

## Inhaltsverzeichnis

---

1. Einführung.....	2
2. Vorgeschlagenes System.....	2
2.1. Übersicht: .....	2
2.2. Funktionale Anforderungen.....	2
2.3. Nichtfunktionale Anforderungen.....	4
2.4. Systemmodelle .....	4
2.4.1. Szenarien .....	4
2.4.2. Anwendungsfallmodell: Use case-Diagramm + Use case Beschreibungen .....	7
2.4.3. <i>Statisches Modell</i> :.....	7
2.4.4. Dynamisches Modell: Sequenzdiagramme + Zustandsdiagramm(e) .....	7
3. Glossar .....	7
Anhang A: GUI-Skizzen.....	7

## 1. Einführung

---

Das hier vorgestellte System ist eine Social-Media-Plattform auf der Benutzer Musik von Künstlern hören und in persönlichen Wiedergabelisten speichern können. Jeder Benutzer hat ein eigenes Profil und kann sich die Profile aller anderen Benutzer anschauen. Der Benutzer kann durch eine Follow-Funktion über Beiträge und neue Wiedergabelisten oder Alben von Benutzern und Künstlern informiert werden.

Ein Benutzer kann zum Künstler werden und dann selbst Musik publizieren, verbreiten und für sich werben. Diese wird dann in seinem Profil angezeigt und ist für jeden sicht- und hörbar.

Neben den Benutzer existieren auch sogenannte Labels, die die Profile von den ihnen zugehörigen Künstlern in deren Namen verwalten können. Über diese Funktion hinaus haben Labels ebenfalls ein eigenes Profil wo sie eigene Playlist zusammenstellen und für ihre Künstler werben können.

Jedes Label wird von mindestens einem Manger verwaltet, der dadurch über mehr Funktionalität verfügt als der normale Benutzer. Er kann, wie der Künstler auch, Musik hochladen, in Alben gruppieren und in Genre einteilen.

Alle erzeugten oder hochgeladenen Daten werden vom System auf einem lokalen Server konsistent gespeichert.

Im Hintergrund des Systems existiert eine Kontrollinstanz in Form von realen Administratoren. Ihre Aufgabe ist es Beschwerden zu bearbeiten und die Authentizität von Künstlern, Labels und Label-Managern mit Hilfe eines Drittsystems zu verifizieren.

## 2. Vorgeschlagenes System

---

### 2.1. Übersicht:

Kurze textuelle Beschreibung

### 2.2. Funktionale Anforderungen

*Benutzerfreundlichkeit*

- ~ Kenntnisstand des Benutzers:  
Er ist in der Lage Anwendungen zu bedienen.
- ~ Normen für die Benutzerschnittstelle:  
Wir haben in der Vorlesung Normen jeglicher Art nicht behandelt.
- ~ Welche Dokumentation soll dem Benutzer übergeben werden:  
Eine Video Einführung in das Programm.

*Zuverlässigkeit, Verfügbarkeit, Robustheit*

- ~ Wie zuverlässig, verfügbar, robust soll das System sein:  
Das System soll nach Möglichkeit Abstürze vermeiden.
- ~ Ist ein Systemneustart im Falle eines Fehler akzeptabel:  
Nein.
- ~ Wie viele Daten darf das System verlieren:  
Keine bereits gesicherten Daten.
- ~ Wie soll das System mit Ausnahmen umgehen:  
Ausnahmen sollen abgefangen und dem Benutzer als Dialog angezeigt werden.

- ~ Gibt es Betriebssystem Anforderungen:  
Das System soll auf einem Computer funktionsfähig sein.
- ~ Gibt es Datenschutzanforderungen:  
Der Zugriff darf nur durch eingeloggte (insbesondere registrierte) Benutzer erfolgen dürfen. Des Weiteren sollen die persönlichen Passwörter verschlüsselt gespeichert werden.

#### *Leistungsanforderungen*

- ~ Wie schnell muss das System reagieren:  
Es sollte so schnell wie möglich reagieren.
- ~ Gibt es zeitkritische Benutzeraufgaben:  
Nö.
- ~ Wie viele Anwender soll das System gleichzeitig unterstützen:  
Eine Person.
- ~ Welche Latenz wird noch akzeptiert:  
Eine Latenz von 42.
- ~ Wie groß sind Daten Speicher vergleichbarer Systeme:  
Petabyte.

#### *Unterstützungen*

- ~ Welche Erweiterungen sind geplant:  
Keine.
- ~ Wer wartet das System:  
Die Administratoren (bis auf weiteres die Gruppenmitglieder).
- ~ Gibt es Pläne zur Portierung auf andere Systeme:  
Nein.

