

# Entwurf

Vorläufige Abgabe

## Team 2

SEP WS 2021/22



Betreuer:

PROF. DR. CHRISTIAN BACHMAIER

Projektphase	Leiter
Pflichtenheft	Johann Schicho
Entwurf	Stefanie Gürster
Feinspezifikation	Johannes Garstenauer
Implementierung	Thomas Kirz
Validierung	Sebastian Vogt

5. November 2021

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Systemarchitektur</b>	<b>1</b>
2.1	Schichten . . . . .	1
2.2	Pakete . . . . .	2
2.2.1	de.lases.control . . . . .	2
2.2.2	de.lases.business . . . . .	2
2.2.3	de.lases.persistence . . . . .	2
2.2.4	de.lases.global . . . . .	3
2.3	Fehlerbehandlung . . . . .	3
2.3.1	Geprüfte Ausnahmen . . . . .	3
2.3.2	Ungeprüfte Ausnahmen . . . . .	3
2.4	Frameworks . . . . .	3
2.4.1	Jakarta Server Faces . . . . .	3
2.4.2	Context and Dependency Injection . . . . .	3
2.5	Patterns . . . . .	4
2.5.1	Model View Controller . . . . .	4
2.5.2	Singleton . . . . .	4
2.5.3	Data Transfer Object . . . . .	4
2.5.4	Repository . . . . .	4
2.5.5	Object Pool . . . . .	5
2.5.6	Observer . . . . .	5
<b>3</b>	<b>Klassendiagramm</b>	<b>5</b>
3.1	Klassendiagramm . . . . .	5
3.2	Klassenbeschreibungen . . . . .	6
3.2.1	de.lases.business.service . . . . .	6
3.2.2	de.lases.business.utils . . . . .	7
3.2.3	de.lases.global.util . . . . .	7
3.2.4	de.lases.business.internal . . . . .	7
3.2.5	de.lases.control.validation . . . . .	8
3.2.6	de.lases.global.transport . . . . .	8
3.2.7	de.lases.persistence.repository . . . . .	9
3.2.8	de.lases.persistence.util . . . . .	9
3.2.9	de.lases.persistence.exception . . . . .	10
3.2.10	de.lases.control.backing . . . . .	10
<b>4</b>	<b>Facelets</b>	<b>11</b>
4.1	Templates . . . . .	11
4.2	Seiten . . . . .	13
<b>5</b>	<b>Systemfunktionen</b>	<b>26</b>
5.1	Logging . . . . .	26
5.2	Sicherheit . . . . .	26
5.3	Asynchrone Threads . . . . .	27
5.4	Systemstart . . . . .	27
5.5	Systemstopp . . . . .	27

<b>6</b>	<b>Datenfluss</b>	<b>28</b>
6.1	Hochladen einer Revision . . . . .	28
6.2	Ablehnung einer Einreichung . . . . .	28
<b>7</b>	<b>ER-Modell</b>	<b>29</b>
7.1	Entities . . . . .	29



## 2.2 Pakete

### 2.2.1 de.lases.control

Dieses Paket enthält alle Klassen der Kontrollschicht.

*de.lases.control.conversion* enthält alle Klassen, die zur Konvertierung von Nutzereingaben in Facelets verwendet werden.

*de.lases.control.validation* enthält alle Klassen, die zur Validierung von Nutzereingaben in Facelets verwendet werden.

*de.lases.control.backing* enthält alle Managed Beans welche zur Darstellung der Facelets aus der View Verwendung finden. Zu jedem Facelet [facelet] existiert genau ein Backing-Bean, welches [facelet]Backing genannt wird.

### 2.2.2 de.lases.business

Dieses Paket enthält alle Klassen der Anwendungslogikschicht.

*de.lases.business.service* enthält alle Klassen, die den Hauptteil der Anwendungslogik umsetzen. Das sind also alle Dienste welche den Backing Beans zur Verfügung gestellt werden und auf die auf die Repositories zugreifen. *→ statische Utilities?*  
*Transition steuerung hier, oder?*

*de.lases.business.util* enthält die Hilfsklassen, welche Dienste zur Verfügung stellen, die von der Anwendungslogik selbst getrennt werden können.

*de.lases.business.exception* enthält die Exceptionklassen, die ein Fehlverhalten im System repräsentieren.

*de.lases.business.internal* enthält die Klassen, welche zur internen Verwaltung verwendet werden. Hierzugehört beispielsweise die Sessionverwaltung, sowie der Systemstart und -stop. *obere Schicht wenn möglich?*

### 2.2.3 de.lases.persistence

Dieses Paket enthält alle Schichten der Persistenzschicht.

*de.lases.persistence.repository* enthält alle Klassen, die dem direkten Zugriff auf die Datenbank dienen. *→ statische Utilities?*

*de.lases.persistence.util* enthält die Hilfsklassen zur Datenbankverwaltung, wie einen Connection-Pool. *CP von Businessschicht aber nicht zugreifbar!*

*de.lases.persistence.exception* enthält die Exceptionklassen, welche ein Fehlverhalten im Umgang mit der Datenbasis repräsentieren.

## 2.2.4 de.lases.global

Dieses Paket enthält alle Klassen welche über mehrere Schichten hinweg verwendet werden.

*de.lases.global.logging* enthält Hilfsklassen für schichtenübergreifende Loggingfunktionen. ✓

*de.lases.global.transport* enthält alle POJO-Klassen, die Daten in der Anwendung repräsentieren und zu deren Transport verwendet werden.

Nur "Data Beans" wie Logik

## 2.3 Fehlerbehandlung

Bei Verwendung der Anwendung können Fehler auftreten. Diese werden zur Nachvollziehbarkeit immer geloggt. Daraufhin wird auf verschiedene Weisen verfahren.

### 2.3.1 Geprüfte Ausnahmen

Geprüfte Ausnahmen repräsentieren Ausnahmesituationen, mit denen gerechnet werden muss und auf die reagiert wird. Treten solche auf, dann werden sie mit einer Ausnahmeklasse und Fehlermeldung an den Aufrufer weiterpropagiert. In Service-Klassen werden auftretende Ausnahmen weiter zu den Backing-Beans propagiert, wo Faces-Messages generiert werden. Diese werden dann von den zugehörigen Facelets gerendert.

es gibt auf geprüfte Ausnahmen der unteren Schicht

✓ welche gibt es

Problem: Jedes hat aus Control-Schicht den Service braucht try-catch & Fallkanten Code  
→ Abstraktion

### 2.3.2 Ungeprüfte Ausnahmen

Bei ungeprüften Ausnahmen handelt es sich um fatale Fehler, wie sie beispielsweise durch Programmierfehler entstehen können. Diesen werden vom anwendungsweiten `ExceptionHandler` aufgegriffen. Der User wird daraufhin auf die Fehlerseite weitergeleitet.

✓ welche gibt es

## 2.4 Frameworks

### 2.4.1 Jakarta Server Faces

Jakarta Server Faces (JSF; früher JavaServer Faces) ist ein Framework-Standard zur Entwicklung von grafischen Benutzeroberflächen für Webanwendungen in Java, basierend auf Servlets und JSP-Technik. Es wird die Referenzimplementierung *Mojarra* verwendet.

### 2.4.2 Context and Dependency Injection

Die *Context and Dependency Injection* ist eine Technologie welche die Abhängigkeitsverwaltung zwischen Objekten vereinfacht. Im Sinne der *Inversion of Control* werden dem Entwickler Werkzeuge an die Hand gelegt, mithilfe derer er Objektreferenzierung tätigen kann, ohne diese Objekte vorher explizit zu erstellen. Diese referenzierten Objekte können in andere Objekte zur Laufzeit *injiziert* werden. Somit können Objekte in dieser Anwendung ohne Initialisierung oder umständlichen Transport der Referenzierung verwendet werden. Dies ermöglicht eine losere Kupplung zwischen Klassen und somit

✓ wo genau verwendet

Erweiterbarkeit und Austauschbarkeit. Die *CDI* verwaltet in dieser Anwendung alle Objekte der Backing-Beans und Services. Es wird die Referenzimplementierung  *Weld*  von *JBoss* verwendet.

## 2.5 Patterns

### 2.5.1 Model View Controller

Dieses Entwurfsmuster trennt die Klassen und Komponenten der Anwendung in drei verschiedenen Teile, welche jeweils eigene Aufgaben übernehmen:

- Die **View-Komponenten** übernehmen alle Aufgaben, welche mit der grafischen Darstellung der Anwendung zusammenhängen. In dieser Anwendung sind dies die Facelets.
- Die **Controller-Komponenten** übernehmen alle Aufgaben, welche die Interaktion zwischen der View und dem Model betrifft. Sie behandeln Nutzereingaben, Validation und Konvertierung etc. In dieser Anwendung befinden sie sich in den `de.lases.control` Paketen.
- Die **Model-Komponenten** beinhalten die Modelldaten, sowie jene Klassen welche sich mit deren Beschaffung und Verarbeitung beschäftigen. In dieser Anwendung befinden sie sich in den `de.lases.business` und `de.lases.persistence` Paketen.

Die Verwendung dieses Patterns bietet viele Vorteile hinsichtlich der Modularität und Erweiterbarkeit der Anwendung.

