

Quoi De Neuf ?

- Projet Long BDD 2015/2016.
- Binôme : Alexandre VASTRA, Zakaria ZEMMIRI.
- Description : *KoiDe9* est un système de messagerie WEB permettant aux utilisateurs de s'envoyer des messages et des photos.

PLAN :

1. Déploiement
2. Synthèse
 - Objectif du travail
 - Ce qui a été fait
 - Les améliorations à apporter
 - En quoi ce projet était difficile
 - Conclusion
3. La description technique du projet
 - Objectif techniques du travail
 - Principe de réalisation
 - Difficultés techniques rencontrées et solutions apportées.
4. Conclusion
 - La documentation utilisateur du point de vue de ce dernier (tutoriel)

1. DEPLOIEMENT

1. Importer le fichier script.sql dans la base de données afin de créer les tables nécessaires, ainsi que les utilisateurs par défaut.
2. Copier le fichier KoiDe9.war dans le dossier webapps de Tomcat.
3. Configurer l'accès à la base de données dans le fichier META-INF/context.xml.
4. Lancer tomcat.
5. Lancer l'application : <http://localhost:8080/KoiDe9/>

Liste utilisateurs par défaut (pour tester l'application) :

Login	Mot de passe
zakaria	123
alexandre	123
philippe	123

2. SYNTHÈSE

2.1 OBJECTIF DU TRAVAIL

Réaliser un système de messagerie WEB permettant aux utilisateurs de s'envoyer des messages et des photos. Un utilisateur doit s'inscrire afin de pouvoir utiliser l'application. Après inscription il dispose :

- D'un avatar par défaut qu'il peut changer ultérieurement.
- De 4 groupes par défaut : Amis, Famille, Pro et Autre.

Dès lors, l'utilisateur peut :

- Personnaliser son profil.
- Rechercher/Ajouter des nouveaux contacts.
- Consulter le profil des autres utilisateurs et voir la liste de leurs amis.
- Ajouter un utilisateur à un ou plusieurs groupes.
- Créer/Supprimer des groupes d'utilisateurs.
- Envoyer des messages ou des photos aux contacts ou aux groupes.
- Visualiser les anciens messages déjà envoyés à un contact ou groupe.
- Changer le thème de l'application.

2.2 CE QUI A ÉTÉ FAIT

- Tout !
- Tests de montée en charge.
- Refactoring et optimisation du code.

2.3 LES AMÉLIORATIONS À APPORTER

Avec un peu plus de temps, On aurait pu apporter quelques améliorations :

- Intégrer la création des tables dans l'application.
- Récupération des mots de passes par Mail.
- Voir les utilisateurs connectés en temps réel.
- Améliorer la compatibilité avec certains navigateurs.
- SSL, Changer de cryptage du mot de passe car md5 est dépassé.
- Accessible selon les critères Opquast.
- Gestion des Emojis.
- Possibilité d'envoyer un flux vidéo.
- Pouvoir jouer à des mini-jeux comme sur msn.

2.4 EN QUOI CE PROJET ÉTAIT DIFFICILE

- La difficulté principale été de réaliser un MCD correct qui répond à l'ensemble des problématiques de cette application.
- Faire un choix d'architecture pour l'application.
- Le fait que le développement de ce projet est été à faire en parallèle d'autres.

2.5 CONCLUSION

Ce projet c'est révélé bien plus riche qu'il y parait de prime abord. Et ce sur n'importe qu'elle plan. Que ce soit humain, en effet la répartition du travail au sein du groupe est un des sujets épineux car il y a eu beaucoup à faire mais les capacités de chacun n'étaient pas complémentaires dans tous les domaines. Cela s'est également ressenti dans les discussions pour les différents choix techniques. Sur le plan technique puisque beaucoup, si ce n'est l'ensemble, des différentes technologies vues au cours de l'année ont dû trouver une place dans ce projet, ce qui a permis d'en appréhender un peu plus leurs utilités au sein d'un travail concret ainsi qu'une meilleure maîtrise de ces dernières. Mais également la découverte de nouvelles technologies.

