

Entwurfsbeschreibung

hg17b - Android App f r Weiterbildungsmanagement

Julian Dietz, Marcel Herhold, Nikolai Kortenbruck

29. Januar 2018

Inhaltsverzeichnis

1 Allgemeines	2
2 Produkt bersicht	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 läuft 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. Nachfolgend ein Überblick der Klassen, die für den Server verantwortlich sind.

4.1.1 Server(Klasse)

Diese Klasse ist die Mainklasse des Servers und erstellt die Datenbank und startet den Server, außerdem wird ein Handlerobjekt erstellt, wenn sich ein Client mit dem Server verbindet und es kann eine Liste von allen Clients angezeigt werden.

Julian Dietz, Marcel Herhold, Nikolai Kortenbruck

4.1.2 Commands

Durch diese Klasse werden die Befehle für die Serveradministratoren zur Überwachung des Servers bereitgestellt. Im moment während dies :

"list" (eine liste aller verbundenen Clients)

"stop" zum herrunterfahren des Servers

"help" um alle Befehle aufzulisten.

4.1.3 Handler

Diese Klasse kümmert sich um die Kommunikation zwischen Client, Server und Datenbank. Dabei wartet sie auf Anfragen des Clients und sendet Daten aus PupilDB und wird beendet wenn der Client offline ist

4.1.4 PupilDB

Diese Klasse ist die Datenbank des Servers und liefert Methoden zum lesen und manipulieren der Schülerdaten die als Triple in Graphen gespeichert werden. Die Methoden nutzen SPARQL-Querys und Update Statements.

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. Nachfolgend ein kurzer Überblick von den Klassen der App

4.2.1 Client

Die Klasse Client erstellt einen Hintergrundthread, welcher sich mit dem Server verbindet. Sie sendet und empfängt Daten von der Datenbank des Servers

4.2.2 StartActivity

Diese Klasse ist die Startseite der App und definiert beim starten das Layout und erstellt eine Instanz der Klasse Client. Nach dem klicken auf „Bestätigen“ wird die eingegebene ID mit der Datenbank abgeglichen und wenn diese vorhanden ist wird zur Klasse Main weitergeleitet

4.2.3 Main

Die Klasse Main bestimmt das Design unserer Hauptseite. Das ist die Seite welche ein Schüler nach dem Einloggen als erstes sieht. Oben wird die eigene ID angezeigt, in der Mitte erfährt der Schüler wie viele Punkte er hat und unten werden bevorstehende Veranstaltungen angezeigt, für welche der Schüler sich angemeldet hat

4.2.4 Nextevents

Julian Dietz, Marcel Herhold, Nikolai Kortenbruck

Die Klasse NextEvents soll die bevorstehenden Veranstaltungen anzeigen. Zum jetzigen Zeitpunkt besitzt die Klasse keinerlei Funktionalität. Diese folgt im zweiten Release mit der Einbindung der Veranstaltungsdatenbank.

4.2.5 Lastevents

Die Klasse LastEvents soll die vom eingeloggten Schüler besuchten Veranstaltungen anzeigen. Auch hier besitzt die Klasse noch keine Funktionalität, da die Veranstaltungsdatenbank noch nicht integriert ist.

4.2.6 Ranking

Die Klasse zeigt die Topliste, also die höchsten Punktzahlen in absteigender Reihenfolge, geliefert aus der Klasse PupilDB. Unter der Rangliste wird die Platzierung des angemeldeten Schülers gezeigt.

