

CS Seminar Registration

Im Modul *Workshop Distributed Software Systems* werden die Themen der Vertiefungsrichtung *Verteilte Software Systeme* in einem durchgehenden Beispiel angewendet und umgesetzt. Dieses Dokument beschreibt die Aufgabenstellung für die Durchführung 2017.

1. Problembeschreibung

In diesem Jahr soll im Rahmen des Moduls DSS Workshop eine Plattform entwickelt werden, mit der die im Studiengang Informatik durchgeführten Seminar-Anlässe verwaltet werden können. Dazu gehören die Ankündigung der Vorträge und das Registrieren der Teilnehmenden. Früher wurde für diese Aufgabe Doodle verwendet, aber dieses System konnte leicht missbraucht werden (Registrierung von Phantasie-Teilnehmern), daher werden die Anmeldungen aktuell über ein Excel-Sheet erfasst, welches auf Inside bereitgestellt ist. Der Anmeldeprozess ist jedoch ziemlich kompliziert:

Anmeldungsschritte:

https://groups.inside.fhnw.ch/workspace_sites/I_CSSeminar/Dokumente/Forms/AllItems.aspx

1. Ordner "Dokumente" öffnen
2. Dokument "CS-Seminar-Anmeldungen" öffnen
3. Im Excel, oben links, "Edit work book" anklicken
4. Im Menue "Edit in Excel Webapp" anklicken und Namen eintragen (Inhalt wird automatisch gespeichert)
5. Ausloggen mit x anklicken

Falls Sie ein Login-Problem haben melden Sie sich bitte bei markus.oehninger@fhnw.ch

Ziel ist es nun, eine neue kleine Plattform zu entwickeln, die speziell für die Seminaranmeldung genutzt werden kann und die bisherigen Lösungen (Doodle und Inside) ersetzt.

2. Zielsetzung

Das Ziel dieser Arbeit liegt darin, ein relativ einfaches Geschäftsmodell mit Hilfe von Internet-Technologien komponentenbasiert abzubilden. Wir wollen uns dabei auf die im Unterricht vorgestellten Plattformen konzentrieren. Für das GUI verwenden Sie einen Web-Container (Servlets; JSP, JSF oder RIA mit REST) und für die Geschäftslogik Spring Boot (mit Spring-Data). Alternativ könnte die Anwendung auch auf der Basis der Java Enterprise Edition (JEE) entwickelt werden.

Da die Aufgabe ziemlich umfangreich ist, soll sie in Gruppen zu 3 Studierenden bearbeitet werden. Die Arbeit wird benotet und ergibt die Modulnote.

3. Anforderungsbeschreibung

Funktionale Anforderungen

Ein Seminarvortrag hat folgende Eigenschaften:

- Datum / Zeit
- Raum
- Referent (intern oder extern)
- Titel und Abstract
- Folien
- Link mit weiterführenden Infos (z.B. <https://www.youtube.com/watch?v=Tyd0FO0tko8>)

Typischerweise vor dem Start des Semesters (aber bei gewissen Vorträgen auch während des Semesters) werden die Seminarvorträge von einem Koordinator erfasst. Daraufhin soll der Referent selbständig den Titel und das Abstract erfassen bzw. ändern können, und nach dem Vortrag soll er auch die Folien hochladen können.

Etwa zwei Wochen vor dem Vortrag werden die Studierenden vom Studiengangleiter (SGL) per Email zum Seminarvortrag eingeladen (es kann mehrere Personen mit dieser Rolle geben). Diese Einladung soll ebenfalls über diese Plattform verschickt werden können. Die Studierenden registrieren sich dann für den Vortrag und geben an, ob sie ein Fleisch- oder Vegi-Sandwich und ob sie ein Getränk wünschen. Die Studierenden können sich bis zu einem bestimmten Zeitpunkt anmelden, bei einem Vortrag am Montag typischerweise bis am Freitag davor, 12 Uhr (diese Deadline wird bereits mit der Einladung den Studierenden mitgeteilt).

