

Université Nice Sophia Antipolis

IUT département Informatique

Lisez la description de l'étude de cas en entier et toutes les questions jusqu'au bout avant de commencer.

Le barème est donné à titre indicatif, mais il peut être modifié.

Chaque fois que vous butez sur un manque de spécification, faites un choix et explicitez ce choix par une note.

Contrôle -- M3105- S3T

Conception et Programmation Objet Avancées

janvier 2017 - Durée 1h30

Une page recto/verso au format a4 autorisée

Le courrier électronique : un impact climatique important. En 2009, il s'est échangé 247 milliards de courriers électroniques chaque jour dans le monde (SPAM compris). Dans une entreprise de 100 personnes en France, chaque collaborateur reçoit environ 58 courriels et en envoie 33 par jour. L'impact climatique de l'envoi d'un courriel avec pièces jointes augmente sensiblement avec le poids des pièces jointes, le nombre de destinataires et leur temps de stockage sur un serveur. L'envoi de 33 courriels d'1 Mo à 2 destinataires par jour et par personne génère annuellement des émissions équivalentes à 180 kg de CO₂, ce qui équivaut à plus de 1000 km parcourus en voiture. (*Internet, courriels : réduire les impacts ADEME, 2014*)

« *Globalisation des messageries !* »

Vous recevez des messages sur facebook, whatsapps, email, twitter, ... Quand vous voulez être sûre de joindre une personne vous essayez sur plusieurs réseaux. Quand vous voulez adresser une communauté, le problème est encore pire ! Les conséquences sont non seulement la difficulté pour joindre une personne mais également cela multiplie les échanges et a un impact climatique non négligeable.

Votre défi est de concevoir une application GlobalEMailing qui permet d'adresser un membre ou des communautés sans savoir sur quel canal l'information leur sera diffusée, chacun pouvant l'adapter à sa guise.

1^{re} partie : Une application de messagerie globalisée

Chaque membre de ce service mondial reçoit des « contenus de messages » sur son compte *GlobalEMailing* qui propage ensuite le message sur un compte choisi associé. Les membres « classic » ne peuvent associer qu'un compte email de leur choix. Les membres « premium » peuvent au choix faire suivre leur message sur un compte email, un compte facebook ou un compte Whatapp, mais un seul des trois pour des raisons d'économie d'énergie. N'importe qui peut envoyer des messages sur un compte *GlobalEMailing*.

La Figure 1 présente le scénario imaginé par le concepteur initial de l'application.

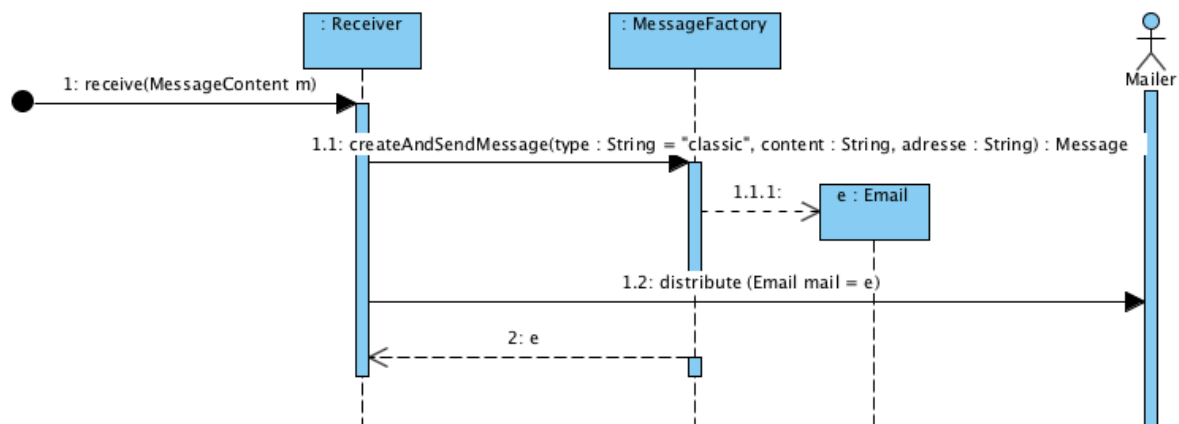


Figure 1 : Scénario de réception d'un message par un membre "classic".

Scénario ** : Un internaute pour s'inscrire doit renseigner son nom, son prénom, une adresse email valide, puis choisir son type de compte («classic » ou « premium »). Si ces informations sont valides (pas de membre avec les mêmes nom et prénom ou la même adresse email, adresse bien formée, ...), le membre est créé et, en fonction du type choisi, il précise si besoin le compte sur lequel il veut recevoir les messages et les informations afférentes comme l'identifiant facebook ou whatsapp.

Un membre peut ensuite consulter son profil et le modifier, par exemple pour changer le compte sur lequel recevoir ses messages.

2^e partie : Des communautés

L'application doit permettre de communiquer avec des communautés et des individus. Une communauté regroupe des individus et éventuellement des sous-communautés.

Un membre peut créer des communautés et/ou sous-communautés, dont il devient le gestionnaire, si la demande de création est validée par un administrateur.

Un membre peut demander à s'inscrire à une communauté. Le gestionnaire de cette communauté doit alors valider sa demande d'inscription.

Lorsqu'une communauté reçoit un « contenu de messages », elle le diffuse à chacun des individus et des communautés qui la compose, en ajoutant à l'objet du message, le nom de la communauté. Mais pour réduire le trafic, dans tous les cas, si un membre appartient à plusieurs sous-communautés, il ne reçoit le message qu'une seule fois.

