

Dokumentation JProject

Entwurf und Implementierung einer Projektverwaltung mittels JEE 5 Webtechnologien

Team:

Andy Klay

Michael Koppen

Tino Reuschel



Inhaltsverzeichnis

1. Einführung.....	4
2. Portal-Funktionen.....	5
2.1 Funktionale Sicht.....	5
2.2 Funktionen Detailsicht.....	7
3. Architektur.....	11
3.1 Schichten.....	11
3.2 Komponenten.....	13
3.2.1 GUI	14
3.2.2 Controller	14
3.2.3 Manager	14
3.2.4 Model	15
3.2.5 DA	16
3.2.6 DAO	16
3.2.7 Datenbank	16
4. Systemvoraussetzungen.....	18
4.1 Allgemeine Voraussetzungen.....	18
5. Anhang.....	19
A1 Datenbank ERR-Diagramm.....	20
A2 Sequenz-Diagramm.....	21
A3 Package-Diagramm.....	22
A4 Klassendiagrammübersicht	23



Dokument-Historie

Version	Datum	Autor	Erläuterung
1.0	09.07.2011	Klay, Koppen	



1. Einführung

JProject ist in erster Linie eine allgemeine Projektverwaltungsumgebung die von jedem Rechner, der Internet hat, erreicht werden kann.

JProject wird in 3 Teile unterteilt: JProject, Content und Wiki.

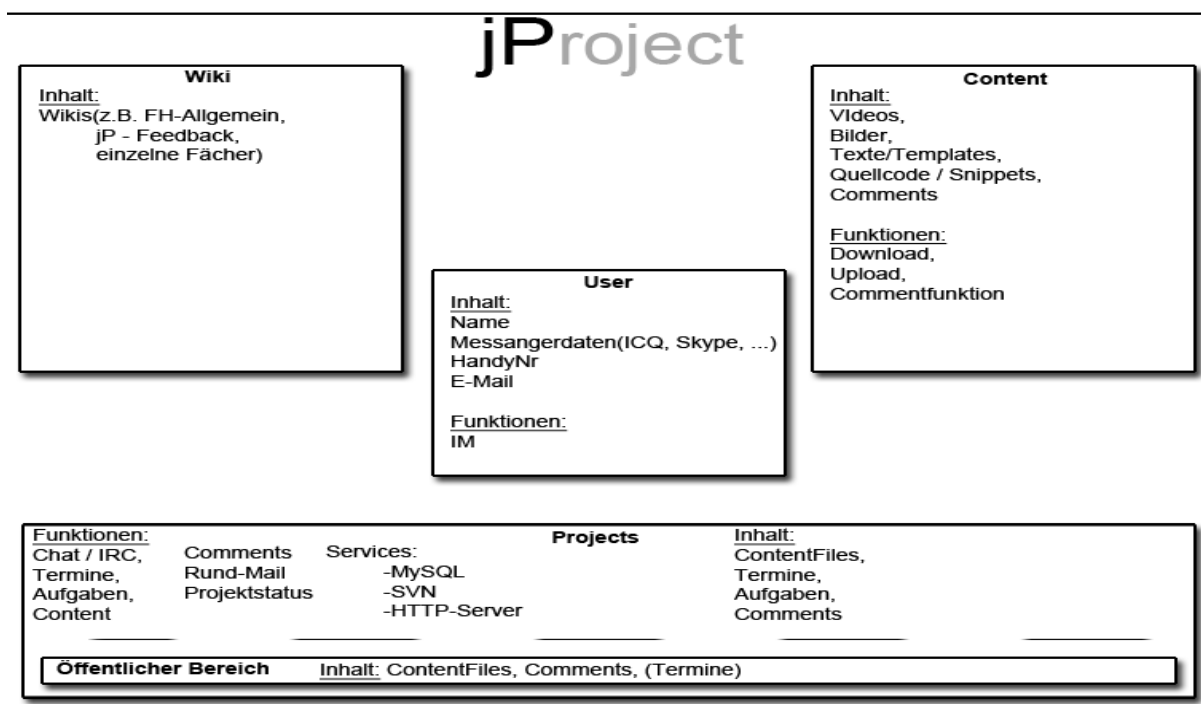
Mit JProject kann eine Benutzergruppe Teams zur Realisierung eines Projektes bilden und sich selbst organisieren. Die User können Projekte erstellen, Dokumente/Sourcelcode hochladen und Aufgaben anlegen und zuordnen. Dazu steht ein flexibles Rechtesystem zur Seite.

Zum späteren Funktionsumfang soll eine umfangreiche Datenbank mit Hilfestellungen, Anleitungen bzw. Tutorials zur Verfügung stehen. Dieser Bereich nennt sich Content.

Des Weiteren soll später ein Bereich namens Wiki eingerichtet werden, in dem die User allgemeine Informationen zum Ablauf und zur Organisation, in deren Hierarchie das Projekt entworfen werden soll, finden.

Funktionaler Umfang der Idee:

- Rechtesystem für die Projekte
- Öffentlicher Bereich für Projekte
- Download/Upload des Contents (später mit Versionskontrolle)
- Synchrones und Verlustfreies arbeiten am Projekt
- Termin- und Aufgabenverwaltung innerhalb der Projekte



(Abb. 1 Ursprünglicher Ideen-Entwurf)



2. Portal-Funktionen

2.1 Funktionale Sicht/User-Sicht auf das Portal

In dem Usecase-Diagramm (Abb. 2) ist nur eine starre Konstellation der Rollen zu erkennen. Implementiert wurde ein Rollensystem welches erlaubt dynamisch Rollen hinzu zufügen.

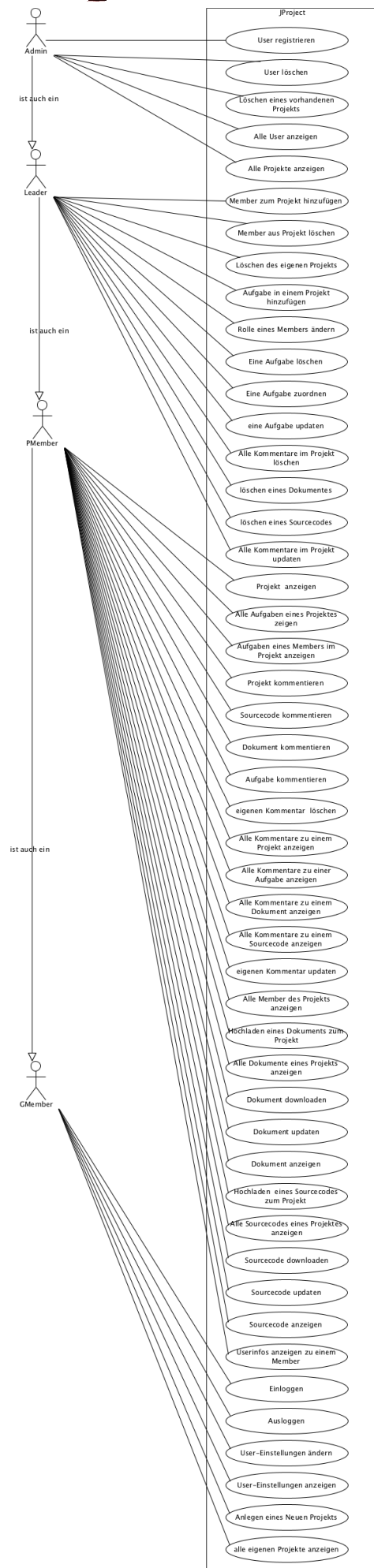
Es gibt ein globales Rollensystem (GlobalRoles). In ihm wird unterschieden welche Aktionen der Benutzer im Allgemeinen tätigen darf. Beispielsweise ist hier im Usecase-Diagramm zu sehen der GMember (User ohne Projekt) und der Admin, dies sind zwei mögliche globale Rollen.

Als Weiteres gibt es aber noch ein Rollensystem nur für Projekte (ProjectRoles). In ihm wird unterschieden welche Aktionen der Benutzer innerhalb eines Projektes tätigen darf. Als Beispiel dafür ist im Usecase-Diagramm der Leader und der PMember zu sehen.

Die globalen Rollen und Projektrollen sind für den Betrieb des Systems zwingend erforderlich d.h. es sind mindestens 2 Rollen global (Admin und Member) und 2 Projektrollen (Leader und Member) erforderlich.

Der Admin hat durch seine globale Rolle eine übergeordnete Funktion und darf grundsätzlich alles was andere Rollen auch dürfen außer Tasks zuordnen, dann kommt der Projekt-Leader, der innerhalb eines Projektes wiederum mehr Rechte hat als ein Projekt-Member. User die in einem Projekt nicht involviert sind, also weder Member noch Leader (GMember) sind und haben daher nur grundlegende Aktionen wie z.B. Einloggen, Ausloggen, Projekt anlegen.

Die Rollen können derzeit von Ersteller des Systems umbenannt und auch deren Rechte dynamisch geändert oder auch komplett neue Rollen hinzugefügt werden.
















(Abb. 2UseCase-Diagramm)
















2.2 Funktionen Detailsicht

Es wird hier im Folgenden nicht mehr immer explizit erwähnt, dass alle Aktionen die der Projekt-Leader machen kann, auch der Admin über die AdminConsole tätigen kann. Ausgeschlossen davon ist das zuordnen von Tasks an PMember.