3. LA DESCRIPTION TECHNIQUE DU PROJET

3.1 OBJECTIF TECHNIQUES DU TRAVAIL

- Utilisé un maximum des technologies imposées pour mener à bien ce projet.
-

3.2 PRINCIPE DE RÉALISATION

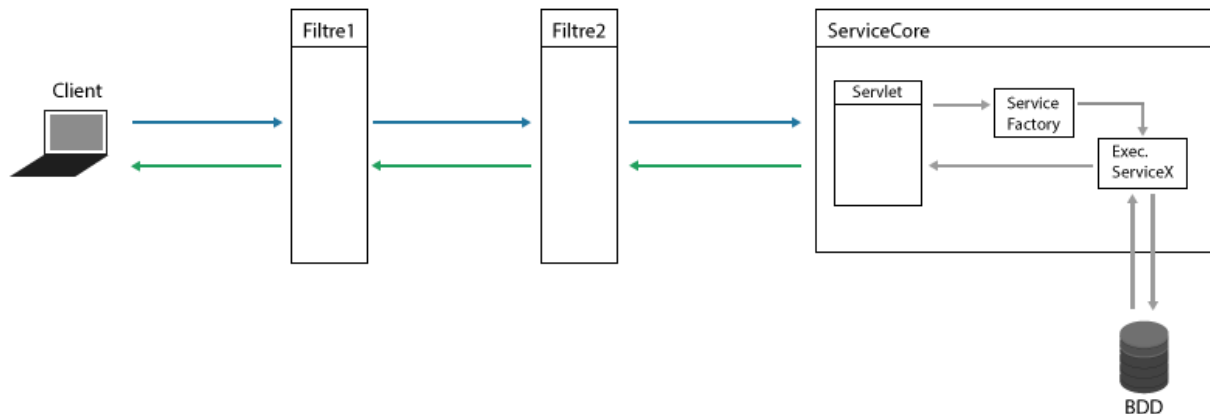
3.2.1 Bibliothèques utilisées

- Gson : bibliothèque Google pour la gestion des flux json.
- jQuery : pour les appels Ajax, et autres manipulations du DOM.
- Log4j : pour la génération de logs sur la sortie standard et dans un fichier.
- El expressions + JSTL : pour plus de simplicité d'écriture dans les jsp.
- Bootstrap.

3.2.2 Sécurité & Infos

1. ☒ Protection contre les injections SQL (PreparedStatement).
2. ☒ Protection contre les injections XSS.
3. ☒ Vérification des données envoyées côté client et serveur.
4. ☒ Filtre pour limiter l'accès aux pages privées.
5. ☒ Log de toute activité : Adresse IP, port, date, etc.....
6. ☒ Hachage des mots de passes.
7. ☒ Gestion des erreurs : ErrorPage.
8. ☒ Validation des formulaires.
9. ☒ Ergonomie du site.

3.2.3 Architecture



L'application est construite selon le pattern MVC. Avec deux parties distinctes client (front-end) et serveur (contient l'ensemble des services disponibles). Chaque partie est construite selon un ou plusieurs patterns pour faciliter le développement et la maintenance. Les deux parties communiquent en utilisant un protocole ad hoc écrit en Json à mi-chemin entre celui des communications REST et l'enveloppe SOAP.

- La communication avec les services se fait donc via des requêtes HTTP, avec un format prédéfini.

Format de communication :

- Requête : {nomService : "NomDuService", data : {} }
- Réponse : {returnCode : 0, message : "", data : [] }

Détail Format Requête :

nomService : celui qui est le servicefactory.
data : données envoyées au services.

Détail Format Réponse :

Codes Retour :

0 : OK : Tout s'est bien passé.

4 : Warning : une remarque ou une notification sur le contenu de la réponse.

8 : Error : un problème s'est produit.

Message : Permet de donner des informations complémentaires sur le code retour.

data : contient les véritables données envoyées par le service.

Une requête client passe par deux filtres :

