|  |  |
| --- | --- |
|  | tud_logo |
|  | Dokumentation Übung 2 |
|  |  |
|  |  |
|  | 25. Juni 2016  Informatik im Bauwesen II – Gruppe 01  Florian Saumweber, 2354534  Benjamin Krauß, 2388173 |
|  |  |

# Anforderungen an die Software

Die Webanwendung soll die Verwaltung von Bauschäden, Handwerkern und Gutachtern ermöglichen, sowie als Informationsportal für Eigentümer dienen.

Das webbasierte System zur Dokumentation von Bauschäden und ihrer Behebung kennt drei verschiedene Nutzerrollen: Gutachter, Handwerker und Eigentümer.

Diese Benutzergruppen erhalten nach der Anmeldung vom System spezifische Interaktionsmöglichkeiten bzw. einen unterschiedlichen Funktionsumfang.

Folgende Daten werden hierbei verwaltet:

* Schadensarten
* Lokalisierung des Schadens (Gebäude, Stockwerk, Raum, etc.)
* Detaillierte Informationen zum Schaden
* Zusätzliche Informationen zu den Schäden (Bilder, Film, etc.)
* Involvierte Personen (Gutachter, Wohnungs-/Hausbesitzer, Handwerker)
* Spezialisierung und Qualifikation der Handwerker
* Aufträge der Handwerker zur Schadensbehebung

Folgende Funktionen des Systems werden zur Verfügung gestellt:

Für den Nutzer *Gutachter*:

* Aufnehmen neuer Gebäude, Schäden und Handwerker, sowie deren Daten ändern
* Eingeben von Verträge mit den Handwerkern
* Bestätigen von Schäden, die der Kunde meldet

Für den Nutzer *Handwerker*:

* Einsehen der Aufträge und der zu behebenden Schäden (mit Schadensart, -informationen, Status und Umfang des Auftrags)

Für den Nutzer *Eigentümer*:

* Überblicken seiner Einheiten und Schäden
* Melden weiterer Schäden, falls der Eigentümer schon Kunde ist

Die Realisierung findet in der Programmiersprache Java statt. Mittels der Datenbanksprache SQL ist für die persistente Datenhaltung eine MySQL-Datenbank vorgesehen. Die Datenbankschnittstelle soll über JavaBeans unter der Zuhilfenahme von JDBC erfolgen.

Die Präsentation und Abfrage der Daten wird mit Java Server Pages erstellt.

# Entity-Relationship-Modell

Die Datenmodellierung innerhalb des Entity-Relationship-Modells (ERM) dient dazu die Anforderungen aufzugreifen und einheitlich zu visualisieren. Sie bildet die Grundlage des Datenbankentwurfes.

Die Entitäten in diesem Modell sind *Beruf, Handwerker, Gutachter, Eigentümer, Adresse, Gebäude, Auftrag, Schaden, Schadenslage, Schadensart* und *Multimedia*. Die Verfeinerung durch weitere Attribute ist der *Abbildung 1* zu entnehmen. Die Relationen setzen die Entitäten zueinander in Beziehung, werden durch die Kardinalitäten in der Semantik erweitert und sind in der *Abbildung 1* grün gekennzeichnet.

Prinzipiell wird in der Modellierung danach gestrebt die Entitäten durch einen eindeutigen Pfad miteinander zu verbinden. Da in unserem Fall allen Nutzern der Webapplikation eine Adresse zugewiesen wurde, konnte dieses Ziel nur eingeschränkt erreicht werden.

Es wird davon ausgegangen, dass an einer Adresse z. B. mehrere Personen beheimatet sind. Jedoch besitzt eine Person oder ein Gebäude nur eine Adresse.

Der Beruf eines Handwerkers wurde als Entität gestaltet, um die Berufsbezeichnung getrennt von der Spezialisierung betrachten zu können. Ein Beispiel ist der *Anlagentechniker für Sanitär-, Heizungs- und Klimatechnik* mit der Spezialisierung *Wassertechnik* oder *Wärmetechnik*.

Die Entitäten *Multimedia*, *Schadensart* und *Schadenslage* beschreiben über ihre Relation den *Schaden* genauer. In diesem Zuge wurde die *Schadenslage* und das *Gebäude* getrennt modelliert, um sie besser verwalten zu können bzw. Abfragen leichter nachvollziehen zu können.



Abbildung : ERM der Übung 1.2

# Relationaler Datenbankentwurf

Der relationale Datenbankentwurf leitet sich aus dem ERM in *Abbildung 1* ab. Eine Übersicht zeigt *Abbildung 2*, die das MySQL Modell widerspiegelt.

