies between packages. • Both Fat and Tangles are measured as the percentage by which the threshold is exceeded. This percentage is multiplied by the size of the item so that the Average XS can be read as a percentage of the code-base. • Read more about the concepts behind XS.
Tip:	<ul style="list-style-type: none"> • Use the Measurement perspective to obtain detailed information on the XS in your project. • Change the metrics and thresholds used to calculate XS by putting the XS Configuration into "Edit mode" from the drop down menu, bottom-left, in the Measurement perspective. Save the changes as a new named configuration. • If you have a very large project and find the metrics calculations too time-consuming, you can exclude this section of the report via the Options menu.

XS breakout by metric (and scope)

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Tangled (design)	0	0 of 3	0%	0%
Fat (design)	120	0 of 3	0%	0%
Fat (leaf package)	120	0 of 6	0%	0%
Fat (class)	120	0 of 27	0%	0%
Fat (method)	15	0 of 0	0%	0%

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Total				100%

Note:	<ul style="list-style-type: none"> The Metric (and scope) column contains all the metrics that are enabled in the selected XS Configuration (named in the previous section). The "Threshold" is taken from the selected XS Configuration The "#Offenders" indicates the number of items that exceed the threshold. The "Offenders (%)" indicates the number of items that exceed the threshold as a percentage of the total number of items at that level. The "XS contribution" indicates the degree to which design tangles and fat at each level contribute to the total amount of XS in the project.
Tip:	<ul style="list-style-type: none"> Use the Measurement perspective to obtain detailed information on the XS in your project.

Tangled (design)

No items exceed the threshold for Tangled at the design level.

Note:	<ul style="list-style-type: none"> The "design" level items are high-level packages (i.e. packages that contain sub-packages). Tangles are only measured at this level. The value is calculated as the number of dependencies that go bottom-up as a percentage of all the dependencies on the child graph for the item.
Tip:	<ul style="list-style-type: none"> To view and analyze these tangles, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu. The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Tangled (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (design)

No items exceed the threshold for Fat at the design level.

Note:	<ul style="list-style-type: none">• The "design" level items are high-level packages (i.e. packages that contain sub-packages).• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the sub-packages of the measured package.
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (leaf package)

No items exceed the threshold for Fat at the leaf package level.

Note:	<ul style="list-style-type: none">• The "leaf package" level items are packages that contain classes.• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the classes in the measured package
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (leaf package) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (class)

No items exceed the threshold for Fat at the class level.

--	--

Note:	<ul style="list-style-type: none">• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the methods and fields in the class.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (class) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (method)

No items exceed the threshold for Fat at the method level.

Note:	<ul style="list-style-type: none">• The value is calculated as the Cyclomatic Complexity (McCabe's metric) of the method.• The value is calculated from the bytecode and so may differ from the value that would be calculated from the corresponding source code in some (rare) cases.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the class that contains the method in any perspective.• Click on the "Show source" button on the main toolbar to view the source.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (method) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

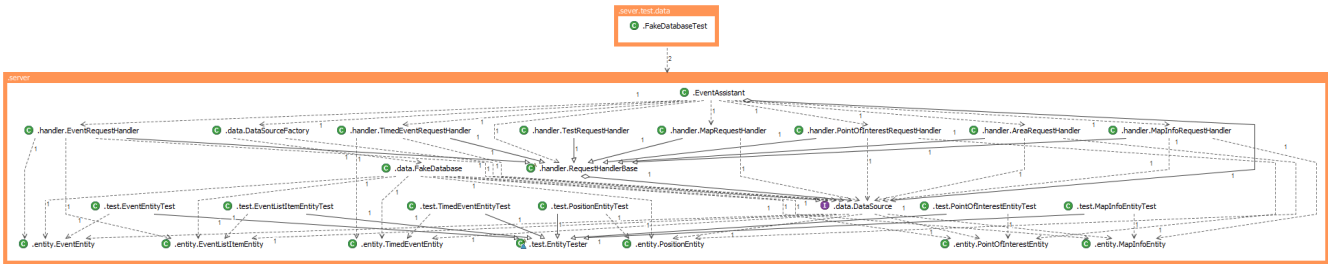
Items with highest XS

None

Note:	<ul style="list-style-type: none">• XS is related to the value and the size of the item so that bigger items (usually higher in the hierarchy) score higher. Since excessive structural complexity on bigger items (e.g. packages) will tend to disrupt development activities more than on smaller items (e.g.
--------------	---

	<p>methods), you should consider fixing these first.</p> <ul style="list-style-type: none">• Values in red exceed threshold.
Tip:	<ul style="list-style-type: none">• The complete list is available in the Measurement perspective - make sure you are at the root of the "Drill-down" in the top-left, and do not select any items in the pie.• You can integrate the generation of this summary into your build process. See help for details.