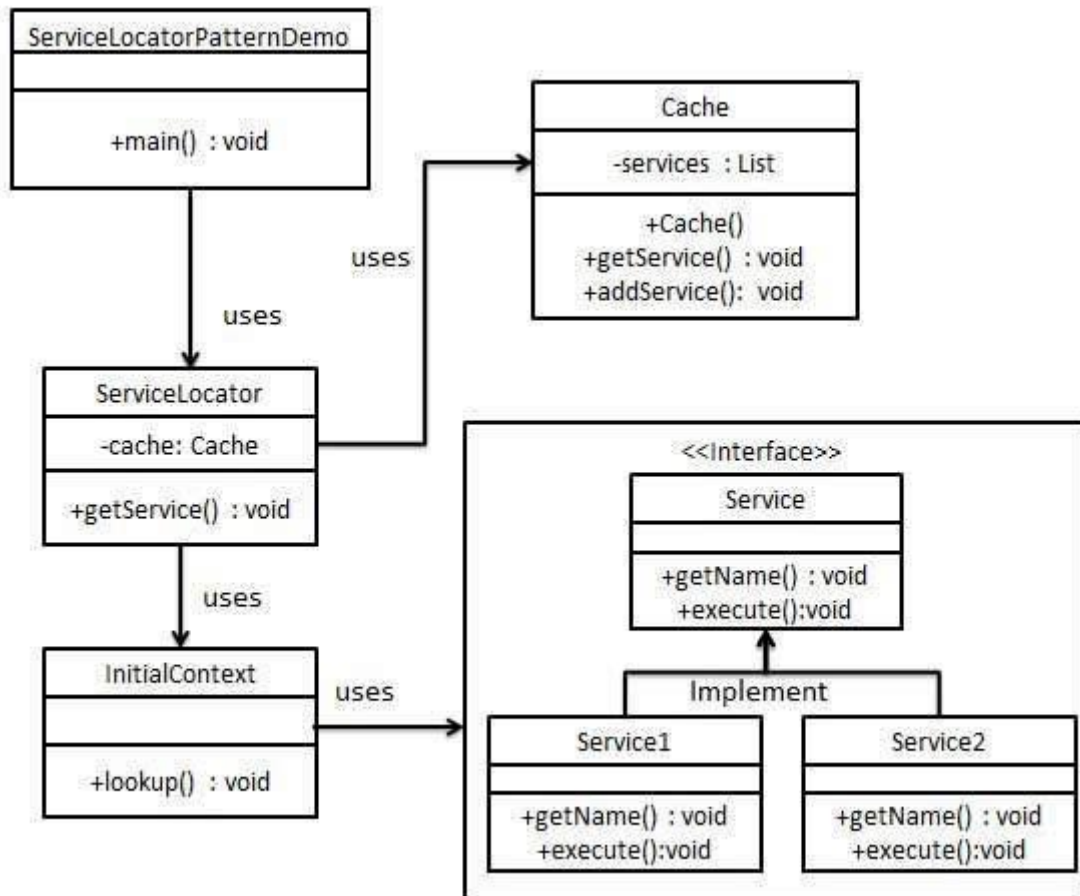
- Le premier filtre vérifie si l'utilisateur dispose des droit d'accès à la page demandée. Si l'utilisateur est connecté la requête est autorisée a passée sinon elle est bloquée et l'utilisateur se voit diriger vers la page de connexion.
- Le deuxième filtre permet de vérifier la validité de la requête. Une requête valide doit respecter le format du protocole précédemment détaillé. Si ce format n'est pas respecté, la requête est bloquée et une réponse est envoyée au client lui indiquant la nature du problème.

Une autre vérification est réalisée au niveau du service. Elle permet de vérifier si le client a bien envoyé les données requises à l'exécution du service et aussi dans le bon format, si ce n'est pas le cas, la requête est bloquée et une réponse est envoyée au client lui indiquant la nature du problème.

La servlet agit comme un « contrôleur ». Elle permet la redirection des requêtes vers le bon service, mais aussi l'envoi de la réponse au client.

Remarque : Toute données est vérifié en plus sur le client avant envoie au serveur.

• Coté Serveur :



L'implémentation des services est une variante du pattern "Locator" semblable à celui qu'on peut trouver lors de l'utilisation des composants JNDI. L'avantage avec ce type de méthode est le fait que tous les appels service se font vers une seule et unique url. Et c'est le contrôleur qui sélectionne le bon service.

• Coté Client :

Pour l'utilisateur, L'application est constituée de deux parties :

- Une partie publique que tout le monde peut consulter, qui est la page de connexion.
- Une partie privée, accessible uniquement par les utilisateurs déjà inscrits. Un filtre est utilisé pour éviter toute tentative d'accès à cette partie si l'utilisateur ne s'est pas connecté.

Coté développement, le client ne dispose que deux tâches :

- Appeler les services dont il a besoin.
- Faire la mapping entre la réponse du service et l'IHM.

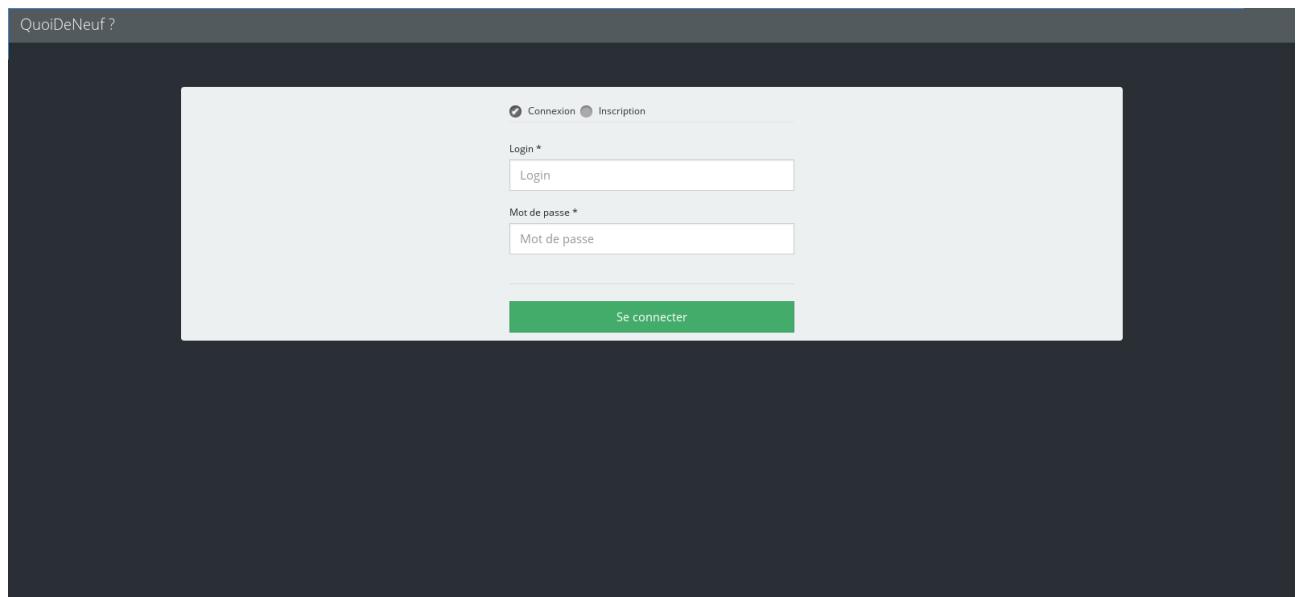
3.3 DIFFICULTÉS TECHNIQUES RENCONTRÉES ET SOLUTIONS APPORTÉES.

