

GENIE LOGICIEL II

Travail pratique #3



14 DECEMBRE 2015

CABANNES JORDY / Noël DO VAN

Table des matières

[1. Diagramme de classes 2](#_Toc437819155)

[1.1. Package Logique 3](#_Toc437819156)

[1.2. Package DataMapper 4](#_Toc437819157)

[1.3. Package InterfaceUtilisateur et Context 5](#_Toc437819158)

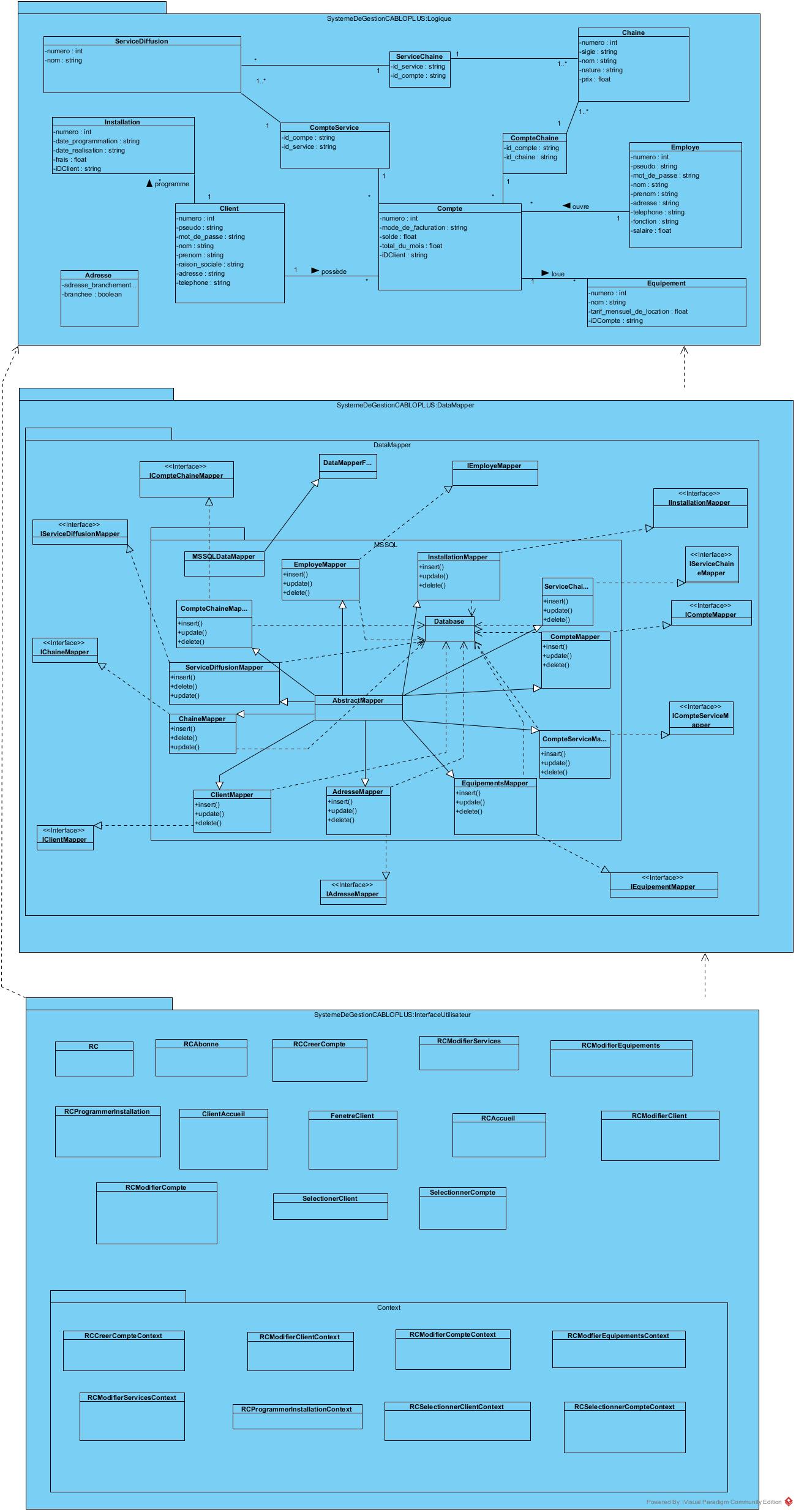
[2. La génération XSLT 6](#_Toc437819159)

[3. Code source et exécutable 7](#_Toc437819160)

[3.1. Guide d’utilisation 7](#_Toc437819161)

[4. Les scripts SQL 8](#_Toc437819162)

# 1. Diagramme de classes



Voici le diagramme de classes en entier.

