

# System Design

## Kickercup

*Version:*                1.0  
*Datum:*  
*Autoren:*            *Christoper Heid, Moritz Lugbauer, Jonas Jahns, Lucas Wierer, Johannes  
Schweer*

**Änderungsverzeichnis:**

Version	Datum	Autor(en)	Änderung

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>3</b>
<b>2</b>	<b>Überblick .....</b>	<b>4</b>
<b>3</b>	<b>Funktionales Modell .....</b>	<b>5</b>
3.1	Benutzerdaten-Management .....	5
3.2	Turnierdaten-Management .....	5
3.3	Spielerdaten-Management .....	5
3.4	Team-Making .....	6
3.5	Matchmaking.....	6
3.6	Continuous Integration.....	6
3.7	Ranglisten-Berechnung .....	7
3.8	PDF-Export .....	7
3.9	Skill Level-Berechnung .....	7
3.10	Share via Twitterintegration .....	7
3.11	Programm-Feedback .....	8
3.12	Share via Facebookintegration .....	8
3.13	Design Konfigurator .....	8
3.14	Spielerstatistiken .....	8
3.15	Benutzerauthentifizierung .....	9
<b>4</b>	<b>Dialoglandkarte.....</b>	<b>10</b>
4.1	Startdialoge .....	10
4.2	Verwaltungsdialoge .....	11
4.3	Dialog während des Spiels .....	13
4.4	Auswertungsdialo g .....	13
<b>5</b>	<b>IT Architektur .....</b>	<b>14</b>
5.1	Kontextabgrenzung.....	15
5.2	Bausteinsicht / Systemansicht .....	16
5.3	Klassendiagramm .....	17
<b>6</b>	<b>Schnittstellen und Integration .....</b>	<b>18</b>
6.1	Schnittstellenspezifikation.....	18
6.2	Schnittstellenformat .....	18
<b>7</b>	<b>Datenmodell .....</b>	<b>20</b>
7.1	Datenspeicherungskonzept .....	20
7.2	Physikalische Datenbankstruktur.....	20
7.3	Datenmodell.....	21
<b>8</b>	<b>Testspezifikation.....</b>	<b>21</b>
8.1	Teststrategie .....	21
8.2	Testfälle .....	21
<b>9</b>	<b>Projektmanagement .....</b>	<b>23</b>
9.1	Projektplan .....	23
9.2	Aufgabenverteilung .....	23
<b>10</b>	<b>Glossar .....</b>	<b>24</b>

# 1 Einleitung

## Ziel dieses Dokuments

Dieses Projekt hat zum Ziel eine Software zur Verwaltung von Tischkicker-Turnieren zu erstellen. In diesem Dokument werden die jeweiligen Use Cases und die dazugehörige Architektur spezifiziert.

## Referenzierte Dokumente

Folgende Bilder werden zwar in klein gezeigt, sind aber zur besseren Lesbarkeit separat beigefügt:

- Fachliches\_Klassenmodell\_Drawlo.jpg
- Fachliches\_Komponentendiagramm\_Drawlo.jpg
- Fachliches\_Datenmodell\_Drawlo.jpg
- Swagger\_File

## **2 Überblick**

Das „Kickercup“-System hat zum Ziel, Nicht-Fach-Anwendern das Erstellen und Austragen von Tischkicker-Turnieren zu erleichtern.

Nutzer können sich über den Browser anmelden und in einem Web-Portal Ihre Turniere konfigurieren. Über das System sollen Analysen über die Spielstärke der Spieler gemacht werden können. Nach dem Abschließen eines Turniers sollen diverse Statistiken angezeigt werden. Die Ergebnisse sollen auf Social-Media teilbar und über einen PDF-Export Druckbar sein. Das System soll von mehreren Nutzern gleichzeitig verwendbar sein und über einen Login und Logout verfügen. Die genauen Use-Cases wird im nachfolgenden Kapitel spezifiziert.

## 3 Funktionales Modell

### 3.1 Benutzerdaten-Management

<b>Use Case Nummer</b>	1
<b>Name</b>	Benutzerdaten-Management
<b>Ziel</b>	Benutzer konfigurieren
<b>Eingangsbedingung / Anstoß</b>	GUI-Auswahl „Konto erstellen“ / manuell
<b>Erfolgreiche Ausgangsbedingung</b>	Benutzer wird in Konto angemeldet
<b>Akteure</b>	Nicht angemeldeter Benutzer
<b>Benutzerinteraktion</b>	<ol style="list-style-type: none"> <li>1. Eingabe über GUI: <ol style="list-style-type: none"> <li>a. Benutzername</li> <li>b. Geschlecht</li> <li>c. Passwort</li> <li>d. Email</li> </ol> </li> <li>2. GUI Auswahl „Speichern“</li> </ol>

### 3.2 Turnierdaten-Management

<b>Use Case Nummer</b>	2
<b>Name</b>	Turnierdaten-Management
<b>Ziel</b>	Turnier konfigurieren
<b>Eingangsbedingung / Anstoß</b>	GUI-Auswahl „Turnier erstellen“ / manuell
<b>Erfolgreiche Ausgangsbedingung</b>	Benutzer startet Spiel
<b>Akteure</b>	Benutzer
<b>Benutzerinteraktion</b>	<ol style="list-style-type: none"> <li>3. Eingabe über GUI: <ol style="list-style-type: none"> <li>a. Anzahl der Tische</li> <li>b. Wie viele Siege man zum Matchgewinn braucht.</li> <li>c. Wie viele Tore man zum Gewinnen braucht</li> <li>d. Anzahl der Teams</li> <li>e. Gewertetes oder ungewertetes Turnier</li> </ol> </li> <li>4. Spieler hinzufügen / erstellen</li> </ol>

Gestartete Turniere werden nicht in der Datenbank, sondern nur im Session Storage persistiert. Nur die Konfiguration eines Turniers wird in der Datenbank gespeichert.

