



HSR

HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Bachelorarbeit

Architekturkonzepte moderner Web-Applikationen

Hochschule für Technik Rapperswil
Frühjahressemester 2013

Erstellt: 10. März 2013, 21:54

Autoren Manuel Alabor
Alexandre Joly
Michael Weibel

Betreuer Prof. Hans Rudin
Experte Daniel Hildebrand
Gegenleser tbd.

Our fancy abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Inhaltsverzeichnis

1. Einleitung	5
1.1. Involvierte Personen	5
1.1.1. Team	5
1.1.2. Betreuung & Bewertung	5
2. Projektplanung	7
2.1. Infrastruktur	7
2.1.1. Projektverwaltung	7
2.1.2. Entwicklungsumgebung	7
2.1.3. Git Repositories	7
2.1.4. Continuous Integration	7
2.2. Meetings	8
2.2.1. Regelmässiges Statusmeeting	8
2.3. Phasenplanung	9
2.4. Meilensteine	11
2.5. Artefakte	12
3. Aufgabenfindung	13
4. Anforderungsanalyse	14
4.1. Funktionale Anforderungen	14
4.2. Nicht-Funktionale Anforderungen	15
4.3. Use Cases	16
4.3.1. UC1: Anmelden	17
4.3.2. UC2: WG erstellen	18
4.3.3. UC3: WG beitreten	19
4.3.4. UC4: WG verlassen	20
4.3.5. UC5: Aufgabe erstellen	21
4.3.6. UC6: Aufgabe bearbeiten	22
4.3.7. UC7: Aufgabe erledigen	23
5. Domainmodel	24

6. Technische Architektur	25
A. Abbildungen, Tabellen & Quellcodes	26
B. Literatur	27
C. Glossar	28
D. Meetingprotokolle	29
E. Aufgabenstellung	36

Kapitel 1 **Einleitung**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1.1. Involvierte Personen

1.1.1. Team

Manuel Alabor

Lorem Ipsum

Alexandre Joly

Lorem Ipsum

Michael Weibel

Lorem Ipsum

1.1.2. Betreuung & Bewertung

Prof. Hans Rudin

Lorem Ipsum

Kevin Gaunt

Lorem Ipsum

Daniel Hildebrand

Lorem Ipsum

Kapitel 2 Projektplanung

2.1. Infrastruktur

<i>Ressource</i>	<i>URL</i>
<i>Projektverwaltung</i>	http://redmine.alabor.me/projects/ba2013
<i>Code: Git Repository</i>	https://github.com/mweibel/ba
<i>Code: CI</i>	https://travis-ci.org/mweibel/BA
<i>Thesis: Git Repository</i>	https://github.com/mweibel/BA-Dokumentation
<i>Thesis: PDF</i>	http://mweibel.github.com/BA-Dokumentation/thesis.pdf
<i>Thesis: CI</i>	https://travis-ci.org/mweibel/BA-Dokumentation
<i>Meeting Protokollierung</i>	https://github.com/mweibel/BA-Dokumentation/wiki/Meetings

Tabelle 2.1.: Projektrelevante URL's

2.1.1. Projektverwaltung

Für die komplette Projektplanung, die Zeitrapportierung sowie das Issue-Management wird Redmine eingesetzt.

2.1.2. Entwicklungsumgebung

Zur Entwicklung von Quellcode-Artefakten steht eine mit Vagrant[Has] provisionierte Virtual Machine bereit. Sie enthält alle notwendigen Abhängigkeiten und Einstellungen.

Eine IDE im klassischen Sinne wird nicht vorgeschrieben.

2.1.3. Git Repositories

Sowohl Quellcodeartefakte als auch die in LaTeX formulierte Thesis (dieses Dokument) wird in auf GitHub abgelegten Git Repositories versioniert bzw. zentral gespeichert.

2.1.4. Continous Integration

Dieses Projekt verwendet Travis CI als Continous Integration Lösung.

Beide Git Repositories (Code & Thesis) verfügen über einen Push-Hook welcher automatisch einen Build im CI-System auslöst.

2.2. Meetings

2.2.1. Regelmässiges Statusmeeting

Während der gesamten Projektdauer findet jeweils am Mittwoch um 10 Uhr ein wöchentliches Statusmeeting statt. Die Sitzung wird abwechselungsweise jeweils von einer Person aus dem Projektteam geführt sowie von einer anderen protokolliert.

Das Projektteam stellt die Agenda der aktuellen Sitzung bis spätestens am vorangehenden Dienstag Abend bereit.

2.3. Phasenplanung

Die Phasenplanung orientiert sich grob am RUP und ist unterteilt in eine *Inception*-, *Elaboration*-, fünf *Construction*- sowie jeweils eine *Transition*- und *Abschlussphase*.

