



Kurze Projektdokumentation

Praxisanwendung Entwicklung Basics – KUD Karadjordje Bern

Termin- & Formularverwaltung • JavaFX Desktop-App mit PostgreSQL

Studiengang / Schule	HF Informatik (HFTM)
Sozialform	Einzelarbeit
Module/Kurse	System Modelling • Requirements Engineering • Programming 1–3 • Relational Databases
Autor	Milos Stanojevic
Abgabetermin	27.01.2026
Repository	<https://github.com/milos-hftm/Praxisanwendung_Entwicklung_Basics>

Stand: 24.12.2025

Projektziel und Kontext

Ziel des Projekts ist die Entwicklung einer Desktop-Anwendung mit JavaFX zur Verwaltung von Vereinsdaten. Die Anwendung kommuniziert mit einer relationalen Datenbank (PostgreSQL) und ermöglicht CRUD-Operationen (Create, Read, Update, Delete) über eine grafische Benutzeroberfläche.

Beteiligte Kurse:

- System Modelling
- Requirements Engineering
- Programming 1–3
- Relational Databases

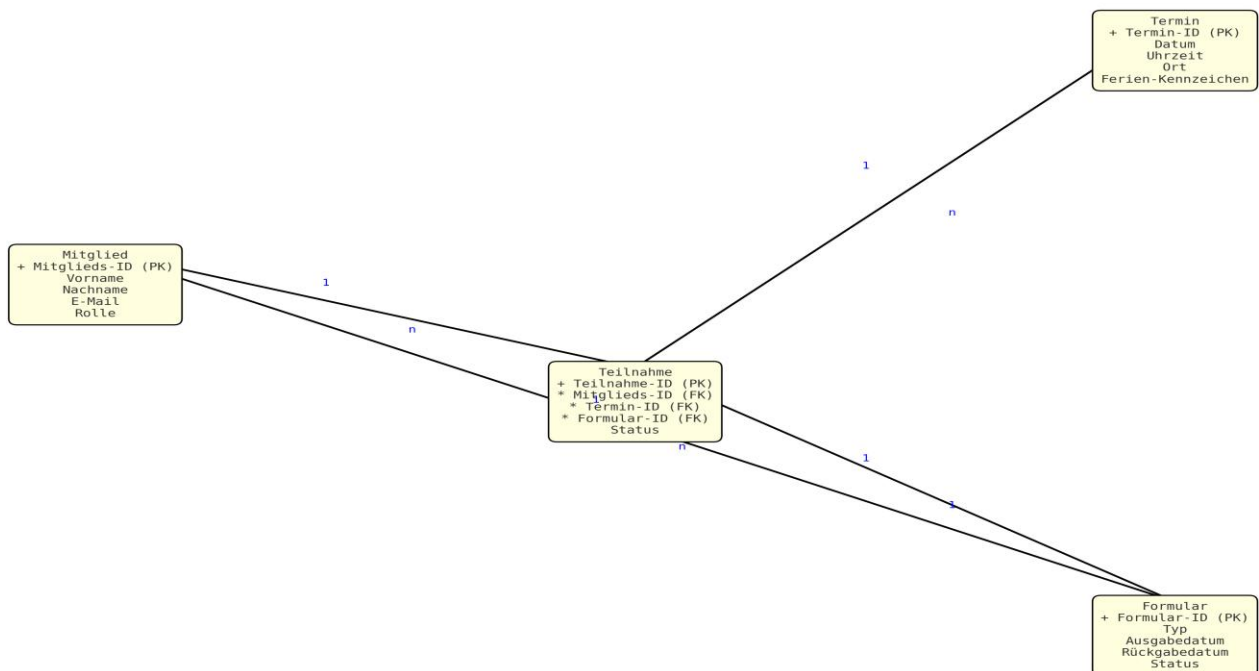
Anforderungen und Umsetzung

Die folgenden Punkte beziehen sich auf die Anforderungen aus Programming 1–3 (funktional und nicht-funktional) und zeigen, wie sie in der Applikation umgesetzt werden.