Zusammenfassung

Name	Dokumentation
 Admin	Admin = d.h. also ein User mit entsprechenden Rechten, muss nicht Admin heißen.
 User löschen	Über die AdminConsole kann ein Admin einen User aus dem System löschen, damit dieser das System nicht mehr nutzen kann.
 Alle Projekte anzeigen	Nur der Admin kann über die AdminConsole alle Projekte anzeigen.
 Alle User anzeigen	Nur der Admin kann über die AdminConsole alle User anzeigen.
 Löschen eines vorhandenen Projekts	Ein Projekt kann direkt gelöscht werden vom Admin über die AdminConsole.
 User registrieren	Über die AdminConsole kann ein Admin einen User im System erstellen, damit dieser das System nutzen kann.
 Leader	Projekt-Leader =d.h. also ein User mit entsprechenden Rechten der in einem Projekt mehr Rechte hat als ein einfacher Teilnehmer(Projekt-Member).
 Eine Aufgabe updaten	Der Projekt-Leader kann über das Projekt und über den Reiter „Task“ eine Aufgabe des Projekts updaten. Durch Auswahl des Tasks links, dem Eintragen der neuen Daten rechts und dem Klicken auf Update-Button unten kann dies bewerkstelligt werden.
 Eine Aufgabe zuordnen	Der Projekt-Leader kann über das Projekt und über den Reiter „Task“ eine Aufgabe des Projekts einem Member zuordnen. Durch die Auswahl der vorhandenen Aufgabe links und Auswahl des Members rechts und klicken auf Zuweisen kann dies bewerkstelligt werden.
 Löschen des eigenen Projekts	Ein Projekt kann gelöscht werden, indem der Projekt-Leader alle Member des Projektes löscht und zuletzt sich selbst.
 Member zum Projekt hinzufügen	Der Projekt-Leader kann über das Projekt einen User als Projekt-Member hinzufügen.
 Löschen eines Sourcecodes	Das Mitglied mit den benötigten Rechten, kann über das Projekt und dann Reiter „Sourcecode“ die Sourcecodes des Projektes anzeigen lassen. Durch Klick auf das Kreuz vor dem Namen des Sourcecodes wird die Sourcecode Datei gelöscht.
 Alle Kommentare im Projekt updaten	Der Projekt-Leader kann alle Kommentare eines Projektes updaten (alle Kommentare zu allen Tasks, allen Dokumenten, allen Sourcecodes und die Kommentare des Projektes selber)durch das Anzeigen eines Kommentars, dem Ändern des
















	Contents im Textfeld und dem Klick auf den Update Button darunter.
 Eine Aufgabe löschen	Der Projekt-Leader kann über das Projekt und über den Reiter „Task“ in der Leiste links eine Aufgabe des Projekts löschen durch klicken auf das Kreuz neben einer Aufgabe.
 Member aus Projekt löschen	Der Projekt-Leader kann über das Projekt einen Projekt-Member aus dem Projekt entfernen.
 Alle Kommentare im Projekt löschen	Der Projekt-Leader kann über das Projekt alle Kommentare zu allen Tasks, allen Dokumenten, allen Sourcecodes und die Kommentare des Projektes selber löschen durch das Anzeigen dieser Kommentare und dem Button Delete darüber.
 Aufgabe in einem Projekt hinzufügen	Der Projekt-Leader kann über das Projekt und über den Reiter „Task“ eine Aufgabe zum Projekt hinzufügen.
 Löschen eines Dokumentes	Das Mitglied mit den benötigten Rechten, kann über das Projekt und dann Reiter „Dokument“ die Dokumente des Projektes anzeigen lassen. Durch Klick auf das Kreuz vor dem Namen des Dokumentes wird das Dokument gelöscht.
 Rolle eines Members ändern	Der Projekt-Leader kann über das Projekt und über den Reiter „Project“ im zweiten Abschnitt die Rolle der vorhandenen Member im Projekt ändern. Des Weiteren kann der Leader seine eigene Rolle im Projekt ändern, jedoch muss min. ein Projekt-Leader vorhanden sein.
 GMember	Grund-User, der kein Projektteilnehmer ist
 Einloggen	Wenn man die Startseite des Systems aufruft so findet man rechts in der Navigationsleiste ein Eingabefeld für den Benutzernamen dort trägt man auch sein Benutzernamen ein, des Weiteren findet man ein Eingabefeld für das Passwort, wo man sein Passwort einträgt und dann klickt man auf „einloggen“.
 Ausloggen	Wenn ein User eingeloggt ist, so findet er rechts in der Navigationsleiste den Button „ausloggen“.
 User-Einstellungen ändern	Wenn ein User eingeloggt ist so findet er links von Button zum ausloggen, ein Link zu seinen User-Einstellungen, dort kann er Änderungen vornehmen und danach speichern.
 User-Einstellungen anzeigen	Wenn ein User eingeloggt ist so findet er links von Button zum ausloggen, ein Link zu seinen User-Einstellungen, wenn er dort drauf klickt so werden ihm seine User-Einstellungen angezeigt.
 Anlegen eines neuen Projekts	Wenn ein User sich gerade eingeloggt hat so kann er auf dieser Seite ein neues Projekt anlegen, indem er rechts den Namen des Projektes einträgt und dann auf „addnew Project“ klickt.
 alle eigenen Projekte anzeigen	Wenn ein User eingeloggt ist, so werden alle seine Projekte auf der linken Seite unter dem Punkt Projekte angezeigt.



 PMember	Ist ein einfacher Projektteilnehmer(Projekt-Member).
 Sourcecode anzeigen	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Sourcecode“ sich alle Sourcecodes anzeigen lassen, wenn er nun auf den Namen eines Dokumentes klickt, so wird ihm das Dokument angezeigt.
 Alle Sourcecodes eines Projektes anzeigen	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Sourcecode“ sich alle Sourcecodes anzeigen lassen.
 Projekt anzeigen	Sobald der User sich ans System eingeloggt hat, werden seine Projekte auf der linken Seite angezeigt.
 Dokument downloaden	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Dokumente“ sich alle Dokumente anzeigen lassen, wenn er nun auf den Namen eines Dokumentes klickt wird ihm das Dokument angezeigt, dort findet er dann auch den Download Button.
 Aufgabe kommentieren	Wenn ein Mitglied eines Projektes sich eine Aufgabe anzeigen lässt, so findet er am unteren Ende die Kommentare und kann selber ein Kommentar schreiben.
 Sourcecode downloaden	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Sourcecode“ sich alle Sourcecodes anzeigen lassen, wenn er nun auf den Namen eines Sourcecodes klickt wird ihm der Sourcecode angezeigt, dort findet er dann auch den Download Button.
 Alle Kommentare zu einem Dokument anzeigen	Wenn ein Mitglied eines Projektes sich ein Dokument anzeigen lässt, so findet er am unteren Ende alle Kommentare, wenn man auf „show Comments“ klickt.
 Sourcecode updaten	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Sourcecode“ sich alle Sourcecodes anzeigen lassen, wenn er nun auf den Namen eines Sourcecodes klickt wird ihm der Sourcecode angezeigt, dort findet er dann auch den Update Button.
 Dokument kommentieren	Wenn ein Mitglied eines Projektes sich ein Dokument anzeigen lässt, so findet er am unteren Ende die Kommentare und kann selber ein Kommentar schreiben.
 Dokument anzeigen	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Dokumente“ sich alle Dokumente anzeigen lassen, wenn er nun auf den Namen eines Dokumentes klickt, so wird ihm das Dokument angezeigt.
 Dokument updaten	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Dokumente“ sich alle Dokumente anzeigen lassen, wenn er nun auf den Namen eines Dokumentes klickt wird ihm das Dokument angezeigt, dort findet er dann auch den Update Button.
 Aufgaben eines Members im Projekt	Ein Member des Projektes sieht auf der Projekt-Seite

