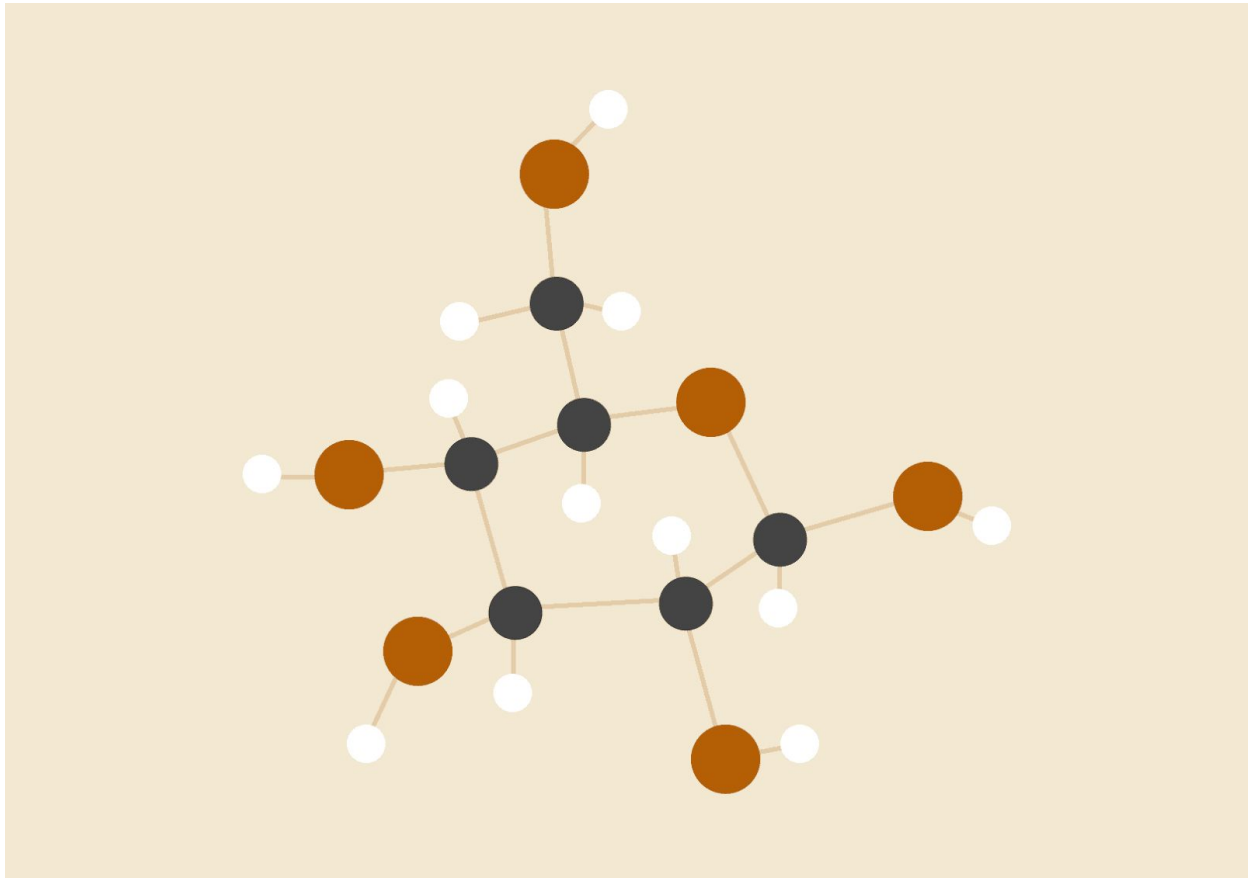


RAPPORT D'Architectures des Systèmes D'Information

Application de partage de recettes de cuisine



**Kake Abdoulaye - Mariam Konaté - Michel Orel
N'gatimo**

16/05/2018

Département Informatique - Polytech Tours – CLASSE DE 4e

Technologies utilisés :

