

Applications concurrentes mobiles  
Projet informatique : Murs de messages (*twittes*)

CACCIOTELLA Simone  
GADHOUMI Shérazade  
JACQUELIN Augustin  
PERROT Thomas

Professeur : Dr BICHOT Charles-Edmond

26 avril 2015

# Table des matières

<b>1</b>	<b>Cahier des charges</b>	<b>3</b>
1.1	Contexte . . . . .	3
1.2	Définition fonctionnelle de l'application . . . . .	3
1.3	Fonctionnalités pour les utilisateurs . . . . .	3
1.4	Schéma UML : Diagramme de classes . . . . .	4
<b>2</b>	<b>Manuel d'utilisation</b>	<b>6</b>
2.1	Installation . . . . .	6
2.1.1	Logiciels nécessaires . . . . .	6
2.1.2	Installation sur son ordinateur . . . . .	6
2.1.3	Fonctionnalités . . . . .	10
2.1.4	Application en local . . . . .	13
<b>3</b>	<b>Organisation du code source</b>	<b>14</b>
3.1	Arborescence . . . . .	14
3.2	Organisation de l'application : Pattern MVC et fichiers XML . . . . .	14
3.3	Version répartie . . . . .	16

# Chapitre 1

## Cahier des charges

### 1.1 Contexte

Afin de favoriser la communication à distance entre individus, il a été décidé de créer une sorte de réseau social permettant chacun de suivre en temps réel les activités de ses proches. Ce réseau social nommé Twitter intègre en fait des éléments de différents réseaux sociaux tels Twitter et Facebook par exemple.

### 1.2 Définition fonctionnelle de l'application

La fonction principale de l'application Twitter est de donner la possibilité à toute personne possédant un ordinateur de communiquer avec autrui et de suivre l'actualité de ses amis. Pour cela, l'utilisateur devra d'abord créer un compte. Il pourra ensuite accéder à son mur contenant les messages de ses amis, qu'il sélectionnera parmi la base de données des utilisateurs de Twitter. Ce qui est désigné comme mur est en fait un fil d'actualité propre à chaque utilisateur, où ce dernier peut s'exprimer et où apparaissent les publications des personnes suivies. Chaque message posté sur son propre mur peut être supprimé par le posteur mais il peut aussi être "liker" par les personnes qui suivent le posteur. Lorsque l'utilisateur désire s'adonner à d'autres activités, il peut se déconnecter facilement de Twitter.

### 1.3 Fonctionnalités pour les utilisateurs

Plusieurs fonctionnalités sont proposées à l'utilisateur afin que ce dernier puisse interagir avec autrui et donner l'image qu'il désire aux autres utilisateurs.

Ces fonctionnalités sont les suivantes :

- créer un compte
- se connecter
- suivre d'autres utilisateurs de Twitter
- ne plus suivre une personne
- poster des messages
- aimer des messages postés par des personnes suivies
- supprimer un message dont on est l'auteur
- supprimer son compte
- se déconnecter

#### Créer un compte

Lors de la connexion sur Twitter une page d'identification devra apparaître. Elle permettra à l'utilisateur soit de créer un compte, soit de se connecter. Dans le cas d'une première connexion,

l'utilisateur devra renseigner le pseudo et le mot de passe choisis. Ces identifiants lui permettront de se connecter ultérieurement.

### **Se connecter à son compte**

Si l'utilisateur a déjà créé son compte, il pourra se connecter directement sur la page d'accueil précédente grâce à ses identifiants.

### **Suivre d'autres utilisateurs sur Twitter**

Lorsque l'utilisateur s'identifiera, il accèdera à une page qui comprend son mur, la liste des personnes qu'il suit, ainsi que la liste des utilisateurs de Twitter. L'utilisateur pourra alors sélectionner dans la liste des utilisateurs de Twitter les personnes qu'il souhaite suivre en cliquant simplement sur son pseudo.

### **Ne plus suivre une personne**

Lorsque l'utilisateur décidera de ne plus suivre une personne, il lui suffira de cliquer sur le pseudo de la personne concernée au sein de la liste des personnes que l'utilisateur suit.

### **Poster des messages**

Sur son mur, tout utilisateur pourra poster des messages auxquels toutes les personnes qui le suivent peuvent avoir accès.

### **Aimer des messages postés par des personnes suivies**

L'ensemble des messages postés par les personnes que l'utilisateur suit apparaîtra sur le mur. Il est alors possible d'approuver un de ces messages postés en cliquant sur "J'aime".

### **Supprimer un message dont on est l'auteur**

À côté de chaque message dont le posteur est l'auteur apparaîtra un bouton "supprimer" qui permettra d'effacer le message du fil d'actualité du posteur ainsi que de toutes les personnes qui suivent le posteur.