Phase	Dauer	Beschreibung
<i>Inception</i>	3 Wochen	Projektsetup, genauere Definition der Aufgabe, Vorbereitungen & Planungen
<i>Elaboration</i>	3 Wochen	Anforderungsanalysen, Entwicklung eines Architekturprototypen und genauere technische Evaluationen. Guidelines für Quellcode und Testing werden erstellt.
<i>Construction 1</i>	2 Wochen	Umsetzung des Applikationsfundaments, UI Design, Umsetzung erster als <i>Hoch</i> priorisierter Use Cases. Qualitätssicherung in Form von Reviews & Unit Testing.
<i>Construction 2</i>	2 Wochen	Fertigstellung der restlichen als <i>Hoch</i> priorisierten Use Cases. Qualitätssicherung in Form von Reviews & Unit Testing.
<i>Construction 3</i>	2 Wochen	Umsetzung des <i>Gameification</i> -Teils der Applikation. Qualitätssicherung in Form von Reviews & Unit Testing.
<i>Construction 4</i>	2 Wochen	Implementation der restlichen als <i>Mittel</i> priorisierten Use Cases. Qualitätssicherung in Form von Reviews & Unit Testing.
<i>Construction 5</i>	2 Wochen	Umsetzung aller restlichen als <i>Tief</i> priorisierten Use Cases sowie erstes Bugfixing gem. geführter Issueliste. Qualitätssicherung in Form von Reviews & Unit Testing.
<i>Transition</i>	1 Wochen	Abschliessende Bugfixing-Arbeiten. Codefreeze und Erstellung von Deploymentpaketen.
<i>Abschluss</i>	2 Wochen	Finalisierung der Dokumentation sowie Erstellung der HSR Artefakte <i>A100</i> sowie <i>A101</i> .

Tabelle 2.2.: Projektphasenbeschreibung

Jede einzelne Phase wird jeweils von einem Meilenstein abgeschlossen, was in folgendem Gantt-Diagramm ersichtlich ist.

Eine Ausnahme bildet der Meilenstein *M9: Abgabe HSR Artefakte*. Zu diesem Zeitpunkt müssen die entsprechenden Artefakte *A100* sowie *A101* im HSR abgabebereit sein.



Abbildung 2.1.: Phasenübersicht mit Meilensteinen, Kalenderwochen Februar bis Juli 2013

2.4. Meilensteine

<i>ID</i>	<i>Meilenstein</i>	<i>Termin</i>	<i>Beschreibung</i>
<i>M1</i>	Ende Inception	10.03.2013	Die Aufgabenstellung wurde gem. Auftrag klar definiert und die Projektinfrastruktur ist aufgesetzt. Eine initiale Projektplanung besteht.
<i>M2</i>	Ende Elaboration	17.03.2013	Konkrete Technologie und Guidelines sind definiert. Anforderungsdokumente sind erstellt und abgenommen. Initiale SAD und Architekturprototyp bereit.
<i>M3</i>	Ende Construction 1	31.03.2013	Das Fundament der Applikation wurde implementiert. Weiter wurden die ersten Use Cases der Priorität <i>Hoch</i> umgesetzt.
<i>M4</i>	Ende Construction 2	14.04.2013	Alle Use Cases der Priorität <i>Hoch</i> sind umgesetzt.
<i>M5</i>	Ende Construction 3	28.04.2013	
<i>M6</i>	Ende Construction 4	12.05.2013	
<i>M7</i>	Ende Construction 5	26.05.2013	
<i>M8</i>	Ende Transition	02.06.2013	Deploymentpakete und zugehörige Anleitungen sind bereit. Bugfixing abgeschlossen resp. ausstehende Bugs dokumentiert.
<i>M9</i>	Abgabe HSR Artefakte	07.06.2013	Das A0-Poster sowie die Kurzfassung der Bachelorarbeit sind dem Betreuer zugestellt.
<i>M10</i>	Abgabe Bachelorarbeit	14.06.2013	Alle abzugebenden Artefakte sind dem Betreuer zugestellt worden.

Tabelle 2.3.: Meilensteine

2.5. Artefakte

Dieser Abschnitt beschreibt alle Arbeitsprodukte (Artefakte), welche zwingend erstellt und abgegeben werden müssen.

Falls nicht anders vermerkt sind alle Artefakte Teil der Dokumentation.

ID	Meilenstein	Artefakt	Beschreibung
A20	M2	Projektplanung	Projektablauf, Infrastrukturbeschreibung & Phasenplanung
A21	M2	Aufgabenfindung	Bericht zu den Arbeiten während der Inception-Phase, Beschreibung wie zu verwendende Technologie gewählt wurde
A22	M2	Guidelines	Quellcode- und Testing-Guidelines
A23	M2	Anforderungsanalyse	Funktionale & Nichtfunktionale Anforderungen, Use Cases
A24	M2	Domainmodel	Analyse der Problemdomäne
A25	M2	SAD	Beschreibung der angestrebten Architektur für die Beispielapplikation.
A26	M2	Architekturprototyp	Exemplarische Implementierung der angestrebten Technologie/Architektur <i>Typ: Quellcode/Applikation</i>
A80	M8	Quellcode Paket	Quellcode der Beispielapplikation zum eigenen, spezifischen Deployment. Bereits zur Weiterentwicklung. <i>Typ: Quellcode</i>
A81	M8	Vagrant Paket	VM-Image mit lauffähiger Version der Beispielapplikation. <i>Typ: Vagrant Image</i>
A82	M8	Heroku Paket	Beispielapplikation ist so vorbereitet, dass ein Deployment auf Heroku problemlos möglich ist. <i>Typ: Quellcode</i>
A83	M8	Installationsanleitung	Anleitung wie die verschiedenen Deploymentpakete (Artefakte A80-82) eingesetzt/installiert werden können.
A100	M10	A0-Poster	Gem. HSR Vorgaben zu erstellendes Poster mit Übersicht zu dieser Bachelorarbeit.
A101	M10	Kurzfassung	Gem. HSR Vorgaben zu erstellende Kurzfassung dieser Bachelorarbeit.
A102	M10	Dokumentation	Alle bisherigen Dokumentationsartefakte zusammengefasst in einem Bericht. Wo nötig, sind entsprechende Kapitel dem Projektablauf entsprechend nachgeführt (bspw. A25 SAD etc.)

