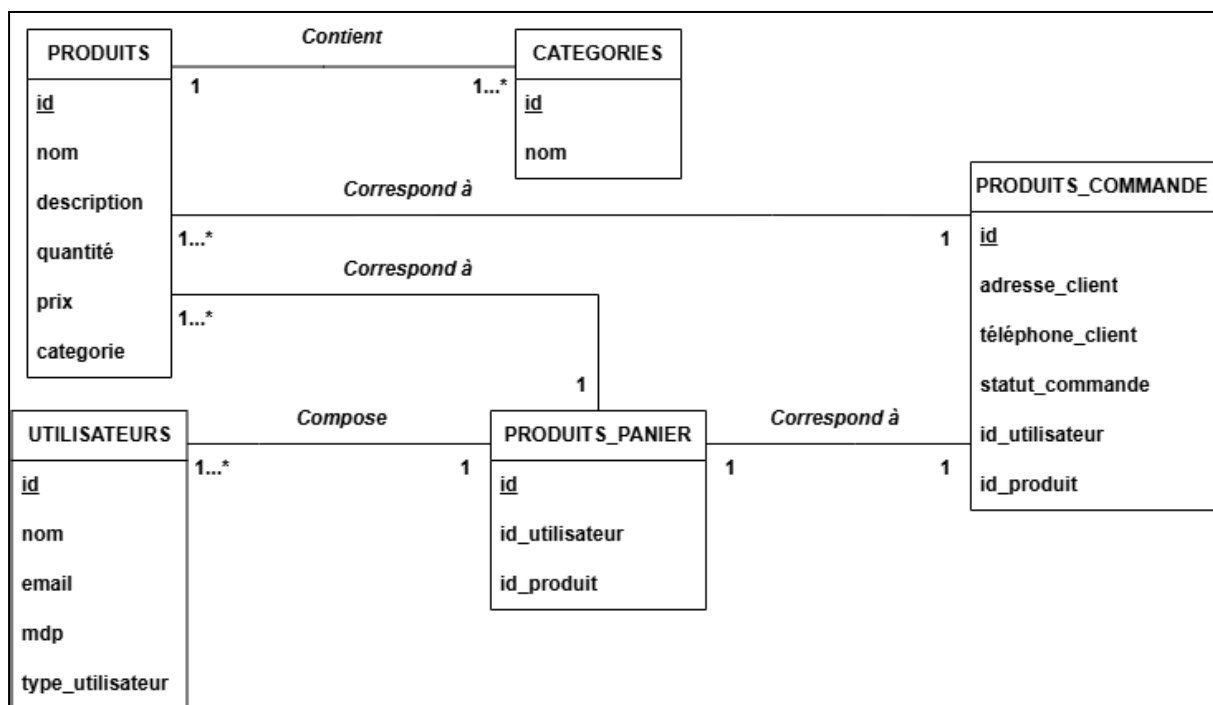


↳ **Modélisation de la base de données (MEA, UML et MR) liée au site web d'e-commerce:**

- Réaliser un projet de site web dynamique implique de créer et de maîtriser sa base de données. Pour pouvoir la mettre en place, il nous faut d'abord connaître en amont la structure approximative, si ce n'est exacte, de ses tables, des types de données qu'elles contiennent et des relations entre elles. À cet effet, nous devons modéliser ces tables.

- La modélisation, c.a.d: représentation graphique et théorique de l'agencement et du contenu des tables, doit se faire selon les standards de modélisation MEA/UML/MR, respecter la théorie des cardinalités (Un à Plusieurs/Plusieurs à Plusieurs), et intégrer des clés étrangères là où il le faut. Elle permet de prévisualiser la base de données, et les interactions entre les tables.

- Pour le cas de *Fullfort Crousty*, on suppose l'existence d'utilisateurs et de produits, et de catégories. On suppose également la possibilité pour les utilisateurs d'effectuer des commandes et/ou paiement et d'ajouter des produits aux paniers. Certains utilisateurs possèdent le statut administrateur et ont pour rôle d'importer de nouveaux produits disponibles. Il faut donc modéliser en conséquence.



