

(weitere) Verwendung von 3rd-Party Libs ohne deren "Genehmigungspflicht":
Elimination des Download-Servlets

Herr Kirz vgl. wenig Verantwortung

Feinspezifikation

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

19. November 2021

Inhaltsverzeichnis

1	Einleitung	1
2	Systemarchitektur	1
2.0.1	Paketdiagramm	1
2.0.2	Ordnerstruktur	2
3	Klassendiagramm	3
4	Bibliotheken	4
5	Konfiguration	4
5.1	Konfiguration der Anwendung	5
5.1.1	Konfiguration des Mailings	5
5.1.2	Konfiguration der Datenbank	6
5.1.3	Konfiguration des Logging	6
5.1.4	Sprachkonfiguration	7
6	Facelets	8
6.1	Namenskonvention	8
6.2	Templates	9
6.3	Seiten	9
6.3.1	Anonymer Nutzer	9
6.3.2	Angemeldeter Nutzer	9
6.3.3	Gutachter	9
6.3.4	Editor	9
6.3.5	Admin	9
7	DB Schema	9
7.1	Konvertierung des ER-Diagramms	50
7.1.1	SQL Standard	50
7.1.2	Normalform	50
7.1.3	Namenskonventionen	50
7.1.4	Relationship-Typen	50
7.1.5	Vererbung	50
7.2	Entities	51
7.3	Indizes	53
7.4	Relationships	53
8	Datenfluss	55
8.1	Hochladen einer Revision	55
8.2	Fehlerfall	56
9	Security	57
9.1	Cross Site Scripting (XSS)	57
9.2	SQL-Injection	57
9.3	Session Hijacking & Fixation	58
9.4	Passwörter	58
9.5	Information Leakage	58

9.6	Access Control	59
9.7	Vergabe von Administratorrechten	59

1 Einleitung

Johann Schicho

In diesem Dokument ist der Entwurf aus der vorherigen Projektphase genauer und detaillierter spezifiziert. Beispielsweise wird von grafischen Darstellungen, wie dem ER-Diagramm nun in *Data Definition Language* übersetzt. Das Klassendiagramm wird ebenfalls detaillierter beschrieben, indem nun auch Rückgabewerte und *JavaDoc* angegeben werden.

2 Systemarchitektur

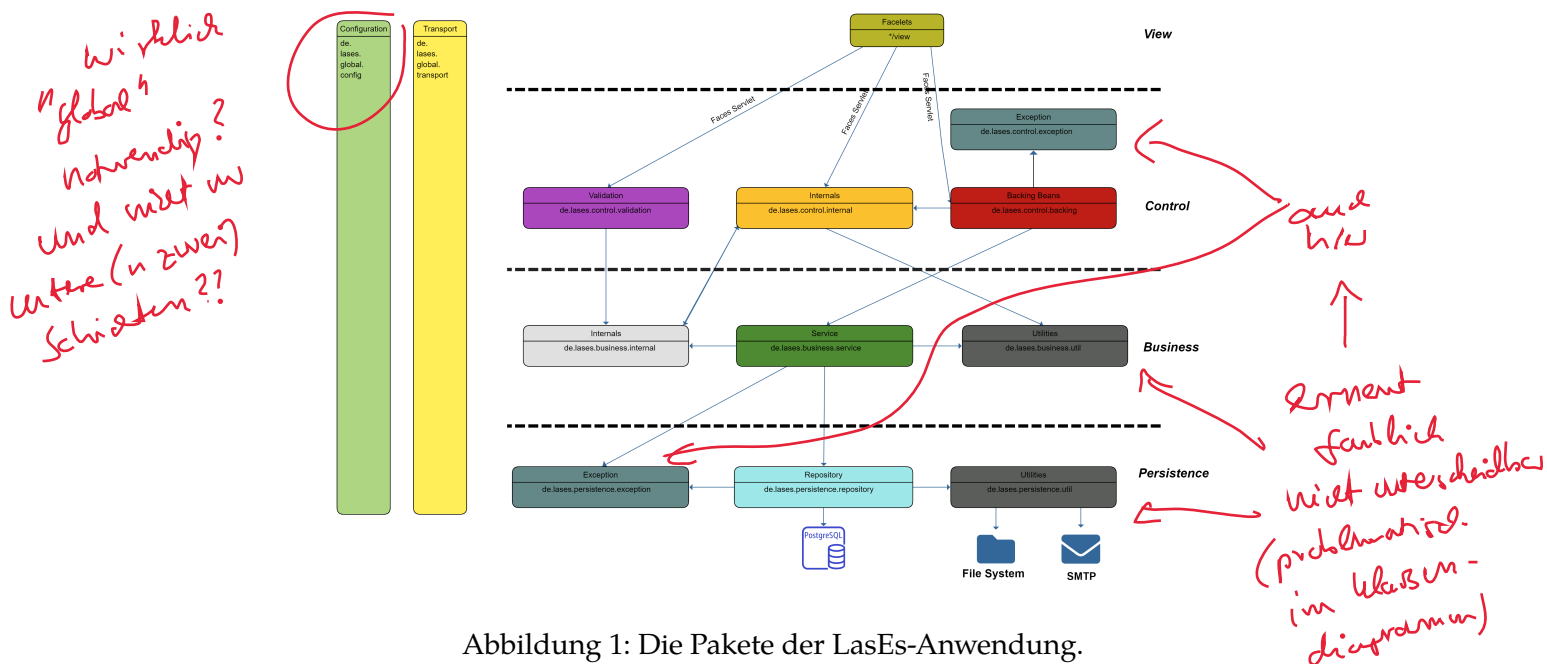
Johannes Garstenauer

2.0.1 Paketdiagramm

Die LasEs-Anwendung verwendet eine Vierschichtenarchitektur.

- Die *view*-Schicht enthält die Komponenten zur grafischen Darstellung.
- Die *control*-Schicht enthält die Komponenten zur Steuerung der grafischen Darstellung und Reaktion auf Nutzereingaben.
- Die *business*-Schicht enthält die Komponenten, welche die Anwendungslogik umsetzen.
- Die *persistence*-Schicht enthält die Komponenten zum Zugriff auf den Datenbankserver.

Diese Schichten folgen der *MVC-Architektur* wie in **Abbildung 1** dargestellt. Die Persistenz- und Businessschichten gehören hierbei dem *Model* an. Die Kontrollschicht stellt den *Controller* und die Faceletschicht die *View* dar.



Eine Schicht kommuniziert jeweils nur mit direkt angrenzenden Schichten. Diese Kommunikation (also der Datenfluss) erfolgt mittels *Data Transfer Objects*. Somit sind diese eine Ausnahme von der Schichtenarchitektur, da Sie sich keiner einzelnen Schicht zuordnen lassen. *DTOs* sind ein schichtenübergreifender *cross-cutting concern*.

~~2.0.2~~ Ordnerstruktur *! von was*

Die Ordnerstruktur dieser Anwendung ist ausgelegt auf klare Trennungen und Übersichtlichkeit. In Abbildung 1 werden Quellcodedateien ausgespart.

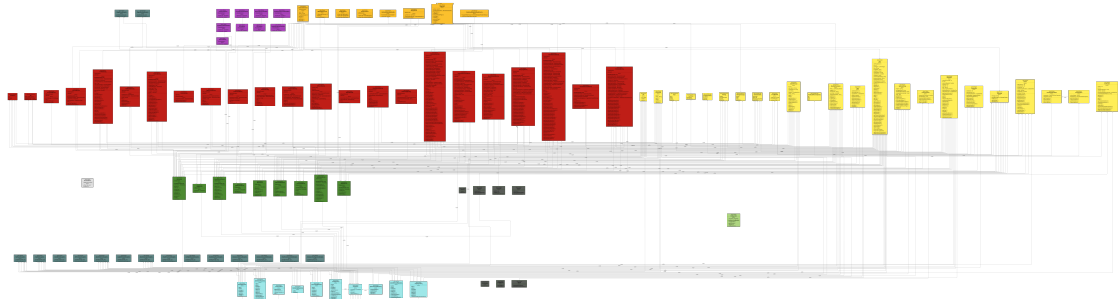


Abbildung 2: Die Ordner- und Dateistruktur der LasEs-Anwendung.

3 Klassendiagramm

- RSA: Sebastian Vogt
- Control Schicht: Sebastian Vogt
- Business Schicht: Johannes Garstenauer

- Persistence Schicht: Sebastian Vogt
- Cross-Cutting Concerns: Sebastian Vogt



Das *JavaDoc* befindet sich in der zip Datei *java-doc.zip*.

4 Bibliotheken

Johann Schicho

Zur Entwicklung und Verwendung von LasEs werden einige Softwarebibliotheken benötigt. Um keine weiteren Gebühren geltend machen zu müssen, werden nur Bibliotheken mit freien Lizenzen verwendet.

Name	Beschreibung	Lizenz
Mojarra v3.0.1	Jakarta Server Faces Referenzimplementierung der Eclipse Foundation	EPL v2.0
JBoss Weld v4.0.2.Final	Context and Dependency Injection Referenz Implementierung	Apache 2.0
Primefaces v10.0.0	UI Komponenten Bibliothek	MIT
Bootstrap v5.1.3	CSS Framework	MIT
PostgreSQL JDBC v42.3.1	Datenbank Treiber für PostgreSQL	BSD 2-Clause
Jakarta Mail API v2.0.1	Automatisierte E-Mail Versendung	EPL v2.0
JUnit v5.8.1	Unit Tests für Java Anwendungen	EPL v2.0
Mockito v4.0.0	Erzeugung von Mockups zum Testen	MIT
Selenium v4.0.0	Automatisierte Browsertests	Apache 2.0

5 Konfiguration

Johannes Garstenauer

JSF lässt sich auf verschiedenste Art & Weisen konfigurieren, wie beispielsweise mithilfe der *faces-config.xml*. Die LasEs-Anwendung bietet nebenher weitere selbstdefinierte Konfigurationsmöglichkeiten an. Die Konfigurationsdateien befinden sich im *webapp/WEB-INF/classes* Verzeichnis.

Klassendesign

private Antefakte besser nicht mit mit aufnehmen

Vgt. ToolBox Binding

- init() welche DTOs werden initialisiert?
- onLoad & View action? so nicht erlauben kann aufgeben
- wo Methode, die Parameter Eingabefeld für ^{z.B.} neue Editor "füllt"?

Review DTO
ConnectionState (für was jetzt??)

VerbindungsBdZ

- init() kann nicht URL parsieren lesen
- view Action darf nicht (erlaubt)

Exceptions: Default-Handler fehlt jeweils
meistens Stil "Firs" "is thrown" ...
Inge Sendet gibt es weiterhin vs. PF durchgezogen
Timeouts und Unvollständigkeit

Regel: Default - und das ist in
Menge der Stütz wissen immer das sein

Global Config

Config Reader (Daten fehlt)

init Cases?

z.B. Pfad für Config File

keine Methode auf Methode was gibt etwas zurück

Submissen oder Review DTO: \neq equals, clone etc?

Grantenwesen: Periodic Update

Avatar Util File als Wrapper um byte[]?

Hashing Util \neq Bit Länge, Hashverfahren, Buchstaben Länge z.B. für verschlüsselte i.D.B. wichtig

Scalst Länge

→ Implementieren was ist, was er machen soll

ReviewService - download File lädt File?

- add Review File-Parameter → i.w. wir wollen das hier fließen
Parameter (nur zur Steuerung)

Die Konfigurationsdateien werden intern in *Properties*-Pakete umgewandelt, wobei jedes *Properties*-Paket eine semantische Aufgabe übernimmt. So dient beispielsweise die *mail.properties* der Konfiguration des Mailing. Wenn ein Paket bestimmte *Properties* auslagert, so landen diese in der Anwendungskonfiguration *config.properties*.

