

Application Desktop pour la Gestion de la Facturation d'eau des forages

Dans ce travail, nous vous demandons de développer en C++ une application pour la gestion d'un forage. L'application permet la gestion des abonnés, des compteurs, et de la facturation. Pour les interfaces graphiques, je vous suggère d'utiliser Qt ou wxWidgets. Vous allez trouver beaucoup de tutoriel sur internet qui vont vous montrer comment les utiliser.

L'application aura comme principales fonctionnalités :

- **La Gestion des abonnés**
- **La Gestion des compteurs**
- **La Gestion de la facturation**

I. Gestion des abonnés

Un abonné (ou client) est personne désirant d'aller s'abonner pour bénéficier d'un compteur et pouvoir l'utilisée. Notre application doit nous permettre de faire les taches suivantes :

- Créer un abonné (client) : c'est pour ajouter un client dans la table **Client**
- Afficher la liste des abonnés :
- Modifier les informations concernant un abonné
- Tri des abonnés par collectivité : la possibilité de pouvoir ranger par ordre croissant les abonnés selon leurs collectivités auxquelles qu'elles appartiennent.
- Afficher la liste des fiche abonnée/compteur cela permet de voir qu'un tel client à s'abonner un compteur ainsi d'identifier son numéro compteur qui lui est assigné.

II. Gestion des compteurs

L'application doit permettre :

- D'ajouter des compteurs disponibles
- D'archiver ou transférer un compteur.
- De faire un abonnement : assigner un compteur à un abonnée (Faire une insertion dans la table **Abonnement**)

Note : Un abonné peut avoir plusieurs compteurs dans différents endroits.

III. Gestion de la facturation

Établir la facturation du client à partir de la consommation.

Permettre d'effectuer le paiement lié à une facture.

