

**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/User-Sicht auf das Portal ......................................................5

2.2 Funktionen Detailsicht .......................................................................................5

2.2.x Use-Case x .................................................................................................5

3. Architektur ..............................................................................................................6

3.1 Schichten ..........................................................................................................6

3.2 Komponenten ....................................................................................................6

3.2.x Komponente <Name> (z.B. UserManager, ShoppingCard, etc) .................6

4. Systemvoraussetzungen ........................................................................................7

4.1 Systemspezifische Komponenten .....................................................................7

4.2 Allgemeine Voraussetzungen ............................................................................7

Anhang .......................................................................................................................8

A1 Execution Sicht/Deployment Sicht .....................................................................8

A2 Schnittsellen-Sprache (falls vorhanden z.B. DTD) .............................................8

A3 Konfigurationseinstellungen ...............................................................................8

A4 Testszenarien .....................................................................................................8

A5 Monitoring (welche Ports sind erreichbar, Prozesse, Directories, ...) .................8

was unbedingt in die technische doku MUSS

das projekt ist optimiert auf firefox 4,5

also 4 und 5

**1. Einführung**

- Was ist die Zielaufgabe des Projektes?

Konzeptuelle Sicht - sehr knapp nur zum Verständnis wichtig, siehe Projekt-Idee

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/Sourcecode 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 Herarchie 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

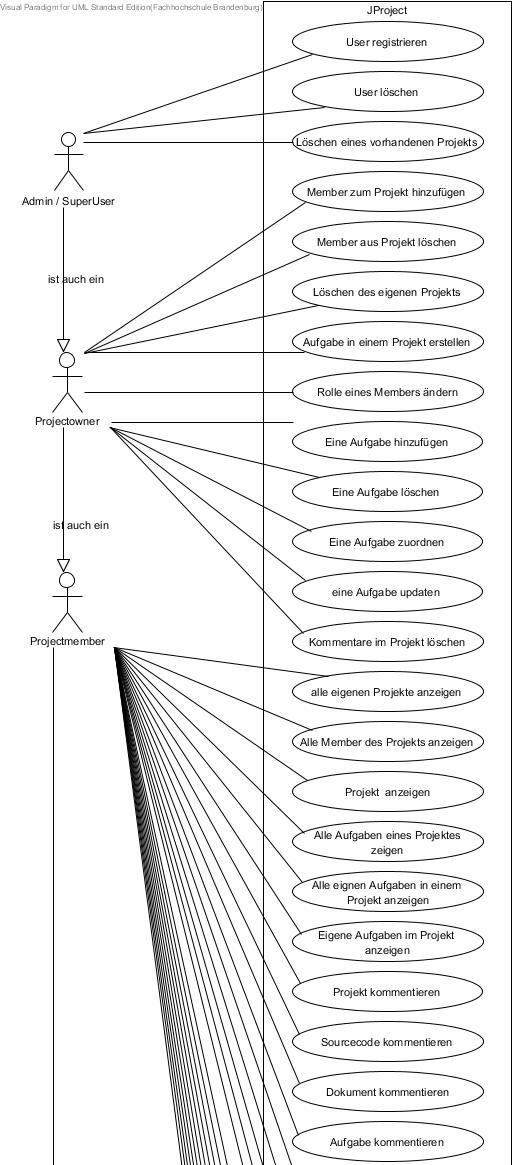
(- Vorbereitung für verschiedene Sprachen)

