Dokumentation Fallbeispiel

DVD Archiv

Jasmin Thevathas, Marc Hutzli, Robin Schmid
JARCHITECTS

29. Januar in Burgdorf

Inhaltsverzeichnis

1 Einführung

Dieses Dokument dient zum besseren Verständnis unseres Projekts respektive Fallbeispiels im Modul „Java Soft. Entwicklung mit Open Source 1“.

1.1 Idee

Wir wollten als Projekt ein DVD-Verleihsystem umsetzen. Dabei soll es Usern möglich sein, eigene DVDs zu erfassen, welche man verleihen möchte. Diese DVDs lassen sich einem Film zuordnen. Andere User wiederum können diese DVDs ausleihen und schliesslich, wenn sie den Film gesehen haben, ein Review (Beurteilung) verfassen. Diese wird wiederum im System dargestellt. Falls von einem Film gerade keine DVD frei verfügbar ist, kann der User eine reservieren.

Der Nutzen ist, dass User einen Überblick über die Filmsammlung haben und andererseits neue Streifen entdecken könne, die sie noch nicht besitzen. Dank des Reviews können sie sich bereits ein Bild darüber machen, was andere über den Film denken und ob sich eine Ausleihe lohnt.

1.2 Methodik

Das Exempel diente nicht nur zur Adaption der im Unterricht behandelten Themen, sondern auch, um die Arbeit und Kommunikation im Team zu schulen. Zuerst wurde ein Domain Model erstellt, welches aber laufen angepasst wurde. Auch die spätere Implementation erfuhr mehrere Lebenszyklen. So wurde nicht nur eine im Unterricht erarbeitete Lösung abgeben, sondern man traf sich auch noch ausserschulisch und kann schliesslich eine Umsetzung mit SpringBoot vorweisen.

2 Domain Model

Das Domain Model wurde während der Entwicklung mehrfach angepasst. In Abbildung 2.1 ist die erste und ursprüngliche Version zu sehen. In Abbildung 2.2 ist die schliesslich umgesetzte Version zu betrachten.