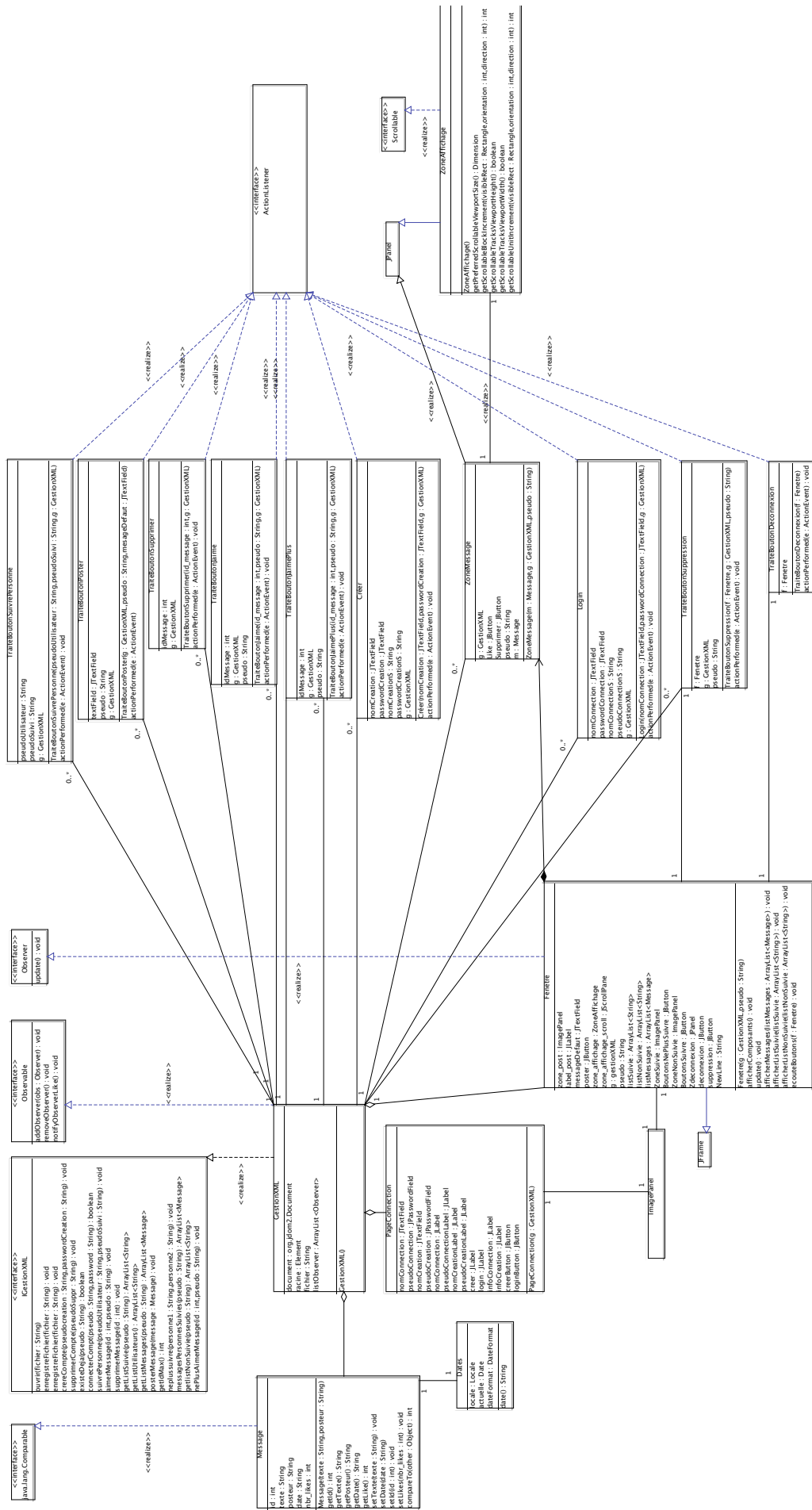
### **Supprimer son compte**

L'utilisateur aura la possibilité de supprimer son compte en cliquant simplement sur un bouton qui apparaîtra en haut de sa fenêtre principale.

### **Se déconnecter**

Il sera possible de se déconnecter à tout moment grâce à un bouton "déconnexion" qui apparaît sur la page principale.

## **1.4 Schéma UML : Diagramme de classes**



## Chapitre 2

# Manuel d'utilisation

### 2.1 Installation

#### 2.1.1 Logiciels nécessaires

L'utilisateur doit posséder le logiciel de développement Eclipse avec au minimum le JRE 1.7. On rappelle ci-dessous les différents liens internet pour installer ces deux entités.

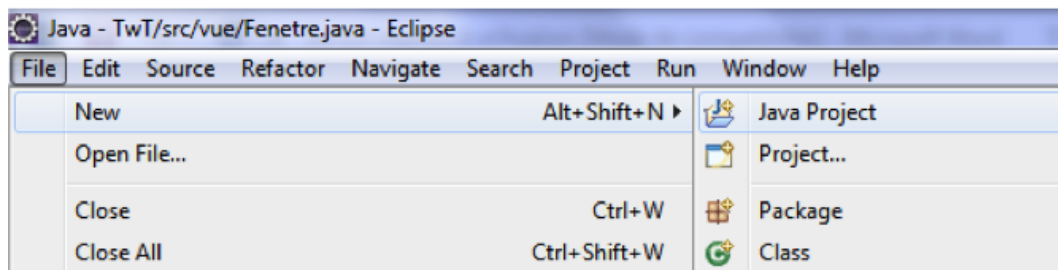
Eclipse : <https://eclipse.org/downloads/>

JRE : <http://www.01net.com/telecharger/windows/Programmation/java/fiches/8138.html>