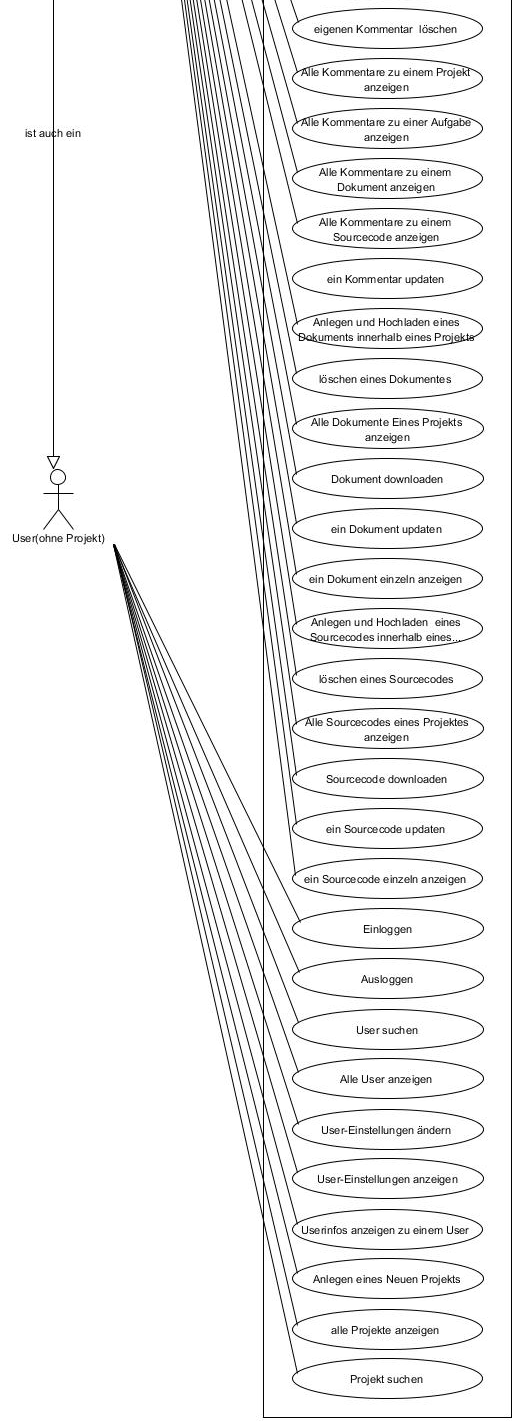


(Ursprünglicher Ideen-Entwurf)

**2. Portal-Funktionen**

2.1 Funktionale Sicht/User-Sicht auf das Portal





Projectmember -> ein kommentar updaten (nur sein eigenes!!!!!)

Kann kein sourcecode oder dokumente löschen!

ProjectOwner -> kann Alle kommentare updaten!

Kann sourcecode/Dokument löschen (auch welche die nicht er selber gem8 hat)

User (ohne projekt)-> kann keine user suchen

Kann nicht alle User anzeigen

EIGENE usereinstellungen ändern/anzeigen

Kann nicht alle projekte anzeigen

Kann keine projekte suchen

In dem UseCase-Diagramm ist nur eine starre Konstellation der Rollen zu erkennen.

Implementiert wurde ein Rollensystem welches erlaubt dynamisch Rollen hinzu zufügen.

Es gibt ein globales Rollensystem (siehe auch GlobalRolesManager)in ihm wird unterschieden welche Aktionen der Benutzer im Allgemeinen tätigen darf hier(<-?) als Beispiel zu sehen der User ohne Projekt und der Admin, dies sind zwei mögliche globale Rollen.

(Global = 2 Rollen sind ein Muss -> Admin + Member

Projekt = 2 Rollen sind ein Muss -> Leader + Member)

Als weiteres gibt’s es aber noch ein Rollensystem nur für Projekte (siehe auch ProjectRolesManger) in ihm wird unterschieden welche Aktionen der Benutzer innerhalb eines Projektes tätigen darf. Als Beispiel dafür ist im UseCase-Diagramm(?->) zu sehen der Projektowner/Leader und der Projectmember/Member.

Der Admin hat durch seine Globale Rolle eine übergeordnete Funktion und darf grundsätzlich alles was andere Rollen auch dürfen, dann kommt der Projektowner (und) innerhalb eines Projektes hat dieser mehr Rechte als ein Member. User die in einem Projekt nicht involviert sind, also weder Member noch Leader, sind hier dann nur ein User ohne Projekt(<-?).