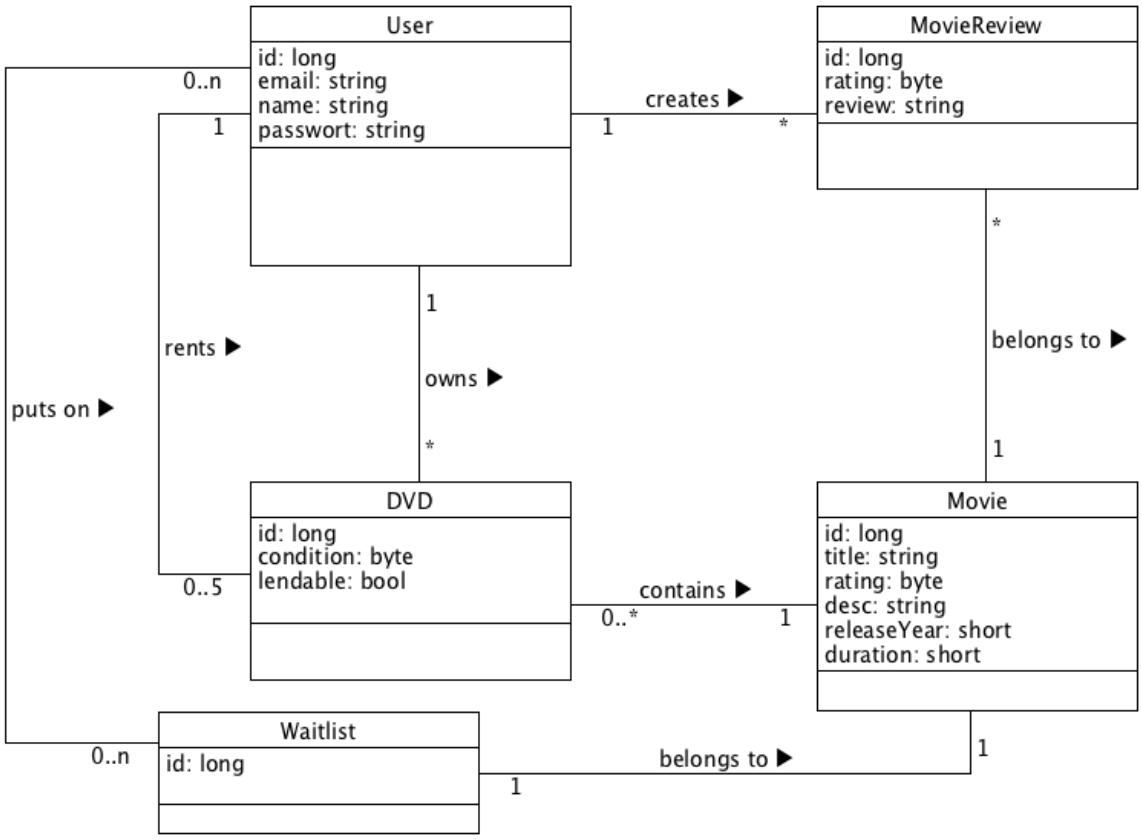

Abbildung 2.1: Erste Version des Domain Models

2.1 Entitäten

User

Beschreibt einen Benutzer des Systems. Er besitzt einen Namen, E-Mail Adresse und ein Passwort. Ausserdem kann er Besitzer (*owns*) von mehreren **DVD**s sein. Zudem kann ein User bis zu 5 DVDs ausgeliehen haben (**RentRecord** pro Ausleihe). Weiter

kann ein User einen Anspruch auf einen Film erheben, falls gerade kein DVD „frei“ ist. Dann entsteht ein **ReservationRecord**. Schliesslich kann ein User Beurteilungen über Film erfassen, welche dann als **MovieReview** modelliert sind.

DVD

Stellt ein Model eines DVDs dar, mit einem Zustand (*condition*), welcher aussagt, wie gut die DVD erhalten ist (neu, verkratzt, etc). Schliesslich kann noch erfasst werden, ob der DVD verleihbar ist (*lendable*). Ein DVD lässt sich jeweils einem Film zuordnen.

Movie

Diese Entity stellt einen Film dar und besitzt Attribute für Titel, Länge, Rating (FSK, PEGI), Veröffentlichungsjahr und Beschreibung. Ein Movie besitzt Beurteilungen von Usern (**MovieReview**) und es kann von einem Movie mehrere **DVD**s geben. Ausserdem besitzt jeder Movie **ReservationRecords**) für alle User, die auf den Film warten, um eine DVD davon auszuleihen.

MovieReview

Wird von Usern zu einem bestimmten Film erstellt, besitzt ein *Rating* von 1 - 10, wie gut dem User der Film gefallen hat sowie ein kurzes *Review*, wo der User einen Kommentar verfassen kann.

RentRecord

Stellt eine Zwischen-Entität dar, die abbildet, welcher User welchen **DVD** ausgeliehen hat.

ReservationRecord

Ist ebenfalls eine Zwischen-Entität, die darstellt, welcher User auf welchen Film wartet, bzw. für welchen Film er einen DVD reserviert hat.

