

# Architektur-Beschreibung: Bookstore

Michael Schoderer

19 Januar 2016

# Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>3</b>
1.1	Beschreibung der Anwendung . . . . .	3
1.2	Architektur der Anwendung . . . . .	3
<b>2</b>	<b>Komponenten</b>	<b>4</b>
<b>3</b>	<b>Package-Struktur</b>	<b>5</b>
<b>4</b>	<b>Datenbank</b>	<b>7</b>
<b>5</b>	<b>Webservices</b>	<b>9</b>
5.1	REST-API . . . . .	9
5.2	SOAP-API . . . . .	9

# Abbildungsverzeichnis

2.1	Darstellung der Komponenten . . . . .	4
3.1	Packetstrukturdiagramm . . . . .	5
4.1	Datenbankdiagramm . . . . .	7

# Kapitel 1

## Allgemeines

### 1.1 Beschreibung der Anwendung

Die Java-EE Anwendung 'Bookstore' wurde für die Vorlesung Implementierung von Informationssystemen an der Technischen Hochschule Ingolstadt entwickelt und ermöglicht das Speichern und Verwalten von Dokumenten (PDF, MOBI, TXT, ...).

Die Anwendung ist unter der folgenden URL erreichbar: <http://mschoderer.de:8080/bookStore>

### 1.2 Architektur der Anwendung

Bei der Konzeption und Entwicklung wurde die Anwendung in die im folgenden aufgelisteten Schichten unterteilt:

- **GUI** Die Oberfläche wurde mithilfe von JavaServer Faces (JSF) umgesetzt. Bei der Gestaltung der Oberfläche wurden Templates eingesetzt, damit ein einheitliches Bild der einzelnen Seiten gewährleistet werden kann
- **Geschäftslogik** Die Geschäftslogik wurde unabhängig von der graphischen Oberfläche entwickelt, um eine Trennung nach dem MVC-Model zu erreichen
- **Datenbank** Die Datenbankschicht wurde durch mehrere Interfaces gekapselt und stellt die benötigten Funktionen bereit, um die Domain-Objekte in die jeweilige Datenbank zu speichern

# Kapitel 2

## Komponenten

Die nachfolgende Abbildung 2 stellt die Komponenten der Anwendung und deren Beziehungen zueinander dar.

