



**HOCHSCHULE TRIER**  
Trier University of Applied Sciences  
**Informatik - Computer Science**

---

## Entwicklung mobiler Applikation zur zentralen Verwaltung WG-typischer Aufgaben

Development of an mobile application for central administration to  
manage flat sharing tasks

Tobias Barwig, Robert Raschel, Simon Ritzel

Bachelor-Projektarbeit

Betreuer: Prof. Dr. Georg Rock

Trier, 18.12.2014

---

## Kurzfassung

In dieser Projektarbeit wird die Problematik der Verwaltung von Haushaltsaufgaben in einer Wohngemeinschaft mit meist jungen erwachsenen Studenten behandelt. Ziel ist das oftmals auftretende Chaos (durch Zettelwirtschaft und schlechter Kommunikation) möglichst gering zu halten und die anstehenden Aufgaben und Termine jederzeit klar definiert jedem Mitbewohner zugänglich zu machen. Dieses Ziel soll durch eine mobile Applikation auf Basis des von Google entwickelten Betriebssystems Android erreicht werden. Als Trägermedium dient hierbei ein handelsübliches Smartphone. Die App bietet jedem Mitbewohner die Möglichkeit Notizen sowie Kalendereinträge einzusehen und zu erstellen. Außerdem ist es möglich eine Einkaufsliste zu verwalten in der Artikel hinzugefügt, gelöscht und als gekauft markiert werden können. Des Weiteren zeigt ein Putzplan die noch anstehenden bzw. bereits erledigten Aufgaben aus dem WG Haushalt an. Für den Putzplan muss vor dem ersten Gebrauch der WG-Administrator, der auch unter anderem die Benutzer verwaltet, selbst definierbare Aufgaben mit Beschreibung, Zyklus und Mitbewohner anlegen.

This project thesis displays the often problematic administration of the various household tasks within a flatsharing student community. Objective is to reduce the mostly chaotic atmosphere (be it due to jumble of bits of paper or bad communication) by defining due tasks and deadlines and publishing them to every flatmate in an easy manner. This should be reached by using a mobile application based in an operation system from Google, called "Android". Carrier medium should be a common smartphone. The application out every roommate into position to manage and create different notes and calendar appointments. Furthermore it is possible to manage a shopping list where articles can easily be added, deleted or marked. In addition to this shows

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Problemstellung</b>	<b>1</b>
1.1	Zielsetzung	2
1.2	Ähnliche Apps	3
<b>2</b>	<b>Prototyp</b>	<b>4</b>
2.1	Implementierung	4
2.1.1	Datenbankstruktur	5
2.1.2	PHP-Skripte	5
<b>3</b>	<b>Implementierung</b>	<b>7</b>
3.1	App	8
3.2	Putzplan	9
3.2.1	Umsetzung	9
3.2.2	Probleme und Lösungen	9
<b>4</b>	<b>Zusammenfassung und Ausblick</b>	<b>10</b>
	<b>Glossar</b>	<b>11</b>
	<b>Erklärung der Kandidaten</b>	<b>12</b>

