Philipps-Universität Marburg

Sommersemester 2005
Informatikpraktikum im Grundstudium

Entwicklung einer Literaturdatenbank

Entwurfsdokument
Gruppe: Anwendungslogik (AL)

Betreuer: Prof. Dr. Bernhard Seeger

Autoren: Manuel Haim, Marc Koch, Jakub Lidke, Sami Madani, Nan Meng, Stefan Sauer,Doru Vormicescu, Evgenui Zinoviev

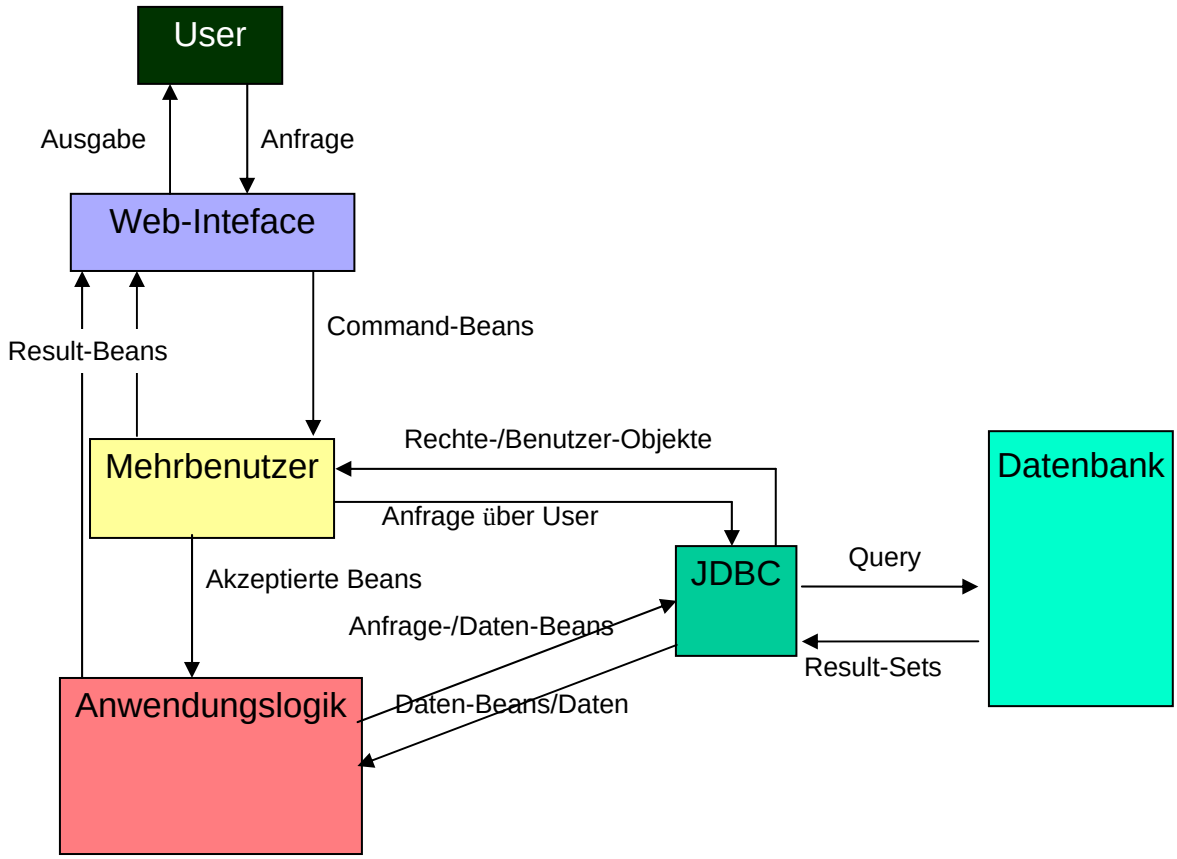
Abgabe: 27.05.2005
Letzte Aktualisierung: 04.06.2005

Inhalt

1. Ablaufschema

1.1. grobes Ablaufschema

Das grobe Ablaufschema einer Anfrage sieht wie folgt aus:



Wie zu erkennen ist, erfolgt der Datenaustausch zwischen den einzelnen Bausteinen hauptsächlich über Beans. Diese werden durch die Anwendungslogik erstellt[1]. In 1.3. gibt es ein detailliertes Schaubild zu den Beans.