- Les appels asynchrones imbriqués, Solution ? Utilisation des promesses en JavaScript.

4. CONCLUSION

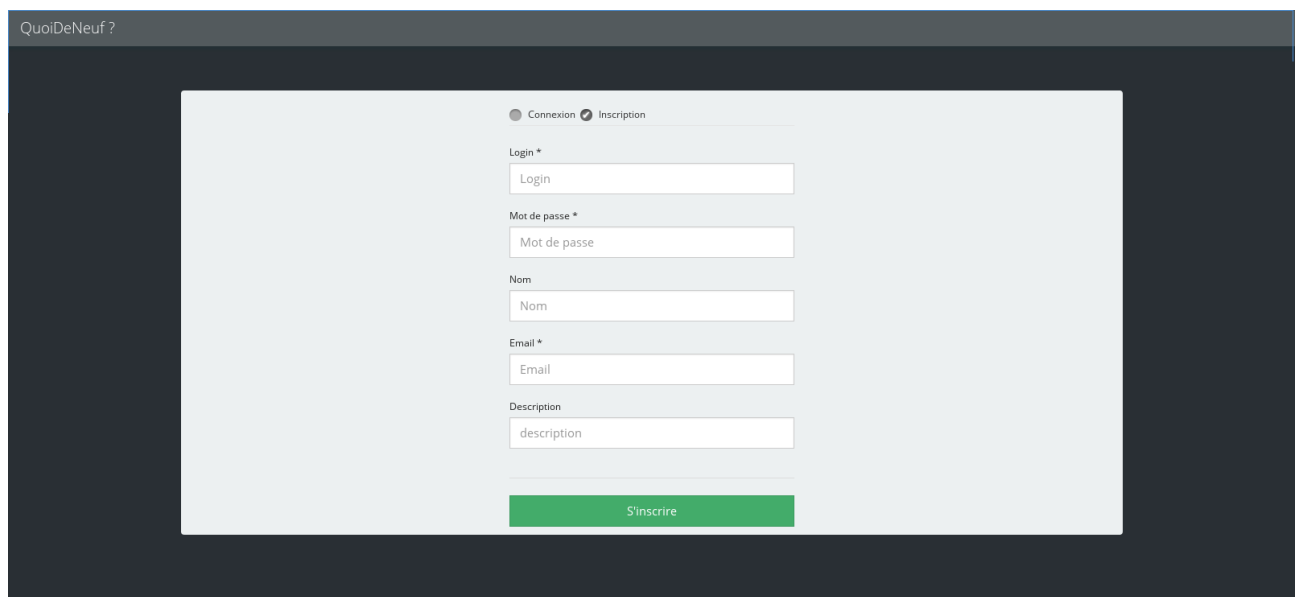
4.1 Tutoriel

Nous voici devant la première fenêtre de l'application KoiDe9, à savoir la page d'accueil :



