

Hochschule Ostwestfalen-Lippe
Fachbereich 8 - Umweltingeneurwesen und Angewandte Informatik
Fachgebiet Angewandte Informatik
Programmieren 3
5. Semester WS 2015/16

Dokumentation

Implementierung eines Prototypen für eine Verwaltungssoftware für ein Call-Center

Von

Robin Hake
15306070

und

Benedikt Brüntrup
15306067

Erstprüfer: Prof. Dr. Ralf Hesse
Eingereicht am: 9. Dezember 2015

Inhaltsverzeichnis

| | | |
|----------|--------------------------------|----------|
| 1 | Aufgabenstellung | 2 |
| 2 | Projektplanung | 2 |
| 2.1 | Anforderungsanalyse | 2 |
| 2.2 | Erster Designentwurf | 3 |
| 2.3 | Datenbankentwurf | 3 |
| 2.4 | Architektur | 4 |

1 Aufgabenstellung

Als Abschlussarbeit des Moduls „Programmiersprachen 3“ sieht die Prüfungsordnung die Abgabe eines in JavaEE implementierten Software-Projekts vor. Hierzu ist von den Professor Ralf Hesse eine Liste mit möglichen Abschlussthemen bereitgestellt worden. Von dieser Liste sollte sich ein Thema ausgesucht werden und zwei Anwendungsszenarien des Themas in einen Software-Prototyp umgesetzt werden. Der Software-Prototyp ist dabei auf Basis eines dreischichtigen Ansatz umzusetzen. Die Gruppe Robin Hake und Benedikt Brüntrup entschloss sich das Thema „Verwaltungssoftware für ein Call-Center“ umzusetzen.

2 Projektplanung

Dieser Abschnitt beschreibt, wie bei der Projektplanung vorgegangen wurde. Es wird auf die Anforderungsanalyse, den ersten Designentwurf, den Datenbankentwurf und das gewählte Entwurfsmuster eingegangen.

2.1 Anforderungsanalyse

Zu Beginn der Projektplanung wurde mit Hilfe eines Use-Case-Diagramms ermittelt was das Projekt alles können soll. Die Ermittlungen haben ergeben, dass es einen Administrator und Standardbenutzer geben soll. Der Administrator soll Benutzer verwalten können und die Standardbenutzer sollen mit Tickets arbeiten können. Genauere Details zur Anforderungsanalyse kann den Use-Case-Diagramm der *Abbildung 1* entnommen werden.

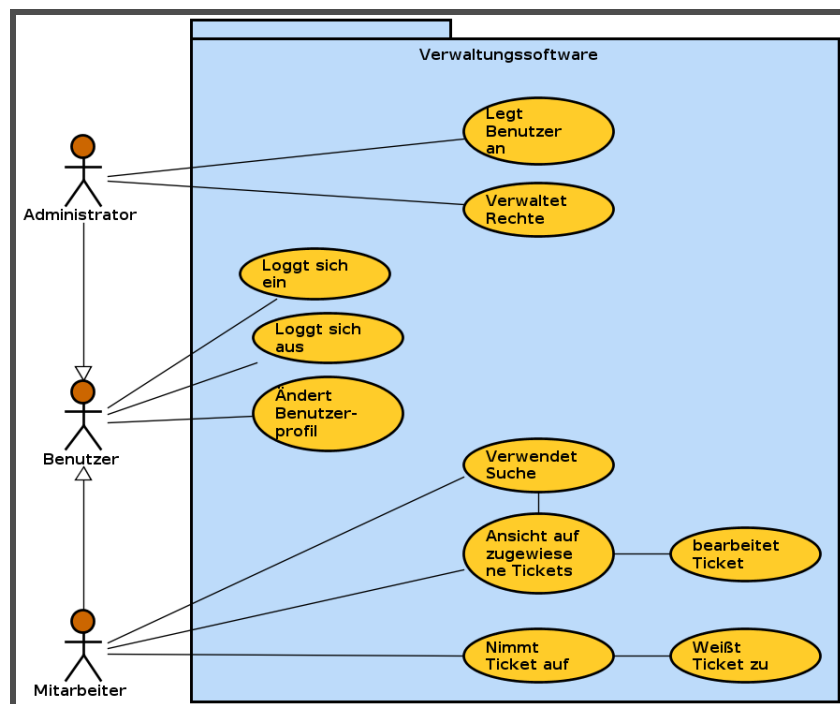


Abbildung 1: Anforderungsanalyse als Use-Case-Diagramm

2.2 Erster Designentwurf

Als nächstes wurde mit dem Programm PowerPoint ein erster Designentwurf für die Weboberfläche des Projektes entworfen. Hierbei wurde sich für ein fensterbasiertes Design entschlossen. Dieses bietet den Vorteil, dass das Projekt einfach erweitert werden kann. Soll das Software-Produkt eine neue Funktion haben kann einfach ein neues Fenster hinzugefügt werden. Die folgende Abbildung zeigt den ersten Designentwurf in PowerPoint.

