

# Technische Dokumentation Basket Rolling Software

## 1. Projektbeschreibung

Dieses Projekt implementiert ein umfassendes Verwaltungssystem für den Basketballverein "*Basket Rolling*".

Die Software ermöglicht die Verwaltung von **Spielern, Trainern, Teams intern/extern, Spielen, Hallen, Trainingsterminen, Elternkontakten, Mitgliedsbeiträgen** und **Spielerstatistiken**.

Über eine grafische Benutzeroberfläche können Benutzer Daten eingeben, speichern und verwalten. Die Daten werden in einer Datenbank abgelegt und stehen somit zentralisiert und konsistent zur Verfügung.

Ein zentrales Merkmal der Anwendung ist das **rollenbasierte Zugriffssystem**:

- **Administratoren** haben volle Rechte, können Datensätze anlegen, bearbeiten, löschen und Benutzer verwalten.
- **Normale Benutzer (User)** haben eingeschränkte Rechte, beispielsweise nur Leserechte oder die Möglichkeit, bestimmte Informationen einzusehen, ohne Änderungen vornehmen zu können.

Dieses Rollenmodell sorgt für **Datensicherheit**, klare **Verantwortlichkeiten** und schützt die Anwendung vor unbefugten Änderungen.

Durch die Einführung dieser Software wird die Verwaltung im Verein effizienter, transparenter und weniger fehleranfällig. Hauptnutzer sind Vereinsadministratoren und Trainer, die mit wenigen Klicks alle relevanten Daten erfassen und abrufen können.

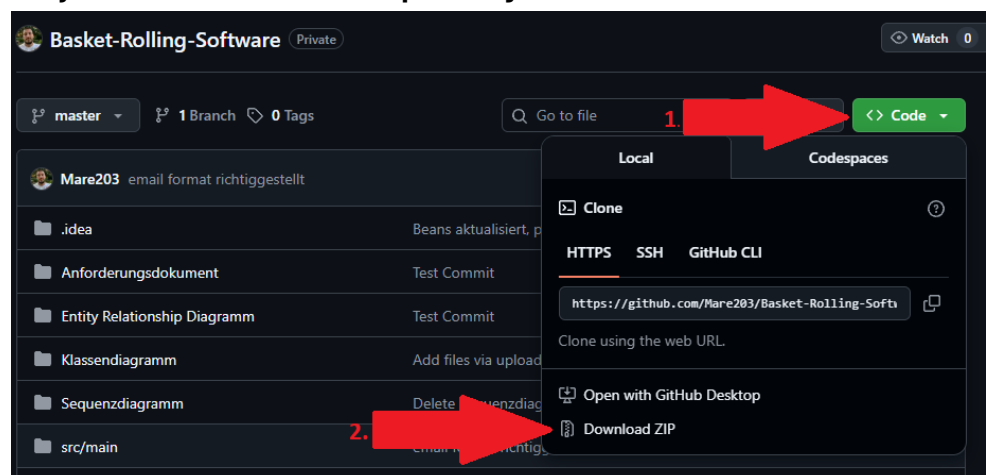
## 2. Systemanforderungen

- **Betriebssystem**  
Windows 10/11, Linux oder MacOS (plattformunabhängig, da Java-basiert)
- **Java-Version**  
Java Development Kit (JDK) 21 oder höher

- **Datenbank**
  - PostgreSQL 15.x (getestet mit JDBC-Treiber **org.postgresql:postgresql:42.7.6**)
  - DB-Name: *basketrolling*
  - Konfiguration über *persistence.xml*
- **Frameworks / Libraries**
  - **JavaFX 21.0.7** (GUI – Controls & FXML)
  - **Hibernate Core 7.0.0.Final** (ORM für Datenbankzugriff)
  - **Jakarta Persistence API 3.2.0** (JPA-Schnittstelle)
  - **Jakarta CDI 4.1.0** (Dependency Injection API)
  - **Password4j 1.8.2** (Passwort-Hashing für sichere Authentifizierung)
  - **Lombok 1.18.38** (Code-Generierung, Getter/Setter, Builder usw.)
  - **Maven** (Build & Dependency Management)
  - **javafx-maven-plugin** (Starten der Anwendung über Maven)
- **Hardwareanforderungen**
  - Bildschirmauflösung: 1920×1080, empfohlen für optimale GUI-Darstellung
- **Benutzeranforderungen**
  - Admin-Zugang (muss initial in der Datenbank angelegt werden)
  - Danach können Admins User im System erstellen

### 3. Installation & Setup

- **Voraussetzungen**
  - Installiertes **Java Development Kit (JDK) 21** oder höher
  - Installiertes **PostgreSQL** (z. B. Version 15.x)
  - Installiertes **Maven** (für Build und Dependency Management)
  - Entwicklungsumgebung: IntelliJ IDEA oder NetBeans
- **Schritt für Schritt Anleitung**
  1. **Projekt aus dem GitHub Repository klonen oder herunterladen**



## 2. Abhängigkeiten installieren

- Das Projekt verwendet Maven,  
alle benötigten Libraries (JavaFX, Hibernate, JDBC-Treiber,  
Password4j, Lombok usw.) werden automatisch heruntergeladen.