Wenn diese Anmeldefrist abgelaufen ist, dann werden vom System folgende Emails verschickt:

- Information an den CBW Anlasskoordination (raumreservation.windisch@fhnw.ch) mit der Angabe der Anzahl Studierenden.

Am CS-Seminar vom <dd.mm.yyyy> hat es <n> Teilnehmende.

[Falls Referent extern]

Um <Zeit-15Min> sollte jemand zur Verfügung stehen, um dem externen Referenten beim Bedienen des Beamers behilflich zu sein.

[Falls Anzahl Teilnehmer > 40]

Bitte Palette mit genügend Stühlen im Gang bereitstellen.

Mit freundlichen Grüßen

<Koordinator>

- Information an die SV-Group mit der Angabe der Anzahl Getränke sowie der Anzahl Fleisch- und Vegi-Sandwiches.

Am CS-Seminar vom <dd.mm.yyyy> haben sich <n> Teilnehmende angemeldet.

Wir bitten Sie, um <Zeit-15Min> im Raum <Raum> folgende Verpflegung bereitzustellen:

<g> Getränke (5dl PET-Flaschen)

<v> Vegi-Sandwich

<f> Fleisch-Sandwich.

Mit freundlichen Grüßen

<Koordinator>

- Email an den Referenten mit der Information, wieviele Teilnehmer erwartet werden.

Am CS-Seminar vom <dd.mm.yyyy> haben sich <n> Teilnehmende angemeldet. Das CS-Seminar findet im Raum <Raum> statt.

[Falls Referent extern]

Um <Zeit-15Min> Uhr ist jemand vom Hausdienst da, der Ihnen wenn nötig behilflich ist, den Beamer einzurichten.

Wir freuen uns auf Ihr Referat mit Beginn um <Zeit> zum Thema <Titel>. Nach dem Vortrag bitten wir Sie, die Folien über den Link <URL> auf unsere Plattform hochzuladen.

Mit freundlichen Grüßen

<Koordinator>

Die Empfänger dieser Emails sollen über ein Property-File definiert werden, d.h. es ist nicht nötig, diese über eine Webseite editieren zu können. Für die SV-Group sind dies aktuell die Adressen

- Nicole.Hinden@sv-group.ch
- Karin.Huesler@sv-group.ch.

Die Email soll dabei an beide (bzw. generell an alle erfassten Empfänger einer Kategorie) geschickt werden. Das gilt auch für den Hausdienst, wobei dort aktuell die Information nur an die eine Adresse

- raumreservation.windisch@fhnw.ch

verschickt werden muss.

Der SGL wird die Einladung zum Vortrag verschicken, und zwar an alle Mitarbeitenden der Informatik-Institute sowie an alle Studierenden (sowie Cc an die SV-Group Ansprechpersonen). Der Text dazu könnte wie folgt aussehen:

Liebe Studierende der Informatik/iCompetence und liebe Institutsangehörige,

Am nächsten <Datum> findet um <Zeit> das CS-Seminar zum Thema <Titel> im <Ort> statt.

Bitte melden Sie sich für diesen Anlass bis <Deadline> über folgenden Link an:
<Link>

Beste Grüsse
Christoph Denzler

All diese Texte sollen als Templates auf der Plattform abgelegt sein. Es muss jedoch keine Webseite bereitgestellt werden, über welche diese Templates geändert werden können. Wenn der SGL die Einladung verschickt, so soll der generierte Einladungstext (sowie die Liste der To: und der Cc: Empfänger) editiert werden können.

Die Einladung wird zusätzlich auch an registrierte Nutzer verschickt (typischerweise externe Personen, die ebenfalls am Seminar interessiert sind). Diese Personen werden vom Koordinator erfasst, sie können sich jedoch selbständig abmelden wenn sie nicht mehr an weiteren Seminar-Ankündigungen interessiert sind (drop out).