### 3.3 Spielerdaten-Management

<b>Use Case Nummer</b>	3
<b>Name</b>	Spielerdaten-Management
<b>Ziel</b>	Teilnehmer anlegen
<b>Eingangsbedingung / Anstoß</b>	GUI-Auswahl „Teilnehmer erstellen“ / manuell
<b>Erfolgreiche Ausgangsbedingung</b>	Spieler ist in Datenbank gespeichert
<b>Akteure</b>	Benutzer
<b>Benutzerinteraktion</b>	<ol style="list-style-type: none"> <li>1. Name, Alter, Geschlecht eingeben</li> <li>2. Selbsteinschätzung des Skills</li> </ol>

### 3.4 Team-Making

<b>Use Case Nummer</b>	4
<b>Name</b>	Team-Making
<b>Ziel</b>	Spieler werden Teams zugewiesen
<b>Eingangsbedingung / Anstoß</b>	Turnier wird gestartet / automatisch
<b>Erfolgreiche Ausgangsbedingung</b>	Alle Teams haben zwei Spieler
<b>Akteure</b>	Anwendungsintern
<b>Benutzerinteraktion</b>	<p>Gewertetes Turnier:</p> <ol style="list-style-type: none"> <li>1. Spieler werden anhand ihres Skill Levels in Teams aufgeteilt. -&gt; Teams sollen annähernd gleiche Skill Level Summe haben</li> </ol> <p>Ungewertetes Turnier:</p> <ol style="list-style-type: none"> <li>1. Spieler werden zufällig aufgeteilt.</li> </ol>

### 3.5 Matchmaking

<b>Use Case Nummer</b>	5
<b>Name</b>	Match-Making
<b>Ziel</b>	Match wird erstellt
<b>Eingangsbedingung / Anstoß</b>	Zwei Teams haben kein Match / automatisch
<b>Erfolgreiche Ausgangsbedingung</b>	Match ist erstellt und hat zwei Teams zugewiesen bekommen
<b>Akteure</b>	Anwendungsintern
<b>Benutzerinteraktion</b>	<ol style="list-style-type: none"> <li>1. Gegebenenfalls fliegt ein Team raus (je nach vorherigem Ergebnis).</li> <li>2. Match wird erstellt.</li> <li>3. Anhand von Spielmodus werden zwei Teams dem Match zugewiesen.</li> </ol>

### 3.6 Continuous Integration

<b>Use Case Nummer</b>	6
<b>Name</b>	Continuous Integration
<b>Ziel</b>	CI Pipeline bereitstellen
<b>Eingangsbedingung / Anstoß</b>	Git Push
<b>Erfolgreiche Ausgangsbedingung</b>	CI Pipeline hat Artefakt erstellt und sind downloadbar
<b>Akteure</b>	Entwickler
<b>Benutzerinteraktion</b>	<ol style="list-style-type: none"> <li>1. Entwickler pusht Code</li> <li>2. Unit Tests werden ausgeführt</li> <li>3. Front- und Backend werden gebaut</li> </ol>

### 3.7 Ranglisten-Berechnung

<b>Use Case Nummer</b>	7
<b>Name</b>	Ranglisten-Berechnung
<b>Ziel</b>	Rangliste berechnen
<b>Eingangsbedingung / Anstoß</b>	Finale beendet / automatisch
<b>Erfolgreiche Ausgangsbedingung</b>	Rangliste ist berechnet
<b>Akteure</b>	Anwendungsintern
<b>Benutzerinteraktion</b>	1. Turnierbaum wird in Rangliste umgewandelt und auf GUI angezeigt

### 3.8 PDF-Export

<b>Use Case Nummer</b>	8
<b>Name</b>	PDF Export
<b>Ziel</b>	Ranglisten und Urkunden als PDF exportieren
<b>Eingangsbedingung / Anstoß</b>	Mindestens ein Turnier abgeschlossen / manuell
<b>Erfolgreiche Ausgangsbedingung</b>	PDF Datei ausgegeben
<b>Akteure</b>	Benutzer
<b>Benutzerinteraktion</b>	1. Turnier auswählen a. Rangliste exportieren 2. Teilnehmer auswählen a. Urkunde exportieren

### 3.9 Skill Level-Berechnung

<b>Use Case Nummer</b>	9
<b>Name</b>	Skill Level-Berechnung
<b>Ziel</b>	Individuelle Skill Level für Teilnehmer anhand des Matchergebnis
<b>Eingangsbedingung / Anstoß</b>	Matchende / automatisch
<b>Erfolgreiche Ausgangsbedingung</b>	Das neue Skill Level wird für den Teilnehmer in der Datenbank beendet
<b>Akteure</b>	Anwendungsintern
<b>Benutzerinteraktion</b>	1. Skill Level berechnen 2. Skill Level abspeichern

### 3.10 Share via Twitterintegration

<b>Use Case Nummer</b>	10
<b>Name</b>	Share via Twitterintegration
<b>Ziel</b>	Turnierergebnisse auf Twitter posten
<b>Eingangsbedingung / Anstoß</b>	Mindestens ein Turnier abgeschlossen / manuell
<b>Erfolgreiche Ausgangsbedingung</b>	Erfolgreiches Posting auf Twitter
<b>Akteure</b>	Benutzer
<b>Benutzerinteraktion</b>	1. Turnier auswählen 2. Twitter Account verbinden 3. Aktion „Teilen“ ausführen

### 3.11 Programm-Feedback

Use Case Nummer	11
Name	Programm-Feedback
Ziel	Benutzer soll Feedback hinterlassen können
Eingangsbedingung / Anstoß	Mindestens ein Turnier abgeschlossen / manuell
Erfolgreiche Ausgangsbedingung	Erfolgreiches Posting auf Facebook oder Twitter
Akteure	Benutzer
Benutzerinteraktion	<ol style="list-style-type: none"><li>1. Turnier auswählen</li><li>2. Facebook/Twitter Account verbinden</li><li>3. Aktion „Teilen“ ausführen</li></ol>

