

## Rapport livrable 2 :

PHILIP Samuel

PERINEL Maxence

PERCIO Sarah

Avancement :

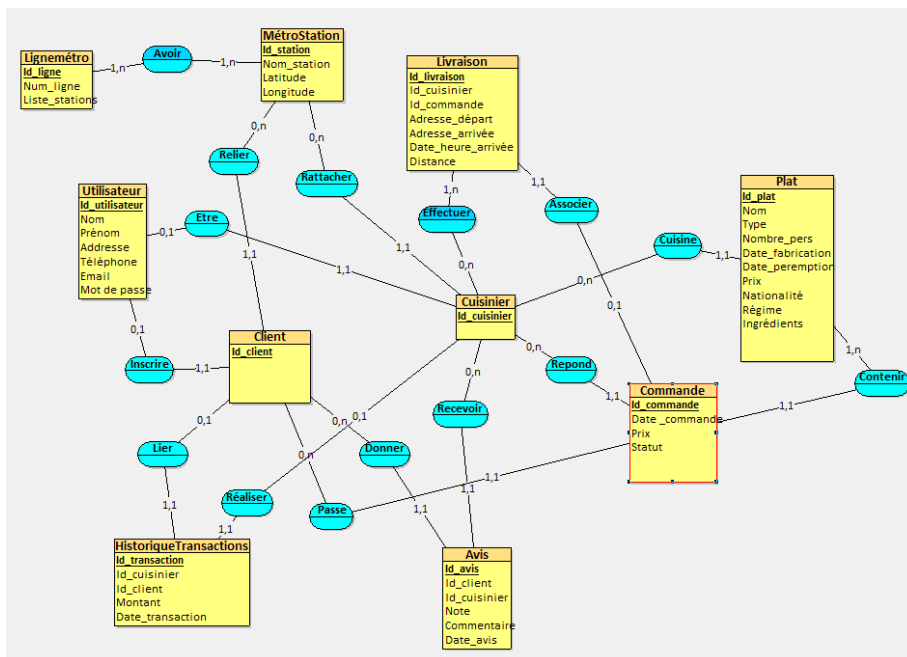
La majorité de la logique permettant le fonctionnement de l'application est fonctionnel.

Pour visualiser notre travail sur le graphe :

Lancer l'app => S'inscrire et crée un profil cuisinier => Se connecter au profil cuisinier que l'on vient de crée => Proposer un nouveau plat => S'inscrire et crée un profil client => Voir les plats disponibles => En choisir un pour crée une commande.

Le mode développeur n'est pas encore fonctionnel.

Schéma entité association modifié pour mieux répondre à nos besoins :



Script de création de la base :

```

CREATE TABLE Utilisateur(
    Id_utilisateur VARCHAR(50),
    Nom VARCHAR(50),
    Prénom VARCHAR(50) NOT NULL,
    Adresse VARCHAR(70) NOT NULL,
    Téléphone INT NOT NULL,
    Email VARCHAR(70) NOT NULL,
    Mot_de_passe VARCHAR(50) NOT NULL,
    PRIMARY KEY(Id_utilisateur)
);

CREATE TABLE MétroStation(
    Id_station VARCHAR(50),
    Nom_station VARCHAR(50) NOT NULL,
    Latitude VARCHAR(50) NOT NULL,
    Longitude VARCHAR(50) NOT NULL,
    PRIMARY KEY(Id_station)
);

CREATE TABLE Lignemétro(
    Id_ligne VARCHAR(50),
    Num_ligne INT NOT NULL,
    Liste_stations VARCHAR(2000) NOT NULL,
    PRIMARY KEY(Id_ligne)
);

CREATE TABLE Cuisinier(
    Id_cuisinier VARCHAR(50),
    Id_station VARCHAR(50) NOT NULL,
    Id_utilisateur VARCHAR(50) NOT NULL,
    PRIMARY KEY(Id_cuisinier),
    UNIQUE(Id_utilisateur),
    FOREIGN KEY(Id_station) REFERENCES MétroStation(Id_station),
    FOREIGN KEY(Id_utilisateur) REFERENCES Utilisateur(Id_utilisateur)
);

CREATE TABLE Client(
    Id_client VARCHAR(50),
    Id_station VARCHAR(50) NOT NULL,
    Id_utilisateur VARCHAR(50) NOT NULL,
    PRIMARY KEY(Id_client),
    UNIQUE(Id_utilisateur),
    FOREIGN KEY(Id_station) REFERENCES MétroStation(Id_station),
    FOREIGN KEY(Id_utilisateur) REFERENCES Utilisateur(Id_utilisateur)
);

CREATE TABLE Plat(
    Id_plat VARCHAR(50),
    Nom VARCHAR(50) NOT NULL,
    Type VARCHAR(50) NOT NULL,
    Nombre_pers INT NOT NULL,
    Date_fabrication DATE NOT NULL,
    Date_peremption DATE NOT NULL,
    Prix DECIMAL(15,2) NOT NULL,
    Nationalité VARCHAR(50) NOT NULL,
    Régime VARCHAR(50) NOT NULL,
    Ingrédients VARCHAR(50) NOT NULL,
    Id_cuisinier VARCHAR(50) NOT NULL,
    PRIMARY KEY(Id_plat),
    FOREIGN KEY(Id_cuisinier) REFERENCES Cuisinier(Id_cuisinier)
);