---

## Abbildungsverzeichnis

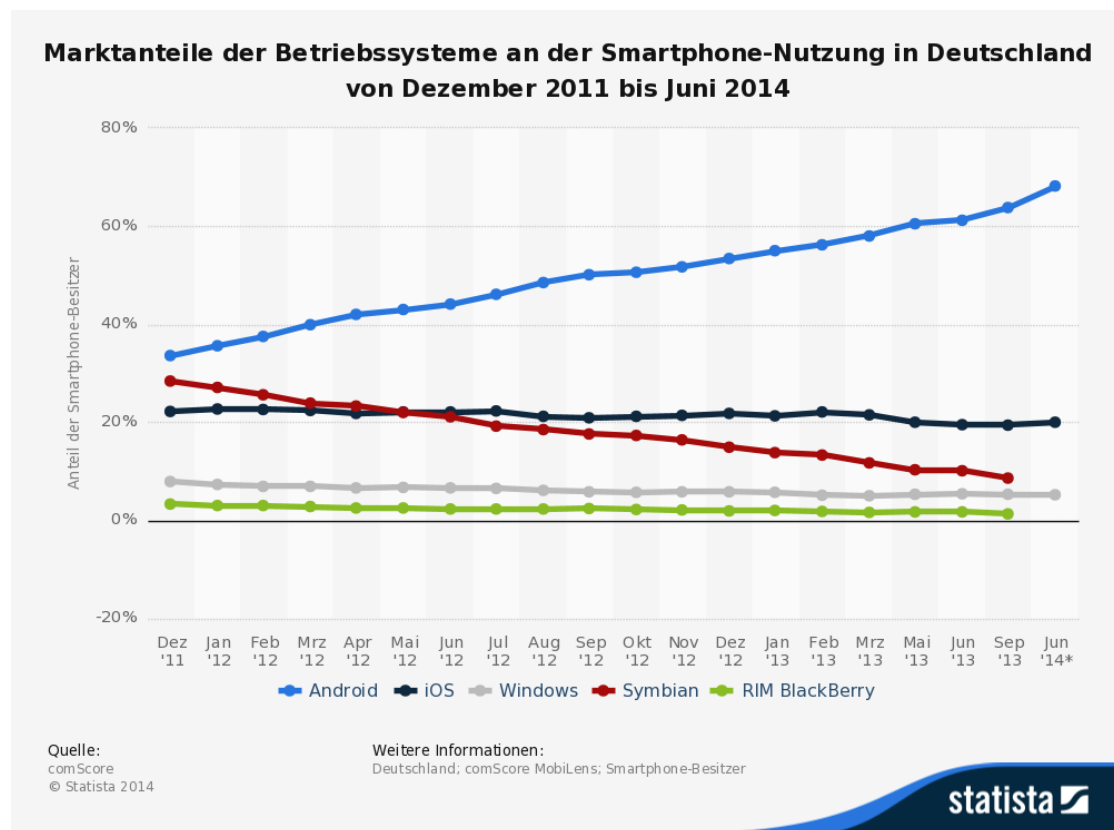
1.1	Marktanteile der Betriebssysteme an der Smartphone-Nutzung in Deutschland von Dezember 2011 bis Juni 2014.....	2
2.1	Aufbau der Tabelle tp_test .....	5
2.2	Aufbau der Tabelle users .....	5
2.3	Struktur der PHP-Skripte im Dateisystem .....	6

## Einleitung und Problemstellung

Mit der steigenden Anzahl von Studierenden an den Bildungseinrichtungen Deutschlands erfreuen sich die Wohngemeinschaften immer größerer Beliebtheit. Mit den vielen neuen Studenten vergrößern sich gleichzeitig die Wohngemeinschaften und die Probleme die mit solch zusammengewürfelten Gemeinschaften einhergehen. Viele Studierende kennen das Problem einer chaotischen Zettelwirtschaft in Küche, Bad und Flur. Für jeden Raum gibt es meistens einen eigenen Plan und jeder dieser Pläne muss nach jeder Woche mit Stift und Papier aufwendig aktualisiert werden. Geht dabei einmal was vergessen so geriet das ganze Rotationssystem ins Schwanken. Dadurch werden Aufgaben nicht erfüllt, Termine nicht eingehalten oder die Verbrauchsartikel für Bad und Küche sind aufgebraucht bevor einer der Mitbewohner überhaupt bemerkt, dass der Artikel bereits fehlt und dringend auf die Einkaufsliste gehört.

