Rapport d'analyse et conception

Binôme :

Melissa LARBI
Augustin MAILLE

Le cahier des charges:

Description:

Notre projet consiste à implémenter une application pour gérer un service de livraison de repas. Cette application permet à un client de se faire livrer un repas où il le souhaite et quand il le souhaite.

Le client:

Le client doit pouvoir se créer un compte sur l'application pour se connecter avec des identifiants uniques et passer des commandes auprès de restaurants par la suite. Un client possède un numéro permettant de l'identifier, un nom, un prénom, un numéro de téléphone(facultatif, s'il souhaite être appelé quand le livreur est arrivé), une adresse mail(pour l'envoi de la facture).

Le restaurant:

Un restaurant est identifié par un numéro et il possède un nom, une adresse ainsi qu'un numéro de téléphone. Le restaurant propose différents articles à vendre sur l'application.

Les articles:

Un article est identifié par un numéro, un nom, un prix et une description. Chaque article possède un type tel que : « Boisson, entrée... ».

Type des articles:

Pour pouvoir regrouper les articles et faciliter la prise de commande pour le client. Un type est identifié par un numéro et un nom.

La commande: Le restaurant prépare la commande qui contient certains articles.

Une commande est identifiée par un numéro de commande et possède une date, l'adresse de livraison, la distance entre le restaurant et le client ainsi que la distance entre le restaurant et le livreur.

État de la commande:

Une commande est associée à un état(identifié par un numéro et possède un nom).

Quand le restaurant a terminé la préparation de la commande son état passe à « prête » sinon elle est en cours de préparation, une fois un livreur accepte de la livrer, son état est «en livraison». Une fois qu'elle est arrivée chez le client, son état est «livrée».

La facture:

À chaque commande est associée une facture, identifiée par un numéro de facture. Elle renseigne le montant de la commande(somme des prix unitaires de chaque articles achetés et des

frais de livraison), les frais de livraison pour détailler le montant payé.

Le livreur et état livreur:

À l'issue de la préparation de la commande, cette dernière sera livrée par un livreur. Un livreur possède aussi un compte sur l'application et renseigne son nom, son prénom, son email, son rib, son numéro de téléphone. Lorsqu'il se connecte sur l'application, il renseigne sa position, et son état passe de hors service à en attente. Une fois qu'il accepte de livrer une commande, son état passe à en livraison.

Comment sont calculés les frais de livraison?

Lorsque le Client se connecte sur l'application il renseigne sa position permettant ainsi de calculer la distance entre l'adresse de livraison et le restaurant. Cette distance permet de calculer les frais de livraison qui seront reversés au livreur. Une fois que le livreur accepte une commande la distance entre sa position et le restaurant sera calculé et il recevra l'itinéraire de la commande. Ainsi il sera possible d'avoir des statistiques telles que les distances parcourues et la rémunération sur 30 jours pour les livreurs.

Comment chercher les livreurs les plus proches d'un restaurant?

À chaque fois que le client commande à un restaurant, on recherche les livreurs disponibles et les plus proches pour leur proposer de livrer la commande.

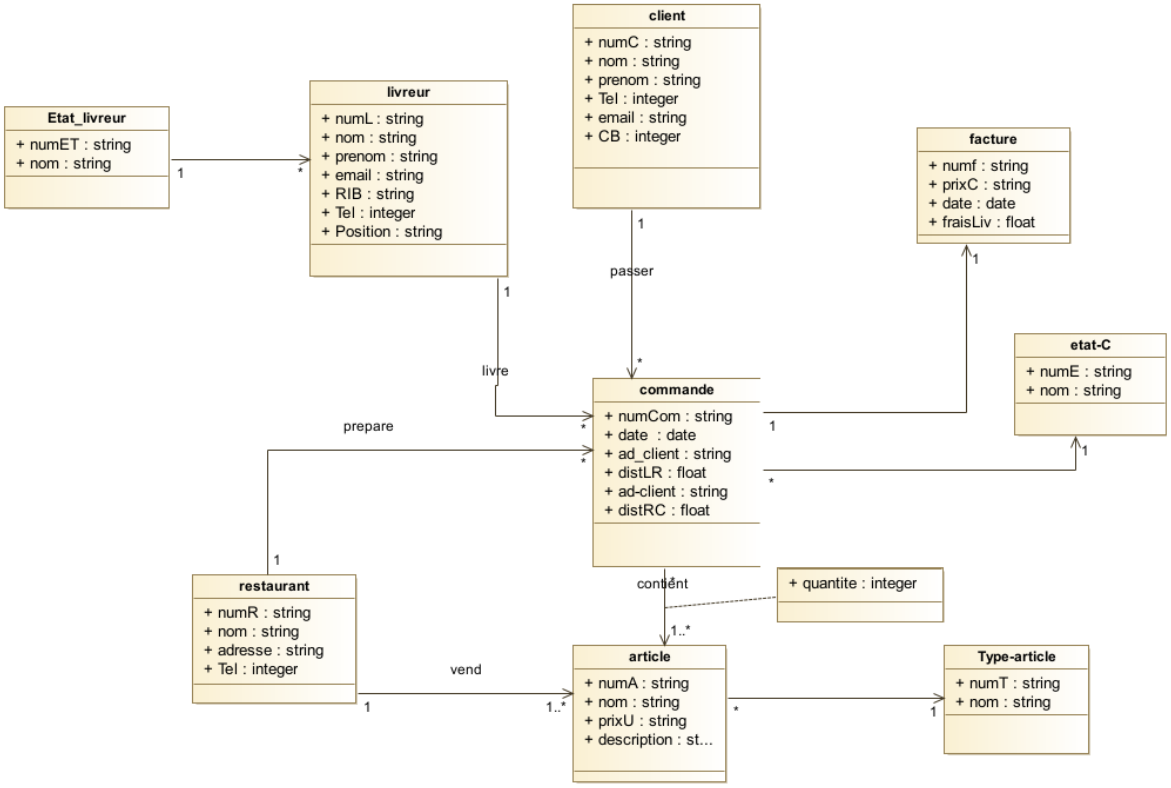
Ce qui sera géré par la base de données :