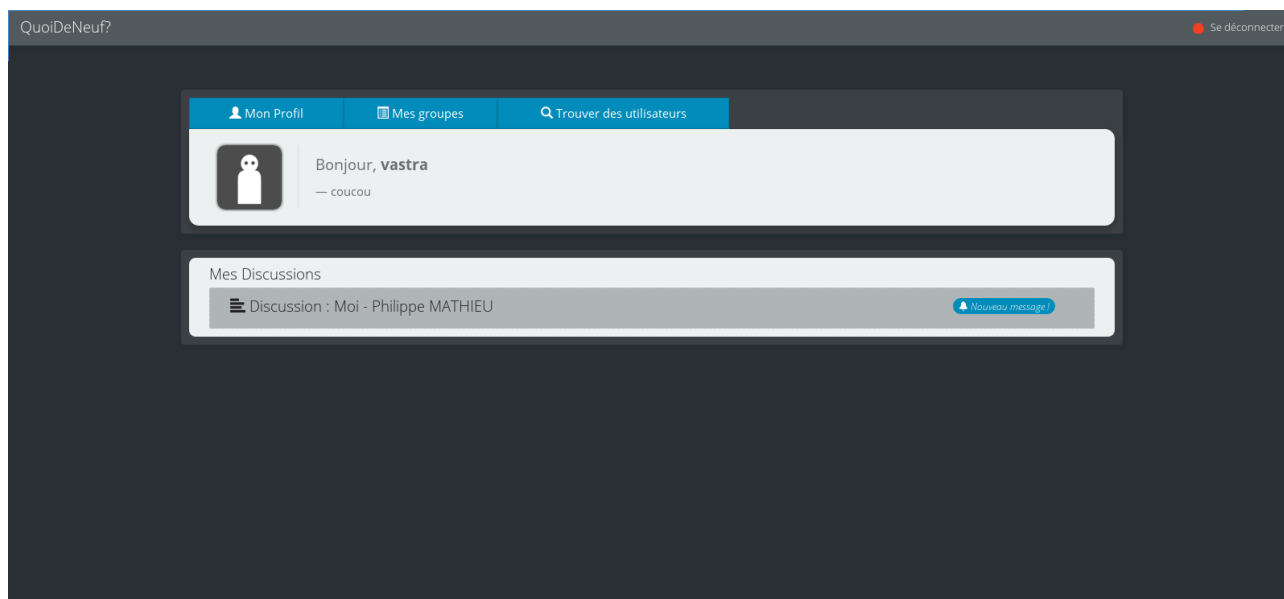
The screenshot shows the login page of the KoiDe9 application. At the top left, the text "QuoiDeNeuf ?" is visible. The main content area is a light gray rectangle centered on a dark gray background. At the top of this rectangle, there are two radio buttons: "Connexion" (selected) and "Inscription". Below these are two input fields: "Login *" and "Mot de passe *". The "Login *" field contains the text "Login". The "Mot de passe *" field contains the text "Mot de passe". At the bottom of the form is a green button labeled "Se connecter".

Si vous ne possédez pas de compte, dirigez-vous vers la section d'inscription en sélectionnant la case correspondante sur la page d'accueil :