### 3.12 Share via Facebookintegration

Use Case Nummer	12
Name	Share via Facebookintegration
Ziel	Turnierergebnisse auf Facebook posten
Eingangsbedingung / Anstoß	Mindestens ein Turnier abgeschlossen / manuell
Erfolgreiche Ausgangsbedingung	Erfolgreiches Posting auf Facebook
Akteure	Benutzer
Benutzerinteraktion	<ol style="list-style-type: none"><li>1. Turnier auswählen</li><li>2. Facebook Account verbinden</li><li>3. Aktion „Teilen“ ausführen</li></ol>

### 3.13 Design Konfigurator

Use Case Nummer	13
Name	Design Konfigurator
Ziel	Design des Spielfelds konfigurieren
Eingangsbedingung / Anstoß	keine
Erfolgreiche Ausgangsbedingung	Erfolgreich Design angepasst
Akteure	Benutzer
Benutzerinteraktion	<ol style="list-style-type: none"><li>1. Konfiguration öffnen</li><li>2. Farbe auswählen</li></ol>

### 3.14 Spielerstatistiken

Use Case Nummer	14
Name	Spielerstatistiken
Ziel	Charts zu Nutzerdaten anzeigen
Eingangsbedingung / Anstoß	Mindestens ein Turnier abgeschlossen / manuell
Erfolgreiche Ausgangsbedingung	Charts werden angezeigt
Akteure	Benutzer
Benutzerinteraktion	<ol style="list-style-type: none"><li>1. Konfiguration öffnen</li><li>2. Farbe auswählen</li></ol>



### 3.15 Benutzerauthentifizierung

<b>Use Case Nummer</b>	15
<b>Name</b>	Benutzerauthentifizierung
<b>Ziel</b>	Authentifizierung des Nutzers
<b>Eingangsbedingung / Anstoß</b>	Starten der Anwendung / manuell
<b>Erfolgreiche Ausgangsbedingung</b>	Benutzer wird in Konto angemeldet Akteure: Nicht angemeldeter Benutzer
<b>Akteure</b>	Benutzer
<b>Benutzerinteraktion</b>	<ol style="list-style-type: none"><li>1. Eingabe über GUI:<ol style="list-style-type: none"><li>a. Benutzername</li><li>b. Passwort</li></ol></li><li>2. GUI Auswahl „Anmelden“</li></ol>

## 4 Dialoglandkarte

### 4.1 Startdialoge

In diesem Dialog kann der Benutzer sich anmelden oder, falls er dies noch nicht getan hat, ein Konto erstellen. Bei der Erstellung eines Benutzerkontos muss ein Benutzername, Vor- und Nachname, Geschlecht und ein Passwort eingegeben werden (schreibender Zugriff).



1.0 Start | KickerCup

**KICKER CUP**

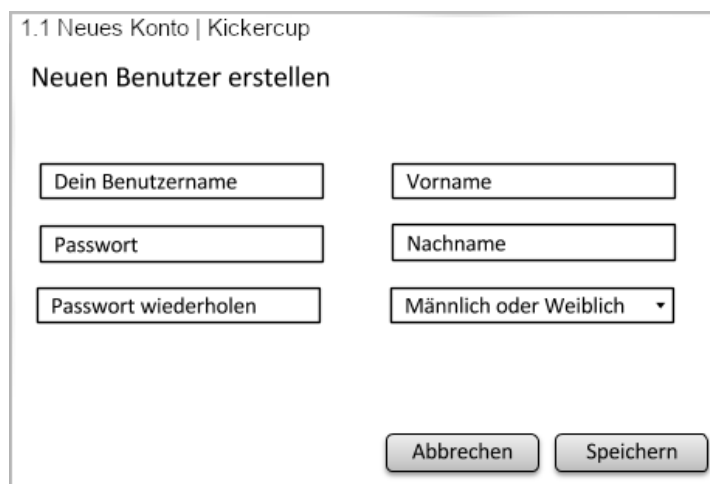
Benutzername

Passwort

2.0 Anmelden

Noch nicht registriert? 1.1 Konto erstellen

Abbildung 1: Startbildschirm und Login



1.1 Neues Konto | KickerCup

Neuen Benutzer erstellen

Dein Benutzername

Vorname

Passwort

Nachname

Passwort wiederholen

Männlich oder Weiblich ▼

Abbrechen Speichern

Abbildung 2: Benutzererstellung

## 4.2 Verwaltungsdialoge

Hier erfolgt im Allgemeinen die Verwaltung der Turniere. Dazu gehören die Turniererstellung, das Hinzufügen und Verwalten von Teilnehmern, das Anzeigen der Turnierhistorie, die grundsätzlichen Benutzereinstellungen und die Möglichkeit zu Datenexporten. Es ist je nach Anwendungsfall lesender oder schreibender Zugriff erforderlich. In diesem Dialog werden die Anwendungsfälle 1, 2 und 7 ausgelöst.