## 3. Datenbank konfigurieren

- Neue Datenbank *basketrolling* anlegen
- Einen Benutzer namens „mare“ als SuperUser erstellen
- Die Verbindungskonfiguration ist in der Datei  
*src/main/resources/META-INF/persistence.xml* hinterlegt

***jdbc:postgresql://localhost:5432/basketrolling***

***Benutzer: mare***

***Passwort: rolling***

## 4. Anwendung starten

1. Die heruntergeladene .zip Datei entpacken
2. Den Ordner „**Basket-Rolling-Software-master**“ in der bevorzugten IDE (z. B. IntelliJ IDEA oder NetBeans) öffnen.

### 3. **Erste Ausführung (einmalig erforderlich):**

*RollingDemo.java* zum Anlegen eines Standard Admin Users.

**Pfad:** *src\main\java\org\basketrolling\RollingDemo.java*

- **Benutzername:** admin

- **Passwort:** Admin

4. Nach dem Anlegen des ersten Administrators kann die Anwendung über die Klasse *LoginMenu.java* gestartet werden.

**Pfad:** *src\main\java\org\basketrolling\gui\menues\LoginMenu.java*

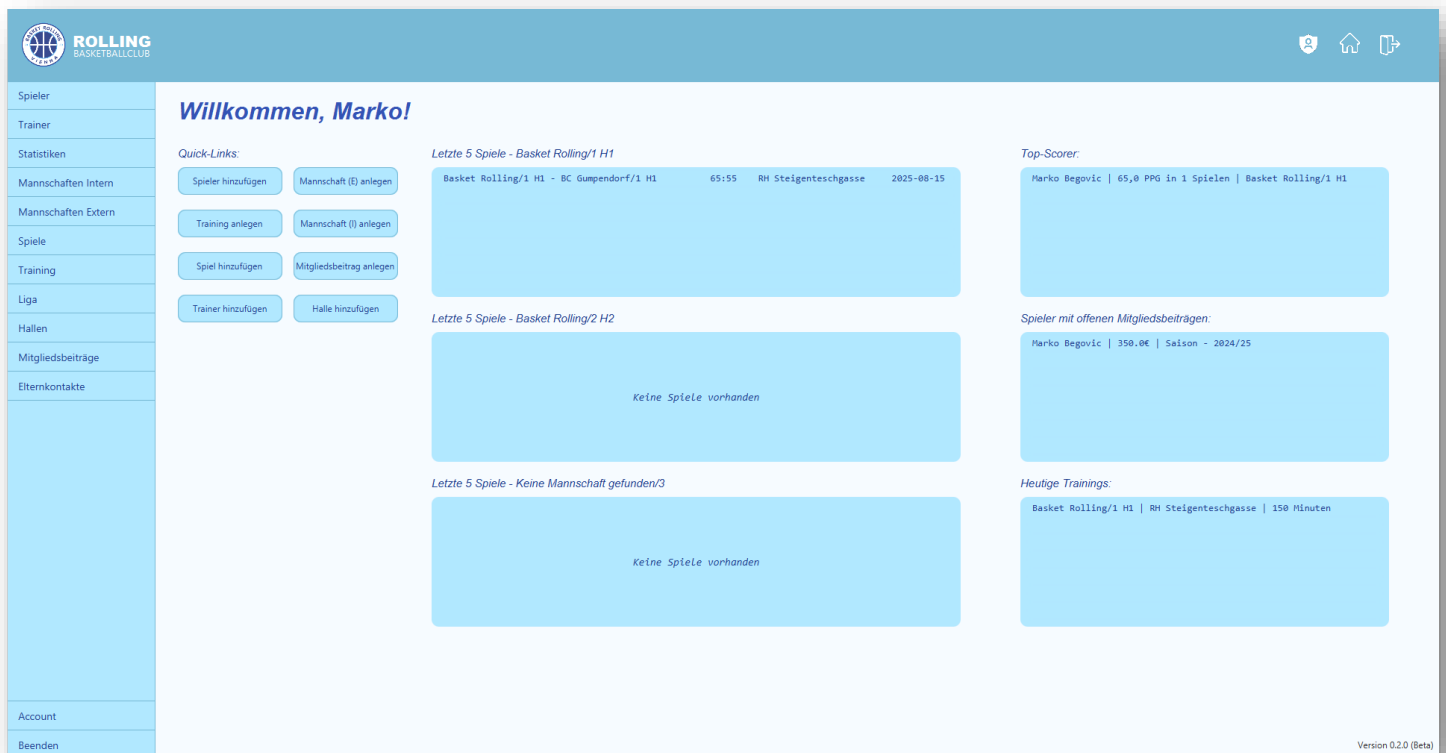
## 4. Benutzeranleitung

Die Benutzeroberfläche der *Basket Rolling Software* ist rollenbasiert aufgebaut. Nach dem Login unterscheiden sich die Rechte zwischen **Administrator** (voller Zugriff, CRUD) und **User** (nur Lesezugriff).