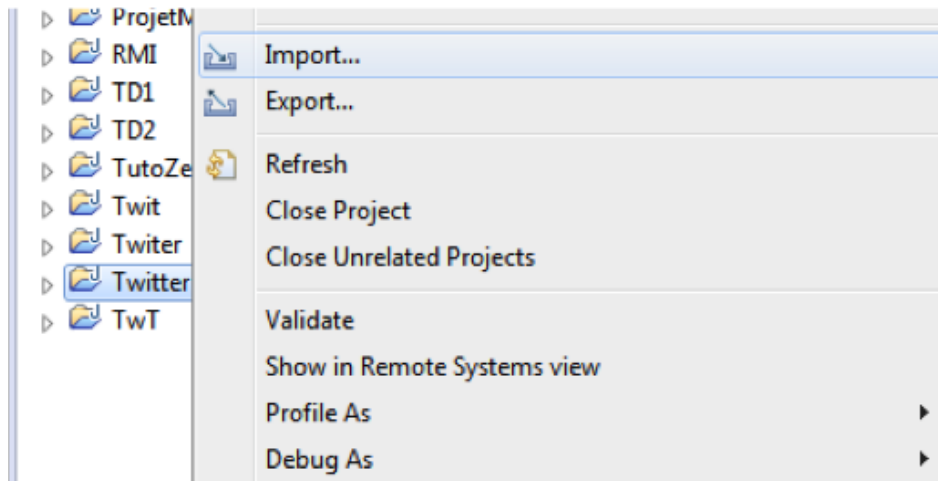
#### 2.1.2 Installation sur son ordinateur

Afin de pouvoir profiter pleinement des fonctionnalités de l'application, l'utilisateur doit impérativement :

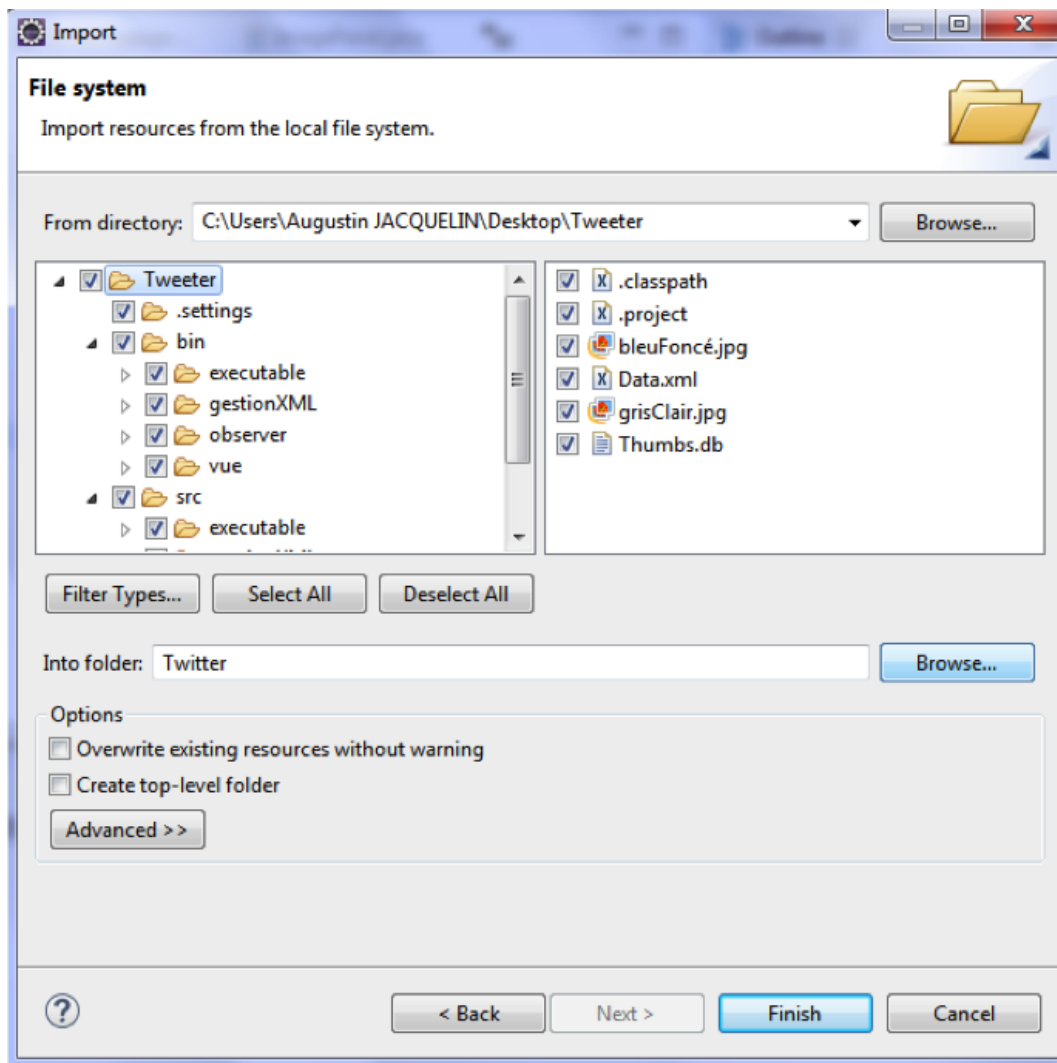
1. Télécharger et dé-zipper le fichier « Tweeter.zip » dans un répertoire bien connu.
2. Se rendre dans Eclipse, faire File -> New -> Java Project



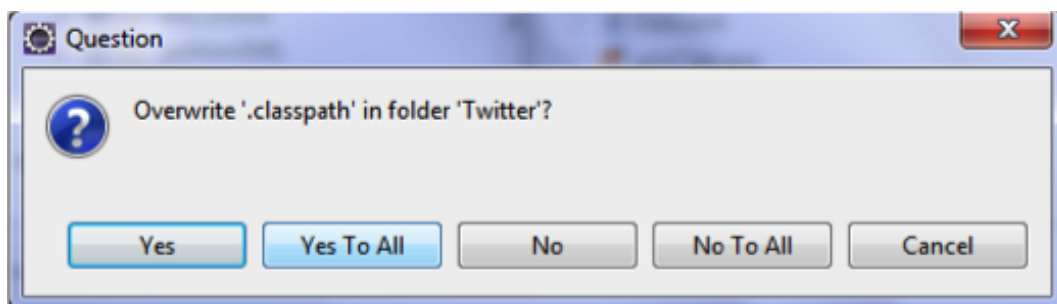
3. Créer un nouveau projet « Twitter »