#### *Implementierung*

- ~ Gibt es Beschränkungen auf gewisse Hardware- und oder Softwareplattform  
Java 7
- ~ Gibt es Beschränkungen durch die Wartungsarbeitsgruppe  
Loginsperre während der Wartung
- ~ Gibt es Beschränkungen durch die Testarbeitsgruppe  
Nein, es wird ausschließlich in Testumgebung getestet

#### *Schnittstelle*

- ~ Soll das System mit anderen bereits vorhandenen Systemen interagieren  
Nein
- ~ Wie werden Daten in das System exportiert/ aus das System exportiert  
Mit Hilfe einer Datenbank und dem Dateisystems.
- ~ Welche Normen sollen vom System unterstützt werden:  
Wir haben in der Vorlesung Normen jeglicher Art nicht behandelt.

#### *Betrieb*

- ~ Wer kümmert sich um das laufende System:  
Die Administratoren (bis auf weiteres die Gruppenmitglieder).

*Installation*

- ~ Wer installiert das System:  
Der Benutzer.
- ~ Wie viele Installationen sind geplant:  
Eine.
- ~ Gibt es zeitlich Beschränkungen für die Installation:  
Nein.

*Rechtliches*

- ~ Wie soll das System lizenziert werden:  
Es wird keine Lizenzierung erforderlich sein.
- ~ Wer trägt die Verantwortung bei einem Systemfehler:  
Keiner.
- ~ Werden durch die Benutzung bestimmter Algorithmen oder Komponenten Lizenzen fällig:  
Nein.

## 2.3. Nichtfunktionale Anforderungen

### Textuelle Beschreibung der relevanten nichtfunktionalen Anforderungen

## 2.4. Systemmodelle

### 2.4.1. Szenarien

Anwendungsfall Listener:

*Anwendung starten*

- ~ User lädt die Anwendung herunter, installiert und führt sie aus.

*Registrierung*

- ~ User registriert sich über das Registrierungsformular und gibt dabei seine persönlichen Informationen (Password, Benutzer Name, etc...) an.

*Einloggen/Abmelden*

- ~ User logt sich mit seinen Login Daten ein.

*Suche Medium nach Kriterium XY*

- ~ User gibt in die Suchleiste ein Kriterium (Genre, Band, Label) ein und erhält eine Liste mit Ergebnissen. Aus diesen Ergebnissen kann er nacheinander eins auswählen und anschließend betrachten.

*Suche von Benutzern*

- ~ User gibt einen anderen Benutzernamen oder eine andere Email Adresse in das Suchfeld ein und erhält eine Liste mit übereinstimmenden Benutzerprofilen.

*Playlist erstellen*

- ~ Der User wählt die Funktion: neue Playlist erstellen. Diese wird unmittelbar darauf benannt. anschließend kann der User beliebige Titel der wiedergabeliste hinzufügen. Gespeichert wird automatisch.

*Playlist bearbeiten*

- ~ der User fügt einen beliebige Titel der wiedergabeliste hinzu. Gespeichert wird automatisch.

*Playlist löschen*

- ~ der User lässt sich die Wiedergabeliste anzeigen und wählt einen Titel oder mehrere Titel aus, die/den er gerne löschen möchte und bestätigt die Sicherheitsabfrage. gespeichert wird automatisch.

*Persönliche Daten bearbeiten/...*

- ~ der User wählt die Maske „profilbearbeiten“ und verändert den Inhalt der Textfelder, die sein Profil beschreiben/darstellen.
- ~ nach dem erfolgreichen speichern durch Bestätigung wird das veränderte Profil gespeichert und ist nun online für alle anderen User verfügbar.

*Abspielen von Titeln/Playlists/*

- ~ Der User wählt eine Playlist aus und wählt die abspielen-Funktion. Das System spielt nun die Wiedergabe liste ab.

*Interagieren mit anderen Benutzer/Informationen teilen*

- ~ User sucht nach Olaf und sieht sich sein Profil an. da er ihm als sehr sympathisch erscheint möchte er ihn als Freund hinzufügen und klickt auf den entsprechenden Button. Nun ist er Freund von Olaf.

*Vorschläge für Titel/Playlists/Benutzer/ unterbreiten*

- ~ Der User schickt Olaf eine private Nachricht und verweist auf den Titel/Playlist/Benutzer in Form eines links.

*Bewertungen abgeben an Titel/Künstler/Label*

- ~ Der User wählt die Bewerte-Funktion für einen Titel/Künstler/Label aus und wählt eine Bewertungsstufe. Das System speichert die Bewertungen.

## Anwendungsfall Künstler

## Künstlerbewerbung:

- ~ Künstler als angemeldet als Listener und will Künstler werden.
- ~ Künstler tritt durch ein Dritt-System (oder durch System selbst) in Kontakt mit den Admin. Admin prüft Anfrage und ernennt ihn zum Künstler, falls Prüfung angemessen.

*Musik hochladen:*

- ~ Künstler will ein Titel hochladen. Künstler loggt sich ein und klickt auf Hochladen. Dort gibt er die Metadaten ein und wählt die MP3-Datei und lädt sie hoch.

*Labelbewerbung:*

- ~ Künstler will zum Label Olabel. Er loggt sich ein, sucht nach dem Label und klickt auf Bewerben. Er wartet einige Zeit und erhält vom LabelManager ein Ablehnung bzw. eine Bestätigung.

