



Entwurf

DocThesisTracker



DOCTHESIS TRACKER
DTT

Client: Dr. Armin Größlinger

Team 2

Hadi Abou Hassoun
Haroun Alswedany
Alaa Qasem
Johannes Silvennoinen
Stefan Witka

Inhaltsverzeichnis

1	Einleitung	4
2	Systemarchitektur	5
2.1	Architektur-Überblick	5
2.2	Paketaufteilung	5
2.3	Datenübertragung	6
2.4	Verbindungspool	6
2.5	Fehlerbehandlung	6
2.6	Diagramme	7
2.7	Fazit	8
3	Statisches Klassendiagramm in UML	9
3.1	Klassendiagramm	9
3.2	Klassenbeschreibungen	10
4	JF-Seiten	17
4.1	Login-Seite	17
4.2	PasswortVergessen-Seite	17
4.3	resetPassword-Seite	18
4.4	Registrierung-Seite	18
4.5	Email Freischaltung-Seite	19
4.6	Profile	19
4.7	Error-Seite	19
4.8	Header	20
4.9	Template	20
4.10	footer	20
4.11	Pagination.xhtml	21
4.12	Seitenverwaltung	21
4.13	Benutzerverwaltung	21
4.14	Circulationlist	22
4.15	Umlaufferstellung-Seite	22
4.16	circulationDetails.xhtml	23
4.17	Hilfe	24
5	Systemfunktionen	25
5.1	Systemstart	25
5.2	Systemstop	25
5.3	Wartungsthread	26
5.4	Zugriffsrechte	26

5.5	Logging	27
5.6	Sicherheitsaspekte	27
6	Datenfluss	28
6.1	Anzeigen eines Umlaufs	28
6.2	Anzeigen eines Umlaufs (Fehlerfall)	29
7	ER-Modell	30
7.1	Entities	31

1 Einleitung

Stefan Witka

Dieses Dokument stellt den ersten Entwurf des Online Umlaufprogramms für Dissertationen DocThesisTracker dar.

Im folgenden werden die Systemarchitektur, die Systemkomponenten und die Funktionsweise des Systems erläutert oder Graphisch dargestellt.

2 Systemarchitektur

Johannes Silvennoinen

2.1 Architektur-Überblick

Unsere Architektur basiert auf dem Model-View-Controller (MVC) Pattern, das eine Trennung von Anwendungslogik, Daten und Darstellung ermöglicht. Die Model-Schicht enthält die Geschäftslogik und den Datenzugriff, während die View-Schicht für die Präsentation der Benutzeroberfläche zuständig ist. Der Controller verbindet die beiden Schichten und koordiniert die Aktionen des Benutzers.

Um eine effiziente und sichere Verbindung zur Datenbank herzustellen, verwenden wir einen Connection Pool als Singleton. Dies stellt sicher, dass mehrere Verbindungen zur Datenbank gleichzeitig genutzt werden können, ohne dass es zu Konflikten kommt. Der Pool wird auch verwendet, um Verbindungsfehler zu vermeiden und die Ressourcen effektiv zu nutzen.

Um Daten zwischen den Schichten zu übertragen, verwenden wir Data Transfer Objects (DTOs). Diese sind einfache POJOs (Plain Old Java Objects), die nur Daten und keine Geschäftslogik enthalten. Sie dienen als Schnittstelle zwischen der Model-Schicht und der View-Schicht und erleichtern die Datenübertragung und -manipulation.

2.2 Paketaufteilung

Die gesamte Anwendung ist in drei Hauptpakete unterteilt: View, Controller und Model.

Im View-Paket befinden sich alle XHTML-Seiten, die der Benutzer sehen und mit denen er interagieren kann. Diese Seiten sind für die Darstellung von Informationen und die Benutzereingabe verantwortlich.

Der Controller wird von Jakarta Server Faces (JSF) verwaltet und ist für die Steuerung der Interaktionen zwischen der View und dem Model zuständig.

Das Model-Paket ist in zwei Schichten unterteilt: Business Layer und Data Access Layer. Der Business Layer enthält die Backing Beans und ist für die Implementierung der Geschäftslogik zuständig. Diese Schicht enthält auch Validatoren

für die Eingabevalidierung und die Fehlerbehandlung.

Der Data Access Layer ist für die Datenbankbindung zuständig und ist in Unterpakete für jede unterstützte Datenbankplattform unterteilt. Das PostgreSQL-Unterpaket enthält alle Klassen und Schnittstellen, die für die Interaktion mit einer PostgreSQL-Datenbank erforderlich sind. Es ist jedoch möglich, in Zukunft Unterpakete für andere Datenbankplattformen hinzuzufügen, um die Kompatibilität der Anwendung zu erweitern.

Durch diese Paketaufteilung wird eine klare Trennung von Aufgaben und Verantwortlichkeiten erreicht, was die Wartbarkeit und Erweiterbarkeit der Anwendung verbessert.

2.3 Datenübertragung

Zur Übertragung von Daten zwischen den Schichten werden Data Transfer Objects (DTOs) genutzt. Diese sind einfache POJOs (Plain Old Java Objects), die keine JEE- oder JPA-Managed Beans sind und außerhalb der Data Access Schicht und Business Schicht liegen.

2.4 Verbindungspool

Für die Verbindung zur Datenbank wird ein Connection Pool als Singleton (thread-sicher) verwendet. Dadurch wird sichergestellt, dass die Verbindung zur Datenbank effizient genutzt wird und keine unnötigen Verbindungen aufgebaut werden.

2.5 Fehlerbehandlung

Die Fehlerbehandlung in der Systemarchitektur wird mithilfe von Validatoren implementiert. Dabei wird für jedes Eingabeformular ein eigener Validator erstellt, der die Eingaben des Nutzers auf Plausibilität und Vollständigkeit prüft. Falls ein Fehler auftritt, wird dem Nutzer eine entsprechende Fehlermeldung angezeigt und er wird aufgefordert, die fehlerhaften Eingaben zu korrigieren.

Die Validatoren sind in den Backing Beans für den Business Code integriert. Diese Beans sind für die Verarbeitung der Geschäftslogik und der Nutzerinteraktion zuständig. Wenn eine Eingabe vom Nutzer empfangen wird, ruft das entsprechende Backing Bean den passenden Validator auf.