4. Le projet « Twitter » s'affiche dans la barre de gauche d'Eclipse. Sélectionner alors ce projet en faisant clic droit, Import.
5. Dans « Select an import source », sélectionner Général -> File System. Cliquez alors sur Next. Dans « From directory » de la fenêtre « File system », cliquer sur « Browse? » et sélectionner le fichier dé-zippé qui se situe dans votre répertoire bien connu. S'affiche alors tous les fichiers du dossier, sélectionner les tous.

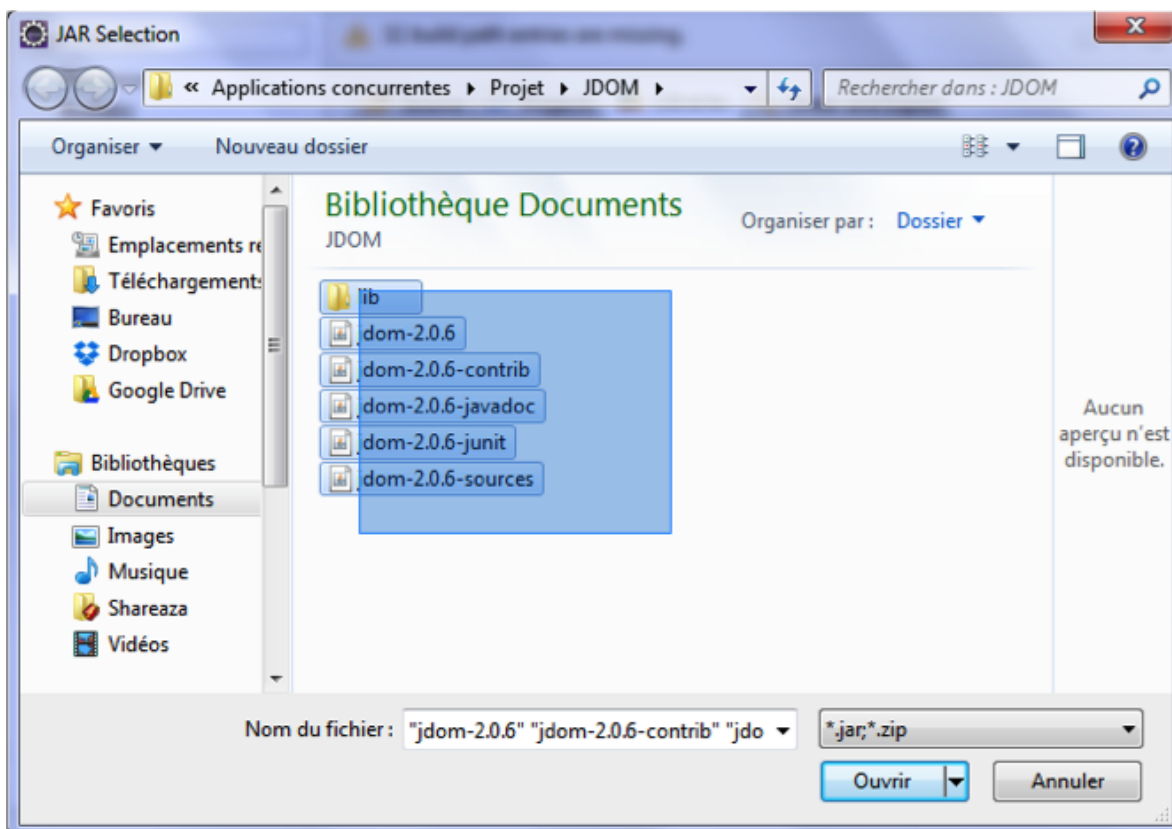
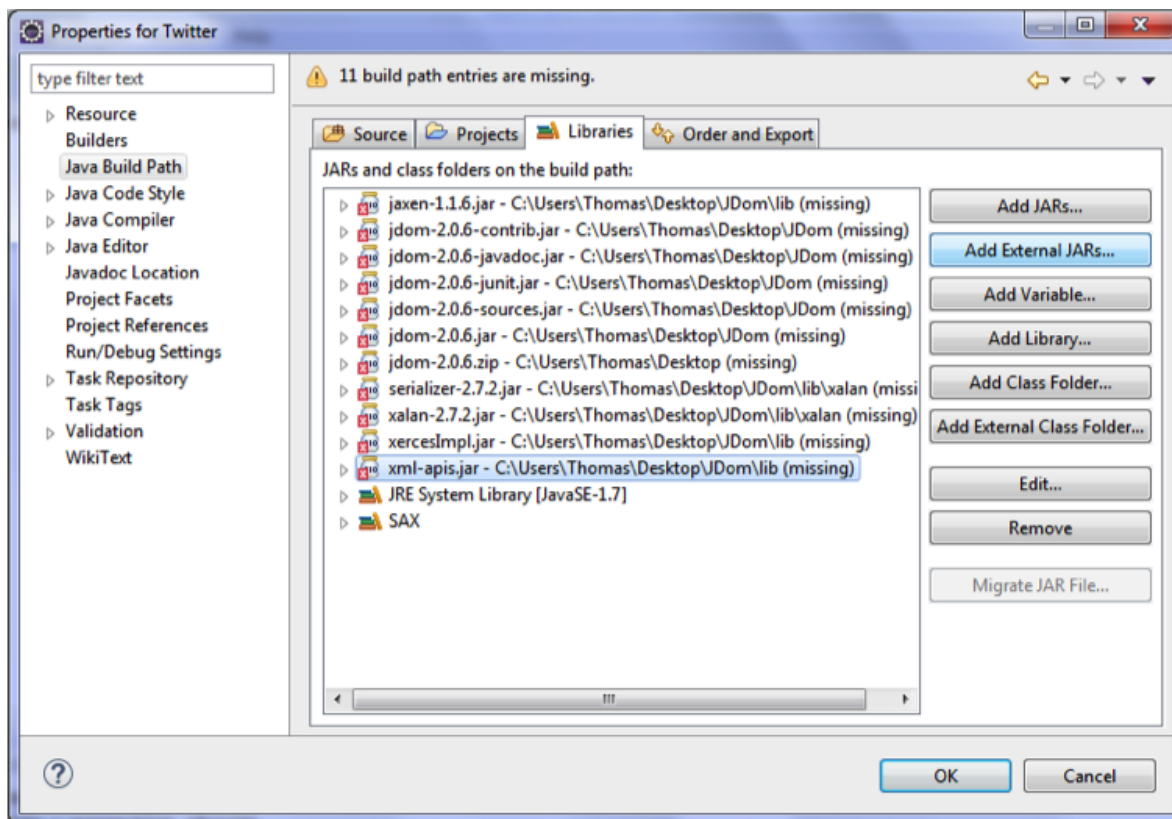


