

Projektmanagement

- Die neue Form der Firmenorganisation -

1 Lastenheft

Bei dem firmeninternen Projektmanagementsystem werden Mitarbeiter der Firma unter Angabe ihrer Ausbildung (Elektroniker etc.) und einer Beschreibung ihrer Tätigkeit vom Administrator registriert. Zudem müssen sich die Mitarbeiter mit ihrem Vornamen, Nachnamen, ihrem Geburtsdatum sowie ihrer E-Mail Adresse und Telefonnummer registrieren. Das erstellte Profil des Mitarbeiters kann von ihm selbst bearbeitet werden.

Nur der Administrator des Systems kann Mitarbeiter eintragen und diese wieder entfernen.

Des Weiteren müssen alle Räume, ihre Funktionen und ob sie frei sind, eingetragen sein. Räume können zu jedem Projekt hinzu gebucht werden.

Jeder Benutzer kann ein Projekt erstellen und andere Benutzer eintragen. Zu jedem Projekt muss ein Zeitplan erstellt werden der angibt, in welchem Zeitraum das Projekt läuft.

Nur der Projektleiter und der Administrator kann das Projekt bearbeiten und löschen.

Der vom Administrator eingetragene Benutzer kann auf seiner Benutzeroberfläche seine Daten und die Projekte an denen er beteiligt ist, sowie den Zeitplan, einsehen. Projekte die ein Benutzer leitet werden in der Liste ganz oben angezeigt.

Der Systemadministrator kann eine Tabelle der Benutzer einsehen, hinzufügen oder löschen.

2 Pflichtenheft

Hier wird eine Auflistung der derzeitig berücksichtigten Anwendungsfälle innerhalb des Projektmanagementsystems gezeigt.

Benutzeroberfläche

- Benutzerprofil einsehen / bearbeiten
- Projekte anzeigen (an denen er beteiligt ist)
- Projekt bearbeiten / löschen (in denen er Projektleiter ist)
- Projekt erstellen
- Zeitplan hinzufügen
- Raum hinzu buchen

Administrator

- Benutzer hinzufügen / Administratorrechte verwalten
- Benutzer löschen
- Benutzer editieren
- Projekte erstellen / löschen / bearbeiten
- Zeitplan löschen
- Räume eintragen / löschen / bearbeiten

2.1 Benutzerverwaltung

2.1.1 Benutzerprofil einsehen

A1 Das System soll es dem Benutzer ermöglichen sein Benutzerprofil einzusehen.

2.1.2 Benutzerprofil bearbeiten

A2 Das System soll es dem Benutzer ermöglichen sein Benutzerprofil zu bearbeiten.

A2.1 Das System soll es dem Benutzer ermöglichen nur seine Beschreibung der Tätigkeit, seine Ausbildung, seine Telefonnummer und sein Passwort zu bearbeiten.

2.2 Administratorenverwaltung

Es gibt festgelegte Administratoren im System. Ein Administrator hat seine ganz eigenen Rechte. (siehe „Administrator“ Aufzählung bei Punkt 2.)

2.2.1 Benutzer hinzufügen

A3 Das System soll es dem Administrator ermöglichen, Benutzerprofile in das System hinzuzufügen.

A3.1 Wenn der Administrator bei der Erstellung eines Profils eine bereits vorhandene E-Mail-Adresse eingibt, soll das System eine Fehlermeldung anzeigen. (Wird als SQL-Error ausgegeben)

2.2.2 Benutzer löschen

A4 Das System soll es dem Administrator ermöglichen, Benutzerprofile aus dem System zu löschen.

A4.1 Wenn der Administrator versucht ein Profil zu löschen, welches zurzeit in einem Projekt eingetragen ist, soll das System eine Warnmeldung anzeigen. (SQL-Error)

2.2.3 Benutzer editieren

A5 Das System soll es dem Administrator ermöglichen, Benutzerprofile im System zu editieren.