### 4.1 Login

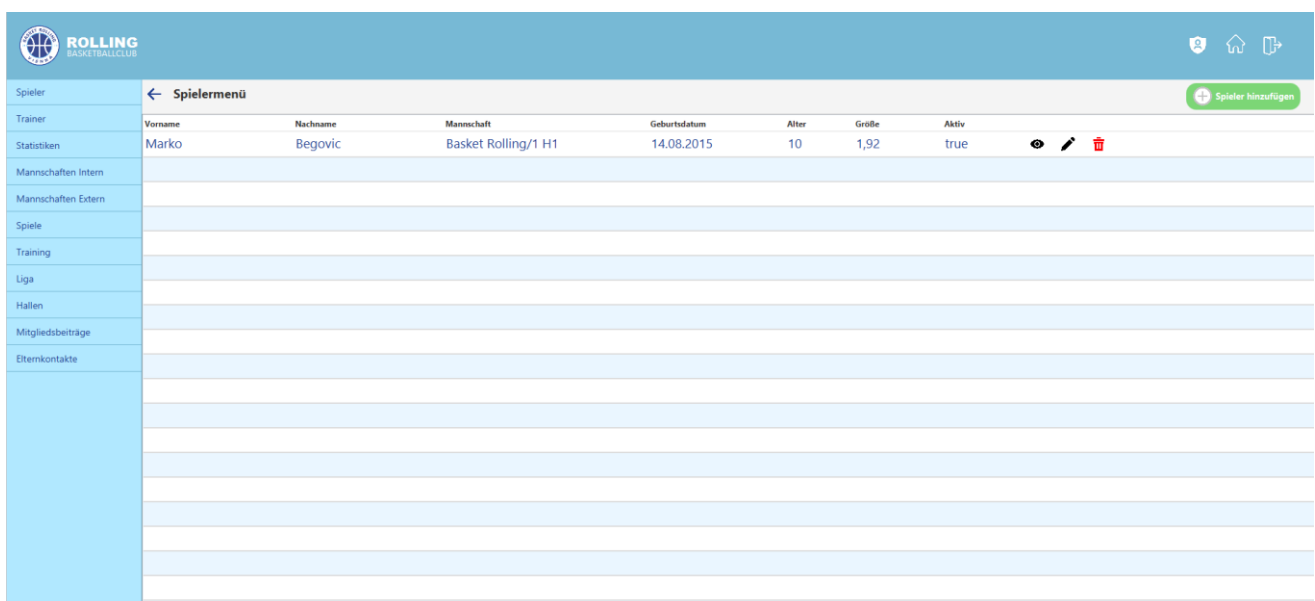
- Beim Start der Anwendung öffnet sich das **Login-Fenster**.
- Anmeldung erfolgt mit Benutzername und Passwort.
- Admins können weitere Benutzer anlegen.

Nach erfolgreichem Login erscheint das **Hauptmenü**.







Von hier aus können die zentralen Funktionen über Buttons oder Menüpunkte aufgerufen werden:

- Spieler verwalten



- Trainer verwalten


**ROLLING**  
BASKE TBALLCLUB









←
**Trainermenü**

+ Trainer hinzufügen

Spieler	Vorname	Nachname	Mannschaft	Telefon	E-Mail
<ul style="list-style-type: none"> <li>Trainer</li> <li>Statistiken</li> <li>Mannschaften Intern</li> <li>Mannschaften Extern</li> <li>Spiele</li> <li>Training</li> <li>Liga</li> <li>Hallen</li> <li>Mitgliedsbeiträge</li> <li>Elternkontakte</li> </ul>	Kein Inhalt in Tabelle				

- Interne / externe Teams verwalten






**ROLLING**  
 BASKETBALLCLUB


Spieler

← Mannschaftenmenü Intern


+ Mannschaft hinzufügen


Trainer	Mannschaftsname	Liga	Trainer
Statistiken	Basket Rolling/1 H1	1. Herrenliga (H1)	 
Mannschaften Intern	Basket Rolling/2 H2	2. Herrenliga (H2)	 
Mannschaften Extern			
Spiele			
Training			
Liga			
Hallen			
Mitgliedsbeiträge			
Elternkontakte			