6. Dans « Into folder », cliquer sur « Browse? » et sélectionner le Java project « Twitter » fraîchement créé. Si une fenêtre « Questions » s'ouvre, cliquez sur « Yes To All ».



7. Tous les fichiers .java étant dans le Java project, il reste plus qu'à installer la bibliothèque JDOM. Si elle n'est pas déjà installée, télécharger la dernière version la plus stable et dé-zipper le contenu dans un répertoire bien connu.  
JDOM : [http ://www.jdom.org/downloads/](http://www.jdom.org/downloads/)
8. Faire clic droit sur le Java project « Twitter » créé, et aller dans « Properties ».

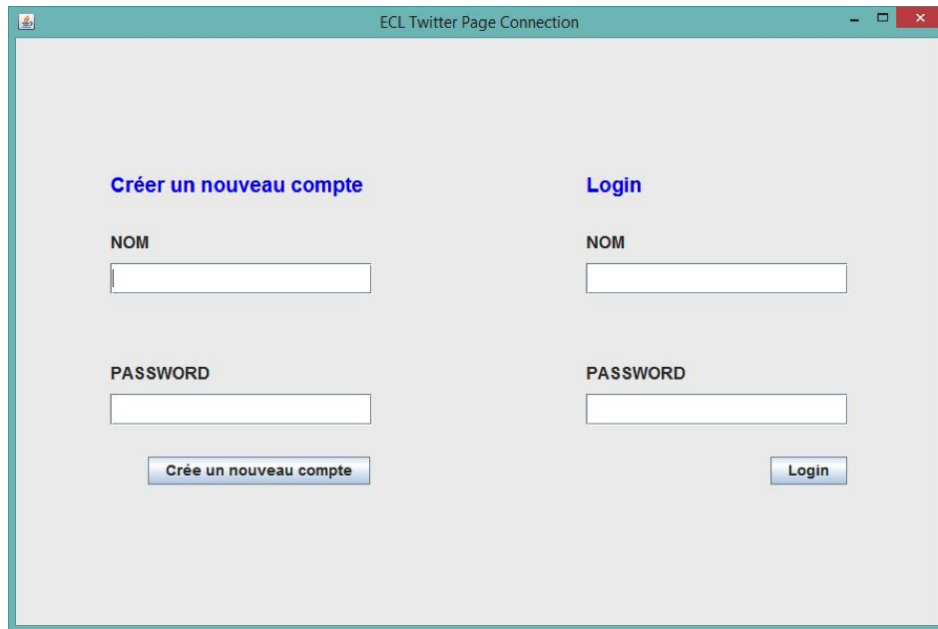




9. Cliquer alors sur « Add External JARs? », et sélectionner les fichiers dé-zippés.
10. Cliquer sur ouvrir, puis Ok, et c'est bon !

### 2.1.3 Fonctionnalités

Lorsqu'on lance l'application par le biais d'Eclipse, une fenêtre de connexion s'ouvre sur laquelle on peut soit créer un nouveau compte pour profiter de l'application, soit se connecter si le compte a déjà été créé.



The screenshot shows a window titled "ECL Twitter Page Connection". It contains two distinct sections for user authentication. The left section, titled "Créer un nouveau compte" in blue, includes labels for "NOM" and "PASSWORD" above text input fields, and a button labeled "Crée un nouveau compte". The right section, titled "Login" in blue, includes labels for "NOM" and "PASSWORD" above text input fields, and a button labeled "Login". The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

#### Création d'un compte

L'utilisateur non-encore enregistré est amené à créer un compte dans lequel il doit entrer uniquement un pseudo et un mot de passe. La création de compte est alors très simple et intuitive et permet à chacun de profiter de l'application très simplement. Une gestion des pseudos permet de bloquer le fait que deux utilisateurs aient le même pseudo.