Anforderung	Umsetzung (Kurz)	Nachweis
Daten anzeigen	Anzeige aller gespeicherten Daten in Tabellen (TableView).	Screenshots (Kap. 5)
Sortieren/Filtern	Sortierung über Spaltenkopf, Filter über Suchfelder/Filterlogik.	Screenshots + Controller
Suche	Volltext-/Live-Suche über Datenbestände.	Screenshot + Suchfeld
Daten hinzufügen	Formular-Eingabe inkl. Validierung (Pflichtfelder/Formate).	Screenshot + Validierung
Daten bearbeiten	Auswahl aus Tabelle → Formular vorbefüllt → Update.	Screenshot + Update
Daten löschen	Löschen mit Sicherheitsabfrage (Confirmation Dialog).	Screenshot Delete-Dialog
Stabilität/Fehler	Fehlerbehandlung bei DB-Verbindungsproblemen/SQL-Fehlern.	Fehlerdialog/Logging
MVC	Trennung von Model, View (FXML), Controller.	Projektstruktur

Datenmodell (ER-Diagramm)

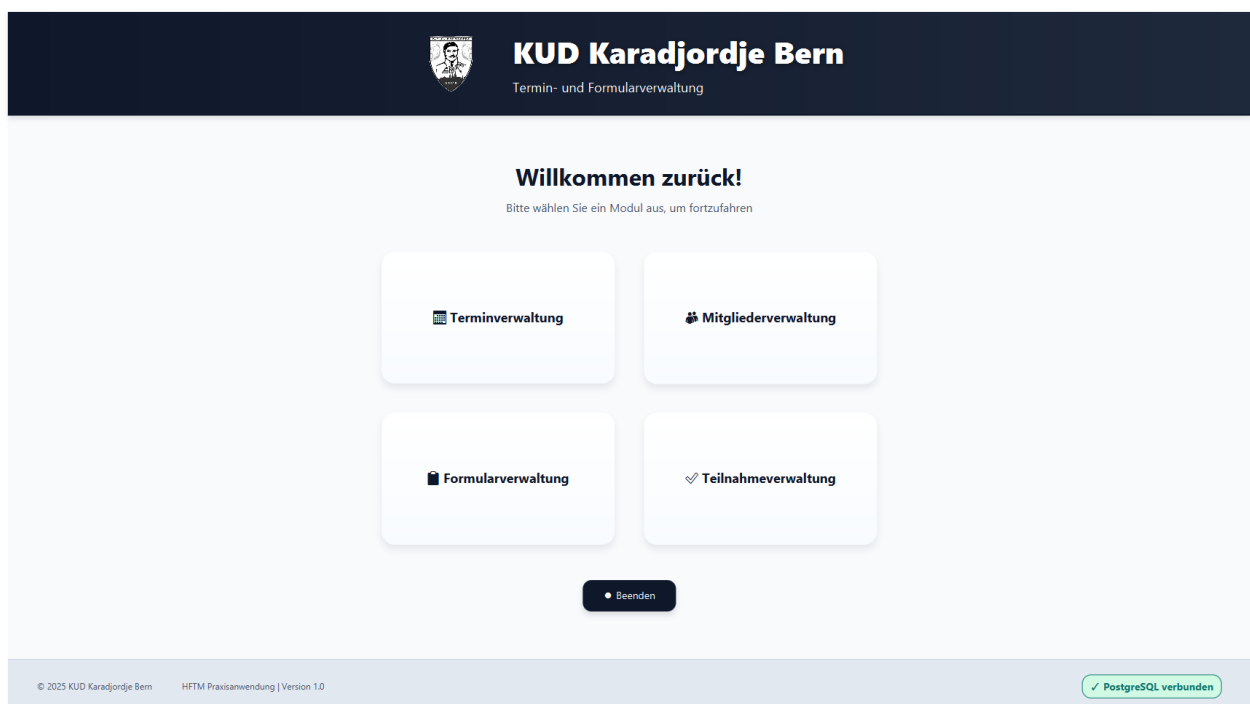
Das Datenmodell basiert auf den Entitäten Mitglied, Termin, Teilnahme und Formular. Die N:M-Beziehung zwischen Mitglied und Termin wird über Teilnahme abgebildet. Das ER-Diagramm ist als **Fachübergreifendes Transfer-Projekt.pdf** im Repository enthalten.



Screenshots / Nachweise

Screenshot 1 – Startansicht / Modulübersicht

Die Startseite dient als zentraler Einstiegspunkt. Der Benutzer wählt ein Modul aus, um Daten zu verwalten. Unten rechts wird der Datenbank-Status angezeigt („PostgreSQL verbunden“) als Feedback, dass die App korrekt mit der DB kommuniziert.



Alle Termine werden in einer Tabelle angezeigt (Read). Über die Such-/Filterleiste können Termine **gefiltert** (z. B. nach Ort/Datum) und die Ansicht **sortiert** werden (Sortierung über TableView-Spalten). Rechts befindet sich das Detail-/Eingabeformular für **Create/Update**, inklusive Buttons für **Speichern/Aktualisieren/Löschen** sowie **CSV-Export**.

Die Mitglieder werden in einer TableView angezeigt und können über die Suche (z. B. Name/E-Mail) gefiltert werden. Das Detailformular zeigt Pflichtfelder (z. B. Vorname, Nachname, E-Mail, Rolle) und unterstützt Create/Update/Delete. Zusätzlich steht ein CSV-Export zur Verfügung.