5.1 Konfiguration der Anwendung

Die Anwendung kann auf einige Weisen personalisiert konfiguriert werden. Hierzu dient die *config.properties*-Datei, welche zentrale Parameter der Anwendung definiert.

```
# The configuration for the LasEs application.
# The convention to follow is: <key> = <value>.

#----- GENERAL -----#

# The maximum filesize for a pdf upload.
MAX_PDF_FILE_SIZE_MB = 20

# The maximum filesize for a image upload.
MAX_AVATAR_FILE_SIZE_MB = 5

# The maximum length of paginated lists.
MAX_PAGINATION_LIST_LENGTH = 25

#----- MAIL -----#

MAIL_ADDRESS_FROM = system@lases.de

#----- DATABASE -----#

# The URL with which the database can be accessed.
DB_URL = jdbc:postgresql://bueno.fim.uni-passau.de/sep21g02

# The database driver.
DB_DRIVER = org.postgresql.Driver

# The minimum amount of database connections provided
# by the ConnectionPool.
MIN_DB_CONNECTIONS = 10

# The maximum amount of database connections provided
# by the ConnectionPool.
MAX_DB_CONNECTIONS = 50
```

Listing 1: Die Anwendungskonfiguration *config.properties*

5.1.1 Konfiguration des Mailings

Damit die LasEs-Anwendung Mails versenden und Mailto-Links generieren kann müssen einige Konfigurationsentscheidungen getroffen werden. Hierzu dient die *mail.properties*-Datei.

```
# The configuration for the mailing API.
```

```
# The convention to follow is: <key> = <value>.
# For documentation see:
# https://jakarta.ee/specifications/mail/1.6/apidocs/index.html?com/
#   sun/mail/smtp/package-summary.html
```

2.0.1 wäre heute... - Schritt
siehe Bibliotheken

```
#----- SMTP -----#

mail.smtp.host = mail.fim.uni-passau.de

mail.smtp.port = 587

mail.smtp.auth = true  # Credentials

mail.smtp.starttls.enable = true
```

Listing 2: Die Mailingkonfiguration *mail.properties*

5.1.2 Konfiguration der Datenbank

aha → hier in config?

Damit die LasEs-Anwendung einen persistenten Datenbankzugriff besitzen kann, muss dieser konfiguriert werden. Nennenswert sind ebenfalls die Konfigurationsentscheidungen bezüglich der SSL-Verschlüsselung. Hierzu dient die *database.properties*-Datei.

```
# The configuration for the database.
# The convention to follow is: <key> = <value>.
# For documentation see:
# https://jdbc.postgresql.org/documentation/head/connect.html

#----- ACCESS -----#

# The username for database access.
user = sep21g02

# The password for database access.
password = ieQu2aeShoon

#----- SSL -----#

# Is SSL protection being used.
ssl = true

# The SSL factory used for SSL protection.
sslfactory = org.postgresql.ssl.DefaultJavaSSLFactory
```

Listing 3: Die Datenbankkonfiguration *database.properties*

5.1.3 Konfiguration des Logging

Damit die LasEs-Anwendung aussagekräftige *Log*-Dateien ausgeben kann, wird der *Logger* konfiguriert.

OT Prototyping: Einlesen des (obj.) Daten beim Start des Problems
→ da gibt es oft Probleme, z.B. den 1. Pfad zu finden
(soll nicht in Classpath liegen sondern unter WEB-INF)

```
# The configuration for the logger.
# For documentation see:
# https://docs.oracle.com/javase/7/docs/api/java/util/logging/FileHandler.html

#----- LOGGING -----#

# The handler for logging.
handlers= java.util.logging.FileHandler

# The level of logs that are to be logged. With 'Info' all logs
# with higher priority will also be logged.
.level= INFO

# The directory where the logs will be placed.
# We need help deciding where this should be.
java.util.logging.FileHandler.pattern = %h/lases%u.log
```

Listing 4: Die Loggingkonfiguration *logger.properties*

5.1.4 Sprachkonfiguration

Die Dateien, welche die dargestellten *Strings* enthalten, befinden sich im *src/main/resources*-Verzeichnis. Die initiale Standardsprache (Defaultwert Englisch) wird in der *faces-config.xml* im Verzeichnis *webapp/WEB-INF* festgelegt.

- *Resource Bundle 'label'* enthält solche *Strings*, welche als *Labels* für die Benutzeroberfläche verwendet werden.
- *Resource Bundle 'help'* enthält solche *Strings*, welche zum Anzeigen der *Tooltips* verwendet werden.
- *Resource Bundle 'message'* enthält solche *Strings*, welche zum Anzeigen von Hinweisen und Warnungen an den Nutzer verwendet werden.

Ein *Resource Bundle* enthält jeweils drei *.properties*-Dateien. Jeweils eine für:

- *Strings* in deutscher Sprache.
- *Strings* in englischer Sprache.
- sprachneutrale *Strings*.

Diese *Strings* werden nach der Konvention: **<key> = <value>** abgelegt, wobei der **<key>** in *camelCase* geschrieben ist.

Hier ein Beispiel mit einigen ausgewählten *key-value*-Paaren aus dem *Resource Bundle 'labels'*.

```
# Language-neutral translations of labels displayed in the UI.
# The convention to follow is: <key> = <value>.

appName = LasEs
```

Listing 5: Beispielhafter Ausschnitt aus *label.properties*

```
# German language translations of labels displayed in the UI.  
# The convention to follow is: <key> = <value>.
```

```
login = Anmelden  
register = Registrieren  
logout = Abmelden  
save = speichern  
acceptSubmission = Annehmen  
rejectSubmission = Ablehnen  
prevPage = Nächste Seite  
lastPage = Letzte Seite
```

Listing 6: Beispielhafter Ausschnitt aus *label_de.properties*

```
# English language translations of labels displayed in the UI.  
# The convention to follow is: <key> = <value>.
```

```
login = Login  
register = Register  
logout = Logout  
saveEditor = Save  
acceptSubmission = Accept  
rejectSubmission = Reject  
prevPage = Previous Page  
lastPage = Last Page
```

Listing 7: Beispielhafter Ausschnitt aus *label_en.properties*

6 Facelets

Stefanie Gürster

6.1 Namenskonvention

Zur Vereinheitlichung der Namensgebung der Komponenten in den Facelets verfahren wir nach dem *kebab-case* Prinzip. Um den genauen Typ in die ID mit aufzunehmen, wird das entsprechende Suffix angehängt.

Typ	Suffix
graphicImage	gimg
link	lnk
inputText	itxt
outputText	otxt
commandButton	cbtn
messages	msgs
dataTable	dtbl
selectOneMenu	slctom
panelGrid	pgrd
commandLink	clnk
inputSecret	iscrt
inputFile	ifile
column	clmn
selectBooleanCheckbox	slctbc

✓

6.2 Templates

Wechsel des Viewers Schicho Template auf Editor und
nicht abgesprochen -

6.3 Seiten

6.3.1 Anonymer Nutzer

6.3.2 Angemeldeter Nutzer

6.3.3 Gutachter

Johann Schicho ff.

6.3.4 Editor

6.3.5 Admin

7 DB Schema

Johann Schicho

hätte ich auch
Labels
wie vor weglassen
gibt es sow mit im DTO

Tabelle 1: **navigation.xhtml** ist die Kopfzeile der Webanwendung. Diese bietet die Suchfunktion, Links zu verschiedenen Listen und dem Profil an.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
logo-gimg	<h:graphicImage>	Logo der Applikation.	value="#{navigationBacking.systemSettings.logoImage}"	/	/
direct-to-home-link	<h:link>	Weiterleitung zur Homepage.	outcome="/facelets/authenticated/homepage" title="#{help.toHome}"	/	/
search-fm	<h:form>	Formular zur Suche.	/	/	/
search-Field-txt	<h:inputText>	Suchleiste	value="#{navigationBacking.resultListParameters.globalSearchWord}" title="#{help.search}"	/	/
search-cbbtn	<h:commandButton>	Suche ausführen.	action="#{navigationBacking.search}"	/	/
user-list-link	<h:link>	Link zur Übersichtseite aller Nutzer	value="#{label.userList}" outcome="/editor/userList" title="#{help.userList}"	rendered="#{navigationBacking.sessionInformation.user.editor}"	/
forum-list-link	<h:link>	Link zur Übersichtseite alle Journale und Konferenzen	value="#{label.forumList}" outcome="/facelets/authenticated/scientificForumList" title="#{help.forumList}"	/	/
logout-cbbtn	<h:commandButton>	Loggt den Nutzer aus dem System aus und leitet zur Loginseite weiter.	value="#{label.logout}" action="#{navigationBacking.logout}" title="#{help.logout}"	/	/
profile-link	<h:link>	Link zur Profilübersicht	value="#{label.nav.profile}" outcome="/facelets/authenticated/profile"	/	/

Convention für Sh:Navigation-Feldern

Tabelle 2: **main.xhtml** ist das Template, welches den Inhalt der Seiten zwischen Kopf- und Fußzeile einbettet.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
/	<ui:include>	Kopfzeile	src="navigation.xhtml"	/	/
global-msgs	<h:messages>	Anzeigen globaler Nachrichten.	/	/	/
/	<ui:insert>	Seiteninhalt	name="main-content"	/	/
/	<ui:include>	Fußzeile	src="footer.xhtml"	/	/

Tabelle 3: footer.xhtml ist die Fußzeile der Webanwendung und für alle sichtbar.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
imprint-link	<h:link>	Link zur Seite des Impressumss	value="#{label.imprint}" outcome="/facelets/anonymous/imprint" title="#{help.imprint}"	/	/
language-otxt	<h:outputText>	Anzeige der eingestellten Sprache.	value="#{footerBacking.sessionInformation.locale.getLanguage()}"	/	/

Tabelle 4: **toolbar.xhtml** ist die Seitenleiste für Editoren und Administratoren auf der Einreichungsübersicht. Hier wird die Einreichung verwaltet.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
add-reviewer-iTxt	<h:input-Text>	Eingabefeld zur Angabe einer E-Mail-Adresse	value="#{toolbarBacking.reviewerInput.email}" title="#{help.reviewerEmail}"	/	emailAddressExistsValidator emailAddressLayoutValidator required="true" requiredMessage="#{message.addReviewer}"
add-deadline-reviewer-iTxt	<h:input-Text>	Eingabefeld zur Angabe einer Deadline.	value="#{toolbarBacking.reviewerByInput.timestampDeadline}" title="#{help.reviewerDeadline}"	/	converterDateTime pattern="dd/MM/yyyy"
add-reviewer-cbbtn	<h:commandButton>	Knopf zum Hinzufügen des Gutachters	value="#{label.addReviewer}" action="#{toolbarBacking.addReviewer}" title="#{help.addReviewer}"	/	/
reviewer-dtbl	<h:dataTable>	Liste der Gutachter	value="#{toolbarBacking.reviewer}" var="review"	/	/
remove-reviewer-cbbtn	<h:commandButton>	Entferne einen bestimmten Gutachters.	action="#{toolbarBacking.removeReviewer}" title="#{help.removeReviewer}"	/	/
select-editor-selctom	<h:selectOneMenu>	Auswahländerung des verwaltenden Editors	value="#{toolbarBacking.editorInput.firstName + toolbarBacking.editorInput.lastName}" title="#{help.selectEditor}"	/	/
/	<f:selectItems>	Liste der möglichen Editoren.	value="#{toolbar.editors}" var="entry" itemLabel="#{entry.firstName + entry.lastName}" itemValue="#{entry}"		
save-editor-cbbtn	<h:commandButton>	Speichern der Änderung des Editors.	value="#{label.saveEditor}" action="#{toolbarBacking.chooseNewManagingEditor}" title="#{help.saveEditor}"	/	/
show-editor-otxt	<h:outputText>	Anzeige des aktuellen Editors.	value="#{toolbarBacking.currentEditor}"	/	/
revision-deadline-iTxt	<h:input-Text>	Eingabe zur Angabe einer Deadline.	value="#{toolbarBacking.submission.deadlineRevision}" title="#{help.revisionDeadline}"	/	converterDateTime pattern="dd/MM/yyyy"
require-revision-cbbtn	<h:commandButton>	Fordert den Einreicher auf, seine Einreichung zu überarbeiten	value="#{label.requireRevision}" action="#{toolbarBacking.requireRevision}" title="#{help.requireRevision}"	/	/
accept-submission-cbbtn	<h:commandButton>	Akzeptiere die Einreichung	value="#{label.acceptSubmission}" action="#{toolbarBacking.acceptSubmission}" title="#{help.acceptSubmission}"	/	/
reject-submission-cbbtn	<h:commandButton>	Lehne die Einreichung ab	value="#{label.rejectSubmission}" action="#{toolbarBacking.rejectSubmission}" title="#{help.rejectSubmission}"	/	/