- JAVA EE , avec Eclipse JEE
- MySql avec un serveur comme Easyphp
- Axis pour le web service SOAP
- Jersey pour le web service REST
- JDBC pour la communication de l'application java avec la base de données MySql
- Tomcat 8.5 pour le lancement côté serveur des webservices

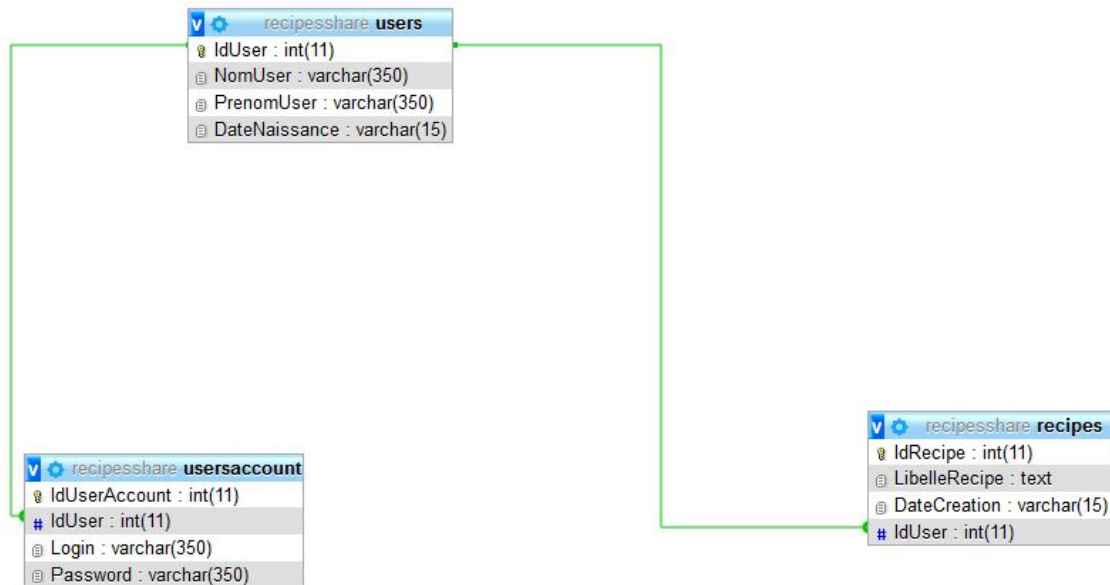
Chargement de la base de données :

Utilisation d'un serveur mysql serait utile pour le chargement de la base de données, dans notre cas nous avons utilisé EasyPHP. Il faut créer une base de données de nom : "recipesshare" et importer le script (.sql) qui se trouve dans le dossier ScriptDataBase dans lequel il y a des requêtes sql pour la création tables de la base de données et insérer quelque données .

SCHÉMA DE LA BASE DE DONNÉES :

la base de données "RecipesShare" est constitué de trois tables qui permettent de stocker les informations et données de l'application ,

- Users : qui contient les information nécessaire d'un utilisateur .
- UsersAccount : contient les informations permettant un utilisateur de se connecter à l'application .
- Recipes : permet de stocker toutes les recettes d'un utilisateur.