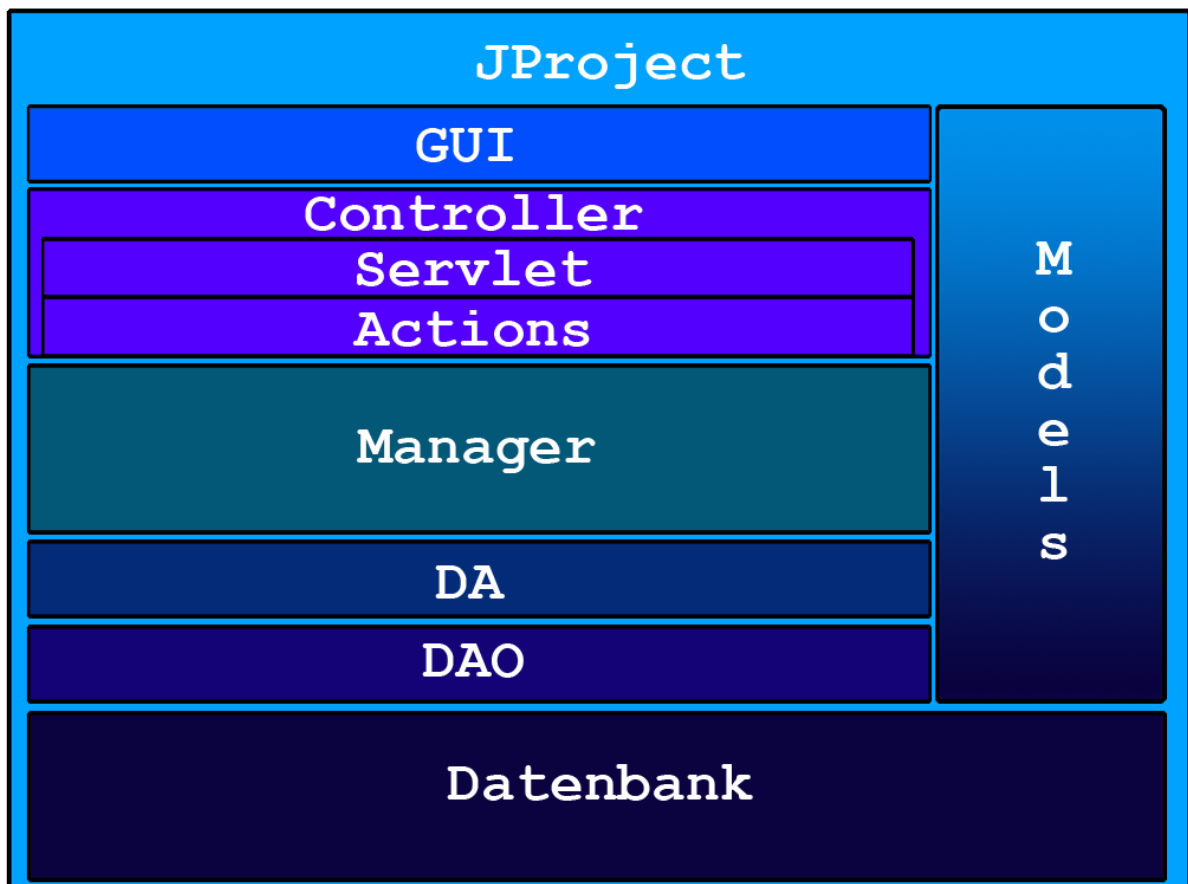


anzeigen	durch das Anklicken eines Members im Member-Abschnitt unten dann auch dessen zugewiesenden Aufgaben.
 Alle Kommentare zu einem Sourcecode anzeigen	Wenn ein Mitglied eines Projektes sich ein Sourcecode anzeigen lässt, so findet er am unteren Ende alle Kommentare, wenn man auf „show Comments“ klickt.
 Sourcecode kommentieren	Wenn ein Mitglied eines Projektes sich ein Sourcecode anzeigen lässt, so findet er am unteren Ende die Kommentare und kann selber ein Kommentar schreiben.
 Alle Member des Projekts anzeigen	Der Projekt-Leader sieht wenn er auf das Projekt klickt alle Member des Projektes.
 Alle Dokumente eines Projekts anzeigen	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Dokumente“ sich alle Dokumente anzeigen lassen.
 Hochladen eines Sourcecodes zum Projekt	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Sourcecode“ ein Sourcecode hochladen, dazu muss er dann auf „Durchsuche“ klicken und sucht sich die Datei raus die man hochladen möchte.
 Alle Aufgaben eines Projektes zeigen	Jedes Mitglied eines Projektes kann über den Reiter „Task“ sich alle Aufgaben eines Projektes anzeigen lassen.
 eigenen Kommentar löschen	Wenn ein Mitglied eines Projektes ein Kommentar geschrieben hat, hat er die Möglichkeit diesen Kommentar auch wieder zu löschen, indem er sich den Kommentar anzeigt und auf das dazugehörige Kreuz klickt.
 eigenen Kommentar updaten	Der Member eines Projektes kann einen eigenen Kommentar updaten durch das Anzeigen eines Kommentars, dem Ändern des Contents im Textfeld und dem Klick auf den Update Button darunter.
 Projekt kommentieren	Klickt ein User auf eines seiner Projekte, so wird ihm das Projekt angezeigt und am unteren Ende findet er alle Kommentare zum Projekt und kann dort auf ein eigenen Kommentar schreiben
 Alle Kommentare zu einer Aufgabe anzeigen	Wenn ein Mitglied eines Projektes sich eine Aufgabe anzeigen lässt, so findet er am unteren Ende alle Kommentare, wenn man auf „show Comments“ klickt.
 Hochladen eines Dokuments zum Projekt	Ein Mitglied eines Projektes kann über Projekt und den Reiter „Dokumente“ ein Dokument hochladen, dazu muss er dann auf „Durchsuche“ klicken und sucht sich die Datei raus die man hochladen möchte.
 Userinfos anzeigen zu einem Member	Wenn man auf den Namen eines Member klickt, so werden Informationen zu diesem Member angezeigt.
 Alle Kommentare zu einem Projekt anzeigen	Wenn ein User auf eines seiner Projekte klickt, so wird ihm das Projekt angezeigt und am unteren Ende kann er über „show Comments“ sich alle Kommentare anzeigen lassen.

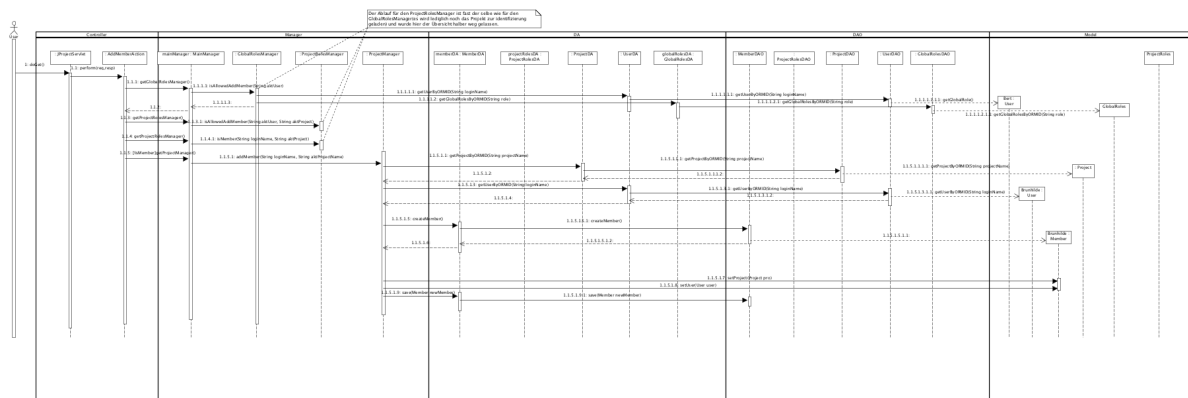


3. Architektur

3.1 Schichten



(Abb.3 Schichten-Übersicht)



(Abb. 4 Sequenz-Diagramm)



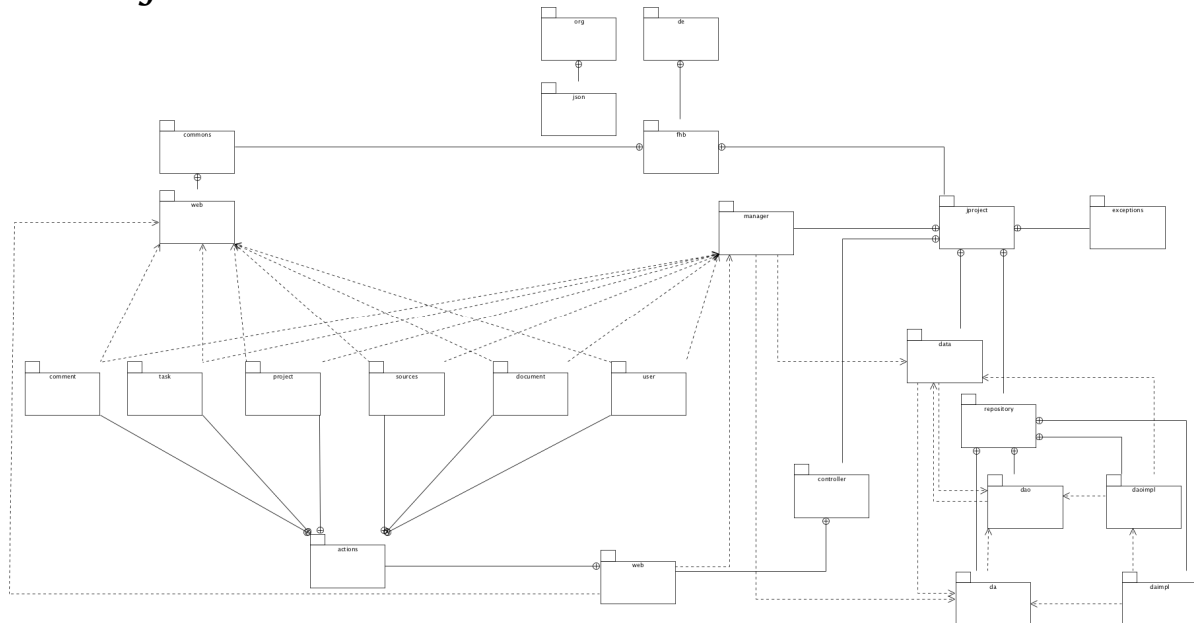
Beschreibung des Anwendungsfalls mit Sequenzdiagramms