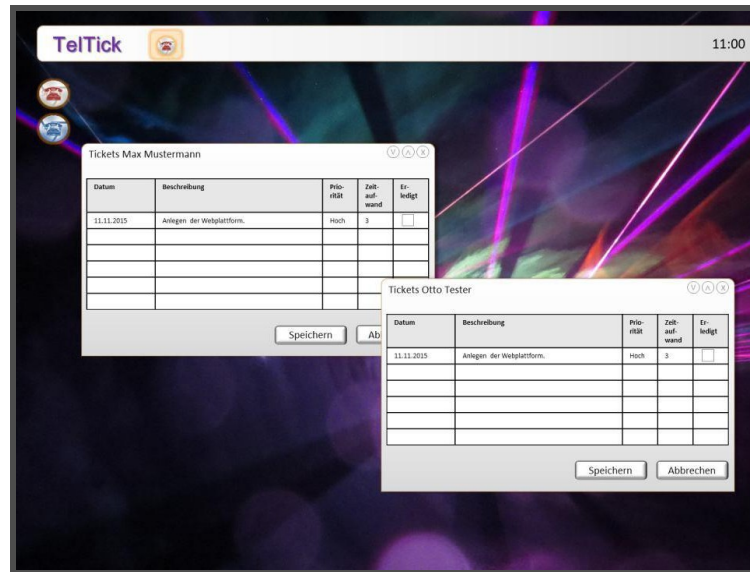


Abbildung 2: Erster Designentwurf in PowerPoint

2.3 Datenbankentwurf

Nachdem das grundlegende Design der Webseite festgelegt worden war, wurde sich über die Datenhaltung Gedanken gemacht. Es wurde ermittelt welche Daten das Software-Projekt persistent speichern soll und wie diese Daten im Zusammenhang stehen. Aus dieser Erkenntnis wurde die in der *Abbildung 3* gezeigte Datenbankstruktur festgelegt.

Die *Tabelle 1* beschreibt wofür welche Relation in der Datenbank benötigt wird.

| Tabellenname | Beschreibung |
|-----------------|---|
| Mitarbeiter | In dieser Relation werden die Mitarbeiter hinterlegt, die sich an der Webseite anmelden können sollen. |
| Rechte | In dieser Relation werden die Fenster eingetragen, auf die ein bestimmter Mitarbeiter Zugriffsrechte haben soll. |
| Ticket | Enthält die erstellten Tickets. |
| Fenster | Enthält die Abmaße, die Titel und die Pfade zu den JSP-Dateien der Fenster, die auf der Webseite angezeigt werden sollen. |
| Ticketzuweisung | Weißt ein Ticket einen bestimmten Benutzer zu. |

Tabelle 1: Beschreibung der einzelnen Relationen

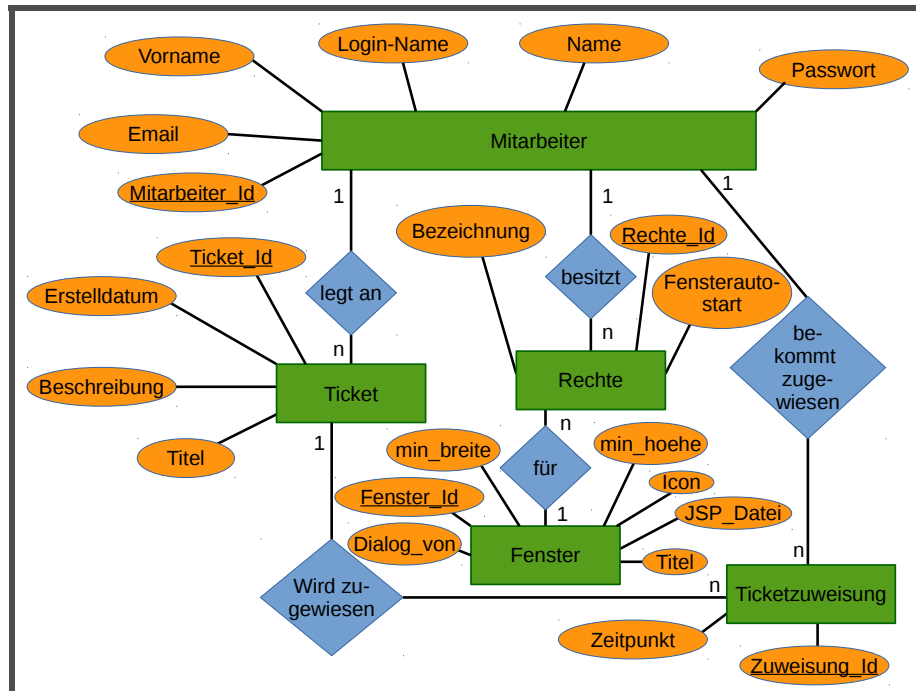


Abbildung 3: Datenbankstruktur

2.4 Architektur

Als Architektur-Modell wurde das MVC-Modell (**M**odel, **V**iew, **C**ontrol) gewählt. Hierbei handelt es sich, wie in der Aufgabenstellung erwünscht, um ein Dreischichtenmodell. Das Use-Case-Diagramm der *Abbildung 4* beschreibt wofür genau welche Schicht zuständig ist.