Nous allons découper chaque ensemble (package) afin de mieux décrire leurs utilités.

Vous pouvez cependant voir comment ils sont reliés entre eux ici même.

Nous utilisons le patron de conception « **MVVM** » (Modèle Vue Vue-Modèle) en application WPF afin de séparer la logique de l’interface.

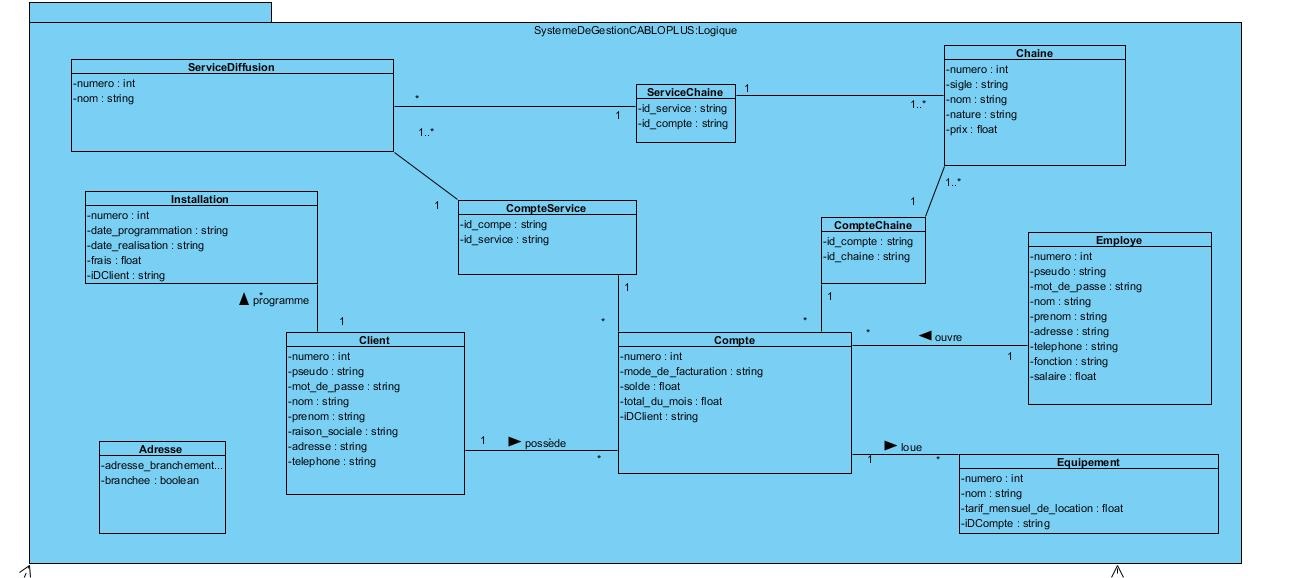
Le **Modèle** étant le package Logique.

La **Vue** étant le package InterfaceUtilisateur.

La **Vue-Modèle** étant le package Context à l’intérieur du package InterfaceUtilisateur.

Le package DataMapper servira à communiquer avec la base de données.

