

Appli'Web : rapport

Yao Yann Vianney Bohoussou - Duncan Deveaux - Keran Asseko Nguema - François Lepetit

INP-ENSEEIH

Résumé Groupe F : Rapport du projet d'appli'web

Table des matières

Résumé	1
Introduction	1
1 Principe	1
2 Architecture	2
3 Implémentation	2

Introduction

Ce rapport présente notre travail de développement d'application web dans le cadre du projet du même nom 2016/2017.

1 Principe

L'application développée est un réseau social géographique, où des utilisateurs, enregistrés ou non, peuvent placer des post-its visibles par tous sur une carte du monde.

Les post-its peuvent contenir du texte, une image et une vidéo. (lien extérieur ou youtube à fournir).

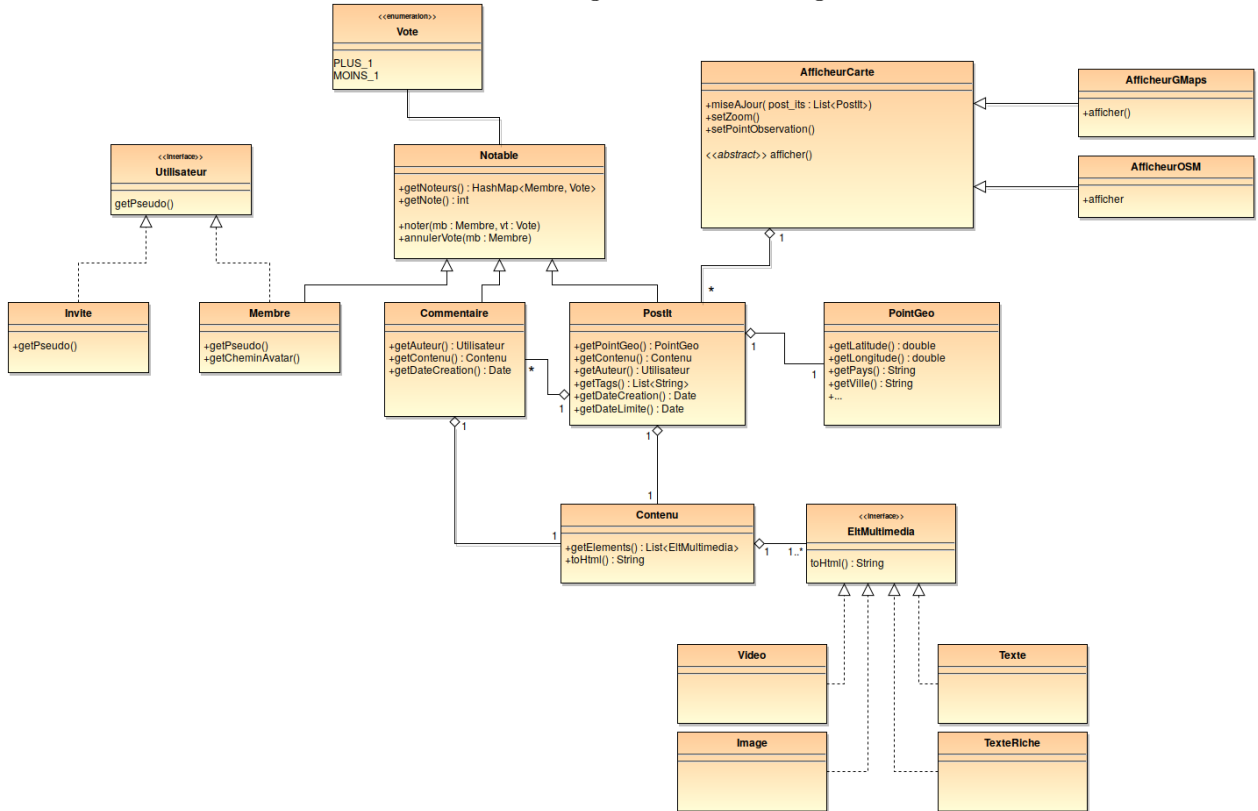
Ils peuvent être commentés par des commentaires contenant des éléments similaires.

Les commentaires, utilisateurs inscrits et les post-its peuvent être notés par les utilisateurs inscrits.

On peut aussi filtrer l'ensemble des post-its pour ne garder que ceux d'un certain auteur ou suffisamment proche d'un endroit donné ou avec une note (cf après) suffisamment haute.

2 Architecture

FIGURE 1. Le diagramme de classe original.



Nous avons décidé de créer deux Servlets :

- **ServletPostIt**, qui gère deux façades :
 - **FacadeCarte**, qui permet de manipuler l'ensemble des post-its sur la carte via les méthodes **placer** et **chercher**.
 - **FacadePostIt**, qui gère les Post-it individuellement via les méthodes **afficher**, **créer** et **commenter**.
- **ServletMembre** qui gère le système de membres via la façade **FacadeMembre**, qui possède les méthode **inscrire**, **connexion**, **afficherProfil**, **modifier** et **voter**.

Pour les entity beans on a **Membre**, **Commentaire**, **PostIt** et **PointGeo**.