1.2. Gruppenbeschreibungen

1.2.1. Webinterface (WI)

Das Webinterface stellt die Schnittstelle Mensch – Maschine her. Der Zugriff wird über ein Webinterface realisiert, welches mit JSP erstellt wird. Das Webinterface übergibt die Befehle des Benutzers mit Hilfe der Beans an die MU, die Benutzer- und Anfragedaten werden in der Bean übergeben.

1.2.2. Mehrbenutzer (MU)

Die Mehrbenutzergruppe stellt die Schnittstelle zwischen dem Web-Interface und der Anwendungslogik dar. Diese Gruppe nimmt die Anfragen, verpackt in Beans, vom Web-Interface entgegen und prüft darauf hin, ob der User die Notwendigen Rechte hat, die Operation durchzuführen, ob die Datei evtl. anderweitig noch gebraucht.

[1] Siehe Kapitel 2, 4.1ff

Nach positiver Verifizierung der Anfrage wird diese an die AL weiter geleitet, andernfalls wird an das WI eine Exception geliefert.
Die Mehrbenutzergruppe bearbeitet ausschließlich Anfragen die die Benutzerverwaltung betreffen und stellt damit, wie der Name sagt, die Mehrbenutzerfähigkeit des Systems sicher.

1.2.3. Anwendungslogik (AL)
Die Anwendungslogik stellt die Beans als Befehls-/Datencontainer zur Verfügung, mit den entsprechenden Methoden. Sie ist dafür verantwortlich, dass alle Programmteile reibungsfrei kommunizieren können. Die genaue Funktionsweise der AL wird in den folgenden Kapiteln genauer erläutert.

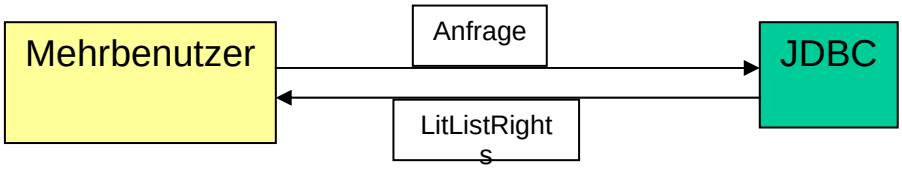
1.2.4. JDBC Gruppe (JDBC)
Die JDBC reagiert auf Anfragen, die ihr von der MU und der AL übergeben werden. Dazu greift sie auf die von der DB erstellte Datenbank zu. Die Ergebnisse der Anfragen werden an die MU und AL zurückgegeben.
Die Anfragen an die JDBC werden zumeist als Bean übergeben.

1.2.5. Datenbankgruppe (DB)
Die Datenbank stellt das Lagerhaus für die JDBC zur Verfügung, diese tätigt dann den Zugriff auf die Daten. Genauere Daten zur Funktionsweise und zur Kommunikation zwischen der JDBC und der DB sind in den entsprechenden Entwurfsdokumentationen zu finden.

Es gibt verschiedene Bean-Sorten, das Zusammenspiel der einzelnen Beans sorgt für einen klar zu identifizierenden Befehl

1.3. Anwendungsfallbeispiel - Speichern einer Literaturliste

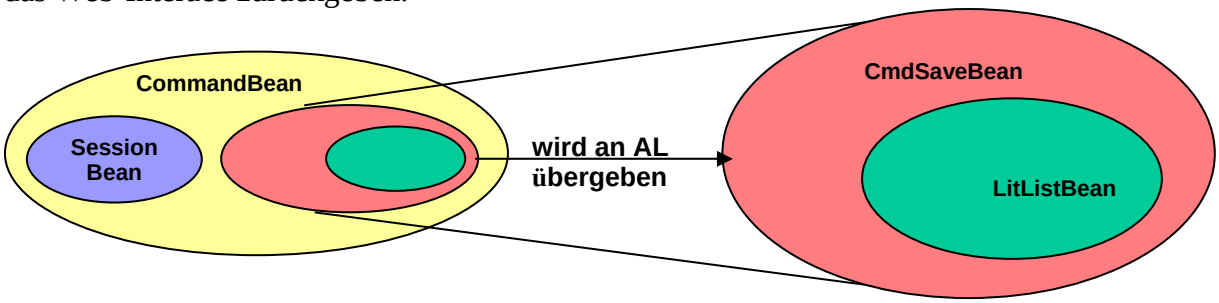
Nachdem das Web-Interface die Anfrage-Bean, bestehend aus der CmdSaveBean, die die LitListBean enthält, SessionBean mit der UserID übergeben hat, erfragt die MU über die UserID den Status des Users von der JDBC und fordert eine LitListRights-Bean an.