Name	AddMemberAction
Kurzbeschreibung	Dieses Sequenz-Diagramm (Abb. 2) stellt dar, wie der ein Projekt-Leader (Bert: User bzw. Bert: Member) einen Member (Brunhilde : User bzw. Brunhilde : Member) zu seinem Projekt hinzufügt.
Akteure	User
Auslöser	Eintragung des Usernamen und Klicken auf Button „Member hinzufügen“
Vorbedingung	User eingeloggt, User hat Projekt indem er Leader ist, der hinzu zu fügende User ist nicht bereits im Projekt
Ergebnis	User wird dem Projekt zugeordnet (Member wurde erstellt)
Nachbedingung	keine
Essentieller Ablauf	<p>Ablauf:</p> <ul style="list-style-type: none"> • 1. doGet() <ul style="list-style-type: none"> - Der Request wird an das JProjectServlet gesendet. - Es überprüft ob der User einer Session zugeordnet ist. - Es ordnet dem Request die Aktion „AddMemberAction“ zu. • 1.1 perform(req, resp) <ul style="list-style-type: none"> - Die Action zieht aus dem Request alle erforderlichen Parameter heraus. - Die Action AddMemberAction überprüft ob der User eingeloggt ist. - 1.1.1 getGlobalRolesManager() und folgende - Die Action überprüft, ob der User Globale-Rechte hat um diese Action auszuführen. - 1.1.3 getProjectRolesManager() und folgende - Die Action überprüft, ob der User Projekt-Rechte hat um die Action auszuführen, sofern er nicht schon Globale Rechte hatte. - 1.1.4 getProjectRolesManager() und folgende - Die Action überprüft, ob der angegebene User eventuell schon Member in diesem Projekt ist. - 1.1.5 getProjectManager() Sofern der User <u>nicht</u> Member in diesem Projekt ist, wird der Member hinzugefügt (1.1.5.1) - 1.1.5.1.1 getProjectByORMID(String projectName) und folgende - Das Projekt wird geladen. - 1.1.5.1.3 getUserByORMID(String loginName) und folgende - Der hinzu zu fügende User wird geladen. - 1.1.5.1.5 createMember() und folgende - Ein neuer Member wird erzeugt. - 1.1.5.1.7 setProject(Project pro) – Das Attribut „Projekt“ (PK) im neuen Member wird gesetzt. - 1.1.5.1.8 setUser(User user) – Das Attribut „User“ (PK) im neuen Member wird gesetzt. - 1.1.5.1.9 save(Member newMember) und folgende – Der neue Member wird in die Datenbank gespeichert.



3.2 Komponenten

Paketdiagramm






(Abb. 5Paket-Diagramm)

Zusammenfassung

Name	Dokumentation
org	Sprachraum/Organisationform
de	Sprachraum/Organisationform
json	beinhaltet Klassen für die JSON-Funktionalität
commons	beinhaltet allgemeine Klassen für immer wiederkehrende Aufgaben
fhh	Organisation
web	Allgemeine Klassen bezogen auf das Web
jproject	Projektpackage - beinhaltet alles Projektspezifisches
exceptions	beinhaltet eigene Exceptions für JProject
manager	beinhaltet alle Manager die für JProject benötigt werden
data	beinhaltet alle Models/VO's für JProject
comment	beinhaltet alle Actions die sich auf Kommentare beziehen
task	beinhaltet alle Actions die sich auf Tasks beziehen
project	beinhaltet alle Actions die sich auf Projekte beziehen
sources	beinhaltet alle Actions die sich auf Sourcecode beziehen



 document	beinhaltet alle Actions die sich auf Dokumente beziehen
 user	beinhaltet alle Actions die sich auf den User beziehen
 repository	beinhaltet Zugriffsschichten zur Datenbank
 controller	beinhaltet Controller für JProject
 dao	beinhaltet alle Interfaces und Factory für die DAO-Schicht(Datenbankzugriff), bietet die Funktionalität für Standardzugriffe auf die Datenbank
 daoimpl	beinhaltet alle Implementationen und Factory für die DAO-Schicht(Datenbankzugriff), bietet die Funktionalität für Standardzugriffe auf die Datenbank
 actions	beinhaltet alle Actions für JProject
 web	beinhaltet alle Controller/Servlets die für das Web nötig sind
 da	beinhaltet alle Interfaces und Factory für die DA-Schicht(Datenbankzugriff), bietet die Funktionalität für erweiterte Zugriffe auf die Datenbank
 daimpl	beinhaltet alle Implementationen und Factory für die DA-Schicht(Datenbankzugriff). bietet die Funktionalität für erweiterte Zugriffe auf die Datenbank

3.2.1 GUI

Das GUI stellt die Schnittstelle für den User zum System. Das JProjectServlet und AdminServlet leiten alle Anfragen an die „index.jsp“. Diese zieht aus dem Response alle nötigen Informationen um zu wissen welche Informationen dargestellt werden sollen.

3.2.2 Controller

Die Controllerschicht mit Servlets und Actions regelt den Zugriff auf das System und regelt was gemacht wird bei welchem Zugriff und greift dann auf die Manager zu.
(weitere Details siehe Klassendiagramm im Anhang A4)

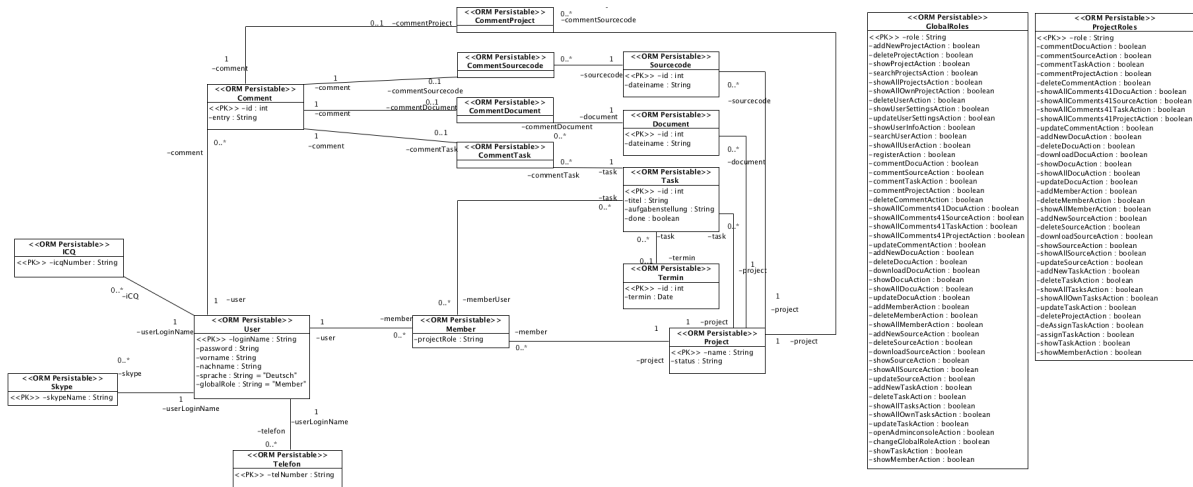
3.2.3. Manager

Die Manager regeln wie Zugriffe im System ablaufen und greifen über die DA-Sicht mit Operationen und Zugriffe auf das Model zu.
(weitere Details siehe Klassendiagramm im Anhang A4)















3.2.4. Model

Domänenmodell



(Abb.6 Domänenmodell)

Zusammenfassung

Name	Dokumentation
 CommentProject	spezieller Kommentar zu einem Projekt
 GlobalRoles	Eine GlobalRole ist eine Rolle, die ein User im System einnehmen kann. D.h. welche Aktionen dieser tätigen kann und darf bzw. nicht darf.
 ProjectRoles	Eine ProjectRole ist eine projektspezifische Rolle, die ein User des Systems innerhalb eines Projekts spielen kann. D.h. welche Aktionen dieser tätigen kann und darf bzw. nicht darf in einem Projekt.
 Sourcecode	Ein Sourcecode ist ein in der Informatik, für Menschen lesbare, in einer Programmiersprache geschriebene Text eines Computerprogramms.
 CommentSourcecode	spezieller Kommentar zu einem Sourcecode
 Comment	Kommentar zu Sourcecode/Document/Project/Task. Beinhaltet spezielle Kommentare.
 CommentDocument	spezieller Kommentar zu einem Dokument
 Document	Ein Document ist eine vom Menschen verfasste Datei mit Informationen, dessen Darstellung unterschiedlich sein kann.
 CommentTask	spezieller Kommentar zu einem Task
 Task	Ein Task ist eine Aufgabe die es in einem Projekt zu erledigen gilt.
 ICQ	ICQ stellt eine Adresse für einen Messenger dar.
 Termin	Ein Termin ist ein festgelegtes Kalenderdatum.



User	Der User stellt jeglichen Benutzer des Systems dar mit nötigen Daten zum Einloggen und zum Identifizieren.
Member	Member ist Assoziationsklasse zwischen User und Project. Ein Member ist also ein User der in ein Projekt involviert ist.
Project	Ein bis mehrere User können in einem Project involviert sein zum Lösen einer Problemstellung.
Skype	Skype stellt eine Adresse für einen Messenger dar
Telefon	ist eine Telefonnummer

3.2.5 DA

DA bietet für erweiterte Zugriffsfunktionalität auf die Datenbank.

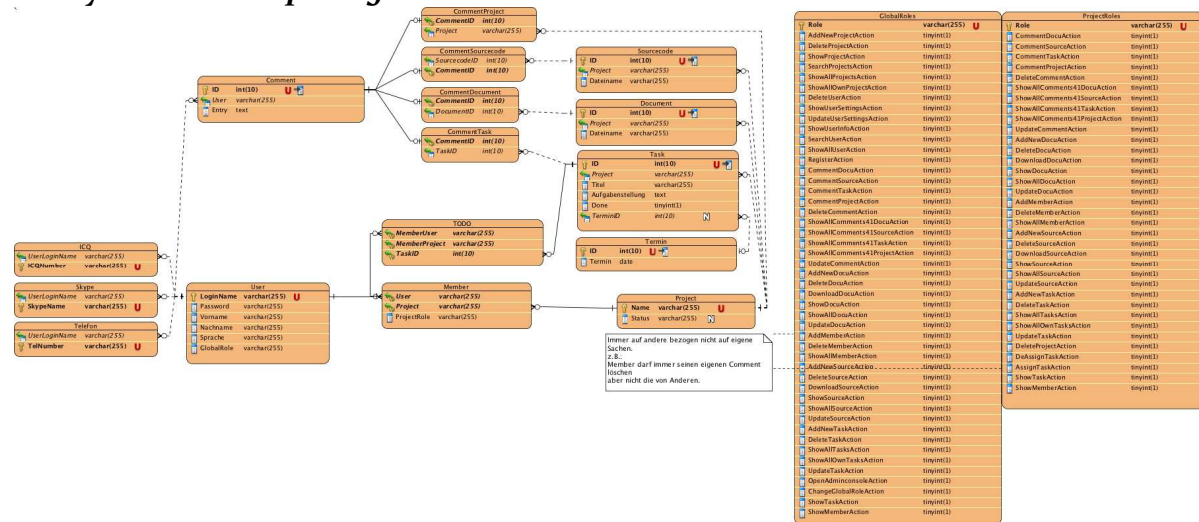
3.2.6 DAO

DAO bietet die Grundfunktionalität für Datenbankzugriffe.