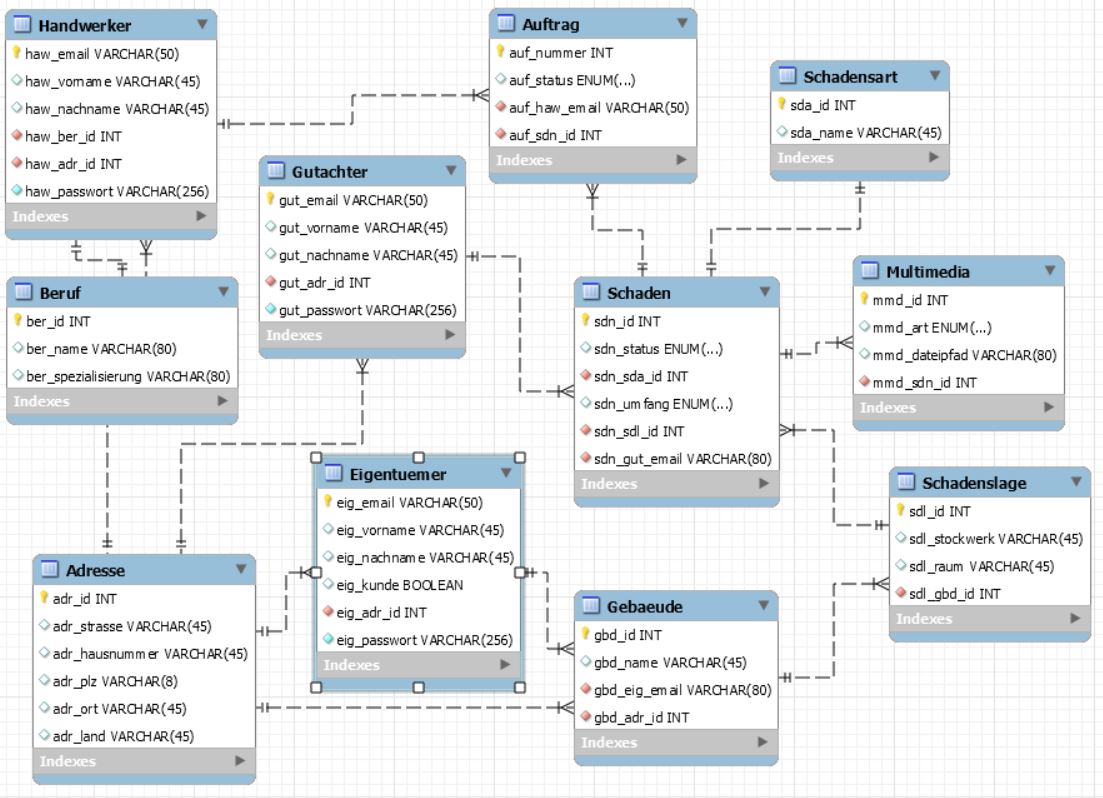


Abbildung : MySQL Modellierung

Die entsprechenden Entitäten mit den zugehörigen Attributen wurden in Beziehung gesetzt und spezifiziert.

Aus dem Modell ergeben sich die konkreten Befehle und Anweisungslisten zur Erstellung der Datenbank. Der SQL-Code zur Generierung der Datenbank inklusive der eingefügten Beispieldaten findet sich in der geforderten .sql‑Datei.

# Konzept und Umsetzung

Konzeptionell wird das Projekt in verschieden Aufgabenbereiche unterteilt:

Es ist eine Logik zur Erstellung von dynamischen Webseiten vorgesehen. Diese Logik ist für die grafische Benutzeroberfläche verantwortlich. Der Nutzer hat im Webbrowser die Möglichkeit zur Interaktion und kommt daher mit dieser Logik direkt in Berührung.

Der Datenbank-spezifische Code regelt die Zugriffe auf die Datenquelle, um die in der Datenbank hinterlegten Beispieldaten für den Nutzer zugänglich zu machen. Dabei handelt es sich z. B. um Methoden, die SQL Befehle ausführen können oder auch die Verbindungen aufbauen können.

Eine dritte Komponente stellen Container dar, die zur Datenübertragung genutzt werden und eine korrekte Datenkapselung sicherstellen sollen.

Die Umsetzung erfolgt in diesem Projekt folgendermaßen:

Java Server Pages (JSP) sind HTML Seiten mit eingebettetem Java Code, die die grafische Benutzeroberfläche erzeugen. Die JSPs haben unter anderem folgende Aufgaben:

Dashboard.jsp ist eine Zusammenfassung der Anlege- und Einsehen-Operationen des Nutzers.

