Prototyping Projektdokumentation

Name: Adonis Shala

E-Mail: shalaado@students.zhaw.ch

URL der deployten Anwendung: <https://bundesligaprototyping.netlify.app/>

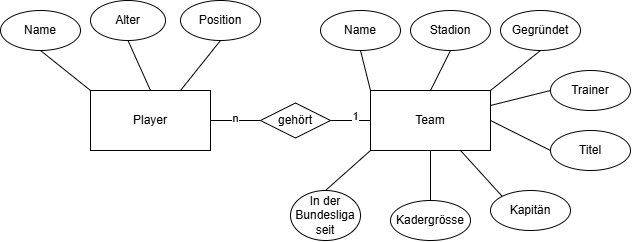
GitHub Repository: <https://github.com/shalaado/Prototyping_Abschlussarbeit>

# Einleitung

Die Anwendung „Bundesliga Vereinsverwaltung“ ist eine Webapplikation zur Verwaltung von Bundesliga-Teams und deren Spielern. Benutzer können Teams ansehen, erstellen, bearbeiten und löschen. Zusätzlich ist es möglich, Spieler zu verwalten und diesen Teams zuzuordnen. Ziel des Projekts war es, ein übersichtliches Interface mit klaren Workflows und einer professionellen, farblich an die Bundesliga angelehnten Darstellung zu entwickeln.

Die Anwendung wurde mit SvelteKit als Frontend-Framework und MongoDB als Datenbank realisiert. Bootstrap wurde zur Gestaltung des Layouts eingesetzt.

# Datenmodell



# Beschreibung der Anwendung

## Startseite

Route: /Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Startseite dient als Einstiegspunkt der Bundesliga Vereinsverwaltung. Sie bietet eine kurze Einführung in die Anwendung und ermöglicht durch zwei prominente Buttons den direkten Zugriff auf die Teams- oder Spielerübersicht. Die Benutzeroberfläche ist responsiv gestaltet und hebt das Bundesliga-Branding mit einem zentral platzierten Logo hervor.

Funktionalität:

* Überblick über den Zweck der Anwendung
* Navigation zu /teams oder /players
* Statische Willkommensnachricht

Dateien:

* routes/+page.svelte
* routes/+layout.svelte

## 3.2 Teamübersicht

Route: /teams

Ein Bild, das Text, Screenshot, Logo, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Seite «Teams» wird eine Übersicht aller Vereine der deutschen Fussball-Bundesliga angezeigt. Jeder Verein wird in einer eigenen Karte dargestellt. Die Karte enthält den Vereinsnamen, das entsprechende Vereinslogo sowie einen Button mit der Aufschrift «Details ansehen».

Durch Klicken auf diesen Button gelangt man zur Detailseite des jeweiligen Vereins, auf der weiterführende Informationen einsehbar sind. Die Logos visualisieren die Vereine klar erkennbar und sorgen für eine ansprechende Darstellung der Übersicht.

Dateien:

* routes/teams/+page.svelte
* routes/teams/+page.server.js
* lib/components/Team.svelte

## Team-Details

Route: /teams/[id]

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Beim Aufrufen dieser Seite werden alle Informationen zum ausgewählten Team übersichtlich dargestellt. Dazu zählen der Teamname, das Stadion, das Gründungsjahr, der aktuelle Trainer, der Kapitän, die Kadergrösse, das Bundesliga-Debütjahr sowie die Anzahl der gewonnenen Meistertitel.

Am unteren Rand der Karte befinden sich drei Schaltflächen:

* Mit dem Button «zurück zur Übersicht» gelangt man zurück zur Teamübersicht.
* Über den Button «Team bearbeiten» wird die Detailseite zur Bearbeitung des Teams geöffnet (routes/teams/[id]/edit).
* Durch Klicken auf «Team löschen» kann der angezeigte Verein vollständig aus der Datenbank entfernt werden.

Diese Seite dient der detaillierten Einsicht und Verwaltung einzelner Teams.

