Lan-Talk Forum

Semester 6 – INF14B – Linda Bopp, Andreas Schick, Niklas Nikisch – Java EE Projekt

Was ist LAN-Talk?

LAN-Talk ist eine Website, auf der man als registrierter Benutzer ein Gespräch (Talk) beginnen kann und vorhandene Talks kommentieren kann.

Das Konzept orientiert sich an der App "Jodel".

Benutzer können Talks anlegen, Talks kommentieren und unangemessene Beiträge melden (einen Report erstellen).

Moderatoren und Administratoren können überdies Reports einsehen und bestätigen, wodurch der Beitrag gelöscht wird, oder einen nicht berechtigten Report verwerfen.

Architektur

LAN-Talk ist eine klassische JavaEE-Applikation, die auf einem Applikationsserver ausgerollt wird (in diesem Falle Wildfly).

Die Web-Oberfläche wurde mithilfe Primefaces 5.3 realisiert.

Die Persistenzschicht basiert auf der Java Persistence API.

Alle Beans sind CDI-Beans, die mit @Named annotiert sind.

Konfiguration des Projekts

Die MySQL-Datenbank muss ein Schema "talk" besitzen.

CREATE SCHEMA talk;

Die standalone.xml des Wildfly liegt im Projekt-Root.

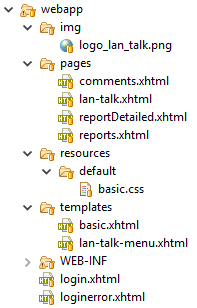
In der standalone.xml sind folgende Zugangsdaten für die talkDS hinterlegt:

Username: root

Passwort: toor

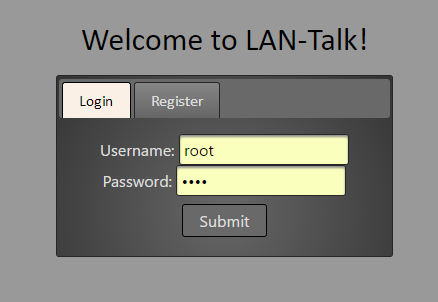
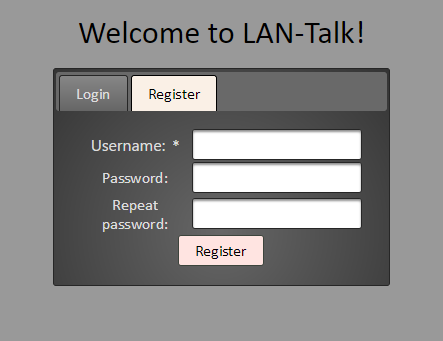
Präsentationsschicht

Die Präsentationsschicht wurde mit Primefaces 5.3 umgesetzt und in verschiedene Module gegliedert.

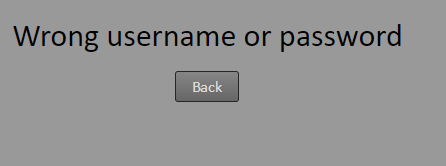


Das zugrunde liegende Template ist basic.xhtml. Hier sind die grundlegenden Konfigurationen wie die CSS Ressource festgelegt.