Via
disum
Pattern
nicht
input -
nicht
binden

Tabelle 5: pagination.xhtml ist ein Composite Component zur Paginierung von Listen.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
/	<composite:attribute> ✓	Paginator Attribut.	name="paginator" type="de.lases.control.internal.Pagination" ✓	required="true"	/
is-empty-otxt	<h:outputText>	Hinweis wenn die Tabelle leer ist.	value="#{label.emptyList}"	rendered="#{cc.attrs.paginator.entries.isEmpty()}"	/
pagination-dtbl	<h:dataTable>	Hier werden die Spalten eingefügt. Durch Klicken auf die Spaltennamen werden die Einträge sortiert. Zusätzlich kann durch ein Textfeld die Liste gefiltert werden.	value="#{cc.attrs.paginator.entries}" var="var" <i>↳ item ?</i>	rendered="#{cc.attrs.paginator.entries.isEmpty()}"	/
/	<composite:insertChildren>	Einfügen benötigter Spalten.	/	/	/
apply-cbtn	<h:commandButton>	Mit diesem Knopf können gewählte Filter angewandt werden.	value="#{label.applyFilter}" action="#{cc.attrs.paginator.loadData}" title="#{help.applyFilter}"	/	/
navigation-pgrd	<h:panelGrid>	Hier befinden sich die Knöpfe zur Navigation zwischen den Seiten der Paginierungen.	columns="5"	rendered="#{cc.attrs.paginator.entries.isEmpty()}"	/
first-page-cbtn	<h:commandButton>	Zurück zur ersten Seite.	value="#{label.firstPage}" action="#{cc.attrs.paginator.firstPage}" title="#{help.firstPage}"	/	/
prev-page-cbtn	<h:commandButton>	Eine Seite zurück.	value="#{label.prevPage}" action="#{cc.attrs.paginator.previousPage}" title="#{help.prevPage}"	/	/
current-page-otxt	<h:outputText>	Nummer der aktuellen Seite.	value="#{cc.attrs.paginator.currentPage}" title="#{help.currentPage}"	/	/
next-page-cbtn	<h:commandButton>	Eine Seite weiter.	value="#{label.nextPage}" action="#{cc.attrs.paginator.nextPage}" title="#{help.nextPage}"	/	/
last-page-cbtn	<h:commandButton>	Zur letzten Seite.	value="#{label.lastPage}" action="#{cc.attrs.paginator.lastPage}" title="#{help.lastPage}"	/	/

Tabelle 6: `sortfiltercolumnn.xhtml` ist ein Composite Component, das eine sortierbare und filterbare Spalte einer Liste modelliert.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
/	<composite:attribute>	Implementation der abstrakten Klasse.	name="paginator" type="de.lases.control.internal.Pagination"	required="true"	/
/	<composite:attribute>	Anzeigenname im Spaltenkopf.	name="columnIdent" type="java.lang.String"	required="true"	/
/	<composite:attribute>	String Identifier der Spalte.	name="columnName" type="java.lang.String"	required="true"	/
/	<composite:attribute>	Spaltennummer, null indiziert.	name="columnNo" type="java.lang.String"	required="true"	/
column-name-clnk	<h:commandLink>	Durch Klicken auf den Spaltennamen wird nach dieser Spalte aufsteigend, bzw. absteigend sortiert.	value="#{cc.attrs.columnNamei18n}" action="#{cc.attrs.paginator.sortBy[cc.attrs.columnNo]}"		/
column-filter-ixt	<h:inputText>	Hier kann ein Wert eingegeben werden, nach dem die Spalte dann gefiltert wird.	value="#{cc.attrs.paginator.searchWords[cc.attrs.columnNo]}"	/	/

~~maximal~~
 wichtig für validierung i. DDL
 maximal

Tabelle 7: **welcome.xhtml** Auf der Login- bzw. Welcome-Seite wird das System 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	Binding	Constraints	Validator Converter
welcome-heading-otxt	<h:outputText>	Überschrift der Welcomepage.	value="#{welcomeBacking.systemSettings.headlineWelcomePage}"	/	/
welcome-otxt	<h:outputText>	Bewerten der Anwendung mithilfe einer Kurzbeschreibung.	value="#{welcomeBacking.systemSettings.messageWelcomePage}"	A/	/
email-ixt	<h:inputText>	Textfeld für E-Mail Eingabe.	value="#{welcomeBacking.loginInput.emailAddress}" title="#{help.loginEmail}"	/	emailAddressExistsValidator, emailAddressLayoutValidator, required="true" requiredMessage="#{message.email}"
password-isrt	<h:inputSecret>	Textfeld für Passwort Eingabe.	value="#{welcomeBacking.loginInput.passwordNotHashed}" title="#{help.loginPassword}"	/	passwordValidator required="true" requiredMessage="#{message.password}"
login-cbtn	<h:commandButton>	Ausführen des Login-Prozesses.	value="#{label.login}" action="#{welcomeBacking.login}" title="#{help.login}"	/	/
register-cbtn	<h:commandButton>	Weiterleitung zur Registrierung.	value="#{label.register}" action="#{welcomeBacking.goToRegister}" title="#{help.register}"	/	/

Tabelle 8: registration.xhtml Seite zur Registrierung anonymer Nutzer.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
title-ixt	<h:input-Text>	Angabe eines Titels.	value="#{registrationBacking.newUser.title}" title="#{help.title}"	/	/
first-name-ixt	<h:input-Text>	Angabe des Vornamens.	value="#{registrationBacking.newUser.firstName}" title="#{help.firstName}"	/	required="true" requiredMessage="#{message.firstName}"
last-name-ixt	<h:input-Text>	Angabe des Nachnamen.	value="#{registrationBacking.newUser.lastName}" title="#{help.lastName}"	/	required="true" requiredMessage="#{message.lastName}"
password-isrt	<h:input-Secret>	Angabe eines Passwortes.	value="#{registrationBacking.newUser.passwordNotHashed}" title="#{help.password}"	/	passwordValidator required="true" requiredMessage="#{message.password}"
email-ixt	<h:input-Text>	Angabe einer validen E-Mail.	value="#{registrationBacking.newUser.emailAddress}" title="#{help.email}"	/	emailAddressExistsValidator emailAddressLayoutValidator required="true" requiredMessage="#{message.email}"
register-cbfn	<h:commandButton>	Ausführen des Registrierungsprozesses.	value="#{label.register}" action="#{registrationBacking.register}"	/	/
welcomepage-link	<h:link>	Weiterleitung zur Login Seite.	value="#{label.welcomeLink}" outcome="/facelets/anonymous/welcome" title="#{help.welcomepage}"	/	/

Stil:
bei Methodennamen

Tabelle 9: **verification.xhtml** Wird bei erfolgreicher Verifikation der E-Mail angezeigt.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
go-to-home-cbtn	<h:commandButton>	Button zur Weiterleitung auf die Homepage bei erfolgreicher Verifizierung. Ansonsten wird auf <i>welcome.xhtml</i> weitergeleitet.	/	/	



Endbutton der Bindung macht hier Sinn

Tabelle 10: **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	Binding	Constraints	Validator Converter
error-message-otxt	<h:outputText>	Anzeige einer Fehlermeldung.	value="#{errorPageBacking.errorMessage.message}"	/	/
stack-trace-otxt	<h:outputText>	Stacktrace für den "Development Mode".	value="#{errorPageBacking.errorMessage.stackTrace}"	/	rendered="#{errorPageBacking.isDevelopmentMode()}"

V

2. ist das schon im Header

Tabelle 11: imprint.xhtml Das Impressum gibt die vom Administrator angegebenen Kontaktdaten des Betreibers wieder.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
logo-gimg	<h:graphicImage>	Logo des Betreibers.	value="#{imprintBacking.systemSettings.logoImage}"	/	/
imprint-heading-otxt	<h:outputText>	Überschrift der Ansicht	value="#{imprintBacking.systemSetting.companyName}"	/	/
imprint-otxt	<h:outputText>	Impressum des Betreibers.	value="#{imprintBacking.systemSetting.imprint}"	/	/

Wel Name formatieren als Admin?

Tabelle 12: `homepage.xhtml` Startseite die den Überblick über alle Einreichungen, aufgeteilt in Reiter, beinhaltet.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
your-submission-clnk	<h:commandLink>	Reiter zum Anzeigen der eigenen Einreichungen.	value="#{label.yourSubmission}" action="#{homepageBacking.showOwnSubmissionsTab}" title="#{help.yourSubmissions}"	/	/
your-reviews-clnk	<h:commandLink>	Reiter zum Anzeigen der zu begutachteten Einreichungen	value="#{label.yourReviews}" action="#{homepageBacking.showSubmissionToReviewTab}" title="#{help.yourReviews}"	rendered="#{homepageBacking.sessionInformation.userReviewer}"	/
editorial-overview-clnk	<h:commandLink>	Reiter zum Anzeigen aller Einreichungen, welche man editiert.	value="#{label.editorialOverview}" action="#{homepageBacking.showSubmissionToEditTab}" title="#{help.editorialOverview}"	rendered="#{homepageBacking.sessionInformation.editor}"	/

Tabelle 13: Die Tabellen *submissionPagination*, *reviewPagination* und *editorPagination* sind nach den gleichen Punkten gegliedert und daher nur einmal ausführlich aufgelistet: Titel des Papers, Datum, Deadline und Status der Einreichung und Name des zugehörigen Forums.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
submission-pg	<pg:pagination>	Liste aller eignen Einreichungen.	paginator="#{homepageBacking.submissionPagination}"	/	/
title-ssc	<ssc:sortsearchcolumn>	Spalte für den Titel des Papers.	paginator="#{homepageBacking.submissionPagination}" columnNamei18n="#{label.paperTitle}" value="#{var.title}" outcome="/facelets/authenticated/submission" includeViewParams="true"	/	/
title-lnk	<h:link>	Titel des Einreichung.	value="#{var.title}" outcome="/facelets/authenticated/submission" includeViewParams="true"	/	/
/	<f:parameter>	Parameter für id der Einreichung.	value="#{var.id}" name="id"	/	/
date-ssc	<ssc:sortsearchcolumn>	Spalte für das Abgabedatum des Papers.	paginator="#{homepageBacking.submissionPagination}" columnNamei18n="#{label.date}"	/	/
date-scltom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit eines Datums.	value="#{homepageBacking.dateFilterSelectSub}" title="#{help.dateSelect}"	/	/
/	<f:selectItems>	Mögliche Filterung des Datums.	value="#{homepageBacking.dateSelects}" var="entry" itemLabel="#{entry.toString()}" itemValue="#{entry}"	/	/
date-otxt	<h:outputText>	Abgabedatum des Papers.	value="#{var.submissionTime}"	/	convertDateTime pattern="yyyy-MM-dd"
deadline-ssc	<ssc:sortsearchcolumn>	Spalte für die Deadline der Revision.	paginator="#{homepageBacking.submissionPagination}" columnNamei18n="#{label.deadline}"	/	/
deadline-otxt	<h:outputText>	Deadline des Papers.	value="#{var.deadlineRevision}"	/	convertDateTime pattern="yyyy-MM-dd"