2.2.4 Administratorrechte verwalten

A6 Das System soll es dem Administrator ermöglichen Benutzer Administratorstatus zu geben und zu entfernen. (hierdurch könnte man sich theoretisch eine Monarchie aufbauen)

2.2.4 Räume hinzufügen

A7 Das System soll es dem Administrator ermöglichen, Räume dem System hinzuzufügen.

2.2.5 Räume entfernen

A8 Das System soll es dem Administrator ermöglichen, Räume aus dem System zu entfernen.

A8.1 Wenn ein Raum belegt ist, kann er nicht gelöscht werden.

2.2.6 Räume bearbeiten

A9 Das System soll es dem Administrator ermöglichen Räume zu editieren.

2.3 Projektverwaltung

2.3.1 Projekt anzeigen

A10 Das System soll es dem Benutzer ermöglichen, alle Projekte an denen er beteiligt ist anzeigen zu lassen. Projekte in denen er Projektleiter ist, werden besonders hervorgehoben.

2.3.2 Projekt erstellen

A11 Das System soll es dem Benutzer ermöglichen unter Angabe des Projektnamens ein Projekt zu erstellen.

A11.1 Wenn der Benutzer beim Erstellen eines Projekts einen bereits verwendeten Namen eingibt, soll das System eine Fehlermeldung anzeigen.

A11.2 Wenn ein Projekt (erfolgreich) erstellt wird, soll das System Datum und Uhrzeit der Erstellung speichern. Der Projektleiter kann frei gewählt werden und muss direkt angegeben werden.

2.3.3 Projekt löschen

A12 Das System soll es dem Benutzer ermöglichen, ein von ihm erstelltes Projekt zu löschen. Dies kann nur geschehen wenn er in dem Projekt auch der Projektleiter ist.

A12.1 Wenn ein Projekt (erfolgreich) gelöscht wird, soll das System die Zuordnung der Benutzer entfernen (unabhängig davon, wer diese erstellt hat).

2.3.3 Projekte bearbeiten

A13 Das System soll es dem Administrator ermöglichen, alle Projekte zu editieren.

2.4 Projektleiterverwaltung

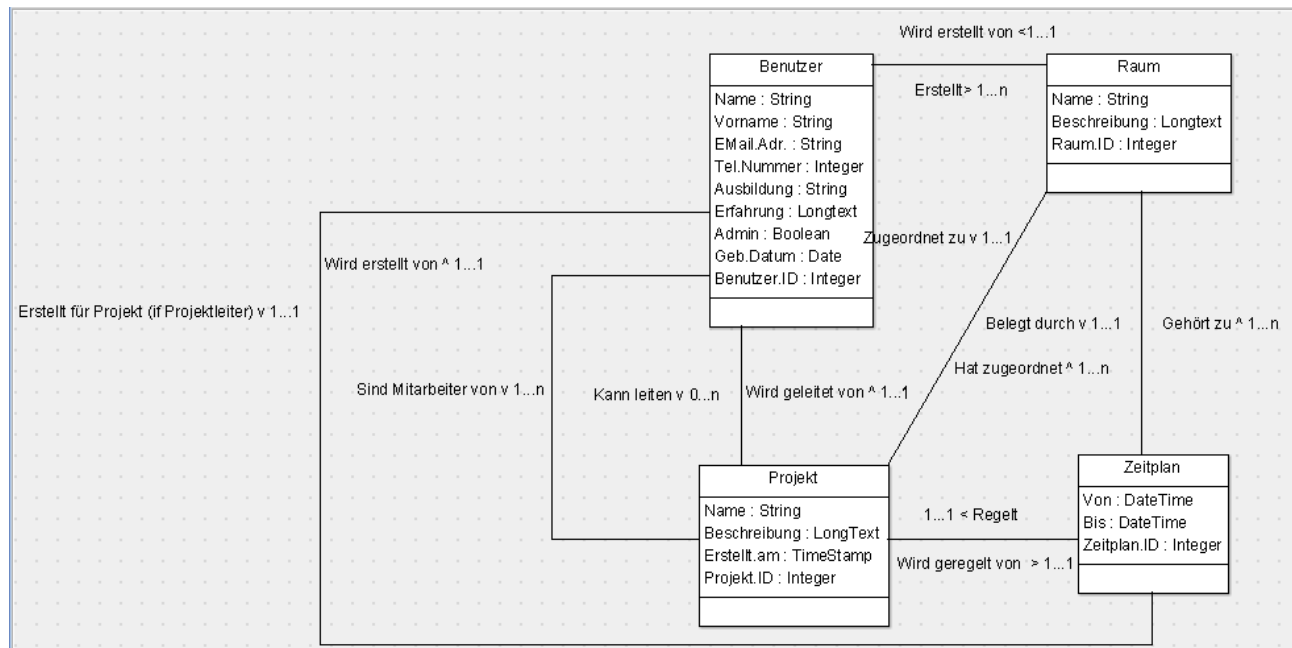
Ein Projektleiter erbt alle Anwendungsmöglichkeiten eines Benutzers.

