

Objektmodell: Design

Felix Höffken, Sebastian Raitza, Nico von Geyso

1 Klassendiagramm

Alle Klassen sind auch als Diagramm in den Dateien *objektdiagramm_entity.pdf* und *objektdiagramm_permission.pdf* zu finden.

2 Packet: **db.project.entity**

Advisor Ein Advisor repräsentiert einen Betreuer, also eine Person, die für ein Team arbeitet (z.B. Trainer, Teamarzt, ...).

Country Country repräsentiert ein Land.

GroupMatch Ein GroupMatch ist ein Spiel, dass in der Gruppenphase ausgetragen wird.

GroupStage Die GroupStage ist die Gruppenphase des Turniers.

KnockOutMatch Ein KnockOutMatch ist ein Spiel, dass in der KO-Runde ausgetragen wird.

Match Ein Match ist ein Spiel, also die Oberklasse von KnockOutMatch und GroupMatch. Sie ist *abstract* kann also nicht implementiert werden.

MatchEvent Ein MatchEvent ist ein Event, dass während eines Spiels auftreten kann. Es ist *abstract* kann also nicht direkt implementiert werden. Es gibt mehrere Unterklassen, die später erklärt werden.

Person Eine Person ist eine Person, die in dem Turnier auftritt. Sie ist *abstract* und die Überklasse von Player und Advisor.

Player Ein Player repräsentiert einen Spieler im Turnier.

Stadium Ein Stadium repräsentiert ein Stadion.

Team Ein Team ist eine Gruppe aus Advisors und Spielern, usw. Sie repräsentieren die Mannschaften, die gegeneinander spielen.

Tournament Ein Tournament ist ein Turnier - da mehrere Turniere in unsere Datenbank passen sollen ist sie nötig.

TournamentGroup Eine TournamentGroup repräsentiert eine Gruppe in der Gruppenphase (Gruppe A, B, C, ...) .

3 Packet: `dbb.project.entity.event`

Diese Entitäten implementieren die abstrakte Klasse `MatchEvent`.

MatchEndEvent Ein `MatchEndEvent` repräsentiert den Schlusspfiff in einem Spiel.

PlayerEvent Ein `PlayerEvent` ist ein im Spiel auftretendes Ereignis, das einen Spieler betrifft. Die Klasse ist *abstract* und wird von `CardEvent`, `GoalEvent`, `LineUpEvent` und `SubstitutionEvent` implementiert.

4 Packet: `dbb.project.entity.event.player`

Diese Klassen implementieren die abstrakte Klasse `PlayerEvent`.

CardEvent Ein `CardEvent` repräsentiert die Vergabe einer Karte an einen Spieler.

GoalEvent Ein `GoalEvent` repräsentiert ein (Eigen-)Tor.

LineUpEvent Ein `LineUpEvent` repräsentiert das Aufstellen eines Spielers in der Startelf.

SubstitutionEvent Ein `SubstitutionEvent` repräsentiert das Auswechseln bzw. Einwechseln eines Spielers.

5 Packet: `dbb.project.entity.permission`

Diese Entitäten werden zur Authentifizierung und Rechtevergabe benötigt.

Actor Ein `Actor` repräsentiert einen Akteur, der Daten in die Datenbank speichern, oder Daten aus ihr lesen will.

Permission `Permissions` verwalten den Zugang zu gewissen Ressourcen im System.

Resource Eine `Ressource` spiegelt ein Recht auf eine Entität wieder. Hat ein Akteur also die Berechtigung (`Permission`) für eine `Ressource`, so kann er diese je nach vergebener Berechtigung einsehen oder verwalten.

Role Akteuren können gewisse Rollen zugewiesen werden. Diese haben wieder je nach Einstellung Berechtigungen auf Ressourcen.

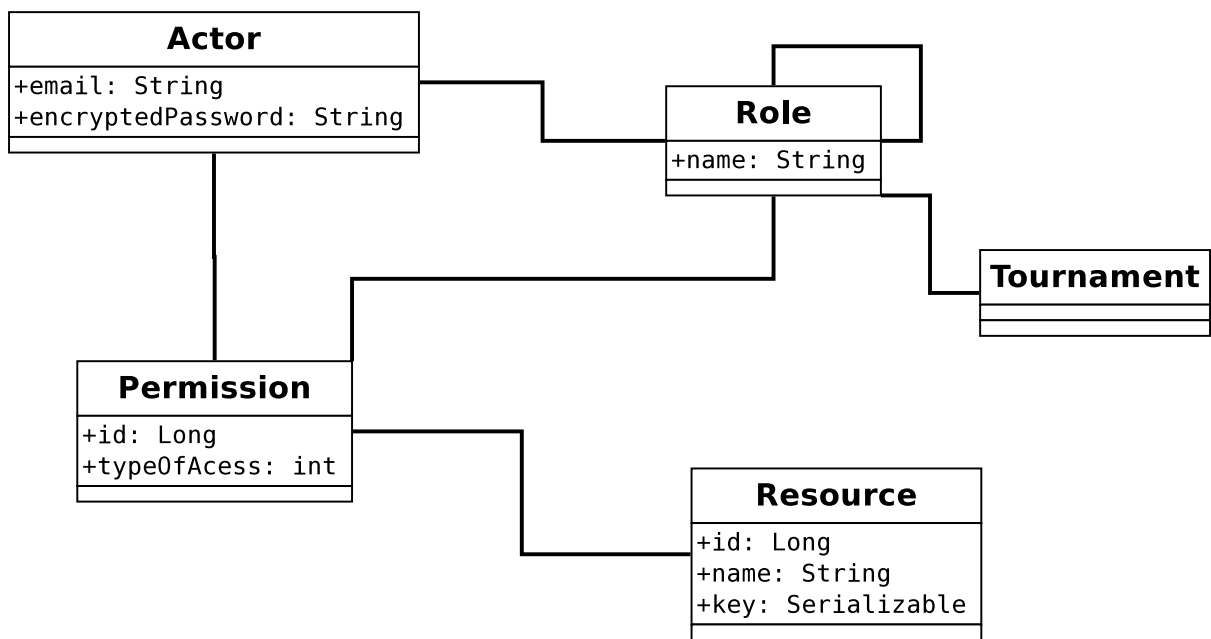


Figure 2: Objektmodell: Authentifizierung und Rechtevergabe