3.2.7 Datenbank

In der Datenbankschicht wird MySQL und Hibernate OR-Mapping genutzt.

Entity-Relationship Diagramm



(Abb.7 Entity-Relationship Diagramm)

Zusammenfassung

Name	Dokumentation
CommentProject	spezieller Kommentar zu einem Projekt
GlobalRoles	Eine GlobalRole ist eine Rolle, die ein User im System einnehmen kann. D.h. welche Aktionen dieser tätigen kann und darf bzw. nicht darf.



 ProjectRoles	Eine ProjectRole ist eine projektspezifische Rolle, die ein User des Systems innerhalb eines Projekts spielen kann. D.h. welche Aktionen dieser tätigen kann und darf bzw. nicht darf in einem Projekt.
 CommentSourcecode	spezieller Kommentar zu einem Sourcecode
 Sourcecode	Ein Sourcecode ist ein in der Informatik, für Menschen lesbare, in einer Programmiersprache geschriebene Text eines Computerprogramms.
 Comment	Kommentar zu Sourcecode/Document/Project/Task. Beinhaltet spezielle Kommentare.
 CommentDocument	spezieller Kommentar zu einem Dokument
 Document	Ein Document ist eine vom Menschen verfasste Datei mit Informationen, deren Darstellung unterschiedlich sein kann.
 CommentTask	Spezieller Kommentar zu einem Task
 Task	Ein Task ist eine Aufgabe die es in einem Projekt zu erledigen gilt.
 TODO	Zuordnung welcher Member welchen Task zu erledigen hat.
 Termin	Ein Termin ist ein festgelegtes Kalenderdatum.
 ICQ	ICQ stellt eine Adresse für einen Messenger dar.
 Skype	Skype stellt eine Adresse für einen Messenger dar
 User	Der User stellt jeglichen Benutzer des Systems dar mit nötigen Daten zum Einloggen und zum Identifizieren.
 Member	Member ist Assoziationsklasse zwischen User und Project. Ein Member ist also ein User der in ein Projekt involviert ist.
 Project	Ein bis mehrere User können in einem Project involviert sein zum Lösen einer Problemstellung.
 Telefon	ist eine Telefonnummer
 Notiz zu den Rollen	Immer auf andere bezogen nicht auf eigene Sachen. z.B.: Member darf immer seinen eigenen Comment löschen aber nicht die von Anderen.



4. Systemvoraussetzungen

4.1 Allgemeine Voraussetzungen

Voraussetzungen für JProject:

Das GUI wurde optimiert für Firefox 4 und 5.

Es wird mindestens Java Version 1.5 benötigt.

Es wird ein Container benötigt z.B. Glassfish oder Tomcat.

Des Weiteren wird eine MySQL Datenbank benötigt – ein entsprechender JDBC-Connector liegt bei.

JProject benutzt folgende Frameworks und APIs:

- Hibernate
- Log4j
- Visual Paradigm JPA
- JSTL
- Apache commons Fileupload & IO
- JavaScript mit Mootools
- MySQL JDBC-Connector



5. Anhang

A1 Datenbank ERR-Diagramm

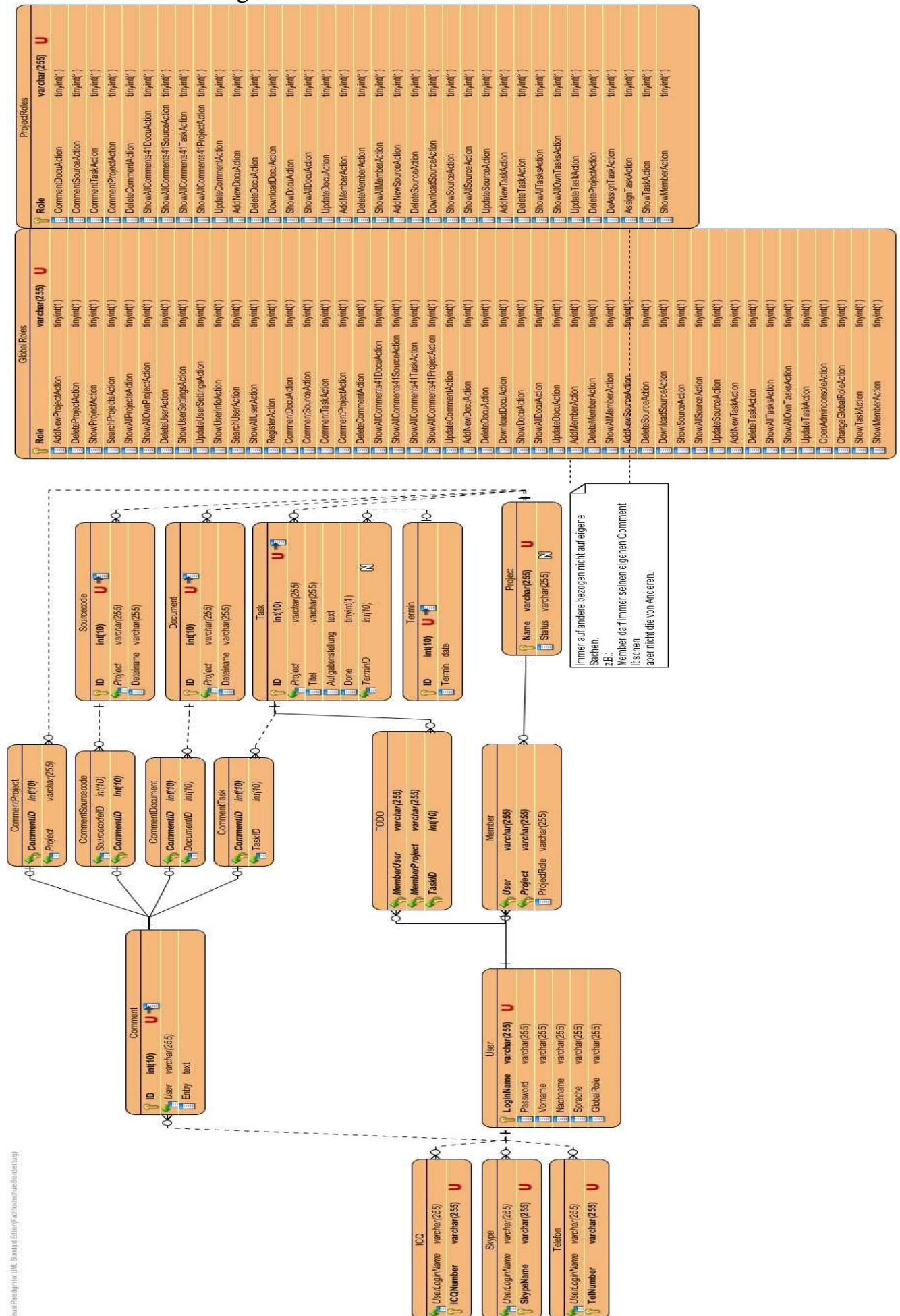
A2 Sequenz-Diagramm(AddMemberAction)

