

TD4 : DÉVELOPPEMENT D'UNE APPLICATION DE SALON DE DISCUSSION (CHAT)

Suite Du TD 3
sur deux séances

Objectifs

2

- Le projet a comme finalité le développement d' une application de salon de discussion (chat).
 - Il est séparé en deux parties:
 - ▣ Partie 1 : Développement de la gestions des utilisateurs, la planification, l'édition, l'affichage de la liste des salons de discussion (chats)
 - Structuration de l'application en architecture MVC (JSP/Servlet/JAVA BEANS) et implémentation de la couche accès aux données
 - ▣ Partie 2 : Développement de serveur de chat avec le protocole WebSocket
- À rendre avant le 01/05**
- L'application est structuré selon une architecture MVC
 - ▣ Modèle : classes métiers (JAVA BEANS) avec une couche accès aux données (Active Record ou DAO)
 - ▣ Vue : des pages HTMLs/JSPs
 - ▣ Contrôleur : Servlets
 - Dans ce document, on utilisera le terme « chat » pour désigner « un salon de discussion »

Premier partie : application web de chat

3

- ❑ On souhaite étendre le résultat du TD3 afin de développer une application de chat. Dans cette application, chaque utilisateur peut programmer un chat à une date avec une durée de validité et en invitant d'autres utilisateurs.
- ❑ On ne souhaite pas sauvegarder les messages mais elle seront seulement diffuser sur les sessions (canaux connectés) de tous les utilisateurs connectés à ce chat. Une personne qui se connecte tardivement à un chat ne verra pas les messages publiés antérieurement.
- ❑ Un chat a identifiant unique, un titre, une description, une date avec un horaire et une durée de validité.

Le modèle

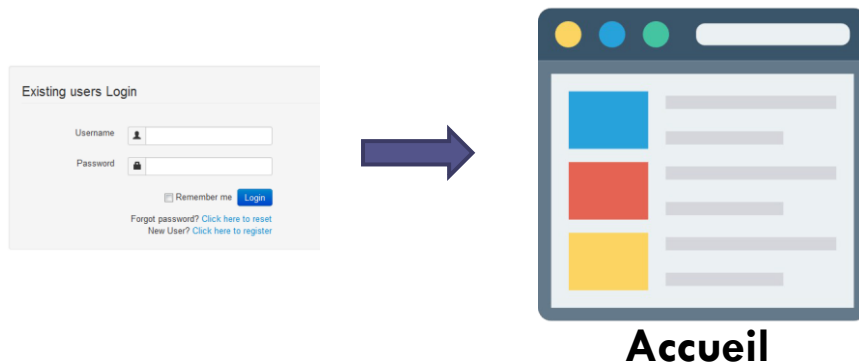
4

- Développer le modèle (JAVA BEANS) et la couche accès aux données en basant sur le design pattern « active record » ou « DAO » en complétant l'implémentation de l'application de chat :
 - ▣ Gestion des chats (ajouter, supprimer)
 - ▣ Retourner la liste de tous les chats d'un utilisateur dont il est propriétaire
 - ▣ Retourner la liste de tous les chats d'un utilisateur dont il est invité

Page d'accueil utilisateur

5

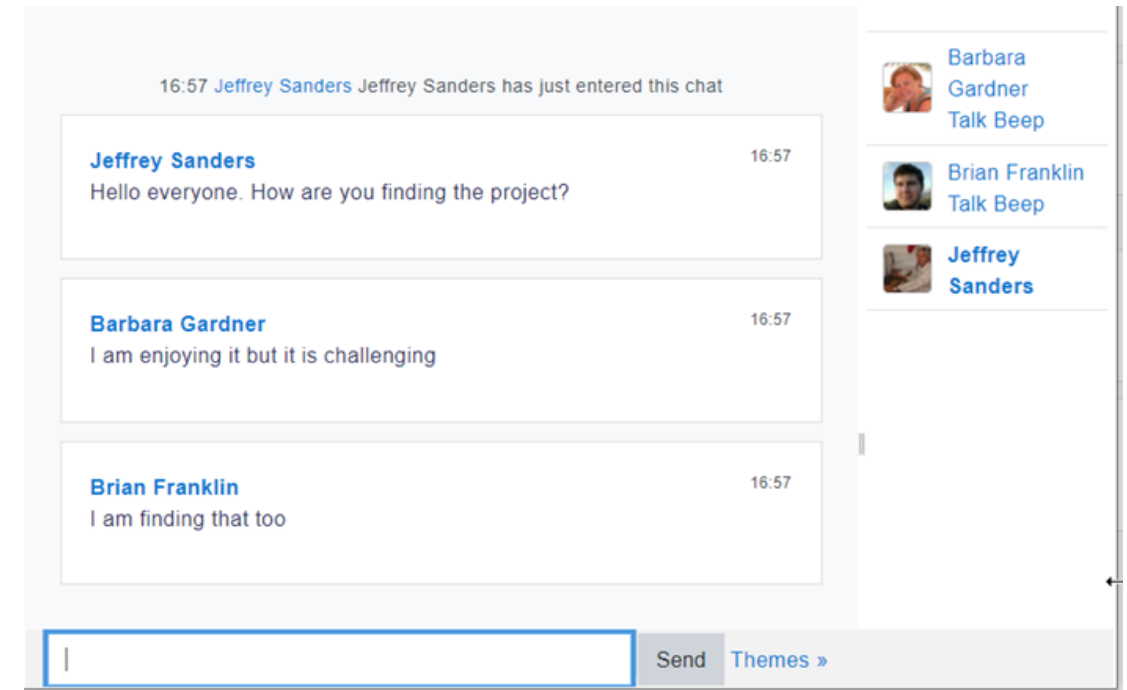
- Un menu avec les liens suivants :
 - ▣ Accueil : Affiche tous les chats dont l'utilisateur est invité ou il est le propriétaire.
 - ▣ Mes chats : afficher la liste (items) des chats dont l'utilisateur est le propriétaire. Chaque item a un lien qui pointe vers la page du chat désigné et des liens pour modifier ou supprimer le chat.
 - ▣ Mes invitations : afficher la liste (items) des chats dont l'utilisateur est un invité. Chaque item a un lien qui pointe vers la page du chat désigné.



Fenêtre (page) de chat (HTML 5 + JAVASCRIPT)

6

- Lorsque l'utilisateur clique sur un lien chat une nouvelle fenêtre est ouverte qui est composé d'
 - ▣ un fil de discussion (TEXTAREA) : une suite de messages consécutifs sur le même chat, classés de manière arborescente.
 - ▣ un formulaire pour éditer et poster un nouveau message sur le chat : un champ de texte avec un bouton pour envoyer le message
 - ▣ une liste d'utilisateurs connectés



Deuxième partie : serveur de chat

7

- ❑ Développement de serveur de chat avec le protocole WebSocket.
- ❑ On utilise l'API javax.websocket (package): le côté serveur sera codé en Java et le client en HTML5/JavaScript (fenêtre de chat).
- ❑ Cette classe serveur websocket est définie en utilisant les annotations disponibles dans le package javax.websocket
- ❑ Voir le document introduction WebSocket en java sur moodle avec les codes de l'exemple d'un seul chat.
 - ❑ Etendre et intégrer ces codes pour supporter plusieurs chats en même temps et afficher la liste des utilisateurs connectés dans votre application de chat