La Figure 2 « visualise » un exemple de communauté. Ainsi la communauté des « PokeStudents » est composée à la fois de la communauté des « MusicianCommunity » et des « PlayerCommunity», elle-même composée des « NoLife » et des « OnlyAtNight ». Un individu peut appartenir à plusieurs communautés par exemple Ben est à la fois dans la communauté des « MusicianCommunity » et des « NoLife ». Thibs est dans la communauté PokeStudents.

Le contenu du message dont l'objet est « Soirée » et la description est « RDV à 18h ... » est envoyé à la communauté « PokeStudents ». Thibs qui a choisi de recevoir les messages sur facebook, le reçoit avec comme contenu : « objet : [PokeStudents] Soirée ; RDV à 18h ... ». Ben ne reçoit le message qu'une fois sur son compte email avec comme objet : «[PokeStudents] [NoLife] Soirée » et le contenu « RDV à 18h ... ».

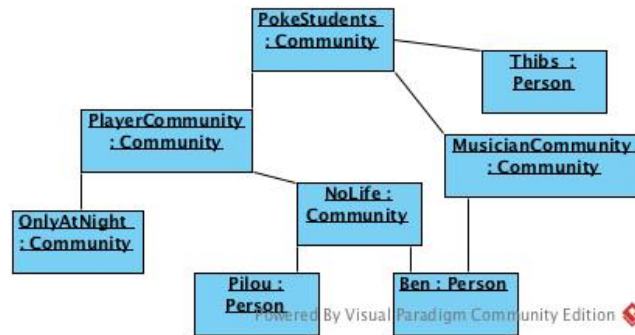


Figure 2 : Un exemple de communauté et d'individus sous la forme d'un diagramme d'objets (nom : classe)

3^e partie : Vers une messagerie moins consommatrice en énergie

Nous voulons à présent contrôler la consommation des individus et des communautés. Il existe plusieurs types d'outils : visualiser le trafic d'une communauté (nombre et poids des messages), identifier les « cibles » gloutonnes, apprendre la répartition des taux d'échanges dans le temps. Nous ne voulons pas que notre système de messagerie dépende de tous ces outils, mais nous aimerions pouvoir superviser notre messagerie.

4^e partie : Connexion à twitter

Voici des codes d'exemples permettant d'envoyer des messages sur Twitter.

Sending / Receiving Direct Messages

```

// The factory instance is re-useable and thread safe.
Twitter sender = TwitterFactory.getSingleton();
DirectMessage message = sender.sendDirectMessage(recipientId, message);
System.out.println("Sent: " + message.getText() + " to @" + message.getRecipientScreenName(
));
  
```

Question 1 : 3 pts Identifiez les acteurs et cas d'utilisation de haut niveau. Représentez acteurs et cas d'utilisation sur un diagramme en vous basant sur les 2 premières parties de l'énoncé.

Question 2 : 3 pts Dessinez un diagramme de séquence **en conception** qui explicite le scénario de la partie 1(**). Ne gérez pas les cas d'erreurs, mais faites attention à l'attribution des responsabilités.

Question 3 : 8 pts Définissez un diagramme de classes en conception en vous limitant au cœur du métier. Préciser les éléments mis en jeu dans des design patterns et leur rôle. *Attention il doit être cohérent relativement au scénario précédent sur la partie métier sans pour autant représenter les objets de l'interface graphique.*

Question 4 : 3 pts Dessinez le diagramme de classes correspondant au code qui vous est donné en partie 4. Ce code fait référence à des classes qui ne vous sont pas données. Représentez-les en fonction de ce que vous en savez uniquement.

Question 5 : 1 pt Que proposez-vous pour prendre en compte les exigences de la partie 3 ?

Question 6 : 2 pts Quel code correspond à *DirectMessageJSONImpl* relativement au diagramme UML de la Figure 3 ?

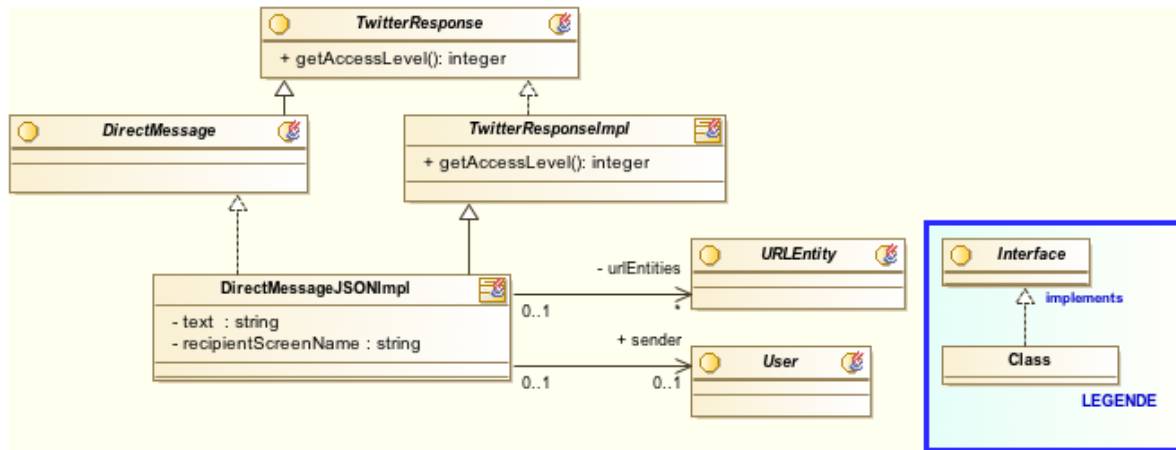


Figure 3 : Extrait de Twitter4J