### 2.5.2 Singleton

Eine Klasse, welche nach dem Singleton-Entwurfsmuster implementiert wurde zeichnet sich dadurch aus, dass es nur genau eine Instanziierung von dieser Klasse geben kann und darf. Hierdurch werden vorrangig Ressourcen gespart, außerdem wird die Verwaltung der im Objekt gekapselten Funktionalität erleichtert. Ein Beispiel hierfür ist der Connection Pool, damit sichergestellt wird, dass eine genau festgelegte Anzahl an Datenbankverbindungen existiert und nicht mehr. *+ Rechte (globales/persistenten) Zugriff*

### 2.5.3 Data Transfer Object

Dieses Entwurfsmuster schreibt die Verwendung von *Data Transfer Objects* zur einheitlichen Darstellung der Entitäten in den Modelldaten und dem Transport von Daten vor. Diese sind POJOs, welche private Felder zur Repräsentation der Daten mit Getter- und Settermethoden haben. Der Vorteil der Verwendung dieses Entwurfsmusters liegt darin, dass eine einheitliche und übersichtliche Schnittstelle für Datenrepräsentation und Transport in der Anwendung besteht. Die zugehörigen Klassen lagern im `de.lases.global.transport` Paket.

*↳ Parameter der (Schichten-) Methoden immer DTOs und keine "flachen"?*

### 2.5.4 Repository

Beim *Repository* Pattern wird für jedes Business-Objekt oder jede Entität eine Klasse definiert welche <sup>der</sup> Datenbankzugriff auf die zugehörigen Daten steuert. Für diesen Zugriff werden <sup>oft</sup> die *CRUD*-Methoden, das heißt

- *CREATE* zur Erstellung,

*was bei join über mehrere Entitäten  
wo sind dann solche Methode?*

das nicht denke ich nicht, wo geht es z.B. eine Liste von Journalen



- READ zum Erhalten,
- UPDATE zum Updaten,
- und DELETE zum Löschen

eines oder mehrerer Datenbankeinträge der zugehörigen Entität, angeboten. Weiterhin findet das Mapping auf die DTO-Objekte dieser Anwendung statt. In dieser Anwendung finden sich die *Repository-Pattern* Klassen im Paket `de.lases.persistence.repository`. Durch die mit der Verwendung dieses Patterns gewonnenen Übersichtlichkeit erleichtern wir uns die Implementation und ermöglichen leichte Erweiterbarkeit der Daten-schemas. und gelösten Instanz der jetzigen Datenhaltung..

### 2.5.5 Object Pool

Das *Object Pool* Erzeugungsmuster wird dazu verwendet, Objekte nach initialer Erzeugung vorzubehalten, da dies sinnvoller ist als sie bei jeder Verwendung neu zu erzeugen. Somit werden Ressourcen gespart. In dieser Anwendung findet dieses Entwurfsmuster beim *Connection Pool* Verwendung und bietet den zusätzlichen Vorteil, dass eine feste Maximalanzahl an Datenbankverbindungen festgelegt werden kann.

### 2.5.6 Observer

Das *Observer* Verhaltensmuster beschreibt Methoden, welche auf Änderungen im Zustand eines Systems warten und daraufhin handeln. In unserem System findet das beispielsweise beim LifeTimeListener zur Reaktion auf das Herunterfahren oder Starten des Systems Verwendung. Font texttt

Handelt Klassenmethoden verwenden wie hier ✓

## 3 Klassendiagramm

Sebastian Vogt, Johannes Garstenauer, Johann Schicho

Im Folgenden Kapitel wird für jedes Package der Autor dieses Abschnitts des Klassendiagramms genannt.

geht in den durchgehenden Seiten innerhalb

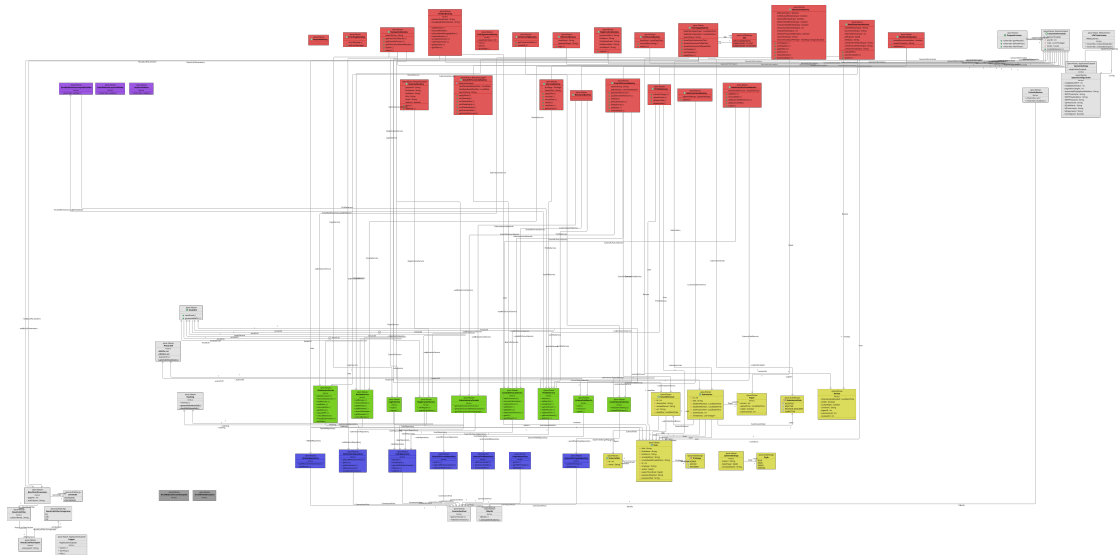
### 3.1 Klassendiagramm

Sebastian Vogt Im folgenden Klassendiagramm wurden folgenden Konventionen verwendet:

- Properties sind als private Attribute modelliert. Konstruktoren sind ausgespart. ✓
- Referenzattribute auf andere Klassen des Diagramms sind als Pfeile dargestellt. Diese haben standardmäßig keine Getter und Setter, außer anders angegeben. ✓
- Konstruktoren sind ausgespart, außer dieser ist privat. anders herum erwartet ...
- In den Backingbeans sind Attribute, die lediglich einen Anzeigetext aus einem Ressourcebundle laden, nicht angegeben.

so was gibt es doch gar nicht  
→ Facet greift über Expression doch direkt in  
auf das Resource-Bundle



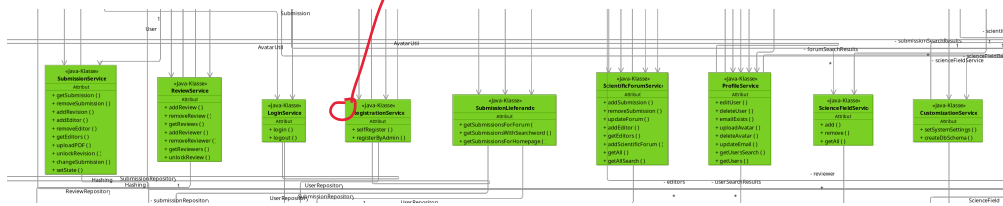


## 3.2 Klassenbeschreibungen

### 3.2.1 de.lases.business.service

Johannes Garstenauer

*manuelle RST-Logik oder, mehr Grund nicht gibt es RST-Probleme?*



*generierung? Scope? static?*

Klassenname	Beschreibung
CustomizationService	Provides methods for the manipulation of system settings.
ScienceFieldService	Provides methods for adding and removing scientific categories.
ProfileService	Provides methods for the manipulation and removal of users.
ScientificForumService	Provides methods for the manipulation and creation of scientific forums.
SubmissionLieferando?	Provides methods for delivering lists of submissions, taking into account their filtering, sorting and calling user.
RegistrationService	Provides methods for the creation of users.
LoginService	Provides methods for the login and logout operations.
ReviewService	Provides methods for the delivery, creation and removal of reviews.
SubmissionService	Provides methods for the delivery, creation, removal and manipulation of submissions.

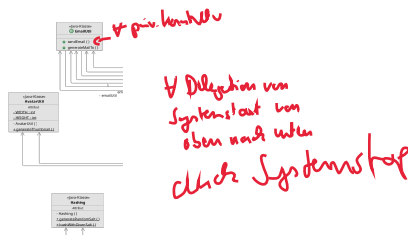
*Listenabstraktion durch Schnittstellen*

### 3.2.2 de.lases.business.utils

Johannes Garstenauer

Utility-Klassen?

(hoff. in anderer  
schicht)  
- Util bei prog. Zugriff  
auf Resource Bundle

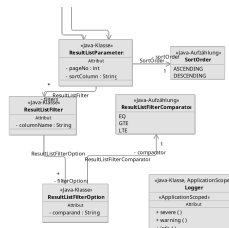


Klassenname	Beschreibung
AvatarUtil	Provides support in the generation of a thumbnail from an image.
Hashing	Provides support in the hashing of passwords.
EmailUtil	Provides support in the sending of emails and creation of mailto links.

### 3.2.3 de.lases.global.util

Johann Schicho

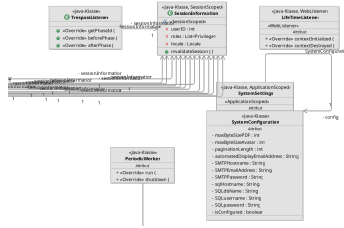
Logging



Klassenname	Beschreibung
Logging	This Logger allows logging in different log-levels.
ResultListParameters	This class bundles parameters for requesting lists from the database. This includes page number, sorting of the results and filtering of the results
SortOrder	A List can be sorted in ascending or descending order
ResultListFilter	A filter that can be applied on a column of a table
ResultListFilterOption	A single comparison that can be used to construct a ResultListFilter
ResultListFilterComparator	The three comparators equal to, less than or equal and greater than or equal

### 3.2.4 de.lases.business.internal

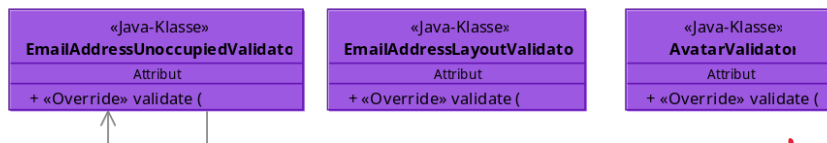
Johann Schicho



Klassenname	Beschreibung
TrespassListener	Manages user access to resources, most importantly webpages.
SessionInformation	Wraps data saved in the session.
LifeTimeListener	Takes care of system start and shutdown operations
PeriodicWorker	Takes care of periodical database cleanup jobs.
SystemSettings	Loads and updates the systems' configuration.
SystemConfiguration	Wraps data used for the system configuration.