Tabelle 2.4.: Abzugebende Artefakte

Kapitel 3 **Aufgabenfindung**

Die Aufgabenstellung (siehe Anhang E) verzichtet bewusst auf funktionale Anforderungen an die zu erstellende Applikation. Desweiteren werden auch keine spezifischen Technologien zur Umsetzung vorgegeben.

Diese sehr offene Ausgangssituation wird lediglich durch die folgenden Ansprüche eingegrenzt:

1. Das Produkt soll unter Verwendung einer oder mehreren Internettechnologien konzipiert und umgesetzt werden.
2. Der zu erstellende Quellcode soll *State Of The Art* Architekturprinzipien [ROC] exemplarisch darstellen und der interessierten Fachperson als Anschauungsmaterial dienen können.

Kapitel 4 **Anforderungsanalyse**

4.1. Funktionale Anforderungen

<i>ID</i>	<i>Name</i>	<i>Beschreibung</i>	<i>Priorität</i>
F1	WG erstellen	Die Applikation erlaubt es eine WG zu erstellen.	hoch
F2	Einladung	Die Applikation erlaubt es, einen Benutzer in eine WG einzuladen.	hoch
F3	Aufgabe erstellen	Die Applikation erlaubt es, eine Aufgabe zu erstellen.	hoch
F4	Aufgabe erledigen	Die Applikation erlaubt es, eine Aufgabe zu erledigen.	hoch
F5	Verlassen	Die Applikation erlaubt es, eine WG zu verlassen.	mittel
F6	Aufgabe bearbeiten	Die Applikation erlaubt es, eine Aufgabe zu bearbeiten.	mittel
F7	Rangliste anzeigen	Die Applikation erlaubt es, eine Rangliste für die Bewohner einer WG anzuzeigen.	mittel
F8	Erfolge vergeben	Die Applikation erlaubt es, Erfolge aufgrund von Regeln zu vergeben.	mittel
F9	WG auflösen	Die Applikation erlaubt es, eine WG aufzulösen.	niedrig
F10	Bewohnerverwaltung	Die Applikation erlaubt es, die Bewohner einer WG zu verwalten.	niedrig
F11	Inhalte teilen	Die Applikation erlaubt es, Inhalte auf Social Media Kanälen zu teilen.	niedrig

Tabelle 4.1.: Funktionale Anforderungen

4.2. Nicht-Funktionale Anforderungen

<i>ID</i>	<i>Name</i>	<i>Beschreibung</i>
NF1	Antwortzeit	Die Applikation antwortet bei normalen Anfragen innerhalb von 0.2s.
NF2	Desktop Browserkompatibilität	Die Applikation unterstützt Internet Explorer 8 und höher, Chrome 25 und höher, Firefox 19 und höher sowie Safari 6 und höher.
NF3	Mobile Browserkompatibilität	Die Applikation unterstützt Safari 6.0 und Android Browser 4.0.
NF4	Sicherheit	Die Applikation kontrolliert den Zugriff auf geschützte Ressourcen.
NF5	ROCA Prinzipien	Die Applikation entspricht den ROCA [ROC] Prinzipien.