**Haushaltsaufgaben müssen sinnvoll und gerecht auf alle Mitbewohner verteilt werden und der aktuelle Stand zu jedem beliebigen Zeitpunkt für jeden einsehbar sein.**

Ist die Wohngemeinschaft bereits an diesem oder einem ähnlich chaotischen Punkt angelangt, bedarf die Verwaltung des Putzplans, der Einkaufsliste und der Kommunikation einer guten Optimierung. Wichtig bei der Optimierung ist, dass ein Medium gewählt wird, welches von jedem Mitbewohner, zu jeder Zeit und von jedem Ort gleichermaßen gut bedient werden kann. Die Lösung soll eine mobile Applikation auf einem Smartphone sein, in der sich jedes WG-Mitglied einloggen und auf die aktuellsten Informationen der WG zugreifen kann. Dass das Smartphone als neues Trägermedium das Blatt Papier ablösen soll liegt nahe, weil nahezu alle der durchweg jungen erwachsenen Personen unserer Zielgruppe solch ein Gerät besitzen. Außerdem bietet es mit der hohen Konnektivität die perfekte Grundlage alle Mitbewohner jederzeit auf dem gleichen Wissensstand zu halten, auch von unterwegs. Als Betriebssystem kommt die von Google entwickelte Android Plattform zum Einsatz. Durch deren hohen Marktanteil von 68,2% in Deutschland wird hier die größte Anzahl an potentiellen Nutzern erreicht (siehe Abb. 1.1).



**Abb. 1.1.** Marktanteile der Betriebssysteme an der Smartphone-Nutzung in Deutschland von Dezember 2011 bis Juni 2014

## 1.1 Zielsetzung

Einer der WG-Mitbewohner erklärt sich bereit die Aufgaben des WG-Administrators zu übernehmen. Dieser WG-Administrator registriert sich und legt dabei eine neue WG an. Zu seinen Aufgaben gehört unter anderem die Verwaltung der Mitbewohner sowie das Pflegen des Putzplans. Neue Mitbewohner müssen vom WG-Administrator per E-Mail in die WG eingeladen werden. Für den Putzplan definiert er Aufgaben die in einem einstellbaren Rhythmus wiederholt werden und wählt zu jeder Aufgabe einen Mitbewohner aus. Nun beginnt der Rhythmus zu laufen und die Aufgaben wechseln nach Erledigung automatisch zu der nächsten Person. Jeder der Mitbewohner kann nach einem Login alle Informationen der WG einsehen und bearbeiten. Auf dem Schwarzen Brett für Notizen können Einträge angezeigt, erstellt und gelöscht werden. In der Einkaufsliste können Artikel hinzugefügt und entfernt werden. Wurde ein Artikel gekauft, kann derjenige den Artikel als gekauft markieren. Alle Änderungen eines Mitbewohners ist für alle anderen Mitglieder der WG sichtbar. Alle Informationen einer WG werden serverseitig in einer Datenbank und clientseitig auf dem Smartphone des Benutzers gespeichert. Bei jedem Start der App wird ein Datenabgleich der auf Client- und Serversei-

te gespeicherten Informationen durchgeführt und alle voneinander abweichenden Daten auf einer Übersichtsseite dem Benutzer als „Neu“ aufgelistet.

## 1.2 Ähnliche Apps

Es gibt bereits eine Auswahl an Apps, die sich jeweils an einem kleinen Teilbereich unseres Funktionsumfangs orientieren und dies gut umsetzen. Hierbei sind einzelne Apps für Einkaufslisten wie „Shopping List“<sup>1</sup> oder Putzpläne wie „Roomboard - Cleaning Roster“<sup>2</sup>. Jedoch gibt es eine weitere App die sich stark an unserer Idee mit dem Funktionsumfang orientiert. Die App „Flatastic: Die WG-App“<sup>3</sup> bietet neben einer Einkaufsliste, einem Putzplan und einer Pinnwand zusätzlich einen Ausgabenrechner, womit alle für die WG getätigten Einkäufe zusammengerechnet werden können. Wir beschränken uns in dieser Ausarbeitung dennoch weiter auf unseren festgelegten Funktionsumfang und können uns nach der Fertigstellung nach wie vor dazu entscheiden weitere Zusatzfunktionen zu implementieren.

---

<sup>1</sup> „Shopping List“ in Google Play Store

<sup>2</sup> „Roomboard- Cleaning Roster“ in Google Play Store

<sup>3</sup> „Flatastic: Die WG-App“ in Google Play Store

## Prototyp

Um vermeidbare Verzögerungen in der späteren Entwicklung möglichst auszuschließen, wurde der Einsatz von Prototypen entschieden. Dazu werden die benötigten Komponenten genauer betrachtet und auf ihre Umsetzbarkeit hin untersucht. Bei einer Android-App, die auf Inhalte aus einer Datenbank zugreift, ist das die Schnittstelle zur Datenbank, beziehungsweise die Netzwerkverbindung.

