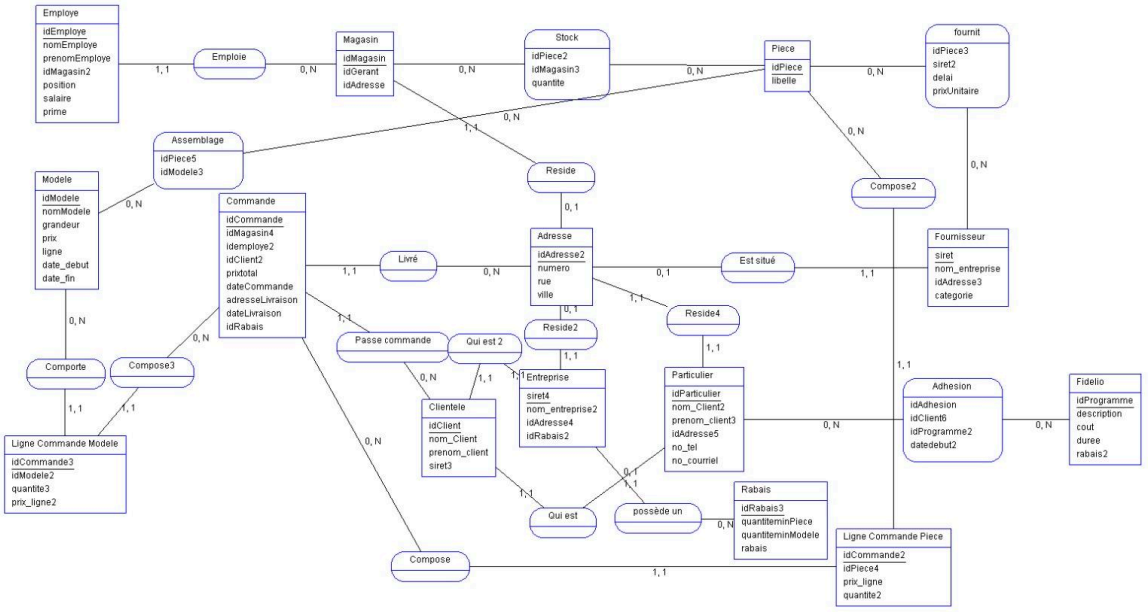
RAPPORT DE PROJET : VELOMAX

I) Organisation de la BDD



Voici la représentation de notre base de donnée avec un diagramme E/A. Ce schéma a été fait avec l'outil gratuit analyse SI.
Ce schéma est la version finale de l'organisation de notre BDD. Il faut savoir qu'il y a eu beaucoup d'itérations et de modifications au fil du code afin de pouvoir obtenir cela.

Notre base de données présente plusieurs classes majeures, comme commande, entreprise, adresse, client, pièce, modèle, fidélité ou encore ligne de commande. Il est inutile de vous donner içi l'entièreté de nos classes, elles sont toutes indiquées sur le schéma.

II) Organisation du code C#

Après avoir lié notre base de donnée SQL avec notre IDE (ici, Visual Studio), nous avons créé autant de bases que de classe et d'association au sein de notre code C#. Après quelques essais, nous avons choisi d'effectuer une présentation sous Forms. La première étape a donc été d'apprendre ce nouveau "language" afin de pouvoir produire un rendu de qualité.
Le forms était un véritable challenge technique, c'est une couche de difficultée supplémentaire qui nous a coûté beaucoup de temps. Cependant, nous avons une belle interface interactive et nous avons appris une nouvelle technique de développement.

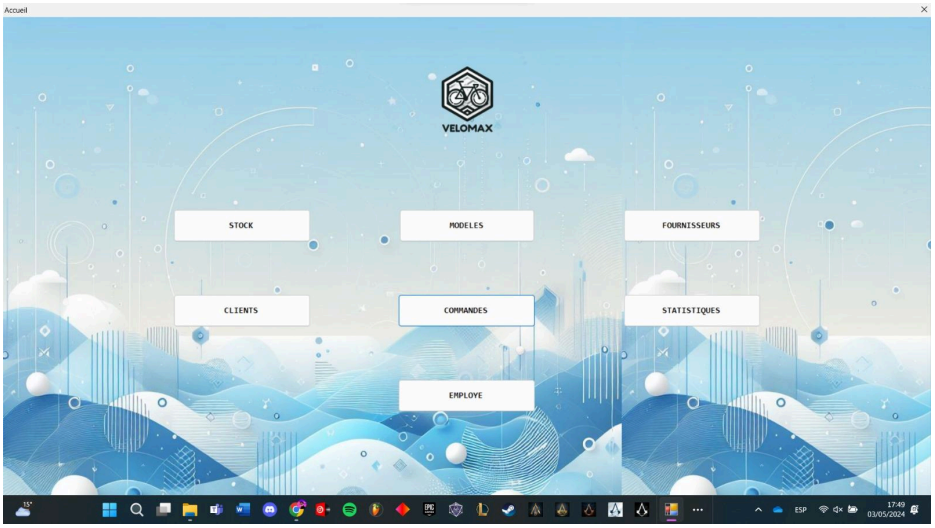
Pour ce qui est de notre code, nous avons donc plus de 28 différentes classes.

```
public void LoadBestClient(MySqlConnection connexion)
        {
            string requete = "SELECT PARTICULIER.nom_client, PARTICULIER.prenom_client,
             COMMANDE.idClient, SUM(COMMANDE.prix_total) AS total FROM COMMANDE JOIN PARTICULIER ON
             PARTICULIER.idParticulier = COMMANDE.idClient GROUP BY COMMANDE.idClient ORDER BY total
             DESC LIMIT 1;";
            MySqlCommand cmd = new MySqlCommand(requete, connexion);

            using (MySqlDataReader reader = cmd.ExecuteReader())
            {
                listBox5.Items.Clear();
                while (reader.Read())
                {
                    string nomC = reader["nom_client"].ToString();
                    string prenomC = reader["prenom_client"].ToString();
                    string montant = reader["total"].ToString();


                    listBox5.Items.Add($"{nomC} {prenomC} {montant} €");
                }
            }
        }
```

