

**Dokumentation**

**Autobday**

*Entwurf und Implementierung einer Systems zum automatischen Versand von Geburtstagsmails*

**Team:**

*Andy Klay*

*Michael Koppen*

*Tino Reuschel*

*Christoph Ott*

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

3.2.2Controller ..............................................................................................................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** |
| 0.1 | 29.11.2011 | Klay | init Dokumentation |
| 0.2 | 10.12.2011 | Klay | Einführung,UseCase, Allg.Voraussetzungen |
|  |  |  |  |
|  |  |  |  |

**1. Einführung**

Autobday ist in erster Linie eine Umgebung zum Automatischen Emailversand die von jedem Rechner, der Internet hat, erreicht werden kann.

Autobday wird erst interessant für Nutzer, wenn Sie einen Google-Account haben und Kontakte im Account eingepflegt haben. Der Nutzer kann dann alle seine Kontakte und Gruppen von Google importieren in das System

Mit Autobday kann ein Benutzer für seine Kontakte Templates erstellen und automatisch Geburtstagsmails vom System versenden lassen. Das System bedient sich der Templates zur Erstellung dieser Emails.

Es soll einen Bereich geben in dem der Nutzer Kontakte in den Automatischen Versand einstellen/aktivieren kann und auch herausnehmen/deaktivieren kann. Auch ganze Gruppen sollen aktivierbar oder deaktivier bar sein.

Auch kann der Nutzer mehrere Google kontakte besitzen und deren Kontakte für den automatischen Emailversand verwalten.

***Funktionaler Umfang der Idee:***

- Aktvieren/Deaktivieren von Kontakten

- Importieren der Kontakte von Google

- Automatischer Versand von Emails

**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.

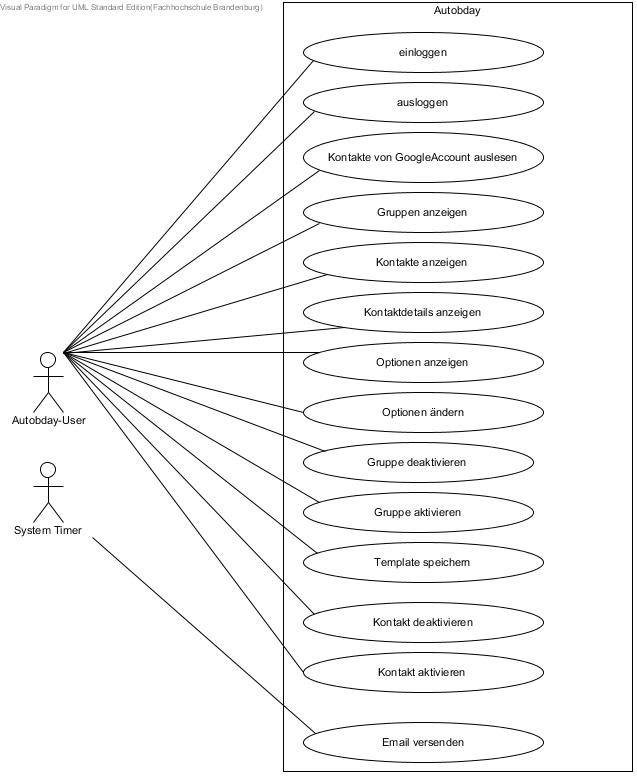
Es gibt ein globales Rollensystem (GlobalRoles). In ihm wird unterschieden welche Aktionen der Benutzer im Allgemeinen tätigen darf. Beispielsweise ist hier im Usecase-Diagramm zu sehen der GMember (User ohne Projekt) und der Admin, dies sind zwei mögliche globale Rollen.