Tabelle 4.2.: Nicht-Funktionale Anforderungen

4.3. Use Cases

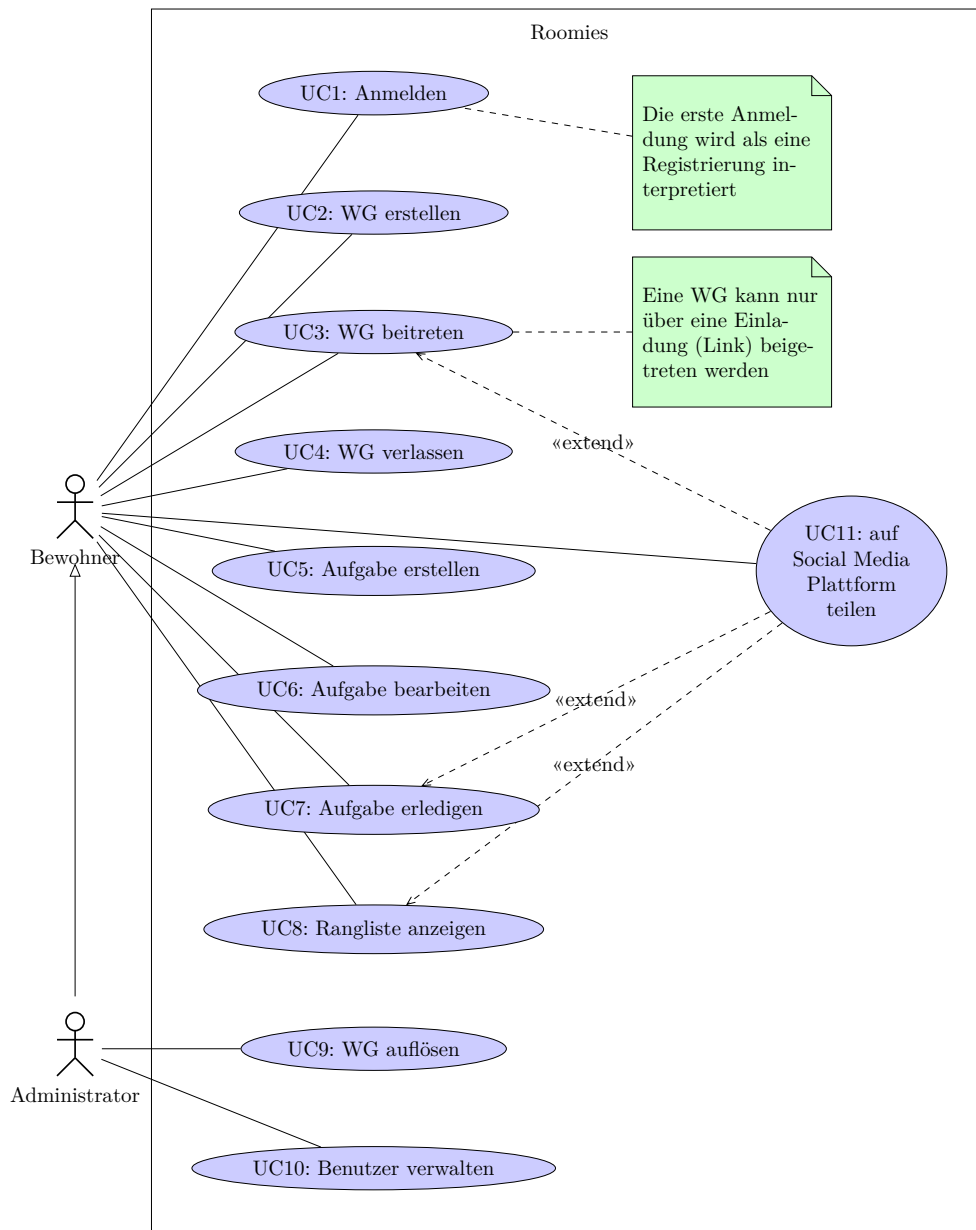


Abbildung 4.1.: Use Case Diagramm

4.3.1. UC1: Anmelden

Use Case Name	UC1: Anmelden
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte sich in das System anmelden.
Preconditions	<ul style="list-style-type: none">▪ Der Bewohner besitzt einen Account bei Facebook.
Postconditions	<ul style="list-style-type: none">▪ Der Bewohner ist in das System eingeloggt.
Main Success Scenario	<ol style="list-style-type: none">1. TODO: login mit FB2. tbd
Extensions	-
Frequency of Occurrence	Oft

Tabelle 4.3.: UC1: Anmelden

4.3.2. UC2: WG erstellen

Use Case Name	UC2: WG erstellen
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte eine WG erstellen.
Preconditions	<ul style="list-style-type: none"> ▪ Der Bewohner ist keiner WG zugeteilt. ▪ Der Bewohner ist in das System angemeldet.
Postconditions	<ul style="list-style-type: none"> ▪ Neue WG ist erstellt. ▪ Der Bewohner ist der WG zugeteilt. ▪ Der Bewohner ist der Administrator der WG.
Main Success Scenario	<ol style="list-style-type: none"> 1. Bewohner klickt auf 'WG erstellen' 2. System zeigt das WG-erstell-Formular an 3. Bewohner trägt der Name der WG ein 4. TODO: weitere infos? 5. Bewohner klickt auf 'erstellen' 6. System zeigt Aufgaben der neu erstellten WG
Extensions	<ol style="list-style-type: none"> 2a. Bewohner gibt keinen Namen an <ol style="list-style-type: none"> 6. System fordert (Fehlermeldung) Bewohner einen Namen für die WG einzugeben
Frequency of Occurence	Selten

Tabelle 4.4.: UC2: WG erstellen

4.3.3. UC3: WG beitreten

Use Case Name	UC3: WG beitreten
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte einer WG beitreten
Preconditions	<ul style="list-style-type: none"> ▪ Der Bewohner besitzt einen Account bei Facebook. ▪ Der Bewohner ist die Internetadresse der Einladung bekannt. ▪ Der Bewohner ist in keine WG aktiv.
Postconditions	<ul style="list-style-type: none"> ▪ Der Bewohner ist in der WG angemeldet ▪ Der Bewohner ist in der WG aktiv.
Main Success Scenario	<ol style="list-style-type: none"> 1. System zeigt Einladungsformular 2. Bewohner klickt auf 'WG beitreten' 3. System zeigt Aufgaben der WG
Extensions	-
Frequency of Occurrence	Gelegentlich

Tabelle 4.5.: UC3: WG beitreten

4.3.4. UC4: WG verlassen

Use Case Name	UC4: WG verlassen
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte einer WG beitreten
Preconditions	<ul style="list-style-type: none"> Der Bewohner ist kein Administrator der WG.
Postconditions	<ul style="list-style-type: none"> Der Bewohner ist in der WG inaktiv
Main Success Scenario	<ol style="list-style-type: none"> Bewohner klickt auf 'Einstellungen' System zeigt die Einstellungen vom Bewohner Bewohner klickt auf 'WG verlassen' System fordert eine Bestätigung des Bewohners Bewohner bestätigt das Verlassen System zeigt 'Auf Wiedersehen' Meldung
Extensions	<ol style="list-style-type: none"> 5a. Bewohner verneigt das Verlassen <ol style="list-style-type: none"> 6 System zeigt die Einstellungen vom Bewohner
Frequency of Occurence	Gelegentlich

Tabelle 4.6.: UC4: WG verlassen

4.3.5. UC5: Aufgabe erstellen

Use Case Name	UC5: Aufgabe erstellen
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte eine Aufgabe der WG hinzufügen
Preconditions	<ul style="list-style-type: none"> ▪ Der Bewohner ist angemeldet
Postconditions	<ul style="list-style-type: none"> ▪ Die Aufgabe wurde erstellt
Main Success Scenario	<ol style="list-style-type: none"> 1. Bewohner klickt auf 'Aufgabe hinzufügen' 2. System zeigt 'Neue Aufgabe'-Formular 3. Bewohner gibt Name der Aufgabe ein. 4. //TODO: fill some more data? 5. Bewohner klickt auf 'Speichern' 6. System zeigt Erfolgsmeldung
Extensions	<ol style="list-style-type: none"> 5a. Bewohner klickt auf "
Frequency of Occurrence	Oft

