

Entwurfsbeschreibung

hg17b - Android App für Weiterbildungsmanagement

Julian Dietz, Marcel Herhold, Nikolai Kortenbruck

29. Januar 2018

Inhaltsverzeichnis

1	Allgemeines	2
2	Produktübersicht	2
3	Grundsätzliche Struktur- und Entwurfsprinzipien	2
4	Struktur- und Entwurfsprinzipien einzelner Pakete	2
4.1	Server	2
4.2	Applikation	2
5	Datenmodell	3
6	Glossar	4

1 Allgemeines

Wir entwickeln eine Android-App für Schüler und Veranstalter von Weiterbildungsereignissen. Schüler bekommen eine Übersicht über ihre bereits besuchten Veranstaltungen und haben die Möglichkeit, kommende Veranstaltungen zu finden und sich unverbindlich anzumelden. Veranstalter können in der App die tatsächlich anwesenden Schüler, mithilfe einer dem Schüler zugeteilten ID, erfassen.

2 Produktübersicht

Unser Produkt besteht aus einem Frontend und einem Backend.

Das Frontend ist die Android App. Beim Start der App wird unterschieden ob der Nutzer ein Schüler oder ein Veranstalter ist. Schüler geben dabei ihre ID ein und erhalten Einblick in ihre Eventdaten, zukünftige Events sowie ihren Platz in der Bestenliste. Veranstalter können sich nach einem Klick auf die untere Schaltfläche autorisieren und anschließend die Teilnehmer einer Veranstaltung erfassen. Das Frontend wurde mithilfe von Android Studio entwickelt.

Unser Backend besteht aus einem Server mit RDF-Triplestore in welchem die Eventdaten und die Daten der Schüler gespeichert werden. Der Server nutzt hierfür das Apache-Jena Framework.

3 Grundsätzliche Struktur- und Entwurfsprinzipien

Die Daten der Schüler (IDs, Score, Rang) liegen zentral auf einem Server. Die Kommunikation zwischen mobiler Android App und Server laufen über eine auf dem Server liegende Java Applikation. Diese stellt einen Server Socket zur Verfügung über den ein Client Socket der App mit dem Server kommuniziert.