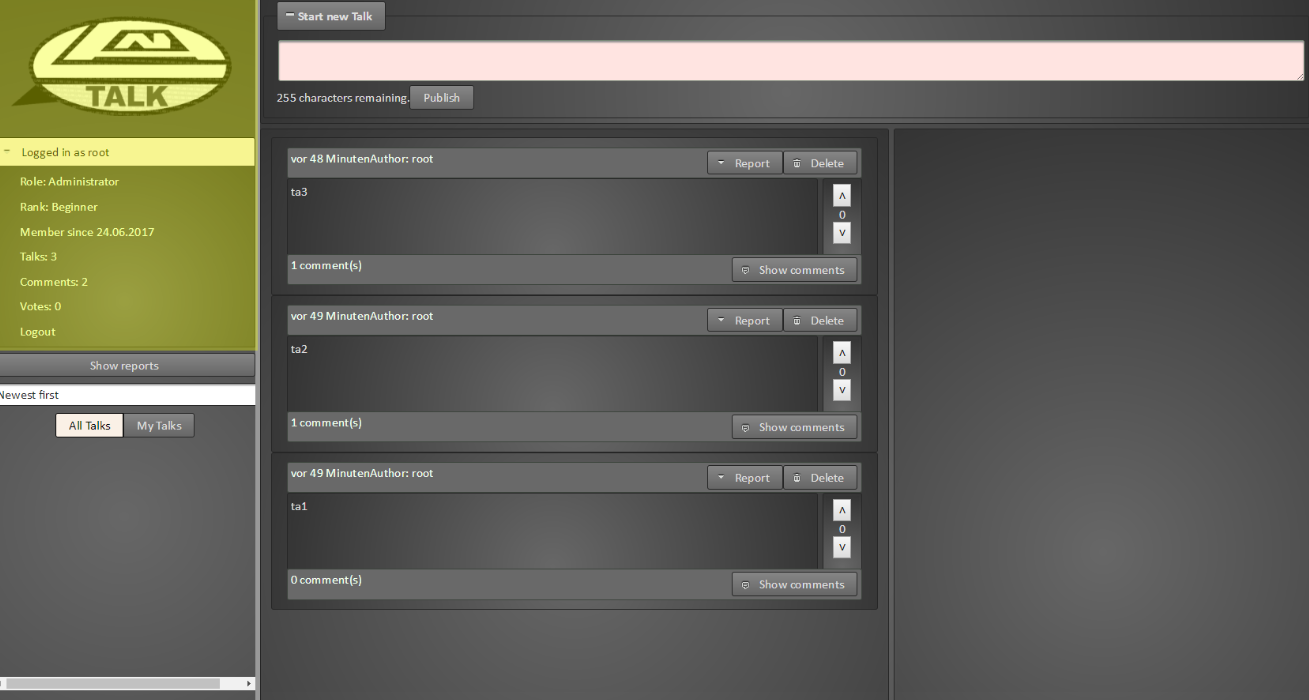
Login.xhtml:

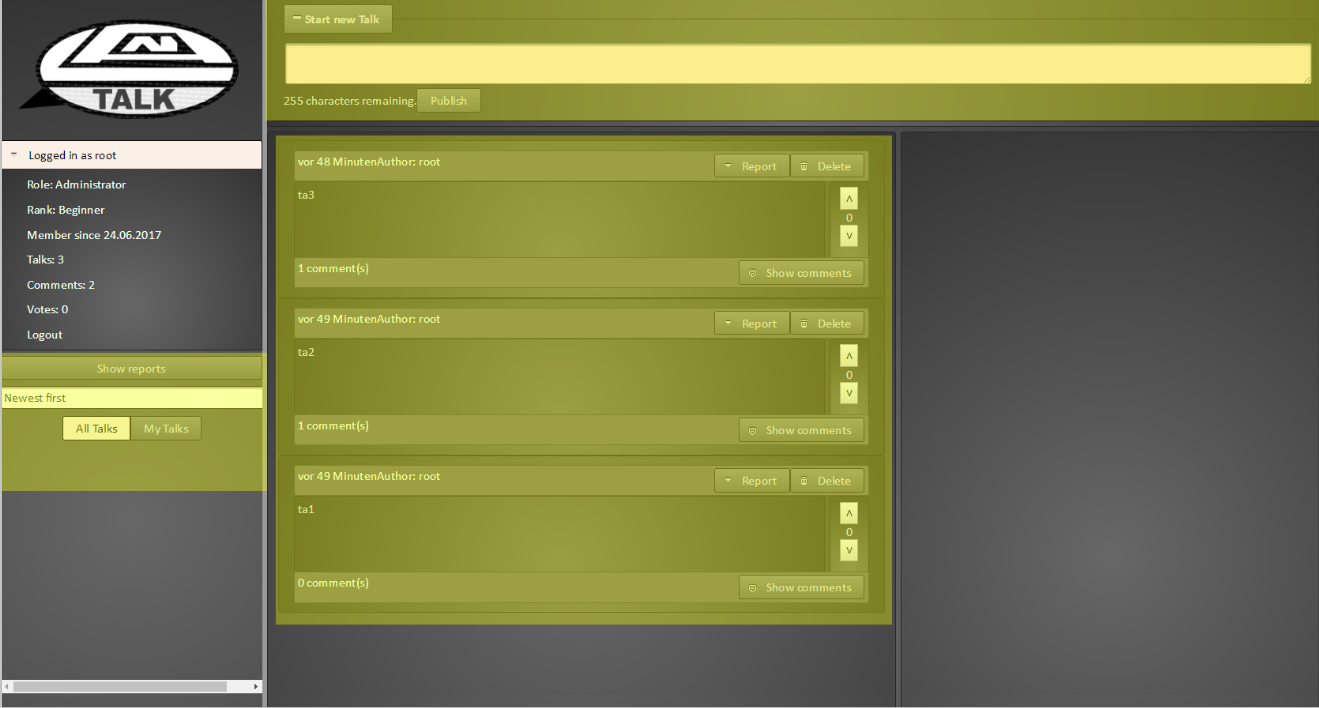
Loginerror.xhtml:



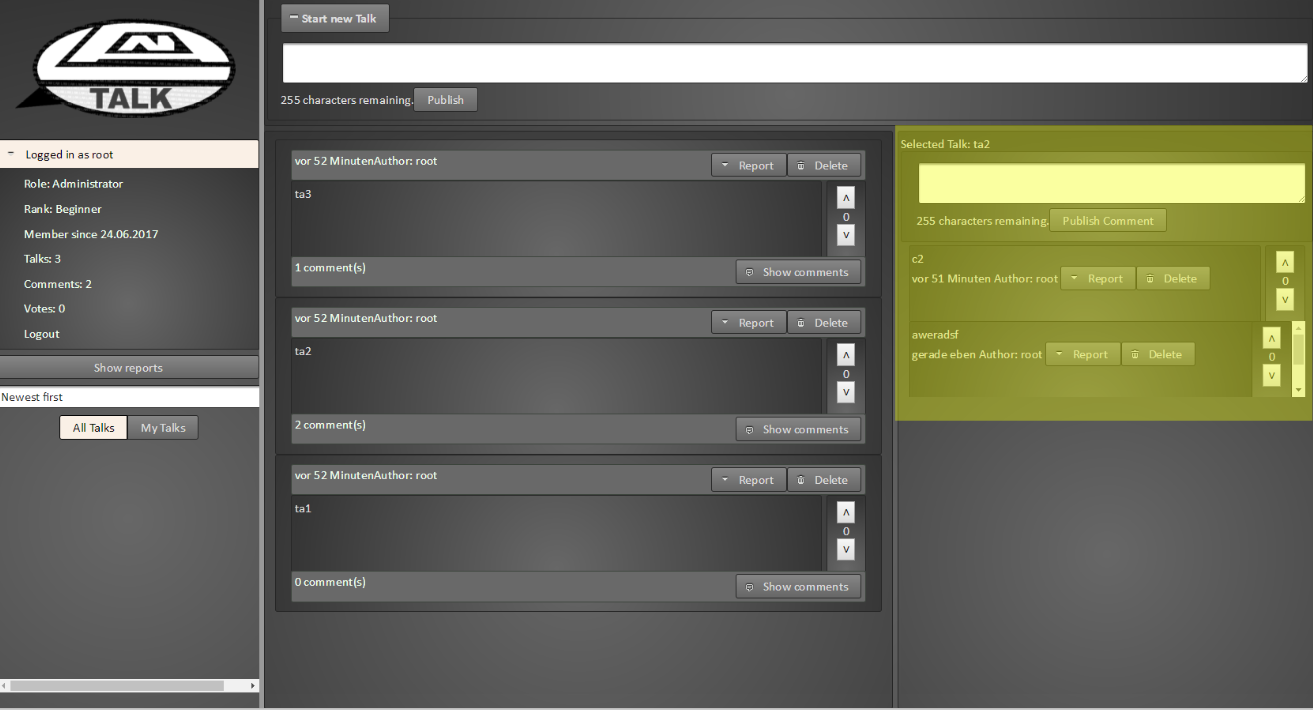
Lan-talk-menu.xhtml:



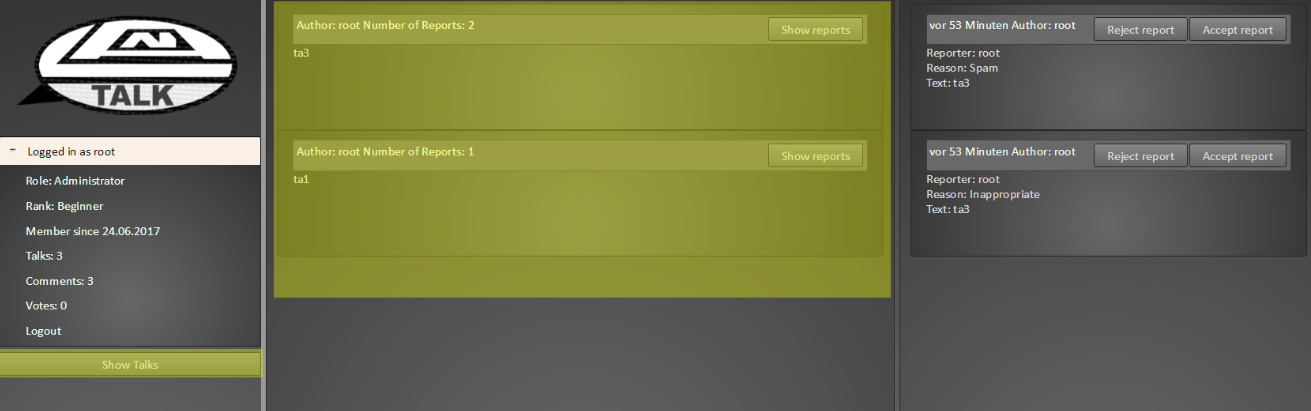
Lan-talk.xhtml:



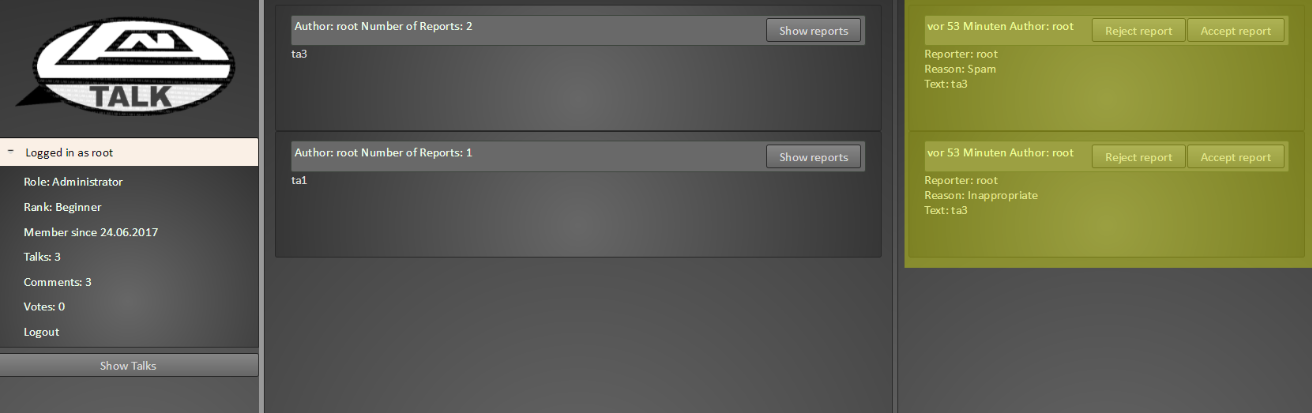
Comments.xhtml:



Reports.xhtml:



reportDetailed.xhtml:



Applikationsschicht

Die Applikationslogik wird über 5 CDI-Beans abgebildet.