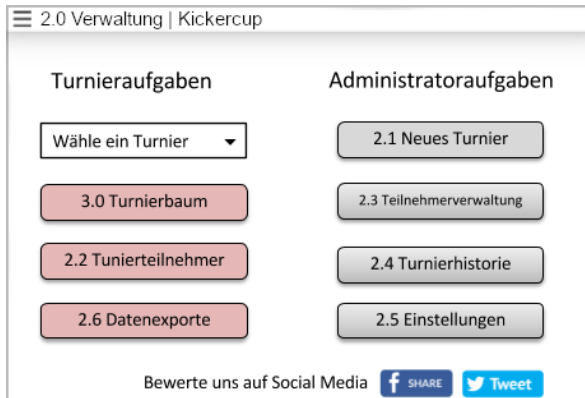


Abbildung 3: Turnierverwaltung

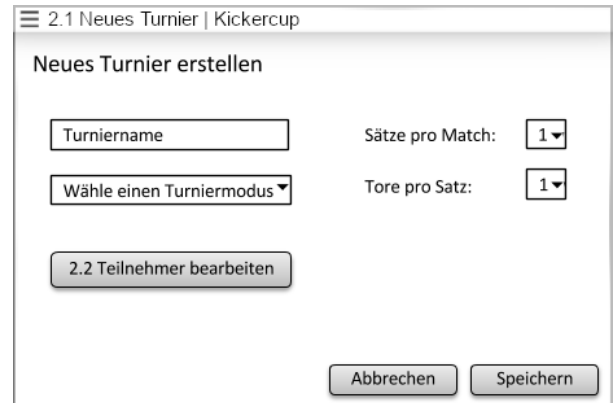


Abbildung 4: Turniererstellung

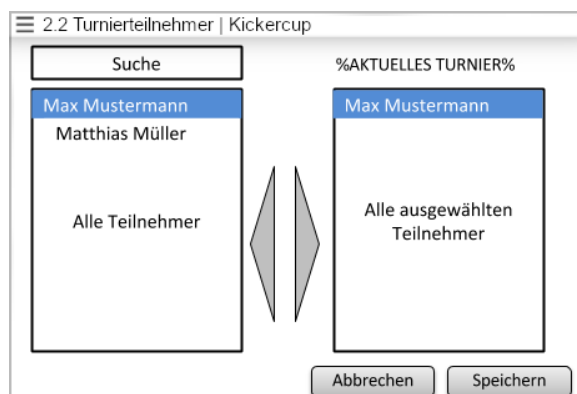


Abbildung 5: Turnierteilnehmer verwalten

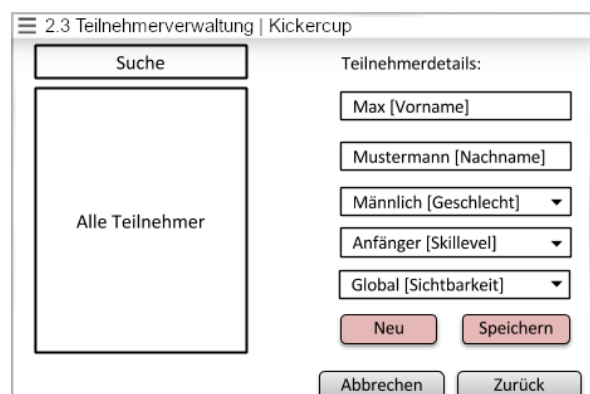


Abbildung 6: Teilnehmer erstellen

2.4 Turnierhistorie | Kickercup

Sonntagscup  
Übungsturnier  
  
Alle Turniernamen

Turniername

Sätze pro Match: 1

Tore pro Satz: 1

Anzahl Teilnehmer: XX

Verbleibende Matches 3/8

3.0 Turnierbaum

Abbrechen Speichern

2.5 Einstellungen | Kickercup

%AKTUELLER MANDANT%

Dein Benutzername

Passwort

Vorname

Passwort wiederholen

Nachname

Passwort setzen

Männlich oder Weiblich

Abbrechen Speichern

Abbildung 7: Turnierhistorie    Abbildung 8: Einstellungen Benutzer

2.6 Datenexporte | Kickercup

Suche

Sonntagscup  
Übungsturnier  
  
Alle Turniere

Teilnehmerliste exportieren

→

Suche

Max Mustermann  
Matthias Müller  
  
Alle Teilnehmer

Urkunde exportieren

Abbrechen Speichern

Abbildung 9: Datenexporte

### 4.3 Dialog während des Spiels

In diesem Dialog können die Ergebnisse der Sätze eingetragen und gespeichert werden, demnach ist schreibender Zugriff erforderlich. Abgedeckt wird damit der Anwendungsfall 5.

Abbildung 10: Ergebnisse eintragen

### 4.4 Auswertungsdialog

Durch diesen Dialog wird der Anwendungsfall 7 ausgelöst. Erforderlich ist lesender Zugriff, sodass der Turnierbaum angezeigt werden kann.