Dateien:

* routes/teams/[id]/+page.svelte
* routes/teams/[id]/+page.server.js
* lib/components/Team.svelte

## Team-Details bearbeiten

Route: /teams/[id]/edit

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite kann ein bestehendes Team inhaltlich bearbeitet werden. Es werden alle relevanten Felder vorausgefüllt angezeigt, darunter der Teamname, das Stadion, das Gründungsjahr, der Trainer, der Kapitän, die Kadergrösse, das Jahr des Bundesliga-Beitritts sowie die Anzahl der Meistertitel.

Nutzer können die gewünschten Daten in den Eingabefeldern anpassen und mit einem Klick auf «Speichern» dauerhaft aktualisieren.  
Am unteren Rand befindet sich ausserdem ein Link «zurück zur Detailseite», der ohne Änderung zur vorherigen Ansicht des Teams zurückführt.

Dateien:

* routes/teams/[id]/edit/+page.svelte
* routes/teams/[id]/edit/+page.server.js
* lib/components/Team.svelte

## Team hinzufügen

Route: /teams/create

Ein Bild, das Text, Screenshot, Zahl, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese Seite dient der Erfassung eines neuen Bundesliga-Teams. Benutzer können hier ein Team vollständig manuell anlegen, indem sie die relevanten Informationen in ein Formular eingeben.  
Folgende Felder stehen zur Verfügung:

* Name des Vereins
* Stadion
* Gründungsjahr
* Trainer
* Kapitän
* Kadergrösse
* Bundesliga seit (Jahreszahl)
* Meistertitel (Anzahl)

Nach dem Ausfüllen der Eingabefelder wird durch Klicken auf den Button «Erstellen» das neue Team gespeichert und zur Übersicht hinzugefügt.  
Über den Link «zurück zur Übersicht» gelangt man ohne Speicherung wieder zur Teamübersicht.

Dateien:

* routes/teams/create/+page.svelte
* routes/teams/create/+page.server.js
* lib/components/Team.svelte

## Spielerübersicht

Route: /players

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite werden alle in der Anwendung erfassten Spieler in einer übersichtlichen Liste dargestellt. Jeder Spieler wird dabei als einzelne Karte angezeigt, welche den Namen und ein Platzhalterbild enthält.

Durch Klicken auf einen Spieler wird man zur Detailseite des jeweiligen Spielers weitergeleitet. Diese Übersicht dient somit als zentrale Anlaufstelle, um alle registrierten Spieler schnell zu überblicken und direkt auf weitere Informationen zuzugreifen.

Dateien:

* routes/players/+page.svelte
* routes/players/+page.server.js
* lib/components/Player.svelte

## Spielerdetails

Route: /players/[id]

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese Seite zeigt alle verfügbaren Informationen zu einem einzelnen Spieler an. Neben dem Namen werden Alter, Position und der zugehörige Verein übersichtlich in einer Karte dargestellt.

Darunter befinden sich drei Buttons:

* zurück zur Übersicht: führt zurück zur vollständigen Spielerübersicht.
* Spieler bearbeiten: öffnet die Bearbeitungsansicht für den aktuellen Spieler (routes/players/[id]/edit).
* Spieler löschen: entfernt den Spieler vollständig aus der Anwendung.

Diese Detailansicht erleichtert die individuelle Verwaltung jedes Spielers.

Dateien:

* routes/players/[id]/+page.svelte
* routes/players/[id]/+page.server.js
* lib/components/Player.svelte

## Spielerdetails bearbeiten

Route: /players/[id]/edit

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite kann ein bestehender Spieler bearbeitet werden. Die Eingabefelder sind mit den aktuellen Daten des Spielers vorausgefüllt. Es können folgende Angaben geändert werden:

* Name
* Alter
* Position (Auswahl aus vordefinierten Rollen)
* Team (Auswahl eines bestehenden Vereins)

Nach der Bearbeitung können die Änderungen mit dem Speichern-Button übernommen werden.  
Ein Link zurück zur Übersicht führt zurück zur Spielerdetailseite.