(nur als Beispiel für Rollen die momentan mit der DB ausgeliefert werden)

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

## 2.2 Funktionen Detailsicht

Hier sollten die einzelnen UseCases spezieller beschrieben werden. Dies kann hier auch aus Usersicht geschehen, so dass man einen Überblick über das System bekommt. Die technische Sicht auf die einzelnen Use-Cases/Komponeten werden dann im 3. Kapitel vorgestellt.

2.2.1 User registrieren

2.2.2 User löschen

2.2.3 Löschen eines vorhandenen Projekts

2.2.4 Member zum Projekt hinzufügen

2.2.6 Member aus Projekt löschen

2.2.5 Löschen des eigenen Projekts

2.2.7 Aufgabe in einem Projekt erstellen

2.2.8 Rolle eines Members ändern

2.2.9 Eine Aufgabe hinzufügen

2.2.10 Eine Aufgabe löschen

2.2.11 Eine Aufgabe zuordnen

2.2.12 eine Aufgabe updaten

2.2.13 Kommentare im Projekt löschen

2.2.14 alle eigenen Projekte anzeigen

2.2.15 Alle Member des Projekts anzeigen

2.2.16 Projekt anzeigen

2.2.17 Alle Aufgaben eines Projektes zeigen

2.2.18 Alle eignen Aufgaben in einem Projekt anzeigen

2.2.19 Eigene Aufgaben im Projekt anzeigen

2.2.20 Projekt kommentieren

2.2.21 Sourcecode kommentieren

2.2.22 Dokument kommentieren

2.2.23 Aufgabe kommentieren

2.2.24 eigenen Kommentar löschen

2.2.25 Alle Kommentare zu einem Projekt anzeigen

2.2.26 Alle Kommentare zu einer Aufgabe anzeigen

2.2.27 Alle Kommentare zu einem Dokument anzeigen

2.2.28 Alle Kommentare zu einem Sourcecode anzeigen

2.2.29 ein Kommentar updaten

2.2.30 Anlegen und Hochladen eines Dokuments innerhalb eines Projekts

2.2.31 löschen eines Dokumentes

2.2.32 Alle Dokumente Eines Projekts anzeigen

2.2.33 Dokument downloaden

2.2.34 ein Dokument updaten

2.2.35 ein Dokument einzeln anzeigen

2.2.36 Anlegen und Hochladen eines Sourcecodes innerhalb eines Projekts

2.2.37 löschen eines Sourcecodes

2.2.38 Alle Sourcecodes eines Projektes anzeigen

2.2.39 Sourcecode downloaden

2.2.40 ein Sourcecode updaten

2.2.41 ein Sourcecode einzeln anzeigen

2.2.42 Einloggen

2.2.43 Ausloggen

2.2.44 User suchen

2.2.45 Alle User anzeigen

2.2.46 User-Einstellungen ändern

2.2.47 User-Einstellungen anzeigen

2.2.48 Userinfos anzeigen zu einem User

2.2.49 Anlegen eines Neuen Projekts

2.2.50 alle Projekte anzeigen

2.2.51 Projekt suchen

# **3. Architektur**

## 3.1 Schichten

Struktur/Architekur dokumentiert:

- Welche Schichten gibt es? (hier kann man bereits in einem Package-Diagramm die wichtigsten Klassen andeuten)



- Mindestens ein Sequenzdiagramm, das einen interessanten Ablauf über Schichten/Rechnergrenzen hinweg veranschaulicht, anhand dessen die Funktionalität des Systems erklärt wird (die genaue Sicht der einzelnen Kommunikationsszenarien würde man unter 3.2 angeben)

//TODO Sequenzdiagramm mit erklärungen

Das Ganze nicht nur mit Diagrammen, sondern auch schriftlich erkären, was die Diagramme ausdrücken.

Dieses Unterkapitel soll zunächst einen schnellen Überblick über die Gesamtarchitektur vermitteln. In 3.2 wird dann näher auf die einzelnen Komponenten/Klassen eingegangen.