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

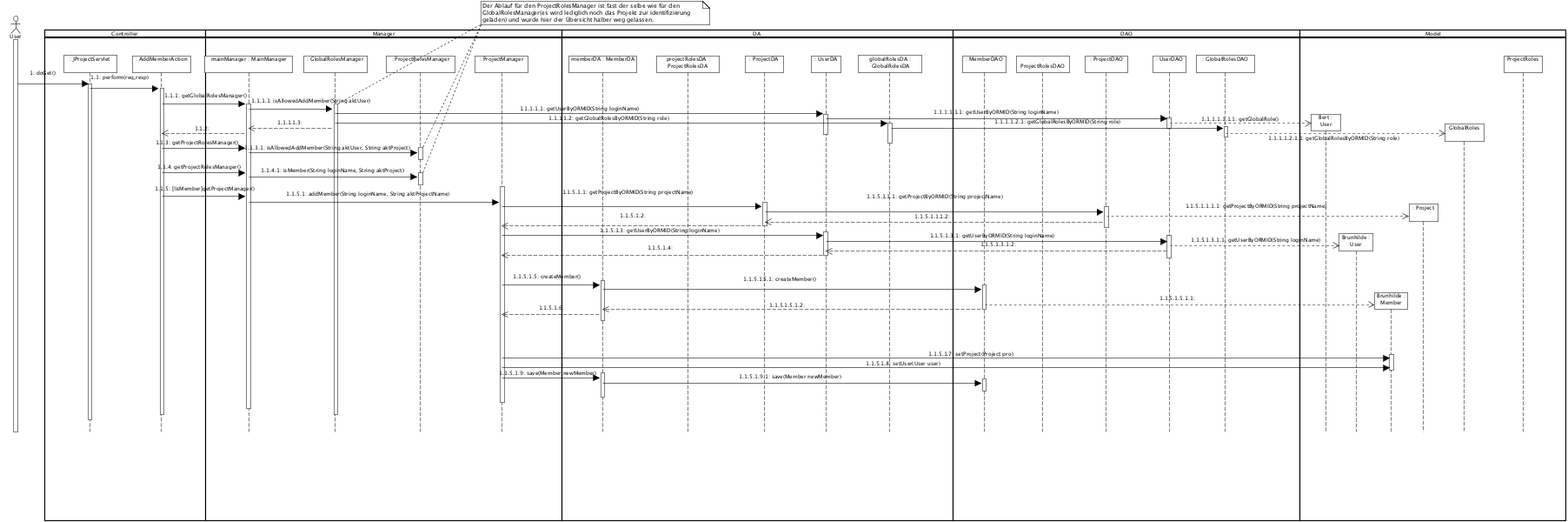
|  |  |
| --- | --- |
| Name | Dokumentation |
| Image1.png Admin | **Admin**= d.h. also ein User mit entsprechenden Rechten, muss nicht Admin heißen. |
| Image2.png User löschen | Über die AdminConsole kann ein Admin einen User aus dem System löschen, damit dieser das System nicht mehr nutzen kann. |
| Image2.png Alle Projekte anzeigen | Nur der Admin kann über die AdminConsole alle Projekte anzeigen. |
| Image2.png Alle User anzeigen | Nur der Admin kann über die AdminConsole alle User anzeigen. |
| Image2.png Löschen eines vorhandenen Projekts | Ein Projekt kann direkt gelöscht werden vom Admin über die AdminConsole. |
| Image2.png User registrieren | Über die AdminConsole kann ein Admin einen User im System erstellen, damit dieser das System nutzen kann. |
| Image1.png Leader | **Projekt-Leader**=d.h. also ein User mit entsprechenden Rechten der in einem Projekt mehr Rechte hat als ein einfacher Teilnehmer(**Projekt-Member**). |
| Image2.pngEine 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. |
| Image2.png 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. |
| Image2.png Löschen des eigenen Projekts | Ein Projekt kann gelöscht werden, indem der Projekt-Leader alle Member des Projektes löscht und zuletzt sich selbst. |
| Image2.png Member zum Projekt hinzufügen | Der Projekt-Leader kann über das Projekt einen User als Projekt-Member hinzufügen. |
| Image2.pngLöschen 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png Member aus Projekt löschen | Der Projekt-Leader kann über das Projekt einen Projekt-Member aus dem Projekt entfernen. |
| Image2.png Alle Kommentare im Projekt löschen | Der Projekt-Leader kannüber 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. |
| Image2.png Aufgabe in einem Projekt hinzufügen | Der Projekt-Leader kann über das Projekt und über den Reiter „Task“ eine Aufgabe zum Projekt hinzufügen. |
| Image2.pngLöschen 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. |
| Image2.png 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. |
| Image1.png GMember | Grund-User, der kein Projektteilnehmer ist |
| Image2.png 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“. |
| Image2.png Ausloggen | Wenn ein User eingeloggt ist, so findet er rechts in der Navigationsleiste den Button „ausloggen“. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png Anlegen eines neuen Projekts | Wenn ein User sich gerade eingeloggt hat so kann er auf dieser Seiter ein neues Projekt anlegen, indem er rechts den Namen des Projektes einträgt und dann auf „addnew Project“ klickt. |
| Image2.pngalle eigenen Projekte anzeigen | Wenn ein User eingeloggt ist, so werden alle seine Projekte auf der linken Seite unter dem Punkt Projekte angezeigt. |
| Image1.png PMember | Ist ein einfacher Projektteilnehmer(**Projekt-Member**). |
| Image2.png 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. |
| Image2.png Alle Sourcecodes eines Projektes anzeigen | Ein Mitglied eines Projektes kann über Projekt und den Reiter „Sourcecode“ sich alle Sourcecodes anzeigen lassen. |
| Image2.pngProjekt anzeigen | Sobald der User sich ans System eingeloggt hat, werden seine Projekte auf der linken Seite angezeigt. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png Aufgaben eines Members im Projekt anzeigen | Ein Member des Projektes sieht auf der Projekt-Seite durch das Anklicken eines Members im Member-Abschnitt unten dann auch dessen zugewiesenden Aufgaben. |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png Alle Member des Projekts anzeigen | Der Projekt-Leader sieht wenn er auf das Projekt klickt alle Member des Projektes. |
| Image2.png Alle Dokumente eines Projekts anzeigen | Ein Mitglied eines Projektes kann über Projekt und den Reiter „Dokumente“ sich alle Dokumente anzeigen lassen. |
| Image2.pngHochladen 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. |
| Image2.png Alle Aufgaben eines Projektes zeigen | Jedes Mitglied eines Projektes kann über den Reiter „Task“ sich alle Aufgaben eines Projektes anzeigen lassen. |
| Image2.pngeigenen 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. |
| Image2.pngeigenen 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. |
| Image2.png 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 |
| Image2.png 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. |
| Image2.png 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. |
| Image2.png Userinfos anzeigen zu einem Member | Wenn man auf den Namen eines Member klickt, so werden Informationen zu diesem Member angezeigt. |
| Image2.png 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)