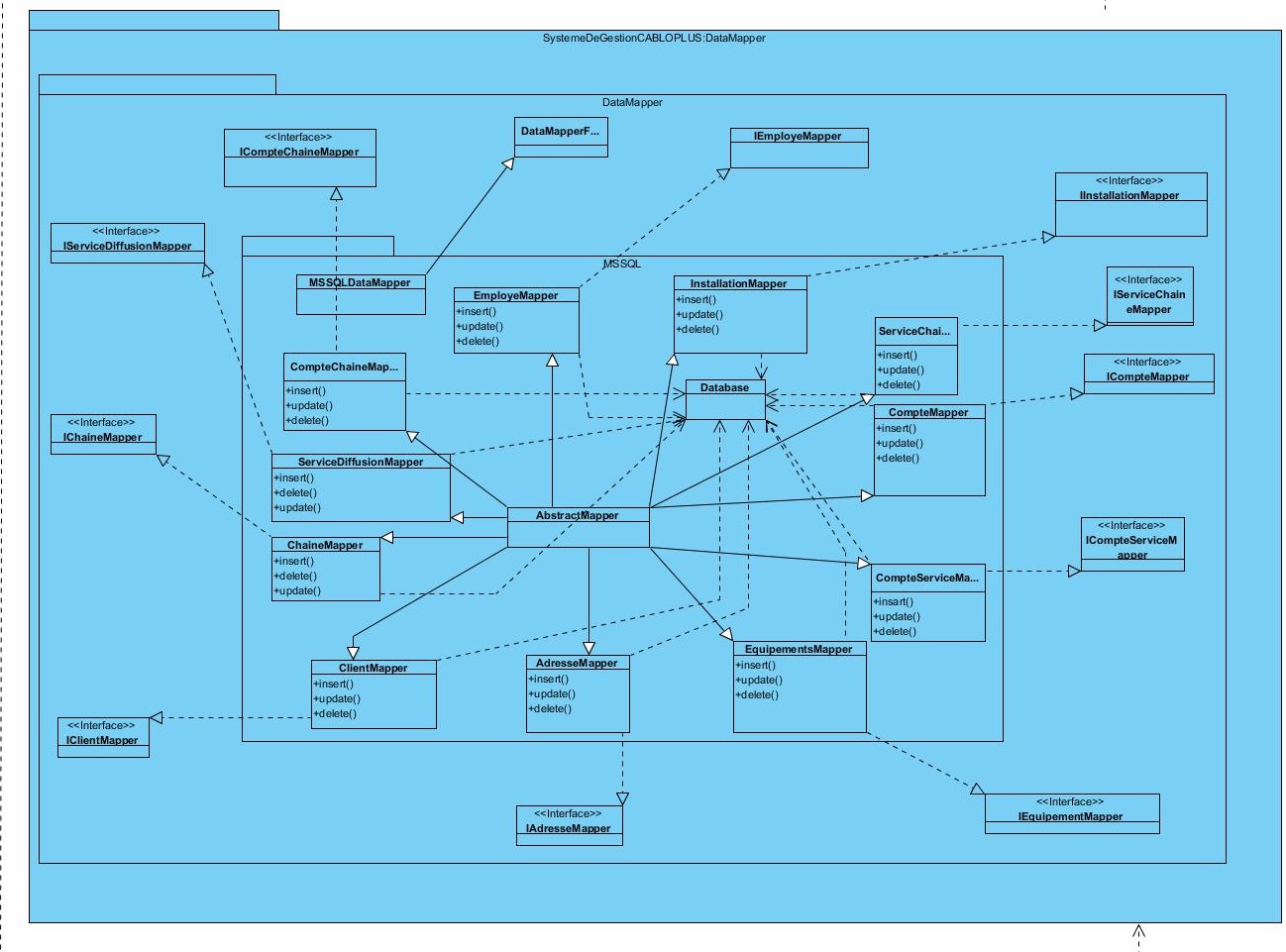
## Package Logique



Le package Logique regroupe les classes qui seront stockées dans la base de données. Nous utilisons ce package pour générer nos classes dans le programme grâce à la technologie **XSLT** (vous trouverez les fichiers nécessaires à la génération en annexe à ce document).

Il y a les classes de base comme ServiceDiffusion, Chaine, Installation etc et il y a les classes servant à faire le lien entre 2 classes, comme les classes CompteService (faisant le lien entre la classe ServiceDiffusion et Compte), CompteChaine et ServiceChaine.