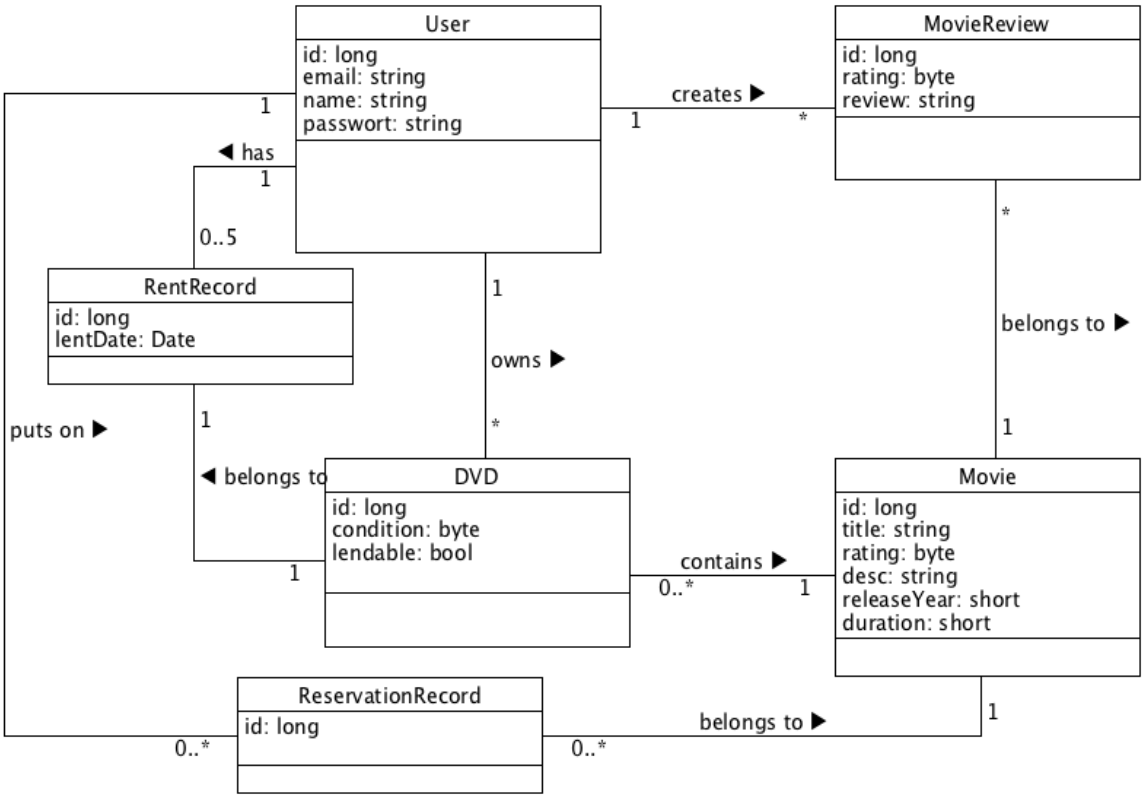

Abbildung 2.2: Finale Version des Domain Models

3 Vorgehen

3.1 Arbeit am Domain Model

Bei Domain Model haben wir uns verschieden Gedanken gemacht. Beispielsweise wurde die Vermietung zuerst als Beziehung zwischen User und DVD modelliert. Somit hätte eine DVD ein Datenbankfeld mit *userID* gehabt, welches entweder gesetzt gewesen wäre (verliehen) oder **null** wenn der DVD noch frei ist. Dies hätte jedoch sehr viele Nullwerte in der Datenbank zur Folge gehabt. Deshalb wurde diese Beziehung explizit als Entität dargestellt.

3.2 Arbeit am Projekt

Gemäss den vorgaben im Unterricht wurde das Projekt zu erst mit Spring alleine umgesetzt. Nach der Lektion „Spring Boot“ entschieden wir uns dann, dieses Framework zu verwenden, da wir auch künftige Projekte in dieser Grössenordnung damit umsetzen würden. Die erste Version war zwar lauffähig im HAL-Browser, da wir aber alle auf unterschiedlichen Branches herumpröbelten, kamen wir nicht zu einer Konsenslösung. Somit begannen wir bei der Abgabe bei Null und konnten somit gleich alle gemachten Lehren in die Neuumsetzung einfliessen lassen.

3.3 Probleme

Die Probleme betrafen unter anderem die **H2**-Datenbank. Wir hatten nicht alle den gleichen Arbeitsstand, unter anderem auf Grund der verschiedenen git-Branches. Als Lösung inkludierten wir die Files im Projekt und verweisen in den *application.properties* direkt auf das lokale File.

Ein weiteres Problem tauchte auf mit gewissen unnötigen Zwischentabellen, welche das Framework automatisch erstellte. Dies konnten wir beheben, in dem wir bei den Properties in der Annotation den Parameter *mappedBy* hinzufügten.

4 Setup

4.1 git Repository

Das Repository des Projektes ist zu finden unter:

https://github.com/tjasmin21/ch.bfh.jarchitects.filmbiblio.boot

Der aktuelle Entwicklungsbranch ist *jasmin*.

4.2 Setup-Anleitung

1. mkdir dvdarchiv && cd dvdarchiv
2. git clone *repo-link* .
3. git checkout jasmin
4. mvn install
5. Projekt in IDE öffnen und runnen
6. localhost:8080 im Browser öffnen
7. Bier öffnen!