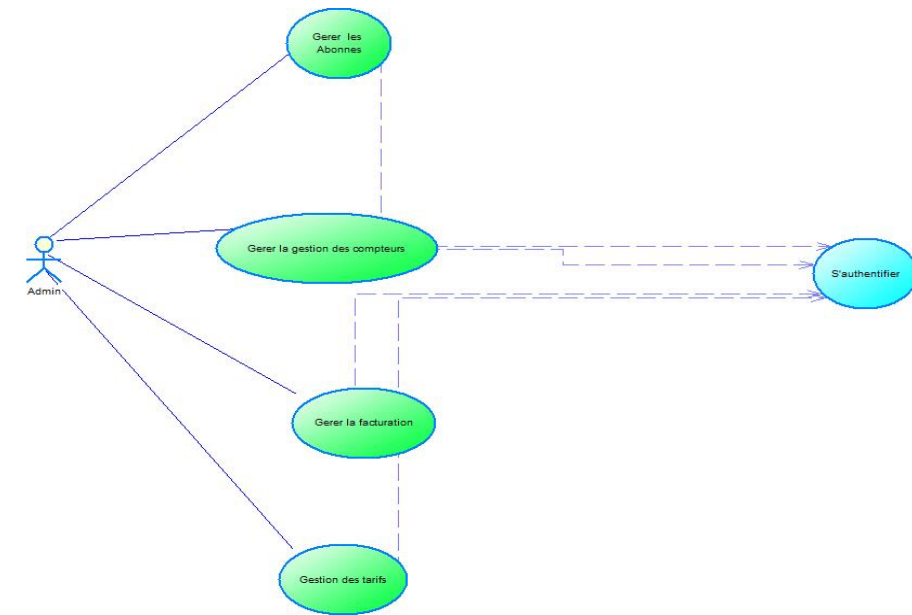


Diagramme de cas d'utilisation

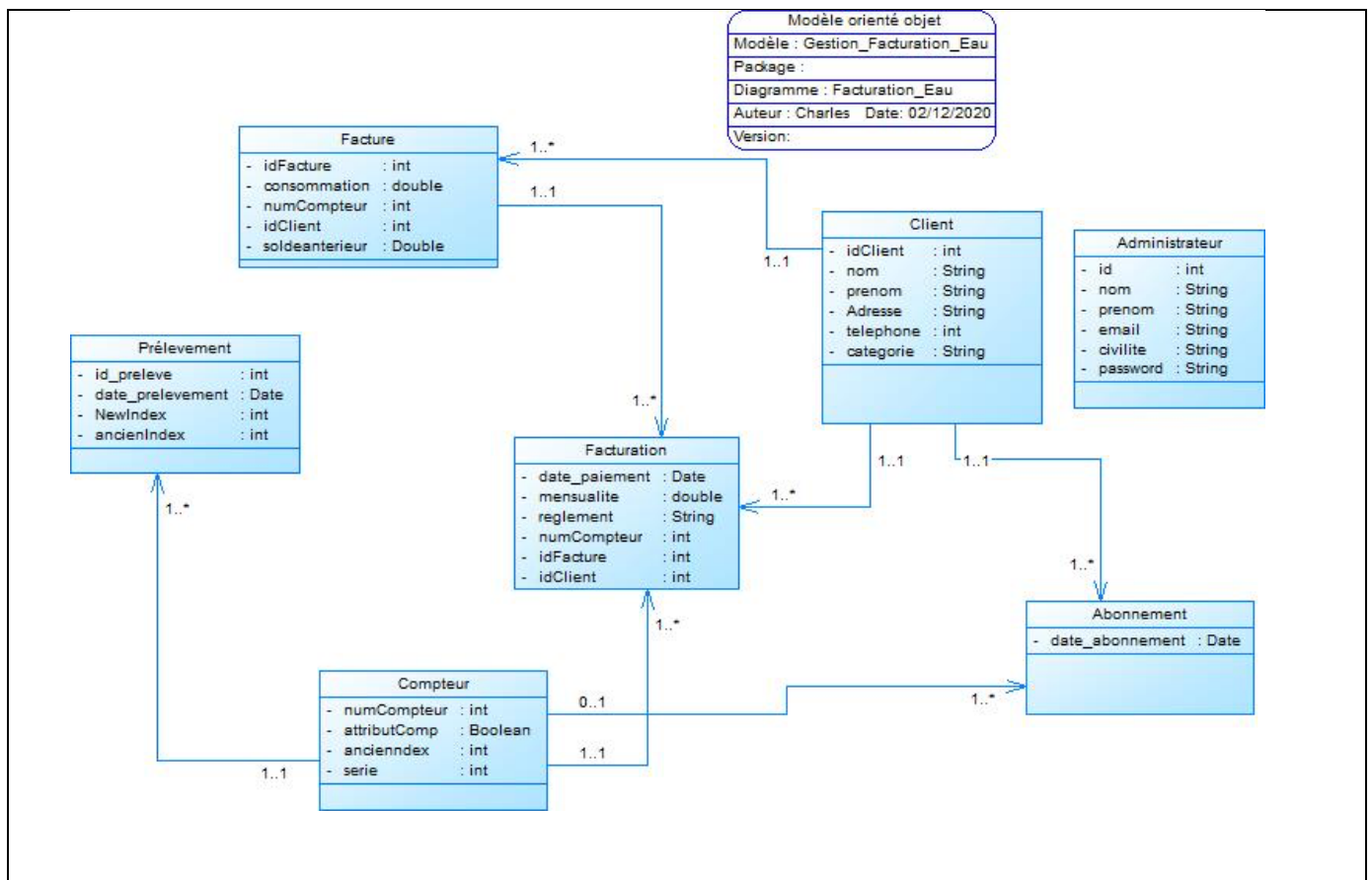


Diagramme de classe de l'application

Modèle logique des données(MLD)