## 3.2 Komponenten

Eventuell hat Ihre Anwendung ausschließlich eine Komponente, da Se keine weiteren Schnitte zwischen einzelnen Anwendungsmodulen (z.B. UserManager/UserBO, SoppingCard, ...) vorgenommen haben. Dann würde man sicher hier von Klassen-Sicht und nicht von Komponenten sprechen und - wenn überhaupt - die Schichten als Unterkapitel aufteilen.

Es sollte hier auf jeden Fall herauskommen, wer welches Objekt benutzt, so dass man die Abhängigkeiten erkennen kann. Insbesondere ist das für die Value Objekte interessant, da diese die Datenstruktur für die Anwendung widerspiegeln. Da es sich hier um die technische Sicht geht, sollte man auch bereits mögliche Aggregationen und Kompositionen (siehe auch UML) berücksichtigen.

3.2.x Komponente <Name> (z.B. UserManager, ShoppingCard, etc)

-- Komponente und deren Hauptfunktionen sollten klar werden (Logische Sicht) -- Für die jeweiligen Komponenten ein Klassendiagramm oder grobes Klassendiagramm als Übersicht.

--Bitte bei verwendeten Mustern, diese beschreiben und deren Verwendung begründen, d.h. hier etwas mehr ins Detail gehen. li

3.2.1 GUI

3.2.2 Controller

3.2.2.1 Servlet

3.2.2.2 Actions

3.2.3. Manager

3.2.3.1 CommentManager

3.2.3.2 DocumentManager

3.2.3.3 GlobalRolesManager

3.2.3.4 MainManager

3.2.3.5 ProjectManager

3.2.3.6 ProjectRolesManager

3.2.3.7 SourceManager

3.2.3.8 TaskManager

3.2.3.9 UserManager

3.2.4. Model

//TODO Diagramm Domänenmodell

3.2.4.1 Comment

3.2.4.2 CommentDocument

3.2.4.3 CommentProject

3.2.4.4 CommentSourcecode

3.2.4.5 CommentTask

3.2.4.6 Document

3.2.4.7 GlobalRoles

3.2.4.8 ICQ

3.2.4.9 Member

3.2.4.10 Project

3.2.4.11 ProjectRoles

3.2.4.12 Skype

3.2.4.13 Sourcecode

3.2.4.14 Task

3.2.4.15 Telefon

3.2.4.16 Termin

3.2.4.17 User

3.2.5 DA

3.2.6 DAO

3.2.7 Datenbank

//TODO hier ERR Diagramm

+ 1,2 sätze zu Hibernate

**4. Systemvoraussetzungen**

4.1 Systemspezifische Komponenten

- welche Komponenten sind von der eigenen Software notwendig (die Komponenten liegen ür die Installation typischerweise als jar-/war-Dateien vor)

- externe Libraries, die von Ihrer Applikation benötigt werden

4.2 Allgemeine Voraussetzungen

- Ablaufumgebung (Java-Version)

- Frameworks

- Container

- DB

- Betriebssystem

**Anhang**

muss nicht abgegeben werden, steht hier nur zur Vervollständigung

A1 Execution Sicht/Deployment Sicht

Ist typischerweise individuell für den jeweiligen Kunden und sollte daher im Anhang hinzugefügt werden

A2 Schnittsellen-Sprache (falls vorhanden z.B. DTD)

Falls Sie eine API nach außen zur Verfügung stellen (z.B. enen REST-Service), sollte klar sein in welchem Format die Daten ausgetauscht werden

A3 Konfigurationseinstellungen

Hier sollten für den jeweiligen Kunden die speziellen Konfigurationsdaten aufgeführt sein, so dass ein Admin des Kunden nachvollziehen kann, was geändert werden muss, wenn sich die Systemlandschaft ändert

A4 Testszenarien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Die Testszenarien dienen zur Überprüfung der Installation, so dass systematisch nachvollzogen werden kann, ob das System sich entsprechend verhält Test-Nr | Test-Name | Beschreibung | Voraussetzung | Erwartetes Ergebnis |