- Spiele verwalten



ROLLING  
BASKETBALLCLUB








Spieler

← Spielemenü



Spiel hinzufügen

Trainer

Rolling Mannschaft

Ergebnis

Gegner

Liga

Datum

Halle

Statistiken

Basket Rolling/1 H1


65 : 55


BC Gumpendorf/1 H1

1. Herrenliga (H1)

2025-08-15

RH Steigenteschgasse





Mannschaften Intern

Mannschaften Extern

Spiele

Training


Liga

Hallen

Mitgliedsbeiträge


Elternkontakte


- Ligen verwalten




ROLLING

BASKETBALLCLUB







Spieler

Trainer

Statistiken

Mannschaften Intern

Mannschaften Extern

Spiele

Training

Liga

Hallen

Mitgliedsbeiträge

Elternkontakte

← Ligamenü


Liga

1. Herrenliga (H1)

2. Herrenliga (H2)


+ Liga hinzufügen


- Trainings verwalten




ROLLING

BASKETBALLCLUB











Spieler

← Trainingsmenü

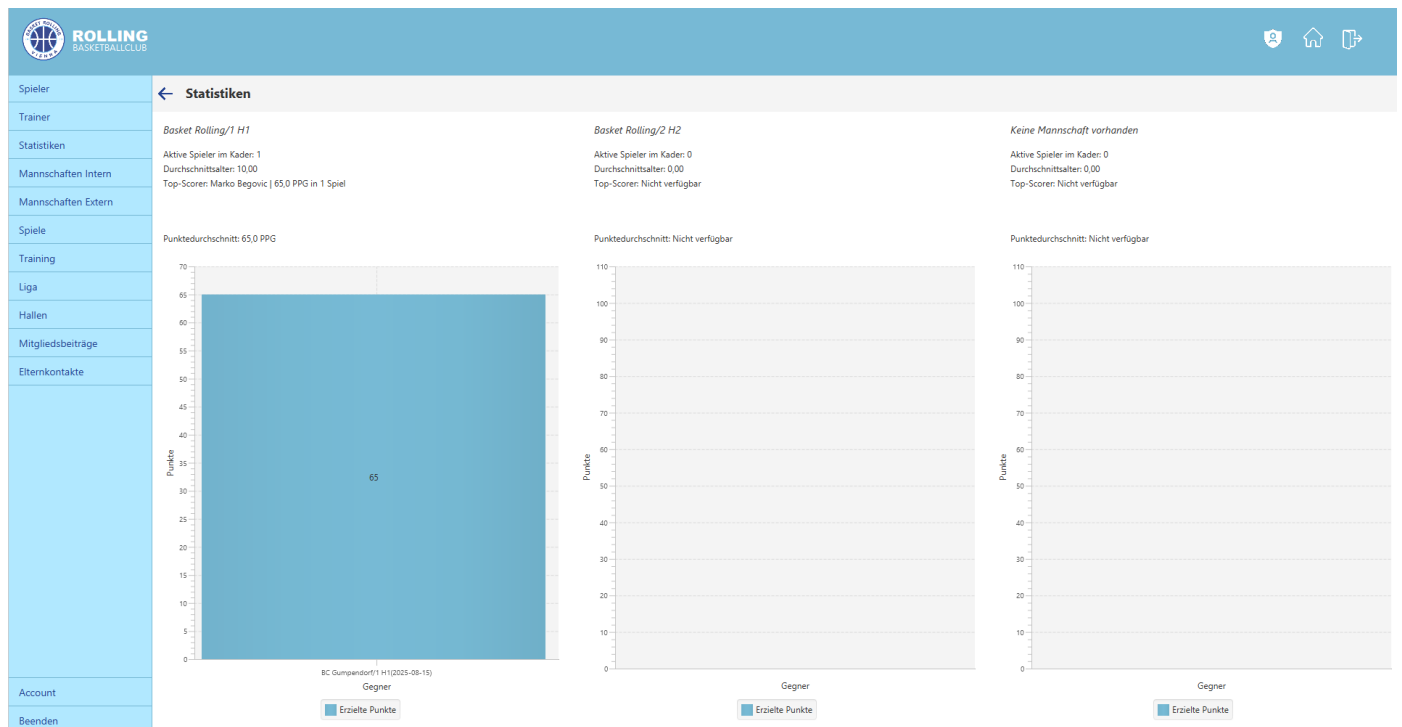
+ Training hinzufügen

Mannschaft	Halle	Tag	Dauer	
Basket Rolling/1 H1	RH Steigenteschgasse	Dienstag	150	 
</				

- Mitgliedsbeiträge verwalten

ROLLING BASKETBALLCLUB				
Spieler	← Mitgliedsbeiträge		+ Mitgliedsbeitrag hinzufügen	
Trainer	Saison	Betrag		
Statistiken	2024/25	350,00 €		
Mannschaften Intern				
Mannschaften Extern				
Spiele				
Training				
Liga				
Hallen				
Mitgliedsbeiträge				
Elternkontakte				

- Statistiken verwalten



**ACHTUNG:** Statistiken werden bei den Spielen eingetragen. Unter Statistik kann man nur die Statistik einsehen, nicht bearbeiten.