Unvollständig
zu werden

Spalten-
Tabelle 14: Fortsetzung
Zurück zu Diskussion 19

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
state-ssc	<ssc:sort-searchcolumn>	Spalte für den Status der Einreichung.	paginator="#{homepageBacking.submissionPagination}" columnName="18n="#{label.stateSub}"	/	/
state-scltom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit eines Status.	value="#{homepageBacking.stateFilterSelectSub}" title="#{help.stateSubSelect}"	/	/
/	<f:selectItems>	Mögliche Filterung des Status.	value="#{homepageBacking.submissionStates}" var="entry" itemLabel="#{entry.toString()}" itemValue="#{entry}"	/	/
state-obtxt	<h:outputText>	Status der Einreichung.	value="#{var.state}"	/	/
forum-ssc	<ssc:sort-searchcolumn>	Spalte für zugehöriges Forum.	paginator="#{homepageBacking.submissionPagination}" columnName="18n="#{label.forumSub}"	/	/
forum-lnk	<h:link>	Zugehöriges Forum der Einreichung.	value="#{homepageBacking.scientificForumFormSubmission}outcome=/facelets/authenticated/-scientificForum" includeViewParams="true"	/	/
/	<f:parameter>	Parameter für id des Forums.	value="#{var.scientificForumId}"name="id"	/	/

Tabelle 15: **submission.xhtml** ist die Übersichtsseite einer Abgabe. Hier werden alle mit der Einreichung verbundenen Aktivitäten abgebildet.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
/	<ui:include>	Seitenleiste mit Tools.	rendered="#{submissionBacking.sessionInformation.user.editor}"	/	
/	<f:viewParam>	Id der Einreichung.	value="#{submissionBacking.submission.id}" name="id"	/	/
/	<f:viewAction>	Laden der entsprechenden Einreichung.	action="#{submissionBacking.onLoad}"		
upload-revision-ifile	<h:inputFile>	Upload einer Revision.	value="#{submissionBacking.uploadedRevisionPDF}"	rendered="#{submissionBacking.isViewerSubmitter}"	required="true" requiredMessage="#{message.inputFile}"
upload-clnk	<h:commandLink>	Ausführen des Uploads.	value="#{label.uploadFile}" action="#{submissionBacking.uploadPDF}" includeViewParams="true" title="#{help.uploadRevision}"	rendered="#{submissionBacking.isViewerSubmitter}"	/
new-review-cbbtn	<h:commandButton>	Weiterleitung zur Einreichung eines neuen Gutachtens.	value="#{label.uploadReview}" action="#{submissionBacking.uploadReview}" title="#{help.uploadRevision}"	rendered="#{submissionBacking.sessionInformation.user.viewer}"	/
deadline-review-otxt	<h:outputText>	Deadline der Einreichung.	value="#{submissionBacking.reviewedBy.timestampDeadline}"	rendered="#{submissionBacking.sessionInformation.user.viewer}"	/
accept-reviewing-cbbtn	<h:commandButton>	Annahme der Rolle eines Gutachters für diese Einreichung.	value="#{label.acceptReviewing}" action="#{submissionBacking.acceptReviewing}" title="#{help.acceptReviewing}"	rendered="#{submissionBacking.sessionInformation.user.viewer}"	/
decline-reviewing-cbbtn	<h:command>	Ablehnen der Rolle eines Gutachters für diese Einreichung.	value="#{label.declineReviewing}" action="#{submissionBacking.declineReviewing}" title="#{help.declineReviewing}"	rendered="#{submissionBacking.sessionInformation.user.viewer}"	/
state-selctom	<h:selectOneMenu>	Status der Submission.	value="#{submissionBacking.submission.state}"	readonly="#{not submissionBacking.sessionInformation.user.editor}"	/
/	<f:selectItems>	Auswahl des Status.	value="#{submissionBacking.submissionState}" var="entry" itemLabel="#{entry.toString()}" itemValue="#{entry}"	/	/
state-otxt	<h:outputText>	Status der Einreichung.	value="#{submissionBacking.submission.state}"	/	/

Binding der "Beitrag"

Tabelle 16: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
forum-lnk	<h:link>	Name des Forums und Weiterleitung zur Forumsübersicht.	value="#{submissionBacking.scientificForum.name}" outcome="/facelets/authenticated/scientificForum" includeViewParams="true" title="#{help.directToForum}"	disabled="#{not submissionBacking.sessionInformation.user.editor}"	/
/	<f:param>	Parameter für id des Forums.	value="#{submissionBacking.scientificForum.scientificForumId}"name=id"	/	/
author-lnk	<h:link>	Name des Autors und Weiterleitung zum Profil.	value="#{submissionBacking.author.firstName + submissionBacking.author.lastName}" outcome="/facelets/authenticated/profile" includeViewParams="true" title="#{help.directProfile}"	disabled="#{not submissionBacking.sessionInformation.user.editor}"	/
/	<f:param>	Parameter für id des Profil.	value="#{submissionBacking.author.id}"name=id"	/	/
email-lnk	<h:link>	E-Mail-Adressen der Co-Autoren und Mailto-Link.	value="#{submissionBacking.author.emailAddress}" outcome="#{'mailto:' + concat(submissionBacking.author.emailAddress)}"	disabled="#{not submissionBacking.sessionInformation.user.editor}"	/
deadline-revision-otxt	<h:outputText>	Deadline der Revision	value="#{submissionBacking.submission.deadlineRevision}"	/	/

Tabelle 17: *paperPagination* enthält die Spalten: Version, Datum, Sichtbarkeit(E) und Download.Die Elemente der Liste, der unterschiedlichen Versionen der Einreichung, sieht wie folgt aus:

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
paper-pg	<pg:pagination>	Liste aller eignen Einreichungen.	paginator="#{submissionBacking.paperPagination}"	/	/
version-ssc	<ssc:sortsearchcolumn>	Spalte für die Version des Papers.	paginator="#{submissionBacking.paperPagination}" columnNamei18n="#{label.paperVersion}"	/	/
version-otxt	<h:outputText>	Version des Papers.	value="#{var.submissionId}"	/	/
date-ssc	<ssc:sortsearchcolumn>	Spalte für das Abgabedatum des Papers.	paginator="#{submissionBacking.paperPagination}" columnNamei18n="#{label.date}"	/	/
date-scltom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit eines Datums.	value="#{submissionBacking.dateFilterSelectPaper}" title="#{help.dateSelect}"	/	/
/	<f:selectItems>	Mögliche Filterung des Datums.	value="#{submissionBacking.dateSelect}" var="entry" itemLabel="#{entry.toString()}" itemValue="#{entry}"	/	/
date-otxt	<h:outputText>	Abgabedatum des Papers.	value="#{var.uploadTime}"	/	convertDateTime pattern="yyyy-MM-dd"

Bindung an "Befehl"

Tabelle 18: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
release-revision-ssc	<ssc:sort-searchcolumn>	Spalte für den Freigabestatus.	paginator="#{submissionBacking.paperPagination}" columnName="#{label.releaseRevision}"	rendered="#{submissionBacking.sessionInformation.user.editor submissionBacking.sessionInformation.user.viewer}"	/
release-revision-selectom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit Freischalten der Revision	value="#{submissionBacking.visibleFilterInputPaper}" "title="#{help.releaseRevisionSelect}"	rendered="#{submissionBacking.sessionInformation.user.editor submissionBacking.sessionInformation.user.viewer}"	/
/	<f:selectItem>	Revision freigeschaltet.	itemValue="true" itemLabel="#{label.releaseRevisionTrue}"	/	/
/	<f:selectItem>	Revision nicht freigeschaltet.	itemValue="false" itemLabel="#{label.releaseRevisionFalse}"	/	/
release-review-otxt	<h:outputText>	Status der Freischaltung.	value="#{label.releaseRevision}"	rendered="#{var.visible && submissionBacking.sessionInformation.user.viewer}"	/
release-revision-cbbtn	<h:commandButton>	Freischalten einer Revision.	value="#{label.releaseRevision}action="#{submissionBacking.releaseRevision(var)}"title="#{help.releaseRevision}"	rendered="#{submissionBacking.sessionInformation.user.editor}"	/
download-file-clmn	<h:column>	Spalten für den Download der Datei.	/	/	/

Tabelle 19: *reviewPagination* enthält die Spalten: Gutachter, Datum, Status, Empfehlung, Kommentar und Download. Ist ein Gutachten freigeschaltet, so erhält auch ein Nutzer Leserechte. Die Elemente der Liste von Gutachten sind folgende:

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
review-pg	<pg:pagination>	Liste aller oder der eignen Gutachten.	paginator="#{submissionBacking.reviewPagination}"	/	/
reviewer-ssc	<ssc:sortsearchcolumn>	Spalte für den Namen des Gutachters.	paginator="#{submissionBacking.reviewPagination}" columnName="#{label.reviewName}"	rendered="#{submissionBacking.sessionInformation.user.editor && submissionBacking.sessionInformation.user.viewer}"	/
reviewer-link	<h:link>	Namen des Gutachters.	value="#{submissionBacking.getReviewerForReview(var)}" outcome="/facelets/authenticated/profile" includeViewParams="true"	rendered="#{submissionBacking.sessionInformation.user.editor && submissionBacking.sessionInformation.user.viewer}"	/
/	<f:parameter>	Parameter für id des Profil.	value="#{submissionBacking.getReviewerForReview(var).id}" name="id"	/	/
date-ssc	<ssc:sortsearchcolumn>	Spalte für das Abgabedatum des Gutachters.	paginator="#{submissionBacking.reviewPagination}" columnName="#{label.date}"	/	/
date-scltom	<h:selectedOneMenu>	Auswahl der Filtermöglichkeit eines Datums.	value="#{submissionBacking.dateFilterSelectReview}" title="#{help.dateSelect}"	/	/
/	<f:selectedItems>	Mögliche Filterung des Datums.	value="#{dateSelect}" var="entryItem" value="#{entry.itemLabel}" "#{entry.toString()}"	/	/
date-otxt	<h:outputText>	Abgabedatum des Gutachters.	value="#{var.timestampUploaded}"	/	convertDateTime pattern="yyyy-MM-dd"

Tabelle 20: Fortsetzung 1

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
release-review-ssc	<ssc:sort-searchcolumn>	Spalte für den Freigabestatus.	paginator="#{submissionBacking.review.Pagination}" columnName="#{label.releaseReview}"	rendered="#{submissionBacking.sessionInformation.user.editor}"	/
release-review-sdltom	<h:selectOneMenu>	Auswahl der Freischaltmöglichkeit der Gutachten	value="#{submissionBacking.visibleFilterInputPaper}" title="#{help.releaseReviewSelect}"	rendered="#{submissionBacking.sessionInformation.user.editor submissionBacking.sessionInformation.user.viewer}"	/
/	<f:selectItem>	Gutachten freigeschaltet.	itemValue="true" itemLabel="#{label.releaseReviewTrue}"	/	/
/	<f:selectItem>	Gutachten nicht freigeschaltet.	itemValue="false" itemLabel="#{label.releaseReviewFalse}"	/	/
release-review-otxt	<h:outputText>	Status der Freischaltung.	value="#{label.releaseReview}"	rendered="#{var.visible}"	/
release-review-cbtn	<h:commandButton>	Freischalten eines Gutachtens.	value="#{label.releaseReview}" action="#{submissionBacking.releaseReview(var)}" title="#{help.releaseReview}"	rendered="#{submissionBacking.sessionInformation.user.editor}"	/

Tabelle 21: Fortsetzung 2

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
recommen- dation- review-ssc	<ssc:sort- searchco- lumn>	Spalte für die Emp- fehlung.	paginator="#{submissionBacking.reviewPagina- tion}" columnNamei18n="#{label.recommendation}"	rendered="#{submis- sionBacking.sessionIn- formation.user.editor submissionBacking.session- Information.user.reviewer	/
recommen- dation- scltom	<h:selec- tOneMe- nu>	Auswahl der Filter- möglichkeit Empfeh- lung.	value="#{submissionBacking.recommendation- FilterInputReview}" title="#{help.recommendationSelect}"	rendered="#{submis- sionBacking.sessionIn- formation.user.editor submissionBacking.session- Information.user.reviewer	/
/	<f:selectI- tem>	Positive Empfehlung.	itemValue="true" itemLabel="#{label.recommendationTrue}"	/	/
/	<f:selectI- tem>	Negative Empfeh- lung.	itemValue="false" itemLabel="#{label.recommendationFalse}"	/	/
recommen- dation- slctbc	<h:sel- ectBoole- anCheck- box>	Empfehlung eines Gutachters.	value="#{var.acceptPaper}"readOnly="true"	rendered="#{submis- sionBacking.sessionIn- formation.user.editor submissionBacking.session- Information.user.reviewer	/
comment- ssc	<ssc:sort- searchco- lumn>	Spalte für den Kom- mentar des Gutach- ters.	paginator="#{submissionBacking.reviewPagina- tion}" columnNamei18n="#{label.comment}"	/	/
comment- otxt	<h:out- putText>	Kommentar des Gut- achters.	value="#{var.comment}"	/	/
download- file-clmn	<h:co- lumn>	Spalten für den Download der Datei.	/	/	/