Classtructure



N.3 Codemeteriken Common

Project Summary: Common

Created: 21. Dezember 2010 13:27 by [Structure101](#)
Classpath: C:\Users\fegli\HSR-EventAssistant\src\EventAssistant-Common\bin
Granularity: "Overview" - no inner classes, methods or fields (fat classes and methods not detected)
Externals: "Hide" - referenced items are not included
Excludes: None
Transformations: None
Hierarchy: "Package"
Notes and tips: On

- Tip:**
- This summary has been generated with "Show notes and tips" enabled - these are shown with a yellow background, like this one. You can switch them off via the drop-down menu in the Overview perspective (or with the -notips switch from the command line).
 - The Collaboration perspective is great for tracking down your dependency on third party libraries. Set "Show externals" in the project properties and then select the library item in the right-hand tree - anything that doesn't use it will be grayed out in the left-hand tree.

Size

Jars (and/or classpath directories):	1	NI (Number of bytecode Instructions):	379
Packages (that contain classes):	1	LOC (Non Comment Non Blank Lines Of Code):	~163
Classes (outer):	9		
Classes (all):	9		
Classes (external):	0		

- Note:**
- Though less commonly used than LOC, NI is an excellent, style-independent, relative measure of code size.
 - The value for LOC is estimated from the NI.

Flat Tangles

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
-------	--------	----------	----------------	---------	--------

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
Leaf package	1	n/a	n/a	n/a	n/a
Jar	1	n/a	n/a	n/a	n/a
Outer class	9	0	0	0	0%

Note:	<ul style="list-style-type: none"> "Flat" tangles are detected and measured from the flat dependency graphs of all items at a specific level. These are independent of the "Design" levels which are created by the higher-level packaging and are measured and reported in the "Excessive Structural Complexity (XS)" section below. Tangles at the Jar or Package-level are highly disruptive. The bigger the tangle, the more problems it is likely to cause. Larger tangles at the class level are likely to cross package boundaries and so may also indicate a problem. The degree is calculated so as to give a higher value for larger tangles - that is, one large tangle will give a higher degree than the same number of items involved in several smaller tangles.
Tip:	<ul style="list-style-type: none"> The "Leaf package" and "Outer class" tangles can be directly viewed and analyzed in the Slice perspective. To view and analyze jar-level tangles, select the Jar hierarchy in the Composition perspective.

Architecture

The architecture for this project is not defined

Note:	<ul style="list-style-type: none"> The architecture is defined in the form of a number of architecture diagrams that define layering and composition, and map to the physical code. Architecture diagrams are not automatically created - that must be done by users (e.g. you). The "#Violations" value is the number of class-to-class dependencies if the project granularity is "Overview" (see the top of this report). If the granularity is "Detail" then it is the number of method/field/class dependencies. The "Violation frequency" is the number of violations as a percentage of all the dependencies in the model.
Tip:	<ul style="list-style-type: none"> You can initialize new diagrams wherever you see the "new architectural diagram" button - in the Composition perspective, the Slice perspective and the Architecture perspective. Create architecture diagrams from dependency diagrams that most closely reflect the intended architecture.

- You can also create architecture diagrams from scratch in the Architecture perspective.
- Once you have created architectural diagrams, you can edit them in the Architecture perspective.
- If you publish Architecture diagrams to a repository, the Structure101 IDE plug-ins will warn developers if they make code changes that are inconsistent with them, plus they will be browsable via the Structure101 web application.