Nachdem der Zugriff der JDBC auf die DB erfolgt ist[2] und eine LitListRightsBean zurück gegeben wurde. Verfolgt die MU wie folgt:
Prüfung der Rechte, wenn der User das Recht hat, die Literatur-Liste zu speichern oder wenn er Administrator ist wird die LitListBean an eine entsprechende Methode der Anwendungslogik übergeben.

[2] siehe dazu insbesondere das Entwurfsdokument der JDBC und DB

Falls er die Rechte nicht hat, wird eine ResultBean mit Exception Exception erzeugt und an das Web-Interace zurückgeben.



Die Anwendungslogik erhält die Daten also nach obigem Schema.
Die AL übergibt die Daten direkt weiter an die JDBC. Sie „schiebt“ diese einfach weiter an die JDBC, welche dann mit Hilfe der Beans den eindeutigen Befehl heraus lesen kann:

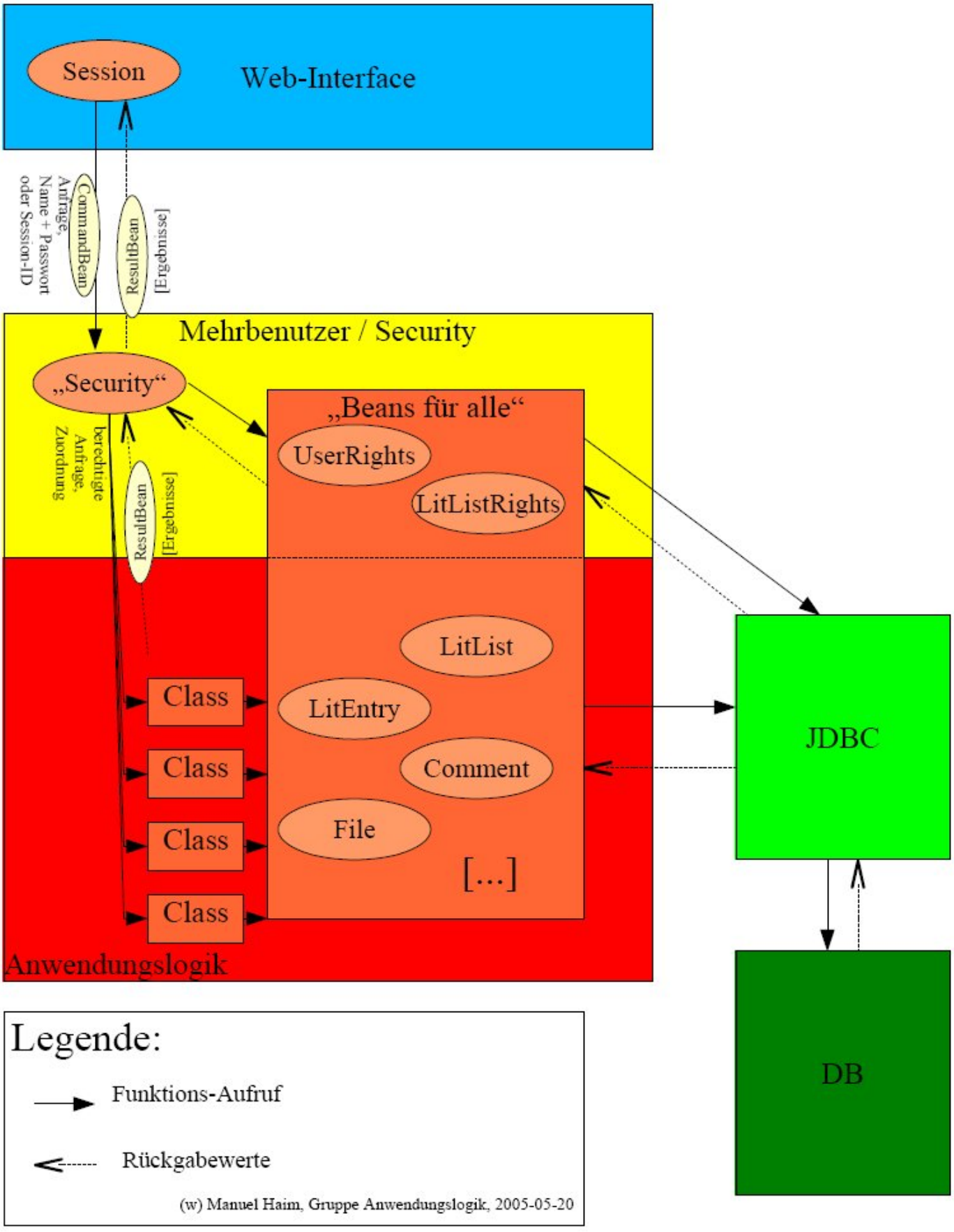
CmdBean → „schreibe“
LitListBean → „schreibe Litliste mit den Daten der LittListBean“