### 4.3 Spielerverwaltung

- Übersicht aller Spieler in tabellarischer Form
- Funktionen für Admins: **Anlegen, Bearbeiten, Löschen**
- Eingabe von: Name, Geburtsdatum, Größe, E-Mail, Mannschaft, Elternkontakte
- Pflichtfelder werden geprüft

Spieler hinzufügen

**Spieler hinzufügen**

Vorname: \*

Nachname: \*

Geburtsdatum: \*

Größe: \*

E-Mail:

Aktiver Spieler? ☐

Mannschaft: \*

Mitgliedsbeitrag:

Mitgliedsbeitrag bezahlt? ☐

\* Pflichtfeld

Speichern Abbrechen

### 4.4 Trainerverwaltung

- Übersicht aller Trainer
- Funktionen für Admins: **Anlegen, Bearbeiten, Löschen**
- Eingabe von: Name, Telefonnummer, E-Mail

Trainer hinzufügen

**Trainer hinzufügen**

Vorname: \*

Nachname: \*

Telefon: \*

E-Mail:

\* Pflichtfeld

Speichern Abbrechen



## 4.5 Teamverwaltung

- **Interne Teams:** mit Zuordnung zu Liga, Training, Trainer
- **Interne Teams** müssen mit /1, /2 oder /3 gekennzeichnet werden
- **Externe Teams:** Gegner teams
- Teams können anschließend Spielen zugeordnet werden

The screenshot shows a dialog box titled "Mannschaft hinzufügen (Intern)". It contains the following fields and controls:

- Mannschaftsname: \***: A text input field with a blue border and a small question mark icon to its right.
- Liga: \***: A dropdown menu.
- Trainer:**: A dropdown menu.
- \* Pflichtfeld**: A label indicating that the name field is mandatory.
- Buttons**: "Speichern" and "Abbrechen".

The screenshot shows a dialog box titled "Mannschaft hinzufügen (Extern)". It contains the following fields and controls:

- Mannschaftsname: \***: A text input field.
- Liga: \***: A dropdown menu.
- \* Pflichtfeld**: A label indicating that the name field is mandatory.
- Buttons**: "Speichern" and "Abbrechen".

## 4.6 Hallenverwaltung

- Übersicht der verfügbaren Hallen
- Eingabe von Namen, Adresse, Postleitzahl
- Validierung verhindert unvollständige Eingaben

The screenshot shows a dialog box titled "Halle hinzufügen". It contains the following fields and controls:

- Halle: \***: A text input field.
- Straße: \***: A text input field.
- PLZ: \***: A text input field.
- Ort: \***: A text input field.
- \* Pflichtfeld**: A label indicating that all fields are mandatory.
- Buttons**: "Speichern" and "Abbrechen".

## 4.7 Spieleverwaltung

- Neue Spiele anlegen mit Datum, Halle, Liga, internem und externem Team
- Ergebnisse (Punkte) eintragen
- Zuordnung zur Liga
- Dubletten Prüfung (gleiche Teams + Datum darf nur einmal vorkommen)

The screenshot shows a web application window titled 'Spiel hinzufügen'. The form has a light blue header with the title. Below the header, there are several input fields: 'Liga: \*' (dropdown), 'Datum: \*' (text input with a calendar icon), 'Halle: \*' (dropdown), 'Mannschaft Intern: \*' (dropdown), 'Mannschaft Extern: \*' (dropdown), 'Punkte: \*' (text input) for both internal and external teams. At the bottom, there are two buttons: 'Speichern' (Save) and 'Abbrechen' (Cancel). A note '\* Pflichtfeld' (required field) is located at the bottom left.