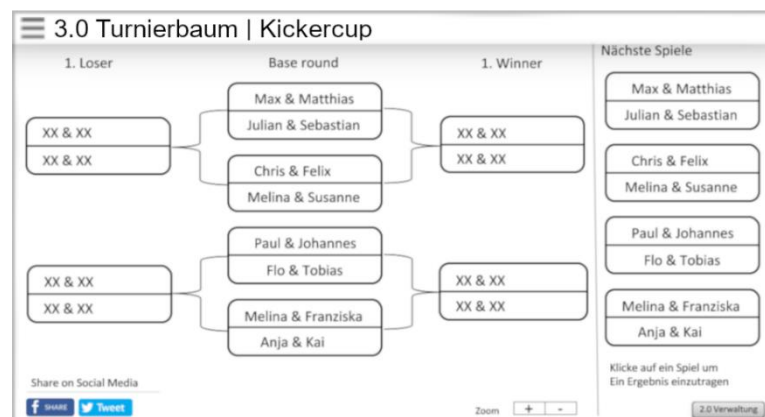


Abbildung 11: Turnierbaum

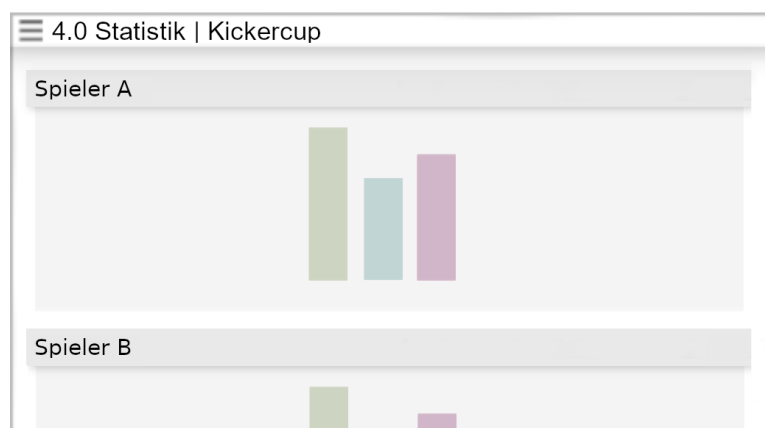
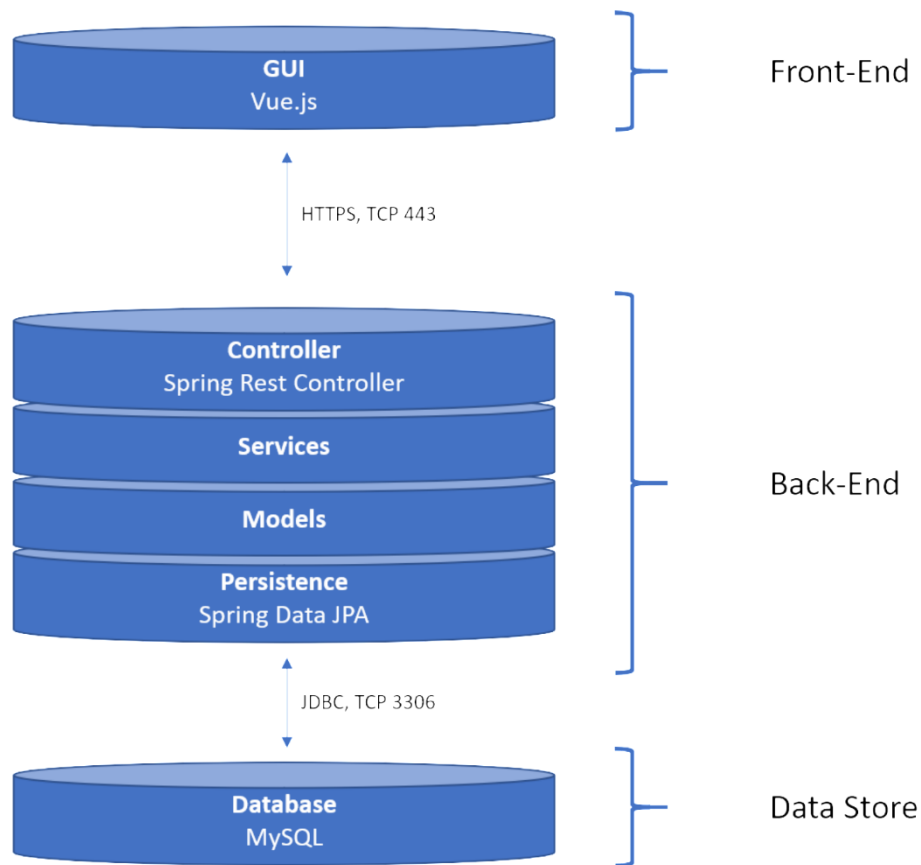


Abbildung 12: Statistiken

## 5 IT Architektur

Dieses Kapitel spezifiziert die angestrebte Architektur des Projektes.

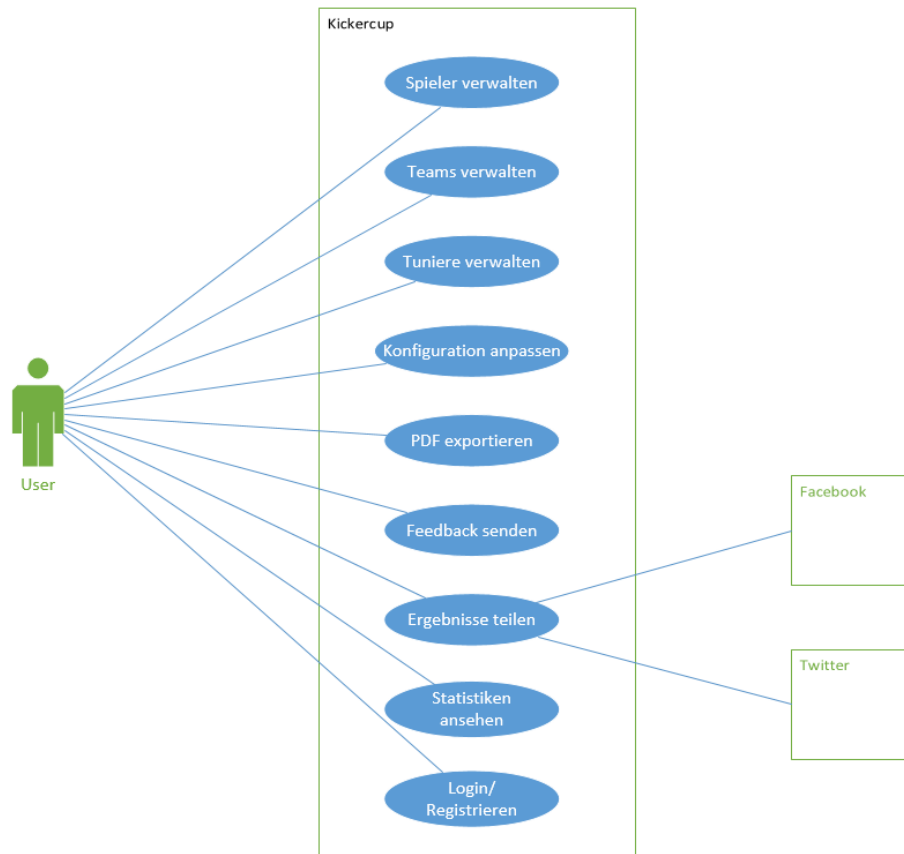


Das Projekt besteht aus einer klassischen drei Schichtenarchitektur

- Front-End
- Back-End
- Database

## 5.1 Kontextabgrenzung

In der Kontextabgrenzung werden zunächst die Use Cases sowie der dazugehörige Anwender beschrieben. Zusätzlich werden die eingebundenen Fremdsysteme angegeben.