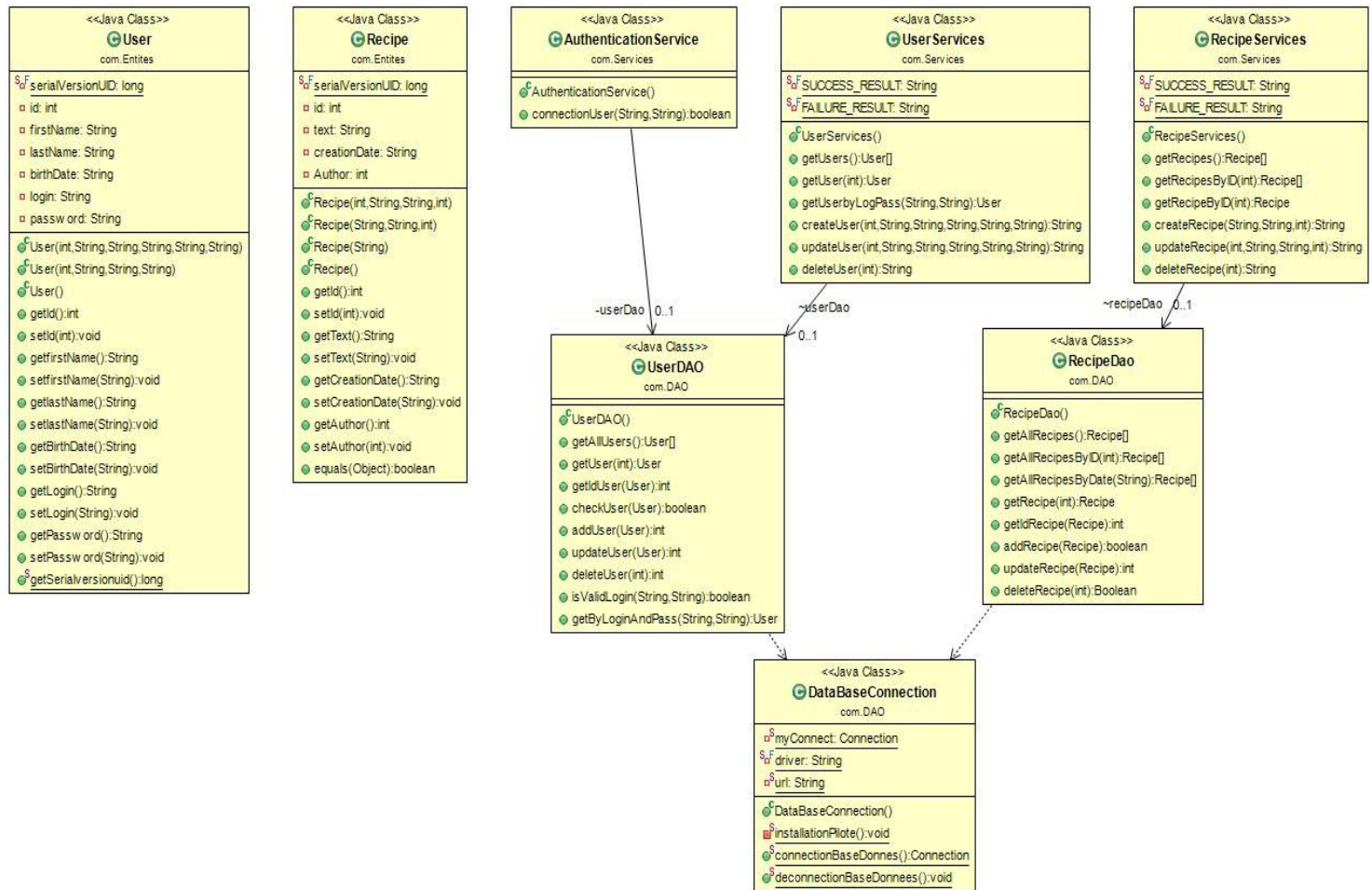
Structure de l'application :

Vous trouverez dans le dossier ... ,le code sources des deux webservices :
RecipesShareSOAP et RecipeShareRest avec leurs clients respectif :
ClientSoapRecipeShare et ClientRestRecipeShare .

1. L'architecture côté serveur :

Nous avons mis des classes DAO qui communiquent avec la base de données MySQL , des classes Services qui contiennent toutes les fonctionnalités de l'application qui font appeller à ces classes DAO en utilisant les Model : User , Recipe .