UserManagerBean

Die UserManagerBean ist zuständig für das Verwalten der Benutzer. Sie kümmert sich um die Authentifizierung der Benutzer, sowie das Löschen und das "Befördern" von Benutzern. Weitere Funktionen sind:

* Anzahl der Posts des eingeloggten Users ermitteln
* Anzahl der Kommentare des eingeloggten Users ermitteln
* Anzahl der Upvotes, die der eingeloggte User bekommen hat ermitteln
* Liste der Kommentare eines Users erhalten
* Liste der Posts eines Users erhalten

NewUserBean

Die NewUserBean ist für die Registration eines neuen Users zuständig. Hier werden übergebene Parameter überprüft (existiert der Benutzername bereits?/stimmen die übergebenen Passwärter überein?) und ein neuer Benutzer mit der Rolle "User" angelegt.

PostManagerBean

Die PostManagerBEan kümmert sich um das Verwalten der Posts. Sie ist zuständig für das Laden, Erstellen, Sortieren, Löschen, Up-/Downvoten der Posts.

CommentManagerBean

Die PostManagerBEan kümmert sich um das Verwalten der Comments. Sie ist zuständig für das Laden, Erstellen, Löschen, Up-/Downvoten der Posts.

ReportBean

Die ReportBean kümmert sich um die Reports die in der Applikation eingereicht werden. Sie ist zuständig für das Reporten eines Posts/Comments und Löschen eines Reports. Außerdem kann ein Report "akzeptiert" werden. Wird ein Report dabei als gerechtfertigt eingestuft wird der Report und alle dazugehörigen Komponenten (Post/Comments/User) gelöscht. Des Weiteren ist es möglich sich über die ReportBean alle zu einer TextComponent zugehörigen Reports zurückzugeben.

Um die Zeitstempel in ein menschenlesbares Format zu übertragen wurde die Bibliothek PrettyTime (<http://www.ocpsoft.org/prettytime/>) genutzt. So wird aus 20.06.2017 bspw. "vor 4 Tagen" (wenn heute der 24.06.2017 ist)

Persistenzschicht

Die Datenbank wurde mit MySQL aufgesetzt. Die Kopplung zwischen JavaEE und der Datenbank findet mithilfe der Java Persistence API statt.

Java seitig wurden die Objekte

* User
* TextComponent
* Post
* Comment
* Point
* Report

Angelegt. Der User repräsentiert den eingeloggten Nutzer. Über ihn können alle geschriebenen Kommentare und Posts eigesehen werden. Des Weiteren kann über den User nachvolzogen werden, welche Bewertungen von dem zugehörigen Nutzer vergeben wurden. Die TextComponent stellt eine Verallgemeinerung von Kommentaren und Post da. Sie beinhaltet den geschriebenen Text, die abgegebenen Bewertungen und den Autor. Post beinhalten zudem eine Liste von zugehörigen Kommentaren. Kommentare halten im Umkehrschluss eine Referenz auf den zugehörigen Post. Das Objekt Point beinhaltet Informationen über die Bewertung. Hinzukomment kennt ein Point die bewertete TextComponent und den User, welcher die Bewertung abgegeben hat. Das Report Objekt speichert gemeldete Nutzer. Hierbei beinhaltet es Daten wie, den Melder der TextComponent , die TextComponent , sowie den Typ der Meldung. Alle Objekte beinhalten zudem noch einen Zeitstempel, der das Erstellungsdatum repräsentiert.

-commentList 
Comment 
Report 
-reporter 0.. 
-user 0..1 
User 
-textComponent Cl.. 1 
-commentList 0 * 
Textomponent 
-textComponent 1 
-user O.. 
-points 
-pointList 
Point 
-post 
Post 
-postList 

Auf der Datenbankseite werden die Beziehung der Objekte untereinander, so speicherplatzsparsam wie möglich abgebildet. Hierzu referenzieren Identifikationsnummern auf einzelne abhängige Objekte.

0 VARCHAff2SS) 
BIC T 一 20 一 
0 — VARCHAR(S2' 
w 。 「 d VARCHAR(2SS) 
一 ! IT VARCHAR(255' 
◇ 一 · ~ L 21 一 11 一 
◇ user INT()1 一 
OtextComgment-idlNT(11) 
018 3 20 一 
一 一 C , | n11 ) 
0 INT'11 一 
B 20 一 
◇ , 11 · VARCHAR(2SS' 
- 」 : 1 VARCHAR42SS) 