A3 Package-Diagramm

A4 Klassendiagrammübersicht

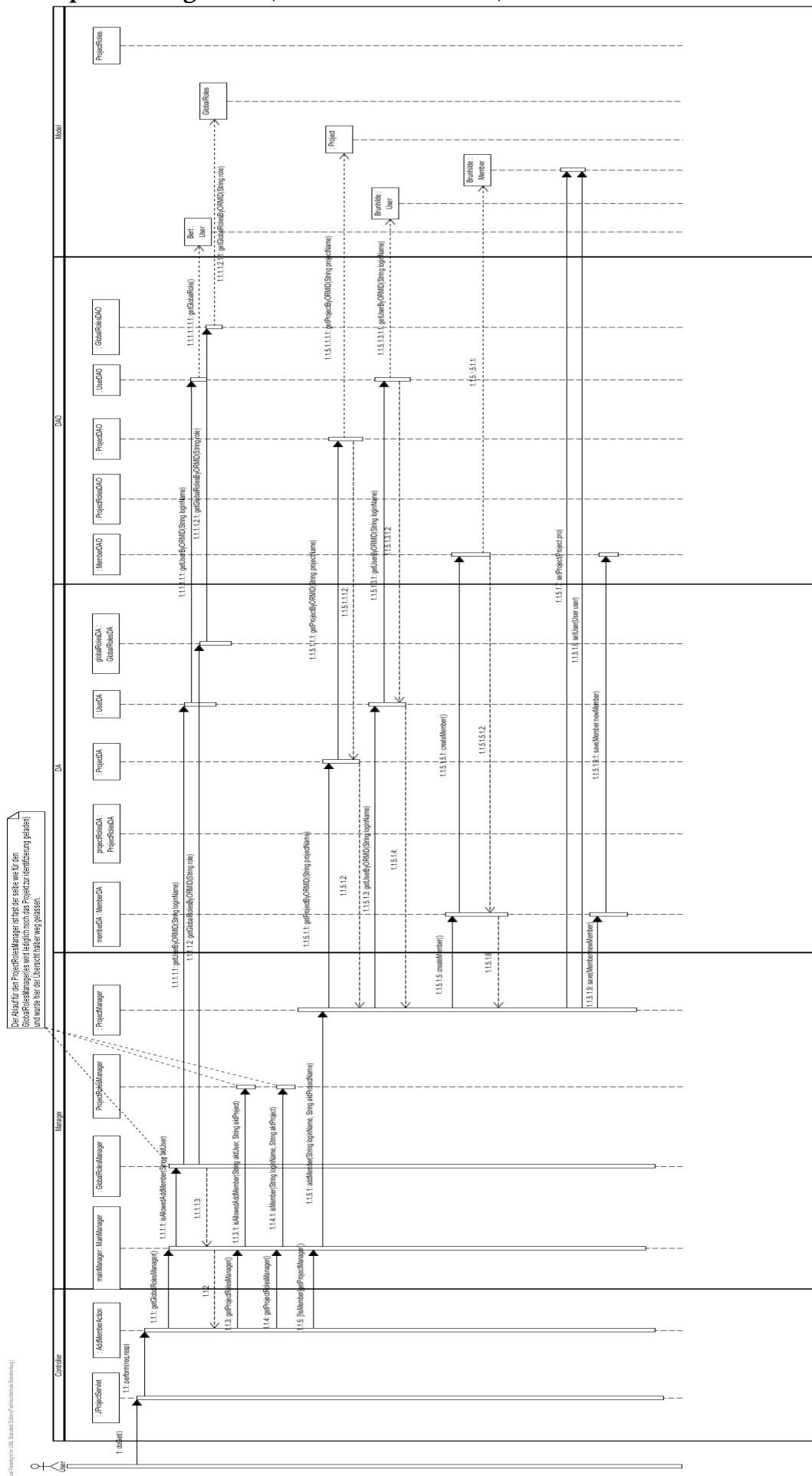


A1 Datenbank ERR-Diagramm



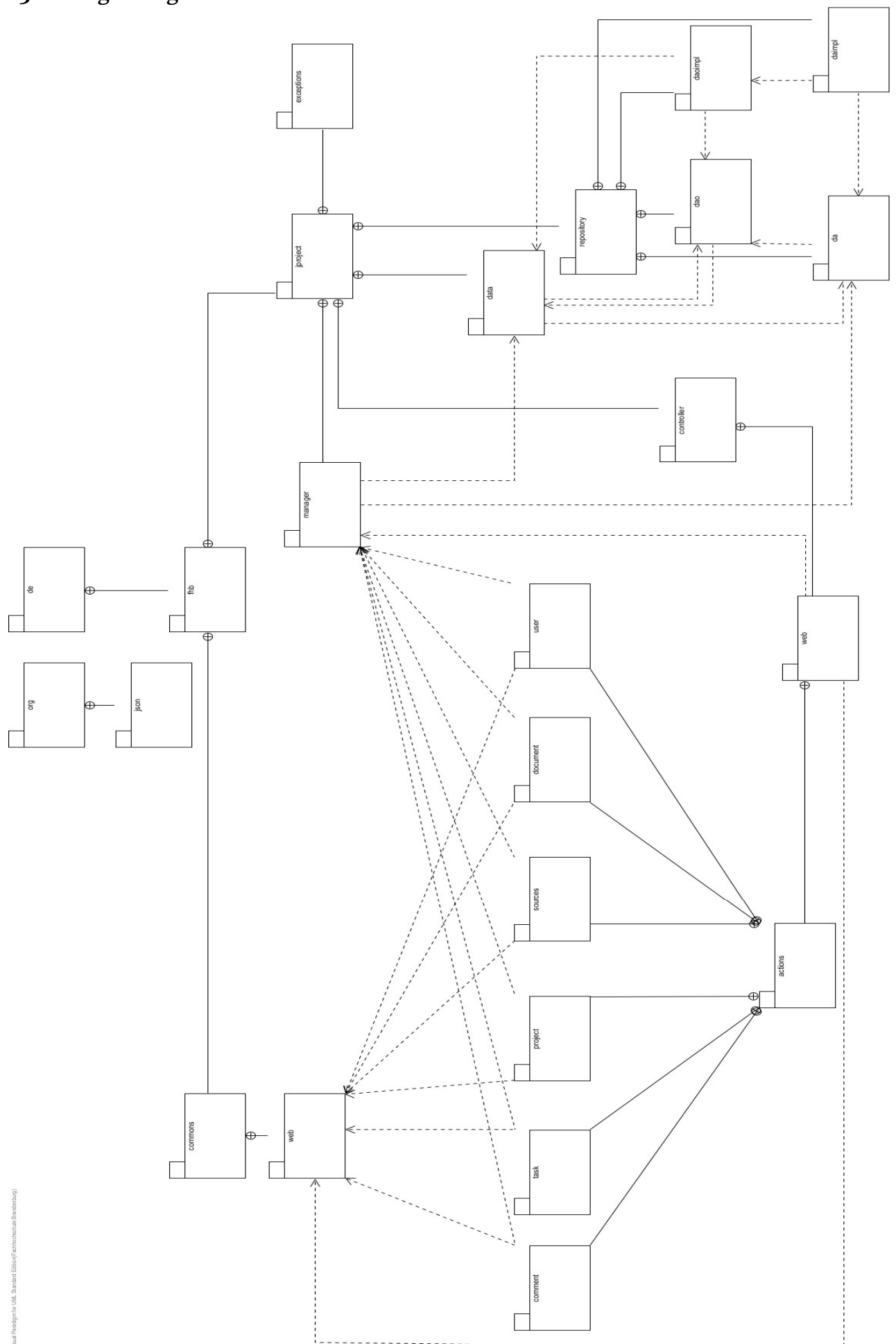


A2 Sequenz-Diagramm(AddMemberAction)





A3 Package-Diagramm

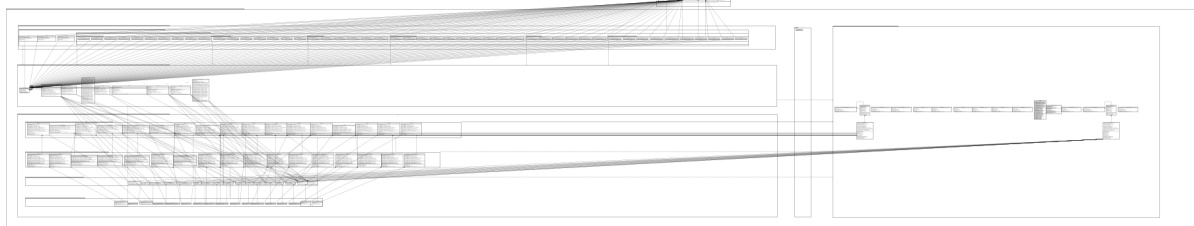


Visual Paradigm for UML, Standard Edition (Fachhochschule Bielefeld)

















A4 Klassen-Diagrammübersicht












Klassendiagramm






Zusammenfassung

Name	Dokumentation
 commons	Beinhaltet allgemeine Klassen für immer wiederkehrende Aufgaben
 web	Allgemeine Klassen bezogen auf das Web.
 HttpServletControllerBase	
 HttpRequestActionBase	Aktion die durch Ableitung definiert werden muss. Die Aktion fuehrt Operationen mit dem Modell aus, und bereitet die Daten fuer die Ausgabe auf. Als letzte Aktion sollte eine Aktion zu einer View den Request weiterleiten, wo dann das Ergebnis der Aktion eingelesen und in eine (HTML-)Seite eingebunden wird.
 CheckString	Diese Klasse ist zum Kontrollieren von Strings.
 HashIt	Diese Klasse generiert einen SHA1- oder MD5-Haswert von einem String.
 jproject	Projektpackage - beinhaltet alles Projektspezifisches.
 controller	Beinhaltet Controller für jProject.
 data	Beinhaltet alle Models/VO's für jProject.
 exceptions	Beinhaltet eigene Exceptions für jProject.
 web	Beinhaltet alle Controller/Servlets die für das Web nötig sind.
 ProjectException	Programmspezifische Exception zur Allgemeinen Fehlerbehandlung
 actions	Beinhaltet alle Actions für jProject.
 JProjectServlet	JProjectServlet ist zum Verwalten der GET und POST Requests. Dieses Servlet is spezialisiert auf alle Anfragen die nicht zu Dateiübermittlungszwecken oder Adminaktionen dient.