Um auf der Plattform aktiv zu werden, muss sich ein Benutzer anmelden. Studierende melden sich dabei mit ihrer EMAIL-Adresse und ihrem AD-Passwort an. Externe Personen (Referenten und externe Teilnehmer) müssen vom Administrator erfasst werden, können ihr Passwort jedoch selber zurücksetzen lassen.

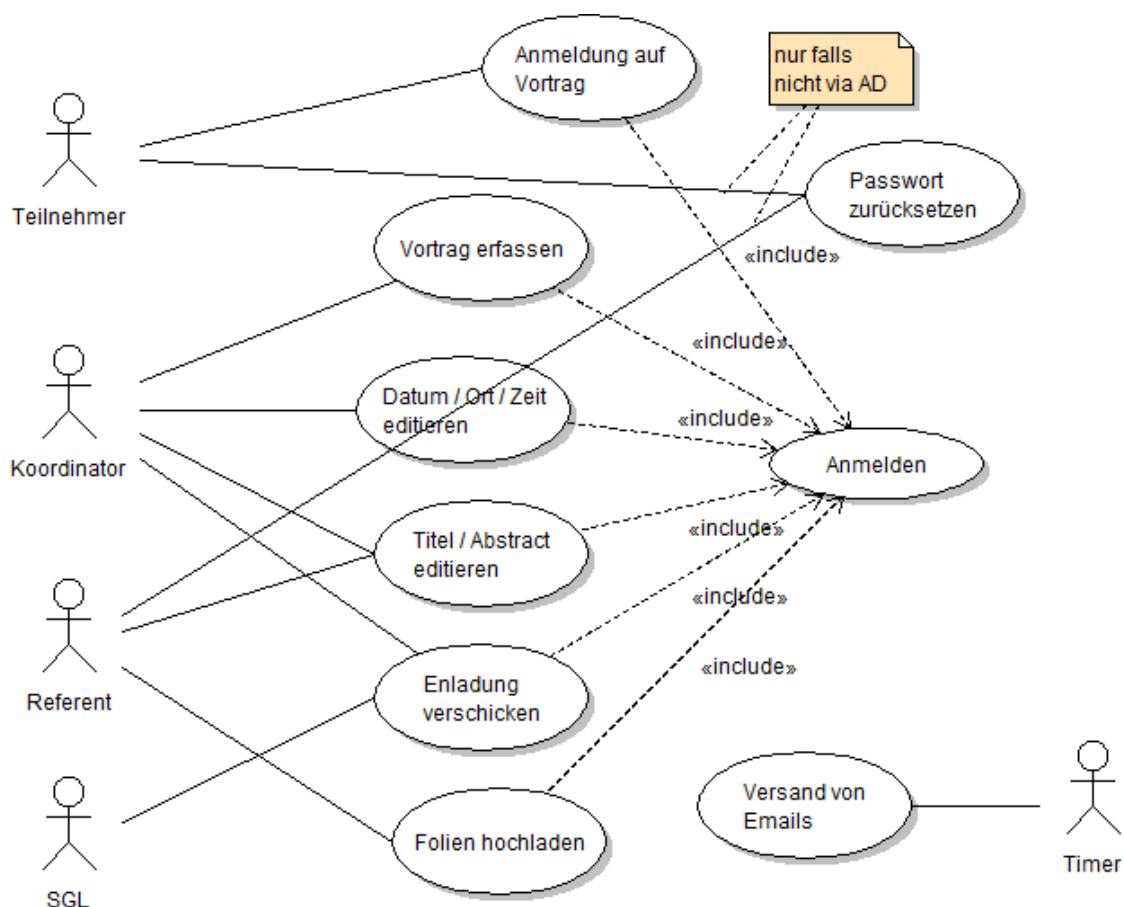


Abbildung 1: Use Case Diagramm

Folgende Rollen sind vorgesehen:

Referent

Kann sich auf der Plattform anmelden und kann die Angaben zu seinem Vortrag (Titel und Abstract) bearbeiten. Zudem kann er seine Folien auf die Plattform hochladen. Beim Referenten kann es sich auch um einen FHNW-Mitarbeiter oder um einen Studierenden halten. Nach Ablauf der Anmeldungs-Deadline erhält er eine E-Mail mit der Angabe der Anzahl Teilnehmer.