- Les informations des clients : nom, prénom, tel, émail, numéro de la CB
- Les informations des restaurants : numéro, nom, adresse, tel.
- Les informations des livreurs ; numéro, nom, prénom, émail, RIB,tel et stockage de sa position
- Les informations sur les commandes : numéro, date, adresse de livraison, distance entre le client et le restaurant et la distance entre le livreur et le restaurant.
- Les informations sur la facture : le numéro de la facture, le prix, la date et les frais de livraison
- Les informations sur l'état de la commande : le numéro de l'état et son nom
- Les informations sur les type des article : le numéro du type et le nom
- Les informations sur les articles : numéro des articles, nom, prix unitaire et une description
- Les informations sur l'état des livreurs : numéro de l'état et son nom.

Ce qui sera géré via du php et sql:

- Toute l'interface utilisateur application.
- Le calcul des distances entre les livreurs et les restaurants.
- Le calcul de la distance entre le restaurant et le client.
- Le calcul du montant de la commande.
- Le calcul des frais de livraison.
- La mise à jour des états de ma commande et du livreur.
- La date à laquelle s'effectue une commande.

Le modèle conceptuel :



Justification du modèle conceptuel :

Le choix	Justificatif
Le livreur renseigne sa position et cette dernière sera stockée dans la BDD.	À chaque fois que le livreur est disponible pour livrer depuis une nouvelle position.
On ajoute l'adresse du client à la table commande	Car pour un client donné, il peut avoir une adresse principale, celle renseignée lors de la création de son compte, mais il peut demander de se faire livrer à une autre adresse. Donc stocker l'adresse de livraison permet de ne pas perdre de l'information dans l'historique des commandes.
On crée une table pour l'état de livreur	ça nous permet de centraliser l'information, et aussi de l'unifier car on a plusieurs état (hors service, en attente …) donc on a jugé qu'il est plus pertinent d'avoir une table etat_ livreur
On crée une table type article	Cela nous permet de regrouper les articles exemple : boisson chaude, boisson froide …
On crée une table état de la commande	C'est plus pertinent, centraliser l'information.
On stocke la distance entre le livreur et le restaurant dans commande	Ca nous permet de sauvegarder l'historique des distances parcourues par un livreur pour chaque commande qu'il a effectué.
On stocke la distance entre le client et le restaurant dans commande	Pour pouvoir calculer les frais de livraison
Le cardinalités entre etat_livreur et livreur	A un instant t, le livreur possède un et un seul état. Plusieurs livreurs peuvent avoir le même état
Les cardinalités entre livreur et commande	Une commande est livrée par un et un seul livreur. Un livreur livre plusieurs commandes
Les cardinalités entre client et commande	Un client peut commander plusieurs fois mais une commande appartient à un et un seul client
Les cardinalités entre commande et état commande	À un instant t, la commande est dans un et un seul état. Plusieurs commandes peuvent se retrouver dans le même état.
Les cardinalités entre commande et restaurant	Un restaurant peut préparer plusieurs commandes, mais une commande est préparée par un et un seul restaurant.
Les cardinalités entre restaurant et article	Un article est vendu par au moins un restaurant et un restaurant vend au moins un seul article.

