

Übung 2

185.A01 Objektorientierte Programmiertechniken

Goran Filcic

Matr. Nr.: 1025112

e1025112@student.tuwien.ac.at

Manuel Schmitt

Matr. Nr.: 1127688

e1127688@student.tuwien.ac.at

Peter Nirschl

Matr. Nr.: 1025647

e1025647@student.tuwien.ac.at

19. Oktober 2012

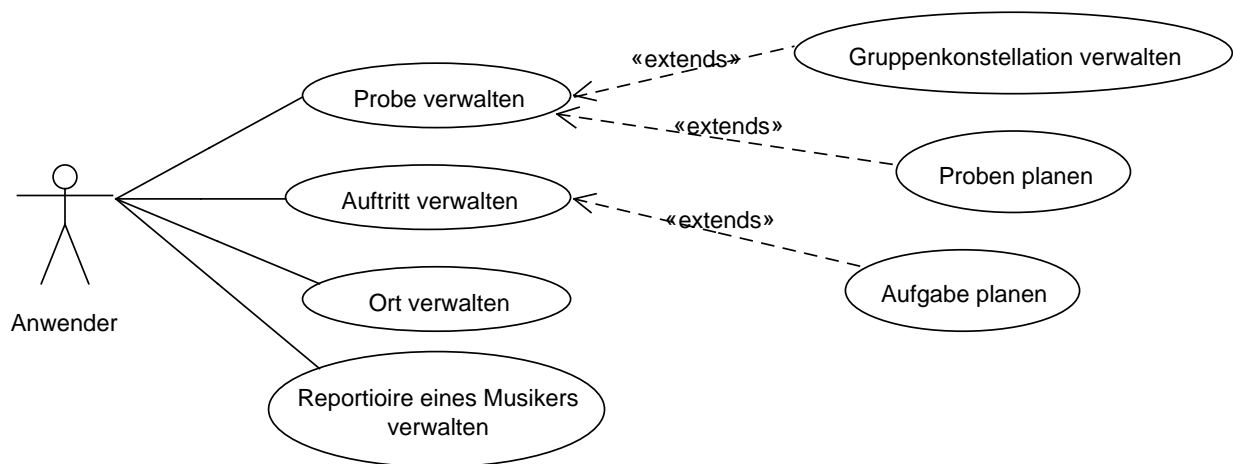
Inhaltsverzeichnis

1	Einleitung	2
2	Analyse der Anwendungsfälle	2
3	Beschreibung der Anforderungen	2
3.1	Annahmen und Voraussetzungen	2
3.2	Allgemeine Beschreibung	2
3.3	Detaillierte Beschreibung	2
4	Details zur Umsetzung	3
4.1	Verteilung der Verantwortung	3

1 Einleitung

Die bestehende Codebase wird von der Gruppe erweitert, sodass es möglich sein wird mit dem Programm effektiv zu arbeiten.

2 Analyse der Anwendungsfälle



3 Beschreibung der Anforderungen

3.1 Annahmen und Voraussetzungen

1. Das System unterscheidet (noch) nicht zwischen verschiedenen Benutzergruppen.
2. Das System wird zu einem Zeitpunkt von genau einem Benutzer verwendet.

3.2 Allgemeine Beschreibung

Folgende Funktionalitäten sind zu implementieren (eine genauere Beschreibung erfolgt weiter unten):

1. Persistenz der Daten
2. Unterscheidung zwischen vergangenen (bestätigten) Ereignissen und zukünftigen Ereignissen
3. Verwaltung von Proben und Auftritten (inklusive Historisierung der Daten)
4. Verwaltung von Orten

3.3 Detaillierte Beschreibung

Persistenz der Daten

Der Zustand der Entitäten soll nach der Manipulation des Datenbestandes persistiert werden. Dazu wird die Klasse `Musikgruppe` dahingehend erweitert, dass sie alle Unterobjekte in einer XML-

Struktur abspeichert. Die Unterobjekte müssen ebenfalls erweitert werden. Damit die Objekte alle gleich strukturiert sind, wird ein Interface **IPersistent** eingeführt:

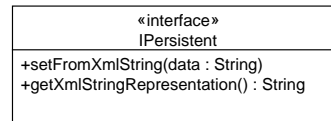


Abbildung 1: Definition der IPersistent Schnittstelle

Vergangenheit und Zukunft

Die Entität **Ereignis** wird so erweitert, dass eindeutig feststellbar ist ob das Ereignis bestätigt ist oder nicht. Bestätigte Ereignisse werden aus Performance-Gründen getrennt von zukünftigen (planbaren) Ereignissen im Speicher gehalten.

Verwaltung von Proben und Auftritten

Proben und Auftritte können verändert und entfernt werden. Veränderungen müssen nachvollziehbar im System abgebildet sein.

Pro Probe (bzw. Auftritt) kann eine eigene Konstellation an Musikern angegeben werden. Die Musiker werden aufgeteilt in permanente Mitglieder und Ersatzmitglieder.

Verwaltung von Orten

Orte werden als zusätzliche Entität im System abgebildet und können von Benutzern verwaltet werden. Zur Nachvollziehbarkeit werden Orte nicht gelöscht. Alle Änderungen an Orten müssen nachvollziehbar sein.

4 Details zur Umsetzung

4.1 Verteilung der Verantwortung

Die Verantwortung zur Umsetzung

Mitglied	Arbeitspakete
Goran Filcic	Trennung vergangener u. zukünftiger Ereignisse
Manuel Schmitt	Verwaltung von Repertoire, Orten und Gruppenkonstellationen
Peter Nirschl	Persistenz d. Daten