



**Westfälische  
Hochschule**

Gelsenkirchen Bocholt Recklinghausen

# **Drunken-Jukebox**

## **Projektdokumentation**

---

**im Master Studiengang Verteilte Systeme**

vorgelegt von: Daniel Hardes, Dennis Miller,  
Fabian Paus, Christian Schlütter

Modul: Middleware A (MWA)

Gutachter: Prof. Dr. Martin Guddat

Abgabetermin: 5. Januar 2015



---

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Projektidee . . . . .	2
<b>2</b>	<b>Funktionale Anforderungen</b>	<b>2</b>
2.1	Anwendungsfälle . . . . .	2
<b>3</b>	<b>Technische Realisierung</b>	<b>3</b>
3.1	Domänenmodell . . . . .	3
3.2	Architektur . . . . .	3
3.3	Datenmodell . . . . .	3
3.4	Benutzerrollen . . . . .	4
<b>4</b>	<b>Fazit</b>	<b>4</b>



## 1 Einleitung

- Aufgabenstellung: Realisierung einer beliebigen Idee unter Einsatz des Applikationsservers JBoss / Wildfly - Experimentelle Verwendung aller Funktionen u.a. Authentifizierung - Überlegungen zur Projektidee „DrunkenJukebox“

### 1.1 Projektidee

Jukebox: Datenbankgestützte Wildfly-Anwendung Generierung und Verwaltung der Playlist inkl. Song Informationen Jukebox, in der die nächsten Songs angezeigt werden Up / Down Voting der Songs innerhalb der Playlists durch Anwender DI ist der Drunken-Index und beeinflusst die Generierung der Playlist

- musikalische Gestaltung einer Party - Gäste sollen Musik beeinflussen können - In Abhängigkeit des Betrunkenhitsgrades und einem Voting - Algorithmus trifft Vorauswahl von Titeln aus Titelsammlung des Gastgebers - Berücksichtigt Betrunkenhitsgrad und Votings vorangegangener Partys.

Dazu wird den Partygästen eine App zur Verfügung gestellt, über die Songs bewertet werden können. Im Rahmen dieses Projektes soll das Backend der Anwendung „Drunken-Jukebox“ entstehen.

## 2 Funktionale Anforderungen

- Einleitender Satz - Unterteilung der Anwender in: - Gastgeber ist Admin - Gast ist Party People

### 2.1 Anwendungsfälle

Party People: Möchte Playlist und Current Song einsehen können. Kann Songs innerhalb der Playlist voten. Kann seinen DI zurück melden Möchte Verlauf seines DIs abrufen können Möchte seine bisherigen Votings abrufen können Admin: Verwaltet globale Song Liste Kann eine Party starten und beenden

Ein  
schönes  
Use-Case-  
Diagramm



## 3 Technische Realisierung

### 3.1 Domänenmodell

Roles: Admin: Verwaltet die Songs auf dem Server Party People: Voten die Songs Ist nach dem ersten Kontakt eindeutig identifizierbar (Party People ID) Jukebox: datenbankgestützte Wildfly-Anwendung Verwaltung der Songs und Song-Informationen Songs hinzufügen Songs löschen Song-Informationen bearbeiten Generierung und Verwaltung der Playlist Song Hat Song-Information Muss abspielbar sein Pfad in Dateisystem Internetstream YouTube-Link Song-Information: Titel, Interpret, Genre, Album, Länge Current Song: Song der gerade gespielt wird. Wenn Current Song abgespielt ist, wird der erste Song aus der Playlist entfernt und zum Current Song. Playlist: Absteigend sortierte Song Liste auf Basis des Votings Playlist wird durch Playlist-Füll-Prozess generiert: Solange Größe Playlist nicht erreicht, wähle nächsten Song via Song-Selection Song-Selection Nächste Song wird auf Basis von DI, Party Time usw. ausgewählt Aus globaler Song Liste der Jukebox Voting Basiert auf Playlist Erlaubt Party People Songs innerhalb der Playlist zu bewerten: Up-Vote Down-Vote  $Vote\ v = \text{Anzahl Up-Votes} - \text{Anzahl Down-Votes}$  Party People darf pro Song max. einmal voten DI: Beschreibt den Betrunkenhitsgrad der Party People ganzzahliger Wert zw. 0 - 100