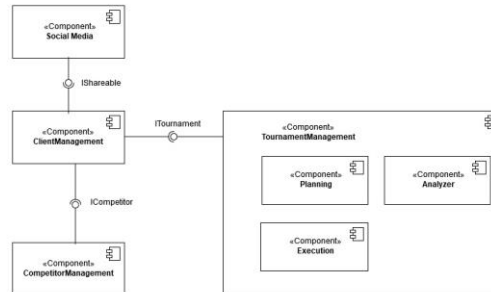
## 5.2 Bausteinsicht / Systemansicht

### Systemansicht

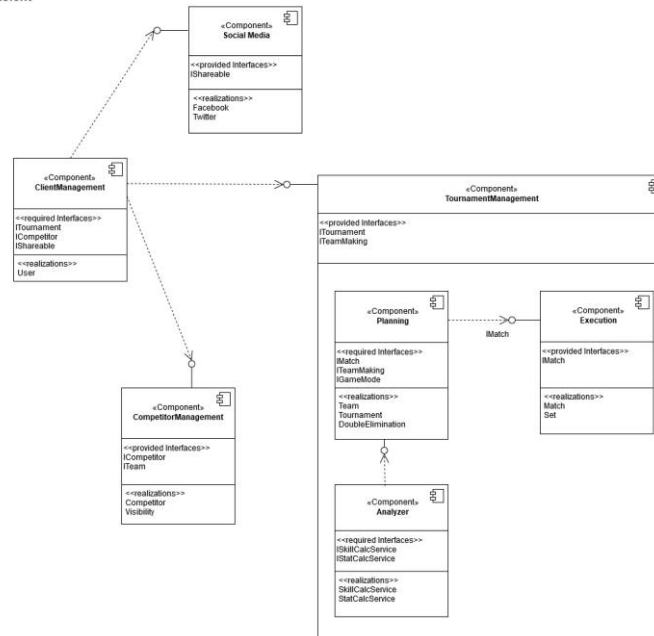
Systeme werden hier in Form von Komponenten dargestellt die nur über ein Interface voneinander abhängen.

Alle Komponenten werden nach dem Entity-Control-Boundary Pattern umgesetzt.

### Außersicht



### Innensicht



Größere Ansicht siehe *Fachliches\_Klassenmodell\_DrawIo.jpg*



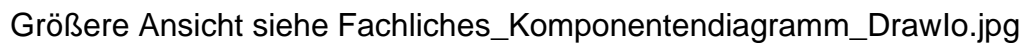
### Einführung:

Ein Client kann Tournaments und Competitors erstellen.

Der Client die Möglichkeit Feedback und Tournaments auf Facebook und Twitter zu teilen.

Schlüssendlich ist die Hauptaufgabe des Clients, Tournaments zu erstellen und diesen Competitors zuzuweisen. Das Tournament und der Competitor implementieren außerdem das Interface `IScoreExportable`. Somit können Umkunden und Ranglisten als CSV exportiert werden.

Bei der Tournament-Erstellung kann ein `IGameMode` ausgewählt werden, der überpruft, ob die Anzahl der Teams mit dem `GameMode` kompatibel ist. Im Verlaufe des Tournaments werden über den `IGameMode` neue Matches erstellt, die wiederum Sets erstellen. Nach jedem Match wird das `SkillLevel` der beteiligten Competitors berechnet.



## 6 Schnittstellen und Integration

### 6.1 Schnittstellenspezifikation

Als externe Schnittstellen kommen Facebook und Twitter zum Einsatz. Folgende Endpunkte wollen wir verwenden

#### TWITTER

/tweet/post  
/tweet/get

#### FACEBOOK

/timeline/post  
/timeline/get

Interne Schnittstellen (REST-Endpunkte).

Eine ausführliche Schnittstellendokumentation befindet sich im angehängten Swagger.json File.

REST-API for KICKERCUP REST API

#### player-controller Player Controller

**GET** /api/v1/players getAllPlayers

**POST** /api/v1/players createPlayer

**GET** /api/v1/players/{id} getPlayerById

**PUT** /api/v1/players/{id} updatePlayer

**DELETE** /api/v1/players/{id} deletePlayer

#### result-controller Result Controller

#### tournament-controller Tournament Controller

#### user-controller User Controller

### 6.2 Schnittstellenformat

Als Schnittstellenformat kommt der Content-Type "application/json" zum Einsatz. Die Serialisierung wird über die Standard-Komponenten des Springboot 2 Frameworks realisiert. Ein Beispiel POST Request sieht folgendermaßen aus:

POST

/api/v1/tournaments createTournament

Parameters

Try it out

Name	Description
<b>Tournament</b> * required (body)	<div>Tournament</div> <div>Example Value   Model</div> <div><pre>{   "amountGoals": 0,   "amountSets": 0,   "gamemode": "string",   "id": 0,   "isFinished": true,   "isRanked": true,   "name": "string",   "userId": 0 }</pre></div> <div>Parameter content type</div> <div>application/json</div>