Tabelle 22: Die Spalte zum Download der einzelnen Paper ist wie folgt definiert und ist beiden Paginations vorhanden:

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
download-header	<f:facet>	Spaltenkopf für den Download.	name="header"	/	/
pdf-submission-cbbtn	<h:commandButton>	Download der Einreichung	value="#{label.downloadFile}action=#{submissionBacking.downloadPaper(var)}"title="#{help.downloadFile}"	/	/

Tabelle 23: newSubmission.xhtml Hier können neue Paper eingereicht werden.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
submission-title-ixt	<h:inputText>	Titel des abzugebenen Papers.	value="#{newSubmissionBacking.newSubmission.title}"	/	required="true" requiredMessage="#{message.titleSub}"
forum-name-ixt	<h:inputText>	Name des Forums, bei welchem abgegeben wird.	value="#{newSubmissionBacking.newSubmission.forumInput.name}"	/	required="true" requiredMessage="#{message.forum}"
editor-slctom	<h:selectOneMenu>	Angabe eines Editors	value="#{newSubmissionBacking.selectedEditor}"	/	/
editor-1st	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="#{newSubmissionBacking.editors}"	/	/
pdf-upload-ifile	<h:inputText>	Angabe der Abgabedatei.	value="#{newSubmissionBacking.uploadedPDF}"	/	required="true" requiredMessage="#{message.inputFile}"
titel-ixt	<h:inputText>	Angabe eines Titels eines Co-Autors.	value="#{newSubmissionBacking.coAuthorInput.title}"	/	/
co-author-firstname-ixt	<h:inputText>	Angabe des Vornamen eines Co-Autors.	value="#{newSubmissionBacking.coAuthorInput.firstName}"	/	/
co-author-lastname-ixt	<h:inputText>	Angabe des Nachnamen eines Co-Autors.	value="#{newSubmissionBacking.coAuthorInput.lastName}"	/	/
co-author-email-ixt	<h:inputText>	Angabe der E-Mail eines Co-Autors.	value="#{newSubmissionBacking.coAuthorInput.emailAddress}"	/	/
submit-co-author-cbtn	<h:commandButton>	Speichern des Co-Autors.	value="#{label.save}" action="#{newSubmissionBacking.submitCoAuthor}"	/	/
co-author-list	<h:dataTable>	Anzeige aller Co-Autoren in einer Liste.	value="#{newSubmissionBacking.coAuthors}"	/	/
delete-co-author-cbtn	<h:commandButton>	Löschen des zugehörigen Co-Autors.	value="#{label.delete}" action="#{newSubmissionBacking.deleteCoAuthor(var)}"	/	/
submit-cbtn	<h:commandButton>	Ausführen des Einreichungsprozesses.	value="#{label.submit}" action="#{newSubmissionBacking.submit}"	/	/

Wird nach den eigenen Einreichungen, nach denen, die man begutachtet oder nach Einreichungen in eigener editorialer Verantwortung gesucht, so erscheinen die Listen im gleichen 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.

Tabelle 24: **searchResult.xhtml**

Tabelle 25: scientificForumList.xhtml Hier erhalten die Nutzer eine Übersicht über alle wissenschaftliche Foren.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
date-select-selctom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit eines Datums.	value="#{scientificForumListBacking.deadlineFilterSelect}"	/	/
deadline-list	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="#{scientificForumListBacking.dateSelects}"	/	/
forums-pagination-pg	<pg:pagination>	Liste aller Foren im System.	paginator="#{scientificForumListBacking.scientificForumPagination}"	/	/
name-ssc	<ssc:sortsearchcolumn>	Spalte für den Namen des Forums.	paginator="#{scientificForumListBacking.scientificForumPagination}" columnNamei18n="#{label.name}"	/	/
name-link	<h:link>	Name des Forums.	value="#{var.name}" includeViewParams="true"	/	/
/	<f:parameter>	URL Parameter Forum ID	name="#{'id'}" value="#{var.id}"	/	/
deadline-ssc	<ssc:sortsearchcolumn>	Spalte für die Deadline des Forums.	paginator="#{scientificForumListBacking.scientificForumPagination}" columnNamei18n="#{label.deadline}"	/	/
deadline	output-Text	Abgabefrist des jeweiligen Forums.	value="#{var.deadline}"	/	/

Tabelle 26: **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	Binding	Constraints	Validator Converter
avatar-gimg	<h:graphicImage>	Aktueller Avatar.	value="'#{https://lases.de/image/' concat(profileBacking.user.id)}'"	/	/
new-avatar-ifile	<h:inputFile>	Hochladen eines neuen Avatars.	value="'#{profileBacking.uploadedAvatar}'"	rendered="'#{profileBacking.hasViewerEditRights}'"	/
submit-avatar-cbtn	<h:commandButton>	Lade neuen Avatar hoch.	value="'#{profileBacking.uploadAvatar}'"	rendered="'#{profileBacking.hasViewerEditRights}'"	/
delete-avatar-cbtn	<h:commandButton>	Löschen des Avatars.	value="'#{profileBacking.deleteAvatar}'"	rendered="'#{profileBacking.hasViewerEditRights}'"	/
role-otxt	<h:outputText>	Angabe der Rolle eines Nutzers	?	/	/
is-admin-cbx	<h:selectBooleanCheckbox>	Zuteilung der Rolle des Administrators.	value="'#{profileBacking.user.admin}'"	rendered="'#{profileBacking.sessionInformation.admin}'"	/
title-ittxt	<h:inputText>	Titel des Nutzenden.	value="'#{profileBacking.user.title}'"	readonly="'#{profileBacking.hasViewerEditRights}'"	/
firstname-ittxt	<h:inputText>	Vorname des Nutzenden.	value="'#{profileBacking.user.firstName}'"	readonly="'#{profileBacking.hasViewerEditRights}'"	required="true" requiredMessage="#{message.firstName}"
lastname-ittxt	<h:inputText>	Nachname des Nutzenden.	value="'#{profileBacking.user.lastName}'"	readonly="'#{profileBacking.hasViewerEditRights}'"	required="true" requiredMessage="#{message.lastName}"

Tabelle 27: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
password-isrt	<h:inputSecret>	Leeres Eingabefeld für ein neues Passwort.	value="{profileBacking.user.passwordNotHashed}"	rendered="{!profileBacking.hasViewerEditRights}"	passwordValidator required="true" requiredMessage="{message.password}"
email-itxt	<h:inputText>	E-Mail des Nutzens.	value="{profileBacking.user.emailAddress}"	readonly="{!profileBacking.hasViewerEditRights}"	emailAddressExistsValidator emailAddressExistsValidator required="true" requiredMessage="{message.email}"
/	<f:attribute>	Parameter zur E-Mail Validierung.	name="id" value="{profileBacking.user.id}"	/	/
submission-number-otxt	<h:outputText>	Anzahl der Einreichungen des Nutzers	?	/	/
employer-itxt	<h:inputText>	Angabe des Arbeitgebers	value="{profileBacking.user.employer}"	readonly="{!profileBacking.hasViewerEditRights}"	/

Tabelle 28: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
science-field-1st	<h:dataTable>	Spezialgebiete des Nutzers	value="#{profileBacking.usersScienceFields}"	/	/
add-science-field-sltolb	<p:selectOneListBox>	Auswahl der Spezialgebiete des Nutzers	value="#{profileBacking.selectedScienceField}"	rendered="#{!profileBacking.hasViewerEditRights}"	/
field-1st	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="#{profileBacking.scienceFields}"	/	/
delete-field-cbtn	<h:commandButton>	Löschen des ausgewählten Fachgebietes.	value="#{label.delete}" action="#{profileBacking.deleteScienceField}"	rendered="#{!profileBacking.hasViewerEditRights}"	/
add-field-cbtn	<h:commandButton>	Füge Fachgebiet zur Liste hinzu.	value="#{label.add}" action="#{profileBacking.addScienceField}"	rendered="#{!profileBacking.hasViewerEditRights}"	/
date-of-birth-1txt	<h:inputText>	Angabe des Geburtstages	value="#{profileBacking.user.dateOfBirth}"	readonly="#{!profileBacking.hasViewerEditRights}"	/
save-cbtn	<h:commandButton>	Daten werden gespeichert.	value="#{label.save}" action="#{profileBacking.submitChanges}"	rendered="#{!profileBacking.hasViewerEditRights}"	/
delete-cbtn	<h:commandButton>	Nutzer und alle damit verbundenen Daten werden gelöscht.	value="#{label.deleteUser}" action="#{profileBacking.deleteProfile}"	rendered="#{!profileBacking.hasViewerEditRights}"	/

Tabelle 29: 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	Binding	Constraints	Validator Converter
password-admin-isrt	<h:inputSecret>	Angabe eines Passworts.	value="{#{profileBacking.adminPasswordInPopup.passwordNotHashed}}"	/	passwordValidator required="true" requiredMessage="{#{message.password}}"
admin-note-otxt	<h:outputText>	Hinweis zur Veränderung der Adminrolle.	value="{#{label.adminNote}}"	/	/
abort-cbtn	<h:commandButton>	Abbruch der Änderung.	value="{#{label.save}}" action="{#{profileBacking.saveInPopup}}"	/	/
save-cbtn	<h:commandButton>	Speichern aller Änderungen.	value="{#{label.abort}}" action="{#{profileBacking.abortInPopup}}"	/	/

Tabelle 30: **scientificForum.xhtml** Die Ansicht eines Forums dient zur Ausgabe von Informationen über die jeweilige Konferenz oder das jeweilige Journal. Das Anzeigen und Filtern der Tabellen *editorialPagination*, *reviewPagination* und *submissionPagination* wird genauso gehandhabt, wie auf *homepage.xhtml*. Ausnahme dabei ist die Spalte Forum, welche hier nicht angezeigt wird.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
/	<f:view-Param>	URL Parameter holen	name="'#{id}'" value="'#{scientificForumBacking.scientificForum.id}'"		required="true"
/	<f:view-Action>	Action Methode ausführen.	action="'#{scientificForumBacking.onLoad}'"	/	/
forum-name-ixt	inputText	Name des Forums.	value="'#{scientificForumBacking.scientificForum.name}'"	readonly="'#{!scientificForumBacking.isViewerEditor}'"	required="true" requiredMessage="'#{message.forumName}'"
editor-1st	<h:dataTable>	Liste der verantwortlichen Editoren.	value="'#{scientificForumBacking.editors}'" title="'#{help.managingEditors}'"	/	/
new-editor-ixt	<h:inputText>	Angabe der E-Mail eines Editors.	value="'#{scientificForumBacking.newEditorInput.emailAddress}'"	rendered="'#{scientificForumBacking.sessionInformation.user.editor}'"	emailAddressExistsValidator emailAddressLayoutValidator required="true" requiredMessage="'#{message.email}'"
add-editor-cbbtn	<h:commandButton>	Füge Editor zur Liste hinzu.	value="'#{label.add}'" action="'#{scientificForumBacking.addEditor}'"	rendered="'#{scientificForumBacking.sessionInformation.user.editor}'"	/
remove-editor-cbbtn	<h:commandButton>	Lösche ausgewählten Editor.	value="'#{label.remove}'" action="'#{scientificForumBacking.removeEditor}'"	rendered="'#{scientificForumBacking.sessionInformation.user.editor}'"	/
deadline-ixt	<h:inputText>	Deadline des Forums.	value="'#{scientificForumBacking.scientificForum.deadline}'"	readonly="'#{!scientificForumBacking.isViewerEditor}'"	required="true" requiredMessage="'#{message.deadline}'"
description-ixt	inputText-Area	Kurzbeschreibung des Forums.	value="'#{scientificForumBacking.scientificForum.description}'"	readonly="'#{!scientificForumBacking.isViewerEditor}'"	required="true" requiredMessage="'#{message.description}'"
url-lnk	<h:outputLink>	Link zur Konferenz oder zum Journal.	value="'#{scientificForumBacking.scientificForum.url}'"	/	/
change-url-ixt	inputText	Angabe einer neuen URL.	value="'#{scientificForumBacking.scientificForum.url}'"	rendered="'#{!scientificForumBacking.isViewerEditor}'"	/
review-instructions-ixt	<h:inputText-Area>	Anleitung für eine Begutachtung.	value="'#{scientificForumBacking.scientificForum.reviewManual}'"	readonly="'#{!scientificForumBacking.isViewerEditor}'"	/