3 Implémentation

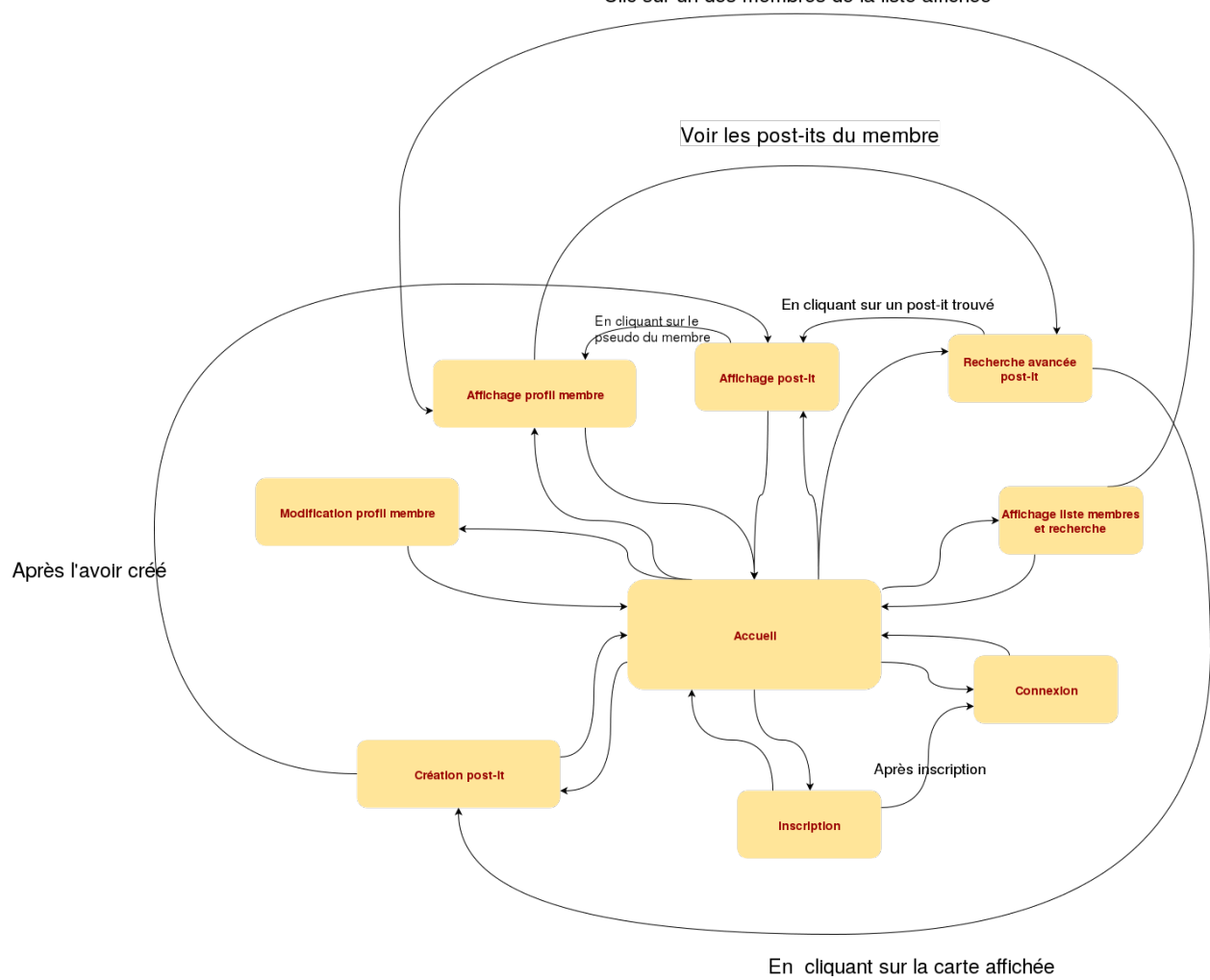
Pour inclure une carte, nous avons utilisé l'API javascript Google Maps qui nous a permis de manipuler et afficher une carte. Notamment, on peut récupérer la latitude et la longitude d'un point sur la carte et inversement. On peut placer un marqueur sur la carte (représentant un post-it), réagir au clic sur celui-ci, ...

La taille de la carte change au moment de la mise en ligne du post-it pour que l'on puisse avoir accès aux différentes données géographiques.

On affiche des images défilantes qui pourraient aisément devenir les premiers post-it à être mis en ligne.

FIGURE 2. L'enchaînement des pages.

Clic sur un des membres de la liste affichée



Donc en JEE, on a réussi à inscrire des membres, à rajouter des post-its par rapport soit à un membre connecté soit à un invité et à les conserver dans notre base de données. Ces post-its sont affichables et visibles par tous les utilisateurs qui viennent sur le site internet. Toutes les données relatives aux membres, noms, prénoms et pseudo sont conservées dans la base de données et dans l'idéal leur avatar de profil y est conservé aussi.

En plus du modèle MVC, nous avons utilisé le modèle DAO.

On a aussi implémenté les différentes obligations demandées en cours c'est à dire plusieurs façades, 2 servlets une sur la gestion des membres abonnés ou non au site et l'autre sur la gestion des post-its uploadés sur le site.

Un système de pagination a été mis en place dans l'affichage de post-it pour n'afficher les commentaires que par paquets de $N = 5$.

Les vidéos sont incluses à partir de liens externes youtube, on ne permet pas l'upload d'images mais l'utilisateur peut donner un lien vers une image externe qu'il souhaite afficher.

FIGURE 3. Un screenshot de l'ajout / affichage des commentaires