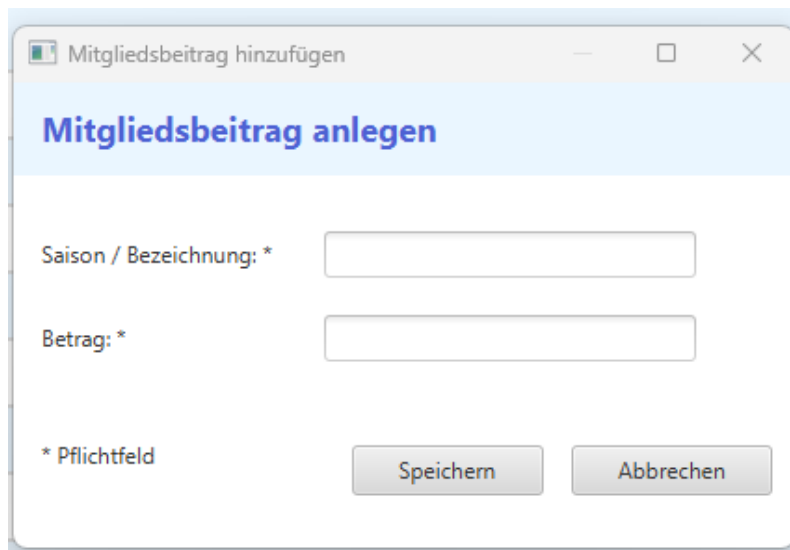
## 4.8 Training

- Zuordnung von Trainings zu Mannschaften
- Eingabe: Wochentag, Dauer, Halle
- Dauer darf zwischen 1–150 Minuten betragen

The screenshot shows a web application window titled 'Training hinzufügen'. The form has a light blue header with the title. Below the header, there are several input fields: 'Mannschaft: \*' (dropdown), 'Halle: \*' (dropdown), 'Tag: \*' (dropdown), and 'Dauer in Minuten: \*' (text input). At the bottom, there are two buttons: 'Speichern' (Save) and 'Abbrechen' (Cancel). A note '\* Pflichtfeld' (required field) is located at the bottom left.

## 4.9 Mitgliedsbeiträge

- Verwaltung der jährlichen Mitgliedsbeiträge
- Eingabe: Saison / Bezeichnung (z.b. 2024/25), Betrag



Mitgliedsbeitrag hinzufügen

### Mitgliedsbeitrag anlegen

Saison / Bezeichnung: \*

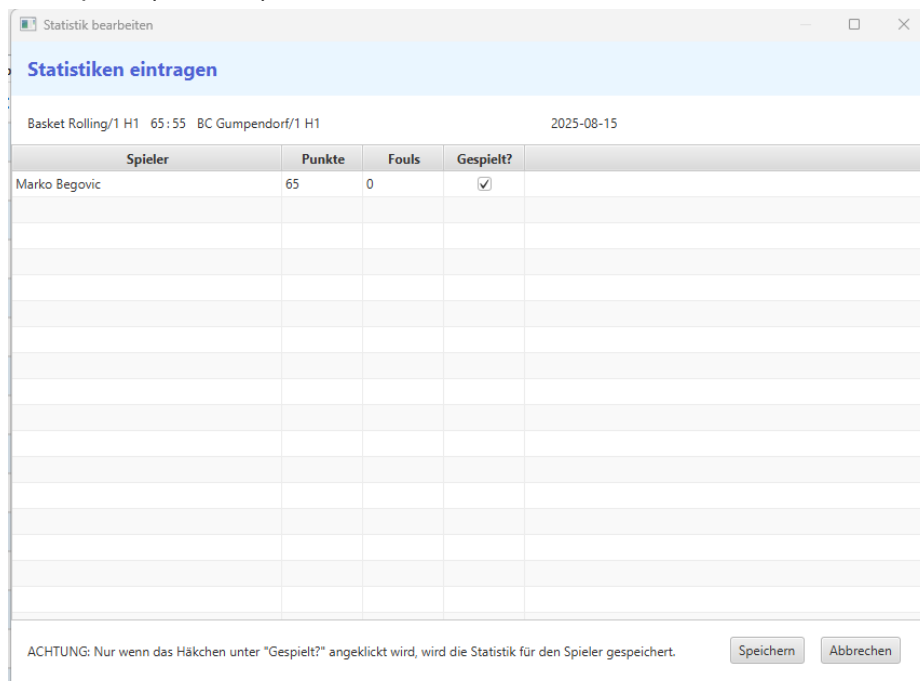
Betrag: \*

\* Pflichtfeld

Speichern Abbrechen

## 4.10 Statistiken

- Pro Spiel individuelle Statistiken pro Spieler erfassen:
- Punkte
- Fouls
- Gespielt (Ja/Nein)



Statistik bearbeiten

### Statistiken eintragen

Basket Rolling/1 H1 65:55 BC Gumpendorf/1 H1 2025-08-15

Spieler	Punkte	Fouls	Gespielt?
Marko Begovic	65	0	<input checked="" type="checkbox"/>

ACHTUNG: Nur wenn das Häkchen unter "Gespielt?" angeklickt wird, wird die Statistik für den Spieler gespeichert.