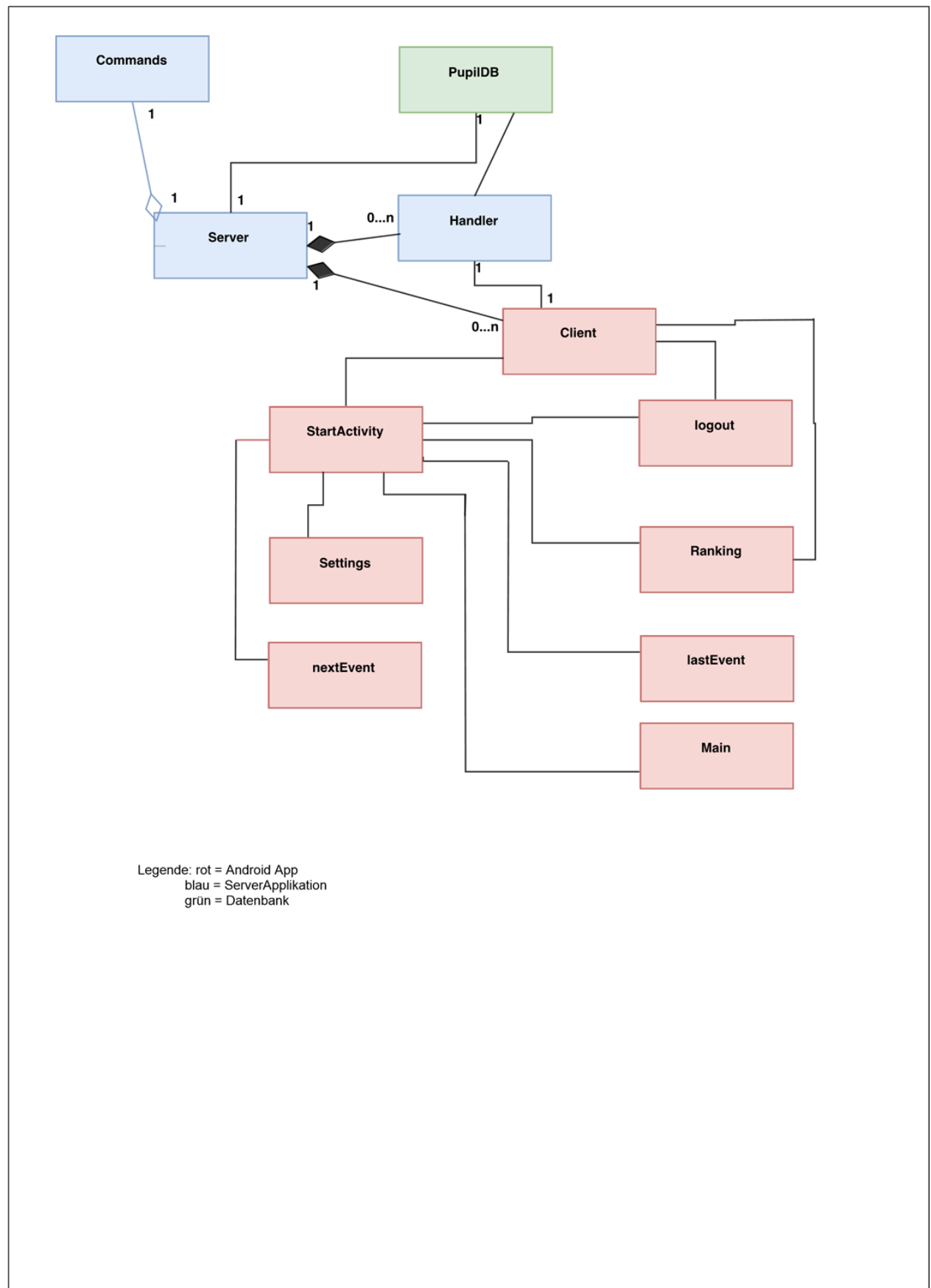
4.2.7 Settings

In dieser Klasse sollen dem Schüler Einstellungsmöglichkeiten zur Individualisierung der App geboten werden. Zurzeit ist die implementierung dieser Funktionen ein Kann-Ziel vom 2. Release.

4.2.8 LogOut

Die Klasse LogOut dient zum trennen der Serververbindung, klickt man auf den Button abmelden, so wird der Hintgrundthread geschlossen und man landet auf der StartActivity Seite um sich erneut anzumelden.

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 Endger t des Nutzers laufende Programm.

ID (Identi kationsnummer) Im Zusammenhang dieses Projektes ist die Identi kationsnummer auf den P ssen der Sch ler gemeint.

RDF RDF steht f r 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 hnelt den klassischen Methoden zur Modellierung von Konzepten wie UML-Klassendiagramme und ER-Modell.

Sch ler Der Sch ler nimmt am Ferienpass teil. Er soll die App leicht bedienen k nnen.

Veranstalter Der Veranstalter organisiert Freizeitaktivit ten f r die Sch ler. Er soll mit der App den Code des Sch lers einscannen.