4 Struktur- und Entwurfsprinzipien einzelner Pakete

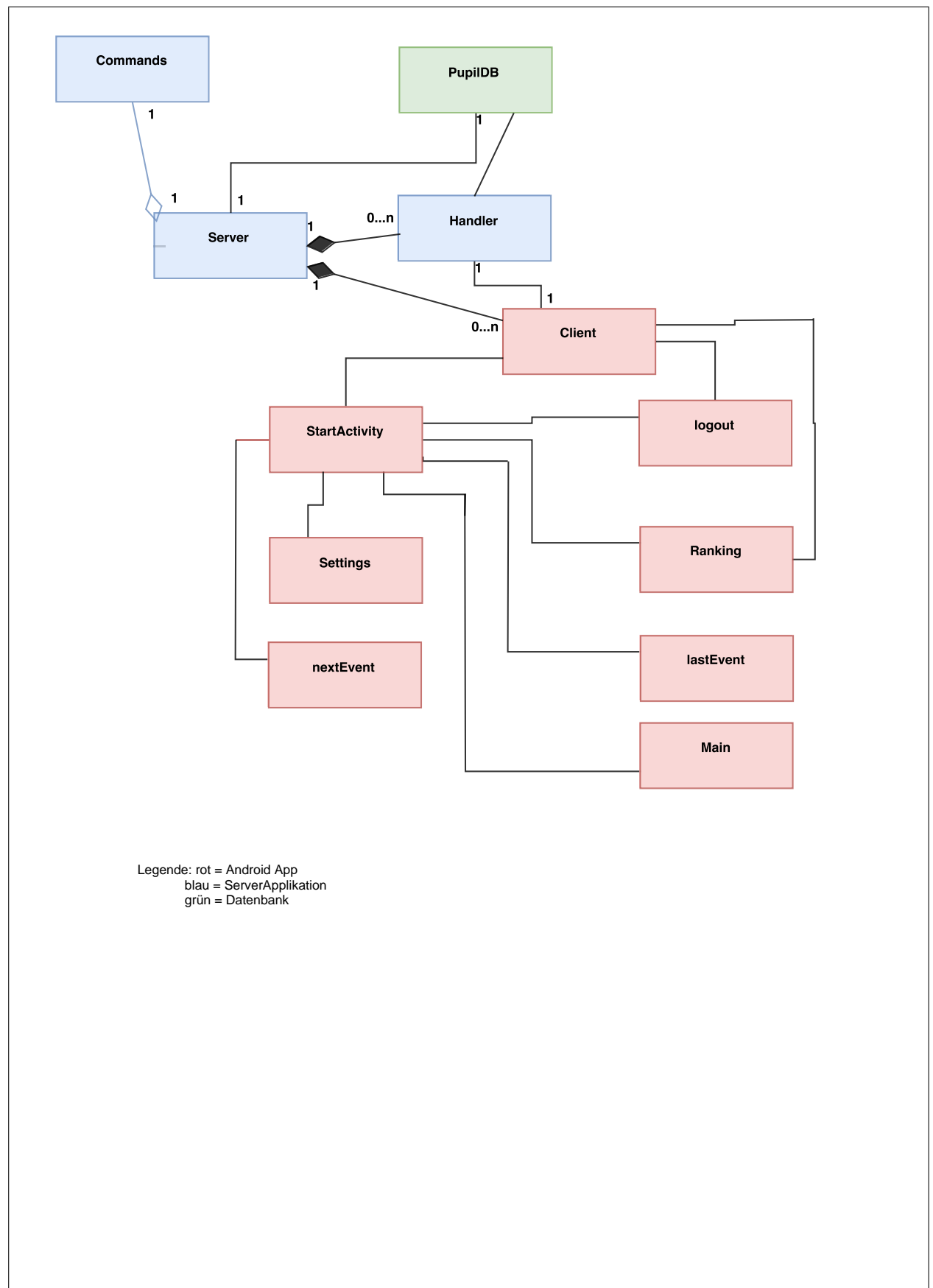
4.1 Server

Die (Schüler-) Daten auf dem Server liegen im SPARQL-Format als Triples in einem Graphen vor und werden von der PupilDB Java-Klasse aus abgerufen und manipuliert. Diese definiert die dafür benötigten SPARQL- Query und Update Statements. Die Server Klasse der auf dem Server laufenden Java Applikation handelt den Verbindungsaufbau und die Abfragen/Antworten vom/an den Client.

4.2 Applikation

Die Kommunikation mit dem Server wird in der Client Klasse der App gehandelt. In der StartActivity Klasse wird das Start Verhalten und der Login der Nutzer definiert. Die Klassen Logout, NextEvents, Ranking und Settings (zu finden in app/src/main/java/hg17b) sind jeweils für die Methoden der in der App vorhandenen Seiten Logout, Nächste Events, Ranking und Einstellungen zuständig. Die Layouts der Seiten werden in den dementsprechenden fragment xml Dateien definiert (zu finden in app/src/main/res/layout). Das Layout des Hauptmenüs hingegen wird in der navigation_menu unter /App/app/src/main/res/menu definiert.

5 Datenmodell



6 Glossar

Backend Backend beschreibt das auf dem Server laufende Programm, ein SPARQL Endpunkt (Apache Jena), sowie Java und Java Applikationen zum bearbeiten diverser Anfragen.

Frontend Frontend beschreibt das auf dem Endgerät des Nutzers laufende Programm.

ID (Identifikationsnummer) Im Zusammenhang dieses Projektes ist die Identifikationsnummer auf den Pässen der Schüler gemeint.

RDF RDF steht für Resource Description Framework. Es wurde von der W3C als Standard zur Beschreibung von Metadaten konzipiert. Es gilt als ein grundlegender Baustein des Semantischen Webs. RDF ähnelt den klassischen Methoden zur Modellierung von Konzepten wie UML-Klassendiagramme und ER-Modell.

Schüler Der Schüler nimmt am Ferienpass teil. Er soll die App leicht bedienen können.

Veranstalter Der Veranstalter organisiert Freizeitaktivitäten für die Schüler. Er soll mit der App den Code des Schülers einscannen.