

Inhaltsverzeichnis

1. Einleitung	2
2. Systementwurf.....	2
2.1 Entwurfsziele	2
2.2 Paketverteilung.....	2
data:.....	2
database:	2
feature:	2
frontend:.....	2
2.3 Verwendung existierender Softwarekomponenten	3
Hibernate/ JPA:.....	3
JUnit:	3
HSQLDB:.....	3
2.4 Management persistenter Daten	3

1. Einleitung

Das hier vorgestellte System hat den Zweck eine Social-Media-Plattform zur Verfügung zu stellen. Auf dieser Plattform soll es dem Benutzer ermöglicht werden Musik von Künstlern zu hören und seine Lieblingsmusik in Wiedergabelisten zu speichern. Um das System zu nutzen erstellt sich jeder Benutzer ein eigenes Profil. Profile von anderen Benutzern können im System angeschaut werden. Das System stellt eine Follow-Funktion zur Verfügung, über welche der Benutzer über neue Beiträge, wie z.B. neue Ankündigungen, neue Wiedergabelisten oder Alben von Benutzern, Künstlern oder Labels, informiert wird.

Wenn ein Benutzer Musik hochladen will, muss dieser erst zum Künstler ernannt werden. Dazu stellt das System eine Funktion zur Verfügung. Ist er nun Künstler kann er Musik veröffentlichen und für sich werben. Die hochgeladene Musik wird nun im Profil des Künstlers angezeigt. Über die im System vorhandene Datenbank hat jeder Benutzer nun die Möglichkeit sich die Musik anzuhören.

Zusätzlich stellt das System noch Labels zur Verfügung. Diese haben ebenfalls eine eigene Profilseite, auf welcher die im Label zusammengeschlossenen Künstler aufgelistet werden. Ebenfalls werden auf der Labelseite auch die Playlists des Labels angezeigt. Die Label-Profil-Seiten werden von Label-Managern bearbeitet. Nach dem einloggen stellt das System diesen die Funktionen zur Verfügung, welche für das Bearbeiten von Playlists und Künstlern benötigt werden.

Der Label-Manager bekommt aber nicht nur die Funktionen zum Bearbeiten der Label-Profil-Seiten, er kann auch im Namen eines Künstlers Musik für diesen ins System uploaden und dessen bereits in der Datenbank vorhandenen Musikstücke bearbeiten.

Alle im System vorhandenen Daten werden lokal gespeichert und über eine Datenbank verwaltet.

2. Systementwurf

2.1 Entwurfsziele

Textuelle Beschreibung der Entwurfsziele und ihre Priorisierung (+ kurze Begründung für Entscheidungen)

2.2 Paketverteilung

de.glurak.data:

Datapaket enthält alle Entitätsklassen. Diese erfüllen die Model-Funktion für die übrigen Klassen.

de.glurak.database:

Database-Paket kümmert sich darum, dass die Entitätsklassen persistent in einer Datenbank verwaltet werden können.

de.glurak.feature:

Das Feature-Paket beinhaltet Klassen, die sich um einen speziellen Aufgabenbereich kümmern (z. B. Mp3-Player).

de.glurak.frontend:

Das Frontend-Paket enthält Klassen, die sich um die Interaktion mit dem Anwender mittels Swing von Java kümmert und Daten entsprechend aufbereitet.

2.3 Verwendung existierender Softwarekomponenten

Hibernate/ JPA:

Hibernate/JPA kümmert sich sehr transparent darum, dass die Entitätsklassen in eine Datenbank unserer Wahl gespeichert werden können.

Laut Aufgabenstellung müssen die Daten einen Neustart überleben. Da Hibernate empfohlen wurde, wurde es auch benutzt.

JUnit:

JUnit kümmert sich um automatische Tests unserer Anwendung, d.h. wir brauchen nach Änderungen nicht explizit alle Funktionen zu testen, sondern können dies automatisiert ausführen und Fehler somit aufspüren.

JUnit ist das Standard-Tool und wird bereits von vielen IDE's unterstützt.

Mp3-Player (Library: javazoom):

HSQLDB:

HSQLDB ist eine Datenbank, die gänzlich in Java implementiert worden ist.

Da es in Java geschrieben ist und auch die Speicherung im Dateisystem unterstützt, ist es sehr komfortabel in Java zu benutzen.

2.4 Management persistenter Daten

Objekte, die persistent zu halten sind:

Medium, Genre, Playlist, User, UserProfile, AdminProfile, ArtistProfile, LabelmanagerProfile, LabelProfile

Zur Beschreibung/ Begründung des gewählten Persistenzmechanismus siehe 2.3.

2.5. Sonstiges

Beschreibung & Begründung von Entwurfsentscheidungen

Kann entfallen!!!

3. Objektentwurf

3.1. Abwägungen des Objektentwurfs

Textuelle Beschreibung der Überlegungen, die zur Entscheidung für den vorliegenden Objektentwurf geführt haben

3.2. Klassenmodell der Entitätsklassen

UML-Klassendiagramm der überarbeiteten Entitätsklassen und ihrer Beziehungen inkl. Klassenschnittstellen (Attribute, Methoden inkl. Typen, Signatur, Sichtbarkeit).

Die Entitätsklassen sind (in der Regel) die Klassen, deren Objekte persistent gehalten werden

3.3. Dokumentation weiterer interessanter Teile des Entwurfsklassenmodells

jeweils UML-Klassendiagramme für Systeme von Klassen mit interessanten/nicht-trivialen Beziehungen, z.B. bei Verwendung von Entwurfsmustern. Zusätzlich jeweils kurze textuelle Erläuterung des gezeigten Ausschnitts des Klassenmodells.

Anmerkung: Die textuelle Beschreibung der obigen und aller weiteren Klassen des Entwurfsklassenmodells erfolgt mit Javadoc im Java-Quelltext.

4. Glossar

5. Anhang

Javadoc des Codes inkl. Spezifikation nicht-trivialer Verträge und Invarianten (textuell; Formulierung in OCL nicht notwendig); Anhang nur elektronisch abgeben