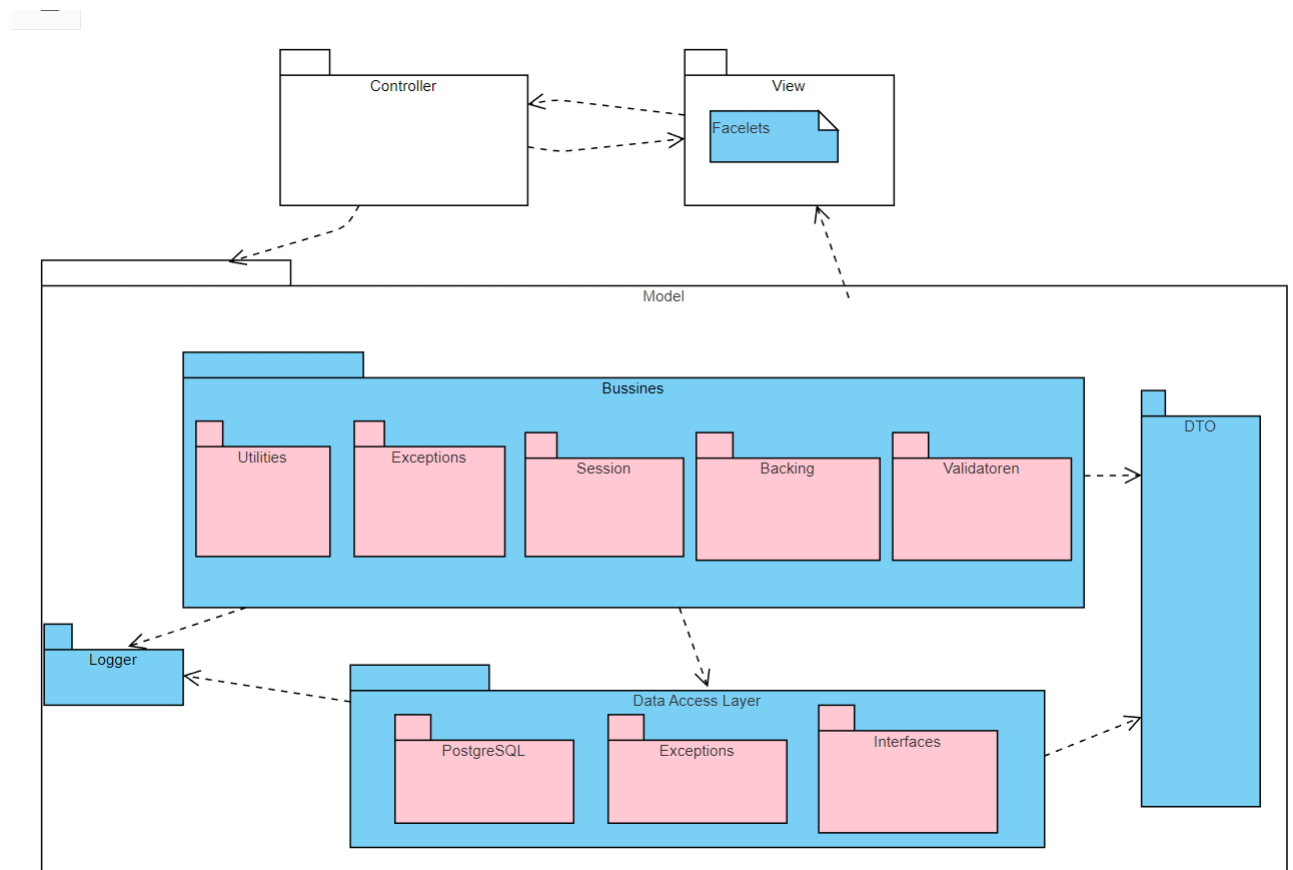
Die Validatoren werden auch für die Datenüberprüfung verwendet, bevor sie in die Datenbank eingefügt werden. Dadurch wird sichergestellt, dass die Daten in

der richtigen Form vorliegen und keine Datenbankfehler auftreten.

Zusätzlich werden auch systemweite Fehlerbehandlungen implementiert, um unerwartete Fehler abzufangen und dem Nutzer eine angemessene Fehlermeldung anzuzeigen. Dabei wird ein Globaler-Fehler-Handler eingesetzt, der alle unerwarteten Fehler abfängt und eine Fehlerseite mit einer entsprechenden Fehlermeldung anzeigt. Die Fehlermeldungen sind dabei für den Nutzer verständlich formuliert und geben Hinweise auf die Ursache des Fehlers.

Insgesamt gewährleistet die Verwendung von Validatoren eine zuverlässige und benutzerfreundliche Fehlerbehandlung in der Systemarchitektur.

2.6 Diagramme



2.7 Fazit

Die Systemarchitektur basiert auf bewährten Entwurfsmustern und ermöglicht eine effiziente und flexible Entwicklung. Die klare Trennung von Aufgaben und die Verwendung von DTOs und DAOs erleichtert die Wartung und Erweiterung des Systems.