2.4.1 Benutzer einsehen

A14 Das System soll es dem Benutzer beim Erstellen eines Projektes ermöglichen, eine Liste aller Benutzer einzusehen. Aus diesen kann er dann Benutzer auswählen, die an dem Projekt beteiligt sein sollen.

3 Datenmodell

Diese Abbildung zeigt das Datenmodell in Form eines UML-Klassendiagramms. Hier werden zunächst die Datentypen *String* für Zeichenketten, *Boolean* für Ja/Nein, *Integer* für Zahlenwerte, *DateTime* für Zeitstempel und *Longtext* für Texteingaben verwendet. Längenbegrenzungen für die Attributwerte werden erst im Rahmen des Software-technischen Entwurfs festgelegt.



4 Software-Architektur

Die Umsetzung des ProManagers erfolgt als Client-Server-Anwendung mittels einer klassischen Drei-Schichten-Architektur. Die *Darstellungsschicht* befindet sich auf dem Client und wird mittels HTML und CSS realisiert. Die *Anwendungsschicht* in Form von PHP-Skripten läuft auf dem Web-Server. Zur Implementierung der *Datenhaltungsschicht* kommt das Datenbankmanagementsystem MySQL zum Einsatz.

In der vorliegenden Version dieses Dokuments wird nur die Realisierung der Datenhaltungsschicht beschrieben.

5 Implementierung

Im Folgenden sind die Anforderungen des Pflichtenhefts den einzelnen SQL-Befehlen mittels ihrer ID zugeordnet. Einige Anforderungen können auf der Ebene der Datenhaltungsschicht allein durch Fremdschlüssel oder UNIQUE-Bedingungen erfüllt werden.

5.1 Datenbank-Schema

Tabellen erhalten einen künstlichen Primärschlüssel vom Typ INTEGER, der beim Einfügen einer neuen Zeile mit Hilfe des MySQL-spezifischen Zusatzes AUTO_INCREMENT erzeugt wird. Da künstliche Primärschlüssel niemals geändert werden sollten, wird bei der Definition von Fremdschlüsseln grundsätzlich die Klausel ON UPDATE RESTRICT verwendet.

Bei allen natürlichen Schlüsselkandidaten wird die Eindeutigkeit durch eine UNIQUE-Bedingung sichergestellt. Im Fall der E-Mail-Adresse war dies aufgrund der zu großen Länge in MySQL leider nicht möglich. Der entsprechende Teil im Befehl zum Anlegen der BENUTZER-TABELLE ist daher auskommentiert.