Da eine direkte Verbindung zum Datenbankserver aus dem Android Betriebssystem nicht möglich ist, wurde die Authentifizierung des Benutzers als weitere Problemstelle markiert. Den die Verwendung von Datenbankbenutzern fällt dadurch weg und muss andersweitig gelöst werden. Um bei diesen kritischen Stellen nicht in bredouille zu geraten, wurde dafür ein Prototyp entwickelt.

### 2.1 Implementierung

Als eine einfache und schnelle Art der Umsetzung haben wir uns für die PHP-Variante entschieden. Dabei werden die Daten von einem PHP-Skript aus der Datenbank gelesen und in eine JSON-Datenformat gebracht. Dieses Objekt wird in einem HTTP-Paket an die App übertragen. In der App sorgt dann ein JSON-Parser für das Auslesen der Daten, welche anschließend direkt verwertet werden oder zunächst in der lokalen SQLite-Datenbank vorgehalten werden.

Alternativ zu dieser Lösung wäre ein RESTful Web Service gewesen. Dieser Lösungsansatz wäre jedoch mit einem größeren programmiertechnischen Aufwand, sowie einer umfangreicheren Serverkonfiguration verbunden gewesen.

Da beide Schwerpunkte in einem Prototyp getestet wurden, wird auf eine weitere Trennung verzichtet.



### 2.1.1 Datenbankstruktur

Begonnen wurde die Umsetzung mit der Definition der Datenbankstruktur sowie deren Umsetzung. Dazu wurden zwei einfache Tabellen angelegt. Die Tabelle `tp_test2.1` wird für die Schreib- und Lesevorgänge verwendet. Um alle nötigen Datentypen testen zu können wurden verschiedene Spalten verwendet. Dadurch konnte auch der spätere Einsatz besser simuliert werden.

```
1 CREATE TABLE IF NOT EXISTS 'test_tp' (  
2   'uid' VARCHAR(23) NOT NULL,  
3   'msg' TEXT,  
4   'nmbr' INT(11) DEFAULT NULL,  
5   'created_at' TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
6 ) ENGINE=MyISAM DEFAULT CHARSET=utf8 ;
```

Abb. 2.1. Aufbau der Tabelle `tp_test`

Um die Benutzerverwaltung für den Prototypen zu simulieren wurden außerdem noch eine Tabelle `users 2.2` angelegt. Darin wurden Informationen zu den Benutzern hinterlegt. Zum Beispiel eine eindeutige ID, sowie Vor- und Nachname. Zum Authentifizieren wurde die E-Mailadresse und ein beliebiges Passwort verwendet. Um das Passwort nicht im Klartext zu speichern, wurde es zusammen mit einem Salt als Hash-Wert abgelegt.