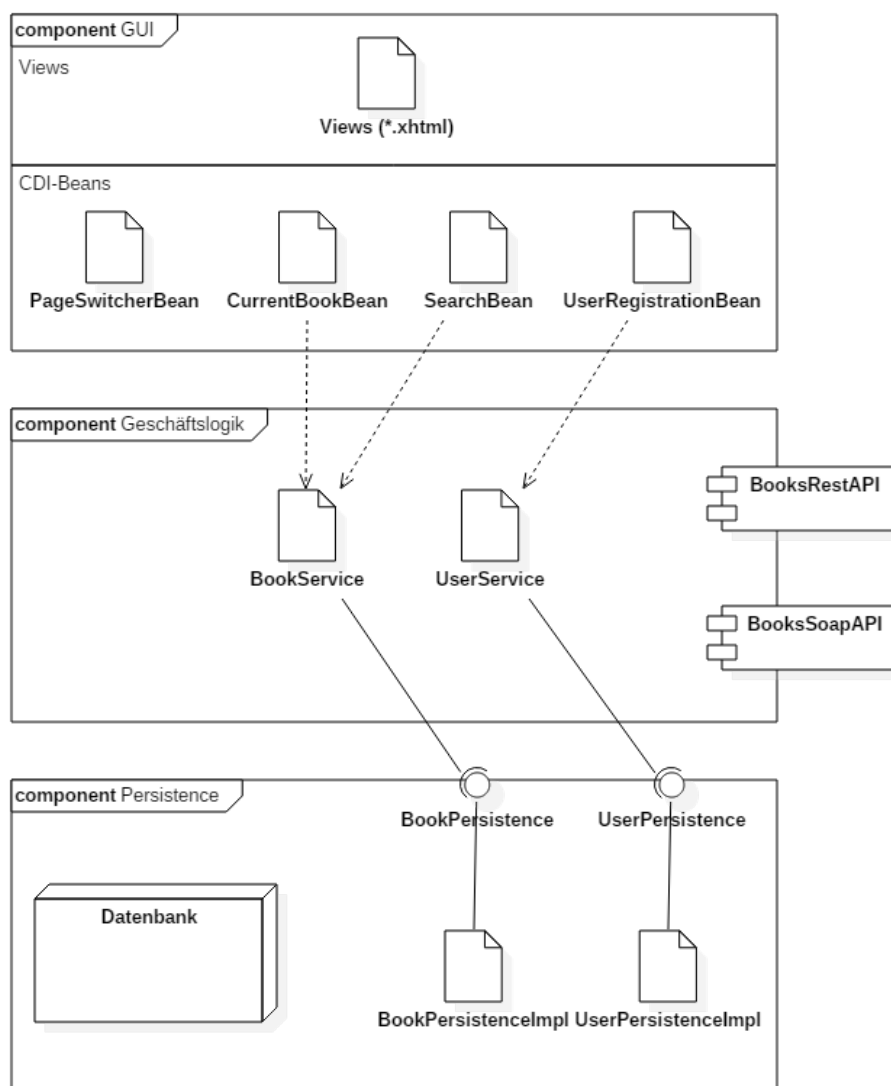


Abbildung 2.1: Darstellung der Komponenten

# Kapitel 3

## Package-Struktur

Die folgende Abbildung 3 zeigt die Struktur der Packages der Anwendung.

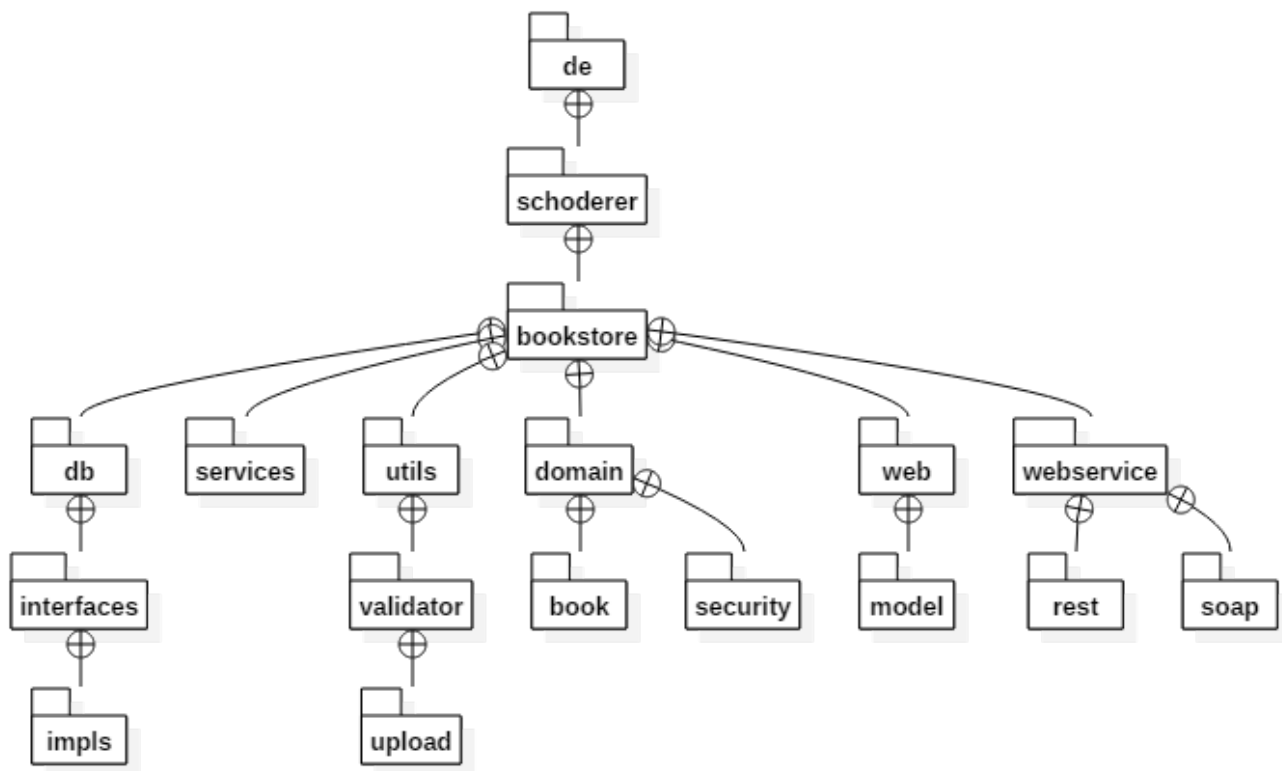


Abbildung 3.1: Darstellung der Packetstruktur der Anwendung

### Beschreibung der Packages

#### **bookstore**

Dies ist das Hauptpackage der Anwendung

#### **db.interfaces**

Hier liegen die Interfaces, welche die Datenbankzugriffe abkapseln

#### **db.interfaces.impls**

Hier liegen die Implementierungen der Datenbankzugriffsinterfaces

**services**

In diesem Package befinden sich die Service-Klassen, welche die Geschäftslogik der Anwendung beinhalten

**utils.validator**

In diesem Package und dem Unterpacket *upload* befinden sich die JSF-spezifischen Validatoren, welche die Eingaben der Benutzer validieren

