

Pflichtenheft: 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. Zielbestimmung	2
2.1 Musskriterien	2
2.2 Kannkriterien	3
2.3 Abgrenzung	4
3. Produkteinsatz	4
3.1 Zielgruppe	4
3.2 Anwendungsbereiche	4
3.3 Betriebsbedingungen	4
4. Produkt	4
4.1 Produktfunktionen bzw. Projektumsetzung	5
4.2 Nichtfunktionale Anforderungen	6
4.3 Benutzeroberfläche	6
5. Qualitätsanforderungen	6
6. Umsetzung	7
6.1 Technische Produktumgebung	7
6.2 Projektplan	7
7. Referenzen	8
8. Quellen	8
9. Glossar	8

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. Zielbestimmung

Die TU München gehört zu den größten technischen Hochschulen in Deutschland. Verteilt über mehrere Standorte und Sondereinrichtungen studieren über 26000 Studenten an der TUM. Neben dem zentralen Online-Informationsmanagementsystem TUMonline soll mit diesem Projekt eine inoffizielle mobile Anwendung für Studenten geschaffen werden, die einen mobilen Zugriff auf Veranstaltungen, Nachrichten, Speisepläne der Mensen, usw. ermöglicht. Diese Anwendung dient vor allem der Aggregation von besonders häufig genutzten Informationsquellen.

2.1 Musskriterien

Folgende Kriterien beschreiben unabdingbare Leistungen die das Produkt erfüllen muss:

[1] Veranstaltungen: Innerhalb der Android-Anwendung sollen Veranstaltungen über einen Menüpunkt im Hauptmenü einsehbar sein. Diese Veranstaltungen werden über eine Internetverbindung von einer Facebook-Seite heruntergeladen und lokal auf dem Android-Gerät abgespeichert. Es werden pro Veranstaltung die Felder Name, Zeit, Ort, Beschreibung und – falls vorhanden – ein Link und ein Foto angezeigt. Heruntergeladen und angezeigt werden die letzten 25 Veranstaltungen. Beispiele: TU-Film, Partys, Demos

[2] Nachrichten: Innerhalb der Android-Anwendung sollen Nachrichten über einen Menüpunkt im Hauptmenü einsehbar sein. Diese Nachrichten werden über eine Internetverbindung von einer Facebook-Seite heruntergeladen und lokal auf dem Android-Gerät abgespeichert. Es wird pro Nachricht der Nachrichtentext und – falls vorhanden – ein Link und ein Foto angezeigt. Heruntergeladen und angezeigt werden die letzten 25 Nachrichten. Beispiele: aktuelle Hinweise zum Skriptenverkauf, Studienberatung, etc.

[3] Mensa Speiseplan: Innerhalb der Android-Anwendung sollen die aktuellen Speisepläne der jeweiligen Mensen über einen Menüpunkt im Hauptmenü angezeigt werden. Die jeweiligen Daten werden über eine bereitgestellte Schnittstelle gelesen und lokal auf dem Android-Gerät abgespeichert. Es wird zu einem bestimmten Datum jeweils pro Gericht der Name und der Typ angezeigt. In den Einstellungen kann ein bestimmtes Gelände (z.B. Garching) festgelegt werden (= Gruppe von Mensen). Heruntergeladen und angezeigt werden jeweils die Daten von 1 Woche. Beispiel: Speiseplan Mensa Leopoldstraße 09.05.2011

[4] Vorlesungstermine: Innerhalb der Android-Anwendung sollen die Vorlesungen des Studenten über einen Menüpunkt angezeigt werden. Die jeweiligen Daten zur Vorlesung werden vom Benutzer aus TUMonline per CSV-Datei exportiert und auf der SD-Karte des Android-Geräts gespeichert. Optional können die Daten hier nochmals über MS Excel vom Benutzer angepasst werden. Von der SD-Karte erfolgt der Import automatisch beim Starten der Anwendung. Es werden pro Veranstaltung die Felder DATUM, VON, BIS, TITEL, ORT und – falls vorhanden – ANMERKUNG angezeigt. Beispiel: 02.05.2011; 09:15; 10:00; Verteilte Anwendungen (IN2102); 00.13.009A, Medien (5613.EG.009A); Übung
Die Semestertermine (Feiertage, Ferien, etc.) werden ebenfalls automatisch per CSV-Datei von der SD-Karte oder dem Ressourcen-Verzeichnis der App importiert.