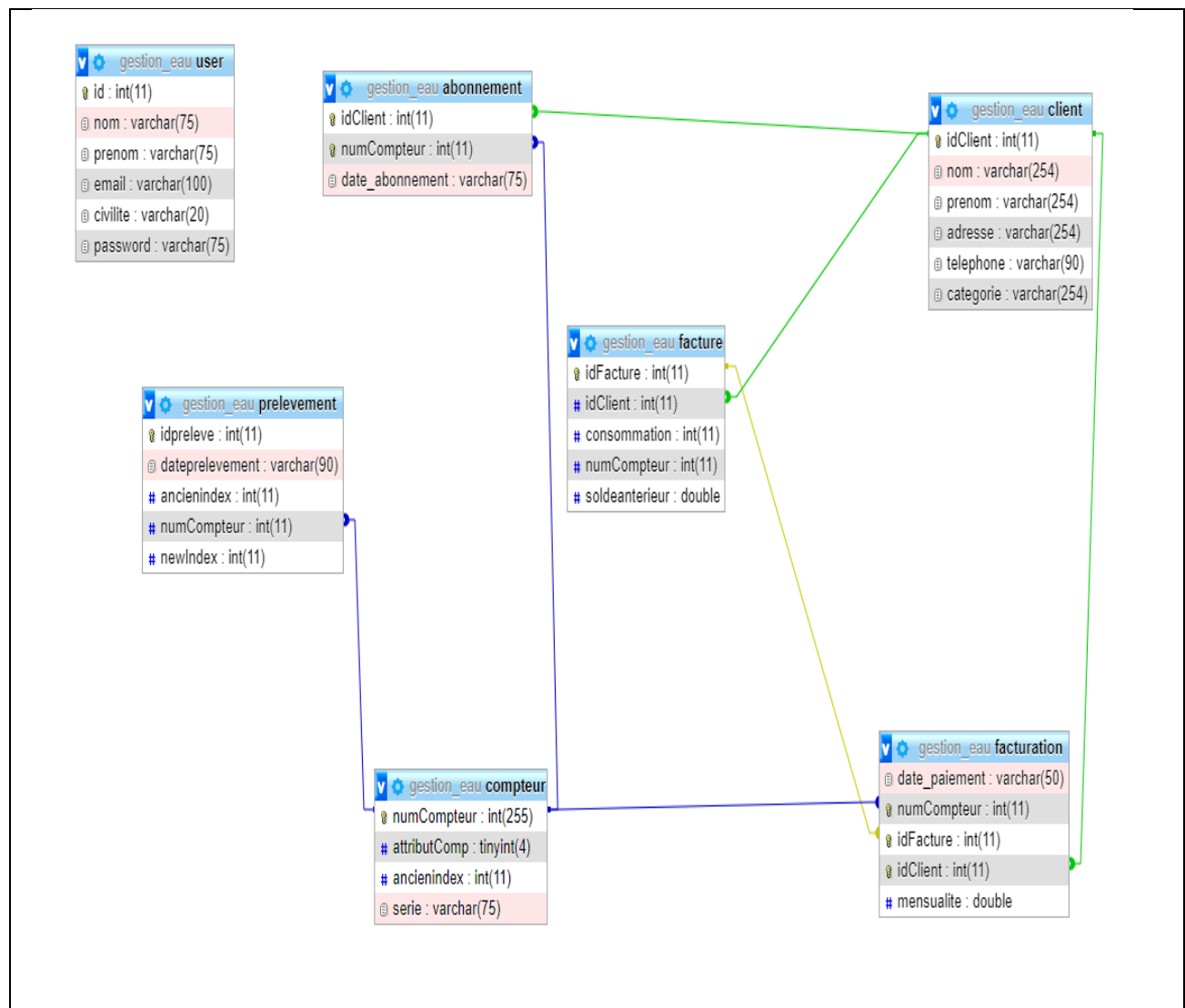
Le MLD est la modélisation logique des données qui tient compte du niveau organisationnel des données. Il s'agit d'une vue logique en terme d'organisation de données nécessaire à un traitement.

Explication de quelques termes : la table **User** désigne l'agent ou l'administrateur c'est-à-dire celui qui s'occupe de l'application pour pouvoir gérer de manière structurel les abonnés. En partant du diagramme de classes présenté ci-dessus, les tables de la base de données relationnelle résultantes sont :

En vue global nous avons sept(7) tables dans la base de données qui sont repartis comme suit suivant la migration des clés étrangères :

- User (**id**, nom, prénom, email, civilite, password)
- Client (**idClient**, nom, prenom, adresse, telephone)

- Compteur (numCompteur, attributComp, ancienindex, serie)
- Abonnement (#idClient, #numCompteur, date_abonnement)
- Facturation (date_paiement, #numCompteur, #idFacture, #idClient, mensualite)
- Prelevement(idpreleve, dateprelevement, #numCompteur, newIndex, ancienIndex)
- Facture (idFacture, #numCompteur, #idClient, soldeanterior, consommation)



Modele de conception de données