Die AL wartet auf die Ausführung dieses Befehls, um dann eine ResultBean erzeugen zu können.
Nach der Ausführung des Befehls[3] gibt die JDBC dieses an die AL zurück, welche dann eine ResultBean erstellt.
Eine ResultBean enthält max. 2 DataBeans, evtl. eine Exception, falls der Speichervorgang fehlgeschlagen ist.
Bei erfolgreichem Speichern wird eine TextBean mit entsprechender Nachricht erstellt.
Die ResultBean wird direkt an das WI weitergeleitet, welche dann das Resultat an den User übergibt.

1.4. Detailliertes Schema

Dieses Ablaufdiagramm verdeutlicht die Zusammenarbeit der einzelnen Arbeitsgruppen. Insbesondere wird hier die Arbeitsweise mit Hilfe der Beans veranschaulicht. Der Datenaustausch findet mit Hilfe der von der Anwendungslogik erstellten Beans statt. Genauere Ausführungen gibt es in den folgenden Kapiteln.
Aus dem Ablaufschema ist zu erkennen, dass die Multiuser-Gruppe der Anwendungslogik vorgezogen ist. Sie fängt alle Anfragen seitens des Webinterfaces auf und kontrolliert diese auf ihre Zulässigkeit hin (Rechtemanagement) und leitet die Anfragen erst dann an die AL weiter. Die Aufgabe der Anwendungslogik ist es also primär, die Algorithmen zur Verfügung zu stellen, die die Beans brauchen, um als Transportbehälter für die Daten zur Verfügung stehen zu können.