**Créer un nouveau compte**

**NOM**

Daho

**PASSWORD**

**Crée un nouveau compte**

**Ce pseudo n'est pas disponible**

### Connection avec un compte déjà créé

L'utilisateur déjà enregistré peut alors directement se connecter sans passer par le gestionnaire de création de compte. Il est averti lorsque le pseudo ou le mot de passe ne correspond pas.

**Login**

**NOM**

Daho

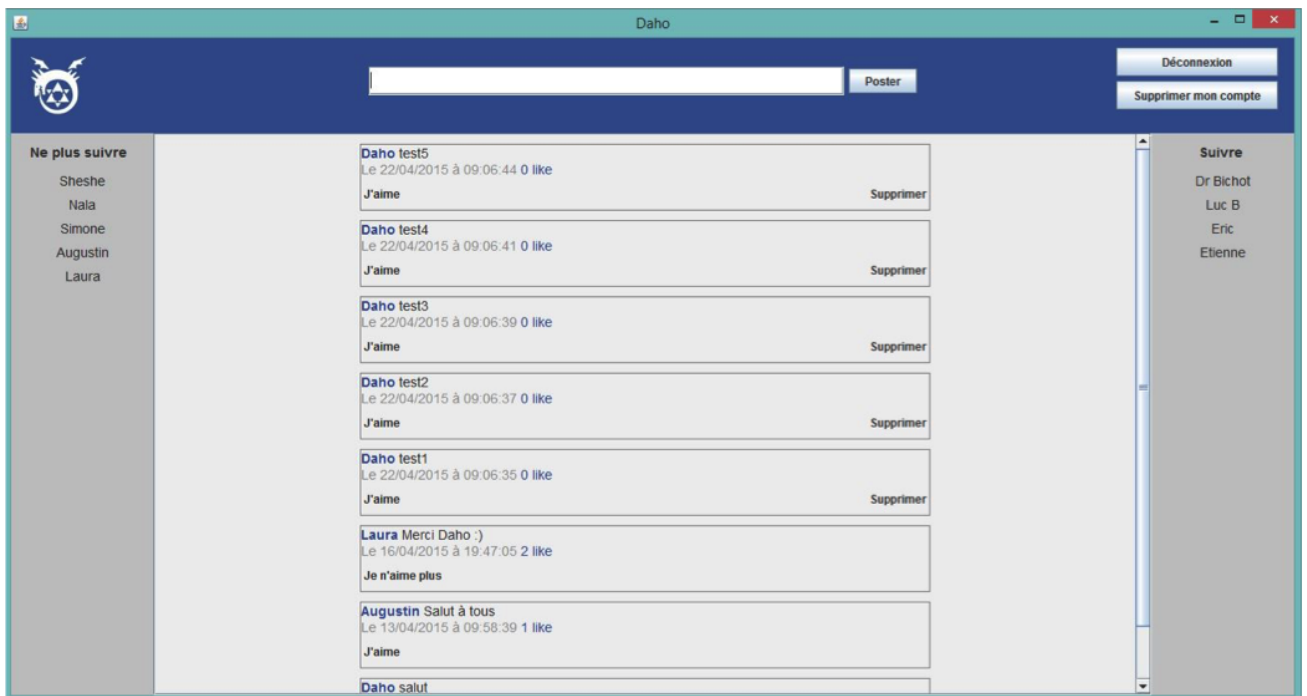
**PASSWORD**

**Login**

**Login ou mot de passe incorrect**

### La page de l'utilisateur

Une fois que l'utilisateur s'est connecté avec succès, une nouvelle fenêtre s'affiche qui correspond au mur de messages de la personne connectée. Le titre de la fenêtre est alors à son nom et il peut commencer à utiliser l'application dans le but de communiquer avec les personnes qu'il souhaite.



Cette fenêtre personnelle est composée de 4 parties :

1. La zone de post
2. La zone qui liste tous les utilisateurs de l'application dont il est possible de suivre (à droite)
3. La zone à gauche qui répertorie les utilisateurs suivis
4. La zone d'expression

### La zone de post

Située tout en haut de la page et s'étalant dans toute la largeur de la fenêtre de l'utilisateur, elle permet d'exprimer ses sentiments dans le but d'être partagé aux utilisateurs suivis. Dans le but d'utiliser cette application comme outil de réaction rapide à ce que postent les utilisateurs, nous avons décidé de limiter le nombre de caractères d'un post. Si le message posté possède un nombre de caractère plus grand que 200, le posteur en est averti et son message ne pourra être partagé à la communauté.

### La liste de tous les utilisateurs

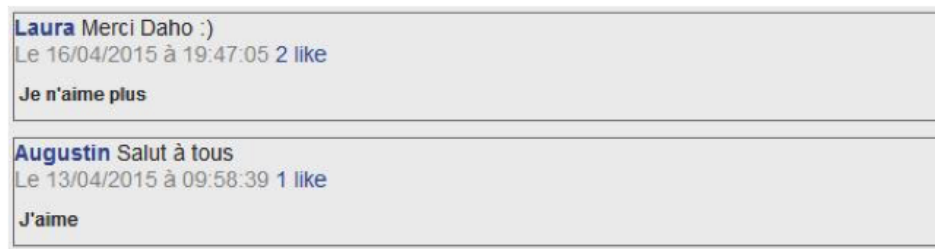
Notre application étant vue comme un moyen de partager ses pensées avec le reste du monde tout en gardant une certaine proximité avec les autres utilisateurs, il est possible de suivre ou non des personnes. Si une personne est suivie, on a alors accès à tous les messages qu'il a posté et réciproquement. Dans le but d'étendre la simplicité de post et de partage des idées, la fonctionnalité « suivre » n'est pas réciproque comme celle « être ami » sur Facebook. On peut suivre une personne qui ne nous suit pas.