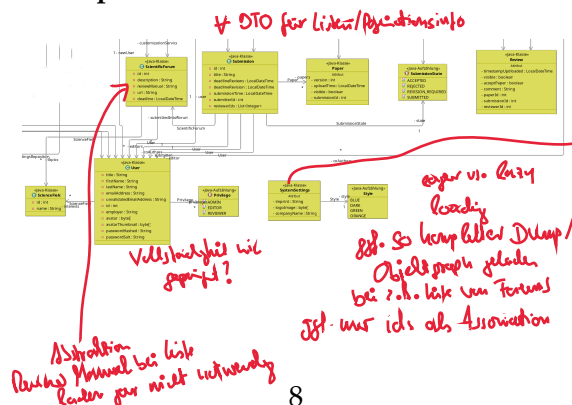
### 3.2.5 de.lases.control.validation

Sebastian Vogt, ff.



Klassenname	Beschreibung
AvatarValidation	Validates that a suitable avatar image has been uploaded.
EmailAddressLayoutValidation	Validates that the email regex pattern is ok.
EmailAddressUnoccupiedValidator	Validates that a given email is not already in use within the system.

### 3.2.6 de.lases.global.transport



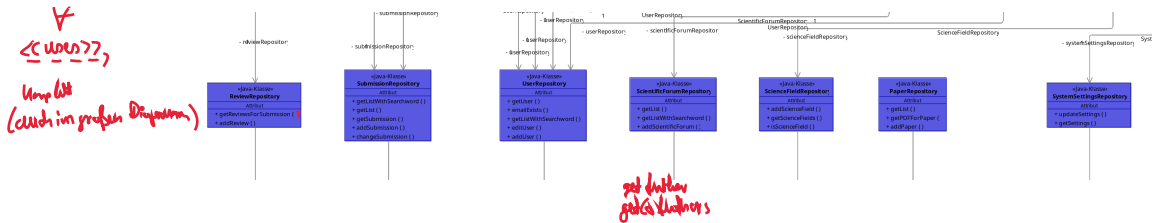
Überschrift und Überschrift hier welcome Page fehlt

Handwritten: *Handwritten DTO*

Handwritten: *haben gleich selbe List DB, repository!*

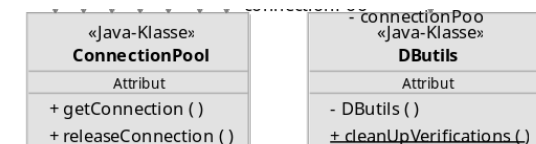
Klassenname	Beschreibung
Paper	This DTO represents a paper.
Privilege	This represents a user privilege.
Review	This DTO represents a review.
ScienceField	This DTO represents a field of Science.
ScientificForum	This DTO represents a forum
Style	This represents a user interface style.
Submission	This DTO represents a submission.
SubmissionState	This represents a submissions state.
SystemSettings	This DTO represents the system settings.
User	This DTO represents a user.

### 3.2.7 de.lases.persistence.repository



Klassenname	Beschreibung
PaperRepository	This repository can get the list of papers for a submission or add new ones.
ReviewRepository	This repository can get the reviews for a certain submission and add new ones.
ScienceFieldRepository	This repository can get or add new fields of science from/to the database.
ScientificForumRepository	This repository can get a list of journals and conferences or add new ones.
SubmissionRepository	This repository allows CRUD operations on submissions.
SystemSettingsRepository	This repository can read or update the system settings.
UserRepository	This repository allows CRUD operations on users.

### 3.2.8 de.lases.persistence.util



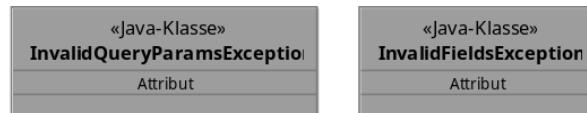
Handwritten: *Handwritten DTO*

Handwritten: *Handwritten: haben gleich selbe List DB, repository!*

Klassenname	Beschreibung
ConnectionPool	Provides and manages connections to the database.

DButils	Provides functions for general database services.
---------	---

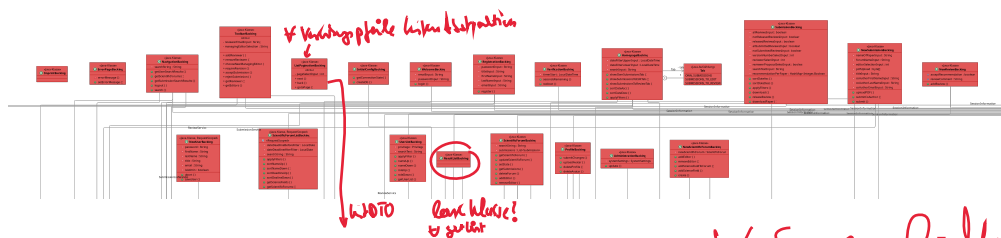
### 3.2.9 de.lases.persistence.exception



Copy  
 -file Not found  
 -Logger kann nicht  
 schreiben

Klassenname	Beschreibung
InvalidFieldsException	Hints at an invalid field in a dto.
InvalidQueryParamsException	Hints at invalid filter or sort parameters given to the query.

### 3.2.10 de.lases.control.backing



if Scope in Manged Beans

Klassenname	Beschreibung
NewSubmissionBacking	Backing bean for the page for creating a new submission.
SubmissionBacking	Backing bean for the submission page.
NewReviewBacking	Backing bean for the page for creating adding a new review.
VerificationBacking	Backing bean for the verification page.
ToolbarBacking	Backing bean for the side toolbar.
NavigationBacking	Backing bean for the navigation bar.
WelcomeBacking	Backing bean for the welcome and login page.
RegistrationBacking	Backing bean for the registration page.
HomepageBacking	Backing bean for the homepage for logged in users.
NewUser	Backing bean for the page for adding a new user.
ScientificForumListBacking	Backing bean for the list of scientific forums.
UserListBacking	Backing bean for the list of users.
ResultListBacking	Backing bean for the search result page.
ScientificForumBacking	Backing bean for the scientific forum page.
NewScientificForumBacking	Backing bean for the page for adding a new forum.
ProfileBacking	Backing bean for the profile page.
AdministrationBacking	Backing bean for the administration page.

✓ Facelet-Komponenten - Controls  
 → meistens per Konvention abhandelbar

## 4 Facelets

Stefanie Gürster, Johann Schicho

Im Folgendem Abschnitt werden Abkürzungen für die verschiedenen Rollen eingeführt: **A** steht für Administrator, **AN** für einen anonymen Nutzer, **N** für einen angemeldeten Nutzer, **E** für einen Editor und **G** steht für einen Gutachter. Ist eine Funktion für alle Benutzerrollen vorgesehen, so werden diese unter dem Begriff **Alle** zusammengefasst.

Verantwortung  
 übertragen

Sind außerdem keine Labels für Komponenten angegeben, dann sind anstelle dieser Icons (Fontawesome) angedacht.

Labels, die dem Administrator  
 würde ich nicht als einzelnes Control spezifizieren

### 4.1 Templates

Johann Schicho

**navigation.xhtml** ist die Kopfzeile der Webanwendung. Diese bietet die Suchfunktion an und Links zu verschiedenen Listen und dem Profil.

Man könnte auch  
 (er kann es)  
 Spezifikation  
 nach Spalten  
 durch...)

ID	Typ	Beschreibung	Sichtbarkeit
searchField	inputText	Suchleiste	Alle
search	commandButton	Suche ausführen.	Alle
userList-Link	link	Link zur Übersichtsseite aller Nutzer	A,E
forumList-Link	link	Link zur Übersichtsseite aller Journale und Konferenzen	Alle
logoutButton	commandButton	Loggt den Nutzer aus dem System aus und leitet zur Loginseite weiter	Alle
profileLink	link	Link zur Profilübersicht	N

**main.xhtml** ist das Template, welches den Inhalt der Seiten zwischen Kopf- und Fußzeile einbettet.

ID	Typ	Beschreibung	Sichtbarkeit
navigation-Bar	include	Kopfzeile	Alle
mainContent	insert	Seiteninhalt, siehe 4.2?	Alle
footerBar	include	Fußzeile	Alle
toolBar	include	Seitenleiste	A,E

✓

**footer.xhtml** ist die Fußzeile der Webanwendung und für alle sichtbar.

ID	Typ	Beschreibung	Sichtbarkeit
imprint-Link	link	Link zur Seite des Impressums	Alle
language	outputText	Anzeige der eingestellten Sprache.	Alle

**toolbar.xhtml** ist die Seitenleiste für Editoren und Administratoren auf der Einreichungsübersicht. Hier wird die Einreichung verwaltet.

reicht mit E, da A per  
kenntnis das  
man kann / darf?