Excessive Structural Complexity (XS)

Cumulative XS: 0

Average XS: 0%

Note:

- The XS Configuration is a set of thresholds for "Fat" and "Tangles" at different levels of composition.
- Fat indicates "too much stuff" in one place and is measurable at every level of composition - Design, Package, Class and Method.
- Cyclomatic Complexity is used as a measure of Fat at the method level - the number of dependencies in the child dependency graph is used at the other levels.
- For the purposes of XS, tangles are only measured at the design level, i.e. cyclic dependencies between packages.
- Both Fat and Tangles are measured as the percentage by which the threshold is exceeded. This percentage is multiplied by the size of the item so that the Average XS can be read as a percentage of the code-base.
- Read [more](#) about the concepts behind XS.

Tip:

- Use the Measurement perspective to obtain detailed information on the XS in your project.
- Change the metrics and thresholds used to calculate XS by putting the XS Configuration into "Edit mode" from the drop down menu, bottom-left, in the Measurement perspective. Save the changes as a new named configuration.
- If you have a very large project and find the metrics calculations too time-consuming, you can exclude this section of the report via the Options menu.

XS breakout by metric (and scope)

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Tangled (design)	0	0 of 1	0%	0%
Fat (design)	120	0 of 1	0%	0%
Fat (leaf package)	120	0 of 1	0%	0%
Fat (class)	120	0 of 9	0%	0%
Fat (method)	15	0 of 0	0%	0%

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Total				100%

Note:	<ul style="list-style-type: none"> The Metric (and scope) column contains all the metrics that are enabled in the selected XS Configuration (named in the previous section). The "Threshold" is taken from the selected XS Configuration The "#Offenders" indicates the number of items that exceed the threshold. The "Offenders (%)" indicates the number of items that exceed the threshold as a percentage of the total number of items at that level. The "XS contribution" indicates the degree to which design tangles and fat at each level contribute to the total amount of XS in the project.
Tip:	<ul style="list-style-type: none"> Use the Measurement perspective to obtain detailed information on the XS in your project.

Tangled (design)

No items exceed the threshold for Tangled at the design level.

Note:	<ul style="list-style-type: none"> The "design" level items are high-level packages (i.e. packages that contain sub-packages). Tangles are only measured at this level. The value is calculated as the number of dependencies that go bottom-up as a percentage of all the dependencies on the child graph for the item.
Tip:	<ul style="list-style-type: none"> To view and analyze these tangles, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu. The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Tangled (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (design)

No items exceed the threshold for Fat at the design level.

Note:	<ul style="list-style-type: none">• The "design" level items are high-level packages (i.e. packages that contain sub-packages).• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the sub-packages of the measured package.
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (leaf package)

No items exceed the threshold for Fat at the leaf package level.

Note:	<ul style="list-style-type: none">• The "leaf package" level items are packages that contain classes.• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the classes in the measured package
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (leaf package) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (class)

No items exceed the threshold for Fat at the class level.

--	--

Note:	<ul style="list-style-type: none">• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the methods and fields in the class.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (class) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (method)

No items exceed the threshold for Fat at the method level.

Note:	<ul style="list-style-type: none">• The value is calculated as the Cyclomatic Complexity (McCabe's metric) of the method.• The value is calculated from the bytecode and so may differ from the value that would be calculated from the corresponding source code in some (rare) cases.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the class that contains the method in any perspective.• Click on the "Show source" button on the main toolbar to view the source.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (method) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

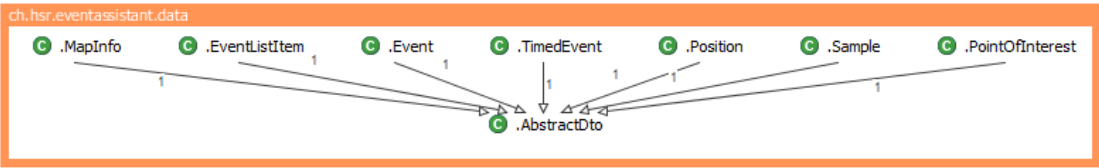
Items with highest XS

None

Note:	<ul style="list-style-type: none">• XS is related to the value and the size of the item so that bigger items (usually higher in the hierarchy) score higher. Since excessive structural complexity on bigger items (e.g. packages) will tend to disrupt development activities more than on smaller items (e.g.
--------------	---

	<p>methods), you should consider fixing these first.</p> <ul style="list-style-type: none">• Values in red exceed threshold.
Tip:	<ul style="list-style-type: none">• The complete list is available in the Measurement perspective - make sure you are at the root of the "Drill-down" in the top-left, and do not select any items in the pie.• You can integrate the generation of this summary into your build process. See help for details.