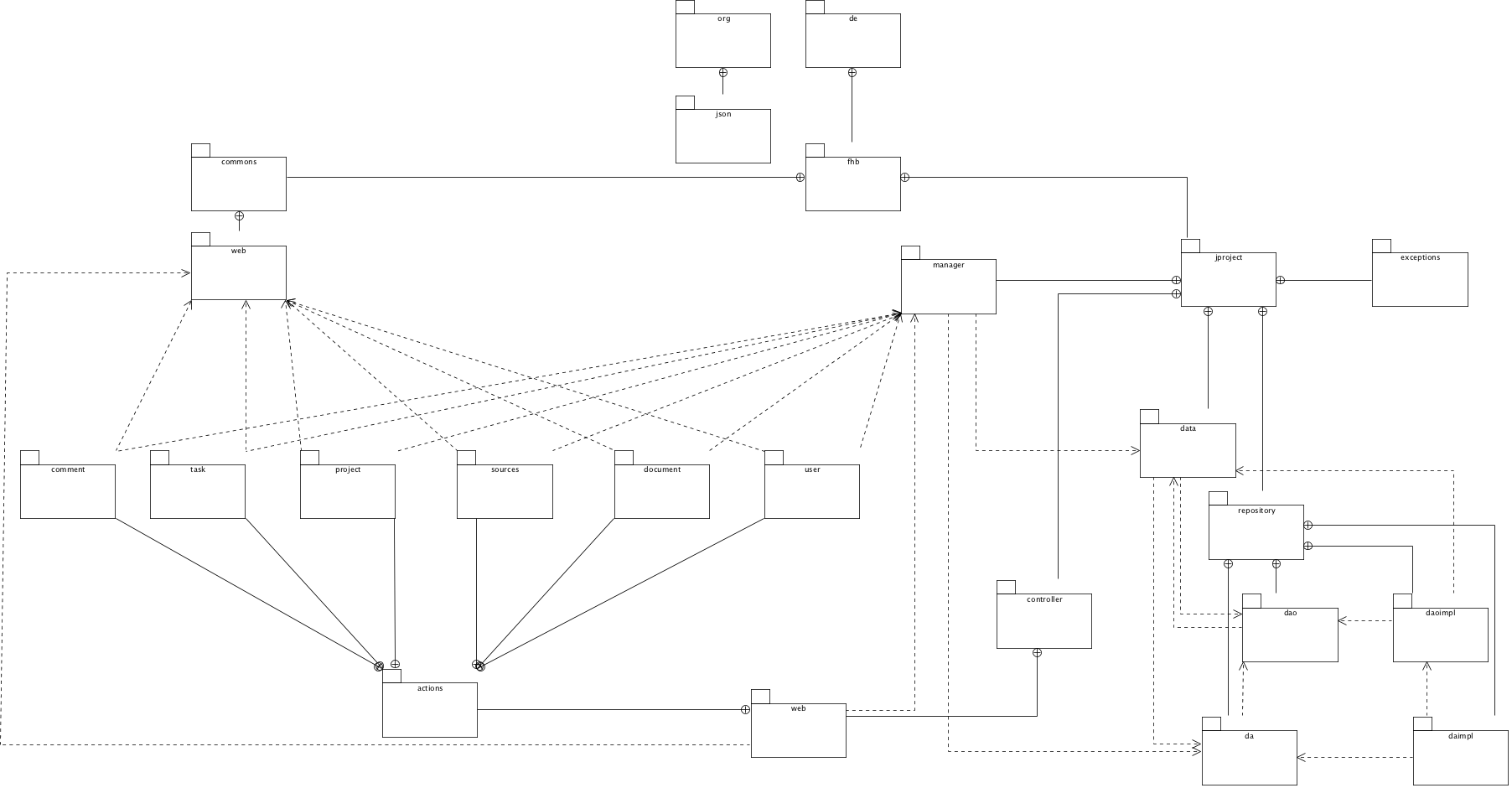
(Abb. 4 Sequenz-Diagramm)