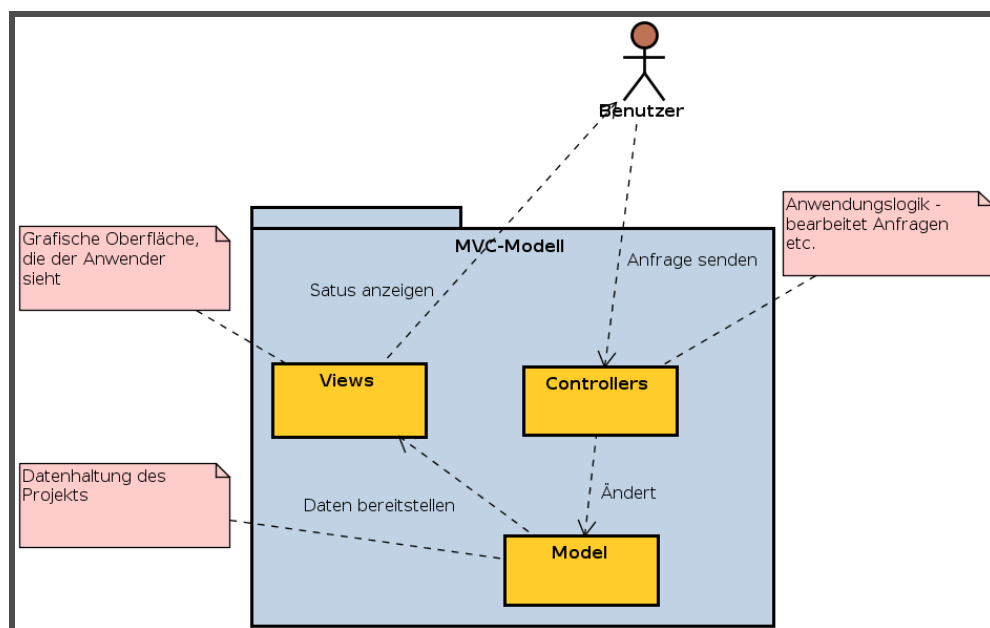


Abbildung 4: MVC-Modell

Die *Abbildung 5* stellt da, wie das MVC-Modell beim Projekt umgesetzt wurde. Die „Views“ wurden im Projekt durch JSP-Seiten umgesetzt, die „Controllers“ durch Servlets und das „Model“ durch DAO-Objekte. Diese DAO-Objekte mappen die Datenbank in normale Java-Objekte, sodass der Anwendungsentwickler wie gewohnt mit Java-Klassen arbeiten kann und nicht im Quellcode direkt Datenbankabfragen senden muss. Dafür bieten die DAO-Objekte einfache „Store“- und „Load“-Methoden. Diese Methoden mappen ein Java-Objekt dann in der Datenbank.

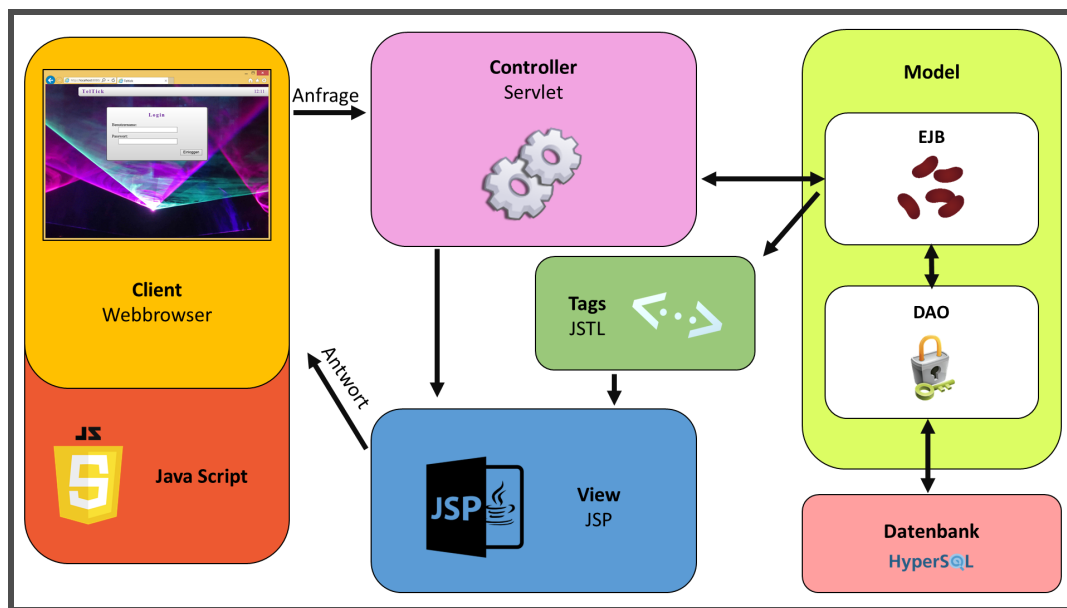


Abbildung 5: Architektur des Projektes