[5] MVV: Innerhalb der Android-Anwendung sollen die aktuellen Abfahrtszeiten für bestimmte Haltestellen aus dem MVV-Bereich angezeigt werden. Der Benutzer soll neben „Garching-Forschungszentrum“ weitere Haltestellen eintragen können. Die aktuellen Daten werden über die Webseite mobil.mvg-live.de gelesen und dargestellt. Es werden jeweils Linie, Zielhaltestelle der Linie und verbleibende Abfahrtszeit in Minuten angezeigt. Beispiel: aktuelle Abfahrtszeiten für Garching-Forschungszentrum

[6] Wichtige Links: Innerhalb der Android-Anwendung sollen dem Benutzer Links zu den Webseiten Infopoint, Studienberatung, Stellenangebote, Fachschaft, OPAC TUM und OPAC LMU angezeigt werden. Weitere Links sollen vom Benutzer in Form von .URL-Dateien (Windows Internetverknüpfung) direkt von der SD-Karte eingebunden werden können. Die Sortierung erfolgt in alphabetischer Reihenfolge.

[7] Weitere News per RSS: Der Benutzer soll über die Einstellungen der App weitere Nachrichtenquellen per RSS-Feeds integrieren können. Diese Nachrichten werden über eine Internetverbindung von der jeweiligen RSS-URL heruntergeladen und lokal auf dem Android-Gerät abgespeichert. Es werden pro Nachricht die Felder Titel, Nachrichtentext, Link und – falls vorhanden – ein Foto angezeigt. Heruntergeladen und angezeigt werden maximal 25 Nachrichten pro RSS-Feed. Beispiel: <http://www.spiegel.de/schlagzeilen/index.rss>

[8] Offline Modus: Über eine asynchrone Hintergrund-Operation sollen alle Quellen aus [1] bis [7] heruntergeladen werden. Über die Benutzeroberfläche kann das Herunterladen für einzelne oder alle Quellen gestartet werden. Die Rückmeldung über den Abschluss bzw. einen Abbruch erfolgt dann ebenfalls über die Benutzeroberfläche. Neue Elemente werden über den jeweiligen Eintrag im Hauptmenü entsprechend markiert. Über die Einstellungen der App soll der Benutzer sämtliche heruntergeladenen Daten löschen können.

[9] App-Info: Über einen Info-Button wird dem Nutzer eine kurze Zusammenfassung über die App sowie ein Impressum angezeigt.

[10] Benutzerhandbuch: Ein Benutzerhandbuch wird in Form einer PDF-Datei bereitgestellt. Es enthält eine Übersicht über die Funktionen und beschreibt kurz, wie diese verwendet werden können. Besonderer Fokus liegt dabei auf Funktionen und Einstellmöglichkeiten, deren Verwendung nicht intuitiv ist.

2.2 Kannkriterien

Die Erfüllung folgender Kriterien wird angestrebt, falls dies zeitlich möglich ist:

Automatische Stummschaltung: Während einer Vorlesung wird das Handy automatisch auf lautlos geschaltet. Die jeweiligen Start- und Endzeiten einer Vorlesung werden aus dem Bereich Vorlesungen entnommen. Die Funktion lässt sich in den Einstellungen aktivieren und deaktivieren.

Übernahme in Kalender: Eine Veranstaltung aus dem Bereich „Veranstaltungen“ kann über einen Button in den Kalender des Endgeräts übernommen werden.