Dateien:

* routes/players/[id]/edit/+page.svelte
* routes/players/[id]/edit/+page.server.js
* lib/components/Player.svelte

## Spieler erstellen

Route: /players/create

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite kann ein neuer Spieler erstellt werden. Die Eingabemaske besteht aus folgenden Pflichtfeldern:

* Name: Vor- und Nachname des Spielers
* Alter: Numerisches Feld für das Alter
* Position: Auswahl aus den möglichen Spielerpositionen (z. B. Torwart, Verteidigung, Mittelfeld, Sturm)
* Team: Auswahl eines vorhandenen Teams aus einer Dropdown-Liste

Durch einen Klick auf den Erstellen-Button wird der Spieler gespeichert und zur Datenbank hinzugefügt.  
Die Seite enthält zudem eine Navigationsleiste zur schnellen Rückkehr zu anderen Bereichen der Anwendung.

Dateien:

* routes/players/create/+page.svelte
* routes/players/create/+page.server.js
* lib/components/Player.svelte

# Erweiterungen

## Dynamisches Laden der Teamlogos

In der Teamübersicht wird zu jedem Verein automatisch das passende Teamlogo angezeigt. Die Logos befinden sich im Verzeichnis static/images/ und sind nach Vereinsnamen benannt (z. B. Borussia Dortmund.png, 1. FC Köln.png). Das Logo wird durch eine Namenskonvention automatisch geladen und angezeigt.

Implementierung:

* Pfad: static/images/
* Verwendung in: lib/components/Team.svelte
* Zugriff via: <img src="/images/{team.name}.png" />

## Responsives Layout mit Bootstrap

Das gesamte UI ist responsiv gestaltet und passt sich auf Desktop- und Tablet-Ansichten an. Als Basis dient das Bootstrap-Framework (eingebunden via CDN). Ergänzend kommt eigenes CSS (global.css) zum Einsatz, z. B. für Farben, Abstände oder die Buttons.

Implementierung:

* Bootstrap via CDN (app.html)
* Datei: src/styles/global.css
* Beispiele: btn btn-primary, card, form-control

## Validierung der Formulareingaben

Alle Eingabeformulare (z. B. Spieler erstellen, Team bearbeiten) beinhalten HTML5-Validierung mittels required-Attributen. Dadurch wird sichergestellt, dass alle Pflichtfelder korrekt ausgefüllt sind, bevor ein Datensatz gespeichert wird.

Implementierung:

* Formulare: routes/teams/create/+page.svelte, routes/players/create/+page.svelte
* Nutzung von: <input required>, <select required>

## Verknüpfung von Spielern mit Teams

Bei der Erstellung oder Bearbeitung eines Spielers kann dieser über ein Dropdown-Menü einem bestehenden Team zugeordnet werden. Die Liste der Teams wird dynamisch aus der Datenbank geladen.

Implementierung:

* Komponenten: Player.svelte
* Routen: routes/players/create/+page.svelte, routes/players/[id]/edit/+page.svelte
* Datenzugriff: lib/db.js

## Vollständige CRUD-Funktionalität

Die Anwendung unterstützt vollständige CRUD-Funktionen für Teams und Spieler:

* Create: Neue Datensätze hinzufügen (/create)
* Read: Detailansicht über /[id]
* Update: Bearbeitung über /[id]/edit
* Delete: Löschung mit Button

Implementierung:

* Verzeichnisstruktur in routes/teams/ und routes/players/
* Serverlogik in +page.server.js
* Navigations-Buttons in jeder Detailansicht

## Navigation & Benutzerführung

Ein fixes Navigationsmenü am oberen Rand erlaubt direkten Zugriff auf:

* Startseite
* Teams
* Team hinzufügen
* Spielerübersicht
* Spieler hinzufügen

Die Navigation ist auf allen Seiten konsistent und benutzerfreundlich gestaltet.

Implementierung:

* Datei: src/routes/+layout.svelte