Tabelle 4.7.: UC5: Aufgabe erstellen

4.3.6. UC6: Aufgabe bearbeiten

Use Case Name	UC6: Aufgabe bearbeiten
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte eine Aufgabe der WG bearbeiten
Preconditions	<ul style="list-style-type: none"> ▪ Der Bewohner ist angemeldet ▪ Aufgabe ist vorhanden ▪ Aufgabe ist nicht als 'Erledigt' markiert
Postconditions	<ul style="list-style-type: none"> ▪ Geänderte Properties wurden gespeichert
Main Success Scenario	<ol style="list-style-type: none"> 1. Bewohner wählt auf Aufgabe aus 2. System zeigt die Eigenschaften der Aufgabe an 3. Bewohner verändert die Eigenschften 4. Bewohner klickt auf 'Speichern' 5. System zeigt Erfolgsmeldung
Extensions	-
Frequency of Occurence	Oft

Tabelle 4.8.: UC6: Aufgabe bearbeiten

4.3.7. UC7: Aufgabe erledigen

Use Case Name	UC7: Aufgabe erledigen
Scope	Roomies
Primary Actor	Bewohner
Stakeholders and Interest	Der Bewohner möchte eine Aufgabe der WG als 'erledigt' markieren.
Preconditions	<ul style="list-style-type: none"> ▪ Der Bewohner ist angemeldet ▪ Aufgabe ist vorhanden
Postconditions	<ul style="list-style-type: none"> ▪ Die Aufgabe ist als 'erledigt' markiert.
Main Success Scenario	<ol style="list-style-type: none"> 1. Bewohner klickt auf 'Erledigt' bei einer ausgewählten Aufgabe 2. System blendet die Aufgabe aus der Aufgabenliste aus
Extensions	-
Frequency of Occurrence	Oft

Tabelle 4.9.: UC7: Aufgabe erledigen

Kapitel 5 Domainmodell

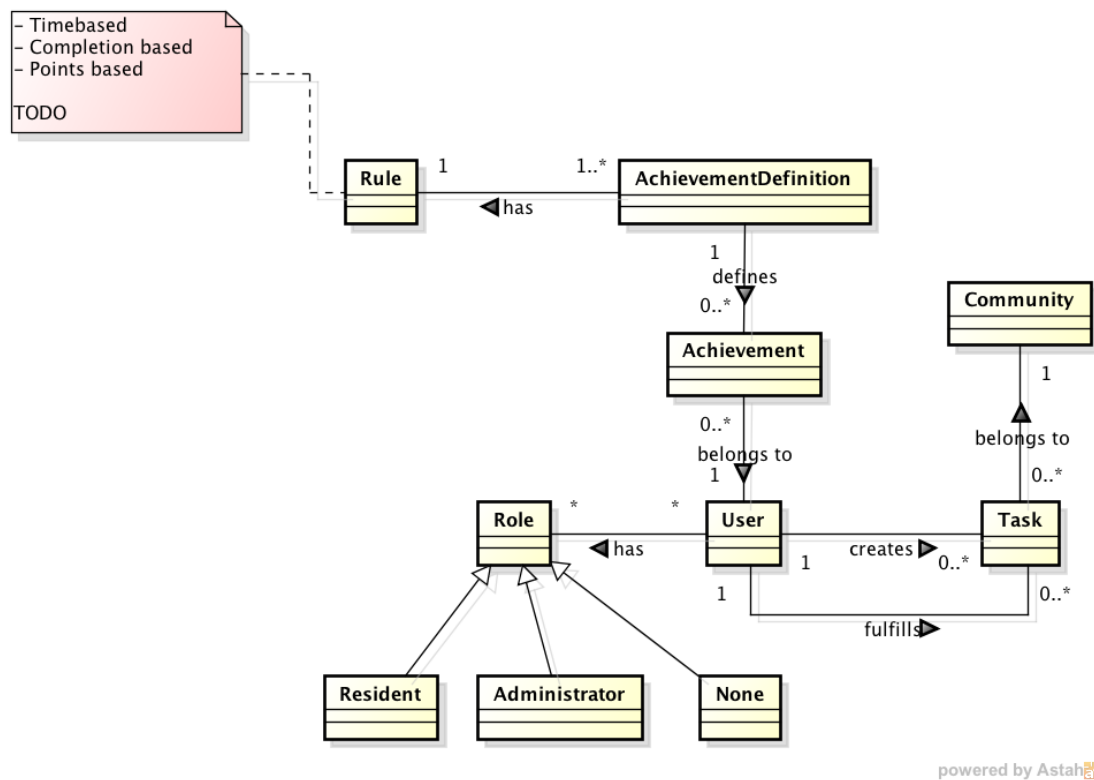


Abbildung 5.1.: Domain Modell

Kapitel 6 **Technische Architektur**

Anhang A **Abbildungen, Tabellen & Quellcodes**

Abbildungsverzeichnis

2.1. Phasenübersicht mit Meilensteinen, Kalenderwochen Februar bis Juli 2013	10
4.1. Use Case Diagramm	16
5.1. Domain Modell	24

