Projet C# Transconnect

Quentin Guignard TDF

Pour ce projet j’ai utilisé une base de données avec SQL, vous retrouverez les tables ainsi que les insertions de données et les essaies de requêtes dans la solution (Le mdp est « root »)

SERVER=localhost;PORT=3306;DATABASE=TransConnect;UID=root;PASSWORD=root

(Voici la ligne de code qui connecte visual studio à MySql) Il vous faudra peut-être installer un package dans visual studio, bien peupler la table pour éviter tout problème

Comment naviguer dans la solution une fois lancée :

* La solution comprend une partie de gestion par le pdg pour y accéder veuiller rentrer « a » dans le mail et « azerty » en mdp (la solution est faite pour que seul le mdp azerty fasse entrer dans la partie gestion)
* Pour la partie client vous pouvez soit vous créer un compte soit en utiliser un existant, vous trouverez les mails et mdp dans la bdd
* Lors de l’affichage de l’organigramme cliquer sur les noms pour voir les postes

1. Les différentes class :

Personne : class abstraite dont va hériter la classe salarie et la classe client

Véhicule : class abstraite dont vont hériter les classes Voiture, Camionnette, Poids-Lourds

Hiérarchie : Joue le rôle de classe nœud pour la construction de l’organigramme elle prend en argument un Salarié et une liste de Hiérarchie

Organigramme : class qui prend en argument le pdg de l’entreprise (Hiérarchie)

Ville : class qui va permettre, à partir du fichier csv, de remplir, à chaque lancement de l’application, la Table Plus\_Court\_Chemin de la bdd comme nous le souhaitons

Commande : class qui contient les informations des commandes

Il y a une multitude d’autres class qui représentent une fenêtre WinForms, il y a bien sur des méthodes essentielles contenues dans ces dernières

**Une grande partie des méthodes est annotée dans la solution si vous ne trouvez pas assez d’explications dans ce document.**

Interface : IDisponibilité, associée aux class Véhicule et Salarié elle permet de savoir si ces derniers sont disponible à une date précise

Délégation : « OperationSubordonne » dans la class Hiérarchie qui permet soit d’ajouter une subordonné soir d’en supprimer un.

Sort et Find\_All : A la fin de la class Hiérarchie

Arbre n\_aire : class organigramme

Dijkstra : Dans la class Ville

Innovations :

* Possibilité d’avoir un espace Client
* L’organigramme, même s’il est demandé je considère sa réalisation comme une innovation car très proche de la photo dans le sujet
* Possibilité de voir tous les véhicules du garage
* Pouvoir réaliser une commande de la page client et de la page PDG
* Ne pas pouvoir créer une commande avant la date d’aujourd’hui (fonction sur les numericupanddown)
* Pouvoir promouvoir certains salariés
* Utilisation d’une bdd SQL
* Affichage WinForms

Test Unitaire :

Pour la partie Test Unitaire, aucun des modèles de projet test unitaire n’a réussis à fonctionner à cause d’un écart dans la version (v4.7.2 et net8.0) cependant j’ai quand même écrit les fonctions dans la class [UnitTest1] en espérant que cela soit suffisant