4

Screenshot 4 – Formularverwaltung (Suche/Filter, Status)

Formulare werden tabellarisch angezeigt und können über die Suche nach **Formulartyp** gefiltert werden. Mit dem Filter „Nur ausstehende Formulare“ lassen sich offene Prozesse schnell finden. Rechts können Formulardaten gepflegt und der **Status** verwaltet werden (CRUD-Workflow).

[illegible]

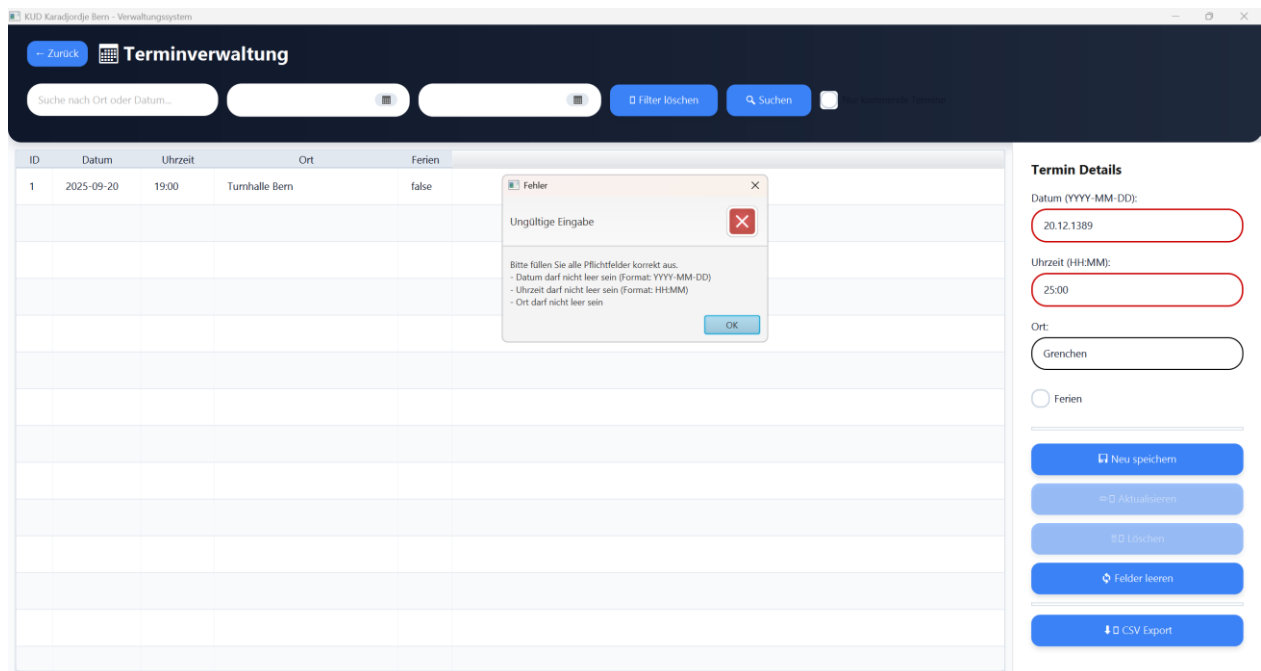
Screenshot 5 – Teilnahmeverwaltung (N:M, Volltextsuche, Filter)

Die Teilnahmeverwaltung bildet die **N:M-Beziehung** zwischen Mitgliedern und Terminen ab. Über die Volltextsuche und gezielte Filter (Mitglied-ID / Termin-ID) können Datensätze schnell gefunden werden. Rechts werden die Details gepflegt, inkl. **Status** (z. B. zugesagt/abgesagt) und optionaler Formular-Verknüpfung.

[illegible]

Screenshot 6 – Validierung & Fehlerbehandlung (Dialog)

Beim Speichern prüft die Anwendung die Eingaben (z. B. Datum/Uhrzeit/Ort). Bei ungültigen Werten erscheint ein Fehlerdialog mit klarer Erklärung, welche Felder korrigiert werden müssen. Das erfüllt die nicht-funktionale Anforderung „stabil und fehlertolerant“ sowie die funktionale Anforderung „Validierung der Eingaben“.



Screenshot 7 – Löschen mit Sicherheitsabfrage

Beim Klick auf „Löschen“ erscheint eine Bestätigung („Termin löschen?“), um versehentliche Löschungen zu verhindern. Erst nach **OK** wird der Datensatz per JDBC aus der Datenbank entfernt; bei **Abbrechen** bleibt alles unverändert.