- **SOAP :**



● REST

L'architecture côté serveur est identique à celle utilisée par SOAP à part quelques nuances par rapport aux ressources qu'il faut annoter par xml ou json. De plus l'appel des services rest passe par le protocole HTTP donc des annotations sont nécessaires pour chaque service.

2. L'architecture côté Client :

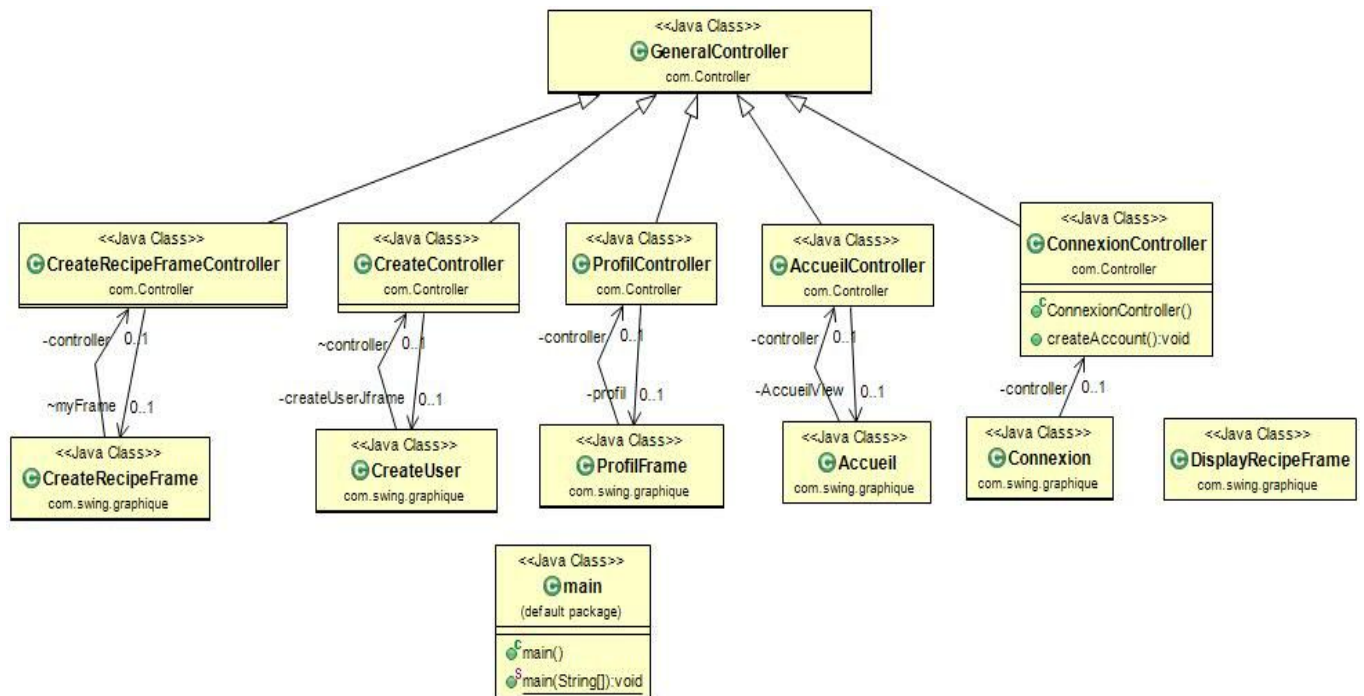
Les clients ont été faites avec une interface graphique Swing qui utilisent les webservices différemment en fonction de l'architecture REST ou SOAP.