In Index.html sind allgemeine Parameter der HTML-Seite hinterlegt.

Index.jsp liefert das Anmeldeformular für den User.

Loginprocess.jsp besitzt die Logik des Login-Prozesses.

MenueHeader.jsp zeigt den Menü Kopf.

ModifyEntity.jsp gibt dem Nutzer die Möglichkeit Daten der Datenbank zu ändern.

NewEntity.jsp ermöglicht das Anlegen neuer Datensätze.

SeeEntity.jsp ist die Logik für die Visualisierung der Daten.

Die JSP-Seiten greifen auf Data Access Objects zurück, in denen der Datenbank-spezifische Code integriert ist.

JavaBeans werden als Container für die Datenübertragung genutzt.

# Erläuterungen zur graphischen Benutzeroberfläche

Im Beispieldatendatensatz wurden verschiedene Benutzergruppen hinterlegt, die sich mit ihrer E‑Mailadresse und ihrem Kennwort in der Applikation anmelden können. An dieser Stelle seien einige Anmeldedaten aufgeführt:

Handwerker:

|  |  |
| --- | --- |
| E-Mailadresse: | Kennwort: |
| khmeier@wassertechnik.de | khmeier |
| hschmidt@heizungsbau.de | hschmidt |
| lspengler@elektrik-spengler.de | lspengler |
| kmaler@maler.de | kmaler |
| kwerner@dachdecker.de | kwerner |

Gutachter

|  |  |
| --- | --- |
| E-Mailadresse: | Kennwort: |
| hschneider@gutachten.de | hschneider |
| afischer@gutachten.de | afischer |
| jhofmann@gutachten.de | jhofmann |

Eigentümer:

|  |  |
| --- | --- |
| E-Mailadresse: | Kennwort: |
| melanie@wolf.de | melanie |
| lukas@schulz.de | lukas |
| gerda@koch.de | gerda |

In der Datenbank werden die Kennwörter nicht im Klartext gespeichert, da sie so für jedermann direkt ersichtlich wären. Um das zu umgehen, kommt die Hashfunktion SHA zum Einsatz.

Bei der Benutzeranmeldung wird die Hashfunktion auf das Kennwort angewandt und mit dem Datenbankeintrag, der ebenfalls in SHA hinterlegt ist, abgeglichen.

Der User trifft zu Beginn auf folgende grafische Benutzerfläche, bei der er sich mit seiner E‑Mailadresse und seinem Passwort anmelden kann (*siehe Abbildung 3*).



Abbildung 3: GUI mit Anmeldefenster

Das Dashboard gibt je nach Benutzergruppe einen Überblick über die verfügbaren Interaktionsmöglichkeiten (*siehe Abbildung 4*):

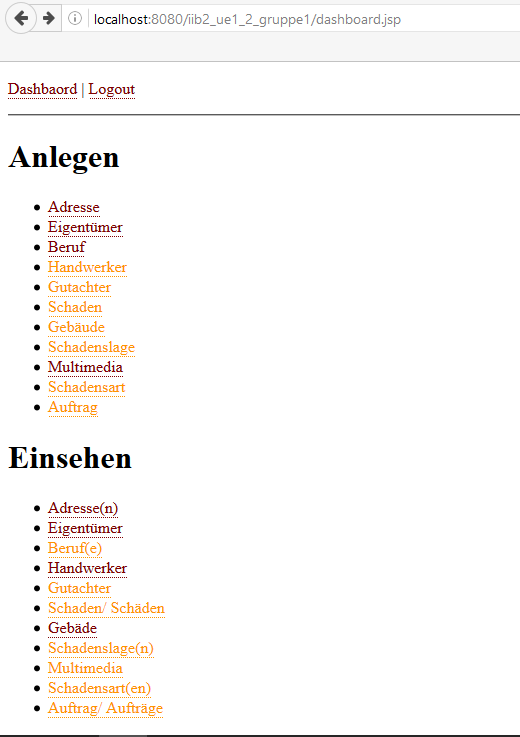


Abbildung 4: GUI mit Dashboard

Die verschiedenen Usergruppen unterschieden sich in ihren Rechten, was das Einsehen, Anlegen oder Bearbeiten von Datensätzen angeht. In *Abbildung 5, 6* und *7* werden die Ansichten der Zugriffe auf die Datenbank visualisiert, die einem Nutzer begegnen können.



Abbildung 5: Einsehen eines Datensatzes



Abbildung 6: Anlegen eines Datensatzes



Abbildung 7: Bearbeiten eines Datensatzes