*(De haut en bas): Diagramme UML (Unified Modeling Language) illustrant l'agencement, le contenu et les interactions théorisées dans la base de données 'fullfort_db'

- La structuration des données de *Fullfort Crousty* tourne majoritairement autour de la table 'produit' et 'user', la première étant la donnée primordiale d'un site de commerce agroalimentaire et la deuxième étant indispensable pour implémenter des

fonctionnalités dynamiques responsives. Par ailleurs, le package *Laravel Breeze*, très utile pour implémenter l'authentification, utilise cette même table 'user'.

```
Schema::create('products', function (Blueprint $table) {
    $table->id();
    $table->string('name')->nullable();
    $table->longText('description')->nullable();
    $table->integer('quantity')->nullable();
    $table->integer('price')->nullable();
    $table->string('image')->nullable();
    $table->string('category')->nullable();
    $table->timestamps();
});
```

*Migration de création de table illustrant les attributs de la table Produits

• Afin de tester la structuration des données modélisées, nous allons coder celle-ci dans un script SQL qui permettra de créer les tables dans le gestionnaire de bases de données de notre choix (en l'espèce, MySQL). L'interface graphique PhpMyAdmin offre également une fonctionnalité "concepteur" permettant la modélisation en temps réel de la base de données effective. Des requêtes SQL seront subséquentement faites pour confirmer la bonne synergie des colonnes (données) des tables.

```
1  -- Creation Tables
2  CREATE TABLE utilisateurs
3  (
4      id BIGINT(20) UNSIGNED PRIMARY KEY AUTO_INCREMENT,
5      nom VARCHAR(255),
6      email VARCHAR(255) UNIQUE,
7      mdp VARCHAR(255),
8      type_user VARCHAR(255) DEFAULT 'user'
9  );
10
11 CREATE TABLE produits
12 (
13     id BIGINT(20) UNSIGNED PRIMARY KEY AUTO_INCREMENT,
14     nom VARCHAR(255),
15     description LONGTEXT,
16     quantite INT(11),
17     prix INT(11),
18     categorie VARCHAR(255),
19     image VARCHAR(255)
20 );
21
22 CREATE TABLE categories
23 (
24     id BIGINT(20) UNSIGNED PRIMARY KEY AUTO_INCREMENT,
25     nom VARCHAR(255)
26 );
```