Nous avons des classes graphique pour l'interaction avec l'utilisateur , des classes

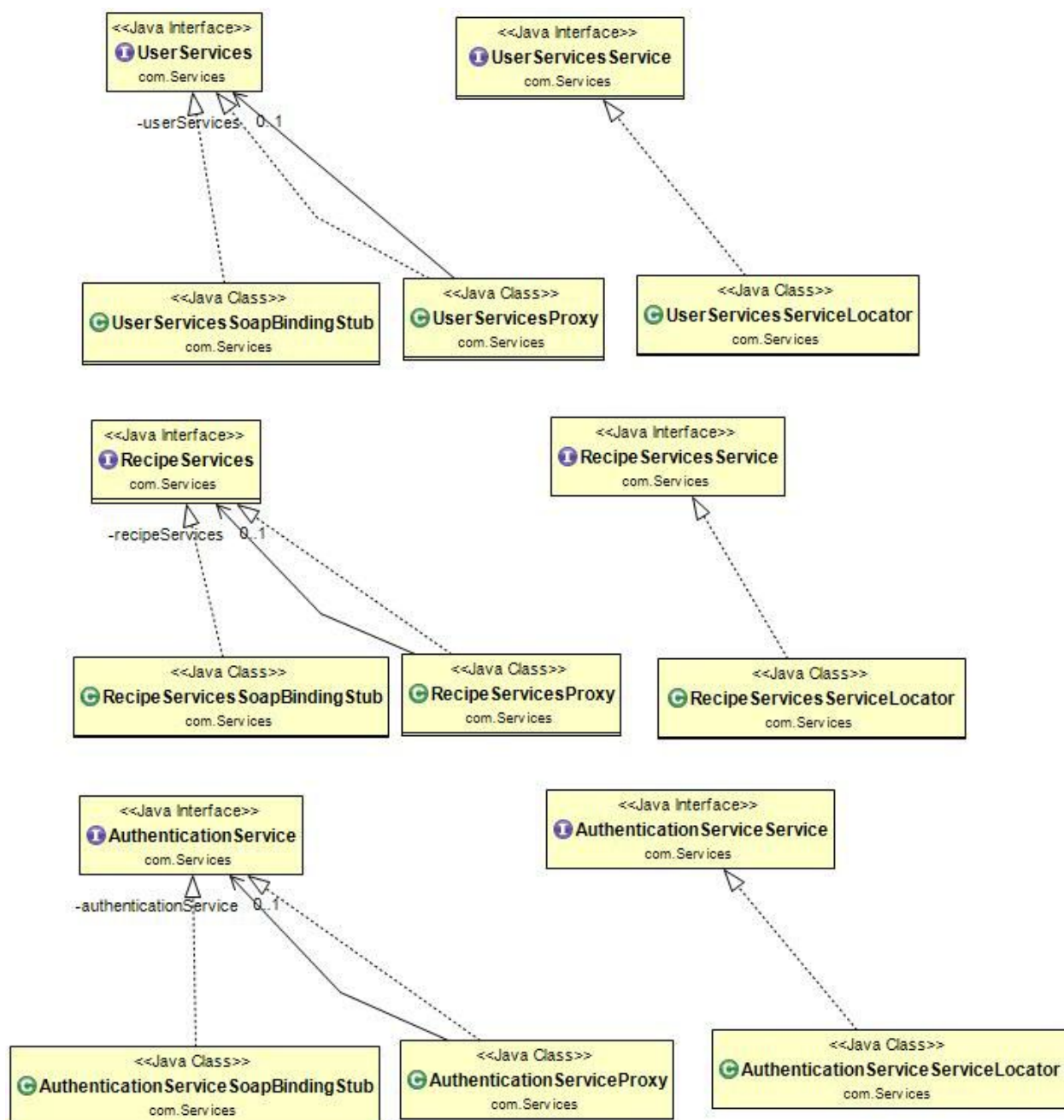
contrôleurs pour le chargement des données et la vérification des informations saisies par l'utilisateur et des classes services en fonction du webservice REST ou SOAP contient des méthodes de communications avec le webservice.

SOAP :

- Partie interface graphique Client



- Partie liaison webservice



REST

L'architecture côté client est identique à celle utilisée par SOAP , à la différence que nous avons des classes services contenant des méthodes avec des URIs pour lire des données sur le serveur.