ID	Typ	Beschreibung	Sichtbarkeit
reviewer-Label	outputLabel	Label für Gutachterangabe.	A,E
addRe-viewerField	inputText	Eingabefeld zur Angabe einer E-Mailadresse	A,E
deadline-Reviewer	outputLabel	Label für Deadline.	A,E
addDead-lineRe-viewer	inputText	Eingabefeld zur Angabe einer Deadline.	A,E
addRe-viewerBtn	commandButton	Knopf zum hinzufügen des Gutachters	A,E
review-erTable	dataTable	Liste der Gutachter	A,E
removeRe-viewer	commandButton	Entferne einen bestimmten Gutachter	A,E
editorLabel	outputLabel	Label für Editorauswahl.	A,E
selectE-di-tor	selectOneMenu	Auswahländerung des verwalten-den Editors	A,E
saveEditor	commandButton	Speichern der Änderung	A,E
deadline-Revision	outputLabel	Label für Deadline.	A,E
revisi-onDeadline	inputText	Eingabe zur Angabe einer Dead-line.	A,E
require-Revisi-onBtn	commandButton	Fordert den Einreicher auf, seine Einreichung zu überarbeiten	A,E
acceptSub-missionBtn	commandButton	Akzeptiere die Einreichung	A,E
rejectSub-missionBtn	commandButton	Lehne die Einreichung ab	A,E

Stefanie Gürster

hier schon im Subsysteme Kontextualität gewandelt, das ist zu fein!

**listPagination.xhtml** ist eine Composite Component zur Paginierung von Listen. *listPa-gination.xhtml* wird in jedem Footer einer Tabelle eingebunden.

ID	Typ	Beschreibung	Sichtbarkeit
back	commandButton	Laden der vorherigen Seite der Lis-te.	Alle
next	commandButton	Laden der nächsten Seite der Liste.	Alle
page	selectOneMenu	Laden einer bestimmten Seite	Alle

↳ knippte zu Seite  
↳ liste den sich  
↳ Filtermöglichkeiten

Nach rel. Untereinheiten

## 4.2 Seiten

*Güster?* Verantwortlichkeit nicht klar erkennbar

**initialConfig.xhtml** Auf dieser Seite landet man beim ersten Systemstart. Sie ist nur für Administratoren zugänglich. Hier kann der Administrator die Datenbankschemata erstellen lassen. ✓

*Hilfflos in Rollen, die Facet übersteigt o.fer diff*

ID	Typ	Beschreibung	Sichtbarkeit
dbConnectionState	outputText	Information über die Verbindung mit der Datenbank	A
createDbBtn	commandButton	Erstellt die Datenbankschemata	A

*↳ Welches Typ von Facet-Controls wäre wie besprochen. erst in Spezifikation spezifiziert.*

**welcome.xhtml** Auf der Login- bzw. Welcome-Seite wird *LasEs* vorgestellt. Zusätzlich gibt es ein Login-Formular zur Anmeldung im System. Für nicht registrierte Nutzer wird man über einen gegebenen Link zu *registration.xhtml* weitergeleitet.

ID	Typ	Beschreibung	Sichtbarkeit
welcomeHeading	outputText	Überschrift der Welcomepage.	Alle
welcomeText	outputText	Bewerben der Applikation mithilfe einer Kurzbeschreibung.	Alle
email	inputText	Textfeld für E-Mail Eingabe.	Alle
password	inputSecret	Textfeld für Passwort Eingabe.	Alle
login	commandButton	Ausführen des Login-Prozesses.	Alle
register	link	Weiterleitung zur Registrierung.	Alle

**registration.xhtml** Seite zur Registrierung anonymer Nutzer.

ID	Typ	Beschreibung	Sichtbarkeit
titleLabel	outputLabel	Label für Titel.	AN
title	inputText	Angabe eines Titels.	AN
firstNameLabel	outputLabel	Label für Vorname.	AN
firstName	inputText	Angabe des Vornamens.	AN
nameLabel	outputLabel	Label für Nachname.	AN
name	inputText	Angabe des Nachnamen.	AN
passwordLabel	outputLabel	Label für Passwort.	AN
password	inputSecret	Angabe eines Passwortes.	AN
emailLabel	outputLabel	Label für E-Mail.	AN
email	inputText	Angabe einer validen E-Mail.	AN
register	commandButton	Ausführen des Registrations-Prozesses.	AN
welcome-page	link	Weiterleitung zur Login Seite.	AN

*Vollständig anhand spezifiziert? DFOs?*



entweder:  
Button?

**verification.xhtml** Wird bei erfolgreicher Verifikation der E-Mail angezeigt.

ID	Typ	Beschreibung	Sichtbarkeit
successText	outputText	Text zur erfolgreichen Verifizierung der E-Mail	Alle
countdown	timer	Timer bis zur Weiterleitung auf die Homepage.	Alle

besten homepage  
besteht aus  
Reiter  
und für jeden  
Reiter eigene  
Beschränkungstabelle

**homepage.xhtml** Startseite die den Überblick über alle Einreichungen beinhaltet.

ID	Typ	Beschreibung	Sichtbarkeit
yourSub- mission	commandLink	Reiter zum Anzeigen der eigenen Einreichungen.	Alle
yourRe- views	commandLink	Reiter zum Anzeigen der zu begutachtenden Einreichungen	G
editorial- Overview	commandLink	Reiter zum Anzeigen aller Einreichungen, welche man editiert.	E
filterLabel	outputLabel	Label für Filter	Alle
stateLabel	outputLabel	Label für Filter vom Status der Einreichung	Alle
stateSelect	selectOneMenu	Filtern des Status von Einreichungen	Alle
dateLabel- Before	outputLabel	Label für Filter nach einem Datum.	Alle
dateBefore	inputText	Eingabe eines <i>spätestens</i> Datum	Alle
dateLabel- After	outputLabel	Label für Filter vor einem Datum.	Alle
dateAfter	inputText	Eingabe eines <i>frühestens</i> Datum	Alle
dateUp	commandButton	Einreichungen werden dem Datum nach aufsteigend angezeigt.	Alle
dateDown	commandButton	Einreichungen werden dem Datum nach absteigend angezeigt.	Alle
searchLa- bel	outputLabel	Label für Suche.	Alle
search	inputText	Durchsuchen der Tabellen mit Matching-Words.	Alle
apply	commandButton	Wende Filtereinstellungen an.	Alle
submis- sionTable	dataTable	Liste aller eignen Einreichungen.	Alle
reviewTa- ble	dataTable	Liste aller Gutachten.	G
editorTable	dataTable	Liste aller Einreichungen für einen Editor.	E
pagination	listPagination	Jede Tabelle enthält eine Paginierung.	Alle

Filter -  
hier abstrahieren

in listenpaginierung

kein:  
eigenes  
tag

so hat  
listenpaginierung  
sichergestellt  
für Paginierung

*(gut): extra Bedingung f. jede Tabelle bzw. den Spalten*

Die einzelnen Tabellen *submissionTable*, *reviewTable* und *editorTable* sind nach den gleichen Punkten gegliedert: Titel des Papers, Datum und Status der Einreichung und Name des zugehörigen Forums. *Miss spalte das 3x fehlt weil, wegen verschiedenen Bindungen*

ID	Typ	Beschreibung	Sichtbarkeit
title	Link	Titel des Papers und Weiterleitung zur Einreichung	Alle
date	outputText	Datum der Einreichung.	Alle
state	outputText	Status der Einreichung.	Alle
forum	Link	Name des zugehörigen Forums und Weiterleitung zum Forum.	Alle

*sehen das in Master-Funktion*  
**submission.xhtml** ist die Übersichtsseite einer Abgabe. Hier werden alle mit der Einreichung verbundenen Aktivitäten abgebildet. Für den **Editor** und den **Administrator** wird zusätzlich eine *toolbar.xhtml* eingebunden. *→ extra include -Eintrag*

ID	Typ	Beschreibung	Sichtbarkeit
title	outputText	Titel des Papers.	Alle
forumLabel	outputLabel	Label für ein Forum.	Alle
forum	Link	Name des Forums und Weiterleitung zur Forumsübersicht.	Alle
authorLabel	outputLabel	Label für Autor.	Alle
author	Link	Name des Autors und Weiterleitung zum Profil.	Alle
coAuthorLabel	outputLabel	Namen der Co-Autoren.	Alle
emailLabel	outputLabel	Label für E-Mail-Adressen der Co-Authoren.	Alle
email	Link	E-Mail-Adressen der Co-Authoren und Mailto-Link.	Alle
filterLabel	outputLabel	Label für Filter	Alle
dateLabel	outputLabel	Label für Filter von Datum.	Alle
dateUp	commandButton	Einreichungen werden aufsteigend angezeigt.	Alle
dateDown	commandButton	Einreichungen werden absteigend angezeigt.	Alle
reviewStateLabel	outputLabel	Label für Filtern der Gutachtenfreischaltung.	E,A
all	checkbox	Anzeigen aller Gutachten.	E,A,G
allLabel	outputLabel	Label für all.	E,A,G
locked	checkbox	Anzeigen nicht freigeschalteter Gutachten.	E,A,G
lockedLabel	outputLabel	Label für locked.	E,A,G
unlocked	checkbox	Anzeigen freigeschalteter Gutachten.	E,A,G

unlocked-Label	outputLabel	Label für unlocked.	E,A,G
reviewSubmitLabel	outputLabel	Label für Filter von eingereichten Gutachten.	E,A
submitted	checkbox	Anzeigen aller abgegebenen Gutachten.	E,A
submitted-Label	outputLabel	Label für submitted.	E,A
notSubmitted	checkbox	Anzeigen noch nicht abgegebener Gutachten.	E,A
notSubmittedLabel	outputLabel	Label für notSubmitted.	E,A
versionLabel	outputLabel	Label für Versionsfilter.	Alle
version	selectOneMenu	Auswahl der Versionsnummern	Alle
reviewer-Label	outputLabel	Label für Reviewerfilter.	Alle
reviewer	selectOneMenu	Auswahl eines bestimmten Gutachters.	E,A
recommendationLabel	outputLabel	Label für Empfehlungen.	E,A,G
recommendation	selectOneMenu	Auswahl der Art von Empfehlungen.	E,A,G
searchLabel	outputLabel	Label für Suche.	Alle
search	inputText	Durchsuchen der Tabelle mit Matching-Words.	Alle
apply	commandButton	Wende Filtereinstellungen an.	Alle
submissionTable	dataTable	Liste aller Versionen	Alle
reviewTable	dataTable	Liste aller/der eigenen Gutachten.	E,A,G
pagination	listPagination	Jede Tabelle enthält eine Paginierung.	Alle