Preferred Action: Darstellung eines bestimmten Buttons direkt unter der Kopfzeile. Auswahl des Buttons nach bestimmten Kriterien: Bei der Uhrzeit „11:30“ wird Mensa angezeigt. 15 Minuten vor Ende einer Vorlesung wird die nächste Vorlesung angezeigt (bzw. MVV falls bereits letzte Vorlesung). Per Default wird ein Button mit der zuletzt aufgerufenen Activity angezeigt.

Vorlesungsräume auf einer Karte zeigen: Verlinkung auf den Roomfinder, z.B. „01.07.023, Seminarraum (5607.01.023)“ => https://portal.mytum.de/displayRoomMap?roomid=01.07.023@5607&disable_decoration=yes

Benutzerhandbuch: Bereitstellung des Benutzerhandbuches im HTML-Format

MVV-Störungsticker: Integration des Störungstickers in die Anzeige der MVV-Abfahrtszeiten (<http://www.mvg-mobil.de/Tickerrss/CreateRssClass>)

MVV-Baustellenhinweise: Integration der Baustellenhinweise (<http://www.mvg-mobil.de/betriebsaenderungen/index.html>)

Android Market: Veröffentlichung der TUM Campus App im Android Market

2.3 Abgrenzung

Die Anwendung wird nur für die Android-Plattform umgesetzt. Die Sprache der Benutzeroberfläche ist Deutsch. Die Anwendung wird lediglich auf den beschriebenen Mobiltelefonen getestet, nicht auf Tablets. Als Zielgruppe werden im Rahmen des Projekts nur Informatikstudenten betrachtet.

3. Produkteinsatz

Das Produkt wird auf Android-Telefonen eingesetzt. Als minimale Version wird Android 2.1 vorausgesetzt. Die Anwendung kann sowohl mit als auch ohne Internetverbindung arbeiten. Für den Abruf aktueller Daten (z.B. MVV) ist eine Internetverbindung notwendig.

3.1 Zielgruppe

Die Zielgruppe des Projekts wird vorerst auf Informatikstudenten der TU München festgelegt. Zukünftig soll die App aber auch für Studenten anderer Fakultäten der TUM angepasst werden. Gleiches gilt für die Erweiterung der Benutzeroberfläche um weitere Sprachen. Weiter wird der Quellcode und die Dokumentation unter der Lizenz GNU GPL v3 offen gelegt, damit eine Portierung auf andere mobile Plattformen (z.B. Apple iOS) schnell und einfach möglich ist.

3.2 Anwendungsbereiche

Unter Betrachtung der Musskriterien und der gewählten Zielgruppe ergeben sich exemplarisch folgende User Stories:

- [1] Veranstaltungen: Als ein Informatik-Student der TUM möchte ich mobil die aktuellen Veranstaltungen für Studenten anschauen, damit ich keine Veranstaltung verpasse.
- [3] Mensa Speiseplan: Als ein Informatik-Student der TUM möchte ich mobil die aktuellen Speisepläne anschauen, um mich zwischen Mensa und Cafeteria zu entscheiden.
- [4] Vorlesungstermine: Als ein Informatik-Student der TUM möchte ich mobil die aktuellen Vorlesungen und Räume nachschauen, damit ich keine Vorlesung verpasse oder im falschen Raum bin. Als ein Informatik-Student der TUM möchte ich mobil die aktuellen Semestertermine anschauen, damit ich am Feiertag nicht versehentlich in die Uni fahre.
- [5] MVV: Als ein Informatik-Student der TUM möchte ich schnell die nächste U-Bahn nachschauen, damit ich pünktlich losgehe.

3.3 Betriebsbedingungen

Die Android-Anwendung soll auf einem Standard-Android-Gerät ohne Root-Rechte laufen. Die Installation bzw. ein Update soll per Download einer .apk-Datei oder per Download und Compilieren des Quellcodes erfolgen. Die Verteilung der Anwendung über den Android-Market ist optional. Ob die Anwendung dauerhaft im Hintergrund laufen muss, wird bei der Umsetzung der automatischen Stummschaltung entschieden (siehe Abschnitt 2.2).