Teilnehmer

FHNW-Mitarbeiter und Studierende sowie externe Teilnehmer können sich für einen Vortrag einschreiben und können angeben, ob sie ein Getränk und ein Vegi- oder Fleisch-Sandwich wünschen. Bis zum Ablauf der Anmeldefrist können Sie diese Angaben einsehen und auch wieder ändern (d.h. sie können so insbesondere prüfen, ob sie sich bereits angemeldet haben). Wir überlassen es ihnen ob man sich auf mehrere unterschiedliche Vorträge einschreiben kann, oder ob immer höchstens eine Ausschreibung offen ist.

Koordinator

Der Koordinator ist der Administrator der Plattform. Er kann neue Vorträge erfassen bzw. ändern. Er wird auch die externen Teilnehmer erfassen.

SGL

Der Studiengangleiter verschickt die Einladung zu einem Seminarvortrag. Die Einladung kann auch vom Koordinator (Admin) verschickt werden. Eine Einladung kann auch mehrfach verschickt werden, z.B. falls nachträglich noch Zusatzinformationen verschickt werden müssen.

Timer

Ein Timer initiiert den Versand der Emails und ändert den Status der Plattform, d.h. nach Ablauf der Anmeldefrist darf man sich nicht mehr für eine Veranstaltung anmelden.

Nichtfunktionale Anforderungen

Technologie:

- Die Applikation muss sowohl auf Unix-, Mac-OS- und auch auf Windows-basierten Servern eingesetzt werden können.
- Es müssen die modernen Browser-Versionen unterstützt werden (Chrome \geq 54, Firefox \geq 50).
- Es soll ein *Java Application Server* verwendet werden (Spring Boot, Tomcat oder JavaEE)

Benutzerinterface:

- Die Applikation soll ohne Benutzerhandbuch bedienbar sein.
- Die Sprache der Applikation ist Deutsch. Eine Unterstützung für die Internationalisierung der Webpage soll aber vorhanden sein.

Architektur:

- Eine saubere Schichtenarchitektur ist notwendig. Es soll möglich sein, die Serviceschicht und die Serverapplikation später auch über weitere Clients nutzen zu können (z.B. mobile Applikation), ohne dass Änderungen im Business-Layer erforderlich werden.
- Alle relevanten Konfigurationseinstellungen müssen ohne Kompilation der Applikation möglich sein und bei einem Neustart des Servers aktiv werden.

Sicherheit:

- allfällige Passwörter sind weder im Klartext noch (reversibel) verschlüsselt abgelegt
 - o https://owasp.org/index.php/Password_Storage_Cheat_Sheet
 - o https://github.com/Squeng/Apsi/blob/4027e82bf754ca206a3cd49becb8f46a48f17d32/app/value_types/Password.java (als Beispiel)
 - o https://owasp.org/index.php/Forgot_Password_Cheat_Sheet
- das Zugriffskontrollmodell ist adäquat
 - o https://owasp.org/index.php/Access_Control_Cheat_Sheet
- die App weist keine SQL-Injection-Schwachstellen auf
 - o https://owasp.org/index.php/Query_Parameterization_Cheat_Sheet
 - o https://owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet

- die Kommunikation erfolgt einzig über HTTPS
 - o entsprechende HTTP-Response-Headers werden genutzt
 - o Falls die Web-App gegenüber einem Web-Service auch als Client fungiert, siehe http://www.cs.utexas.edu/~shmat/shmat_ccs12.pdf
- weitere HTTP-Response-Headers werden sinnvoll genutzt
 - o for Servlet-based apps: <https://github.com/sourceclear/headlines>
- die App weist keine XSS-Schwachstelle auf
 - o [https://owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](https://owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)
- die App weist keine CSRF-Schwachstelle auf
 - o Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet, [https://owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)_Prevention_Cheat_Sheet](https://owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet)

siehe auch:

- Iron-Clad Java: Building Secure Web Applications <http://www.mhprofessional.com/product.php?isbn=0071835881&cat=112>
- <http://blog.squeng.com/2016/01/24/apsi/>

4. Architektur und Aufgaben

Grundsätzlich soll diese Applikation in mehreren logischen Schichten aufgebaut werden:

- Präsentationsschicht
- Geschäftslogikschicht
- Datenbankschicht

Die Datenbankschicht bildet eine standardisierte Schnittstelle zur Datenbank. Das zugrunde liegende Datenbankmodell wird durch diese Schicht verborgen. Diese Schicht ist in Java mit JPA zu realisieren und der Zugriff soll mit Spring Data erfolgen. Alternativ darf auch JOOQ verwendet werden. Falls Sie eine andere Technologie für die Persistenzschicht verwenden möchten (z.B. eine NoSQL Datenbank) dann fragen Sie kurz ihre Betreuer.

Basierend auf der Datenbankschicht baut die Geschäftslogikschicht auf. In der Geschäftslogik werden die Geschäftstransaktionen abgewickelt. Eine Geschäftstransaktion bildet die Geschäftsregeln ab (z.B. das Erfassen eines Seminarvortrags). Die Geschäftslogik soll auf Basis des Spring Frameworks realisiert werden. Zur Laufzeit soll ein aussagekräftiges Logging die Nutzung der Logik aufzeigen.

Die Präsentationsschicht muss die Daten mit einem Webinterface zur Verfügung stellen. Für die Generierung der HTML Seiten kann JSP oder ein MVC Model2 Framework (wie JSF oder Spring MVC) oder ein Ajax-basiertes Framework (z.B. AngularJS mit REST) verwendet werden.

Für den Versand von Emails (Emailbestätigung, Password-Reset) verwenden Sie das JavaMail API. Um die Fristen für das Schliessen der Anmeldemöglichkeit sowie den Versand der Emails zu realisieren muss mindestens ein Batch-Job vorhanden sein, der im Background eine Aufgabe in einem regelmäßigen Intervall ausführt.

Für die Authentifizierung / Autorisierung auf Ihrer Plattform ist JAAS oder Spring Security zu verwenden.

Für Ihre Geschäftslogik sollen Test-Fälle bereitgestellt werden (JUnit-Tests). Ihr Projekt soll am Ende so bereitgestellt sein, dass wir Ihre Projekte lokal installieren und die Testfälle ausführen können.

5. Organisation

Da die Aufgabe ziemlich umfangreich ist, soll sie in Gruppen zu 2-3 Studierenden bearbeitet werden (fünf Gruppen mit drei, eine mit zwei Studierenden). Die Applikation und die Dokumentation werden benotet und ergeben die Modulnote.