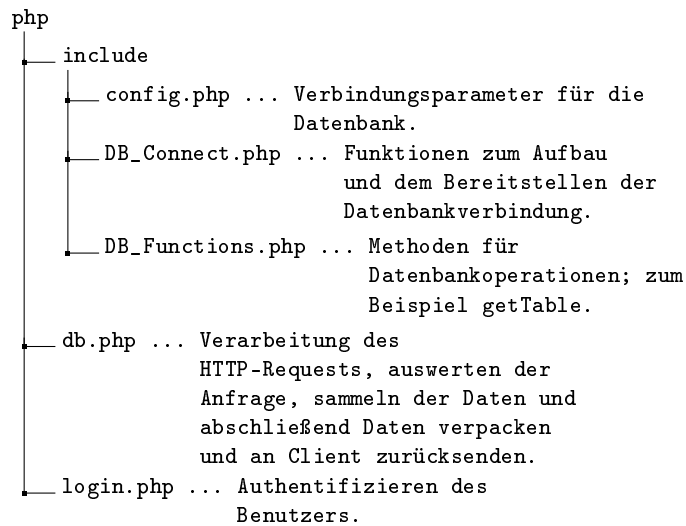
```
1 CREATE TABLE IF NOT EXISTS 'users' (  
2   'uid' INT(11) NOT NULL AUTO_INCREMENT,  
3   'unique_id' VARCHAR(23) NOT NULL,  
4   'firstname' VARCHAR(50) NOT NULL,  
5   'lastname' VARCHAR(50) NOT NULL,  
6   'username' VARCHAR(20) NOT NULL,  
7   'wgId' INT(11) NOT NULL,  
8   'email' VARCHAR(100) NOT NULL,  
9   'encrypted_password' VARCHAR(80) NOT NULL,  
10  'salt' VARCHAR(10) NOT NULL,  
11  'created_at' DATETIME DEFAULT NULL,  
12  PRIMARY KEY ('uid')  
13 ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

Abb. 2.2. Aufbau der Tabelle `users`

### 2.1.2 PHP-Skripte

Der Aufbau der PHP-Schnittstelle ist simpel umgesetzt, da nicht viele Funktionen für den Prototyp benötigt werden. Trotzdem wurde auf eine übersichtliche Datei- und Ordnerstruktur, als auch auf einen modularen Aufbau geachtet. Wie in Abbildung 2.3 Struktur der PHP-Skripte im Dateisystem zu sehen, wurden der Aufbau zunächst in zwei Kategorien aufgeteilt. Funktionen, die direkt auf der Datenbank ausgeführt werden, sowie das Verbinden und Bereitstellen des Datenbankobjekts

übernehmen, sind im Ordner `php/include` untergebracht. Alle weiteren Funktionen die aus der App heraus erreichbar sein sollen, befinden sich im Hauptordner `php`.



**Abb. 2.3.** Struktur der PHP-Skripte im Dateisystem

## Implementierung

Die App wurde zum Großteil in der aktuell von Google empfohlenen Umgebung Android Studio umgesetzt. Zu Beginn fand die Entwicklung noch in Eclipse mit dem entsprechenden Plug-In statt. Jedoch erschwerten die Bugs und die Behäbigkeit der Eclipse-IDE den zügigen Fortschritt. Aus diesem Grund wurde nach dem Legen des Grundsteins das Projekt auf die neue Entwicklungsumgebung migriert, wo es auch fertiggestellt wurde. . . .

---

## 3.1 App

## **3.2 Putzplan**

### **3.2.1 Umsetzung**

### **3.2.2 Probleme und Lösungen**

## Zusammenfassung und Ausblick

In diesem Kapitel soll die Arbeit noch einmal kurz zusammengefasst werden. Insbesondere sollen die wesentlichen Ergebnisse Ihrer Arbeit herausgehoben werden. Erfahrungen, die z.B. Benutzer mit der Mensch-Maschine-Schnittstelle gemacht haben oder Ergebnisse von Leistungsmessungen sollen an dieser Stelle präsentiert werden. Sie können in diesem Kapitel auch die Ergebnisse oder das Arbeitsumfeld Ihrer Arbeit kritisch bewerten. Wünschenswerte Erweiterungen sollen als Hinweise auf weiterführende Arbeiten erwähnt werden.

# A

---

## Glossar

DisASter	DisASter (Distributed Algorithms Simulation Terrain), A platform for the Implementation of Distributed Algorithms
DSM	Distributed Shared Memory
AC	Linearisierbarkeit (atomic consistency)
SC	Sequentielle Konsistenz (sequential consistency)
WC	Schwache Konsistenz (weak consistency)
RC	Freigabekonsistenz (release consistency)

## B

---

### Erklärung der Kandidaten

- ☐ Die Arbeit habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen- und Hilfsmittel verwendet.
- ☐ Die Arbeit wurde als Gruppenarbeit angefertigt. Meine eigene Leistung ist ...

Diesen Teil habe ich selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Namen der Mitverfasser: ...

---

Datum

---

Unterschrift der Kandidaten