3.2 Klassenbeschreibungen

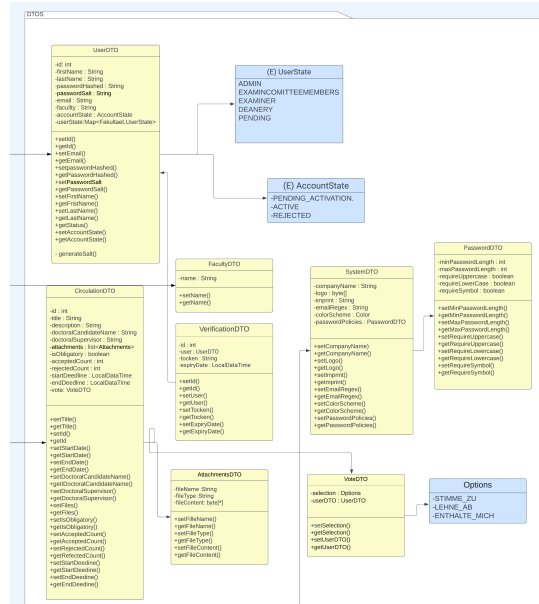


Abbildung 2: DTO's

3.2 Klassenbeschreibungen 3 STATISCHES KLASSENDIAGRAMM IN UML

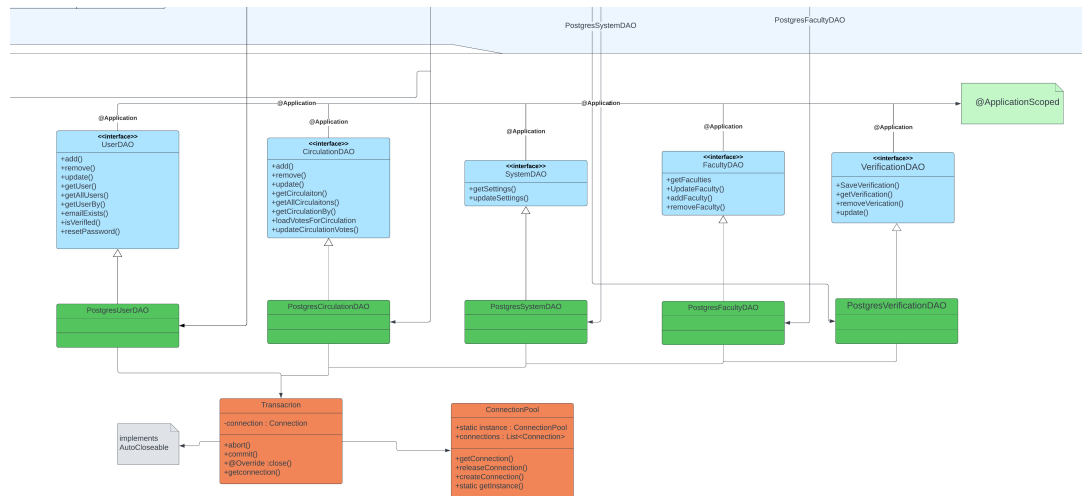


Abbildung 3: DAO'S

Klasse	Beschreibung
SystemDTO	This DTO represents a System.
FacultyDTO	This DTO represents a Faculty.
CirculationDTO	This DTO represents a Circulation.
UserDTO	This DTO represents a User.
(E)UserState	This Enum-Class represents the State of the User .
(E)AccountState	This Enum-Class represents the State of the User Account.
VerificationDTO	Is used to store verification information for a user.
PasswordDTO	To determine password guidelines.
AttachmentsDTO	This DTO represents the attachments associated with a circulation.
Options	This Enum-Class represents the voting options
VoteDTO	This DTO encapsulates a voting action for a circulation.

3.2 Klassenbeschreibungen 3 STATISCHES KLASSENDIAGRAMM IN UML

*Postgres offers a robust implementation of DAOs that facilitates efficient and organized management of database access in object-oriented programming. With this implementation, that makes easily create and manage DAOs for Postgres databases, resulting in improved application performance and scalability.

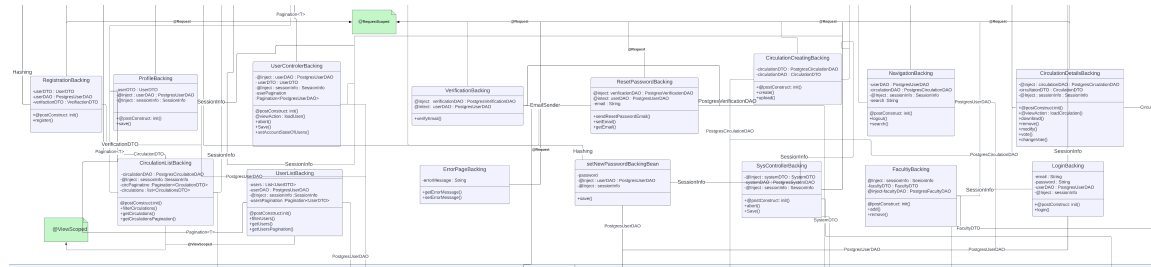


Abbildung 4: BackingBeans

Klasse	Beschreibung
SystemDAO	get and Update the System Setting.
FacultyDAO	Allows CRUD operations on Faculty.
CirculationDAO	Allows CRUD operations on Circulation..
UserDAO	Allows CRUD operations on Users.
VerificationDAO	Bundles information about the verification process .
Transaction	Used to ensure data integrity and consistency by either committing all the changes made during the transaction or rolling back the changes if an error occurs.
connectionPool	Provides a mechanism to efficiently manage connections to the database.

Klasse	Beschreibung
LoginBacking	Backing bean for the Login page.
RegistrationBacking	Backing bean for the registration page.
ProfileBacking	Backing bean for the profile page..
UserControlerBacking	Backing bean for the user management page
CirculationListBacking	Backing bean for the list of Circulations.
CirculationDetailsBacking	Backing bean for the page for vote or modify a Cerculation.
CirculationCreatingBacking	Backing bean for the page for creating a new Circulation
SysControllerBacking	Backing bean for the administration page.
NavigationBacking	Backing bean for the navigation bar.
ErrorMessageBacking	Backing bean To display an error message on a page when an error occurs.
FacultiyBacking	Backing bean for the Faculty management page
setNewPasswordBackingBean	Backing bean to save the new Paswword
ResetPasswordBacking	To validate the email address before resetting the password.
VerificationBacking	Backing bean for verifying the user's email address.
UserListBacking	Backing bean for the list of Users.

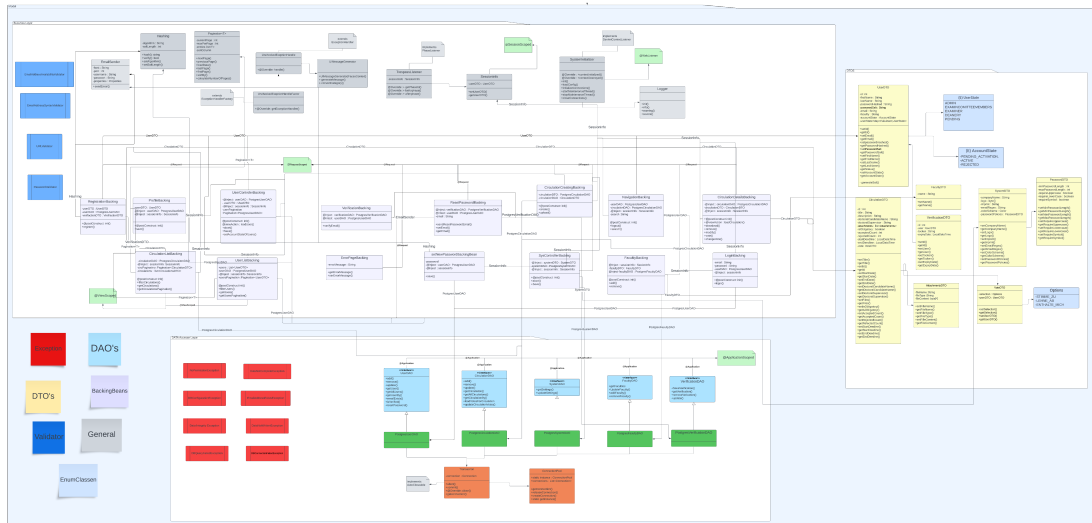


Abbildung 5: Validator

Klasse	Beschreibung
URLValidator	is used to validate whether a given string is a valid.
EmailAddressAvailabilityValidator	Validates that a given email is not already in use within the system.
EmailAddressSyntaxValidator	Ensures that a provided string represents a valid email address.
PasswordValidator	Verifies whether a password satisfies the minimum criteria for password strength.