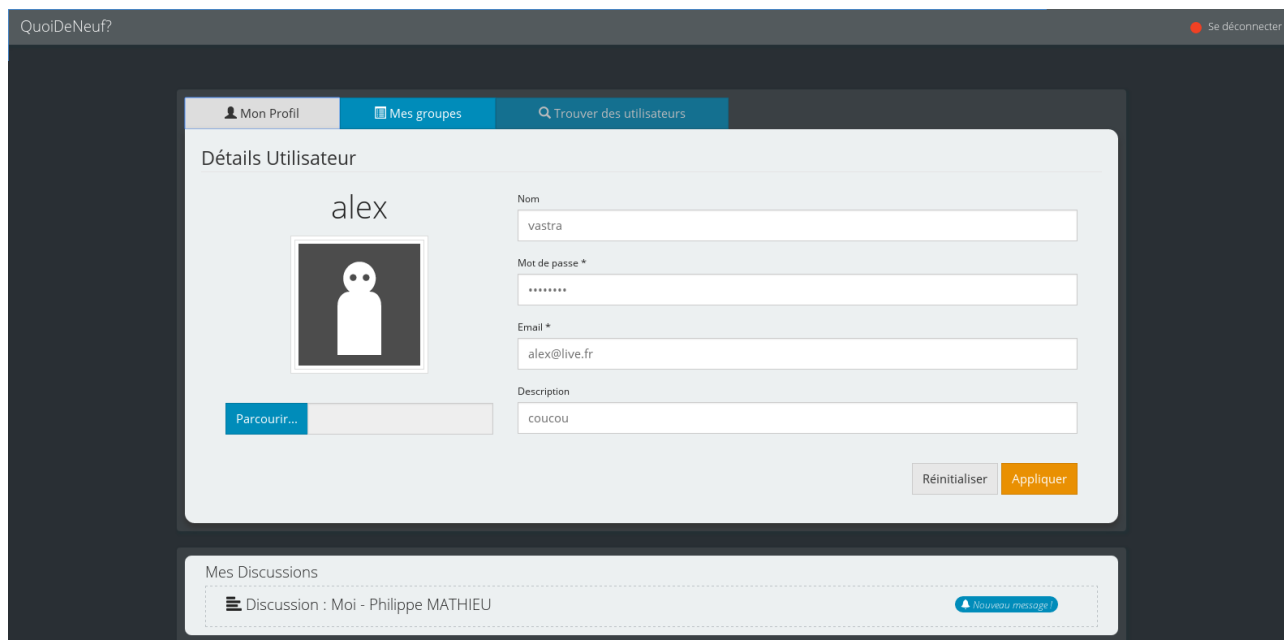


The screenshot shows the registration page of the KoiDe9 application. At the top left, the text "QuoiDeNeuf ?" is visible. The main content area is a light gray rectangle centered on a dark gray background. At the top of this rectangle, there are two radio buttons: "Connexion" and "Inscription" (selected). Below these are five input fields: "Login *" (containing "Login"), "Mot de passe *" (containing "Mot de passe"), "Nom" (containing "Nom"), "Email *" (containing "Email"), and "Description" (containing "description"). At the bottom of the form is a green button labeled "S'inscrire".

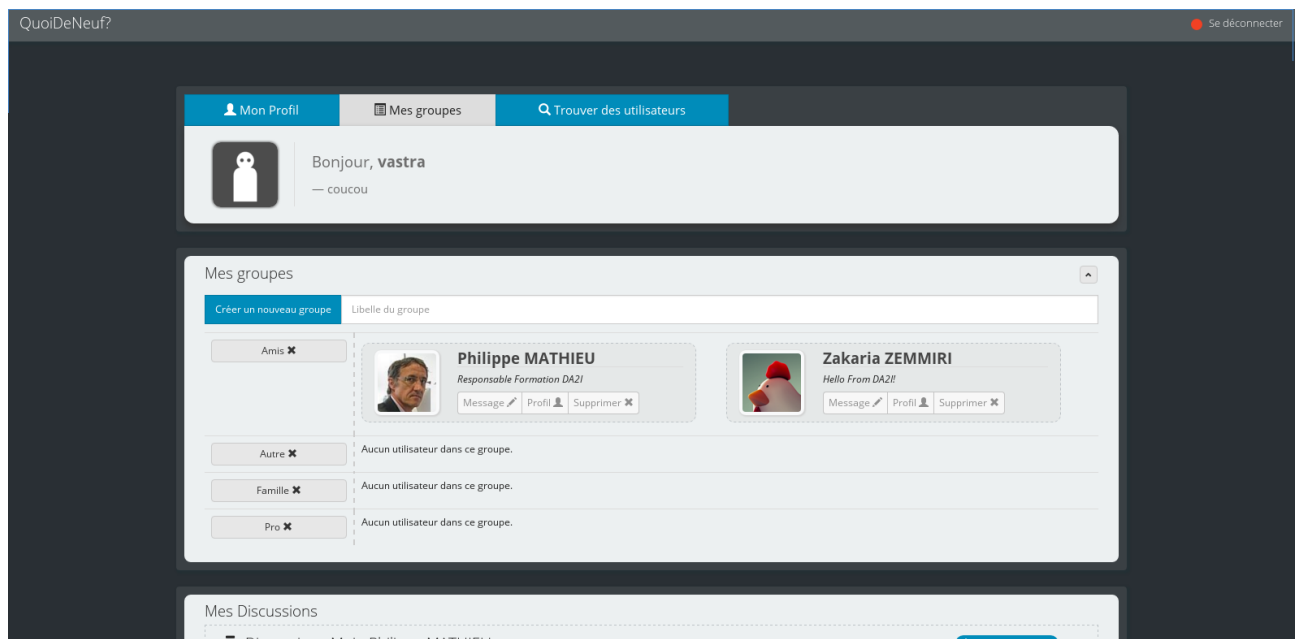
Dans le cas contraire, ou une fois votre inscription complétée, connectez-vous avec votre login et mot de passe. Vous êtes dirigé sur votre page d'accueil :



Vous pouvez customiser votre profil, comme par exemple votre avatar, via l'onglet "Mon Profil" dédié à celui-ci :