4. Produkt

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

- Benutzeroberfläche: Hauptmenü, Info, Settings, Nachrichten, Untermenüs, usw.
- Download-Service: Lädt Daten von den Schnittstellen (Facebook, Mensa, MVV, RSS) und speichert diese auf der SD-Karte (Bilder) bzw. in die SQLite-Datenbank (JSON-Daten).
- Importer: Liest Daten von der SD-Karte, z.B. CSV- oder URL-Dateien und speichert diese in die SQLite-Datenbank
- DataReader: Liest Daten aus der SQLite-Datenbank zur Ausgabe in der GUI

Weitere Details zur Architektur und der Komponenten werden in der Entwurfsphase festgelegt.

4.1 Produktfunktionen bzw. Projektumsetzung

Im Folgenden werden die Funktionen aus dem Abschnitt 2.1 um eine kurze technische Beschreibung der jeweiligen Schnittstellenaufrufe ergänzt. Die Autoren übernehmen keinerlei Haftung für die korrekte Anzeige der Informationen innerhalb der Anwendung, da das Datenformat von den Betreibern der Schnittstellen ohne Rückmeldung geändert werden kann. Ebenso wird keinerlei Haftung für die Funktionalität und die Verfügbarkeit der verwendeten Schnittstellen übernommen, da dies im Verantwortungsbereich der jeweiligen Betreiber liegt.

[1] Veranstaltungen bei Facebook herunterladen: (Facebook Graph API)

- Access-Token bei Facebook abholen (App Login):
https://graph.facebook.com/oauth/access_token?client_id=141869875879732&client_secret=69b21229a5446f5599ffd3470205bb40&grant_type=client_credentials
=> access_token=141869875879732|FbjTXY-wtr06A18W9wfhU8GCkwU
- Veranstaltungen abrufen:
https://graph.facebook.com/162327853831856/events?access_token=...
=> {"data":[{"name": "Demo gegen ...
- Veranstaltungsdetails abrufen (z.B. ID 116109978474015):
https://graph.facebook.com/116109978474015?access_token=...
=> {"id": "116109978474015", "owner": { ...
Fotos: https://graph.facebook.com/116109978474015/photos?access_token=...
=> {"data":[{"id": "...", "source": "http://a2.sphotos.ak.fbcdn.net/..._2037704_n.jpg",
Links: https://graph.facebook.com/116109978474015/feed/?access_token=...
=> {"data":[{"id": "...", "...link": "http://studiengebuehrenbayern.wordpress.com/"...

[2] Nachrichten bei Facebook herunterladen: (Facebook Graph API)

- Access-Token bei Facebook abholen (App Login): siehe [1]
- Nachrichten abrufen:
https://graph.facebook.com/162327853831856/feed/?access_token=...
=> {"data":[{"id": "...", "...message": "Some news demo", ...

[3] Mensa Speiseplan herunterladen: (Mensa-Schnittstelle)

- Mensa-Liste abrufen:
<http://lu32kap.typo3.lrz.de/mensaapp/exportDB.php>
=> {"mensa_mensen":[{"id":"411", "name":"Mensa Leopoldstra\u00dfe"...
- Speisepläne abrufen (Mensa ID 411):
http://lu32kap.typo3.lrz.de/mensaapp/exportDB.php?mensa_id=411
=> {..."mensa_menu":[{"id":"21502", "...date":"2011-05-20", "type_short":"ae", ...

[5] MVV: (Yahoo! Query Language)

- Abfahrtszeiten abrufen:
Query: select content from html where url="http://www.mvg-live.de/ims/dfiStaticAnzeige.svc?haltestelle=**Garching-Forschungszentrum**" and xpath="//td[contains(@class,'Column')]/p"
Url: <http://query.yahooapis.com/v1/public/yql?format=json&q=<Query>>
=> {"query":{"count":10,...{"p":["Garching-Forschungszentrum","04:24"...
- Haltestelle suchen:
Query: select content from html where url="http://www.mvg-live.de/ims/dfiStaticAuswahl.svc?haltestelle=**Gar**" and xpath="//a[contains(@href,'haltestelle')]"
Url: <http://query.yahooapis.com/v1/public/yql?format=json&q=<Query>>
=> {"query":{"count":8,...{"a":["Botanischer'n Garten", ...