6. Termine

- Mo, 27.02.17 Formierung der Teams (Angabe per email)
17 Studierende => typischerweise 3er Teams (ein 2er Team)
- Mo, 06.03.17 Abgabe eines Projektplans (per Email an die Betreuer)
Angaben zu Grobarchitektur und Rollenzuteilung sowie ein ungefährer Zeitplan
- Mo, 27.03.17 Abgabe folgender Dokumente:
- GUI Vorschlag (auch als Papierprototyp möglich, der Ablauf der Webapplikation soll daraus ersichtlich sein).
 - Statisches Klassendiagramm (Datenbankschema)
 - Beschreibung der Schnittstellen der Business-Logik. Hier neben Syntax auch Semantik angeben!
- Mo, 03.04.17 Zwischenreview, für alle Studierenden
Kurze Präsentation der zu realisierenden Lösung und anschliessende Diskussion mit den Dozierenden.
- Mo, 29.05.17 Abgabe der Lösung: Diese enthält folgende Teile:
- Dokumentation (Beschreibung der Lösung)
 - Code auf CD oder Memory-Stick (inkl. Build Umgebung und allen eingesetzten Bibliotheken)
 - URL auf lauffähige Version
- Mo, 29.05.17 Präsentation der Lösungen (für alle Studierenden)
- Mo, 12.06.17 Verteidigungen / Projektklärungen, gruppenweise (15-18 Uhr)
- Di, 13.06.17 Verteidigungen / Projektklärungen, gruppenweise (15-18 Uhr)

Dokumentation

Als Dokumentation erwarten wir eine Beschreibung Ihrer Lösung. Es sollten alle wichtigen Designentscheidungen, die Sie getroffen haben, beschrieben sein. In der Dokumentation sollen auch interessante Aspekte und Lösungen dargestellt werden und die Dokumentation soll helfen, dass wir uns im Code zurechtzufinden. Ein UML-Diagramm Ihrer Klassen ist dazu geeignet (bzw. zwingend nötig).

Den Code geben Sie bitte auf CD oder Memory-Stick ab. Das Projekt soll von uns einfach installiert und getestet werden können (durch Ausführung der Testfälle).

Ihre Lösung soll auch auf einem Rechner zugreifbar sein, damit wir Ihre Lösung (mind. im FHNW Intranet) anschauen können. Geben Sie im Bericht die URL der Startseite an.

Geben Sie im Bericht auch an, wer in Ihrem Team welche Teile realisiert hat.

Präsentation

In der Präsentation soll auf jene Aspekte hingewiesen werden, welche die anderen Gruppen interessieren könnten, z.B. spezielle Probleme, elegante Lösungen (wie Authentisierung/Autorisierung, GUI Framework,), verwendete Technologien, Erfahrungen, etc.
Die Präsentation soll auch eine Live-Demo des Systems enthalten.

Mögliche Themen der Präsentation sind also:

- Architektur Ihrer Lösung
- Erfahrungen mit Web Framework
- Erfahrungen mit Applikationsserver
- Spezielle Sicherheitsmerkmale, die Sie in Ihrer Applikation berücksichtigt haben

Sie haben für diese Präsentation 20 Min Zeit (inkl. Diskussion)

7. Bewertung

Ihre Arbeit (Dokumentation + Code) wird mit einer Projektnote bewertet. In diese Note fließen folgende Aspekte ein:

- Fachliche Umsetzung: 75%
- Dokumentation: 15%
- Präsentation: 10%

Pro Gruppe gibt es im Normalfall *eine* Projektnote.

Nach der Abgabe der Dokumentation findet eine Klärung statt. In dieser Klärung werden die Dozierenden dem Team spezifische Fragen zur Lösung stellen. Fragen zu allen Themengebieten können an alle gerichtet werden. Diese Klärung dauert 60 Min. Anwesend sind alle am Modul beteiligten Dozierenden und die Mitglieder jedes Teams.

Die Projektnote kann auf Grund der Klärung individuell korrigiert werden.

Der Studierende hat das Modul bestanden, falls

- die individuelle Projektnote genügend ist (≥ 3.8)
- die Testat-Bedingung erfüllt ist, d.h. Anwesenheit bei Zwischenreview, Schlusspräsentation und Verteidigung.

8. Betreuung

Die Dozierenden stehen bei Fragen zur Verfügung:

- Sicherheit: Arno Wagner
- Web-Framework / Architektur: Jürg Luthiger
- Services / ORM: Dominik Gruntz

Wir empfehlen bei Problemen einen Termin zu vereinbaren und vorab das Problem per Email zu schildern.