### Liste des utilisateurs suivis

Comme on donne la possibilité aux utilisateurs de se suivre, on leur donne celle de ne plus se suivre. Imaginons qu'une dispute éclate entre deux adeptes de l'application, cette fonctionnalité leur permettra de ne pas voir les messages d'insulte entre les deux parties. Cette zone permet donc à l'utilisateur de voir qui sont les gens qu'il suit, et le cas échéant de ne plus suivre la personne de son choix.

## La zone d’affichage des messages

Cette zone, située en plein centre de l’application, répertorie tous les messages, triés dans l’ordre des plus récents aux moins récents, des personnes suivies et de ses propres messages.



Les messages s’affichent les uns à la suite et on laisse la possibilité à l’utilisateur de supprimer ses messages. Un compteur de like est aussi disponible qui rend compte de la popularité des messages. L’utilisateur ne peut incrémenter le compteur de like que d’une unité et peut aussi enlever le like s’il se rend tout d’un coup compte que le message qu’il a aimé n’est pas à la hauteur d’un like.

## Boutons de déconnexion et de suppression de compte

L’utilisateur peut à tout moment quitter l’application en cliquant sur le bouton « Déconnexion ». Toutes ses données sont alors conservées : ses messages et ceux des autres. Par contre, si un utilisateur souhaite supprimer définitivement son compte, un clic sur le bouton « Suppression compte » réinitialise son compte tant et si bien que toutes ses données sont perdues. Nous avons choisis de mettre une option « Suppression de compte » aussi simple à mettre en pratique pour la confidentialité des données et le respect de l’activité d’un utilisateur. En effet, à l’inverse de Facebook et Twitter qui archivent les données à des fins plutôt obscures, nous laissons la possibilité à l’utilisateur d’effacer totalement et définitivement ses données s’il le souhaite.

### 2.1.4 Application en local

L’architecture de l’application est conçue de telle manière à ce que toutes les fenêtres des utilisateurs actuellement connectés se mettent à jour dès qu’une personne met à jour la sienne. Par exemple si quelqu’un poste un message, alors toutes les personnes qui le suivent voient leur fenêtre se mettre à jour automatiquement. Cette structure permet donc à l’application de fonctionner localement.

## Chapitre 3

# Organisation du code source

Le code a été organisé en fichiers différents de telle sorte à rendre compte de l'architecture du code. Pour cela, le pattern MVC a été utilisé.

### 3.1 Arborescence

L'application Twitter comprend :

- un dossier src contenant les codes de l'application
- la JRE System Library comprenant les bibliothèques logicielles de l'API Java
- un dossier nommé "Referenced Libraries" qui contient la bibliothèque externe JDom téléchargé afin de pouvoir utiliser de fichiers XML et non pas une base de données
- des images "bleuFonce.jpg" et "grisClair.jpg" qui servent au design de l'application
- le fichier "Data.xml" qui contient l'ensemble des données des utilisateurs vis à vis de l'application : pseudos, mots de passe, messages, like , etc.

Cette arborescence est présentée sur la photo ci-dessous :



FIGURE 3.1 – Arborescence du projet Java

### 3.2 Organisation de l'application : Pattern MVC et fichiers XML

Comme présenté précédemment, l'organisation des codes du site apparaissent dans le dossier src. Afin de structurer ce code, nous nous sommes basés sur le pattern MVC (Modèle/Vue/Contrôleur). Le schéma de ce pattern est présenté ci-dessous :

La "vue" représente ce auquel l'utilisateur a accès, c'est-à-dire dans notre cas la partie graphique (Swing et AWT) de l'application. Le "Modèle" représente le coeur du programme : c'est la partie