3.2 Klassenbeschreibungen 3 STATISCHES KLASSENDIAGRAMM IN UML

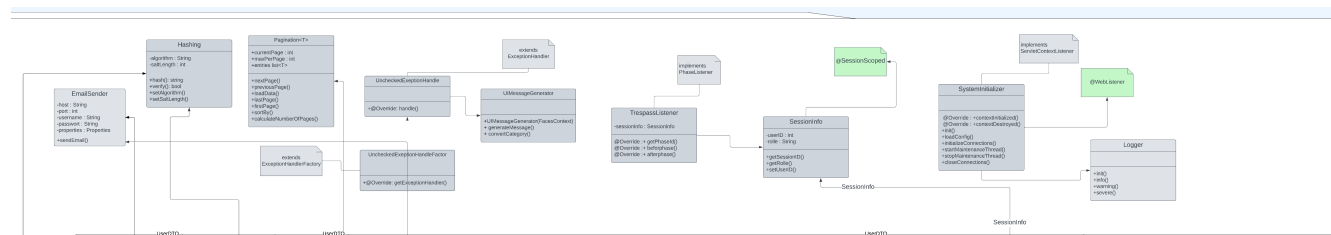


Abbildung 6: General

Klasse	Beschreibung
TrespassListener	Controls the authorization of users to access various resources, particularly web pages.
SessionInfo	Wraps data saved in the Session.
Hashing	Provides support in the hashing of passwords.
EmailSender	Provides support in the Sending of Emails
UncheckedExceptionHandler	Catches all unchecked Exceptions and redirects the User to an error page.
UncheckedExceptionHandlerFactory	A factory for the UncheckedExceptionHandler.
IMessageGenerator	This component observes all events related to IMessage and generates Faces Messages based on them.
Pagination	This is a generic class designed to implement a paginated list of items of a specific type.
Logging	This Logger allows logging in different log Levels.
SystemInitializer	System Initializer is a web listener component responsible for initializing the system when the Tomcat server starts.



Abbildung 7: Exceptions

Klasse	Beschreibung
NoPermissionException	to implement proper authorization, that users or processes can only perform actions or access resources that they are authorized to access.
DBConfigurationException	Is thrown when bad configuration of the database leads to a failure to establish a connection.
DataNotCompleteException	Triggered by the application when it attempts to access or work with data that is not complete
DBQueryFailedException	This exception is usually triggered by the application when it attempts to process a request that contains query parameters that do not conform to the expected format or values. Query parameters are often used in web applications to provide additional information to the server when handling requests.
Data Integrity Exception	This exception is usually triggered by the application or database when it detects that data has been modified or deleted in a way that violates the expected constraints or relationships between data elements.
DBConnectionFailed Exception	Is thrown when the database connection cannot be established or times out.
DataNotWrittenException	This exception is usually triggered by the application when it attempts to write data to a file, database, or other storage medium and the operation fails for some reason.
EmailAddressExistsException	Is thrown when the repository tries to save an email address to the database that already exists in the database.

4 JF-Seiten

Alaa Qasem

Im Folgendem Abschnitt werden Abkürzungen für die verschiedenen Rollen eingeführt: **A** steht für Administrator, **D** für Dekanat, **PK** für einen Prüfungskommissionsmitglieder, **P** für Prüfungsberechtigte und **AN** für anonymen Nutzer. Ist eine Funktion für alle Benutzerrollen, so werden diese Außer anonymen Nutzer unter dem Begriff **Alle** zusammengefasst.

4.1 Login-Seite

login.xhtml Auf der Login-Seite wird das System vorgestellt. Es gibt ein Login-Formular zur Anmeldung im System. Für nicht registrierte Nutzer wird man über einen Button zu registration.xhtml weitergeleitet.

login.xhtml			
Element	Typ	Beschreibung	Rechte
Head	outputText	Überschrift der Login-Seite.	Alle -AN
Email	inputText	Hier kann der Benutzer seine Email eingeben.	Alle -AN
Password	inputSecret	Hier kann der Benutzer seine Passwort eingeben.	Alle -AN
goToForgetPass	link	Weiterleitung zur Passwort Vergessen-Seite	Alle
Login	commandButton	Ausführen des Login-Prozesses.	Alle -AN
Register	commandButton	Wenn dieser Button gedrückt wird, wird der Benutzer zur Registrierung-Seite weitergeleitet.	Alle -AN

4.2 Passwort Vergessen-Seite

forgetPass.xhtml			
Element	Typ	Beschreibung	Rechte
Head	outputText	Überschrift der forgetPass-Seite.	Alle
Email	inputText	Angabe der Email.	Alle
forgetPass	commandButton	Wenn dieser Button gedrückt wird, wird der Prozess zum Zurücksetzen des Passworts gestartet	Alle

4.3 resetPassword-Seite

resetPassword.xhtml			
Element	Typ	Beschreibung	Rechte
Password	inputSecret	Ein leeres Feld für die Eingabe des neuen Passworts des Benutzers.	Alle
savePass	commandButton	Speichert das neue Passwort	Alle
resetSuccess	outputText	Bestätigungstext für das erfolgreiche Passwort-Zurücksetzen.	Alle
failMsg	outputText	Das eingegebene Passwort entspricht nicht den erforderlichen Anforderungen.	Alle
loginPage	link	Weiterleitung zur Login Seite.	Alle

4.4 Registrierung-Seite

registration.xhtml Seite zur Registrierung anonymer Nutzer.

registration.xhtml			
Element	Typ	Beschreibung	Rechte
Titel	inputText	Angabe eines Titels.	AN
Firstname	inputText	Hier kann der Benutzer seinen Name eingeben.	AN
Name	inputText	Hier kann der Benutzer seinen Nachname eingeben.	AN
Email	inputText	Hier kann der Benutzer seine Email eingeben.	AN
Fakultät	selectOneMenu	Hier kann der Benutzer seine Fakultät auswählen.	AN
register	commandButton	Wenn dieser Button gedrückt wird, wird der Registrierungsprozess ausgeführt.	AN
loginPage	link	Weiterleitung zur Login Seite.	AN

