

# Chapter 1

## Bookstore

### 1.1 Beschreibung

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, ...).

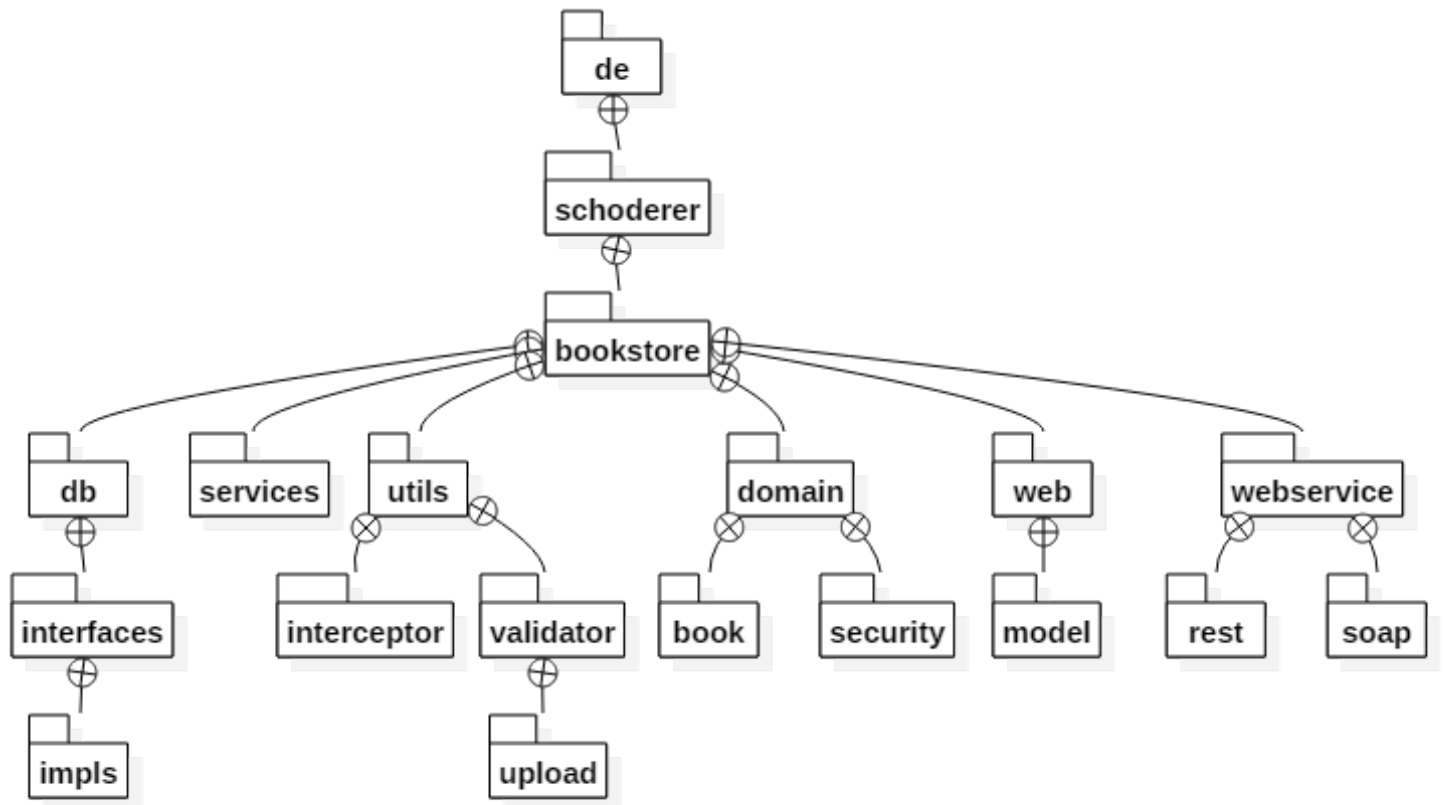
### 1.2 Architektur der Anwendung

#### 1.2.1 Übersicht

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 Webseiten 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 zu in die jeweilige Datenbank zu persistieren

## 1.2.2 Package-Struktur



### Beschreibung der Packages

#### **bookstore**

Dies ist das Hauptpackage der Anwendung

#### **db.interfaces**

Hier liegen die Interfaces, welche die Datenbankverbindungen abkapseln

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

Hier liegen die Implementierungen der Interfaces, welche konkrete Verbindungen zu einer spezifischen Datenbank aufbauen

#### **services**

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

#### **utils.interceptor**

Die Interceptoren befinden sich in diesem Package

#### **utils.validator**

In diesem Package und dem Unterpaket "upload" befinden sich die JSF-spezifischen Validatoren, welche Usereingaben 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 JSF-Beans

#### **webservice.rest**

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

## **webservice.soap**

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