**domain.book**

Hier befinden sich die Domänenobjekte, welche für den Betrieb der Anwendung benötigt werden

**domain.security**

Hier befinden sich die Domänenobjekte, welche die Benutzer und deren Rechte repräsentieren

**web.model**

In diesem Packet befinden sich die CDI-Beans

**webservice.rest**

Hier befinden sich die Klassen, welche benötigt werden um die REST-API der Anwendung bereitzustellen

**webservice.soap**

Hier befinden sich die Klassen, welche benötigt werden um die SOAP-API der Anwendung bereitzustellen

# Kapitel 4

## Datenbank

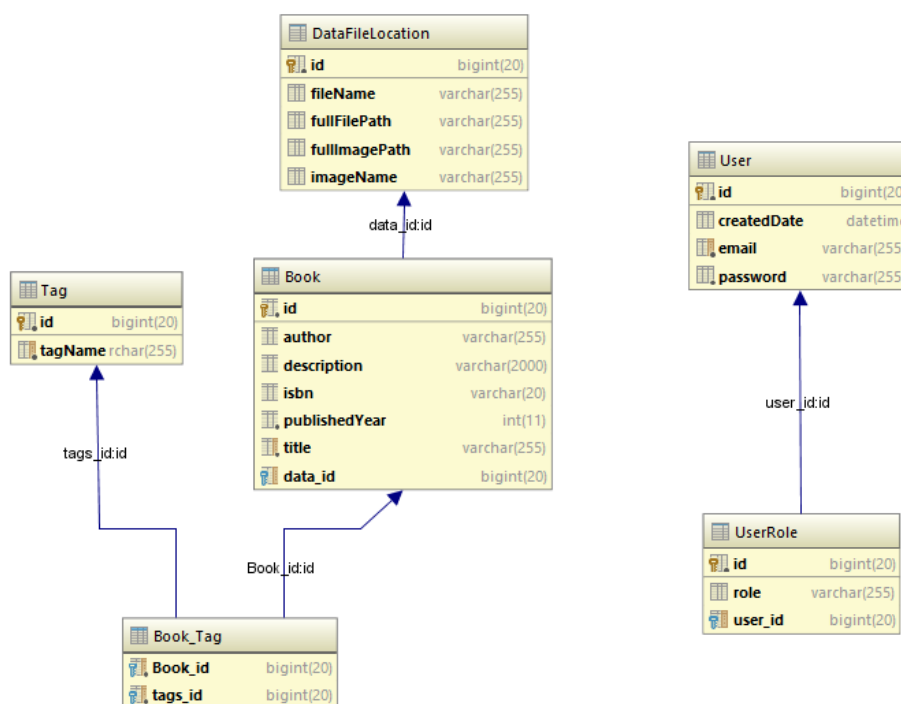


Abbildung 4.1: Modell der Datenbank der Anwendung

### Book

In der Tabelle *Book* werden alle Informationen zu einem Buch gespeichert

### DataFileLocation

In der Tabelle *DataFileLocation* wird der Speicherort der Dateien auf dem Filesystem festgehalten, sowie der Name der Dateien. Die Tabelle *DataFileLocation* steht in einer 1-zu-1-Beziehung mit der Tabelle *Book*

### Tag

In dieser Tabelle werden die Tags zu den Büchern gespeichert. Zwischen der Tabelle *Tags* und der Tabelle *Book* besteht eine n-zu-m-Beziehung, weshalb die Zuordnung der Tags zu den Büchern über die Tabelle *Book\_Tag* verwaltet wird

*Book\_Tag*

Diese Tabelle ist eine Mapping-Tabelle zwischen *Book* und *Tag*

*User*

In dieser Tabelle werden die Daten der registrierten Benutzer der Anwendung gespeichert

*UserRole*

Die *UserRole*-Tabelle ordnet jedem Benutzer in der *User*-Tabelle eine oder mehrere Rollen zu



# Kapitel 5

## Webservices

Die beiden Webservices der Anwendung erlauben Zugriff auf die CRUD-Methoden zum Erstellen, Bearbeiten, Lesen und Löschen eines Buches. Die Webservices greifen direkt auf die Klasse *BookService* der Geschäftslogik zu.

### 5.1 REST-API

Die REST-API ist unter <http://mschoderer.de:8080/bookStore/api/rest/books> erreichbar und bietet über die für REST typischen HTTP-Methoden (GET, POST, PUT, DELETE) zugriff auf die Anwendung. Für die Serialisierung der Objekte verwendet die Schnittstelle das JSON-Format.

### 5.2 SOAP-API

Die wsdl-File für die SOAP-API kann unter <http://mschoderer.de:8080/bookStore/BooksSoapApi?wsdl> erhalten werden.