[3] siehe dazu insbesondere das Entwurfsdokument der JDBC und DB

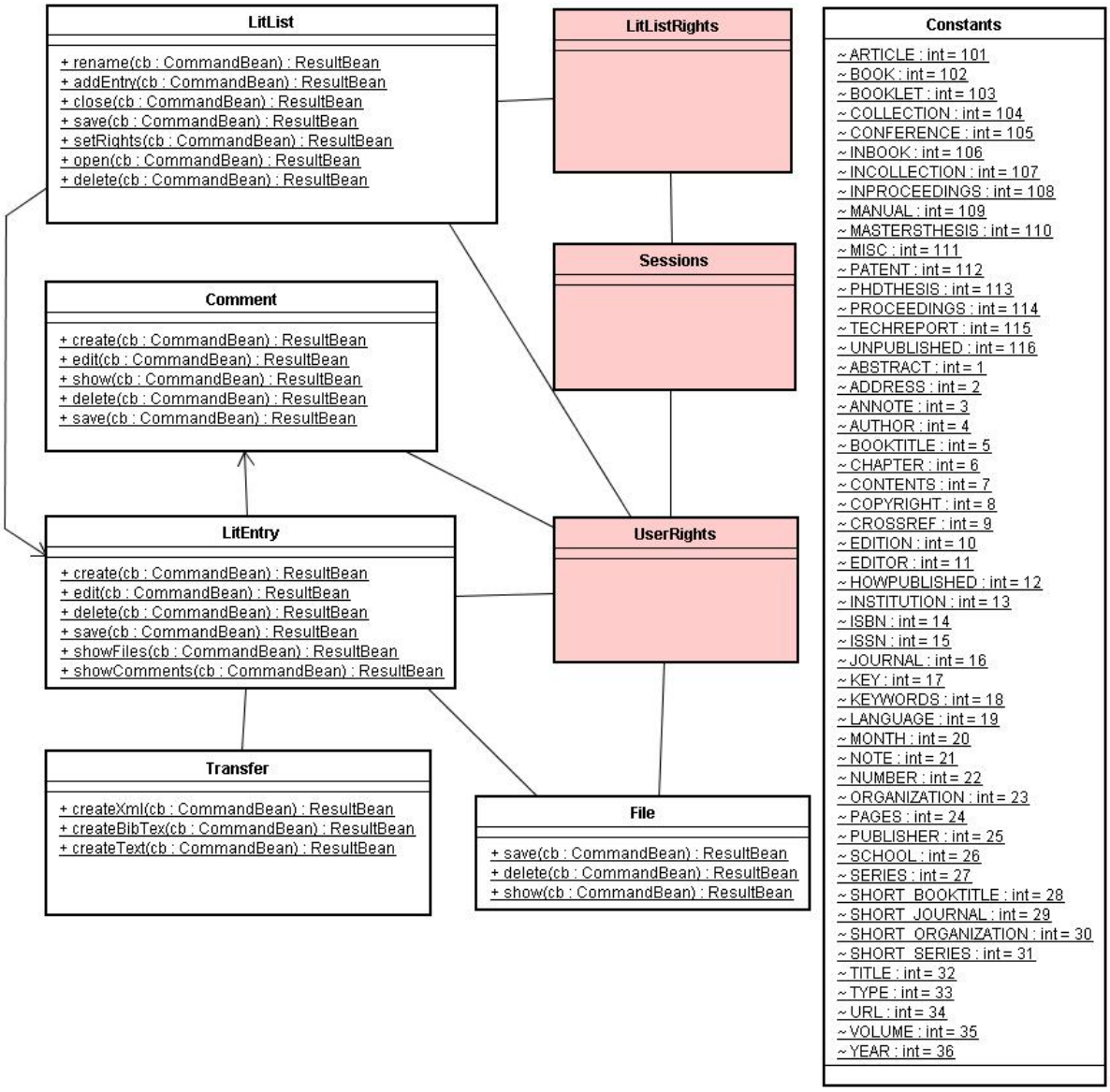

Session
Web-Interface
CommandBean
Anfrage,
Name + Passwort
oder Session-ID
ResultBean
[Ergebnisse]
Mehrbenutzer / Security
„Security“
berechtigte
Anfrage,
Zuordnung
ResultBean
[Ergebnisse]
„Beans für alle“
UserRights
LitListRights
LitList
LitEntry
Comment
File
[...]
Class
Class
Class
Class
Anwendungslogik
JDBC
DB
Legende:
Funktions-Aufruf
Rückgabewerte
(w) Manuel Haim, Gruppe Anwendungslogik, 2005-05-20

2. Bean-Konzept

Die einzelnen Beans dienen, wie im letzten Kapitel angedeutet, zum Datentransport oder der Aufgabenübertragung zwischen den einzelnen Projektteilen (z.B. WI → AL).
Sowohl für den Datentransport als auch für die Aufgabenübermittlung gibt es verschiedene Beans mit jeweils fest spezifizierten Aufgabenbereichen.
So kann die abstrakte CommandBean zu benötigten Beans erweitert werden, mit deren Hilfe effizient die Aufgaben abgewickelt werden können.
Mit Hilfe der DataBeans werden die Daten zu den entsprechenden CommandBeans mit übergeben. Mit der Kombination der DataBean und der CommandBean kann dann von der AL und den anderen Gruppen der gewünschte Anwendungsfall identifiziert und ausgeführt werden.
In den CommandBeans werden die DataBeans als Parameter übergeben. Zusätzlich enthält die CommandBean die SessionBean, welche die Nutzerdaten des Users, der die Anfrage tätigte, enthält. Diese Nutzerdaten werden zunächst bei der MU ausgewertet.
Das Ergebnis dieser Anfrage wird in Form einer ResultBean zurückgegeben, die je nach Ausgang der Anfrage eine DataBean oder eine Exception enthält

Die Typbeschreibung der einzelnen Beans erfolgt im folgenden Kapitel.

3. Klassendiagramm



4. Allgemeines Schnittstellenkonzept

Sämtliche Daten werden in Beans verpackt, das hat den Vorteil, dass die Daten von der Funktionalität getrennt sind. Dies ermöglicht einem einfachen Datenaustausch mit den anderen Programm-Modulen. Im speziellen haben wir eine Verbindung zum WI mittels CommandBeans und ResultBeans und zur JDBC DataBeans.