Mehrere  
Tabelle pro  
Facet  
Vorschau  
(visuell und  
zu Funktionen)

*submissionTable* enthält die Spalten: Version, Datum, Deadline, Status und Download. Die Elemente der Liste, der unterschiedlichen Versionen der Einrichtung, sieht wie folgt aus:

ID	Typ	Beschreibung	Sichtbarkeit
version	outputText	Versionsnummer der Einreichung	Alle
date	outputText	Datum der Einreichung	Alle
deadline	outputText	Deadline der Revision	Alle
state	radioButton	Status der Einreichung	Alle
pdf	inputFile	Download der Einreichung	Alle

*reviewTable* enthält die Spalten: Version, Gutachter, Datum, Deadline, Status, Empfehlung, Kommentar und Download. Die Elemente der Liste von Gutachten ist wie folgt:

ID	Typ	Beschreibung	Sichtbarkeit
version	outputText	Versionsnummer des Gutachtens.	E,A,G
reviewer	outputText	Namen des Gutachters.	E,A,G
date	outputText	Datum des Gutachtens.	E,A,G
deadline	outputText	Deadline des Gutachtens.	E,A,G
state	commandButton	Freigabe des Gutachtens	E,A
stateText	outputText	Freigabestatus des Gutachtens	G
recomen- dation	checkbox	Empfehlung eines Gutachters	E,A,G
comment	outputText	Kommentar des Gutachters.	E,A,G
pdf	commandButton	Download der Einreichung.	E,A,G

**newSubmission.xhtml** Hier können neue Paper eingereicht werden.

ID	Typ	Beschreibung	Sichtbarkeit
subName- Label	outputLabel	Label für Papertitel.	N
submissi- onName	inputText	Name des abzugebenden Papers.	N
forumLabel	outputLabel	Label für Name des Forums.	N
forumNa- me	inputText	Name des Forums, bei welchem ab- gegeben wird.	N
editorLabel	outputLabel	Label für Editor.	N
editorSe- arch	selectOneMenu	Angabe eines Editors	N
pdfOutput	outputText	Angabe des ausgewählten Dateina- mens.	N
pfdUpload	inputFile	Angabe der Abgabedatei.	N
<del>titleLabel</del>	<del>outputLabel</del>	<del>Label für Titel</del>	N
<del>titel</del>	<del>inputText</del>	<del>Angabe eines Titels.</del>	N
<del>coAuthor- Label</del>	<del>outputLabel</del>	<del>Label für Co-Autor.</del>	N
coAuthor- FirstLabel	outputLabel	Label für Vorname des Co-Autors.	N
coAuthor- FirstName	inputText	Angabe des Vornamen eines Co- Autors.	N
coAuthor- LastLabel	outputLabel	Label für Nachname des Co-Autors.	N
coAuthor- LastName	inputText	Angabe des Namen eines Co- Autors.	N
coAuthorE- mailLabel	outputLabel	Label für E-Mail des Co-Autors.	N
coAuthorE- Mail	inputText	Angabe der E-Mail eines Co- Autors.	N

Nicht  
aufrufen,  
sondern  
einmal per  
Konversion  
aufrufen, dann  
es selbst  
gibt

gut; flexibel bel. viele Co-Autoren

Seite für Gutachten annehmen?  
→ eigenes Label für Views-URL-Paradigm

submitCo-Author	commandButton	Ausgabe des Co-Autors in einer Liste.	N
coAuthor-List	outputText	Anzeige aller Co-Autoren in einer Liste.	N
deleteCo-Author	commandButton	Löschen des zugehörigen Co-Autors.	N
submit	commandButton	Ausführen des Abgabe-Prozesses.	N

**newReview.xhtml** Möchte ein Gutachter eines Papers seine Beurteilung abgeben, so ist dies auf dieser Seite möglich.

ID	Typ	Beschreibung	Sichtbarkeit
<del>versionLabel</del>	<del>outputLabel</del>	Label für Versionsnummer	G,A
versionNumber	outputText	Versionsnummer des eingereichten Papers, zu welchem der Gutachter ein Gutachten abgeben kann.	G,A
recommendationLabel	outputLabel	Label für eine Empfehlung.	G,A
recommendation	checkbox	Empfehlung des Gutachters ein Paper zu akzeptieren.	G,A
commentLabel	outputLabel	Label für einen Kommentar	G,A
comment	inputTextarea	Kommentar eines Gutachters	G,A
pdfLabel	outputLabel	Label für Datei	G,A
reviewPdf	inputFile	Einzureichendes Gutachten.	G,A
submitReview	commandButton	Gutachten einreichen.	G,A

**resultList.xhtml** zeigt alle Ergebnisse einer Suche an.

Wird nach den eigenen Einreichungen, nach denen die man begutachtet oder nach Einreichungen in eigener editorialen Verantwortung gesucht, so erscheinen die Listen im gleiche Schema wie auch auf der Homepage. Ergibt die Suche eine Liste von User, so wird diese wie in der Userliste für den Administrator und dem Editor angezeigt. Ist das Ziel der Suche, wissenschaftliche Foren zu finden so erhält man bei passender Eingabe eine Liste im selben Format wie diejenige auf der Seite der Forumsliste.

Johann Schicho

**scientificForumList.xhtml** Hier erhalten die Nutzer eine Übersicht über alle wissenschaftliche Foren.

ID	Typ	Beschreibung	Sichtbarkeit
<del>filterLabel</del>	<del>outputLabel</del>	Label für Filter	Alle
scFieldsLabel	outputLabel	Label für Filter der Themengebiete	Alle

per Konvention, S.O.

das kann groß werden...

scFiel- dSelect	selectOneMenu	Filtern der Themengebiete der Foren	Alle
dead- lineLabel	outputLabel	Label für Filtern nach Deadlines (Abgelaufen/Zukunft)	Alle
deadlin- eSelect	selectOneMenu	Filtern der nach Deadlines	Alle
dateLabel- Before	outputLabel	Label für Deadline nach einem Datum.	Alle
dateBefore	inputText	Eingabe eines <i>spätestens</i> Datum	Alle
dateLabel- After	outputLabel	Label für Deadline vor einem Datum.	Alle
dateAfter	inputText	Eingabe eines <i>frühestens</i> Datum	Alle
searchLa- bel	outputLabel	Label für Suche.	Alle
search	inputText	Durchsuchen der Tabelle mit Matching-Words.	Alle
apply	commandButton	Wende Filtereinstellungen an.	Alle
nameUp	commandButton	Einreichungen werden aufsteigend nach Namen angezeigt.	Alle
nameDown	commandButton	Einreichungen werden absteigend nach Namen angezeigt.	E,A
deadlineUp	commandButton	Einreichungen werden aufsteigend nach Deadline angezeigt.	Alle
deadline- Down	commandButton	Einreichungen werden absteigend nach Deadline angezeigt.	Alle
forumsTa- ble	dataTable	Liste aller Foren im System.	Alle
pagination	listPagination	Die Tabelle enthält eine Paginierung.	Alle

in  
linken-  
abstraktion

so hier  
keine  
Kontextabstraktion

jetzt sagt noch Wechsel des Verantwortungs Inhalts aus Facilit

Stefanie Gürster

Die Liste aller Foren im System wird in folgende Bereiche unterteilt: Name des Forums, Deadline und Kategorie. Dabei beinhaltet jede Zeile dieser Liste folgende Elemente:

ID	Typ	Beschreibung	Sichtbarkeit
name	link	Name des Forums.	Alle
deadline	outputText	Abgabefrist des jeweiligen Forums.	Alle
science- Field	outputText	Kategorie des Forums.	Alle

**profile.xhtml** Die Profilseite eines angemeldeten Nutzers ist nur vom Nutzer selbst und vom Administrator editierbar. Ein Editor besitzt nur Leserechte. Alle zuvor gespeicherten Angaben werden in den jeweiligen Feldern angezeigt.

