

# **Web-basierte Anwendungen 2:**

## **Verteilte Systeme**

Fachhochschule Köln Campus Gummersbach  
Fakultät für Informatik und Ingenieurwissenschaften

### **Phase 2 - Projekt "Social Ticker"**

**Logbuch von Dario Vizzaccaro**

## **Gedanken zum 1. Meilenstein**

Den ersten Meilenstein zur 2. Phase des WBA2 Workshops verbringen mein Teampartner und ich damit, eine Idee zu finden die Interessant und gewissermaßen Innovativ ist. Nach verschiedenen Ideen, die wir gründlich ausdiskutieren, kommen wir auf die Idee eines Chat-basierten Livetickers, der jedem die Möglichkeit bieten soll, auch nur für ein paar Stunden als Admin eines Tickers tätig zu werden. Die Idee finden wir beide sofort bestens geeignet als Projekt, da sie uns verschiedene Möglichkeiten zum Einsatz von synchronem- und asynchronem Datenaustausch bietet. Eine Funktion die unbedingt implementiert werden soll, ist die direkte Kommunikation zwischen verschiedene Nutzern und Admins. Dafür werden „Chatrooms“ verwendet, die jeder Nutzer erstellen kann und denen andere Nutzer anhand einer Liste beitreten können. Nette Zusatzfunktionen wären, Bewertungen abgeben und anderen Nutzer folgen zu können. Noch unklar ist wie das ganze strukturiert werden soll. In Phase 1 haben wir gelernt auf XML residente Daten zu nutzen, aber bei einer viel komplexeren Umgebung werden wir bestimmt genauere Strukturen und Regeln erstellen müssen.

## **Gedanken zum 2. Meilenstein**

Um die Anforderungen des 2. Meilensteins zu meistern habe ich mir das oft empfohlene Buch „REST und HTTP“ von Stefan Tilkov gekauft. Dieses Buch erweist sich als perfekt geeignet für den Einstieg in die Grundlagen des Themas. Die Basis-Operationen wie GET, POST, PUT und DELETE werden ausführlich und verständlich erläutert. Dadurch wird uns auch schnell klar wie wir unsere Hierarchie strukturieren sollten. Um nochmal ganz sicher zu sein, dass unser Konzept in die richtige Richtung geht, haben wir in einem Beratungstermin unsere Struktur vorgestellt und am Ende diese leicht verändert, um alle Anforderungen zu erfüllen.

## **Gedanken zum 3. Meilenstein**

Die Grundlegenden Funktionen haben wir eigentlich bereits schon im 2. Meilenstein teilweise implementiert. Zumindest das Marshalling / Unmarshalling für unsere Events / Users sind bereits drin und konkrete Vorstellungen wie wir noch die Operationen PUT, POST, GET und DELETE nutzen wollen haben wir schon. Diese umzusetzen erweist sich auch nicht als allzu schwer dank des Buches und Youtube Videos. Wir überlegen uns noch wie wir PathParams und QueryParams sinnvoll einsetzen können. Die wichtigste Erkenntnis ist, dass wir am besten noch eine Ressource Eventcontent anlegen, damit wir später gezielter drauf zugreifen können. Den Client und den Server haben wir soweit angemessen getestet, soweit läuft alles wie es sollte.

## **Gedanken zum 4. Meilenstein**

Der Meilenstein ist das einzige was für mich komplettes „Neuland“ ist. Ich habe davor noch nie mit Openfire oder Smack programmiert, also weiß ich nicht so recht was ich erwarten soll. Glücklicherweise stehen uns verschiedene Quellen zur Verfügung, darunter ein Youtube Video, das sich zum erfassen der Grundlagen als sehr hilfreich erweist. Für uns stehen die Subscriber und Publisher schon lange fest. Jeder soll beides machen können, wie wir bereits geplant haben. Somit können wir uns auf die Implementierung der restlichen Funktionen konzentrieren oder zumindest anfangen uns Gedanken drüber zu machen.

## **Gedanken zum 5. Meilenstein**

Nach dem wir uns in Phase 4 eher mit der Theorie von XMPP beschäftigt haben, kommen wir zu einer Reihe von Entschlüssen. Bei der Erstellung eines Events wird ein neues Topic angelegt, wenn User dem Event beitreten, erhalten sie sofort eine Subskription, die sie beim ausloggen wieder verlieren. Somit funktioniert das Abonnieren und Kündigen immer automatisch. Eine Möglichkeit die Topics zu löschen haben wir von Anfang an nicht vorgesehen, so sollte es auch bleiben. Eventuell werden X Tage alte Events automatisch gelöscht. Wir müssen uns noch überlegen wie wir den automatischen „refresh“ der Events code-technisch implementieren und selbstverständlich wie die definitive GUI auszusehen hat.

## **Gedanken zum 6. Meilenstein**

In der letzten Phase geht es um die Implementierung von einer GUI für unsere Applikation. Nach verschiedenen Prototypen entscheiden wir uns für ein Tab-basiertes System. Dieses System, bereits oft in Chatsystemen implementiert scheint uns am geeignetsten. Es erscheint uns von Anfang an sehr funktional, aber auch gleichzeitig einfach zu bedienen. Dieses System ist auch das am schwierigsten zu implementierende System. Besonders die Multi-Tab Funktion korrekt zu implementieren, scheint nicht besonders leicht zu sein. Wir haben uns für JavaFX als Framework entschieden (mehr dazu in der Dokumentation) und trotz der vielen Vorteile, mangelt es JavaFX noch an Dokumentation, was unsere Aufgabe etwas erschwert. Nichtsdestotrotz können wir dieses System wie erwünscht implementieren und setzen die maximale Tab-Anzahl auf sieben fest, um den Anwender, den Server und den Client nicht zu überlasten.

## **Gedanken bis zur Abgabe**

Die wichtigsten Funktionen sind bereits implementiert. Es fehlen nur noch Kleinigkeiten wie Logout Funktion und andere Graphische Details. Außerdem soll der Source-Code ausführlicher beschrieben werden. Minimale XMPP Fehler müssen noch behoben werden. Ben kümmert sich noch um die Such-Funktion, ein netter Zusatz, für den wir uns im letzten Moment entschieden haben. Ansonsten wird noch ein bisschen an der Dokumentation geschrieben, diese muss zur endgültigen Abgabe korrekt formatiert sein. Ich bin der Meinung, dass wir zeitlich soweit gut dabei sind und das Endergebnis pünktlich abgeben werden.