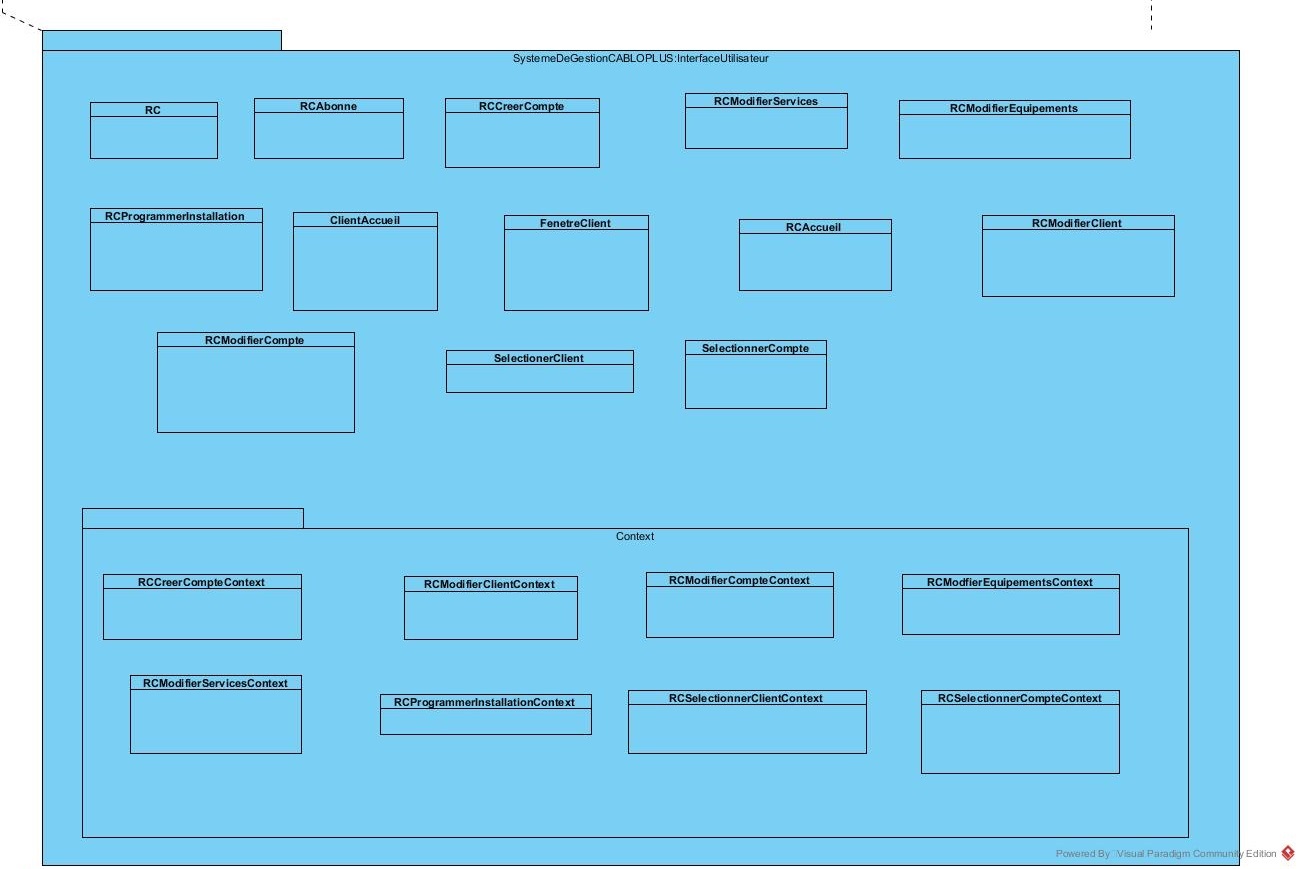
## Package DataMapper



Dans le package DataMapper se trouve les classes qui serviront à communiquer avec la base de données (seule l’interface **MSSQL** est implémentée ici).

Pour chaque classe de la partie logique, il y aura une interface et une classe Mapper. Celle-ci implémentera les méthodes utiles à la modification sur la base de données MSSQL.

## Package InterfaceUtilisateur et Context



Le package InterfaceUtilisateur regroupe les classes XAML/.cs qui serviront de vue au programme.