ID	Typ	Beschreibung	Sichtbarkeit
----	-----	--------------	--------------

avatar	graphicImage	Aktueller Avatar.	N,A,E
newAvatar	inputFile	Hochladen eines neuen Avatars.	N,A,E
deleteAvatar	commandButton	Löschen des Avatars.	N,A,E
roleLabel	outputLabel	Label für Rolle.	N,A,E
role	outputText	Angabe der Rolle eines Nutzers	N,A,E
isAdminLabel	outputLabel	Label für Adminrolle.	A
isAdmin	checkbox	Zuteilung der Rolle des Administrators.	A
titleLabel	outputLabel	Label für Titel	N,A,E
title	inputText	Titel des Nutzenden.	N,A,E
firstNameLabel	outputLabel	Label für Vorname	N,A,E
firstName	inputText	Vorname des Nutzenden.	N,A,E
nameLabel	outputLabel	Label für Nachname	N,A,E
name	inputText	Nachname des Nutzenden.	N,A,E
passwordLabel	outputLabel	Label für Passwort.	N,A
password	inputSecret	Leeres Eingabefeld für ein neues Passwort.	N,A
emailLabel	outputLabel	Label für E-Mail.	N,A,E
email	inputText	E-Mail des Nutzenden.	N,A,E
submissionNumberLabel	outputLabel	Label für Anzahl von Einreichungen.	N,A,E
submissionNumber	outputText	Anzahl der eigenen Einreichungen	N,A,E
employerLabel	outputLabel	Label für Arbeitgeber	N,A,E
employer	inputText	Angabe des Arbeitgebers	N,A,E
scienceFieldLabel	outputLabel	Label für Fachgebiet	N,A,E
scienceField	outputText	Liste aller Fachgebiete	N,A,E
addScienceField	inputText	Spezialgebiete des Nutzers	N,A,E
addFieldButton	commandButton	Füge Fachgebiet zur Liste hinzu.	N,A,E
dateOfBirthLabel	outputLabel	Label für Geburtstag.	N,A
dateOfBirth	inputText	Angabe des Geburtstages	N,A
save	commandButton	Daten werden persistiert.	N,A,E
delete	commandButton	Nutzer und alle damit verbundenen Daten werden gelöscht.	N,A,E

*Handwritten:* Funktion: Neu-Zuweisung der Rolle  
gibt es jeweils 1?

Sollte ein Administrator einem anderen Nutzer die Rolle des Administrators zuweisen oder entziehen, so erscheint beim Auslösen des Save-Buttons ein Popup-Dialog. In diesem Fenster wird der Administrator angewiesen, die Änderung mit seinem Passwort zu bestätigen.

✓

ID	Typ	Beschreibung	Sichtbarkeit
password	inputSecret	Angabe eines Passworts.	A
reference	outputText	Hinweis zur Veränderung der Adminrolle.	A
abort	commandButton	Abbruch der Änderung.	A
save	commandButton	Speichern aller Änderungen.	A

**errorPage.xhtml** Auf diese Seite wird navigiert, wenn auf eine nicht existierende URL oder auf eine URL zugegriffen wird, auf die man keine Zugriffsrechte hat.

ID	Typ	Beschreibung	Sichtbarkeit
errorMessage	outputText	Anzeige einer Fehlermeldung.	Alle

*Handwritten:* ✓ Strukturauswahl im Development-Modus von JSF

Johann Schicho

**userList.xhtml** Editoren und Administratoren erhalten hier einen Überblick über alle Nutzer. Die Seiten sind paginiert.

ID	Typ	Beschreibung	Sichtbarkeit
filterLabel	outputLabel	Label für Filter	E,A
roleLabel	outputLabel	Label für Filter Rolle des Nutzers	E,A
roleSelect	selectOneMenu	Filtern der Rolle des Nutzers	E,A
searchLabel	outputLabel	Label für Suche.	Alle
search	inputText	Durchsuchen der Tabelle mit Matching-Words.	Alle
apply	commandButton	Wende Filtereinstellungen an.	E,A
nameUp	commandButton	Nutzer werden nach Namen aufsteigend angezeigt.	E,A
nameDown	commandButton	Nutzer werden absteigend nach Namen angezeigt.	E,A
roleUp	commandButton	Nutzer werden nach Rolle aufsteigend angezeigt.	E,A
roleDown	commandButton	Nutzer werden absteigend nach Rolle angezeigt.	E,A
userTable	dataTable	Liste aller Nutzer im System.	E,A
pagination	listPagination	Die Tabelle enthält eine Paginierung.	E,A



Stefanie Gürster

Für die Benutzerliste sind die folgenden Elemente vorgesehen, wobei diese in die Bereiche: Nutzerrolle, Name, E-Mail-Adresse, Arbeitgeber und Fachgebiete aufgegliedert ist.

ID	Typ	Beschreibung	Sichtbarkeit
role	outputText	Rolle des Nutzers	E,A
name	link	Name des Nutzers.	E,A
email	link	E-Mail-Adresse des Nutzers.	E,A
employee	outputText	Arbeitgeber des Nutzers.	E,A
science-Field	outputText	Fachgebiete des Nutzers.	E,A

*redundant* ~~newUser.xhtml~~ *gleiches wie zum Erstellen wie bei normalem Nutzer* Der Administrator kann hier einen neuen Nutzer anlegen.

ID	Typ	Beschreibung	Sichtbarkeit
password-Label	outputLabel	Label für Passwort.	A
password	inputSecret	Angabe des Passworts.	A
firstLabel	outputLabel	Label für Vorname.	A
firstName	inputText	Angabe des Vornamen.	A
nameLabel	outputLabel	Label für Nachname	A
name	inputText	Angabe des Nachnamen.	A
titleLabel	outputLabel	Label für Titel	A
titel	inputText	Angabe eines Titels.	A
emailLabel	outputLabel	Label für E-Mail.	A
email	inputText	Angabe einer E-Mail-Adresse.	A
isAdmin-Label	outputLabel	Label für Adminrolle.	A
isAdmin	checkbox	Zuordnung einer Administratorenrolle.	A
abort	commandButton	Abbruch des Vorgangs: Nutzererstellen.	A
save	commandButton	Speichern des neuen Nutzers.	A

**administration.xhtml** Hier kann ein Administrator Konfigurationen vornehmen.

ID	Typ	Beschreibung	Sichtbarkeit
themeHeading	outputText	Überschrift für Farbthemes.	A
orangeTheme	outputLabel	Standart Theme der Applikation.	A
orange	radioButton	Auswahl des Standart Themes.	A
darkTheme	outputLabel	Dark Theme der Applikation.	A
dark	radioButton	Auswahl des Dark Themes.	A
blueTheme	outputLabel	Blue Theme der Applikation	A

fließt der ComboBox die CSS-Files in best. Directory listet

blue	radioButton	Auswahl des Blue Themes.	A
greenTheme	outputLabel	Green Theme der Applikation.	A
green	radioButton	Auswahl des Green Themes.	A
logoHeading	outputText	Überschrift für Logo-Bereich.	A
logo	graphicImage	Logo des Systems.	A
changeLogo	inputFile	Hochladen eines neuen Logos.	A
institutionHeading	outputText	Überschrift für Einstellungen der Einrichtung.	A
institution	inputText	Angabe des Namens der Einrichtung.	A
imprint	inputTextArea	Angabe des Impressums der Einrichtung.	A
abort	commandButton	Abbruch des Änderungsvorgangs.	A
save	commandButton	Speichern der Änderungen.	A
newUser	link	Weiterleitung zu <i>newUser.xhtml</i> .	A
newScientificForum	link	Weiterleitung zu <i>newScientificForum.xhtml</i> .	A

→ gleiches wie beim Editieren?

**newScientificForum.xhtml** Auf der Seite zum Erstellen eines wissenschaftlichen Forums kann ein Administrator dessen wesentlichen Daten festlegen.

ID	Typ	Beschreibung	Sichtbarkeit
forumName	inputText	Angabe des Name des Forums.	A
fName	outputLabel	Label Name des Forums.	A
editorLabel	outputLabel	Label für Editorauswahl.	A
emailEditor	inputText	Angabe der E-Mail-Adresse eines Editor.	A
emailEditorLabel	outputLabel	Label für E-Mail-Adresse eines Editors.	A
addEditor	commandButton	Hinzufügen eines Editors in eine Liste.	A
deleteEditor	commandButton	Löschen des zugehörigen Editors von der Liste.	A
deadline	inputText	Hinzufügen einer Deadline des Forums.	A
deadlineLabel	outputLabel	Label für Deadline des Forums.	A
description	inputTextArea	Angabe einer Kurzbeschreibung.	A
descriptionLabel	outputLabel	Label für Kurzbeschreibung.	A
url	inputText	Link zur Konferenz oder zum Journal.	A

entl. viele...

urlLabel	outputLabel	Label für URL des Forums.	A
instruction-Label	outputLabel	Label für Anleitung	A
reviewInstructions	inputTextArea	Angabe einer Anleitung für eine Begutachtung.	A
scienceFieldLabel	outputLabel	Label für Fachgebiete	A
science-Field	selectOneMenu	Auswahl von Fachgebieten.	A
scienceFieldLabel	outputLabel	Label für Fachgebiete.	A
newScienceField	inputText	Hinzufügen neuer Fachgebiete.	A
addScienceField	commandButton	Ausführen der Hinzufüge-Aktion.	A
save	commandButton	Speichern des neuen Forums.	A
abort	commandButton	Abbruch des Erstellungsprozesses.	A