Tabelle 31: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
science-field-dtbl	<h:dataTable>	Fachgebiet des Forums.	value="{scientificForumBacking.selectedScienceFields}"	/	/
new-science-field	<p:selectOneListBox>	Hinzufügen eines neuen Fachgebietes.	value="{scientificForumBacking.selectedScienceFieldInput}"	rendered="{scientificForumBacking.isViewerEditor}"	/
field-lst	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="{scientificForumBacking.scienceFields}"	/	/
add-science-field-cbtn	<h:commandButton>	Speichern des neuen Fachgebietes.	value="{label.add}" action="{scientificForumBacking.addScienceField}"	rendered="{scientificForumBacking.isViewerEditor}"	/
remove-science-field-cbtn	commandButton	Löschen eines Fachgebietes.	value="{label.remove}" action="{scientificForumBacking.removeScienceField}"	rendered="{scientificForumBacking.isViewerEditor}"	/
save-cbtn	<h:commandButton>	Speichere Veränderungen.	value="{label.save}" action="{scientificForumBacking.submitChanges}"	rendered="{scientificForumBacking.isViewerEditor}"	/

Tabelle 32: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
submission-pg	<pgr:pagination>	Paginierte aller Einreichungen, die der Nutzer eingereicht hat.	paginator="#{scientificForumBacking.submissionPagination}"	/	/
title-ssc	<ssc:sortsearchcolumn>	Spalte für den Titel des Papers.	paginator="#{scientificForumBacking.submissionPagination}" columnName18n="#{label.submissionTitle}"	/	/
title-otxt	<h:outputText>	Titel der Einreichung.	value="#{var.title}"	/	/
date-ssc	<ssc:sortsearchcolumn>	Spalte für das Abgabedatum der Einreichung.	paginator="#{scientificForumBacking.submissionPagination}" columnName18n="#{label.date}"	/	/
deadline-scltom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit eines Datums.	value="#{scientificForumBacking.dateFilterSelectSub}" title="#{help.deadlineSelect}"	/	/
deadline-lst	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="#{scientificForumBacking.dateSelects}"	/	/
date-otxt	<h:outputText>	Abgabedatum der Einreichung.	value="#{var.submissionTime}"	/	convertDateTime pattern=yyyy-MM-dd
deadline-ssc	<ssc:sortsearchcolumn>	Spalte für die Deadline der Revision.	paginator="#{scientificForumBacking.submissionPagination}" columnName18n="#{label.deadline}"	/	/
deadline-otxt	<h:outputText>	Deadline für eine Revision.	value="#{var.deadlineRevision}"	/	convertDateTime pattern=yyyy-MM-dd
state-ssc	<ssc:sortsearchcolumn>	Spalte für den Status der Einreichung.	paginator="#{scientificForumBacking.submissionPagination}" columnName18n="#{label.stateSub}"	/	/
state-scltom	<h:selectOneMenu>	Auswahl der Filtermöglichkeit eines Status.	value="#{scientificForumBacking.stateFilterSelectSub}" title="#{help.stateSubSelect}"	/	/
state-lst	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="#{scientificForumBacking.submissionStates}"	/	/
state-otxt	<h:outputText>	Status der Einreichung.	value="#{var.state}"	/	/

Tabelle 33: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
Weitere Paginierungen mit den gleichen Spalten wie in der vorherigen Tabelle.					
editorial-pg	<pg:pagination>	Paginierte Liste aller Einreichungen, die der Editor verwaltet.	paginator="#{scientificForumBacking.editedPagination}"	rendered="#{scientificForumBacking.isViewerEditor}"	/
review-pg	<pg:pagination>	Paginierte Liste aller Einreichungen, die der Gutachter bearbeitet.	paginator="#{scientificForumBacking.reviewedPagination}"	rendered="#{userListBacking.sessionInformation.user.isReviewer}"	/

Tabelle 34: **newReview.xhtml** Möchte ein Gutachter eines Papers seine Beurteilung abgeben, so ist dies auf dieser Seite möglich.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
version-number-otxt	<h:outputText>	Versionsnummer des eingereichten Papers, zu welchem der Gutachter ein Gutachten abgeben kann.		/	/
recommendation-cbx	<h:selectBooleanCheckbox>	Empfehlung des Gutachters ein Paper zu akzeptieren.	value="#{newReviewBacking.review.acceptPaper}"	/	/
comment-itxt	<h:inputText>	Kommentar eines Gutachters	value="#{newReviewBacking.review.comment}"	/	/
review-pdf-file	<h:inputFile>	Einzureichendes Gutachten.	value="#{newReviewBacking.uploadedPDF}"	/	/
submit-review-btn	<h:commandButton>	Gutachten einreichen.	value="#{newReviewBacking.addReview}"	/	/

ist hier die Grenze zu Schöner?
Schwer erkennbar

Tabelle 35: userList.xhtml Editoren und Administratoren erhalten hier einen Überblick über alle Nutzer. Die Seiten sind paginiert.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
user-pg	<pg:pagi- nation>	Paginierte Liste aller Nutzer	paginator="#{userListBacking.userPagination}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
rol-ssc	<ssc:sort- searchco- lumn>	Spalte für die Nutzer- rolle	paginator="#{userListBacking.userPagination}" columnNamen18n="#{label.role}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
role-otxt	<h:out- putText>	Rolle des Nutzers	value="#{var.role}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
name-ssc	<ssc:sort- searchco- lumn>	Spalte für den Namen des Nutzers	paginator="#{userListBacking.userPagination}" columnNamen18n="#{label.name}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
name-link	<h:link>	Name des Nut- zers. Link mit URL-Parameter.	value="#{(var.firstName.concat().concat(var.last- Name))}" outcome="/facelets/authenticated/profile" includeViewParams="true"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
/	<f:pa- ram>	Nutzer ID als URL Parameter.	name="#{'id'}" value="#{var.id}"	/	/
email-ssc	<ssc:sort- searchco- lumn>	Spalte für die E-Mail des Nutzers	paginator="#{userListBacking.userPagination}" columnNamen18n="#{label.email}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
email-link	<h:link>	Mailto-Link E-Mail-Adresse des Nutzers.	value="#{(var.emailAddress)}" outcome="#{'mailto:'concat(var.emailAddress)}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
employer- ssc	<ssc:sort- searchco- lumn>	Spalte für den Ar- beitsgeber des Nut- zers	paginator="#{userListBacking.userPagination}" columnNamen18n="#{label.employer}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/
employer- otxt	<h:out- putText>	Arbeitgeber des Nut- zers.	value="#{(var.employer)}"	rendered="#{userList- Backing.sessionInformati- on.user.isEditor}"	/

Chwa

Tabelle 36: **initialConfig.xhtml** Auf dieser Seite landet man beim ersten Systemstart. Sie ist nur für Administratoren zugänglich. Hier kann der Administrator die Datenbankschemata erstellen lassen.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
db-connection-state-otxt	<h:outputText>	Information über die Verbindung mit der Datenbank	value="#{initialConfigBacking.getState}"	/	/
create-db-btn	<h:commandButton>	Erstellt die Datenbankschemata	value="#{label.createDB}" action="#{initialConfigBacking.createDataSource}"	/	/

Wird
Redundantes Felder
in eigenes Profil addieren

Min. 6
Max. 16

Tabelle 37: newUser.xhtml Der Administrator kann hier einen neuen Nutzer anlegen.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
password-isrt	<h:inputSecret>	Angabe des Passworts.	value="#{newUserBacking.newUser.passwordNotHashed}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	passwordValidator required="true" requiredMessage="#{message.password}"
firstname-itxt	<h:inputText>	Angabe des Vornamen.	value="#{newUserBacking.newUser.firstName}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	required="true" requiredMessage="#{message.firstName}"
lastname-itxt	<h:inputText>	Angabe des Nachnamen.	value="#{newUserBacking.newUser.lastName}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	required="true" requiredMessage="#{message.lastName}"
titel-itxt	<h:inputText>	Angabe eines Titels.	value="#{newUserBacking.newUser.title}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	/
e-mail-itxt	<h:inputText>	Angabe einer E-Mail-Adresse.	value="#{newUserBacking.newUser.emailAddress}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	emailAddressExistsValidator emailAddressLayoutValidator required="true" requiredMessage="#{message.email}"
is-admin-lbl	<p:outputLabel>	Label für die Administrator Checkbox	value="#{label.setAdmin}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	/
is-admin-cbx	<h:selectBooleanCheckbox>	Zuordnung einer Administratorrolle.	value="#{newUserBacking.admin}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	/
abort-btn	<h:commandButton>	Abbruch des Vorgangs: Nutzer erstellen.	value="#{newUserBacking.abort}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	/
save-btn	<h:commandButton>	Speichern des neuen Nutzers.	value="#{newUserBacking.saveUser}"	render="#{newUserBacking.sessionInformation.user.isAdmin}"	/

Tabelle 38: administration.xhtml Hier kann ein Administrator Konfigurationen vornehmen.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
select-theme-selctom	<h:selectOneMenu>	Auswahl eines Farbthemas.	value="#{administrationBacking.systemSettings.style}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/
styles-1st	<f:selectItems>	Liste des Auswahlmöglichkeiten	value="#{administrationBacking.styles}"	/	/
logo-img	<h:graphicImage>	Logo des Systems.	value="#{administrationBacking.systemSettings.logoImage}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/
change-logo-ifile	<h:inputFile>	Hochladen eines neuen Logos.	title="#{help.uploadNewLogo}" value="#{administrationBacking.uploadedImage}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/
welcome-heading-itxt	<h:inputText>	Angabe einer Überschrift auf der <i>welcomepage</i>	value="#{administrationBacking.systemSettings.headlineWelcomePage}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	required="true" requiredMessage="#{message.welcomeHeading}"
welcome-text-itxt	<h:inputTextArea>	Angabe eines Textes auf der <i>welcomepage</i>	value="#{administrationBacking.systemSettings.shortMessageWelcomePage}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	required="true" requiredMessage="#{message.welcomeText}"
institution-itxt	<h:inputText>	Angabe des Namens der Einrichtung.	value="#{administrationBacking.systemSettings.companyName}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	required="true" requiredMessage="#{message.institution}"
imprint-itxt	<h:inputTextArea>	Angabe des Impressums der Einrichtung.	value="#{administrationBacking.systemSettings.imprint}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	required="true" requiredMessage="#{message.imprint}"
abort-btn	<h:commandButton>	Abbruch des Änderungsvorgangs.	value="#{label.abort}" action="#{administrationBacking.abort}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/
save-btn	<h:commandButton>	Speichern der Änderungen.	value="#{label.save}" action="#{administrationBacking.save}"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/
new-user-link	<h:link>	Weiterleitung zu <i>newUser.xhtml</i> .	value="#{label.gotoNewUser}" outcome="/facelets/authenticated/newUser"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/
new-scientific-forum-link	<h:link>	Weiterleitung zu <i>newScientificForum.xhtml</i> .	value="#{label.gotoNewScientificForum}" outcome="/facelets/authenticated/newScientificForum"	render="#{administrationBacking.sessionInformation.user.isAdmin}"	/

Navigation für Labels für Eingabefelder

redigiert in edit?

Tabelle 39: newScientificForum.xhtml Auf der Seite zum Erstellen eines wissenschaftlichen Forums kann ein Administrator dessen wesentlichen Daten festlegen.