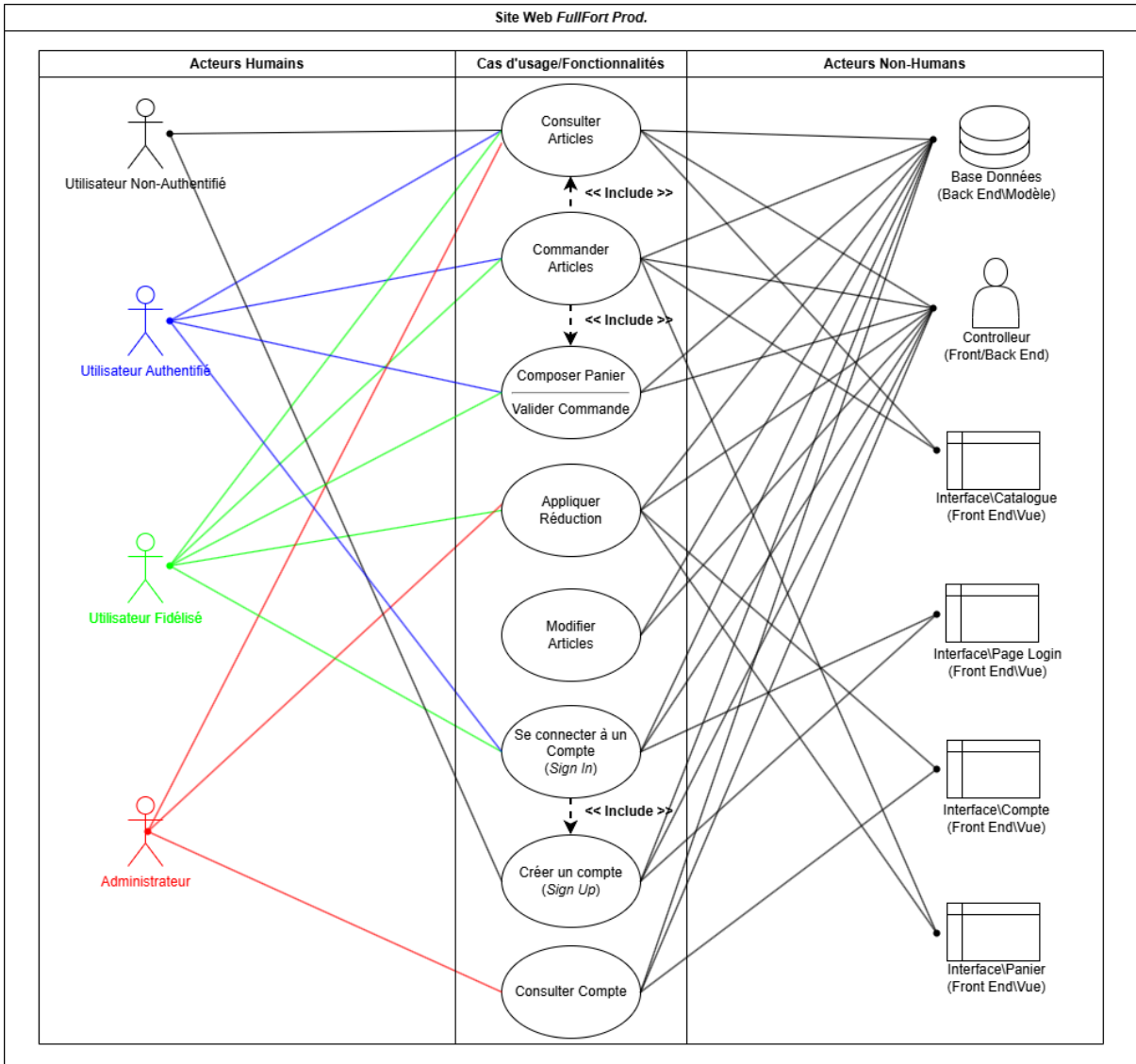
The visual schema diagram shows the following tables and their attributes:

- fullfort_db cart_products**: id (bigint(20) unsigned), user_id (bigint(20) unsigned), product_id (bigint(20) unsigned), created_at (timestamp), updated_at (timestamp)
- fullfort_db order_products**: id (bigint(20) unsigned), client_address (varchar(255)), client_phone (varchar(255)), status (varchar(255)), user_id (bigint(20) unsigned), product_id (bigint(20) unsigned), created_at (timestamp), updated_at (timestamp)
- fullfort_db products**: id (bigint(20) unsigned), name (varchar(255)), description (longtext), quantity (int(11)), price (int(11)), image (varchar(255)), category (varchar(255)), created_at (timestamp), updated_at (timestamp)
- fullfort_db users**: id (bigint(20) unsigned), name (varchar(255)), email (varchar(255)), email_verified_at (timestamp), password (varchar(255)), remember_token (varchar(100)), created_at (timestamp), updated_at (timestamp)
- fullfort_db categories**: id (bigint(20) unsigned), name (varchar(255)), created_at (timestamp), updated_at (timestamp)

Relationships shown in the diagram:

- users (id) to cart_products (user_id)
- users (id) to order_products (user_id)
- products (id) to cart_products (product_id)
- products (id) to order_products (product_id)
- products (category) to categories (name)

*(De gauche à droite): Extrait du code du Script SQL de modélisation et visuel du concepteur de PhpMyAdmin de la base de données 'fullfort_db'



**Diagramme Use Case illustrant les fonctionnalités du site par rapport aux acteurs (y compris les composants techniques dit acteurs "non humains"); (la fonctionnalité de points fidélité/réduction a été annulée dans le projet final)*