### 3.2 Architektur

- Systemübersicht / Komponentendiagramm - Beans und deren Aufgaben

### 3.3 Datenmodell

- Kasten stellen Entity-Beans dar - Pfeile stellen die Beziehungen (Unidirektional, bidirektional) dar - Zwischen den Pfeilen stehen die Cascading Types

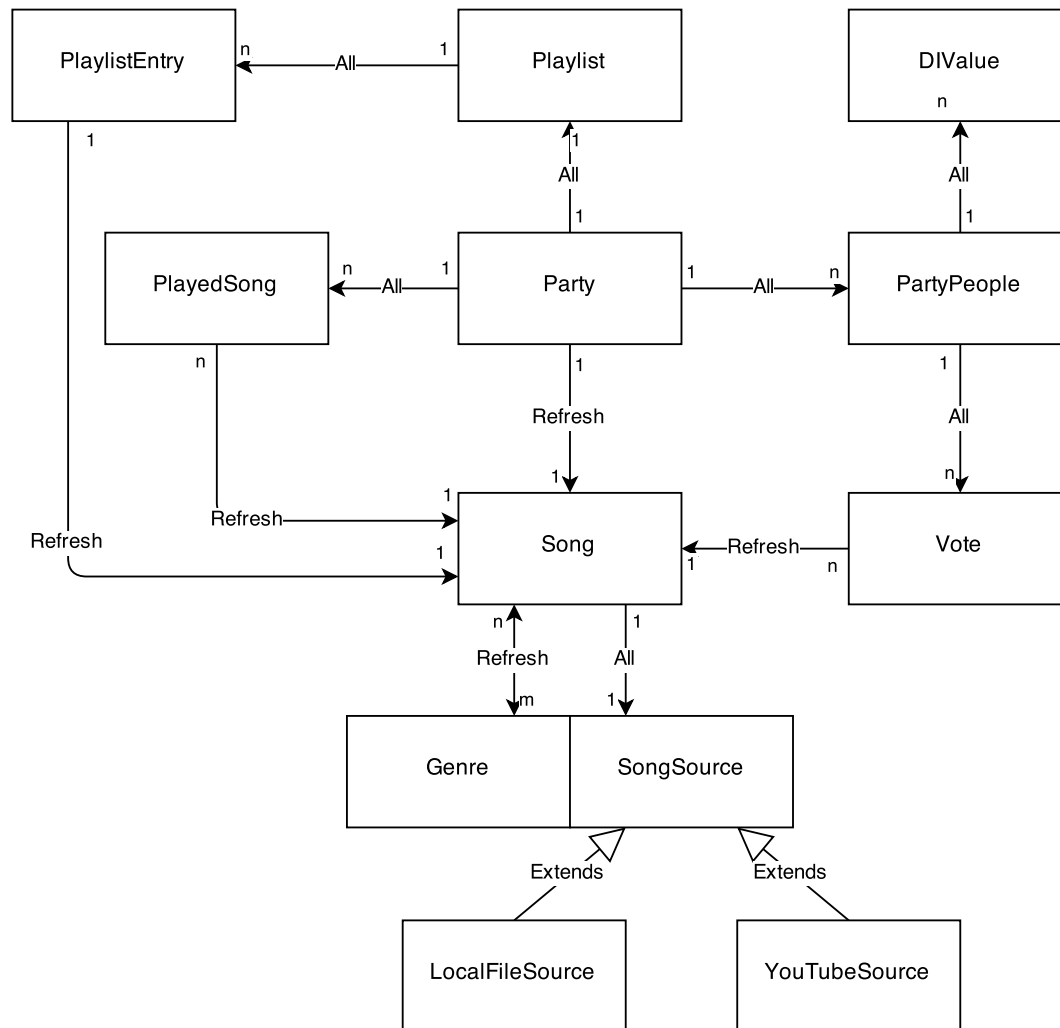


Abbildung 1:

### 3.4 Benutzerrollen

- Authentifizierung via Datenbank im Wildfly - Schritte zur Realisierung - Ausführung der Probleme

### 3.5 Exemplarische Darstellung

- Beschreibung eines Funktionsaufrufes z.B. start Party durch alle „Schichten“ des Systems



## 4 Fazit

- Zusammenfassung - Reflektion (Probleme usw.) - Ausblick

Anhang:  
Doku-  
menta-  
tion der  
API