Voici un exemple d'une fonction permettant d'afficher le nom,prénom et le montant total dépensé par un client dans un listbox (dans le menu statistique). Il est composé d'une requête qui récupère les informations du client puis il l'affiche via la commande listBox.Items.Add();

Nous avons également plusieurs méthodes permettant de récupérer diverses informations utiles à notre application. Notre code est assez complexe et

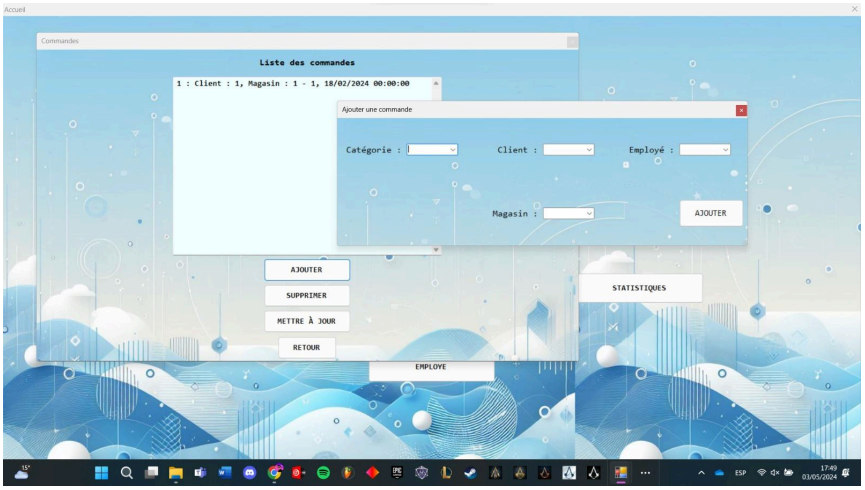
III) Présentation de l'application



Nous avons différents sous-menus disponibles pour le client comme le menu commande, employé ou les statistiques.

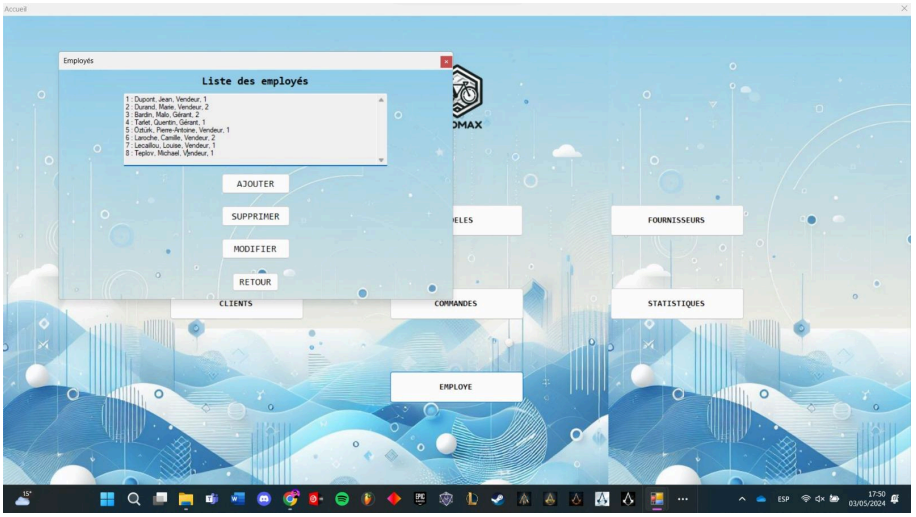
Commande :

- Permet d'ajouter, de supprimer ou de modifier une commande déjà existante. Vous pouvez indiquer la pièce ou le modèle que vous souhaitez ajouter, la quantité, le magasin et le client. Vous pouvez également assigner un employé pour que celui-ci fasse la commande.



Employé :

- Vous pouvez, comme pour les commandes, ajouter, supprimer ou modifier un personnel. Vous pouvez changer le nom, prenom, le rôle de l'employé, le magasin dans lequel il travaille mais également son salaire.



Statistique :

- Le menu de statistique est rempli de plusieurs fonctions analytiques permettant d'avoir plusieurs données sur Vélomax. Ainsi, vous pouvez avoir le revenu moyen de chaque employé, son bonus, le magasin le plus rentable ou le client qui a dépensé le plus.
  Nous avons également d'autres menus mais nous vous laissons le plaisir de les découvrir cher client !