Speichern Abbrechen

## 5. Technische Details

### 5.1 Architektur

Die Anwendung folgt einer klassischen **3-Schichten-Architektur (MVC-Pattern)**:

- **Präsentationsschicht (View/Controller)**
  - Implementiert mit **JavaFX**
  - Enthält alle GUI-Komponenten und Controller, die die Benutzerinteraktionen steuern
- **Geschäftslogik (Service-Schicht)**
  - Enthält die Business-Logik der Anwendung
  - Führt Validierungen durch (z. B. Pflichtfelder, Eingabeformate, Rollenrechte)
  - Vermittelt zwischen Controller und Datenzugriffsschicht
- **Datenzugriffsschicht (DAO)**
  - Zugriff auf die PostgreSQL-Datenbank über **Hibernate (JPA)**
  - CRUD-Operationen auf Entities

Dieses Schichtenmodell ermöglicht eine klare Trennung von GUI, Logik und Datenbank. Dadurch wird der Code besser wartbar und erweiterbar.

### 5.1 Klassenübersicht

**Model (Entities, Beans):**

- **Login** – Benutzerverwaltung (Admin/User)
- **Spieler** – Spieler mit persönlichen Daten und Zuordnung zu Mannschaften
- **Trainer** – Trainer mit Kontaktinformationen
- **MannschaftIntern** – interne Teams des Vereins
- **MannschaftExtern** – externe Teams (z. B. Gegner)
- **Halle** – Hallenverwaltung
- **Training** – Trainingszeiten (Wochentag, Dauer, Halle)
- **Spiele** – Spiele mit Datum, Teams, Liga, Ergebnis
- **Liga** – Ligen, denen Teams und Spiele zugeordnet werden
- **Mitgliedsbeitrag** – Mitgliedsbeiträge
- **MitgliedsbeitragZuweisung** – Zuordnung Beiträge zu Spielern

- **Elternkontakt** – Kontaktinformationen für Spieler (Pflicht bei Jugendmannschaften)
- **Statistik** – individuelle Spielstatistiken (Punkte, Fouls, Starter)

### DAO (Data Access Objects):

- Pro Entity eine DAO-Klasse, die CRUD-Operationen über JPA/Hibernate abwickelt

### Service:

- Pro Entity eine Service-Klasse, die zwischen Controller und DAO vermittelt und zusätzliche Logik enthält (z. B. Validierung, Rollenprüfung)

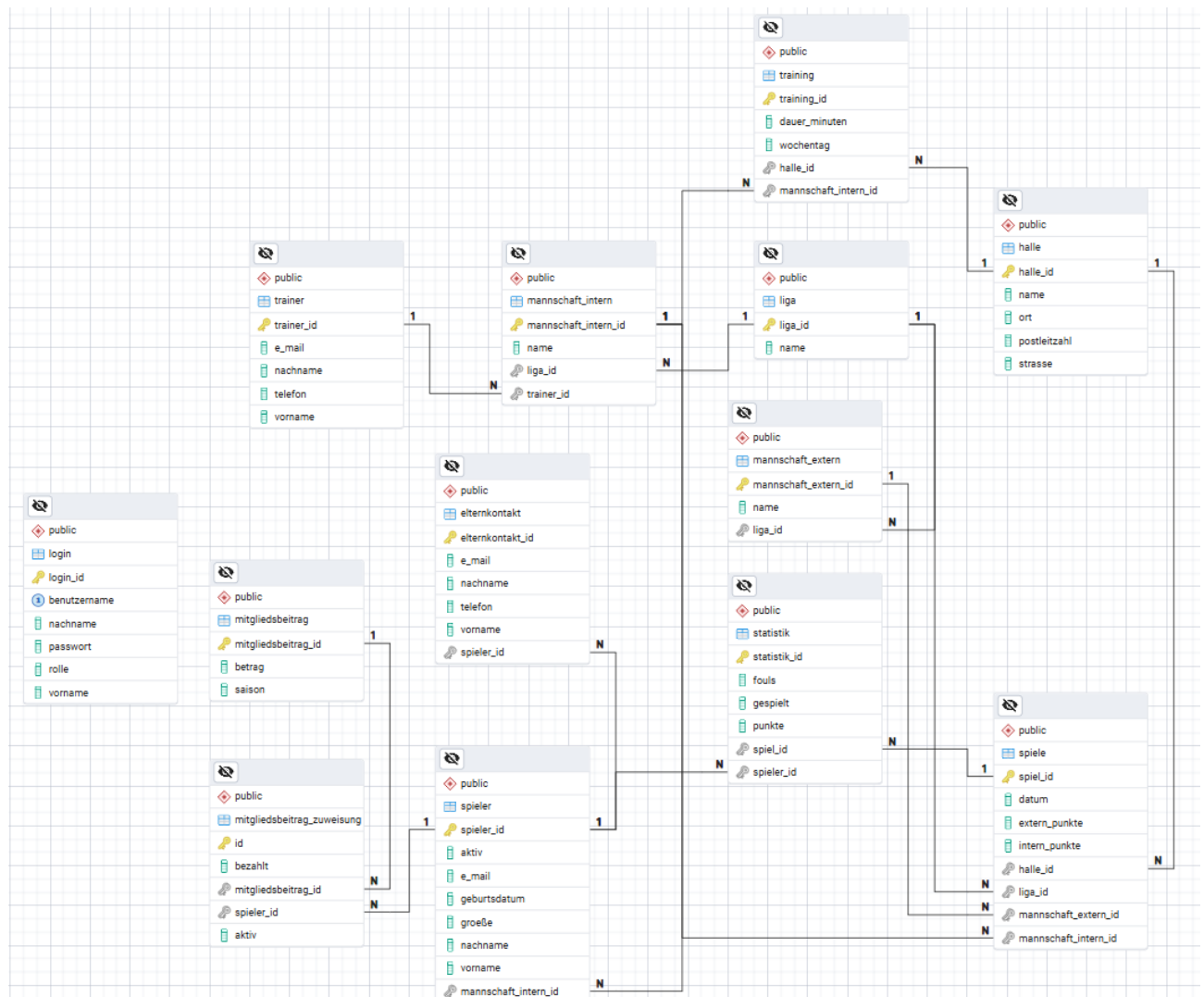
### Controller:

- Pro Eingabemaske/GUI eine Controller-Klasse, die Eingaben ausliest, Validierungen anstößt und Services aufruft

## 5.3 Datenbankmodell

Die Datenbank basiert auf **PostgreSQL**.

Beziehungen:



## 6. Fehlerbehandlung & Einschränkungen

### 6.1 Fehlermeldungen

Alle Fehlermeldungen und Hinweise werden in der Anwendung über die Hilfsklasse **AlertUtil** gesteuert.

Diese Klasse kapselt die Erzeugung von JavaFX-Alerts (Warnung, Information, Bestätigung) und sorgt damit für eine einheitliche Darstellung der Meldungen im gesamten System.

#### **Beispiel:**

*AlertUtil.alertError(„Fehler“, „Der eingegebene Wert ist nicht gültig“);*

### 6.2 Typische Benutzereingabefehler

Eingabe von Text statt Zahl (z. B. bei **Größe** oder **Dauer**)

- Eingabe nur einer Zahl als Straße (z. B. „1“) → wird durch Validierung verhindert
- Vergessen von Pflichtfeldern (z. B. Name, Telefonnummer, Elternkontakt bei U8–U16)
- Versuch, ein Spiel doppelt anzulegen (gleiches Datum + gleiche Teams)

Die Anwendung verhindert in allen Fällen das Speichern und gibt über AlertUtil eine passende Rückmeldung.

### 6.3 Bekannte Einschränkungen

#### **Benutzerregistrierung:**

- Der erste Administrator muss über die Klasse RollingDemo in der Datenbank angelegt werden.
- Danach können nur Administratoren neue Benutzer im System anlegen; eine freie Registrierung für externe User existiert nicht.

#### **Ligensystem:**

- Aktuell kein automatisches Tabellensystem; Ligen dienen nur zur Zuordnung von Mannschaften und Spielen.

#### **GUI:**

- Funktional, jedoch nicht für mobile Endgeräte optimiert.

### Performance:

- Getestet mit kleinen bis mittleren Datenmengen (< 1.000 Spieler). Bei sehr großen Datenbeständen (> 10.000 Spiele/Statistiken) fehlen Optimierungen.

### Sicherheit:

- Passwort-Hashing über *Password4j* ist implementiert, jedoch keine Passwort-Reset-Funktion über E-Mail.

## 7. Wartung & Erweiterung

### 7.1 Zentrale Konfigurationspunkte

- **Datenbankverbindung:**
  - Datei: src/main/resources/META-INF/persistence.xml
  - Enthält Datenbank-URL, Benutzername, Passwort, Hibernate-Konfiguration
  - Änderungen hier ermöglichen den Wechsel auf eine andere Datenbank (z. B. MySQL)
- **Login & Benutzerrollen:**
  - Initialer Admin wird über RollingDemo.java angelegt
  - Weitere Benutzer können im Programm durch Administratoren erstellt werden
  - Rollen sind in der Entity Login als Enum Rolle definiert
- **Projektabhängigkeiten:**
  - Definiert in pom.xml (Maven)
  - Neue Libraries können hier eingebunden werden
- **GUI-Struktur:**
  - JavaFX-Views liegen in den FXML-Dateien
  - Controller-Klassen befinden sich im Package ***org.basketrolling.gui.controller***
  - Änderungen an der Benutzeroberfläche erfolgen hier