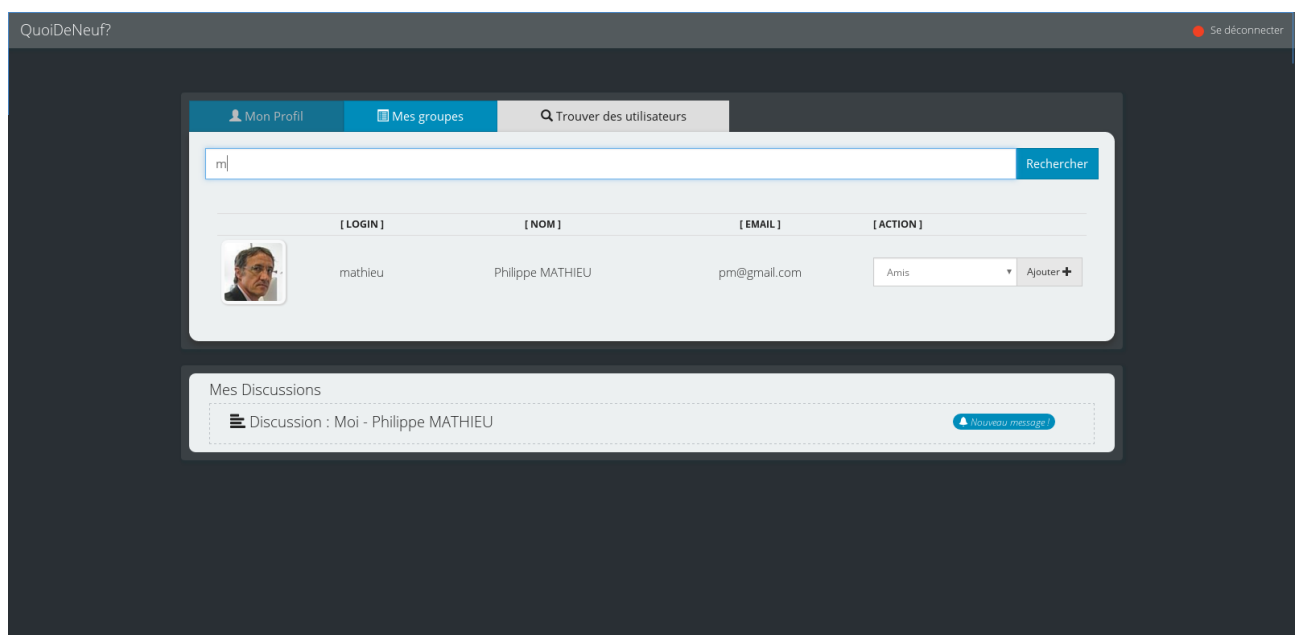


Une fois votre profil à votre convenance, dirigez-vous sur l'onglet "Mes Groupes" :



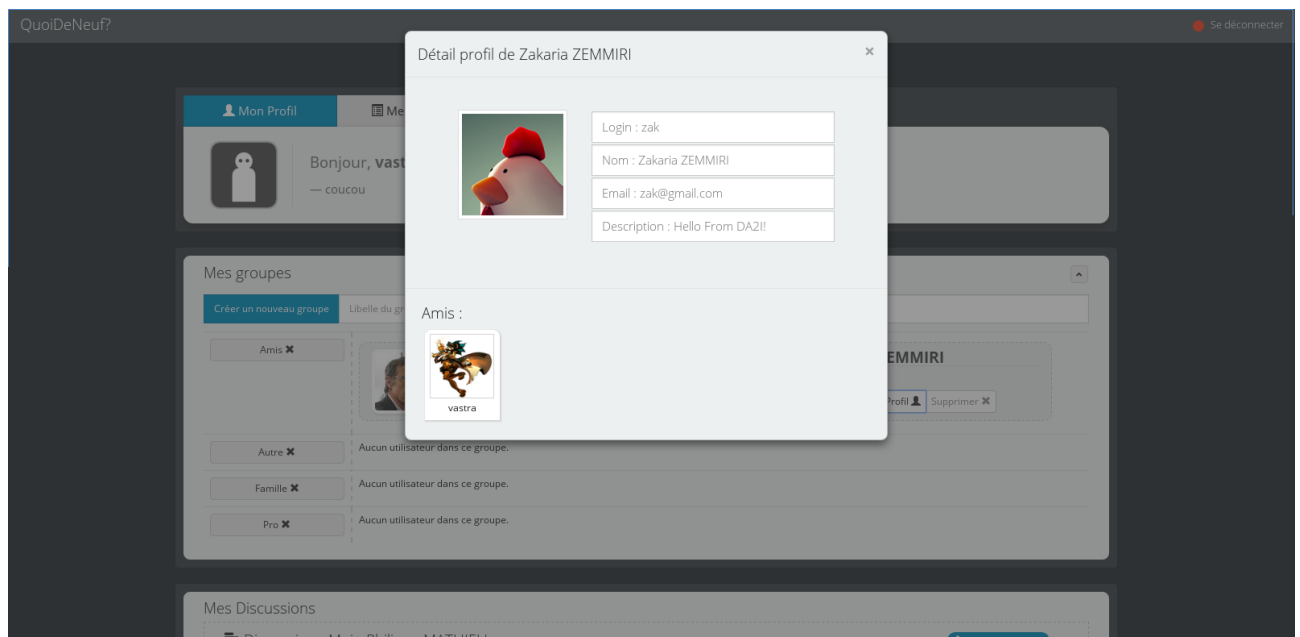
Vous pourrez y voir que des groupes y sont déjà par défaut comme famille par exemple. Vous pouvez bien entendu créer vos propres groupes.