**editForum.xhtml** Hier kann der Administrator eine schon existierende Seite bearbeiten.

ID	Typ	Beschreibung	Sichtbarkeit
forumName	inputText	Angabe des Name des Forums.	A,E
forumNameLabel	outputLabel	Label Name des Forums.	A,E
editorLabel	outputLabel	Label für Editorauswahl.	A,E
emailEditor	inputText	Angabe der E-Mail-Adresse eines Editor.	A,E
emailEditorLabel	outputLabel	Label für E-Mail-Adresse eines Editors.	A,E
addEditor	commandButton	Hinzufügen eines Editors in eine Liste.	A,E
deleteEditor	commandButton	Löschen des zugehörigen Editors von der Liste.	A,E
deadline	inputText	Hinzufügen einer Deadline des Forums.	A,E
deadlineLabel	outputLabel	Label für Deadline des Forums.	A,E
description	inputTextArea	Angabe einer Kurzbeschreibung.	A,E
descriptionLabel	outputLabel	Label für Kurzbeschreibung.	A,E
url	inputText	Link zur Konferenz oder zum Journal.	A,E
urlLabel	outputLabel	Label für URL des Forums.	A,E
reviewInstructLabel	outputLabel	Label für Anleitung einer Begutachtung.	A,E

reviewInstructions	inputTextArea	Angabe einer Anleitung für eine Begutachtung.	A,E
scienceFieldOut	outputText	Liste der Fachgebiete.	A,E
scienceField	selectOneMenu	Auswahl von Fachgebieten.	A,E
scienceFieldLabel	outputLabel	Label für Fachgebiete.	A,E
newScienceField	inputText	Hinzufügen neuer Fachgebiete.	A
addScienceField	commandButton	Ausführen der Hinzufüge-Aktion.	A
save	commandButton	Speichern des Änderungen.	A,E
abort	commandButton	Abbruch des Bearbeitungsprozesses.	A,E
delete	commandButton	Löscht das Forum.	A

*Glück wie bei Schülern/ Lehren?*  
**scientificForum.xhtml** Die Ansicht eines Forums dient zur Ausgabe von Informationen über die jeweilige Konferenz oder das jeweilige Journal.

ID	Typ	Beschreibung	Sichtbarkeit
forumName	outputText	Name des Forums.	Alle
editor	outputText	Liste der verantwortliche Editoren.	Alle
deadline	outputText	Deadline des Forums.	Alle
deadlineLabel	outputLabel	Label für Deadline des Forums.	Alle
description	outputText	Kurzbeschreibung des Forums.	Alle
descriptionLabel	outputLabel	Label für Kurzbeschreibung.	Alle
url	outputLink	Link zur Konferenz oder zum Journal.	Alle
urlLabel	outputLabel	Label für URL des Forums.	Alle
reviewInstructions	outputText	Anleitung für eine Begutachtung.	G,E,A
scienceField	outputText	Fachgebiet des Forums.	Alle
scienceFieldLabel	outputLabel	Label für Fachgebiete.	Alle
filterLabel	outputLabel	Label für Filter	G,E,A
stateLabel	outputLabel	Label für Filter vom Status der Einreichung	G,E,A
stateSelect	selectOneMenu	Filtern des Status von Einreichungen	G,E,A
dateLabel	outputLabel	Label für Filter von Datum.	G,E,A

searchLabel	outputLabel	Label für Suche.	Alle
search	inputText	Durchsuchen der Tabelle mit Matching-Words.	Alle
apply	commandButton	Wende Filtereinstellungen an.	G,E,A
dateUp	commandButton	Einreichungen werden aufsteigend angezeigt.	G,E,A
dateDown	commandButton	Einreichungen werden absteigend angezeigt.	G,E,A
submissionTable	dataTable	Liste aller Einreichungen, welche der Nutzer bearbeitet.	Alle
pagination	listPagination	Die Tabelle enthält eine Paginierung.	Alle

**imprint.xhtml** Das Impressum gibt die vom Administrator angegebenen Kontaktdaten des Betreibers wieder.

ID	Typ	Beschreibung	Sichtbarkeit
imprintHeading	outputText	Überschrift der Ansicht	A
imprint	outputText	Impressum des Betreibers.	A

## 5 Systemfunktionen

Thomas Kirz

### 5.1 Logging

Die Java Logging API wird benutzt, um bei bestimmten Ereignissen Meldungen in eine Logdatei zu schreiben. Dabei kann man verschiedene *Loglevel* einstellen, um nur Meldungen mit bestimmter Wichtigkeit zu speichern. Es gibt folgende Loglevel, wobei jedes Level jeweils die Nachrichten des vorherigen Levels beinhaltet:

**SEVERE** Sehr wichtige Ereignisse, die eine reguläre Benutzung verhindern und deren Auswirkungen auch für Nutzer verständlich sind.

**WARNING** Wichtige Ereignisse, die zu Problemen führen können und für Administratoren und Endnutzer relevant sind.

**INFO** Für den Administrator relevante Meldungen im normalen Betrieb. Diese sind auch fürs Entwickeln und Testen hilfreich.

### 5.2 Sicherheit

Für die häufigsten Sicherheitsschwachstellen bei Webanwendungen, die für LasEs relevant sind, wird im Folgenden erklärt, wie ausnutzende Angriffe verhindert werden.

**Fehlerhafte Zugangskontrolle** Um unerlaubten Zugriff auf Seiten, Informationen und Aktionen zu verhindern, werden Zugriffe auf nicht-öffentliche Ressourcen (das

ist alles außer Start-, Registrierungs- und Verifikationsseite) standardmäßig zuerst abgelehnt. Die Klasse `TrespassListener` entscheidet dann, ob der Nutzer die nötigen Rechte hat und akzeptiert eine Anfrage nur dann, sonst wird auf eine Fehler-Seite weitergeleitet. Das hängt von der Rolle des Nutzers und von seiner Beziehung zur angefragten Ressource ab (ein normaler Nutzer kann z.B. nur sein eigenes Profil oder eine eigene Einreichung sehen). Da dies zentral in dieser Klasse passiert, entstehen weniger Fehler als bei der Prüfung an vielen verschiedenen Stellen des Programms.

✓ anhand was  
Gruppierung der Facets in einer Rolle  
i. Verzeile und schied  
u. d. URL

**Kryptographie-Probleme & Verlust der Vertraulichkeit sensibler Daten** Alle Anfragen und Antworten werden mit dem HTTPS-Protokoll mit TLS-Verschlüsselung durchgeführt. Dafür wird der Tomcat-Server, der diese Technologie unterstützt, entsprechend konfiguriert.

Passwörter werden mit der modernen und sicheren PBKDF2-Funktion gehasht. Dabei wird für jedes Passwort ein eigener zufällig generierter Salt benutzt. Eingebene Passwörter werden so früh wie möglich gehasht, damit die Klartextpasswörter eine minimale Lebenszeit auf dem Server haben. Dafür wird die robuste `javax.crypto`-Bibliothek verwendet.

✓ Klasse verwenden

**Injection & Cross-site-Scripting** Da JSF components für die HTML-Ausgabe verwendet werden, bereinigt JSF alle Nutzereingaben und macht Cross-site-Scripting unmöglich.

Für SQL-Code werden `PreparedStatement`s benutzt, um SQL-Injections zu verhindern.

**Session Hijacking** In der `web.xml`-Datei ist ein Timeout für Sessions konfigurierbar. Dieser wirkt Session Hijacking sowie unerwünschten physischen Zugriffen auf den Browser des Clients entgegen.

✓ das ist nicht  
bei login und wird neues cookie erzeugt werden

### 5.3 Asynchrone Threads

Für das periodische Aufräumen von nicht mehr benötigten Datenbankeinträgen (abgelaufene E-Mail-Verifikationslinks) gibt es einen eigenen Thread. Dieser wird von `business.internal.DBCleaner` gestartet.

→ Klasse für Thread?  
wie wird der regulär beendet?

### 5.4 Systemstart

Beim Systemstart wird das Logging initialisiert und die Konfigurationsdatei gelesen. Die Threads aus Kapitel 5.3 und der `ConnectionPool` werden gestartet.

~~ohne Schichten zu verstehen?~~

Dies passiert, wenn das Servlet bereit ist, dazu hört der `LifetimeListener` aus dem Paket `business.internal` mit der `@WebListener`-Annotation auf das `contextInitialized`-Event.

Systemmodell  
steht,  
das wir für diese  
Sitzung liegen

### 5.5 Systemstopp

Der Server kann über den `shutdown` port heruntergefahren werden. Dies stoppt auch den `Connection Pool` und alle bestehenden Datenbankverbindungen werden geschlossen. Diese Aufgaben führt der `business.internal.LifetimeListener` aus. Damit wird ein geordneter Shutdown sichergestellt.

↳ wann ist es vorbei  
↳ wann ist es vorbei wo Observer versprochen wird, oder ist das  
es beendet und nicht mit JSP-Events?

✓ shutdown hook

## 6 Datenfluss

Thomas Kirz

### 6.1 Hochladen einer Revision

Abbildung 2 zeigt die Interaktionen zwischen den Klassen beim Ablehnen einer Einreichung.