4.2 Nichtfunktionale Anforderungen

Mit Erlaubnis der TU München wird diese Anwendung den Namen und das Logo der „TU München“ verwenden.

Die Anwendung soll stets absturzfrei laufen und keine „Application Not Responding“ Fehler erzeugen. Bei Verbindungsfehlern und Überschreitung einer Verbindungsdauer von 20 Sekunden soll eine Fehlermeldung über die Benutzeroberfläche ausgegeben werden.

Die Anwendung soll personenbezogene Daten nur speichern, wenn der Benutzer diese explizit eingibt oder über ein Import-Verzeichnis auf der SD-Karte zur Verfügung stellt. Die Anwendung soll keinerlei personenbezogene Daten und keinerlei Nutzungsstatistiken an Dritte übermitteln.

4.3 Benutzeroberfläche

Die Oberfläche soll übersichtlich und leicht zu bedienen sein. Das Design soll auf den ersten Blick Assoziationen zur TUM wecken. Um dies zu gewährleisten, werden in erster Linie die Farben blau und weiß verwendet.

Die Abbildung 1 soll hier beispielhaft einen Eindruck vom Hauptmenü vermitteln. Die finale App wird anders aussehen. Das Design ist leicht durch Einfügen neuer Funktionen erweiterbar. Angestrebt wird ein Freibereich auf der rechten Seite. Dieser Bereich soll ein „gefahrloses Scrollen“ ermöglichen, vergleichbar mit dem Startbildschirm von Windows Phone 7. Scrollt man mit dem Finger auf diesen Bereich, klickt man nicht versehentlich etwas an.

In der **Kopfzeile** befinden sich das Logo und der Name der App.

Das **Hauptmenü** soll einen Überblick über die wichtigsten Funktionen verschaffen. Optimalerweise ergänzen sich dafür kleine Logos mit den Namen der Funktionen.

Die erste Funktion oben ist die Preferred Action (siehe 2.2). Möglicherweise wird diese von den Anderen optisch abgehoben.

In der **Fußzeile** werden die Basisfunktionen Nachrichten, Info und Einstellungen dargestellt.

Bei der Integration aller Funktionen soll auf eine möglichst einheitliche Menüführung geachtet werden, die den Anforderungen des Designs und der unterschiedlichen Funktionalitäten gerecht wird. Nach Möglichkeit sollten dabei kaum tief verschachtelte Menüs auftreten.

Weitere Details zur Benutzeroberfläche (insbes. der Punkte aus Abschnitt 2.1) werden im Rahmen der Realisierung ausgearbeitet und in den anschließenden Tests evaluiert.



Abbildung 1: Hauptmenü

5. Qualitätsanforderungen

Die Qualität der Anwendung wird mittels manueller und automatisierter Tests sichergestellt.

Ein manueller Test aller bereits implementierten Funktionen wird jeweils mit dem Erreichen eines Meilensteins durchgeführt (siehe Abschnitt 6.2). Manuelle Tests werden mit den Endgeräten „HTC Google Nexus One“ und „Samsung Google Nexus S“ sowie der jeweils aktuellsten Android-Version durchgeführt. Im Fokus der manuellen Tests stehen Funktionalität und Benutzbarkeit.

Bei der Implementierung soll darüber hinaus auf Änderbarkeit und Erweiterbarkeit der Funktionalität sowie Übertragbarkeit innerhalb der Android-Plattform geachtet werden.

Für jede Schnittstelle sollen automatisierte Testfälle erstellt werden, die bestimmte Eigenschaften der zurückgegebenen JSON-Objekte prüfen (Modultests).