**Beschreibung des Anwendungsfallsmit 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 | Ablauf:   * ***1. doGet()*** * 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)*** * 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 nicht 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 |
| Image12.pngorg | Sprachraum/Organisationform |
| Image12.pngde | Sprachraum/Organisationform |
| Image12.pngjson | beinhaltet Klassen für die JSON-Funktionalität |
| Image12.pngcommons | beinhaltet allgemeine Klassen für immer wiederkehrende Aufgaben |
| Image12.pngfhb | Organisation |
| Image12.pngweb | Allgemeine Klassen bezogen auf das Web |
| Image12.pngjproject | Projektpackage - beinhaltet alles Projektspezifisches |
| Image12.pngexceptions | beinhaltet eigene Exceptions für JProject |
| Image12.pngmanager | beinhaltet alle Manager die für JProject benötigt werden |
| Image12.pngdata | beinhaltet alle Models/VO´s für JProject |
| Image12.pngcomment | beinhaltet alle Actions die sich auf Kommentare beziehen |
| Image12.pngtask | beinhaltet alle Actions die sich auf Tasks beziehen |
| Image12.pngproject | beinhaltet alle Actions die sich auf Projekte beziehen |
| Image12.pngsources | beinhaltet alle Actions die sich auf Sourcecode beziehen |
| Image12.pngdocument | beinhaltet alle Actions die sich auf Dokumente beziehen |
| Image12.pnguser | beinhaltet alle Actions die sich auf den User beziehen |
| Image12.pngrepository | beinhaltet Zugriffsschichten zur Datenbank |
| Image12.pngcontroller | beinhaltet Controller für JProject |
| Image12.pngdao | beinhaltet alle Interfaces und Factory für die DAO-Schicht(Datenbankzugriff), bietet die Funktionalität für Standardzugriffe auf die Datenbank |
| Image12.pngdaoimpl | beinhaltet alle Implementationen und Factory für die DAO-Schicht(Datenbankzugriff), bietet die Funktionalität für Standardzugriffe auf die Datenbank |
| Image12.pngactions | beinhaltet alle Actions für JProject |
| Image12.pngweb | beinhaltet alle Controller/Servlets die für das Web nötig sind |
| Image12.pngda | beinhaltet alle Interfaces und Factory für die DA-Schicht(Datenbankzugriff), bietet die Funktionalität für erweiterte Zugriffe auf die Datenbank |
| Image12.pngdaimpl | 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.6Domänenmodell)