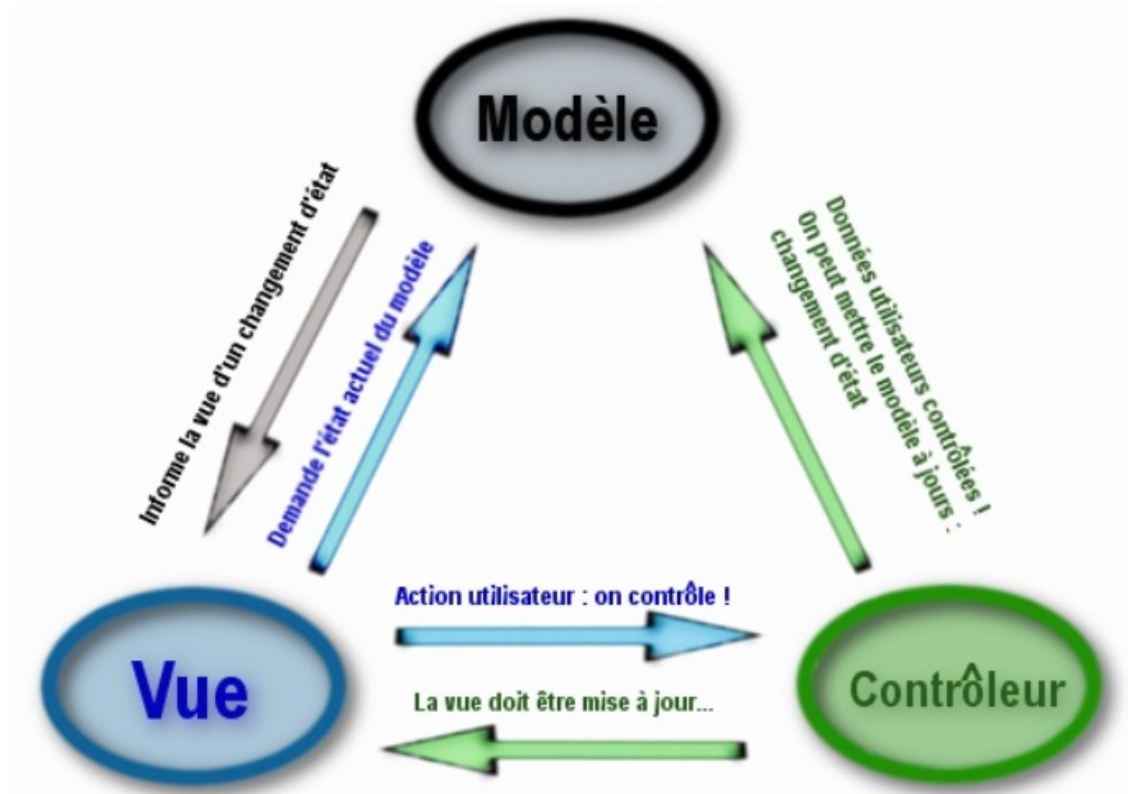


FIGURE 3.2 – Pattern MVC

qui effectuent des traitements complètement transparents à l'utilisateur. Le "contrôleur" permet de faire la liaison entre la vue et le modèle lorsque l'utilisateur intervient sur la vue : cet objet vérifie le bien-fondé des actions de l'utilisateur vis à vis de l'application.

Dans notre cas, on a réparti le programme de l'application en package qui correspondent au pattern MVC : -le modèle est composé du package "gestionXML" et du package "observer" -la vue et le contrôleur sont combinés dans le package "vue"

L'organisation de l'application est présentée ci-dessous :

### Le package "executable"

Ce package contient le "main" : il permet le lancement du programme.

### Le package "gestionXML"

L'ensemble des classes de ce package permet de modifier, écrire ou lire sur fichier "Data.xml" présenté précédemment : c'est le serveur. D'autre part, c'est dans ce package que sont créés les classes qui se chargent d'implémenter les dates et les messages. En effet, ce sont des objets qui sont intégrés au fichier XML et que la vue peut changer via l'utilisateur.

### Le package "observer"

Ce package permet de faire communiquer les différents objets entre eux. Il est constitué de la classe abstraite Observable et de l'interface Observable. La classe GestionXML va alors implémenter l'interface Observable.

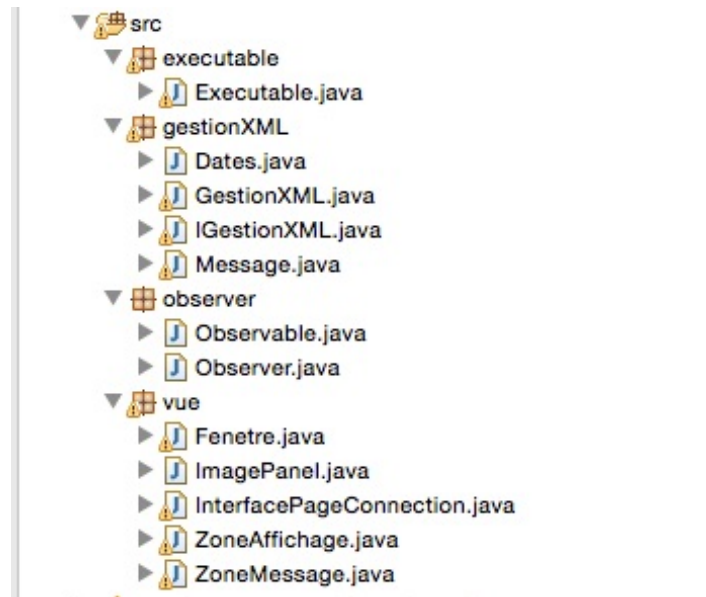


FIGURE 3.3 – Organisation du code Java

## Le package "vue"

Ce package permet de mettre en place l'interface graphique et de vérifier les données des utilisateurs : elle gère ainsi la page de connexion et la page principale.

## 3.3 Version répartie

Nous avons essayé de coder cette application en version répartie. Cependant nous avons rencontré un problème.

En effet, les objets implémentant des interfaces graphiques (package "vue") sont reliés par différentes méthodes et attributs :

- Le serveur possède une liste d'objets "fenetre" qu'il doit actualiser à chaque changement dans le fichier XML qui stocke les données, et une méthode permettant d'ajouter de nouveaux objets à cette liste (donc de rajouter des objets observés)
- La classe "fenetre" possède une méthode update(), qui permet son actualisation par le serveur, et comme attribut le serveur. Chaque instance de fenetre aura le même serveur en commun puisque celui-ci est un des arguments du constructeur de la classe fenetre.

Cependant, pour passer en version répartie, nous sommes confrontés à un problème : nous n'arrivons pas à ajouter les instances de fenetre dans la liste des objets observés par le serveur. En effet, on rencontre une exception de type "notSerializableException". Après quelques recherches, nous nous sommes aperçu que les objets graphiques (dont notre fenetre) ne peuvent pas être sérialisés même en implémentant l'interface Serializable.