```

```

CREATE TABLE Commande(
  Id_commande VARCHAR(50),
  Date__commande DATE NOT NULL,
  Prix DECIMAL(15,2) NOT NULL,
  Statut VARCHAR(50),
  Id_plat VARCHAR(50) NOT NULL,
  Id_cuisinier VARCHAR(50) NOT NULL,
  Id_client VARCHAR(50) NOT NULL,
  PRIMARY KEY(Id_commande),
  FOREIGN KEY(Id_plat) REFERENCES Plat(Id_plat),
  FOREIGN KEY(Id_cuisinier) REFERENCES Cuisinier(Id_cuisinier),
  FOREIGN KEY(Id_client) REFERENCES Client(Id_client)
);

CREATE TABLE Livraison(
  Id_livraison VARCHAR(50),
  Id_cuisinier VARCHAR(50),
  Id_commande VARCHAR(50),
  Adresse_départ VARCHAR(50) NOT NULL,
  Adresse_arrivée VARCHAR(50) NOT NULL,
  Date_heure_arrivée DATETIME NOT NULL,
  Distance DECIMAL(15,2) NOT NULL,
  Id_commande_1 VARCHAR(50) NOT NULL,
  PRIMARY KEY(Id_livraison),
  UNIQUE(Id_commande_1),
  FOREIGN KEY(Id_commande_1) REFERENCES Commande(Id_commande)
);

CREATE TABLE HistoriqueTransactions(
  Id_transaction VARCHAR(50),
  Id_cuisinier VARCHAR(50),
  Id_client VARCHAR(50),
  Montant DECIMAL(15,2) NOT NULL,
  Date_transaction DATE NOT NULL,
  Id_cuisinier_1 VARCHAR(50) NOT NULL,
  Id_client_1 VARCHAR(50) NOT NULL,
  PRIMARY KEY(Id_transaction),
  UNIQUE(Id_cuisinier_1),
  UNIQUE(Id_client_1),
  FOREIGN KEY(Id_cuisinier_1) REFERENCES Cuisinier(Id_cuisinier),
  FOREIGN KEY(Id_client_1) REFERENCES Client(Id_client)
);

CREATE TABLE Avis(
  Id_avis VARCHAR(50),
  Id_client VARCHAR(50),
  Id_cuisinier VARCHAR(50),
  Note INT NOT NULL,
  Commentaire VARCHAR(200),
  Date_avis DATE NOT NULL,
  Id_client_1 VARCHAR(50) NOT NULL,
  Id_cuisinier_1 VARCHAR(50) NOT NULL,
  PRIMARY KEY(Id_avis),
  FOREIGN KEY(Id_client_1) REFERENCES Client(Id_client),
  FOREIGN KEY(Id_cuisinier_1) REFERENCES Cuisinier(Id_cuisinier)
);

```

```

CREATE TABLE Effectuer(
  Id_cuisinier VARCHAR(50),
  Id_livraison VARCHAR(50),
  PRIMARY KEY(Id_cuisinier, Id_livraison),
  FOREIGN KEY(Id_cuisinier) REFERENCES Cuisinier(Id_cuisinier),
  FOREIGN KEY(Id_livraison) REFERENCES Livraison(Id_livraison)
);

CREATE TABLE Avoir(
  Id_station VARCHAR(50),
  Id_ligne VARCHAR(50),
  PRIMARY KEY(Id_station, Id_ligne),
  FOREIGN KEY(Id_station) REFERENCES MétroStation(Id_station),
  FOREIGN KEY(Id_ligne) REFERENCES Lignemétro(Id_ligne)
);

```

Retour sur les algorithmes de recherche :

Sur ma machine les 3 algorithmes s'exécutent de manière instantanée donc nous avons utilisé des stop Watch pour déterminer le plus rapide le tout renforcé par nos recherches sur la complexité des algorithmes.

On note S le nombre de sommet et A le nombre d'arrête

Dijkstra complexité en  $O((S+A)\log(S)) \Rightarrow$  Après nos tests, exécution la plus rapide

Bellman Ford complexité en  $O(S*A) \Rightarrow$  Presque aussi rapide que Dijkstra sur nos données la différence de temps d'exécution est extrêmement faible.

Floyd Warshall complexité en  $O(S^3) \Rightarrow$  Beaucoup plus long que les deux autres