Dans cette version du logiciel, nous avons principalement implémenté la partie Employé RC (les classes commençant par RC) et seulement la page accueil de la partie Client (car les fonctionnalités sont globalement les mêmes que dans la partie Employé RC). Il y a également les 2 fenêtres SélectionnerClient et SélectionnerCompte qui implémentent le système de choix du client/compte lorsque l’on rentre dans un menu.

Le package Context regroupe les classes qui vont relier la Logique à l’InterfaceUtilisateur. En effet, pour chaque classe de la vue, il y aura une classe Context (vue-modèle) qui effectuera les calculs et stockera les informations nécessaires à la vue pour afficher les bonnes informations.

D’autres classes utiles au projet n’ont pas été listées dans les diagrammes de classes. Cependant, nous utilisons également en majorité le patron de conception **Singleton**, notamment pour les classes Context, et pour une classe UtilisateurConnecté (stocke les informations de l’utilisateur connecté en cours).

# La génération XSLT

Afin d’implémenter la partie logique du programme, nous utilisons la génération XSLT sur le package Logique du diagramme de classes.

Vous pourrez trouver les fichiers .xsl utiles à la génération, ainsi que le fichier project.xml dans le dossier « Generation » en annexe à ce document.

Il ne faut pas oublier de changer le chemin du binaire Saxon Transform par rapport à votre système dans le fichier cmd.bat.

# Code source et exécutable

Vous pourrez trouver les fichiers sources et exécutables dans le dossier « TP3\_genie\_noel » en annexe à ce document.

L’interface (boutons, vues) du TP2 a été implémentée afin d’avoir un aspect visuel identique. Cependant, **certains boutons n’ont pas été implémentés dans cette version du logiciel** car ils n’étaient pas assez pertinents par rapport aux travaux à réaliser de ce TP.

## Guide d’utilisation

Une fois le logiciel lancé, vous arrivez sur la page de connexion. Si vous utilisez les scripts SQL servant à générer la base de données liées à ce projet (que nous allons voir plus tard dans ce document Word). Les utilisateurs possibles seront :

* Identifiant : cl1, Mot de passe : toto, pour se connecter en tant que client.
* Identifiant : cl2, Mot de passe : toto, pour se connecter en tant que client.
* Identifiant : emp1, Mot de passe : toto, pour se connecter en tant qu’employé RC.

D’autres utilisateurs pourront être générés dans le programme depuis l’interface Employé RC par la suite (Faire un abonnement).

Si l’on rentre des identifiants clients, le programme redirigera l’utilisateur vers la partie Client. Cette partie recense uniquement la page d’accueil dans cette version du logiciel. Les fonctionnalités sont néanmoins quasiment identiques à la partie Employé RC.

Si l’on rentre des identifiants employé RC, le programme redirigera l’utilisateur vers la partie RC. C’est dans cette partie que l’utilisateur pourra alors faire :

* les abonnements
* la gestion des comptes
* l’ajout d’un compte
* la modification d’un compte
* la modification d’un profil client
* la modification des services
* la modification des équipements
* la planification d’une installation

Il utilisera les menus en haut afin de naviguer entre ces multiples pages.

Veuillez noter que pour effectuer la plupart des tâches ci-dessus, il faudra d’abord ajouter un compte à un abonné (le script SQL ne le fait pas de base car nous n’avons pas la possibilité d’avoir à l’avance, l’id client (GUID) utile à l’ajout d’un compte)

# Les scripts SQL

Vous pourrez trouver les scripts SQL à exécuter lors de l’implémentation de ce système dans le dossier « ScriptSQL » se trouvant dans le dossier « TP3\_genie\_noel » en annexe à ce document.

Nous avons créé 4 scripts, voici leurs utilisations normales à exécuter **dans l’ordre** :

* Script pour tout delete.sql : permet de supprimer le contenu de la base de données avant de commencer. On s’assure ainsi que nous partons sur une base saine.
* Script pour generer les tables.sql : permet de créer les tables que nous utiliserons par la suite dans le système.
* Script pour remplir les tables.sql : permet d’insérer du contenu dans la base de données afin d’avoir une application de base fonctionnelle.
* Script pour voir les tables.sql (optionnelle) : permet de voir les tables présentes dans la base de données.