

Abschlussbericht: TUM Campus App

Betreuung:

Nils Kannengießer
TUM, Fakultät für Informatik
Lehrstuhl 13, Prof. Dr. Uwe Baumgarten
Boltzmannstr. 3
85748 Garching bei München

Inhalt

1. Projektteilnehmer	2
2. Zusammenfassung	2
2.1 Musskriterien	2
2.2 Kannkriterien	2
2.3 Nichtfunktionale Anforderungen	2
3. Produkteinsatz	3
4. Produkt	3
4.1 Projektumsetzung	4
4.2 Benutzeroberfläche	4
5. Qualitätsanforderungen	4
6. Umsetzung	4
7. Ausblick auf die Zukunft	5

1. Projektteilnehmer

- Christoph Richter (Projektumsetzung, GUI-Design, Projektleiter, Student TUM)
Kontakt: christoph_richter@rocketmail.com
- Thomas Bley (Projektumsetzung, SW-Architektur, QA, Student TUM)
Kontakt: thbley@gmail.com
- Andreas Bernhofer (Technischer Partner für Mensa-Schnittstelle, Lehrstuhl 13 TUM)
Kontakt: AndreasBernhofer@googlemail.com
- Nils Kannengießer (Auftraggeber, Betreuung, Lehrstuhl 13 TUM)
Kontakt: nils.kannengiesser@tum.de

2. Zusammenfassung

Im Rahmen des Programmierpraktikums wurde eine mobile Anwendung für Studenten geschaffen, die den mobilen Zugriff auf Veranstaltungen, Nachrichten, Speisepläne der Mensen, usw. ermöglicht. Der Projektplan wurde eingehalten. Ebenso war die technische Umsetzung der Schnittstellen erfolgreich. Die Funktionalität der Schnittstellen sowie der Yahoo! Query Language entsprach der Spezifikation im Pflichtenheft.

2.1 Musskriterien

Zum Pflichtenheft haben sich folgende Abweichungen ergeben:

Nachrichten, RSS-Feeds: Auf die Anzeige kompletter Links wurde auf Grund der besseren Benutzbarkeit verzichtet.

Vorlesungen: Auf den automatischen Import beim Anwendungsstart wurde verzichtet, da hier evtl. bestehende Daten überschrieben werden müssen oder bereits gelöschte Daten durch einen erneuten Import wieder sichtbar werden.

Speisepläne: In den Einstellungen werden die Mensen nur per Suchstring statt per An- und Abwahl einzelner Mensen gefiltert, da die „MultiSelectListPreference“ erst ab Android 3.0 verfügbar ist.

2.2 Kannkriterien

Folgende Kannkriterien wurden umgesetzt: Automatische Stummschaltung, Android Market

Folgende Kannkriterien wurde nicht umgesetzt: Übernahme in Kalender, Preferred Action, Vorlesungsräume auf einer Karte zeigen, Benutzerhandbuch im HTML-Format

Der MVV-Störungsticker wurde als RSS-Feed integriert. Die MVV-Baustellenhinweise und der TUM Roomfinder wurden als Link integriert.

2.3 Nichtfunktionale Anforderungen

Ein festes Zeitlimit für das Herunterladen von Dateien sowie das Aufbauen von Verbindungen wurde nicht implementiert, da je nach Übertragungsstandard (WIFI, EDGE, UMTS, ...) erhebliche Schwankungen in der minimalen und maximalen Übertragungsgeschwindigkeit bestehen.

Die Einhaltung bestimmter Grenzwerte für Metriken (Cyclomatic Complexity, Feature Envy, etc.) wurde auf Grund der geringen Projektgröße vernachlässigt.

Metric	Total	Mean	Std. Dev.	Max...	Resource causing Maximum	Method
▶ McCabe Cyclomatic Complexity		1,831	1,881	17	/tum-campus/src/de/tum/in/tumcampus/services/DownloadService.java	onHandleIntent
▶ Total Lines of Code	4615					
▶ Instability (avg/max per package)		0,502	0,373	1	/tum-campus/src/de/tum/in/tumcampus/receivers	
▶ Number of Parameters (avg/max)		1,025	1,38	10	/tum-campus/src/de/tum/in/tumcampus/models/LectureItem.java	LectureItem

Abbildung 1: Metriken (Ausschnitt)

3. Produkteinsatz

Die Anwendung ist ab Android Version 2.1 lauffähig und erfordert keine Root-Rechte. Die Installation bzw. ein Update kann per Download einer .apk-Datei oder per Download und Compilieren des Quellcodes erfolgen. Die Verteilung über den Android-Market wurde ebenfalls umgesetzt. Ein Service der Anwendung läuft dauerhaft im Hintergrund, um das Gerät während einer Vorlesung automatisch auf lautlos zu schalten.

4. Produkt

Das Produkt umfasst die Android-Anwendung, bestehend aus folgenden Komponenten:

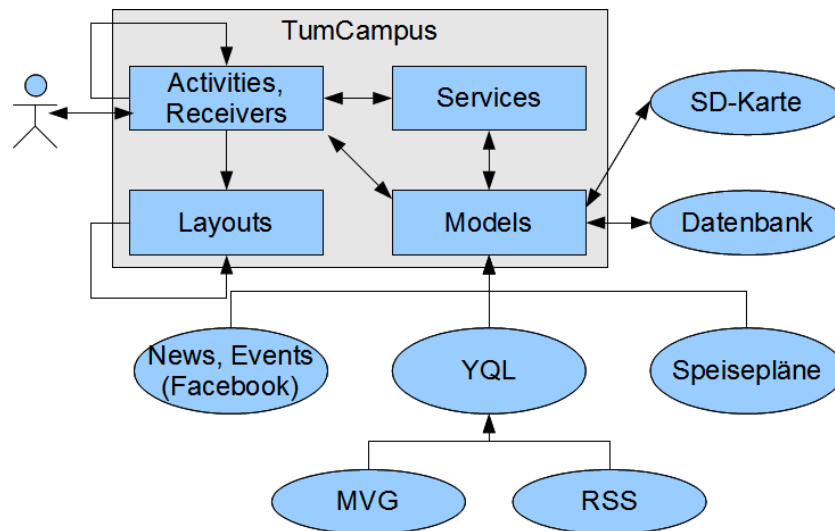


Abbildung 2: Architektur

- Benutzeroberfläche (Activities, Receivers, Layouts):
 - Hauptmenü, App-Info, Einstellungen, Nachrichten, Untermenüs, usw.
- Models:
 - Lesen Daten aus der SQLite-Datenbank zur Ausgabe in der Benutzeroberfläche.
 - Lesen Daten von der SD-Karte und speichern diese in der Datenbank ab.
 - Laden Daten herunter und speichern diese in der Datenbank ab, bzw. schreiben Bilder auf die SD-Karte (Caching).
- Services:
 - Download-Service: Lädt Daten von den Schnittstellen (Facebook, Mensa, MVV, RSS) mit Hilfe der Models herunter und speichert die Daten lokal ab.
 - Import-Service: Lädt Daten von der SD-Karte (CSV- oder URL-Dateien) mit Hilfe der Models und speichert die Daten lokal ab.
 - Silence-Service: Schaltet das Gerät während der Vorlesung auf lautlos.

4.1 Projektumsetzung

Bei der technischen Umsetzung sind folgende Probleme aufgetreten:

Transaktions-Handling der Datenbank

Inserts in die Datenbank sind ohne Transaktionen extrem langsam (<5 pro Sekunde). Ohne Transaktionen würde das Herunterladen der Speisepläne länger als 10 Minuten dauern. Transaktionen beschleunigen die Inserts (~250 pro Sekunde), sperren aber die Datenbank

exklusiv und ermöglichen keinen Zugriff von anderen Threads auf die Datenbank. Dadurch kann es zu Programmabstürzen kommen, wenn sich beispielsweise der Benutzer durch die Speisepläne klickt und der Download-Service im Hintergrund die Datenbank exklusiv gesperrt hat. In der Praxis treten Abstürze aber erst ein, wenn der Benutzer während dem Download mehr als 3 Klicks pro Sekunde macht. Daher wird die Dauer der Transaktionen möglichst kurz gehalten und auf weiteres Error Handling verzichtet.

Eine Lösung für dieses Problem wird leider erst mit Android 3.0 in Form von nicht-exklusiven Transaktionen bereitgestellt (siehe [beginTransactionNonExclusive](#)).

Robotium-Probleme

Beim Robotium-Framework sind Tests manchmal erst beim 2. Durchlauf erfolgreich. Die genaue Ursache konnte bisher noch nicht ermittelt werden.

Die Methode `searchText` kann `ListView`s nicht automatisch scrollen, wenn sich diese in einem `SlidingDrawer` befinden.

Die Methode `goBack` funktioniert nicht bei `Activities` von fremden Prozessen (z.B. Webbrowser).

4.2 Benutzeroberfläche

Die App verwendet das TUM Logo in der Kopfzeile und hat vor allem blau und weiß als Farben. Auf den freien Scrollbereich auf der rechten Seite wurde verzichtet, da die Bedienung auf modernen Touchscreens auch ohne den freien Bereich sehr gut möglich ist. Nachrichten und Info sind im Scrollmenü, Einstellungen über die Menü-Taste erreichbar. Alle Funktionen innerhalb der App sind über maximal 3 Klicks erreichbar. Die Abbildung 1 vermittelt einen kurzen Eindruck des Hauptmenüs.

5. Qualitätsanforderungen

Manuelle Tests wurden mit den Endgeräten „HTC Google Nexus One“ und „Samsung Google Nexus S“ mit der jeweils aktuellsten Android-Version durchgeführt.

Für jede Schnittstelle wurden automatisierte Testfälle erstellt, die die wesentlichen Eigenschaften der zurückgegebenen JSON-Objekte prüfen. Die einzelnen Nutzeraktionen wurden mit Oberflächentests simuliert und erfolgreich überprüft.

Im Rahmen des manuellen Tests traten keine Fehler auf. Alles funktioniert wunschgemäß. Abweichungen vom Pflichtenheft waren an manchen Stellen nötig.

6. Umsetzung

Die technische Produktumgebung wurde auf Android SDK Platform 3.2 revision 1 angehoben. Die Bereitstellung des Quellcodes sowie aller weiteren Dokumente erfolgt über die URL <http://code.google.com/p/tum-campus/>. Als Lizenz wurde die GNU GPL v3 beibehalten. Auf Tests mit dem Android-Emulator wurde verzichtet, da die langsame Ausführungsgeschwindigkeit kein reales Nutzererlebnis widerspiegelt.

7. Ausblick auf die Zukunft

Für die Zukunft soll die Anwendung in ein offenes Community-Projekt überführt werden. Folgende neue Funktionen sind derzeit geplant: Tablet-Version, Internationalisierung, Vorlesungen online suchen, Vorlesungszeiten online importieren und direkt am Gerät ändern, Ergänzung der Speisepläne um Preise und Öffnungszeiten, Mensa-Bewertungssystem, Vorlesungsräume auf einer Karte zeigen, Benutzerhandbuch im HTML-Format, und vieles mehr.



Abbildung 3: Hauptmenü