Classstructure



N.4 Codemeteriken Test

Project Summary: Test

Created: 21. Dezember 2010 13:31 by [Structure101](#)
Classpath: C:\Users\fegli\HSR-EventAssistant\src\EventAssistant-Test\bin
Granularity: "Overview" - no inner classes, methods or fields (fat classes and methods not detected)
Externals: "Hide" - referenced items are not included
Excludes: None
Transformations: None
Hierarchy: "Package"
Notes and tips: On

- Tip:**
- This summary has been generated with "Show notes and tips" enabled - these are shown with a yellow background, like this one. You can switch them off via the drop-down menu in the Overview perspective (or with the -notips switch from the command line).
 - The Collaboration perspective is great for tracking down your dependency on third party libraries. Set "Show externals" in the project properties and then select the library item in the right-hand tree - anything that doesn't use it will be grayed out in the left-hand tree.

Size

Jars (and/or classpath directories):	1	NI (Number of bytecode Instructions):	1,433
Packages (that contain classes):	1	LOC (Non Comment Non Blank Lines Of Code):	~616
Classes (outer):	8		
Classes (all):	8		
Classes (external):	0		

- Note:**
- Though less commonly used than LOC, NI is an excellent, style-independent, relative measure of code size.
 - The value for LOC is estimated from the NI.

Flat Tangles

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
-------	--------	----------	----------------	---------	--------

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
Leaf package	1	n/a	n/a	n/a	n/a
Jar	1	n/a	n/a	n/a	n/a
Outer class	8	0	0	0	0%

Note:	<ul style="list-style-type: none"> "Flat" tangles are detected and measured from the flat dependency graphs of all items at a specific level. These are independent of the "Design" levels which are created by the higher-level packaging and are measured and reported in the "Excessive Structural Complexity (XS)" section below. Tangles at the Jar or Package-level are highly disruptive. The bigger the tangle, the more problems it is likely to cause. Larger tangles at the class level are likely to cross package boundaries and so may also indicate a problem. The degree is calculated so as to give a higher value for larger tangles - that is, one large tangle will give a higher degree than the same number of items involved in several smaller tangles.
Tip:	<ul style="list-style-type: none"> The "Leaf package" and "Outer class" tangles can be directly viewed and analyzed in the Slice perspective. To view and analyze jar-level tangles, select the Jar hierarchy in the Composition perspective.

Architecture

The architecture for this project is not defined

Note:	<ul style="list-style-type: none"> The architecture is defined in the form of a number of architecture diagrams that define layering and composition, and map to the physical code. Architecture diagrams are not automatically created - that must be done by users (e.g. you). The "#Violations" value is the number of class-to-class dependencies if the project granularity is "Overview" (see the top of this report). If the granularity is "Detail" then it is the number of method/field/class dependencies. The "Violation frequency" is the number of violations as a percentage of all the dependencies in the model.
Tip:	<ul style="list-style-type: none"> You can initialize new diagrams wherever you see the "new architectural diagram" button - in the Composition perspective, the Slice perspective and the Architecture perspective. Create architecture diagrams from dependency diagrams that most closely reflect the intended architecture.

- You can also create architecture diagrams from scratch in the Architecture perspective.
- Once you have created architectural diagrams, you can edit them in the Architecture perspective.
- If you publish Architecture diagrams to a repository, the Structure101 IDE plug-ins will warn developers if they make code changes that are inconsistent with them, plus they will be browsable via the Structure101 web application.