*Metadaten ändern:*

- ~ Künstler war zu dumm um Metadaten richtig einzugeben. Deshalb loggt er sich ein und wählt die zu ändernde Musik aus, klickt auf bearbeiten und gibt die richtigen Daten ein und speichert diese, bzw. verwirft falls diese wieder falsch sind.

*Für einen neuen Song werben:*

- ~ Der Künstler will einen tollen Song kreieren. Um sich Druck zu machen wirbt er schon mal aus der Plattform. Dazu loggt er sich ein. Klickt unter seinen Profil auf werben und schreibt sein Text. Danach klickt er auf Speichern. Sogleich ist dieser Text für alle einsehbar.

*Einladung von einem Label:*

- ~ Der Künstler loggt sich ein. Er sieht unter seinem Profil eine Nachricht vom Label. Dort wird er eingeladen in das Label einzutreten. Der Künstler entscheidet sich und klickt entweder auf "Annehmen" oder auf "Ablehnen".

*Neues Genre anlegen:*

- ~ Der Künstler hat ein total neues Genres (bzw. Untergenres) erfunden und will diese zu seinem Song hinzufügen. Er geht dazu (nach dem Einloggen) auf einen noch nicht spezifizierten Ort und fügt dort ein neues Genre ein und speichert diese. Dann fügt er das Genre den Song hinzu und speichert das.

*Album anlegen:*

- ~ Der Künstler will neues Album anlegen. Dazu loggt er sich ein, geht auf "neues Album anlegen". Es erscheint ein neuer Dialog. Dort kann er die Songs auswählen und hinzufügen. Dazu kann er ein Cover oder sonstige Daten hinzufügen. Dann speichert er sein Album und sofort kann jeder Listener das Album suchen oder sonstiges damit anstellen.

## Anwendungsfall Label-Manager

*Künstlerbewerbung bearbeiten:*

- ~ Der Label-Manager Olaf geht zur Arbeit und loggt sich dort in das Programm ein. Dort erhält er eine Nachricht, dass sich ein Künstler beworben hat, um den Label beizutreten. Olaf liest sich die Bewerbung durch und hört sich die Songs des Künstlers an. Er findet den Künstler gut und nimmt ihn in sein Label auf. Olaf verkündet dann auf der Label Seite, dass das Label einen neuen Künstler aufgenommen hat und aktualisiert das Label Profil und listet dort den neuen Künstler auf. Dann erstellt Olaf eine neue Playlist mit den Titel des Künstlers. Weil Olaf sonst keine Benachrichtigungen hat, loggt er sich wieder aus.

*Künstler werben:*

- ~ Olaf geht zur Arbeit und loggt sich dort in das Programm ein. Er sucht nach einen neuen Künstler von dem er gehört hat. Er schaut sich das Profil von Mace Ventura an und hört sich die Songs von ihm an. Weil er sie so gut findet, schreibt Olaf eine Nachricht an Mace Ventura und wirbt ihn an, um sein Label beizutreten. Weil Olaf sonst nichts anderes zu tun hat, loggt er sich wieder aus und macht Feierabend.

## Anwendungsfall Administrator

*Rechte vergeben*

- ~ Admin Olaf meldet sich geschwind im System an. Das System bietet ihm eine Liste der bereits verifizierten Bewerbungen an. So drauf macht sich Admin Olaf, eifrig wie er ist, dran alle Bewerber auf ihren neuen Status zu erheben. Alle Bewerber auf der Liste haben nun neue Rechte.

*Account Sperren*

- ~ Admin Olaf ist bereits angemeldet, hungrig auf Arbeit macht er sich daran alle Störenfriede ausfindig zu machen. Also schaut er gespannt seine Beschwerdeemails durch. Er entdeckt einen User der unzüchtige Inhalte verbreitet. Er besucht das Profil des Users um die Anschuldigung zu verifizieren. Fündig geworden, entschließt er sich den Nutzer zu sperren. Das System fordert Olaf auf, dem besagten Benutzer eine Nachricht, im besten Falle eine Begründung für die Sperre zu hinterlassen. Bei Bestätigung erhält der Benutzer eine E-Mail mit der Benachrichtigung, dass er gesperrt wurde, dem Datum der Sperre und Olafs Begründung. Der Zugang des Users ist nun für unbestimmte Zeit gesperrt. Olaf, glücklich über die Tatsache, die Welt wieder einmal ein wenig besser gemacht zu haben, meldet sich vom System ab und geht einen Kaffee trinken.

2.4.2. Anwendungsfallmodell: Use case-Diagramm + Use case Beschreibungen

2.4.3. Statisches Modell:

Klassendiagramm für Entitätsklassen + Klassenbeschreibungen für Entitäts-, Grenz- und Kontrollklassen

2.4.4. Dynamisches Modell: Sequenzdiagramme + Zustandsdiagramm(e)

## 3. Glossar

---

Lexikonartige Auflistung und Kurzerklärung wichtiger Begriffe

## Anhang A: GUI-Skizzen

---

GUI-Skizzen der zentralen Grenzklassen