4.5 Email Freischaltung-Seite

validation.xhtml Wird angezeigt, wenn die E-Mail-Verifizierung erfolgreich abgeschlossen wurde.

validation.xhtml			
Element	Typ	Beschreibung	Rechte
successText	outputText	Bestätigungstext für die erfolgreiche Verifizierung der E-Mail.	AN

4.6 Profile

profile.xhtml Die Profil-Seite eines angemeldeten Nutzers kann nur vom Nutzer selbst oder vom Administrator bearbeitet werden. Alle zuvor gespeicherten Daten werden in den entsprechenden Feldern angezeigt.

profile.xhtml			
Element	Typ	Beschreibung	Rechte
Firstname	inputText	Hier kann der Benutzer seinen Vorname sehen bzw. ändern.	Alle
Name	inputText	Hier kann der Benutzer seinen Name sehen bzw. ändern.	Alle
Password	inputSecret	Ein leeres Feld für die Eingabe des neuen Passworts des Benutzers.	Alle
Email	inputText	Hier kann der Benutzer seine Email ändern.	Alle
Fakultät	outputText	Der Benutzer erhält hier einen Überblick über die Fakultäten, denen er angehört.	Alle
Save	commandButton	Wenn dieser Button gedrückt wird, wird die neue veränderte Daten gespeichert.	Alle

4.7 Error-Seite

errorPage.xhtml			
Element	Typ	Beschreibung	Rechte
errorHead	outputText	Überschrift der Error-Seite.	Alle
errorMsg	outputText	Was ist fehlgeschlagen.	Alle

Haroun Alswedany

4.8 Header

Header.xhtml			
Element	Typ	Beschreibung	Rechte
logo	graphicImage	Logo der Applikation	alle
logoutButton	commandButton	Loggt den Nutzer aus dem System	alle
Umlaufliste	link	link zur Umlaufliste	A-D
hilfe	link	link zur Hilfseite	alle
searchField	inputText	suchefunktion	alle
search	commandButton	Suche ausführen.	alle
profil	link	Link zum Profil	alle
Hauptseite	link	Link zur Hauptseite	alle

4.9 Template

main.xhtml

main.xhtml		
Element	Typ	Beschreibung
navigation	include	kopfzeile
body	insert	Inhalt der Seite.
footer	include	Fußzeile.

4.10 footer

footer.xhtml

footer.xhtml		
Element	Typ	Beschreibung
imprintLink	link	Link zur Seite des Impressums.
language	outputText	Anzeige der Sprache.

4.11 *Pagination.xhtml*

Pagination.xhtml			
Element	Typ	Beschreibung	Recht
datatable	datatable	...	alle
firstpage	commandButton	Zurück zur ersten Seite.	alle
prevpage	commandButton	Eine Seite zurück.	alle
currentpage	OutputText	aktuelle Seite.	alle
nextpage	commandButton	Eine Seite weiter.	alle
lastpage	commandButton	Zur letzten Seite.	alle

4.12 *Seitenverwaltung*

Seitenverwaltung.xhtml			
Element	Typ	Beschreibung	Recht
selectTheme	selectOneMenu	Auswahl eines Farbthemas.	A
logo	graphicImage	logo des System	A
changeLogo	inputFile	neues Logo hochladen.	A
Passwort Schema	InputText	Änderung des Passwortschema.	A
Passwortregeln	inputText	Änderung des Passwortregeln.	A
kontakt	inputText	Änderung der Email Adresse.	A
abbrechen	commandButton	Abbruch des Änderungsvorgangs.	A
Speichern	commandButton	Speichern der Änderungen.	A

4.13 *Benutzerverwaltung*

Benutzerverwaltung.xhtml		
Element	Typ	Beschreibung
Benutzer	selectManyMenu	Je nach Benutzerrolle wird die entsprechende Benutzergruppe angezeigt.
UserList	datatable	Tabelle mit Paginierung
Benutzer anlegen	link	link zum registrierungsseite
UserList	Pagination	Liste aller Benutzer

4.14 Circulationlist

Circulationlist.xhtml			
Element	Typ	Beschreibung	Recht
Fakultät	selectOneMenu	Wenn man nach bestimmten Fakultäten filtert, werden die entsprechenden Umläufe angezeigt.	alle
Umlaufliste	dataTable	Tabelle mit Paginierung.	alle
Umlauf	link	link zur Umlaufdetails	alle
CirculationList	Pagination	list aller Umläufe	alle

4.15 Umlaufferstellung-Seite

createcirculation.xhtml			
Element	Typ	Beschreibung	Recht
Titel	inputText	Textfeld für Titelname.	A-Ak-I
Doktorand	inputText	Textfeld für Doktorandsname.	A-Ak-I
Doktorvater	inputText	Textfeld für Doktorvatersname.	A-Ak-I
kurzbeschreibung	textarea	Textfeld für kurze Beschreibung über die Dissertation.	A-Ak-I
Start Termin	inputText	Textfeld für das Startdatum	A-Ak-I
Ablauf Frist	inputText	Textfeld für das Enddatum	A-Ak-I
verpflichtend	checkbox	Angabe ob die Teilnahme verpflichtend ist	A-Ak-I
logoutButton	commandButton	Loggt den Nutzer aus dem System	A-Ak-I
pfdUpload	inputFile	Dateien hochladen.	A-Ak-I
create	commandbutton	um den Umlauf zu speichern	A-Ak-I

4.16 circulationDetails.xhtml

Circulationdetails.xhtml						
Element	Typ	Beschreibung	Rechte	ReadOnly	editable	visib
Titel	outputText	Titel der Dissertation	alle	P	A-Ak-D	alle
Doktorand	outputText	Neme der Doktorand	alle	P	A-Ak-D	alle
Doktorvater	outputText	Neme der Doktorvater	alle	P	A-Ak-D	alle
kurzbeschreibung	outputText	kurze Beschreibung über die Dissertation	alle	P	A-Ak-D	alle
Starttermin	outputText	Starttermin	alle	P	A-Ak-D	alle
Ablauftermin	outputText	Ende der First für die Begutachtung	alle	P	A-Ak-D	alle
pdf Herunterladen	commandButton	Download der Dissertation und weitere Unterlagen	alle	-	-	alle
Löschen	commandbutton	Umlauf löschen	A,D,PK	P	A-Ak-D	A-Ak
revote	commandbutton	Umlauf erneut abstimmen	A,D,PK	P	A-Ak-D	A-Ak
pdfUpload	inputFile	un Datein hochzuladen	A,D,PK	P	A-Ak-D	A-Ak