Excessive Structural Complexity (XS)

Cumulative XS: 0

Average XS: 0%

Note:	<ul style="list-style-type: none"> • The XS Configuration is a set of thresholds for "Fat" and "Tangles" at different levels of composition. • Fat indicates "too much stuff" in one place and is measurable at every level of composition - Design, Package, Class and Method. • Cyclomatic Complexity is used as a measure of Fat at the method level - the number of dependencies in the child dependency graph is used at the other levels. • For the purposes of XS, tangles are only measured at the design level, i.e. cyclic dependencies between packages. • Both Fat and Tangles are measured as the percentage by which the threshold is exceeded. This percentage is multiplied by the size of the item so that the Average XS can be read as a percentage of the code-base. • Read more about the concepts behind XS.
Tip:	<ul style="list-style-type: none"> • Use the Measurement perspective to obtain detailed information on the XS in your project. • Change the metrics and thresholds used to calculate XS by putting the XS Configuration into "Edit mode" from the drop down menu, bottom-left, in the Measurement perspective. Save the changes as a new named configuration. • If you have a a very large project and find the metrics calculations too time-consuming, you can exclude this section of the report via the Options menu.

XS breakout by metric (and scope)

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Tangled (design)	0	0 of 1	0%	0%
Fat (design)	120	0 of 1	0%	0%
Fat (leaf package)	120	0 of 1	0%	0%
Fat (class)	120	0 of 8	0%	0%
Fat (method)	15	0 of 0	0%	0%

Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Total				100%

Note:	<ul style="list-style-type: none"> The Metric (and scope) column contains all the metrics that are enabled in the selected XS Configuration (named in the previous section). The "Threshold" is taken from the selected XS Configuration The "#Offenders" indicates the number of items that exceed the threshold. The "Offenders (%)" indicates the number of items that exceed the threshold as a percentage of the total number of items at that level. The "XS contribution" indicates the degree to which design tangles and fat at each level contribute to the total amount of XS in the project.
Tip:	<ul style="list-style-type: none"> Use the Measurement perspective to obtain detailed information on the XS in your project.

Tangled (design)

No items exceed the threshold for Tangled at the design level.

Note:	<ul style="list-style-type: none"> The "design" level items are high-level packages (i.e. packages that contain sub-packages). Tangles are only measured at this level. The value is calculated as the number of dependencies that go bottom-up as a percentage of all the dependencies on the child graph for the item.
Tip:	<ul style="list-style-type: none"> To view and analyze these tangles, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu. The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Tangled (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (design)

No items exceed the threshold for Fat at the design level.

Note:	<ul style="list-style-type: none">• The "design" level items are high-level packages (i.e. packages that contain sub-packages).• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the sub-packages of the measured package.
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective with the Package hierarchy selected from the Model menu.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (design) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (leaf package)

No items exceed the threshold for Fat at the leaf package level.

Note:	<ul style="list-style-type: none">• The "leaf package" level items are packages that contain classes.• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the classes in the measured package
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (leaf package) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (class)

No items exceed the threshold for Fat at the class level.

--	--

Note:	<ul style="list-style-type: none">• The value is calculated as the number of dependencies in the child graph for the item, i.e. the dependency graph of the methods and fields in the class.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the item in the Composition perspective.• Use the "Auto-partition" button on the Dependency Graph to help make sense of big graphs.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (class) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Fat (method)

No items exceed the threshold for Fat at the method level.

Note:	<ul style="list-style-type: none">• The value is calculated as the Cyclomatic Complexity (McCabe's metric) of the method.• The value is calculated from the bytecode and so may differ from the value that would be calculated from the corresponding source code in some (rare) cases.• This is not available if the Granularity for the project is "Overview".
Tip:	<ul style="list-style-type: none">• To view and analyze these Fat items, drill down to the class that contains the method in any perspective.• Click on the "Show source" button on the main toolbar to view the source.• The complete list is available in the Measurement perspective. Make sure you are at the root of the "Drill-down" in the top-left, and then select the Fat (method) slice of the pie. The "Items with XS" will contain all the items that exceed the threshold at this level.

Items with highest XS

None

Note:	<ul style="list-style-type: none">• XS is related to the value and the size of the item so that bigger items (usually higher in the hierarchy) score higher. Since excessive structural complexity on bigger items (e.g. packages) will tend to disrupt development activities more than on smaller items (e.g.
--------------	---

	<p>methods), you should consider fixing these first.</p> <ul style="list-style-type: none">• Values in red exceed threshold.
Tip:	<ul style="list-style-type: none">• The complete list is available in the Measurement perspective - make sure you are at the root of the "Drill-down" in the top-left, and do not select any items in the pie.• You can integrate the generation of this summary into your build process. See help for details.

Classstructure