CREATE TABLE Benutzer

```
(
  Name      VARCHAR(64)    NOT NULL,
  Vorname   VARCHAR(64)    NOT NULL,
  EmailAdr   VARCHAR(64)    NOT NULL  UNIQUE,
  TelefonNr  VARCHAR(64)    NOT NULL,
  Ausbildung VARCHAR(256)   NOT NULL,
  Erfahrung  LONGTEXT       NULL,
  Admin      BOOLEAN        NOT NULL DEFAULT 0,
  GebDatum   DATE           NOT NULL,
  BenutzerID INTEGER        NOT NULL  AUTO_INCREMENT,
  Projekte   INTEGER        NULL,

  PRIMARY KEY (BenutzerID),

  FOREIGN KEY (Projekte)
  REFERENCES ProjektMitarbeiter (ProjektID)
  ON UPDATE RESTRICT
  ON DELETE SET NULL
);
```

CREATE TABLE Projektleiter

```
(
  BenutzerID  INTEGER  NOT NULL,
  ProjektID   INTEGER  NOT NULL,

  PRIMARY KEY (BenutzerID, ProjektID),

  FOREIGN KEY (BenutzerID)
  REFERENCES Benutzer (BenutzerID)
  ON UPDATE RESTRICT
  ON DELETE SET NULL,
```

```
FOREIGN KEY (ProjektID)
REFERENCES Projekt (ProjektID)
ON UPDATE RESTRICT
ON DELETE SET NULL
);
```

```
CREATE TABLE Projekt
(
    Name          VARCHAR(128)    NOT NULL UNIQUE,
    Beschreibung   LONGTEXT        NOT NULL,
    ErstelltAm     TIMESTAMP       NOT NULL DEFAULT CURRENT_TIMESTAMP,
    ProjektID      INTEGER         NOT NULL AUTO_INCREMENT,
    Projektleiter  INTEGER         NOT NULL,

    PRIMARY KEY (ProjektID),

    FOREIGN KEY (Projektleiter)
    REFERENCES Projektleiter (BenutzerID)
    ON UPDATE RESTRICT
    ON DELETE SET NULL
);
```

```
CREATE TABLE ProjektMitarbeiter
(
    BenutzerID     INTEGER NOT NULL,
    ProjektID      INTEGER NOT NULL,

    PRIMARY KEY (BenutzerID, ProjektID),

    FOREIGN KEY (BenutzerID)
    REFERENCES Benutzer (BenutzerID)
    ON UPDATE RESTRICT
    ON DELETE SET NULL,

    FOREIGN KEY (ProjektID)
    REFERENCES Projekt (ProjektID)
    ON UPDATE RESTRICT
    ON DELETE SET NULL);
```

```
CREATE TABLE Raeume
(
    Name          VARCHAR(146)    NOT NULL UNIQUE,
    Beschreibung   LONGTEXT        NOT NULL,
    RaumID         INTEGER        NOT NULL AUTO_INCREMENT,
    Zeitplan       INTEGER        NULL,
    Projekt        INTEGER        NULL,

    PRIMARY KEY (RaumID),

    FOREIGN KEY (Zeitplan)
    REFERENCES Zeitplan (ZeitplanID)
    ON UPDATE RESTRICT
    ON DELETE SET NULL,
```

```
FOREIGN KEY (Projekt)
REFERENCES Projekt (ProjektID)
ON UPDATE RESTRICT
ON DELETE SET NULL
```

```
);
```

```
CREATE TABLE Zeitplan
(
  Anfang      DATETIME NOT NULL,
  Ende        DATETIME NOT NULL,
  ZeitplanID  INTEGER   NOT NULL AUTO_INCREMENT,
  Projektleiter INTEGER NOT NULL,
  Projekt     INTEGER   NOT NULL,
```

```
PRIMARY KEY (ZeitplanID),
```

```
FOREIGN KEY (Projektleiter)
REFERENCES Benutzer (BenutzerID)
ON UPDATE RESTRICT
ON DELETE SET NULL,
```

```
FOREIGN KEY (Projekt)
REFERENCES Projekt (ProjektID)
ON UPDATE RESTRICT
ON DELETE SET NULL
```

```
);
```

5.2 Benutzerverwaltung

5.2.1 Benutzerprofil einsehen

--Anforderung A1 auf Seite 2

```
SELECT Name, Vorname, EmailAdr, TelefonNr, Ausbildung, Erfahrung, GebDatum
FROM Benutzer
WHERE BenutzerID = <BenutzerID>;
```