4.17 Hilfe

Hilfe.xhtml		
Element	Typ	Beschreibung

5 Systemfunktionen

Johannes Silvennoinen

5.1 Systemstart

Der Start des Systems umfasst mehrere Schritte:

1. Starten von Tomcat: Tomcat ist ein Webserver und Servlet-Container, der das System ausführt. Der Server wird durch Ausführen der Start-Skripte gestartet.
2. Weblistener: Der Weblistener ist eine Komponente des Tomcat-Servers, die bestimmte Ereignisse im Lebenszyklus des Servers verfolgt. Beim Start des Systems wird der Weblistener verwendet, um die Verbindung zur Datenbank initialisieren, die Konfiguration zu lesen und den Logger zu starten durch die methode `contextInitalized()`.
3. Logger initialisieren: Als erstes wird der Logger initialisiert, um sicherzustellen, dass wir Log-Meldungen ausgeben können, falls bei dem Systemstart Probleme auftreten. Der Logger wird über die Methode `init()` initialisiert.
4. Konfiguration lesen: Das System muss beim Start die Konfigurationsdatei `config.properties` lesen, um die Einstellungen und Parameter zu erhalten, die für das System erforderlich sind. Die Konfigurationsdatei wird als Java-Properties gespeichert und enthalten Informationen wie Datenbankverbindungsparameter, Sicherheitseinstellungen, Zugriffsrechte und andere Systemeinstellungen. Dies erfolgt durch die methode `loadConfig()`.
5. Verbindung zur Datenbank: Beim Start des Systems muss eine Verbindung zur Datenbank hergestellt werden. Hierfür wird der Connection Pool verwendet. Der Connection Pool ist als Singleton implementiert und threadsicher, so dass mehrere Anfragen gleichzeitig verarbeitet werden können. Für die initialisierung der Verbindungen wird die Methode `initalizeConnections()` aufgerufen.
6. Wartungsthread: Beim Start des Systems wird auch ein Wartungsthread gestartet, der die Systemwartung und -überwachung durchführt. Die Methode `startMaintenanceThread()` wird hierfür aufgerufen.

5.2 Systemstop

Der Stop des Systems umfasst auch mehrere Schritte:

1. Weblistener: Beim Herunterfahren des Systems wird der Weblistener verwendet, um die Verbindung zur Datenbank zu schließen, den Logger zu stoppen und den Wartungsthread zu beenden. Dies erfolgt durch die Methode `contextDestroyed()`.
2. Wartungsthread: Beim Herunterfahren des Systems wird dieser Thread gestoppt. Die Methode `stopMaintenanceThread()` wird hierfür aufgerufen.
3. Datenbankverbindung: Beim Stoppen des Systems wird die Verbindung zur Datenbank geschlossen. Der Connection Pool wird verwendet, um sicherzustellen, dass alle offenen Verbindungen geschlossen werden. Dafür ist die methode `closeConnections()` zuständig.
4. Logger: Beim Herunterfahren des Systems muss auch der Logger gestoppt werden, um sicherzustellen, dass alle ausstehenden Log-Einträge geschrieben werden. Hierfür wird die Methode `stopLogger()` aufgerufen.
5. Tomcat-Server herunterfahren: Schließlich muss der Tomcat-Server heruntergefahren werden, um das System vollständig zu beenden.

5.3 Wartungsthread

Das System verfügt über einen Wartungsthread, der zuständig ist um erinnerungs E-Mails zu Schicken an Prüfungsberechtigte die noch nicht abgestimmt haben, 1 tag vor der Ablauffrist.

5.4 Zugriffsrechte

Das System ist mit verschiedenen Benutzerrollen ausgestattet. Jede Rolle hat unterschiedliche Zugriffsrechte auf die Funktionen des Systems. Administratoren haben die höchsten Zugriffsrechte und können alle Funktionen des Systems nutzen. Andere Benutzerrollen haben eingeschränkte Zugriffsrechte. Das System verwendet eine Kombination von Techniken, um sicherzustellen dass nur autorisierte Benutzer auf das System zugreifen können. Zu diesen Techniken gehören Trespass-Listener sowie die Überprüfung von Zugangsrechten in Backing-Beans.

Der Trespass-Listener überwacht die Anforderungen an das System und überprüft, ob der Benutzer über die erforderlichen Berechtigungen verfügt, um auf die angeforderte Ressource zugreifen. Wenn der Benutzer nicht über die erforderlichen Berechtigungen verfügt, wird eine Zugriffsverletzung ausgelöst.

Darüber hinaus werden die Zugriffsrechte auch in Backing-Beans überprüft. Backing-Beans enthalten den Business-Code des Systems und stellen eine Verbindung zwischen den xhtml Seiten und der Datenzugangs-Schicht her. Bevor ein Benutzer auf eine Ressource zugreifen kann, wird in den Backing-Beans überprüft, ob der Benutzer über die erforderlichen Berechtigungen verfügt. Wenn der Benutzer nicht über die erforderlichen Berechtigungen verfügt, wird ebenso eine Zugriffsverletzung ausgelöst.

Durch die Kombination dieser Techniken wird sichergestellt, dass nur autorisierte Benutzer auf das System zugreifen können und dass die Integrität und Sicherheit des Systems gewahrt bleibt.

5.5 Logging

Das System nutzt die Java Logging API um eine umfassende Protokollierung aller Aktivitäten durch zu führen. Dabei werden sämtliche Aktionen, die von Benutzern und dem System ausgeführt werden, erfasst. Die Protokolldateien werden auf dem Server gespeichert und sind nur für berechtigte Personen einsehbar. Das Logging dient der Überwachung der Systemaktivitäten und der Fehlerbehebung. Die Loglevels sind auf folgende 4 Level aufgeteilt:

- Severe
- Warning
- Info
- Debug

5.6 Sicherheitsaspekte

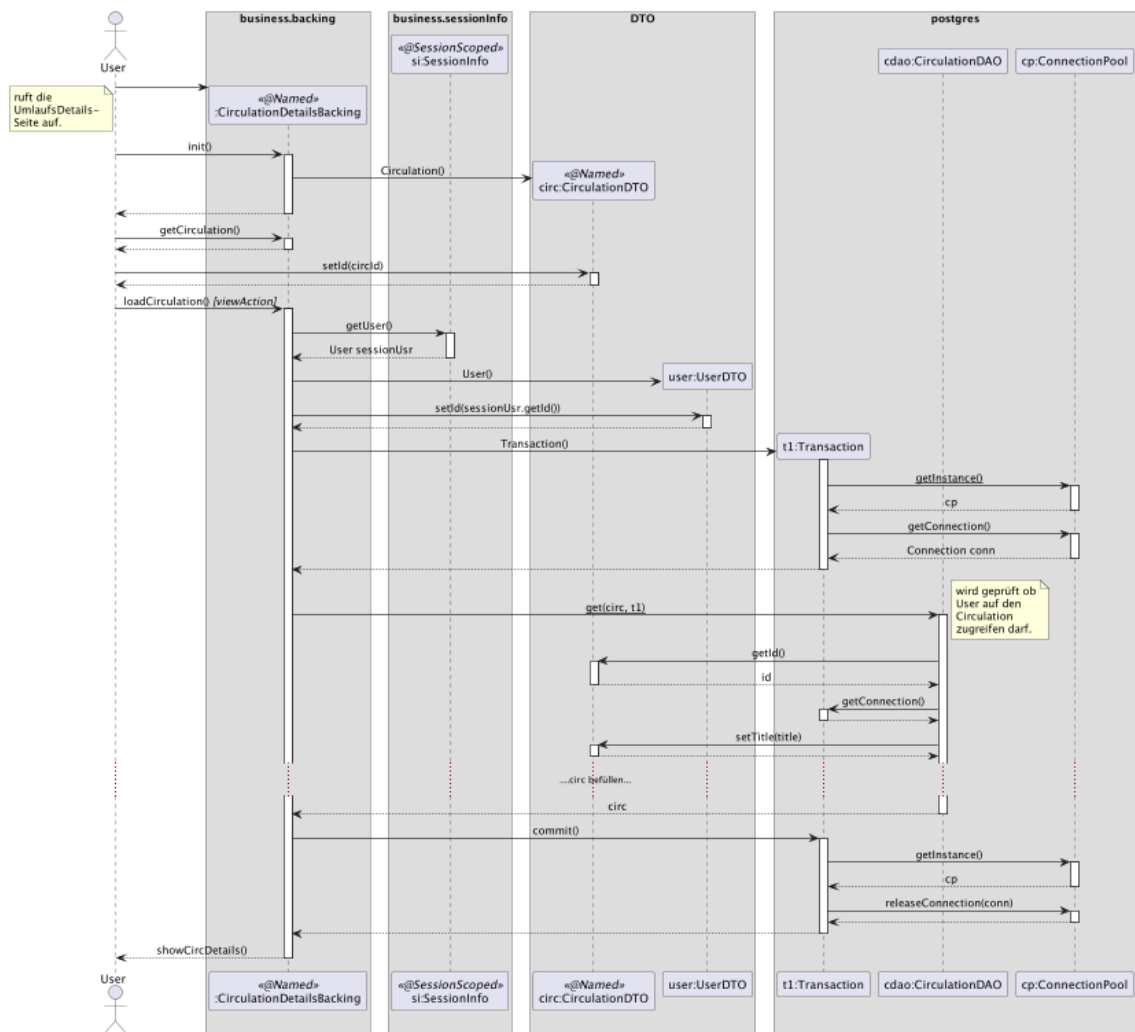
Das System ist gegen Angriffe von außen abgesichert. Der Datenverkehr zwischen dem Server und dem Client erfolgt über eine SSL-verschlüsselte Verbindung, um die Übertragung von Daten abzusichern. Die Verbindung zur Datenbank ist ebenfalls über SSL gesichert, um die Sicherheit der Datenübertragung zu gewährleisten. Das System verfügt über eine sinnvolle Fehlerbehandlung bei fehlerhaften Eingaben, um mögliche Schwachstellen im System zu minimieren.