 DataServlet	DataServlet ist zum Verwalten von POST Requests. Dieses Servlet ist spezialisiert auf alle Anfragen die speziell die Übermittlung von Daten/JSONObjekte betreffen.
 AdminServlet	AdminServlet ist zum Verwalten der GET und POST Requests. Dieses Servlet ist spezialisiert auf alle Anfragen die speziell den Admin betreffen.
 user	Beinhaltet alle Actions die sich auf den User beziehen.
 task	Beinhaltet alle Actions die sich auf Tasks beziehen.
 sources	Beinhaltet alle Actions die sich auf Sourcecode beziehen.
 project	Beinhaltet alle Actions die sich auf Projekte beziehen.
 document	Beinhaltet alle Actions die sich auf Dokumente beziehen.
 comment	Beinhaltet alle Actions die sich auf Kommentare beziehen.
 ShowUserInfoAction	Action, die beim Anzeigen der Info eines Users angesprochen wird. Parameter: Aktueller User: Session -> aktUser loginName: request -> loginName Rechteüberprüfung für GUI: isAllowedShowUserSettings Managemethoden: showUserInfo Beispiel-Aufruf: do=showUserInfo&loginName=loginName
 SearchUserAction	Action, die beim Suchen eines Users angesprochen wird. (UNUSED) Parameter: Aktueller User: Session -> aktUser searchValue: request -> searchValue Rechteüberprüfung für GUI: keine Managemethoden: searchUser Beispiel-Aufruf: do=searchUser&searchValue=loginName
 ShowAllUserAction	Action, die beim Anzeigen aller User angesprochen wird. Parameter: Aktueller User: Session -> aktUser Rechteüberprüfung für GUI:






	<p>isAllowedDeleteUserAction</p> <p>Managermethoden: showAllUser</p> <p>Beispiel-Aufruf: do=showAllUser</p>
 ShowUserSettingsAction	<p>Action, die beim Anzeigen der Settings eines Users angesprochen wird.</p> <p>Parameter: Aktueller User: Session -> aktUser loginName: request -> loginName</p> <p>Rechteüberprüfung für GUI: isAllowedUpdateUserSettings</p> <p>Managermethoden: showUserSettings</p> <p>Beispiel-Aufruf: do=showUserSettings&loginName=loginName</p>
 UpdateUserSettingsAction	<p>Action, die beim Ändern von Settings eines Users angesprochen wird.</p> <p>Parameter: Aktueller User: Session -> aktUser loginName: request -> loginName nachname: request -> nachname vorname: request -> vorname neuesPasswortEins: request -> neuesPasswortEins neuesPasswortZwei: request -> neuesPasswortZwei</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: updateUserSettings</p> <p>Beispiel-Aufruf: do=updateUserSettings&loginName=neuerUser&nachname=Schmidt&vorname=Kurt&neuesPasswortEins=bla&neuesPasswortZwei=bla</p>
 RegisterAction	<p>Action, die beim Hinzufügen eines neuen Users angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser loginName: request -> loginName passwort: request -> passwort passwortWdh1: request -> passwortWdh1 nachname: request -> nachname vorname: request -> vorname</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: register</p> <p>Beispiel-Aufruf:</p>







	<code>do=Register&loginName=neuerUser&passwort=passwort&passwortWdhl=passwort&nachname=Schmidt&vorname=Kurt</code>
 LoginAction	<p>Action, die beim Login eines Users angesprochen wird.</p> <p>Parameter: loginName des Accounts: request -> loginname password des Accounts: request -> password</p> <p>Rechteüberprüfung für GUI: isAllowedOpenAdminconsoleAction</p> <p>Managermethoden: login</p> <p>Beispiel-Aufruf: do=Login&loginName=Bert&password=berta</p>
 OpenAdminconsole	<p>Action, die beim Öffnen der AdminConsole angesprochen wird.</p> <p>Parameter: Aktueller User: Session -> aktUser</p> <p>Rechteüberprüfung für GUI: isAllowedRegister</p> <p>Managermethoden: showAllProjects showAllUser</p> <p>Beispiel-Aufruf: do=OpenAdminconsole</p>
 LogoutAction	<p>Action, die beim Logout eines Users angesprochen wird.</p> <p>Parameter: keine</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: logout</p> <p>Beispiel-Aufruf: do=Logout</p>
 DeleteUserAction	<p>Action, die beim Löschen eines Users angesprochen wird.</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject loginName(LoginName des Users): request -> loginName</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deleteUser</p>






	Beispiel-Aufruf: do=DeleteUser&loginName=Bert
 UpdateTaskAction	<p>Action, die beim Updaten einer Task in einem Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject taskId(Id des Tasks): request -> taskId (optional)titel(titel des Tasks): request -> titel (optional)aufgabenStellung(aufgabenStellung des Tasks): request -> aufgabenStellung (optional)date(Termin des Tasks): request -> date (optional)done(Status des Tasks): request -> done</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: updateTask</p> <p>Beispiel-Aufruf: do=UpdateTask&taskId=5&titel=DeineAufgabe&date=2011-06-10&done=true</p>
 ShowAllOwnTasksAction	<p>Action die ausgeführt wird, wenn die eigenen Tasks zu einem Projekt in dem der User teilnimmt angezeigt werden sollen</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject loginName(für den Member): request -> loginName</p> <p>Rechteüberprüfung für GUI: isAllowedShowAllOwnTasks</p> <p>Managermethoden: deAssignTask</p> <p>Beispiel-Aufruf: do=ShowAllOwnTasks</p>
 AssignTaskAction	<p>Action, die beim Zuordnen eines Tasks zu einem Member des Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject taskId(Id des Tasks): request -> taskId loginName(für Memberauswahl): request -> loginName</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: assignTask</p> <p>Beispiel-Aufruf: do=AssignTask&loginName=karl&taskId=5</p>





 DeAssignTaskAction	<p>Action, die beim Abordern eines Member des Projekts von einem Task, angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject taskId(Id des Tasks): request -> taskId loginName(für Memberauswahl): request -> loginName</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deAssignTask</p> <p>Beispiel-Aufruf: do=DeAssignTask&loginName=karl&taskId=5</p>
 AddNewTaskAction	<p>Action, die beim Hinzufügen eines neuen Tasks in einem Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject titel(Bezeichnung des Tasks): request -> titel aufgabenStellung(formulierte Aufgabenstellung): request -> aufgabenStellung date(Datum der Form "yyyy-mm-dd"): request -> date</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: addNewTask</p> <p>Beispiel-Aufruf: do=AddNewTask&titel=TestAufgabe&aufgabenStellung=Tue%20dies%20und%20das!&date=2011-06-02</p>
 DeleteTaskAction	<p>Action, die beim Löschen eines Tasks angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject taskId(Id des Tasks): request -> taskId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deleteTask</p> <p>Beispiel-Aufruf: do=DeleteTask&taskId=1</p>
 ShowAllTasksAction	<p>Action die angesprochen wird, wenn alle Aufgaben eines Projektes angezeigt werden sollen</p> <p>Parameter: Aktueller User: Session -> aktUser</p>




	<p>Aktuelles Project: Session -> aktProject taskId(Id des Tasks): request -> taskId</p> <p>Rechteüberprüfung für GUI: isAllowedUpdateTaskAction, isAllowedAddNewTaskAction, isAllowedDeleteTaskAction, isAllowedShowAllMemberAction, isAllowedAssignTaskAction, isAllowedDeAssignTaskAction</p> <p>Managermethoden: showAllTasks</p> <p>Beispiel-Aufruf: do=ShowAllTasks&taskId=0</p>
 ShowAllSourceAction	<p>Action, die beim Anzeigen aller Sourcecodes in einem Projekt angesprochen wird, dabei wird das erste Element detailliert angezeigt falls der Parameter sourcecodeId null ist. Wenn der Parameter nicht null ist wird der entsprechende Sourcecode angezeigt.</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject (optional)sourcecodeId: request -> sourcecodeId</p> <p>Rechteüberprüfung für GUI: isAllowedDeleteSourceAction</p> <p>Managermethoden: showAllSource</p> <p>Beispiel-Aufruf: do=ShowAllSourceAction</p>
 UpdateSourceAction	<p>Action, die beim Updaten eines Sourcecodes in einem Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject sourcecodeId(Id des Sourcecodes): request -> sourcecodeId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: updateSource</p> <p>Beispiel-Aufruf: do=UpdateSourceAction</p>
 ShowSourceAction	<p>Action, die beim Anzeigen eines Sourcecode in einem Projekt angesprochen wird (UNUSED)</p>







	<p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject sourcecodeId(Id des Sourcecodes): request -> sourcecodeId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showSource</p> <p>Beispiel-Aufruf: do=ShowSourceAction&sourcecodeId=1</p>
 AddNewSourceAction	<p>Action, die beim Hinzufügen eines neuen Sourcecodes angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject data(List FileItem): request -> data</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: addNewSource</p> <p>Beispiel-Aufruf: do=AddNewSourceAction</p>
 DownloadSourceAction	<p>Action, die beim Download eines Sourcecodes angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject sourcecodeId(Id des Sourcecodes): request -> sourcecodeId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: downloadSource</p> <p>Beispiel-Aufruf: do=DownloadSourceAction&sourcecodeId=1</p>
 DeleteSourceAction	<p>Action, die beim Löschen eines Sourcecodes angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject sourcecodeId(Id des Sourcecodes): request -> sourcecodeId</p> <p>Rechteüberprüfung für GUI: keine</p>






	<p>Managermethoden: deleteSource</p> <p>Beispiel-Aufruf: do=DeleteSourceAction&sourcecodeId=1</p>
 UpdateMemberAction	<p>Action, die beim Updaten eines Members angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject Loginname(für Memberauswahl): request -> loginName rolle(Projectrolle): request -> rolle</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: addMember</p> <p>Beispiel-Aufruf: do=UpdateMember&loginName=Michael&rolle=Leader</p>
 ShowAllProjectsAction	<p>Action, die beim Anzeigen aller Projekte angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showAllProjects</p> <p>Beispiel-Aufruf: do=ShowAllProjects</p>
 ShowProjectAction	<p>Action, die das angeklickte Project darstellt. Es werden allgemeine Informationen zum Project dargestellt und Optionen zur Memberverwaltung bereitgestellt. Zuvor werden jedoch die Rechte überprüft.</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject Projectname(für Wechsel des aktProject): request -> projectName Loginname(für Memberauswahl): request -> loginName</p> <p>Rechteüberprüfung für GUI: isAllowedAddMemberAction isAllowedDeleteMemberAction isAllowedShowAllTasksAction</p> <p>Managermethoden: ShowProject, ShowAllMember,</p>