5.2.2 Benutzerprofil bearbeiten

```
UPDATE Benutzer,
SET EmailAdr = <neue EmailAdr>, TelefonNr = <neue TelefonNr>, Erfahrung = <neue
Erfahrung>
WHERE BenutzerID = <BenutzerID>;
```

5.2.3 Benutzerprofil erstellen

```
SELECT BenutzerID
FROM Benutzer
WHERE Admin = true and BenutzerID = <BenutzerID>;
```

--Anforderung A3 auf Seite 2

--darf nur ausgeführt werden, falls vorige Anforderung 1 liefert

INSERT INTO Benutzer (Name, Vorname, EmailAdr, TelefonNr, Ausbildung, Erfahrung, Admin, GebDatum)

VALUES (<Name>, <Vorname>, <EmailAdr>, <TelefonNr>, <Ausbildung>, <Erfahrung>, <false/true>, <GebDatum>);

5.2.4 Benutzerprofil löschen

SELECT BenutzerID

FROM Benutzer

WHERE Admin = true and BenutzerID = <BenutzerID>;

--Anforderung A4 auf Seite 2

--darf nur ausgeführt werden, falls vorige Anforderung 1 liefert

DELETE FROM Benutzer

WHERE BenutzerID = <BenutzerID>;

5.2.5 Administrator hinzufügen

UPDATE `benutzer` SET `Admin` = '1' WHERE BenutzerID = <BenutzerID>

5.3 Projektverwaltung

5.3.1 Projekt erstellen

INSERT INTO Projekte(Name, Beschreibung)

VALUES (<Name>,<Beschreibung>);

5.3.2 Projekt einsehen

SELECT (ProjektID)

FROM ProjekttMitarbeiter

WHERE BenutzerID = <BenutzerID>

SELECT(Name,Beschreibung,ErstelltAm,Projektleiter)

FROM Projekt

WHERE ProjektID = <ProjektID> *--oben ermittelte ProjektID*

5.3.3 Projekt löschen

SELECT BenutzerID

FROM Benutzer

WHERE Admin = true and BenutzerID = <BenutzerID>;

OR

SELECT BenutzerID

FROM ProjektLeiter

WHERE ProjektID = <ProjektID> and BenutzerID = <BenutzerID>;

--Anforderung A12 auf Seite 3

--darf nur ausgeführt werden, falls vorige Anforderung 1 liefert


```
DELETE FROM Projekte
WHERE ProjektID =<ProjektID>;
```

5.4 Zeitplanverwaltung

5.4.1 Zeitplan erstellen

```
INSERT INTO Zeitplan(Anfang, Ende)
VALUES (<Anfang>,<Ende>;
```

5.4.2 Zeitplan löschen

```
SELECT BenutzerID
FROM Benutzer
WHERE Admin = true and BenutzerID = <BenutzerID>;
```

--darf nur ausgeführt werden, falls vorige Anforderung 1 liefert

```
DELETE FROM Zeitplan
WHERE ZeitplanID =<ZeitplanID>;
```

5.5 Raumverwaltung

5.5.1 Raum erstellen

```
SELECT BenutzerID
FROM Benutzer
WHERE Admin = true and BenutzerID = <BenutzerID>;
```

--Anforderung A7 auf Seite 2
--darf nur ausgeführt werden, falls vorige Anforderung 1 liefert

```
INSERT INTO Raeume(Name, Beschreibung)
VALUES (<Name>,<Beschreibung>;
```

5.5.2 Raum löschen

```
SELECT BenutzerID
FROM Benutzer
WHERE Admin = true and BenutzerID = <BenutzerID>;
```

--Anforderung A8 auf Seite 3
--darf nur ausgeführt werden, falls vorige Anforderung 1 liefert

```
DELETE FROM Raeume
WHERE RaumID =<RaumID>;
```

5.5.3 Raum bearbeiten

```
UPDATE `raeume` SET `Name`=[value-1],`Beschreibung`=[value-2], WHERE RaumID =
<RaumID>
```

5.5.3 Raum buchen

```
UPDATE `raeume` SET `Projekt`=[value-1] RaumID = <RaumID>
```