Les cardinalités entre facture et commande	Une commande a une seule facture, éditée lors de l'acceptation de la commande par le restaurant. Une facture appartient à une et une seule commande.
Les cardinalités entre commande et article	Une commande contient au moins un seul article. Mais un article peut appartenir à plusieurs commandes.
Les cardinalités entre article et type article	Un article appartient à un et un seul type. Un type contient plusieurs articles.

Le modèle relationnel :

Client(numC, nom, prenom, tel, email, cb)
Livreur(numL, nom, prenom, email, rib, tel, numET#)
Commande(numcom, date, distLR, distRC, numE#, numR#, numA#)
facture(numF, prixC, date, fraisLiv, numcom#)
etatC(numE, nom)
restaurant(numR, nom, adresse, tel)
articles(numA,numR# ,nom, prixU, description, quantite, type-article#)
type-article(numT, nom)
etat-livreur(numET, nom)

Les requêtes de création de table avec les contraintes de domaine et d'intégrité référentielle :

La table Client :

```
create table client (
        numC varchar(50),
        nom varchar(50) not null,
        prenom varchar(50) not null,
        tel integer,
        email varchar(50) not null,
        cb varchar(50)
        primary key (numC) );
```

La table état de la commande :

```
create table etatC (
         numE integer ,
         nom varchar(20) check
        (nom IN ('en
        préparation','prête','en
        livraison', 'livrée'))
       primary key(numE))
```

justification : Le check in pour vérifier que la commande ai un des états prédéfinis uniquement.

La table restaurant :

```
create table restaurant(
        numR varchar(20) primary key,
        nom varchar(20) not null,
        adresse varchar(50) not null,
        tel integer not null);
```

Justification:

l'adresse du restaurant, le numéro de téléphone ainsi que le nom doivent être connus pour les clients.

La table typearticle :

```
create table typearticle (
        numT integer primary key,
        nom varchar(20) check
        (nom IN
        ('Boisson','plat','dessert',
        'accompagnements'))
        primary key(numE));
```

Justification : check in pour verifier que l'article appartienne bien à une categorie bien définit.

La table etatlivreur :

```
create table etatlivreur (
        numEL integer primary key,
        nom varchar(20) (check
        nom In ('Hors service', 'en
        attente','en livraison')));
```

La table livreur :

```
create table livreur (
        numL varchar(50) primary key,
         nom varchar(50) not null,
         prenom varchar(50) not null,
         email varchar(50) not null,
         rib varchar(70) not null,
         tel integer,
        numEL integer,
        foreign key (numEL) references etatlivreur);
```

justification : numL en clé primaire pour identifier chaque livreur. numEL en clé étrangère pour avoir l'état du livreur. Les not null pour ne pas avoir de cases vides dans les tables.

La table article :

```
create table article (
        numA varchar(30) primary
        key, nom varchar(20) not null,
        prixU float  not null (check
        (prixU>0)) ,
        description varchar(150),
        numR varchar(20) not null,
         numT integer not null,
         foreign key (numR) references restaurant,
         foreign key (numT) references typearticle);
```

justification : check que le prix ne soit pas négatif pour ne pas avoir de problème lors des calculs. Les clés étrangères pour relié l'article au restaurant qui le propose et à son type.

La table commande :

```
create table commande (
        numCom varchar(50) primary key,
        date date not null,
         distLR integer not null,
        distRC integer not null,
        adclient varchar(50) not
        null,
        numE integer not null,
        numR varchar(20) not null,
        numC varchar(50) not null
        foreign key (numE) references etatC,
        foreign key (numR) references restaurant,
        foreign key (numC) references client);
```

justification : numE, numR, numC en clé étrangère pour relier la commande au client, au restaurant et au livreur.

La table facture :

```
create table facture(
        numF varchar(20) primary key,
        prixC float not
        null(check(prixC>0)),

      date date not null,
      fraisliv float not null
      (check(fraisliv>0)),
      numCom varchar(50) not
      null,
      foreign key (numCom)
      references commande);
```

justification : Clé étrangère numCom pour associer la facture à la bonne commande. Frais liv ne peut pas être négatif.

La table articlecommande :

```
numCom varchar(50),
numA varchar(30) not
null,
quantitie integer not
null(check (quantitie>0)),
primary key(numCom, numA),
foreign key ( numCom) references commande,
foreign key (numA) references article);
```

justification des contraintes :

not null : pour ne pas avoir des null dans les table

ghp_x2HuiBW8v1m1J3FwVCsySyg82aK7EQ
1rm4lc