Tabellenverzeichnis

2.1. Projektrelevante URL's	7
2.2. Projektphasenbeschreibung	9
2.3. Meilensteine	11
2.4. Abzugebende Artefakte	12
4.1. Funktionale Anforderungen	14
4.2. Nicht-Funktionale Anforderungen	15
4.3. UC1: Anmelden	17
4.4. UC2: WG erstellen	18
4.5. UC3: WG beitreten	19
4.6. UC4: WG verlassen	20
4.7. UC5: Aufgabe erstellen	21
4.8. UC6: Aufgabe bearbeiten	22
4.9. UC7: Aufgabe erledigen	23

Quellcodeverzeichnis

Anhang B **Literatur**

- [Has] HashiCorp. *Vagrant*. URL: <http://www.vagrantup.com/> (besucht am 06.03.2013).
- [ROC] ROCA. *Resource-oriented Client Architecture*. URL: <http://roca-style.org> (besucht am 06.03.2013).

Anhang C **Glossar**

Benutzer

Ein Benutzer TOOODOOO. 14

Bewohner

Ein Bewohner TOOODOOO. 14

CI

Continous Integration. 7

IDE

Integrated Development Environment. 7

Push-Hook

In *git* bezeichnet ein Push-Hook ein Script, welches nach jedem Commit ausgeführt wird.. 8

RUP

Rational Unified Process; Iteratives Projektvorgehen. 9

SAD

Software Architektur Dokument. 11, 12

VM

Virtual Machine. 12

WG

Eine WG TOOODOOO. 14

Anhang D **Meetingprotokolle**

Bachelorarbeit Vorbesprechung 14. Februar 2013

Teilnehmer

- Hans Rudin, HRU (HSR)
- Kevin Gaunt, KGA (HSR)
- Daniel Hildebrand, DHI (Crealogix)
- Manuel Alabor, MAL (Team)
- Alexandre Joly, AJO (Team)
- Michael Weibel, MWE (Team, Protokoll)

Traktanden

- StoryboardBuilder - Was ist der aktuelle Stand?
- Oder was habt ihr für Ideen?

Meeting

StoryboardBuilder

Einführung DHI

- Ende Januar Entscheid: Nicht weiterentwickeln
- Allerdings nicht weil Produkt/Markt nicht interessant wäre
- Ziel war: damit die Crealogix UX-Services zu unterstützen
- Crealogix wird keine UX-Services gegen aussen mehr anbieten
 - mehr interne Projekte betreuen
- Zwei Schwerpunkte: Education & Financial Services

- StoryboardBuilder gehört nicht in einen solchen Schwerpunkt
- Dies obwohl gutes Potential gesehen wird für das Produkt
- Anforderungsspezifikation verfeinert bis Ende Januar
- Technischer Prototyp gestartet, aber wieder gestoppt aufgrund der Neuorientierung
- Idee wäre: Storyboardbuilder an externe Firma weitergeben
- Bestehende Mitbewerber bauen ihre Angebote aus

Diskussion BA

- Frage an die Runde: ist es interessant für euch, den Storyboardbuilder in der BA weiter zuentwickeln?
 - MAL: Wie würde das aussehen?
 - DHI: Beispiel aufgrund gewählter Technologie zu entwickeln
 - DHI: Auf Basis der fachlichen Spezifikation der Crealogix
 - HRU: BA sollte nicht zu einer Fleissarbeit werden
 - HRU: Was wären denn die Herausforderungen wenn das weiterentwickelt werden würde in BA?
 - HRU: Die momentane Ausgangslage ist anders, da kein wirklicher Kunde existiert
 - DHI: Es sind sicher einige Ideen da, die technisch Herausfordernd sind
 - DHI: Grafische Repräsentation auf Screen, Objektmodell umsetzen
 - DHI: Wie gesagt, hat auch nichts dagegen, wenn eine neue Arbeit gemacht werden würde
 - DHI: Laut Kenntnisstand von DHI sollte auch von seiten Industriepartner weniger Betreuung beinhalten, mehr von Team
- KGA: Was ist nun der Anspruch an das Meeting? Müssen wir am Ende des Meetings schon wissen was gemacht werden soll?
 - DHI: Ist offen, hat keinen Anspruch auf Entscheid jetzt - sollte aber bald geschehen, da Bachelorarbeit bald startet
 - DHI:

- DHI: Hat div. Alternativen die man anschauen könnte
- MAL: für ihn ist das Durchführen der BA mit dem Storyboardbuilder nicht mehr besonders interessant, aufgrund Änderung seitens Crealogix
 - DHI: Ja das stimmt, BA würde nicht mehr zu einem realen Produkt führen
- MWE: gehts ähnlich wie MAL
- DHI: Thema 1:
 - **Kino reservations system**
 - Kennt Kinobesitzer in rapperswil
 - Buchungssystem
 - Mit anbindung an Kassensystem
 - nicht nur einzelne Plätze
 - sondern auch Firmenanlässe, Frauenkino, Catering etc.
 - Konzeptionell entwerfen
 - soweit wie möglich implementieren
 - könnte sehr interessant sein, DHI's Meinung nach
- DHI: Thema 2
 - **Personal Finance Management**
 - Zusatz zu Ebanking
 - Eigene Zahlungen analysieren
 - Verschiedene Banken
 - Soviel für Versicherung, Einkauf, etc.
 - Basiert auf einer isländisch-schwedischen Firma
 - Integrationsarbeit (in ein Bankensystem einbauen)
 - MAL: Geht es darum, das zu integrieren und anschaulich darzustellen, und Data-Mining ist schon gemacht?
 - DHI: Ja
 - Schwierigkeit Sicherheit
 - Zertifizierung, Authentifizierung
 - nur Teilaspekt möglich zum lösen
 - Versch. Komponenten
 - Aspekt wichtig auf welcher sich die BA konzentrieren soll
 - DHI: Müsste das genauer anschauen wie das machbar wäre
 - Da es ein sehr grosses System wäre
 - Müsste verifizieren ob das gehen soll?