4.1. CommandBeans (abstract)

Eine CommandBean muss eine solche Form haben:

```
public abstract class CommandBean {
	public CmdBean cmdBean;
	public SessionBean sessionBean;
	public CommandBean(CmdBean cmdBean, SessionBean sessionBean) {
		this.cmdBean = cmdBean;
		this.sessionBean = sessionBean;
	}
}
```

Es können daraus folgende Beans abgeleitet werden:

4.1.1. CmdSaveBean

Diese ist für die Speicherung der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
	public CmdSaveBean(DataBean dataBean1, DataBean dataBean2, DataBean
	dataBean3) {
		this.dataBean1 = dataBean1;
		this.dataBean2 = dataBean2;
		this.dataBean3 = dataBean3;
	}
}
```

4.1.2. CmdCreateBean

Diese ist für das Erzeugen der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
	public CmdCreateBean(DataBean dataBean1, DataBean dataBean2, DataBean
	dataBean3) {
		this.dataBean1 = dataBean1;
		this.dataBean2 = dataBean2;
		this.dataBean3 = dataBean3;
	}
}
```

4.1.3. CmdLoadBean

Diese ist für das Laden der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
	public CmdLoadBean(DataBean dataBean1, DataBean dataBean2, DataBean
	dataBean3) {
		this.dataBean1 = dataBean1;
		this.dataBean2 = dataBean2;
		this.dataBean3 = dataBean3;
	}
}
```

4.1.4. CmdDeleteBean

Diese ist für das Löschen der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
        public CmdDeleteBean(DataBean dataBean1, DataBean dataBean2, DataBean
        dataBean3) {
                this.dataBean1 = dataBean1;
                this.dataBean2 = dataBean2;
                this.dataBean3 = dataBean3;
        }
}
```

4.2. DataBean (abstract)

```
public abstract class DataBean {
}
```

Es können folgende Beans abgeleitet werden:

4.2.1. FileBean

Beinhaltet den Pfad einer Datei, enthaltene Datenfelder:

String litEntryKey
String fileName
String path

4.2.2. CommentBean

Beinhaltet einen Kommentar für einen LitEntry, enthaltene Datenfelder:

String commentKey
String litEntryKey
String owner
String title
String text
String date
String userName

4.2.3. LitEntryBean

Beinhaltet den Literatureintrag einer LitList, enthaltene Datenfelder:

int bookType
String bookId
String[] field
String owner
String[][] files

4.2.4. LitListBean

Beinhaltet einen Array mit LitEntryBeans, enthaltene Datenfelder:

String litListkKey
String litListName
String owner
litEntryBean[] litEntrys

4.2.5. XmlBean

Beinhaltet einen StringArray im XML-Format, enthaltene Datenfelder:

String[] xml

4.2.6. BibTexBean

Beinhaltet einen StringArray im BitTex-Format, enthaltene Datenfelder:

String[] bibTex

4.2.7. TextBean

Beinhaltet einen StringArray im Txt-Format, enthaltene Datenfelder:

String[] text

4.2.8. SearchBean

Beinhaltet ein Suchstring und eine Suchtypvariable, enthaltene Datenfelder:

String[] search

4.2.9. ArrayBean

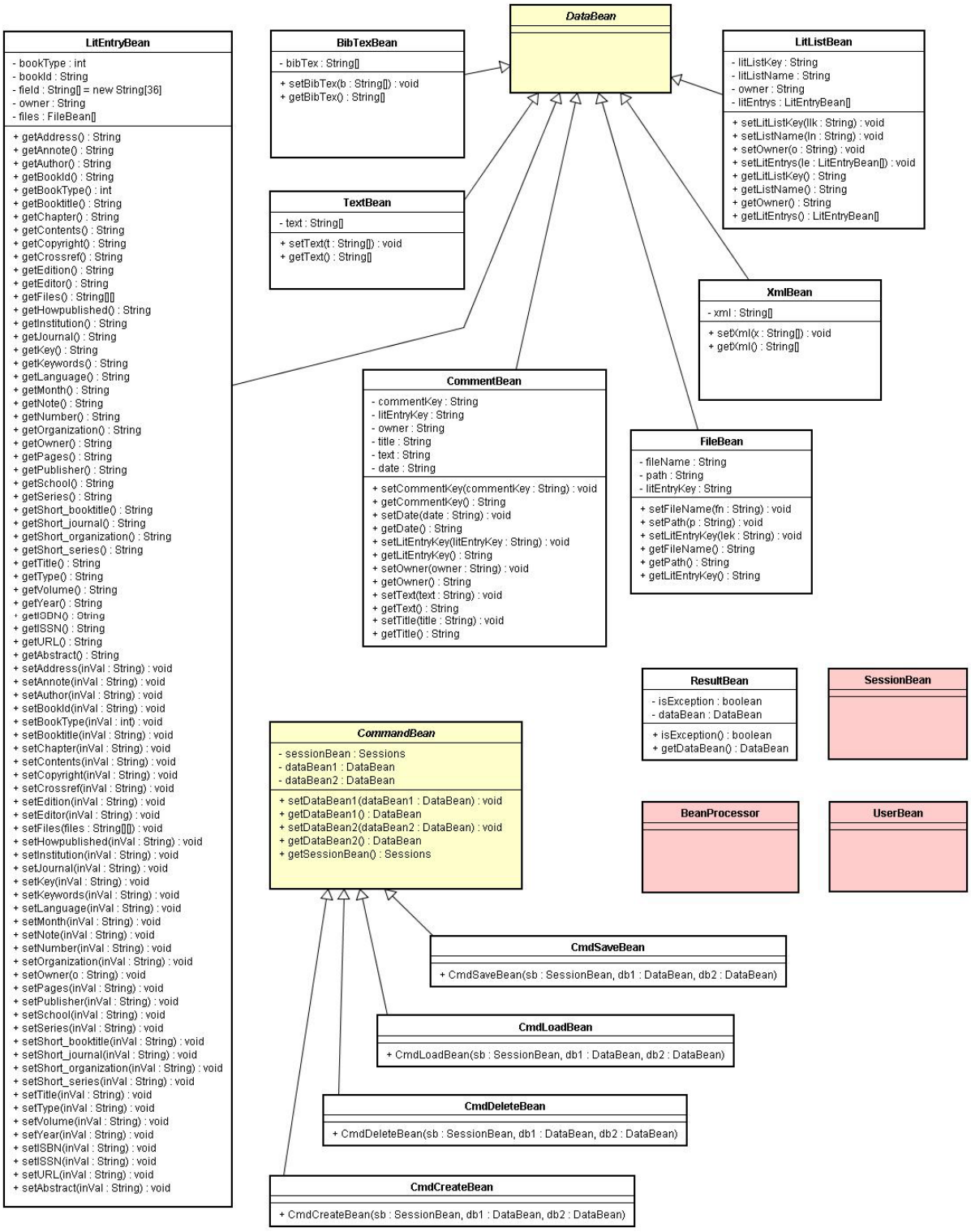
Beinhaltet einen Array von DataBeans (optional), enthaltene Datenfelder:

DataBean[] beanArray

4.3. ResultBean

Die ResultBean liefert das Ergebnis einer Operation an das WI zurück, enthaltene Datenfelder:

Exception e
DataBean db



4.4. Statische Klassen

Die folgenden Klassen enthalten allesamt keine Datenfelder, sondern nur statische Methoden. Diese werden von den Bohnen aufgerufen.
Im Einzelnen sind dies:

4.4.1. LitList

Die Klasse LitList enthält alle Funktionen, die auf einer Literaturliste ausgeführt werden können.

- Liste umbenennen
- Datensatz zur LL hinzufügen
- Schließen LL
- Speichern
- Die Rechte für Benutzer setzen
- Öffnen ein LL
- Liste löschen

LitList	.rename(CommandBean cb)	returns ResultBean
	.addToLitList(CommandBean cb)	returns ResultBean
	.close(CommandBean cb)	returns ResultBean
	.save(CommandBean cb)	returns ResultBean
	.setLitListRights(CommandBean cb)	returns ResultBean
	.open(CommandBean cb)	returns ResultBean
	.delete(CommandBean cb)	returns ResultBean

4.4.2. Comment

Kommentare können von Benutzern zu Datensätzen erstellt werden. Die Klasse kann folgende Operationen durchführen:

- Kommentare erzeugen
- Kommentare editieren
- Kommentare anzeigen
- Kommentare löschen

Comment	.create(CommandBean cb)	returns ResultBean
	.edit(CommandBean cb)	returns ResultBean
	.show(CommandBean cb)	returns ResultBean
	.delete(CommandBean cb)	returns ResultBean

4.4.3. LitEntry

Ein LitEntry enthält alle Informationen die zu einem Buch gespeichert werden sollen. Die Klasse besitzt folgende Methoden:

- Datensatz anlegen
- Datensatz editieren
- Datensatz speichern
- Datensatz löschen
- Datei zum Datensatz anzeigen
- Kommentare zum Datensatz anzeigen

LitEntry	.create(CommandBean cb)	returns ResultBean
	.edit(CommandBean cb)	returns ResultBean
	.delete(CommandBean cb)	returns ResultBean
	.save(CommandBean cb)	returns ResultBean
	.showFiles(CommandBean cb)	returns ResultBean
	.showComments(CommandBean cb)	returns ResultBean

4.4.4. Transfer

Diese Klasse zeichnet sich zuständig für das Übertragen von Datenbankeinträgen in Txt-,Xml-, oder BibTexdateien. Und umgekehrt.