	<p>ShowMember, showUserInfo</p> <p>Beispiel-Aufruf: do=ShowProject&projectName=ProjectName</p>
 ShowAllOwnProjectsAction	<p>Action, die beim Anzeigen aller eigenen Projekte angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showAllOwnProjects</p> <p>Beispiel-Aufruf: do=ShowAllOwnProjects</p>
 ShowAllMemberAction	<p>Action, die beim Anzeigen aller Member in einem Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject loginName(loginName des Users): request -> loginName</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showMember</p> <p>Beispiel-Aufruf: do=ShowAllMember&loginName=Heinz</p>
 DeleteProjectAction	<p>Action, die beim Löschen eines Projektes angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject projectName(Bezeichnung des Projektes): request -> projectName</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deleteProject</p> <p>Beispiel-Aufruf: do=DeleteProject&projectName=Bezeichnung</p>
 DeleteMemberAction	<p>Action, die beim Löschen eines Members angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject userLoginName(loginname eines Users der Member ist): request -> userLoginName</p>







	<p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deleteMember</p> <p>Beispiel-Aufruf: do=DeleteMember&userLoginName=karl</p>
 AddNewProjectAction	<p>Action, die beim Hinzufügen eines neuen Projekts angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser projectName(Bezeichnung des Projektes): request -> projectName status(Projectstatus): request -> status</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: addNewProject</p> <p>Beispiel-Aufruf: do=AddNewProject&projectName=NeuesProjekt&status=beginn</p>
 SearchProjectsAction	<p>Action, die beim Suchen von Projekten angesprochen wird (UNUSED)</p> <p>Parameter: Aktueller User: Session -> aktUser searchValue(Teil einer Bezeichnung eines gesuchten Projektes): request -> searchValue</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: searchProjects</p> <p>Beispiel-Aufruf: do=SearchProjects&searchValue=a</p>
 AddMemberAction	<p>Action, die beim Hinzufügen eines Members in einem Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject loginName(loginName eines Users): request -> loginName rolle(Projectrolle): request -> rolle</p> <p>Rechteüberprüfung für GUI: keine</p>






	<p>Managermethoden: addMember</p> <p>Beispiel-Aufruf: do=AddMember&loginName=Heinz&rolle=Member</p>
 UpdateDocuAction	<p>Action, die beim Updaten eines Dokuments in einem Projekt angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject documentId(Id des Dokuments): request -> documentId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: updateDocu</p> <p>Beispiel-Aufruf: do=UpdateDocuAction</p>
 DownloadDocuAction	<p>Action, die beim Download eines Dokuments angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject documentId(Id des Documents): request -> documentId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: downloadDocu</p> <p>Beispiel-Aufruf: do=DownloadDocuAction&documentId=1</p>
 ShowDocuAction	<p>Action, die beim Anzeigen eines Dokuments in einem Projekt angesprochen wird (UNUSED)</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject documentId(Id des Dokuments): request -> documentId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showDocuContent</p> <p>Beispiel-Aufruf:</p>







	do=ShowDocuAction&documentId=1
 ShowAllDocuAction	<p>Action, die beim Anzeigen aller Dokumente in einem Projekt angesprochen wird, dabei wird das erste Element detailliert angezeigt falls der Parameter documentId null ist. Andernfalls wird das entsprechende Dokument angezeigt.</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject (optional)documentId: request -> documentId</p> <p>Rechteüberprüfung für GUI: isAllowedDeleteDocuAction</p> <p>Managermethoden: showAllDocu</p> <p>Beispiel-Aufruf: do=ShowAllDocuAction&documentId=0</p>
 DeleteDocuAction	<p>Action, die beim Löschen eines Dokuments angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject documentId(Id des Documents): request -> documentId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deleteDocu</p> <p>Beispiel-Aufruf: do=DeleteDocuAction&documentId=1</p>
 AddNewDocuAction	<p>Action, die beim Hinzufügen eines neuen Dokuments angesprochen wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject data(List FileItem): request -> data</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: addNewDocu</p> <p>Beispiel-Aufruf: do=AddNewDocuAction</p>
 ShowAllComments41TaskAction	<p>Action die angesprochen wird, wenn zu einem Task alle Kommentare angezeigt werden</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject</p>







	<p>taskId(Id des Tasks): request -> taskId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: ShowAllComments41Task</p> <p>Beispiel-Aufruf: do=ShowAllComments41Task&taskId=1</p>
 UpdateCommentAction	<p>Action die angesprochen wird, wenn ein Kommentar geupdatet werden soll</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject commentId(Id des Comments): request -> commentId entry(Neuer Inhalt des Comments): request -> entry</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: updateComment</p> <p>Beispiel-Aufruf: do=UpdateComment&commentId=1&entry=aenderungInhalt</p>
 CommentDocuAction	<p>Action die angesprochen wird wenn ein Dokument kommentiert wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject documentId(Id des Dokuments): request -> documentId entry(Inhalt des Comments): request -> entry</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: commentDocu</p> <p>Beispiel-Aufruf: do=CommentDocu&documentId=1&entry=GutesDokument</p>
 ShowAllComments41ProjectAction	<p>Action die angesprochen wenn zu einem Projekt alle Kommentare angezeigt werden</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject</p> <p>Rechteüberprüfung für GUI: keine</p>




	<p>Managermethoden: showAllComments41Project</p> <p>Beispiel-Aufruf: do=ShowAllComments41Project</p>
 ShowAllComments41Source Action	<p>Action die angesprochen wenn zu einem Sourcecode alle Kommentare angezeigt werden</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject sourcecodeId(Id des Sourcecodes): request -> sourcecodeId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showAllComments41Source</p> <p>Beispiel-Aufruf: do=ShowAllComments41Source&sourcecodeId=1</p>
 ShowAllComments41DocuAct ion	<p>Action die angesprochen wird, wenn zu einem Dokument alle Kommentare angezeigt werden</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject documentId(Id des Dokuments): request -> documentId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: showAllComments41Docu</p> <p>Beispiel-Aufruf: do=ShowAllComments41Docu&documentId=1</p>
 DeleteCommentAction	<p>Action die angesprochen wird wenn ein Kommentar gelöscht werden soll</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject commentId(Id des Comments): request -> commentId</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: deleteComment</p> <p>Beispiel-Aufruf: do=DeleteComment&commentId=1</p>
 CommentTaskAction	<p>Action die angesprochen wird wenn eine Aufgabe kommentiert wird</p>



	<p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject taskId(Id des Tasks): request -> taskId entry(Inhalt des Comments): request -> entry</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: commentTask</p> <p>Beispiel-Aufruf: do=CommentSource&taskId=1&entry=DieseAufgabelstDringend</p>
 CommentSourceAction	<p>Action die angesprochen wird wenn ein Sourcecode kommentiert wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject sourcecodeId(Id des Sourcecodes): request -> sourcecodeId entry(Inhalt des Comments): request -> entry</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: commentSource</p> <p>Beispiel-Aufruf: do=CommentSource&sourcecodeId=1&entry=GuterSource</p>
 CommentProjectAction	<p>Action die angesprochen wird wenn in einem Project kommentiert wird</p> <p>Parameter: Aktueller User: Session -> aktUser Aktuelles Project: Session -> aktProject projectId(Id des projects): request -> projectId entry(Inhalt des Comments): request -> entry</p> <p>Rechteüberprüfung für GUI: keine</p> <p>Managermethoden: commentProject</p> <p>Beispiel-Aufruf: do=CommentProject&projectId=ProjectName&entry=KickOff Started</p>
 manager	Beinhaltet alle Manager die für jProject benötigt werden.
 GlobalRolesManager	Dieser Manager kontrolliert den Zugriff auf die Globalen Rechte



 ProjectRolesManager	Dieser Manager kontrolliert den Zugriff auf die Projekt-Rechte
 CommentManager	Dieser Manager kontrolliert den Zugriff auf die Kommentare
 DocumentManager	diese Klasse kontrolliert den Zugriff auf die Documente
 SourceManager	diese Klasse kontrolliert den Zugriff auf den SourceCode
 UserManager	Dieser Manager kontrolliert den Zugriff auf die User
 TaskManager	Dieser Manager kontrolliert den Zugriff auf die Tasks
 ProjectManager	Dieser Manager kontrolliert den Zugriff auf die Projekte
 MainManager	Hauptmanager fuer die Aktionen (Haupt BO-Klasse), besitzt die Untermanager(BOs) zu den einzelnen Teilbereichen an Abfragen und Aktionen des systems
 repository	Beinhaltet Zugriffsschichten zur Datenbank.
 dao	Beinhaltet alle Interfaces und Factory für die DAO-Schicht(Datenbankzugriff). Dieses Package bietet die Funktionalität für Standardzugriffe auf die Datenbank.
 daoimpl	Beinhaltet alle Implementationen und Factory für die DAO-Schicht(Datenbankzugriff). Dieses Package bietet die Funktionalität für Standardzugriffe auf die Datenbank.
 da	Beinhaltet alle Interfaces und Factory für die DA-Schicht(Datenbankzugriff). Dieses Package bietet die Funktionalität für erweiterte Zugriffe auf die Datenbank.
 daimpl	Beinhaltet alle Implementationen und Factory für die DA-Schicht(Datenbankzugriff). Dieses Package bietet die Funktionalität für erweiterte Zugriffe auf die Datenbank.