## Zusammenfassung

|  |  |
| --- | --- |
| Name | Dokumentation |
| Image5.png CommentProject | spezieller Kommentar zu einem Projekt |
| Image5.png 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. |
| Image5.pngProjectRoles | 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. |
| Image5.png Sourcecode | Ein Sourcecode ist ein in der Informatik, für Menschen lesbare, in einer Programmiersprache geschriebene Text eines Computerprogramms. |
| Image5.png CommentSourcecode | spezieller Kommentar zu einem Sourcecode |
| Image5.png Comment | Kommentar zu Sourcecode/Document/Project/Task. Beinhaltet spezielle Kommentare. |
| Image5.png CommentDocument | spezieller Kommentar zu einem Dokument |
| Image5.png Document | Ein Document ist eine vom Menschen verfasste Datei  mit Informationen, dessen Darstellung unterschiedlich sein kann. |
| Image5.png CommentTask | spezieller Kommentar zu einem Task |
| Image5.png Task | Ein Task ist eine Aufgabe die es in einem Projekt zu erledigen gilt. |
| Image5.png ICQ | ICQ stellt eine Adresse für einen Messenger dar. |
| Image5.png Termin | Ein Termin ist ein festgelegtes Kalenderdatum. |
| Image5.png User | Der User stellt jeglichen Benutzer des Systems dar mit nötigen Daten zum Einloggen und zum Identifizieren. |
| Image5.png Member | Member ist Assoziationsklasse zwischen User und Project.  Ein Member ist also ein User der in ein Projekt involviert ist. |
| Image5.png Project | Ein bis mehrere User können in einem Project involviert sein zum Lösen einer Problemstellung. |
| Image5.png Skype | Skype stellt eine Adresse für einen Messenger dar |
| Image5.png Telefon | ist eine Telefonnummer |

**3.2.5DA**

DA bietet für erweiterte Zugriffsfunktionalität auf die Datenbank.

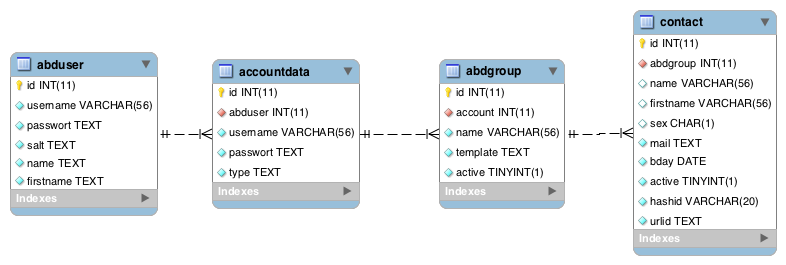
**3.2.6 DAO**

DAO bietet die Grundfunktionalität für Datenbankzugriffe.

**3.2.7Datenbank**

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

# ***Entity-Relationship Diagramm***



(Abb.7 Entity-Relationship Diagramm)

## Zusammenfassung

|  |  |
| --- | --- |
| Name | Dokumentation |
| Image14.png CommentProject | spezieller Kommentar zu einem Projekt |
| Image14.png 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. |
| Image14.png 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. |
| Image14.png CommentSourcecode | spezieller Kommentar zu einem Sourcecode |
| Image14.png Sourcecode | Ein Sourcecode ist ein in der Informatik, für Menschen lesbare, in einer Programmiersprache geschriebene Text eines Computerprogramms. |
| Image14.png Comment | Kommentar zu Sourcecode/Document/Project/Task. Beinhaltet spezielle Kommentare. |
| Image14.png CommentDocument | spezieller Kommentar zu einem Dokument |
| Image14.png Document | Ein Document ist eine vom Menschen verfasste Datei  mit Informationen, deren Darstellung unterschiedlich sein kann. |
| Image14.png CommentTask | Spezieller Kommentar zu einem Task |
| Image14.png Task | Ein Task ist eine Aufgabe die es in einem Projekt zu erledigen gilt. |
| Image14.pngTODO | Zuordnung welcher Member welchen Task zu erledigen hat. |
| Image14.png Termin | Ein Termin ist ein festgelegtes Kalenderdatum. |
| Image14.png ICQ | ICQ stellt eine Adresse für einen Messenger dar. |
| Image14.png Skype | Skype stellt eine Adresse für einen Messenger dar |
| Image14.png User | Der User stellt jeglichen Benutzer des Systems dar mit nötigen Daten zum Einloggen und zum Identifizieren. |
| Image14.png Member | Member ist Assoziationsklasse zwischen User und Project.  Ein Member ist also ein User der in ein Projekt involviert ist. |
| Image14.png Project | Ein bis mehrere User können in einem Project involviert sein zum Lösen einer Problemstellung. |
| Image14.png Telefon | ist eine Telefonnummer |
| Image10.png 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 Autobday:***

Das GUI wurde optimiert für Firefox 7 und 8.

Es wird mindestens Java Version 1.5 benötigt.

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

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

***Autobday benutzt folgende Frameworks und APIs:***

- EasyMock, EasyGloss

- primefaces

- Hibernate, JPA

- JBoss

- Java Server Faces

- 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**

**A4 Klassen-Diagrammübersicht**

# ***Klassendiagramm***

## Zusammenfassung