FONCTIONNALITÉS DE L'APPLICATION :

Principale :

- Poster une recette de cuisine
- Toutes les recettes d'un utilisateur
- Toutes les recettes par rapport à une date

Secondaire :

- Connection utilisateur
- Création d'un utilisateur et d'une recette
- Suppression et éditer une recette

Lancement et utilisation de l'application :

SOAP

- Il faut en premier lieu importer la base de donnée depuis le script et l'appeler recipesshare.
- Puis dans le fichier src/com/RecipesUsers/DAO/DataBaseConnection modifiez l'attribut "url" en fonction de votre port utilisé puis dans la méthode connectionBaseDonnes modifiez les attributs de userInfo en fonction de vos login et password pour votre base de données.
- Ouvrir le client "ClientSoapRecipeShare" et le webservice "RecipeShareSOAP" avec eclipse IDE
- Pour le lancement du test de chaque client , il faut d'abord lancer le serveur Mysql contenant la base de données charger.
- Puis lancer le webservice RecipeShareSOAP sur le serveur Tomcat
- Après aller sur le projet client , exécuter la classe main dans defaultPackage contenant le main avec "Application java"
- Et interagir avec l'interface :
 - Connection avec un utilisateur (login : user , password : user) ou créer un compte user

- Accueil : affiche la liste de toutes les recettes avec leur auteurs , permet de trier par date de création de recette ou par nom d’auteur.
 - Double click sur recette pour l’afficher en entier avec toutes ces informations.
- Profile utilisateur : affiche la liste de toutes les recettes de l’utilisateur , permet de créer , éditer , supprimer une recette.
- déconnection de l’utilisateur.

REST

- Il faut en premier lieu importer la base de donnée depuis le script et l’appeler recipesshare.
- Puis dans le fichier src/com/RecipesUsers/DAO/DataBaseConnection modifiez l’attribut “url” en fonction de votre port utilisé puis dans la méthode connectionBaseDonnes modifiez les attributs de userInfo en fonction de vos login et password pour votre base de données.
- Ouvrir le client “ClientRestRecipeShare” et le webservice “RecipeShareRest” avec eclipse IDE
- Pour le lancement du test du client , il faut d’abord lancer le serveur Mysql contenant la base de données charger.
- Puis lancer le RecipeShareRest sur le serveur Tomcat
- Après aller sur le projet client ClientRestRecipeShare, exécuter la classe Test dans defaultPackage contenant le main avec “Application java”
- Et interagir avec l’interface :
 - Connection avec un utilisateur (login : user , password : user) ou créer un compte user
 - Accueil : affiche la liste de toutes les recettes avec leur auteurs , permet de trier par date de création de recette ou par nom d’auteur.
 - Double click sur recette pour l’afficher en entier avec toutes ces informations.
 - Profile utilisateur : affiche la liste de toutes les recettes de l’utilisateur , permet de créer , éditer , supprimer une recette.
 - déconnection de l’utilisateur.

Erreurs possible :

Après ouverture d’un des projets, s’il y a une erreur cela peut être dû à :

- l'utilisation du JRE ,JDK aller dans les propriétés puis le build path et librairies en fonction de votre machine utilisé celui qui vous convient.
- Le chemin vers les paquets axis déjà installé avec eclipse dans le dossier plugin allez dans les propriétés puis le build path et librairies pour mettre le bon chemin vers ces paquets.

CONCLUSION

Après avoir créé des webservices avec les api Rest et SOAP nous sommes plus à même à comprendre les différences. Nous nous sommes rendus compte qu'un client fonctionnait quasiment de la même manière peu importe la technologie utilisée, juste la façon d'appeler les services diffère.

RÉFÉRENCES

1. <https://dev.mysql.com/downloads/connector/j/>
2. https://www.tutorialspoint.com/restful/restful_methods.htm
3. <https://blog.philippbauer.de/testing-restful-services-java-best-practices/>