- HRU: Thema 2 wohl eher interessant (Team bejaht)
 - HRU: Wäre das so kurzfristig machbar?
 - DHI: müsste angeschaut werden
 - DHI: Verträge bestehen
 - DHI: anderes ist Ebanking in Java mit Oracle
 - DHI: Verfügbar machen möglich
 - HRU: Zwei Komponenten, was ist Webfrontend?
 - DHI: Netty server von airlock für authentifizierung
 - DHI: möglichst HTML das übers web geht
 - DHI: J2EE - JSP vorne
 - HRU: Isländische Software
 - DHI: .NET basiert
 - DHI: hat aber ein WCF/REST/AJAX schnittstelle welche relativ gut ins Frontend integrierbar wäre
 - DHI: von einem System ins andere System transferieren (Oracle zu MSSQL DB)
 - DHI: Statistisch aufwerten, und wieder anzeigen
 - DHI: machbarkeit unklar, muss verifiziert werden
- MAL: Was hat HRU für Projekte
 - MAL: Realtime sachen wären interessant (Mobile, Chat, Messaging?)
 - HRU: hat keine Projekte
- DHI: Hat evtl. noch andere Projekte, müsste das aber noch anschauen
- MAL: Bis Testumgebung steht würden wohl wochen vergehen
 - DHI: stimmt wohl
- MWE: Persönlich interessiert vorallem WebRTC
- HRU: was ist mit web realtimecommunication (WebRTC) gemeint?
- MWE: Near-realtime communication (Daten, Video, Audio) zw. Browsern
- MWE: Was wäre denn für Crealogix interessant - zentrale Frage?
 - DHI: Crealogix muss nicht unbedingt dabei sein, wenn nicht nötig
- DHI: update maus-scanner
 - MAL: wäre denn mobile auch ein Thema?
 - DHI: Mobile ist sehr zentral

- HRU: gibt es fraktionen (web/mobile) im Team?
 - MWE: Web-Mensch, aber Mobile wäre auch sehr interessant
- DHI: Fragt bei Crealogix CEO/entwicklungsleiter nach bzgl. Mobile
- MAL: Elearning wäre auch interessant bzgl. Mobile
 - DHI: könnte nachfragen obs da auch was geben würde?
- AJO: Auch vorallem Mobile interessant
 - arbeitet auch vorallem in Mobile
 - HRU: Auch sie MAL ;)
- DHI: müsste nachfragen, aber wäre sicher interessant
- MAL: Wäre sicher interessant auf DHI's Themen zu warten, andererseits müsste auch Teamintern bzw. mit KGA/HRU geschaut werden
- DHI: wann beginnt die Arbeit? - Montag
- DHI: 3 Bereiche Education
 - Campusmanagement
 - Time to learn
 - 30'000 die mit TTL arbeiten
 - Mit Center for young professional zusammenarbeit (evtl. da was interssantes)
 - In Bubikon
- DHI: was machen wenn nichts herauskommt?
 - DHI: würden gerne mit euch zusammenarbeiten, wenn möglich
- HRU: Wäre schon der Weg zum gehen
 - parallel müssten überlegungen angestellt werden obs was anderes geben würde
 - gibt den 3 Studierenden möglichst freie Hand
- KGA: Termin bis wann die Entscheidung möglich sein
 - HRU: allerspätistens erste Woche
 - DHI: wird noch heute mit den 3 Crealogix leuten schauen
 - DHI: Bis morgen, 15.02. Antwort wenn möglich
 - MAL: Mittwoch zu spät oder zu früh?
 - HRU: Wann sind sie @HSR?
 - MAL: MO/DI/MI

- HRU: Mittwoch wäre nicht unbedingt zu spät
- MAL: Mittwoch wäre Zusammenkunft, um definitiv zu entscheiden.
Aber mit Kommunikation bis dann
- MAL: DHI kann sicher schnell entscheiden, je nach dem wäre pers.
Anwesenheit nicht nötig
- DHI: würde gerne persönlich dabei sein
- DHI: gibt morgen Feedback
- HRU: das ist gut - in der Runde Team/HSR diskutieren was gemacht werden kann
- KGA: Was wird mit UX passieren? (Expert Talks)
 - DHI: Prio 1 hat interne Aufträge und Prio 2 externe

Diskussion im Team

- KGA: also ist keines der beiden Thema sehr interessant für Team?
- Team: Ja, insbesondere auch zu gross für die kurze Zeit (Thema 2)
- HRU: Reservationssystem vor allem Businessanalyse
- HRU: Sicher gut zu schauen was DHI einbringt
- HRU: Aber auch schauen was wir für Ideen haben
- KGA: was wäre ursprüngliche Idee für SA gewesen
- MWE: XMPP Server in node.js modular, flexibel bzgl.
Datenbankanbindung
- MAL: und auch Skalierbarkeit von node.js interessant
- MAL: interessant wäre vielleicht frontend framework
- HRU: Alle miteinander auf dem Laufenden halten bzgl. Ideen

Nächstes Meeting

- Mittwoch, 20. Februar 2013, 10:10 Uhr

Anhang E **Aufgabenstellung**

Die folgenden drei Seiten enthalten die offizielle Aufgabenstellung dieser Bachelorarbeit.