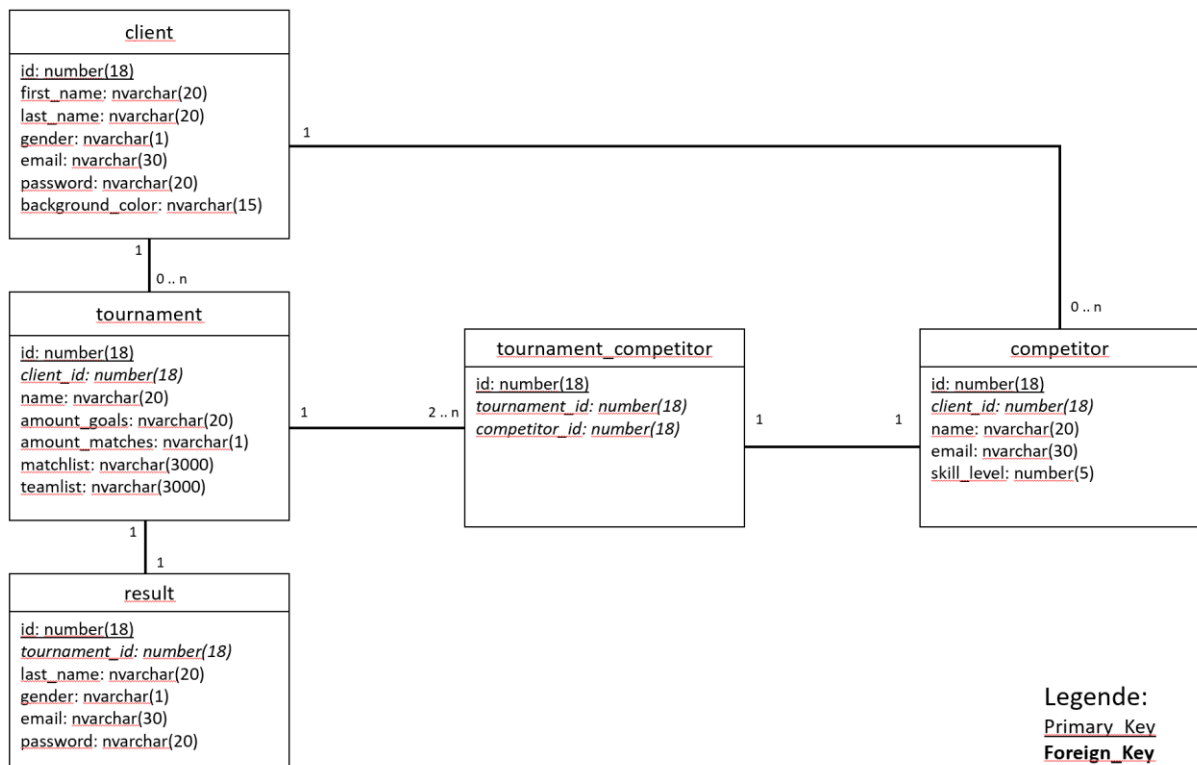
## 7 Datenmodell

### 7.1 Datenspeicherungskonzept

Die Datenspeicherung findet ausschließlich in der DataAccess-Schicht, in der jede Entität ein eigenes Repository besitzt. Mithilfe von JPA werden hier die Datenbank Transaktionen auf eine MySQL Datenbank durchgeführt. Unsere DataAccess-Schicht entspricht hierbei der Entity Schicht des BCE-Patterns.

### 7.2 Physikalische Datenbankstruktur

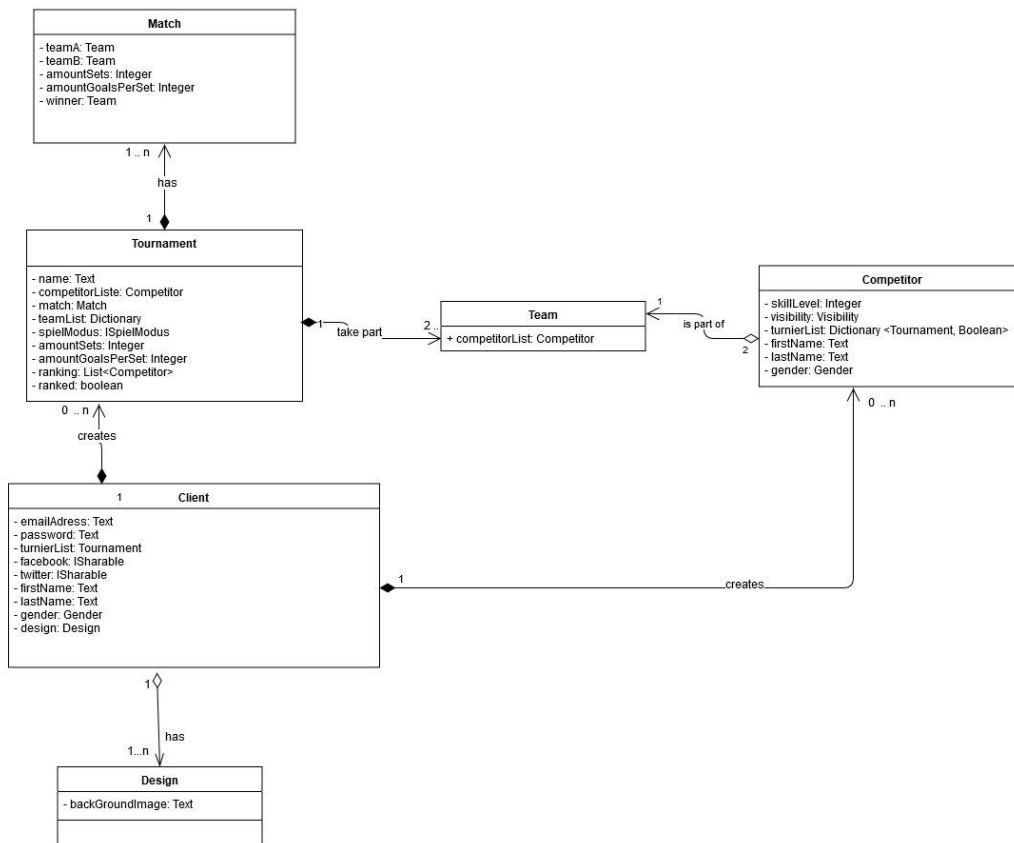
Die folgenden Tabellen entsprechen den Data Entitäts, die nur im Backend und Datenbank verwendet werden.



## 7.3 Datenmodell

Das folgende Modell ist das Modell für die Data Transfer Objects (DTO). Das sind die Objekte die zwischen Frontend und Backend ausgetauscht werden.

### KICKERCUP



Größere Ansicht siehe Fachliches\_Datenmodell\_Drawlo.jpg

## 8 Testspezifikation

### 8.1 Teststrategie

Um die durchgehende Funktionsfähigkeit von Komponenten und Funktionen zu gewährleisten werden Unit Tests geschrieben die automatisiert nach jeder Funktionserweiterung gestartet werden. Um Use Cases auf die Vollständige Funktionalität zu überprüfen werden komplexe End-to-End Tests in Protractor erstellt.

