

Technologiemanager

„FreeSpace“ Projekt SWE

09.11.2016

1. Ziele
 1. Client
 2. Server
2. Implementierung
 1. Coding
 2. Services
 3. Objektübertragung

Ziele

Client:

- Smartphones (auch **lauffähig** auf Tablets) auf denen Android 4.4 oder höher läuft
- Kamera zwingend erforderlich

Server:

- Java-Applikation
- Betriebssystem: JAVA-Laufzeitumgebung

Implementierung

- Android Studio als Entwicklungsumgebung für Client
- In Android Studio eingebauter Emulator sowie eigene Geräte zum Testen
- Eclipse oder IntelliJ als Entwicklungsumgebung für Server
- Codesharing über GitHub

Server:

- REST-Service über Jersey und Grizzly
- SQLite Datenbank
- Maven als Buildmanager

Client:

- Hauptsächlich Android-Standard Methoden zum Design
 - Verringert Dateigröße
 - Weniger Programmieraufwand
- Login wird über Google Play Services realisiert
 - Verwendete API: vermutlich Firebase (wenn lauffähig unter 4.4), ansonsten Google Play Developer API
- Logik wird komplett auf den Server ausgelagert

- JSON zur Objektübertragung
 - Bibliothek zur (De-)Serialisierung: Google GSON

Gründe

Android

- 4.4 – 70% Abdeckung (ausreichend für Kunden)
 - Viele neue API Funktionen

Java

- Plattformunabhängig (einfacher zum Programmieren)
- Viele frei verfügbaren Bibliotheken für unseren Anwendungsfall
- Allen bekannt

Jersey als JAX-RS

- Gut Dokumentiert
- Vielen aus anderen Fächern (GPM) bekannt
- REST-Service

Grizzly als HTTP-Server

- Arbeitet gut mit Jersey zusammen
- Keine 2 Baustellen, die man bspw. mit einem zusätzlichen Apache hätte

SQLite

- Performant und klein
- SQL ist bekannt
- Einfache Implementierung, da kein dedizierter Server notwendig

Android Standard

- Verringert Dateigröße
- Weniger Programmieraufwand

FireBase

- Bietet viele Komfortfunktionen
 - Beispiel: kleinere Änderungen in der App über Webconsole und ohne Kompilierung
- Möglicherweise nicht lauffähig unter 4.4, dann Google Dev API

Logik auf Server

- Ressourcensparend für ältere und schwache Geräte

JSON

- Geringere Übertragungsgröße als bspw. XML dank weniger Overhead
- Sprachenunabhängig falls sich doch gegen Java entschieden wird

GSON

- Einfach zu bedienen
- Frei verfügbar

FH Aachen
Fachbereich Elektrotechnik und Informationstechnik

www.fh-aachen.de