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
forum-name-ixt	<h:inputText>	Angabe des Name des Forums.	value="#newScientificForumBacking.newScientificForum.name"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	required="true" requiredMessage="#{message.forumName}"
editors-list	<h:dataTable>	Liste der Editoren	value="#newScientificForumBacking.editors"	render="#toolbarBacking.sessionInformation.user.isAdmin"	/
email-editor-ixt	<h:inputText>	Angabe der E-Mail-Adresse eines Editor.	value="#newScientificForumBacking.newEditorInput.emailAddress"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	emailAddressValidator emailAddressLayoutValidator required="true" requiredMessage="#{message.email}"
add-editor-btn	<h:commandButton>	Hinzufügen eines Editors in eine Liste.	value="#{label.add}" action="#{newScientificForumBacking.addEditor}" title="#{help.addEditor}"	render="#{newScientificForumBacking.sessionInformation.user.isAdmin}"	/
delete-editor-btn	<h:commandButton>	Löschen des zugehörigen Editors von der Liste.	value="#{label.remove}" action="#{newScientificForumBacking.removeEditor}" title="#{help.removeEditor}"	render="#{newScientificForumBacking.sessionInformation.user.isAdmin}"	/
deadline-ixt	<h:inputText>	Hinzufügen einer Deadline des Forums.	value="#{newScientificForumBacking.newScientificForum.deadline}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	converterDateTime pattern="dd/MM/yyyy"
description-ixt	<h:inputTextArea>	Angabe einer Kurzbeschreibung.	value="#{newScientificForumBacking.newScientificForum.description}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	required="true" requiredMessage="#{message.description}"
url-ixt	<h:inputText>	Link zur Konferenz oder zum Journal.	value="#{newScientificForumBacking.newScientificForum.url}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	/
review-instructions-ixt	<h:inputTextArea>	Angabe einer Anleitung für eine Begutachtung.	value="#{newScientificForumBacking.newScientificForum.reviewManual}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	required="true" requiredMessage="#{message.instruction}"
science-field-list	<h:outputText>	Liste der ausgewählten Fachgebiete.	value="#{newScientificForumBacking.selectedScienceFields}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	/

Tabelle 40: Fortsetzung

ID	Typ	Beschreibung	Binding	Constraints	Validator Converter
science-field-list	<p:selectOneListBox>	Auswahl von Fachgebieten.	value="#{newScientificForumBacking.scienceFields}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	/
new-science-field-ixt	<h:inputText>	Hinzufügen neuer Fachgebiete.	value="#{newScientificForumBacking.newScienceFieldInput.name}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	required="true" requiredMessage="#{message.newScienceField}"
add-science-field-btn	<h:commandButton>	Ausführen der Hinzufüge-Aktion.	value="#{label.add}" action="#{newScientificForumBacking.addScienceField}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	/
save-btn	<h:commandButton>	Speichern des neuen Forums.	value="#{label.save}" action="#{newScientificForumBacking.create}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	/
abort-btn	<h:commandButton>	Abbruch des Erstellungsprozesses.	value="#{label.abort}" action="#{newScientificForumBacking.abort}"	render="#newScientificForumBacking.sessionInformation.user.isAdmin"	/

7.1 Konvertierung des ER-Diagramms

Das ER-Diagramm aus dem Entwurf wird hier in *Data Definition Language* übersetzt. Dabei werden die Details des Übertragungsprozesses dargestellt und erklärt.

7.1.1 SQL Standard

SQL ist vielfach standardisiert und alle Datenbank Management System weichen teilweise von diesen Standards ab und haben dadurch ihren eigenen *SQL Dialekt*. Im Folgenden wird der PostgreSQL-Dialekt verwendet, da PostgreSQL, wie im Pflichtenheft definiert, als SQL Server verwendet wird.

Abweichungen vom ISO SQL Standard sind Enumerationen, `TIMESTAMP` Zeitpunkte, `BYTEA` als binärer Datentyp und einzelne Unterschiede in der Syntax. ✓

7.1.2 Normalform

Das Datenbankschema befindet sich in *Boyce-Codd-Normalform* (BCNF). ✓

Durch die Verwendung von Surrogatschlüsseln in den Entities sind die vorkommenden funktionalen Abhängigkeiten offensichtlich. Für all diese gilt, dass die Surrogatschlüssel Superschlüssel der Relation sind. Die *UNIQUEs* bilden ebenfalls funktionale Abhängigkeiten und sind auch Superschlüssel der jeweiligen Relation.

7.1.3 Namenskonventionen

Die Tabellennamen sind immer im englischen Singular. Also beispielsweise *submission* anstatt *submissions*. Alle Bezeichner in SQL werden in *snake_case* geschrieben. Damit ist eine eindeutige optische Trennung zu Java's *camelCase* möglich. Die Werte von Enumerationen werden in *UPPER_CASE* geschrieben. ✓

7.1.4 Relationship-Typen

1 : n Relationships benötigen keine extra Tabellen in der Datenbank, da diese durch Referenz des Primärschlüssels der 1-Relation als Fremdschlüssel in der n-Relation abgespeichert werden können.

n : m Relationships benötigen dahingegen eine eigene Relation in der die Schlüsselpaare abgespeichert werden.

Abhängigkeiten, wie zum Beispiel Einreichungen, die zu einem Nutzer gehören, werden durch `ON DELETE CASCADE` modelliert. Wird der referenzierte Beziehungspartner (*User*) gelöscht, so werden auch kaskadierend alle abhängigen weiteren Daten (*Submissions*) automatisch gelöscht.

Wie wird User bei kaskadierenwachen Löschen gemacht?

7.1.5 Vererbung

Durch *IS-A* werden in ER-Diagrammen verschiedene Arten eines Typs definiert. Diese können dann in der *DDL* unterschiedlich übersetzt werden.

Nachteil join bei Unstetigkeit als Index
jeweils notwendig
alternativ Rollenspalte und ggf. null in
anderen Spalten

In unserem Fall wird die Rechtezuordnung der registrierten Nutzer durch jeweils eigene Entities Administrator, Editor und Reviewer anstatt einer Enumeration modelliert, da ein Nutzer mehrere Aufgaben in LasEs wahrnehmen kann.

Die Unterscheidung zwischen Journalen und Konferenzen findet nur durch eine gesetzte oder *NULL* Deadline statt, da sich beide funktional nicht weiter unterscheiden.

7.2 Entities

Hier werden die einzelnen Entities in der *Data Definition Language* beschrieben. Durch die Verwendung von *SERIAL* werden die Surrogatschlüssel der Relationen automatisch von PostgreSQL erstellt und die Primärschlüsseleigenschaft sichergestellt.

```
CREATE TABLE "user" (  
  id SERIAL PRIMARY KEY NOT NULL,  
  email_address VARCHAR UNIQUE NOT NULL,  
  firstname VARCHAR NOT NULL,  
  lastname VARCHAR NOT NULL,  
  title VARCHAR,  
  employer VARCHAR,  
  birthdate DATE,  
  avatar_thumbnail BYTEA,  
  is_registered BOOLEAN NOT NULL,  
  password_hash VARCHAR,  
  password_salt VARCHAR,  
  
  CONSTRAINT valid_name CHECK (length(firstname) >= 1 AND length(lastname) >= 1),  
  CONSTRAINT valid_birthdate CHECK (birthdate < CURRENT_DATE),  
  CONSTRAINT email_address_pattern CHECK (email_address LIKE '%@%._.%')  
);  
  
CREATE TABLE reviewer(  
  id INTEGER PRIMARY KEY NOT NULL,  
  FOREIGN KEY (id) REFERENCES "user"(id) ON DELETE CASCADE  
);  
  
CREATE TABLE editor(  
  id INTEGER PRIMARY KEY NOT NULL,  
  FOREIGN KEY (id) REFERENCES "user"(id) ON DELETE CASCADE  
);  
  
CREATE TABLE administrator(  
  id INTEGER PRIMARY KEY NOT NULL,  
  FOREIGN KEY (id) REFERENCES "user"(id) ON DELETE CASCADE  
);
```

notwendig bei primary key?

Lebenslaufdaten (Perfomanz?) wie werden in Faceted

auch in Faceted Content angezeigt

fast schon Rollenlogik, aber ok

Wirds evtl
→ besser
1 Unstetigkeit

Listing 8: DDL von Nutzern

```
CREATE TABLE verification (  
  id INTEGER NOT NULL,  
  validation_random VARCHAR,  
  verified BOOLEAN,  
  timestamp_validation_started TIMESTAMP,
```

bonuss geht nicht in DB!

```
unvalidated_email_address VARCHAR,  
  
PRIMARY KEY (id),  
FOREIGN KEY (id) REFERENCES "user"(id)  
);
```

Listing 9: DDL von Verifizierung

```
CREATE TABLE scientific_forum(  
  id SERIAL PRIMARY KEY NOT NULL,  
  name VARCHAR UNIQUE NOT NULL,  
  description VARCHAR,  
  url VARCHAR,  
  review_manual VARCHAR,  
  timestamp_deadline TIMESTAMP  
);
```

Listing 10: DDL von wissenschaftlichen Foren

```
CREATE TABLE science_field(  
  name VARCHAR PRIMARY KEY NOT NULL  
);
```

Listing 11: DDL von Fachgebieten

```
CREATE TYPE submission_state AS ENUM (  
  'SUBMITTED',  
  'REVISION_REQUIRED',  
  'REJECTED',  
  'ACCEPTED',  
);
```

Umwandlung bei Einlesen in Java Enums gefordert?
OT: Enums & Enums können importiert werden

```
CREATE TABLE submission (  
  id SERIAL PRIMARY KEY NOT NULL,  
  title VARCHAR NOT NULL,  
  state submission_state NOT NULL,  
  timestamp_submission TIMESTAMP,  
  requires_revision BOOLEAN NOT NULL DEFAULT FALSE,  
  timestamp_deadline_revision TIMESTAMP,  
  
  author_id INTEGER REFERENCES "user"(id) ON DELETE CASCADE,  
  editor_id INTEGER REFERENCES editor(id) ON DELETE SET NULL,  
  forum_id INTEGER REFERENCES scientific_forum(id) ON DELETE CASCADE  
);
```

Listing 12: DDL von Einreichungen

```
CREATE TABLE paper (  
  version INTEGER NOT NULL,  
  submission_id SERIAL NOT NULL,  
  timestamp_upload TIMESTAMP NOT NULL DEFAULT NOW(),  
  is_visible BOOLEAN NOT NULL,  
  pdf_file BYTEA NOT NULL,
```



```

PRIMARY KEY (submission_id, version),
FOREIGN KEY (submission_id) REFERENCES submission(id) ON DELETE CASCADE
);

```

Listing 13: DDL von Papers

```

CREATE TABLE review (
  reviewer_id INTEGER REFERENCES reviewer(id) ON DELETE CASCADE,
  version INTEGER NOT NULL,
  submission_id INTEGER NOT NULL,

  timestamp_upload TIMESTAMP NOT NULL DEFAULT NOW(),
  is_visible BOOLEAN NOT NULL,
  is_recommended BOOLEAN NOT NULL,
  comment VARCHAR,
  pdf_file BYTEA,

  PRIMARY KEY (submission_id, version, reviewer_id),
  FOREIGN KEY (submission_id, version)
  REFERENCES paper(submission_id, version) ON DELETE CASCADE
);

```

Listing 14: DDL von Gutachten

```

CREATE TABLE system(
  id INTEGER PRIMARY KEY,
  company_name VARCHAR,
  welcome_heading VARCHAR,
  welcome_description VARCHAR,
  imprint VARCHAR,
  logo_image BYTEA
);

```

Listing 15: DDL von den Systemeinstellungen

7.3 Indizes

Um die E-Mail-Adressen unabhängig von der Groß- oder Kleinschreibung in der Datenbank tatsächlich *UNIQUE* zu halten benötigt es in PostgreSQL einen eigenen Index, da das *UNIQUE* Schlüsselwort selbst *case-sensitive* ist.

```

CREATE UNIQUE INDEX email_unique_case_insensitive
ON "user" (LOWER(email_address));