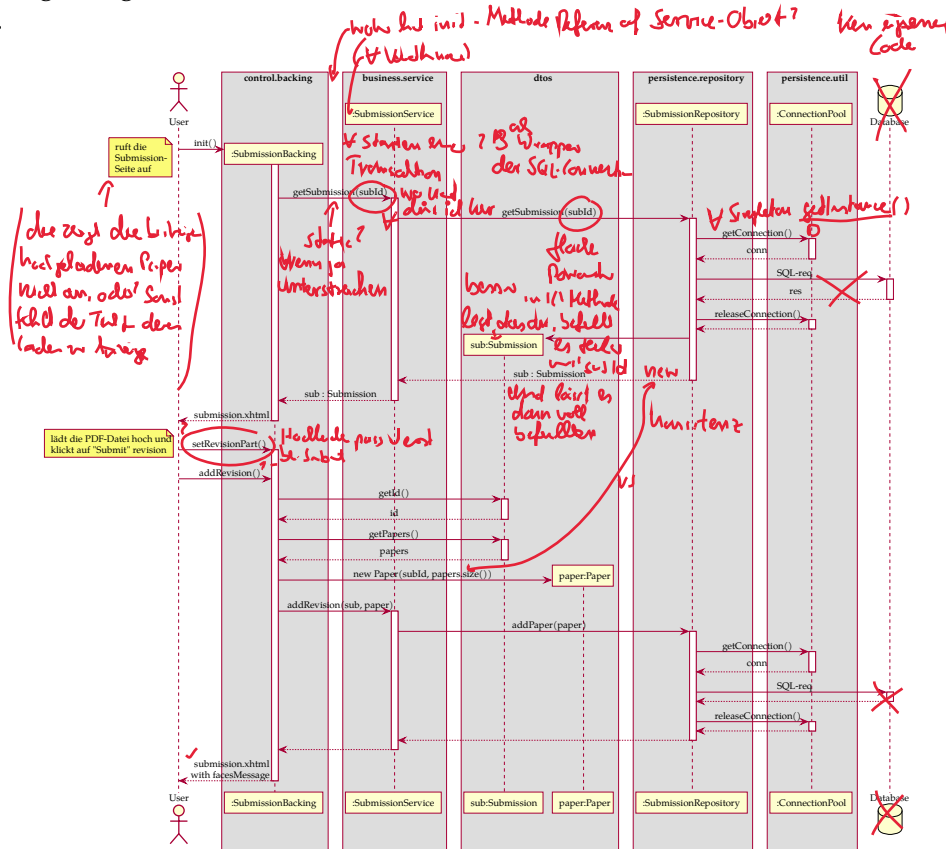


Abbildung 2: Sequenzdiagramm zur Ablehnung einer Einreichung

### 6.2 Ablehnung einer Einreichung

Abbildung 3 zeigt die Interaktionen zwischen den Klassen beim Ablehnen einer Einreichung, die Verbindung zum Datenbankserver schlägt aber fehl und die Einreichung kann nicht verändert werden. Es werden Exceptions bis zum Backing Bean geworfen und der Nutzer wird über den Fehler benachrichtigt.



[illegible]

→ Forward of Felleurte  
↳ ErrorDTO, ErrorBS, CustomExceptionHandler (Tryng)

## Johann Schicho

Im Entity-Relationship-Diagramm in Abbildung 4 sieht man die gesamte persistente Datenstruktur von LasEs. Die Kardinalität der *Relationships* gibt die ganzzahligen Abhängigkeiten zwischen der einzelnen *Entities* an. Die *is-A* Spezialisierungen modellieren die unterschiedlichen Subtypen einer Entität. Ein im Diagramm doppelt umrandetes Entity ist ein *schwaches* Entity. Das heißt, es existiert nur im Zusammenhang mit dem übergeordneten Entity. In unserem Fall kann beispielsweise kein Paper in der Datenbank existieren, ohne einer Einreichung zugeteilt zu sein.

lighting  
water

In Abbildung 4 werden die einzelnen Entities nochmal im Detail beschrieben. Mit der Notation /DXXX/ wird sich auf die im Pflichtenheft aufgeführten Produktdaten bezogen.

gut  
Vermische

**User** Ein Nutzer kann verschiedene Rollen wahrnehmen, diese werden durch die *is-A*-Spezifizierung modelliert. Die Attribute entsprechen den in /D010/ und /DW015/ angegebenen Daten. Hierbei gilt zusätzlich die Einschränkung, dass E-Mail-Adressen im System einzigartig sein müssen. Zusätzlich werden auch noch der Passwort-Hash und das zugehörige Salt zum generieren des Hashes gespeichert.

Wann man  
in der  
Beschreibung  
helfen kann



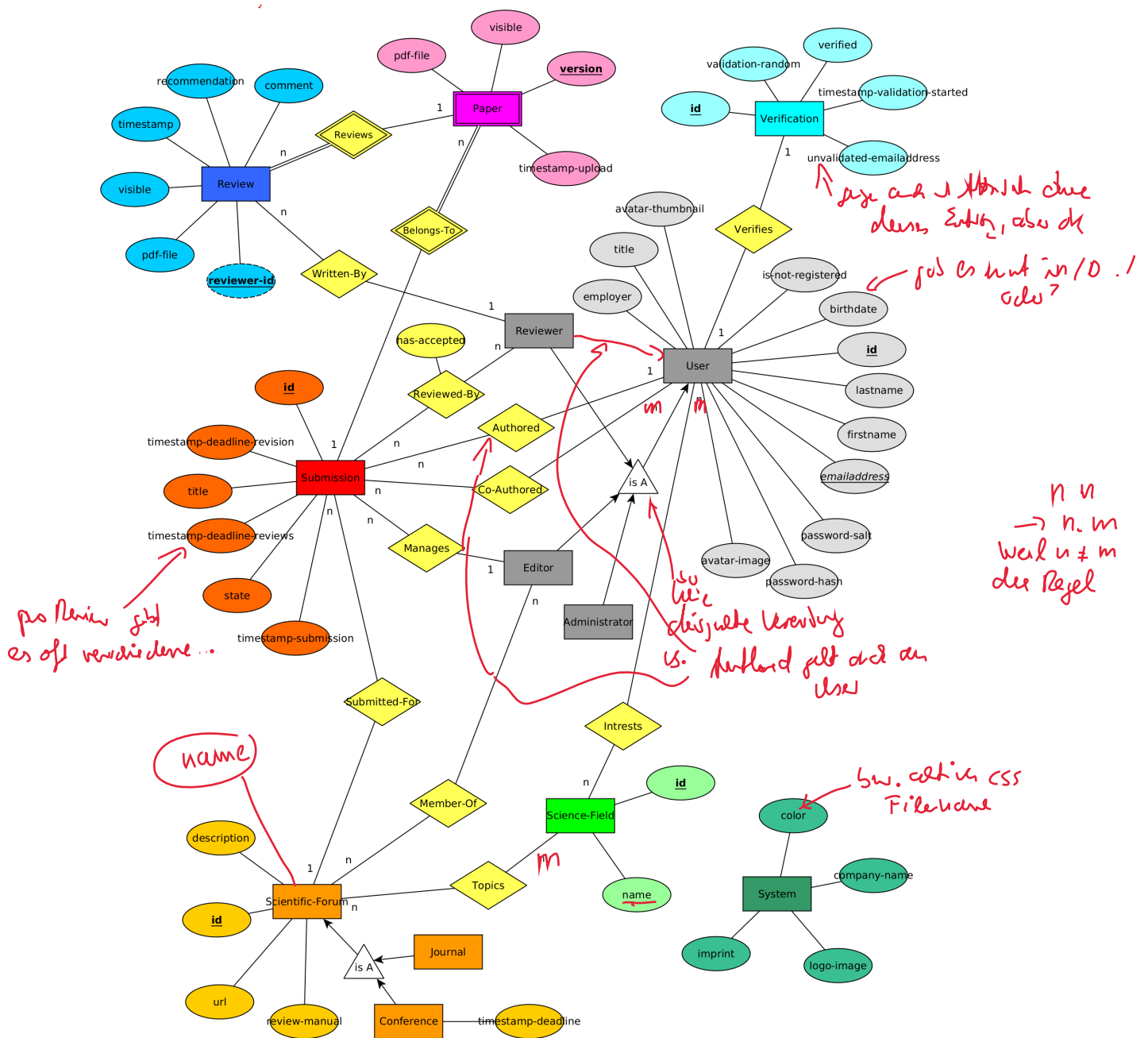


Abbildung 4: Übersicht über die Zusammenhänge zwischen den Entitäts

**Verification** Da jede E-Mailadressenänderung eine Verifizierung benötigt, werden hier die Daten, zu welchen geändert wird, gespeichert. Das *validation-random* ist dabei der String, welcher im Verifizierungslink per E-Mail versendet wird. ✓

**Paper** Ein Paper ist das in einem Einreichungsprozess hochgeladene Paper. Es ist modelliert als eine schwache Entität, abhängig von einer Einreichung, da ein Paper nicht im System existieren kann, ohne zu einem Einreichungsprozess zugeordnet zu werden. Es speichert die in /D020/ angegebenen Daten.

**Submission** Eine Einreichung speichert alle zu einem Einreichungsprozess gehörenden Daten, inklusive der Metadaten, wie den Einreicher und Ko-Autor, also alle in /D025/ und /DW021/ angegebenen Daten.

**Review** Ein Gutachten ist immer genau einem Paper zugeordnet, welches wiederum genau einer Einreichung zugeordnet ist. Es speichert die PDF Datei ab und alle zugehörigen Metadaten. Damit spiegelt es /D040/ wieder.

**Scientific Forum** Ein wissenschaftliches Forum besitzt die Referenz auf alle zu diesem Journal/Konferenz eingereichten Papers. Zusätzlich sind allgemeine Metadaten abgespeichert. Damit erfüllt es /D030/

↖ Verwrt: Bleib in DB oder  
Dateien  
im  
Filesystem?

**Scientific Field** Liste aller Fachgebiete bzw. Interessengebiete, die in LasEs verwendet werden. Diese werden als Fachgebieten der Foren und als Forschungsinteressen der Nutzer verwendet. Nach /D035/.

**System** Systemdaten speichern allgemeine Informationen über die Firma, die LasEs einsetzt. Sie sind in /D050/ spezifiziert. ✓