- Lesen von XML-Datei
- Lesen von BibTex-Datei
- Lesen von Text-Datei
- Erzeugen von XML-Datei
- Erzeugen von BibTex-Datei
- Erzeugen von Text-Datei

Transfer	.read(CommandBean cb)	returns ResultBean
	.read(CommandBean cb)	returns ResultBean
	.read(CommandBean cb)	returns ResultBean
	.create(CommandBean cb)	returns ResultBean
	.create(CommandBean cb)	returns ResultBean
	.create(CommandBean cb)	returns ResultBean

4.4.5. File

Diese Klasse kümmert sich um das Anhängen von Dateien an Literatureinträge und um deren Ausgabe.

- Speichern
- Löschen
- Anzeigen

File	.save (CommandBean cb)	returns ResultBean
	.delete(CommandBean cb)	returns ResultBean
	.show(CommandBean cb)	returns ResultBean

4.4.6. Interface Constants

Hier sind alle nötigen Konstanten zur Speicherung von Daten in der Datenbank enthalten.

Constants

~ ARTICLE : int = 101
- BOOK : int = 102
- BOOKLET : int = 103
~ COLLECTION : Int = 104
~ CONFERENCE : Int = 105
~ INBOOK : Int = 106
~ INCOLLECTION : int = 107
~ INPROCEEDINGS : int = 108
~ MANUAL : int = 109
- MASTERSTHESIS : int = 110
~ MISC : int = 111
~ PATENT : Int = 112
~ PHDTHESIS : Int = 113
~ PROCEEDINGS : Int = 114
~ TECHREPORT : int = 115
~ UNPUBLISHED : int = 116
~ ABSTRACT : int = 1
- ADDRESS : int = 2
~ ANNOTE : Int = 3
~ AUTHOR : Int = 4
~ BOOKTITLE : Int = 5
~ CHAPTER : int = 6
~ CONTENTS : int = 7
~ COPYRIGHT : int = 8
- CROSSREF : int = 9
- EDITION : int = 10
~ EDITOR : Int = 11
~ HOWPUBLISHED : Int = 12
~ INSTITUTION : Int = 13
~ ISBN : int = 14
~ ISSN : int = 15
~ JOURNAL : int = 16
- KEY : int = 17
~ KEYWORDS : int = 18
~ LANGUAGE : Int = 19
~ MONTH : Int = 20
~ NOTE : Int = 21
~ NUMBER : int = 22
~ ORGANIZATION : int = 23
- PAGES : int = 24
- PUBLISHER : int = 25
~ SCHOOL : Int = 26
~ SERIES : Int = 27
~ SHORT_BOOKTITLE : Int = 28
~ SHORT_JOURNAL : int = 29
~ SHORT_ORGANIZATION : int = 30
~ SHORT_SERIES : int = 31
- TITLE : int = 32
- TYPE : int = 33
~ URL : Int = 34
~ VOLUME : Int = 35
~ YEAR : Int = 36

5. Weiterführende Links der AL

Die AL hat auf ihrer Homepage noch weitere zusätzliche Dokumente zur Verfügung gestellt, wie zum Beispiel eine grobe Umschreibung aller Anwendungsfälle. Die Grafiken, die hier zu sehen sind, sind auch unter http://www.mathematik.uni-marburg.de/~haim/it/ipg/docs/?C=M;O=A zu finden.

6. Haftungsausschluss

Die AL-Gruppe übernimmt keine Haftung für die Aktualität, die inhaltliche Richtigkeit sowie für die Vollständigkeit der in ihrem Entwurfsdokument eingestellten Informationen, es sei denn die Fehler wurden vorsätzlich oder grob fahrlässig aufgenommen. Dies bezieht sich auf eventuelle Schäden materieller oder ideeller Art Dritter, die durch die Nutzung dieses Webangebotes verursacht wurden. Änderungen vorbehalten.