6 Datenfluss

Alaa Qasem

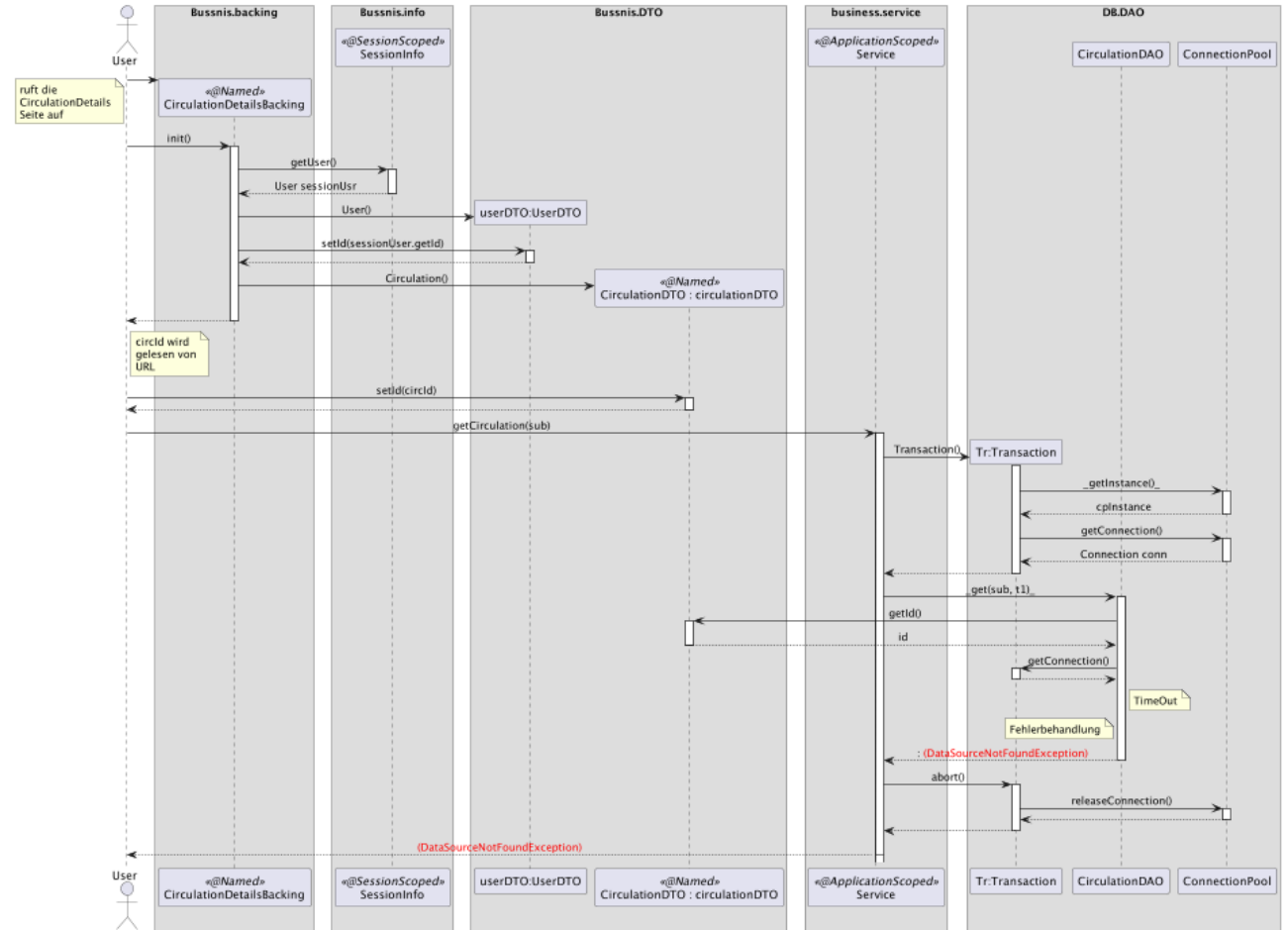
6.1 Anzeigen eines Umlaufs

Ein angemeldeter Nutzer ruft eine UmlaufsDetails-Seite.



6.2 Anzeigen eines Umlaufs (Fehlerfall)

ein Timeout Fall



7 ER-Modell

Stefan Witka

Das in [8] gezeigte Diagramm beschreibt die Datenstruktur von DocThesisTracker. Unterstrichene Attribute sind Primärschlüssel, doppelt umrandete Entities bezeichnen schwache Entity typen.

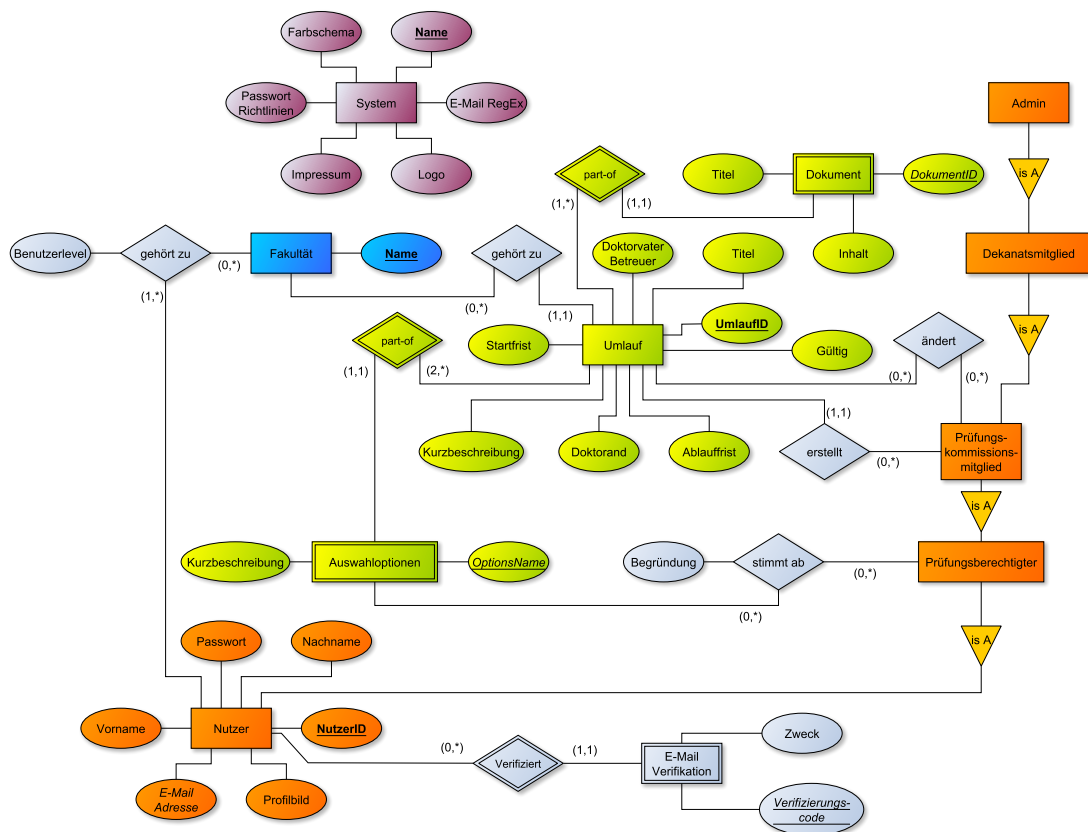


Abbildung 8: Das ER Diagramm zu DocThesisTracker

7.1 Entities

Die Entities aus dem ER-Diagramm [8] werden im folgenden im Detail beschrieben.

Nutzer Personen die das System benutzen. Benutzerlevel der Relation „Gehört zu“ Fakultät bestimmt ob jemand Dekanatsmitglied, Prüfungskommissionsmitglied, oder Prüfungsberechtigter ist. Admins haben alle Rechte an allen Fakultäten.

$$\begin{aligned} \text{Admin} &\xrightarrow{isA} \text{Dekanatsmitglied} \\ &\xrightarrow{isA} \text{Prüfungskommissionsmitglied} \\ &\xrightarrow{isA} \text{Prüfungsberechtigter} \\ &\xrightarrow{isA} \text{Nutzer} \end{aligned}$$

Fakultät Fakultät einer Universität. Nutzer gehören zu mindestens einer Fakultät mit unterschiedlichen Berechtigungsleveln.

Umläufe gehören zu genau einer Fakultät Durch die Beziehungen Zur Fakultät wird, unter anderem, festgelegt, ob ein Nutzer einen Umlauf sehen soll.

Umlauf Bereitgestellte Informationen zu einer Dissertation für Bewertung durch Prüfungsberechtigte.

Dokument Schwacher Entity Typ. Teil eines Umlaufs. Wird durch ID des Umlaufs identifiziert.

Die Dokumente welche einem Umlauf hinzugefügt werden..

Auswahloption Schwacher Entity Typ. Teil eines Umlaufs. Wird durch ID des Umlaufs identifiziert.

Die änderbaren Auswahlmöglichkeiten für umläufe, mit einem boolean ob eine Kurzbegründung für die Option verfügbar sein sollte.

System Daten die zur Systemverwaltung gehören.

E-Mail Verifikation Tokens für Registrierung und Passwortrücksetzung.
Wird eindeutig einem Nutzer zugeordnet.