```

Listing 16: DDL des UNIQUE Index auf E-Mail-Adressen

7.4 Relationships

n : m Relationships benötigen eigenen Relationen.

gut Syntax-Highlighting

```
CREATE TABLE co_authored(  
    user_id INTEGER NOT NULL,  
    submission_id INTEGER NOT NULL,  
  
    PRIMARY KEY (user_id, submission_id),  
    FOREIGN KEY (user_id) REFERENCES "user"(id) ON DELETE CASCADE,  
    FOREIGN KEY (submission_id) REFERENCES submission(id) ON DELETE CASCADE  
);
```

Listing 17: DDL der Ko-Autoren

```
CREATE TABLE interests (  
    user_id INTEGER NOT NULL,  
    science_field_name VARCHAR NOT NULL,  
  
    PRIMARY KEY (user_id, science_field_name),  
    FOREIGN KEY (user_id) REFERENCES "user"(id) ON DELETE CASCADE,  
    FOREIGN KEY (science_field_name) REFERENCES science_field(name)  
    ON DELETE CASCADE  
);
```

Listing 18: DDL der Interessensgebiete

```
CREATE TABLE member_of(  
    editor_id INTEGER NOT NULL,  
    scientific_forum_id INTEGER NOT NULL,  
  
    PRIMARY KEY (editor_id, scientific_forum_id),  
    FOREIGN KEY (editor_id) REFERENCES editor(id) ON DELETE CASCADE,  
    FOREIGN KEY (scientific_forum_id) REFERENCES scientific_forum(id)  
    ON DELETE CASCADE  
);
```

Listing 19: DDL der verwaltenden Editoren

```
CREATE TYPE review_task_state AS ENUM (  
    'PENDING',  
    'ACCEPTED',  
    'REJECTED'  
);  
  
CREATE TABLE reviewed_by(  
    reviewer_id INTEGER NOT NULL,  
    submission_id INTEGER NOT NULL,  
    has_accepted review_task_state,  
    timestamp_deadline TIMESTAMP,  
  
    PRIMARY KEY (reviewer_id, submission_id),  
    FOREIGN KEY (reviewer_id) REFERENCES reviewer(id) ON DELETE CASCADE,  
    FOREIGN KEY (submission_id) REFERENCES submission(id) ON DELETE CASCADE  
);
```

Listing 20: DDL der begutachtenden Gutachter

keine Views? stand procedures? (du nicht)

```
CREATE TABLE topics(  
  science_field_name VARCHAR NOT NULL,  
  scientific_forum_id INTEGER NOT NULL,  
  
  PRIMARY KEY (science_field_name, scientific_forum_id),  
  FOREIGN KEY (science_field_name) REFERENCES science_field(name)  
  ON DELETE CASCADE,  
  FOREIGN KEY (scientific_forum_id) REFERENCES scientific_forum(id)  
  ON DELETE CASCADE  
);
```

Listing 21: DDL der Themengebiete eines Forums



8 Datenfluss

Thomas Kirz

8.1 Hochladen einer Revision

Als Beispiel für einen Datenfluss wird in Abbildung 3 ein Sequenzdiagramm zu folgendem Vorgang betrachtet: Ein angemeldeter Nutzer ruft die Seite einer eigenen Einreichung auf, für die eine Revision angefordert wurde. Dort fügt er eine Revision, also ein neues Paper hinzu. Nach Erstellung der `UIMessage` wird ein Event gefeuert mit:

```
Event<UIMessage>#fire()
```

Dieses wird vom `UIMessageGenerator` observed, welcher dann eine `FacesMessage` hinzufügt, die nach dem Hochladen auf der Einreichungsseite angezeigt wird.

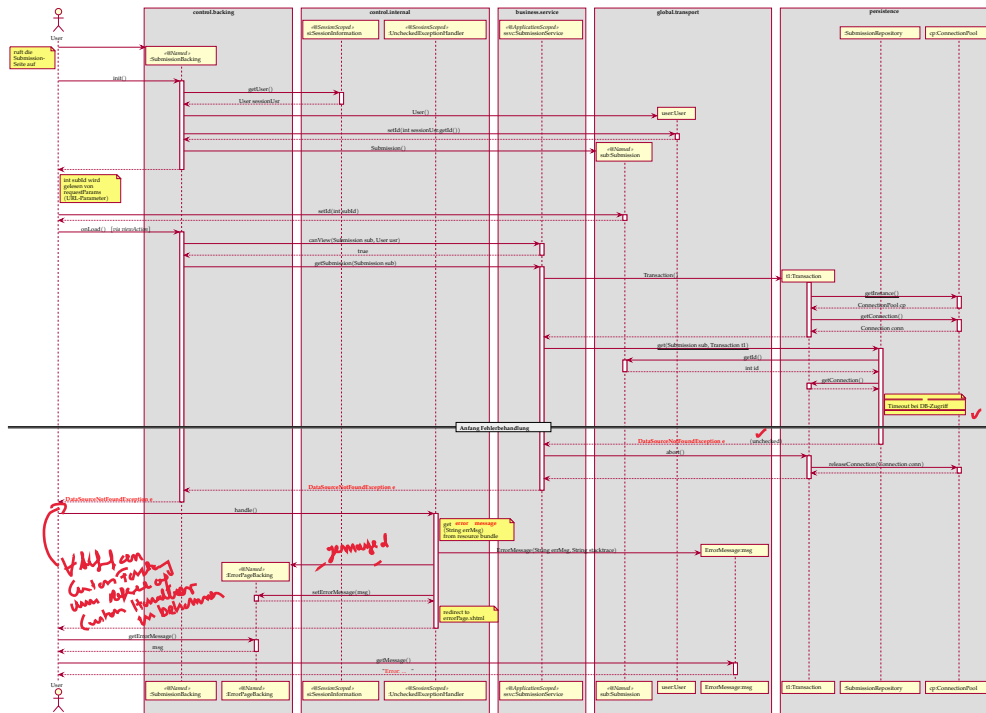


Abbildung 4: Sequenzdiagramm zum Aufruf einer Einreichungsseite mit Fehler beim Datenbankzugriff

9 Security

Thomas Kirz

Im Folgenden wird erklärt, wie verschiedene Sicherheitsangriffe verhindert und Sicherheitsrisiken ausgeschlossen werden.

9.1 Cross Site Scripting (XSS)

Da ausschließlich JSF components für die HTML-Ausgabe verwendet werden, bereinigt JSF alle Nutzereingaben. Zum Beispiel werden '<' und '>' durch '<' und '>' ersetzt, Nutzereingaben können also keine HTML-Tags in der Ausgabe erstellen. Dadurch wird XSS bzw. HTML-Injection unmöglich. Auch JavaScript kann so nicht injected werden, da sich JavaScript-Code in einem HTML-Tag (<script>) befinden müsste.

9.2 SQL-Injection

Für SQL-Code werden PreparedStatement benutzt, damit die SQL-Statements ohne von Nutzereingaben abhängige Parameter vorkompiliert werden. Die Parameter werden dann an die Datenbank gesendet und von ihr getrennt vom Code als Daten betrachtet. Die Ausführung des Statements kann also nicht durch die Nutzereingaben modifiziert werden und eine SQL-Injection wird verhindert

9.3 Session Hijacking & Fixation

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. JSF setzt für das Session-Cookie standardmäßig das `http-only`-Flag, damit kann darauf nicht mit JavaScript zugegriffen werden und Hijacking wird erschwert.

Nach jeder Anmeldung wird die Session verworfen und neu generiert mit ✓

```
FacesContext.getCurrentInstance().getExternalContext()
    .getRequest().changeSessionId()
```

Nach einer Authentifizierung bleibt also keine alte Session bestehen und *Session-Fixation*-Angriffen werden verhindert. ✓

9.4 Passwörter

Passwörter werden mit der modernen und sicheren PBKDF2-Funktion gehasht. Dabei wird für jedes Passwort ein eigener zufällig generierter Salt benutzt, welcher mit dem Hash in der Datenbank gespeichert wird. Eingegebene Passwörter werden so früh wie möglich gehasht, damit die Klartextpasswörter eine minimale Lebenszeit auf dem Server haben. Für das Generieren eines Salts und das Hashing bietet die Klasse `Hashing` aus `de.java.business.util` Methoden an. Sie verwendet die robuste `javax.crypto`-Bibliothek.

Um das Erraten von Passwörtern durch Brute-Force-Angriffe o.ä. zu erschweren, müssen Passwörter 8–100 Zeichen lang sein und Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen enthalten.

Handwritten note: *Verweis auf Verzeichnis* ✓

9.5 Information Leakage


Alle Informationen über das System, die nach außen gelangen, können Angreifern helfen. Dazu gehören Details über die technische Umgebung und verwendete Software des Systems, interne Fehlermeldungen und Implementierungsdetails. Folgende Maßnahmen werden ergriffen, um Leakage solcher Informationen zu verhindern. Fehlermeldungen zeigen im Produktionsmodus keine Stacktraces an, sondern eigene Nachrichten, die nur Informationen enthalten, die für den Endnutzer relevant sind. Wir verwenden eigene Fehlerseiten bei Exceptions und ersetzen die 404 und 500 Fehlerseiten von JSF durch unsere eigenen wie folgt in der `web.xml`-Datei:

```
<error-page>
  <error-code>404</error-code>
  <location>/WEB-INF/errorpages/error404.xhtml</location>
</error-page>
```

✓

Dadurch ist nicht ersichtlich, dass wir JSF/Mojarra verwenden. Das wird auch durch die Verwendung von `.xhtml`-Dateiendungen für unsere Facelets und durch das Ersetzen des Bezeichners `JSESSIONID` für Session-Cookies und URL-Rewriting in der `web.xml`-Datei verborgen:

```
<session-config>
  <cookie-config>
    <name>sessionId</name>
  </cookie-config>
</session-config>
```



9.6 Access Control


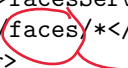
Um unerlaubten Zugriff auf Seiten, Informationen und Aktionen zu verhindern, werden Zugriffe auf nicht-öffentliche Ressourcen (das ist alles außer Start-, Registrierungs- und Verifikationsseite) standardmäßig abgelehnt. Die Klasse `TrespassListener` entscheidet dann, ob der Nutzer die nötigen Rechte hat und akzeptiert eine Anfrage nur dann, sonst wird auf eine 404-Fehler-Seite weitergeleitet. Dazu wird über die angefragte URL das zugehörige Facelet bestimmt und geprüft, ob darauf mit der Rolle des Nutzers zugegriffen werden darf, wie in Kapitel 6 definiert. Einzelne Komponenten können im Facelet mit dem `rendered`-Attribut nur für bestimmte Rollen sichtbar gemacht werden. Auf die Rolle kann der `TrespassListener` über die `SessionInformation` zugreifen. Der Zugriff hängt darüber hinaus von der Beziehung des Nutzers zur angefragten Ressource ab (ein normaler Nutzer kann z.B. nur sein *eigenes* Profil oder eine *eigene* Einreichung sehen). Deshalb prüft für die relevanten Seiten das jeweilige Backing Bean diese Beziehung und leitet ggf. auf eine Fehlerseite weiter. Das verhindert *Insecure Direct Object Reference*, da IDs von Submissions, Scientific Forums und Profilen über URL-Parameter preisgegeben werden.



init()
Methoden

Durch den Eintrag

```
<servlet>
  <servlet-name>facesServlet</servlet-name>
  <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>facesServlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
```



verrät Technologie

in der `web.xml`-Datei werden Aufrufe auf alle unsere Facelet-Dateien dem Faces Servlet weitergeleitet. Damit wird aufgrund der Entscheidung des `TrespassListeners` entweder das Facelet richtig als HTML-Datei gerendert, oder zur 404-Seite weitergeleitet. Ein Herunterladen der eigentlichen auf dem Server gespeicherten Facelet-Datei ist aber nicht möglich.

9.7 Vergabe von Administratorrechten

Administratoren haben die Möglichkeit, Nutzern auf ihrer Profilseite Administratorrechte zu gewähren oder zu entfernen. Da das Ändern von Adminrechten sicherheitskritisch ist, müssen sie dafür ihr Passwort erneut angeben.