A présent dirigez-vous vers l'onglet "Trouver des utilisateurs" : Afin de commencer à discuter avec d'autres utilisateurs vous devez dans un premier temps les ajouter en amis, dans le groupe de votre choix. La recherche s'effectue parmi tous les utilisateurs du service. De plus vous pouvez rechercher une personne en ne connaissant que partiellement son nom ou son login. Vous aurez donc comme résultat tous les utilisateurs correspondant à la recherche :

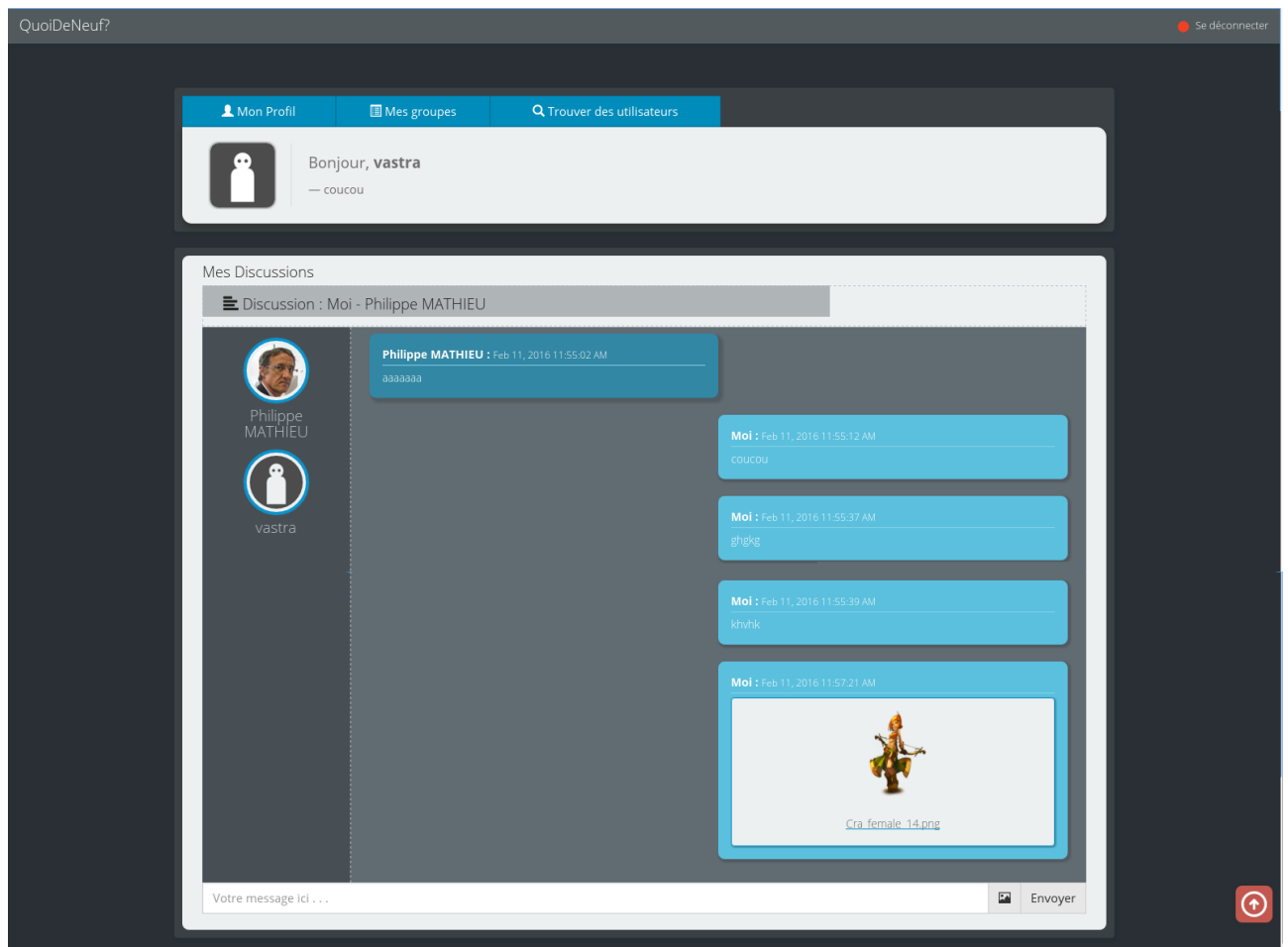


Une fois vos nouveaux amis ajoutés à vos différents groupes, plusieurs choix s'offrent à vous :

- En retournant dans l'onglet "Mes groupes" vous pouvez supprimer les utilisateurs des groupes dans lesquels vous les aviez ajoutés.
- Consulter les détails de son profil :



Et bien entendu discuter avec eux



A tout moment vous pouvez vous déconnecter en cliquant sur le bouton "se déconnecter" disponible en haut à droite depuis n'importe quelle page.