### 8.2 Testfälle

End-to-End Tests:

NR.	Testfall Beschreibung	Testschritte
1	Nutzer loggt sich ein und erstellt ein neues Turnier mit vier bereits existieren-	<ol style="list-style-type: none"> <li>1. Benutzerprofil und vier Spielerprofile in der Datenbank erstellen</li> <li>2. Mit angelegtem Benutzeraccount einloggen</li> </ol>

	den Teilnehmern.	<ol style="list-style-type: none"> <li>3. Neues Turnier mit vorhandenen Spielerprofilen erstellen und notwendigen Parametern erstellen</li> <li>4. Benutzerprofil, Spielerprofil und Turnier löschen</li> </ol>
2	Nutzer loggt sich ein, ruft vorhandenes Turnier auf und trägt Ergebnisse für Spielsätze ein bis das Turnier beendet ist.	<ol style="list-style-type: none"> <li>1. Benutzerprofil, Spielerprofile und Turnier erstellen</li> <li>2. Mit angelegtem Benutzeraccount einloggen und vorhandenes Turnier aufrufen</li> <li>3. Turnierverlauf durchspielen indem Ergebnis für die jeweiligen Runden eingetragen werden bis das Turnier zu Ende ist</li> <li>4. Benutzerprofil, Spielerprofil und Turnier löschen</li> </ol>

### Unit Tests:

NR.	Testfall	Testschritte	Erwartetes Ergebnis
1	Anlegen eines Neuen Benutzers	<ol style="list-style-type: none"> <li>1. Erstellen der Benutzer Parameter</li> <li>2. Aufrufen verschiedener Validierung Methoden</li> <li>3. Speichern des Nutzers mit anschließend Überprüfung</li> <li>4. Löschen des Angelegten Nutzers</li> </ol>	Validierung der einzelnen Parameter und anschließende Speicherung der User in der Datenbank
2	Anlegen eines neuen Turniers	<ol style="list-style-type: none"> <li>1. Benutzerprofil, Turniermodus und vier Spielerprofile müssen erstellt werden</li> <li>2. Restlichen Parameter zusammen mit Spielerprofilen an initiale Methode übergeben</li> <li>3. Überprüfen ob Turnier korrekt gespeichert wurde und anschließend Turnier und Daten aus Punkt löschen</li> </ol>	Validieren der einzelnen Parameter und Speichern der Turnierdaten
3	Team Verteilung bei einem gewerteten Turnier	<ol style="list-style-type: none"> <li>1. Acht Spielerprofile mit unterschiedlichen Skill Levels erstellen</li> <li>2. Team Making Methode aufrufen und überprüfen ob ausgeglichene Teams erstellt werden</li> <li>3. Spielerprofile und Teams löschen</li> </ol>	Es werden ausgeglichene Teams anhand des Skill-Levels erstellt
4	Überprüfen der Matchlogik	<ol style="list-style-type: none"> <li>1. Spielerprofile, Turnier und Teams müssen erstellt werden</li> <li>2. Beide Spiellogiken überprüfen indem kompletter Spielverlauf simuliert wird</li> <li>3. Ergebnisse und Daten aus Punkt eins löschen</li> </ol>	Korrekte Implementierung der ausgewählten Spiellogik
5	Skill-Level Berechnung	<ol style="list-style-type: none"> <li>1. Vier Spielerprofile erstellen und Matchergebnis erstellen</li> <li>2. Anhand des Machergebnis Skill-Level Algorithmus aufrufen</li> </ol>	Algorithmus berechnet auf Basis eines Match Ergebnis den korrekten Skill-Level

		3. Überprüfen ob Skill-Level richtig berechnet und gespeichert wurden 4. Spielerprofile und Matchergebnis löschen	der einzelnen Spieler
6	Berechnung Spielerstatistik	1. Abschluss eines Turniers simulieren 2. Methoden für Chartberechnung aufrufen mit anschließender Überprüfung 3. Simulationsdaten des Turniers löschen	Daten für Charts werden korrekt berechnet und ausgegeben
7	Aktive Sozial-Media Connection	1. Sozial-Media Api ansprechen und überprüfen ob positive Rückmeldung erhalten wird	Status Report ob Facebook und Twitter Schnittstellen ansprechbar sind
8	Ranglistenberechnung	1. Vier Spielerprofile erstellen und Matchergebnis erstellen 2. Methode für Ranglisten Berechnung ausführen und überprüfen ob Ergebnisse richtig ausgegeben werden 3. Spielerprofile und Matchergebnis löschen	Rangliste wird auf Basis eines Turnierdurchlaufs korrekt berechnet

## 9 Projektmanagement

### 9.1 Projektplan

Um den Fortschrittserfolg messbarer zu gestalten, verwenden wir verschiedene Milestones, die mithilfe der Sprints erreicht werden sollen.

Datum	Name
15.11.2019	Verbindung FE, BE, DB -> CRUD ist für alle Entitys möglich
20.11.2019	CI Pipeline
31.11.2019	Turniere können ausgeführt werden
10.12.2019	Benutzerauthentifizierung

### 9.2 Aufgabenverteilung

Das Projektmanagement wird mit SCRUM umgesetzt. Wir setzen hier auf zwei-wöchige Sprints mit folgender Rollenverteilung:

Name	Rolle
Christopher Heid	Entwickler, Product Owner
Moritz Luginbauer	Entwickler
Lucas Wierer	Entwickler
Jonas Jahns	Entwickler
Johannes Schweer	Entwickler, Scrum Master

Zusätzlich sind die Use Cases auf die Projektteilnehmer aufgeteilt. Alle Use Cases sind als Full-Stack Aufgabe gedacht.

Use Case		
K01 - Benutzerverwaltung	groß	Moritz
K02 - Turnierdatenmanagement	klein	Moritz
K03 - Spielerdatenmanagement	klein	Lucas

K04 - Team-Making	mittel	Moritz
K05 - Match-Making	groß	Johannes
K06 - Continuous Integration	groß	Chris
K07 - Ranglistenberechnung	mittel	Jonas
K08 - PDF-Export	mittel	Jonas
K09 - Skilllevel	groß	Lucas
K10 - Share via Twitter Integration	mittel	Johannes
K11 - Programm Feedback	mittel	Jonas
K12 - Share via Facebook Integration	klein	Chris
K13 - Design Konfigurator	mittel	Lucas
K14 - Spielerstatistiken	mittel	Johannes
K15 - Benutzer Authentifizierung	mittel	Chris

## 10 Glossar

BCE: Boundary-Controll-Entity Pattern

BE: Backend

FE: Frontend