Aufgabenstellung Bachelorarbeit für Manuel Alabor, Alexandre Joly und Michael Weibel „Architekturkonzepte moderner Web-Applikationen“

1. Auftraggeber, Betreuer und Experte

Bei dieser Arbeit handelt es sich um eine HSR-interne Arbeit zur Unterstützung des Moduls Internettechnologien.

Auftraggeber/Betreuer:

- Prof. Hans Rudin, HSR, IFS hрудin@hsr.ch +41 55 222 49 36 (Verantw. Dozent, Betreuer)
- Kevin Gaunt, HSR, IFS kgaunt@hsr.ch +41 55 222 4662 (Betreuer)

Experte:

- Daniel Hildebrand, Crealogix

2. Studierende

Diese Arbeit wird als Bachelorarbeit an der Abteilung Informatik durchgeführt von

- Manuel Alabor malabor@hsr.ch
- Alexandre Joly ajoly@hsr.ch
- Michael Weibel mweibel@hsr.ch

3. Ausgangslage

Das Modul Internettechnologien ist stark Technologie-zentriert. Wünschbar ist eine Weiterentwicklung (Aktualisierung, Verbesserung) mit vermehrter Beachtung von konzeptionellen und Architektur-Fragen. In letzter Zeit haben sich Prinzipien und Konzepte herauskristallisiert, nach denen Web-Applikationen am besten aufgebaut werden. Siehe zum Beispiel [1] oder [2]. Um diese Prinzipien und Konzepte anschaulich zu vermitteln, braucht es neben Erläuterungen möglichst anschauliche Beispiele und Übungsaufgaben. Ziel dieser Arbeit ist es, die Weiterentwicklung des Moduls Internettechnologien entsprechend zu unterstützen.

4. Aufgabenstellung

In dieser Arbeit sollen die in [1], [2] und weiteren Quellen dargestellten Prinzipien und Konzepte analysiert werden. Gemeinsam mit dem Betreuer sollen daraus in das Modul Internettechnologien zu transferierende Prinzipien und Konzepte ausgewählt werden, und es soll überlegt werden, wie diese Inhalte anschaulich für den Unterricht aufbereitet werden können. In der Folge sollten entsprechende Resultate erarbeitet werden, welche das Unterrichten der ausgewählten Inhalte möglichst gut unterstützen. Eine wichtige Rolle dürfte dabei eine anschauliche Beispielapplikation bilden.

Details werden im Verlauf der Arbeit zwischen Studierenden und Betreuer vereinbart.

5. Zur Durchführung

Mit dem Betreuer finden wöchentliche Besprechungen statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen.

Alle Besprechungen sind von den Studierenden mit einer Traktandenliste vorzubereiten, die Besprechung ist durch die Studierenden zu leiten und die Ergebnisse sind in einem Protokoll festzuhalten, das den Betreuern und dem Auftraggeber per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsresultate in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsresultate erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Dokumentation.

6. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen (siehe <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html>). Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Alle Resultate sind vollständig auf CD/DVD in 3 Exemplaren abzugeben. Der Bericht ist ausgedruckt in doppelter Ausführung abzugeben.

7. Referenzen

- [1] Stefan Tilkov
Building large web-based systems: 10 Recommendations
Präsentation an der OOP 2013, München
PDF als Beilage
- [2] <http://roca-style.org>
ROCA Resource-oriented Client Architecture - A collection of simple recommendations for decent Web application frontends

8. Termine

Siehe auch Terminplan auf <https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html>.

Montag, den 18. Februar 2013	Beginn der Bachelorarbeit, Ausgabe der Aufgabenstellung durch die Betreuer
7. Juni 2013	Abgabe Kurzbeschreibung und A0-Poster. Vorlagen stehen unter den allgemeinen Infos Diplom-, Bachelor- und Studienarbeiten zur Verfügung.
14. Juni 2013, 12:00	Abgabe der Arbeit an den Betreuer bis 12.00 Uhr. Fertigstellung des A0-Posters bis 12.00 Uhr. Abgabe der Posters im Abteilungssekretariat 6.113.
14. Juni 2012	HSR-Forum, Vorträge und Präsentation der Bachelor- und Diplomarbeiten, 16 bis 20 Uhr
5.8. - 23.08.2013	Mündliche Prüfung zur Bachelorarbeit

9. Beurteilung

Eine erfolgreiche Bachelorarbeit zählt 12 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von 30 Stunden budgetiert (Siehe auch Modulbeschreibung der Bachelorarbeit https://unterricht.hsr.ch/staticWeb/allModules/19419_M_BAI.html).

Für die Beurteilung ist der HSR-Betreuer verantwortlich.

Gesichtspunkt	Gewicht
1. Organisation, Durchführung	1/6
2. Berichte (Abstract, Mgmt Summary, technischer u. persönliche Berichte) sowie Gliederung, Darstellung, Sprache der gesamten Dokumentation	1/6
3. Inhalt*)	3/6
4. Mündliche Prüfung zur Bachelorarbeit	1/6

*) Die Unterteilung und Gewichtung von 3. Inhalt wird im Laufe dieser Arbeit mit den Studierenden festgelegt.

Im Übrigen gelten die Bestimmungen der Abteilung Informatik für Bachelorarbeiten.

Rapperswil, den 20. Februar 2013



Prof. Hans Rudin
 Institut für Software
 Hochschule für Technik Rapperswil