

# **Logbuch**

**Web-basierte Anwendungen**

**Verteilte Systeme**

**2. Phase**

Daniela Kucharczyk

SS 2013

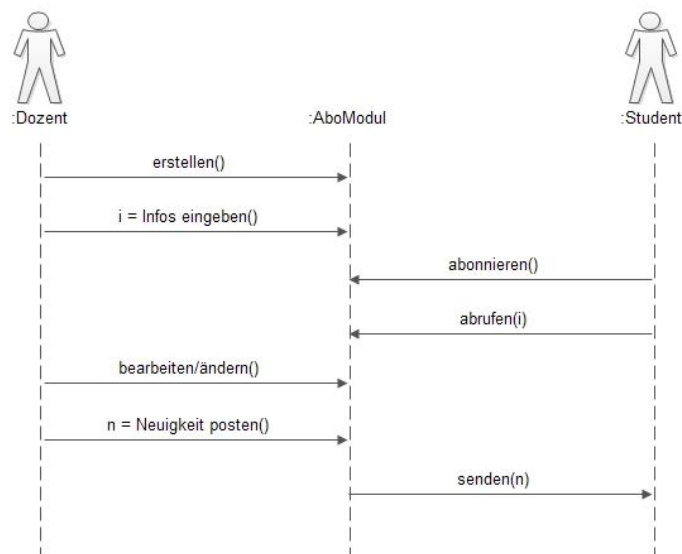
## 1. Projektbezogenes XML Schema / Schemata

Entwicklung und Validierung folgender XML Schemata und XML Dokumenten:

- dozent.xsd, dozent1.xml
- student.xsd, student1.xml
- modul.xsd, modul1.xml
- dozentenliste.xsd, dozentenliste.xml

## 2. Ressourcen und die Semantik der HTTP-Operationen für das Projekt

### Szenario und Interaktion



Dozent

Beschreibung	HTTP Methode	URI
Alle Dozenten laden	GET	/dozenten/
Dozent laden	GET	/dozenten/{id}/
Dozent erstellen	POST	/dozenten/
Dozent aktualisieren	PUT	/dozenten/{id}/
Dozent löschen	DELETE	/dozenten/{id}/
Neuigkeit posten	POST	/dozenten/{id}/news/

Student

Beschreibung	HTTP Methoden	URI
Student erstellen	POST	/student/
Student laden	GET	/student/{id}/
Student aktualisieren	PUT	/student/{id}/
Student löschen	DELETE	/student/{id}/
Abo abschließen	PUT	/student/{id}/
Abo löschen	DELETE	/student/{id}/abos/{id}/
Neuigkeit laden	GET	/news/{id}/
Alle Neuigkeiten laden	GET	/news/

## Modul

Beschreibung	HTTP Methoden	URI
Modul erstellen	POST	/dozenten/{id}/ module /
Modul aktualisieren	PUT	/dozenten/{id}/ module /{id}/
Modul laden	GET	/module/{id}/
Alle Module laden	GET	/ module /
Modul löschen	DELETE	/dozenten/{id}/ module /{id}/
Modul suchen	GET	/ module /
Neuigkeit posten	POST	/dozenten/{id}/module/{id}/news/

### 3. RESTful Webservice

- Hinzufügen weiterer XML Dokumente und Erstellen, der noch fehlenden XML Schemata
- Klasse Ressource.java für das marshalling/unmarshalling
- Implementierung einiger HTTP-Operationen
- Entwicklung eines Testclients für die Ressourcen Dozent und Student, um die HTTP-Methoden über die Konsole zu testen
- Schwierigkeiten beim Pull der Daten des gemeinsamen Projekts über Eclipse
- Rücksprache mit den Mentoren bezüglich der auftretenden Exception bei der Verwendung des Marshallers/Unmarshallers  
--> Entfernen der Namespaces, da sich das Problem nicht anders lösen lässt

### 4. Konzeption + XMPP Server einrichten

- Einrichten des XMPP Servers und Teambesprechung bezüglich der für die Konzeption zu bearbeitenden Punkte

### 5. XMPP - Client

- ConnectionHandler.java : beinhaltet alle Methoden die für den XMPP Client benötigt werden
- ConnectionFrame.java: grafische Oberfläche des Login-Fensters
- ClientFrame: grafische Oberfläche des Hauptfensters
- ItemEventCoordinator.java: Listener für Published Items
- XMPPConfig.java: legt den Hostnamen und die Portnummer fest, die verwendet wird
- XMPPClientMain.java: startet die Anwendung
- Überarbeitung der XML Schemata und Entfernen von Elementen: Ziel dabei ist die XML Daten auf ein Minimum zu reduzieren

### 6. Client - Entwicklung

- Auseinandersetzung mit Swing, um das Layout mit den gewünschten Funktionen umzusetzen
- Durchprobieren der verschiedenen Möglichkeiten, um eine grafische Oberfläche mit Swing zu realisieren

- nach andauernden Problemen bei der Umsetzung mit Swing und einigen Video-Tutorials zu der Verwendung von Windowbuilder später, Start bei null
- mit Hilfe des Windowbuilders schrittweises hinzufügen der grafischen Komponenten und Verknüpfen des Java Codes mit den gewünschten Funktionalitäten
- Fehler beim marshalling/unmarshalling  
--> Grund: fehlende Namespaces
- Überarbeitung der Projektdokumentation und ergänzen von Verbesserungsvorschlägen