Die einzelnen Nutzeraktionen sollen mit Oberflächentests simuliert und geprüft werden (Black-Box-Tests). Als Grundlage dienen hier u.a. die User Stories aus Abschnitt 3.2.

6. Umsetzung

Die Umsetzung erfolgt im Rahmen des Programmierpraktikums während der Vorlesungszeit im Sommersemester 2011. Weitere Rahmenbedingungen sind durch die Musskriterien (Abschnitt 2.1) und die Schnittstellen (Abschnitt 4.1) gegeben.

6.1 Technische Produktumgebung

Der Entwicklung der TUM Campus App erfolgt mit Java SE 6, Eclipse 3.6.2 und der Android SDK Platform 3.1 revision 2.

Für die Verwaltung des Source Codes (SVN) und der Fehlerberichte wird die Plattform Google Code unter der URL <http://code.google.com/p/tum-campus/> verwendet.

Die Schnittstellen werden zum Beginn des Projekts vollständig festgelegt.

Es soll jeweils zum Ende einer Woche eine lauffähige Version der Anwendung erstellt werden.

Zum Testen werden neben dem Android Emulator in der Android-Version 2.1 die Android-Handys Nexus S und Nexus One in der gerade aktuellen Android-Version verwendet.

6.2 Projektplan

Initialisierung

12.5.2011	Kickoff-Meeting, Vorstellung Projektidee
-----------	--

Analyse

12.5. - 22.5.	Erstellung Pflichtenheft
18.5.	Meeting technische Umsetzung

Entwurf

23.5. - 29.5.	GUI-Design, Software-Architektur
---------------	----------------------------------

Realisierung

30.5. - 5.6.	[1] Veranstaltungen
6.6. - 12.6.	[2] Nachrichten, [3] Mensa Speiseplan
13.6. - 19.6.	[4] Vorlesungstermine
20.6. - 26.6.	[6] Wichtige Links, [9] App-Info
27.6. - 3.7.	[7] Weitere News per RSS
4.7. - 10.7.	[5] MVV, [8] Offline Modus
11.7. - 17.7.	[10] Benutzerhandbuch

Testen

18.7. - 24.7.	QA: Testing, Testdaten, Dokumentation
25.7. - 29.7.	Puffer

Einsatz und Wartung

ab 30.7.2011

Meilensteine: 13.6.2011, 27.6.2011, 11.7.2011, 25.7.2011

7. Referenzen

Facebook Authentication: <https://developers.facebook.com/docs/authentication/>

Facebook Graph API: <https://developers.facebook.com/docs/reference/api/>

Facebook Graph API pages: <https://developers.facebook.com/docs/reference/api/page/>

Facebook-Seite für Veranstaltungen, Nachrichten: <https://www.facebook.com/pages/TUM-Campus-App-for-Android/162327853831856>

Yahoo! Query Language: <http://developer.yahoo.com/yql/>

Yahoo! Query Language Console: <http://developer.yahoo.com/yql/console/>

URL File Format: <http://www.cyanwerks.com/file-format-url.html>

Mensa-Schnittstelle: noch nicht fertig

8. Quellen

TUM in Zahlen, 2011, http://portal.mytum.de/cop/tum_in_zahlen/tum_in_zahlen/

Technische Universität München, 2011, http://de.wikipedia.org/wiki/Technische_Universität_München

Über dieses System TUMonline, 2011, <https://campus.tum.de/tumonline/help.showHtml?pHSPK=500090>

RSS, 2011, <http://de.wikipedia.org/wiki/RSS>

CSV, 2011, http://de.wikipedia.org/wiki/CSV_%28Dateiformat%29

JSON, 2011, <http://de.wikipedia.org/wiki/JSON>

Windows Phone 7, 2011, http://en.wikipedia.org/wiki/Windows_Phone_7

9. Glossar

QA: Quality Assurance

MVV: Münchner Verkehrs- und Tarifverbund GmbH (öffentliche Verkehrsmittel, U-Bahn, S-Bahn, Tram, Bus)