

2025

PROJET GESTION COMMANDE 2

DOCUMENTATION TECHNIQUE

Champiau Annaëlle



SOMMAIRE

<u>Contexte & Objectifs</u>	1
<u>Architecture du projet</u>	2
<u>Modèle de donnée & Scripts</u>	3
<u>Diagramme UML</u>	4
<u>Fonctionnalité Client</u>	5

CONTEXT & OBJECTIVES

01

INTRODUCTION



L'entreprise InfoTech Distribution spécialisé dans la vente de matériel informatique et high-tech, souhaite moderniser son processus de gestion de commandes. Actuellement, les clients sont gérer via des fichiers Excel, ce qui entraîne des erreurs fréquentes, une perte de temps et un manque de visibilité sur l'état des stocks et des livraisons.

CONTEXTE

L'enteprise fais appel à L'ESN SlamSJPIInfo (Entreprise de Service du Numérique) afin de concevoir une application web dynamique, ergonomique et sécurisée permettant :

- La gestion des clients et de leurs informations
- La gestion des produits et des stocks disponible
- La prise de commande avec choix des produits et quantités
- La modification et le suivi du statut des commandes

L'ESN vous charge en tant qu'étudiant en BTS SIO, de réaliser intégralement la conception de la Base de Donnée jusqu'au développement de l'application web en PHP avec une Base de donnée MySQL

OBJECTIFS

Développer un site Web permettant :

- L'enregistrement des clients et de leurs coordonnées
- La gestion des produits avec leurs stocks et prix
- La prise en charge des commandes avec la possibilités de sélectionné un client enregistrer, plusieurs produits, et une quantité pour chaque produit
- La gestion des statuts des commandes (en cours, expédiée, livrée, annulée).
- Une interface utilisateur intuitive et ergonomique.

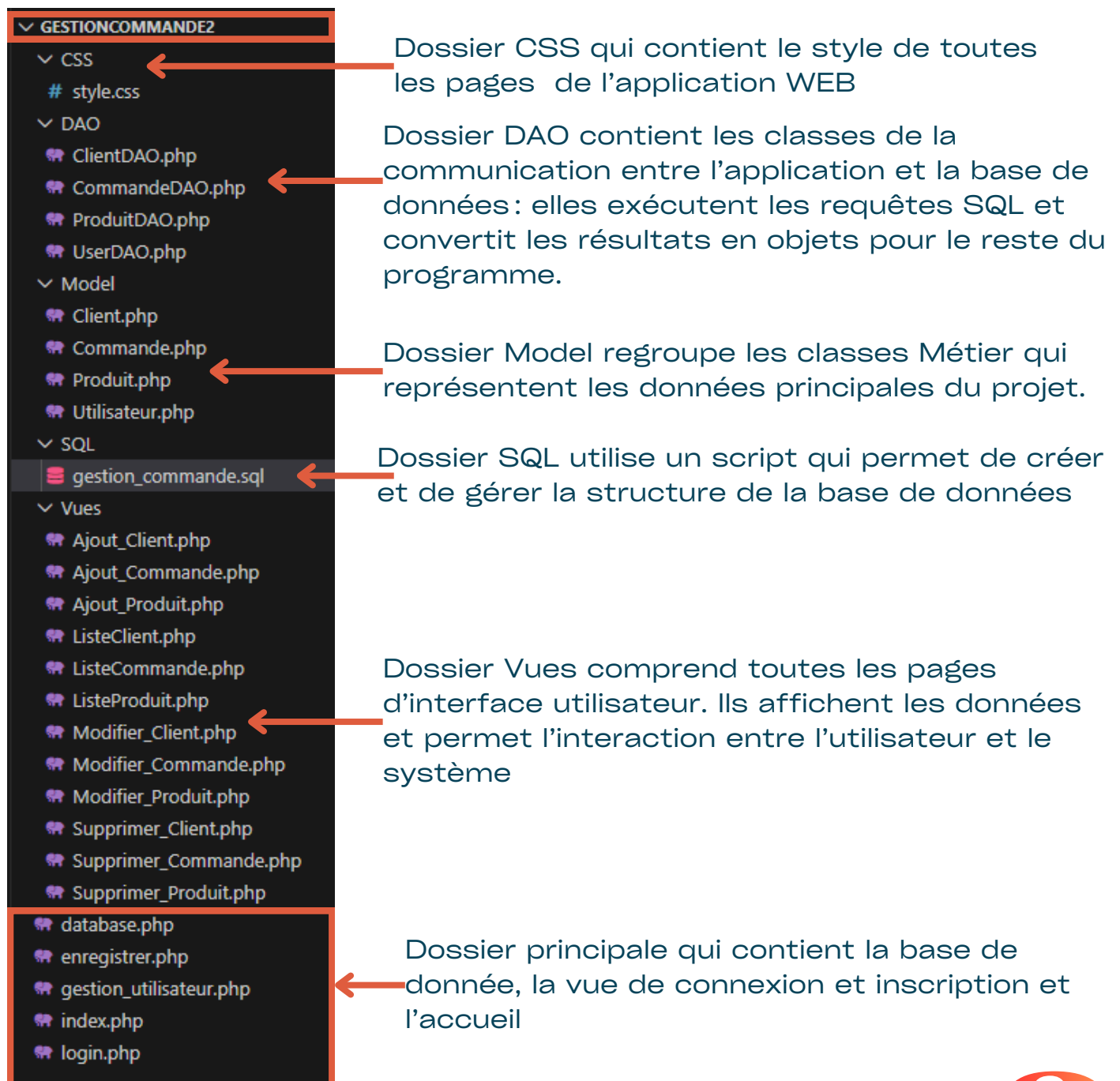


ARCHITECTURE DU PROJET

02

ARCHITECTURE DU PROJET

Ce projet de gestion de commandes, structuré selon une architecture MVC simplifiée, organise le code en dossiers dédiés aux modèles, accès aux données (DAO), vues, et inclut un script SQL d'initialisation, offrant ainsi une base claire et modulaire adaptée aux besoins d'apprentissage des étudiants.



MODÈLE DE DONNÉES + SCRIPTS

03

Modèle de donnée base de donnée PhpMyAdmin

gestion_commande	produits
id : int	
nom : varchar(100)	
description : text	
prix : decimal(10,2)	
stock : int	

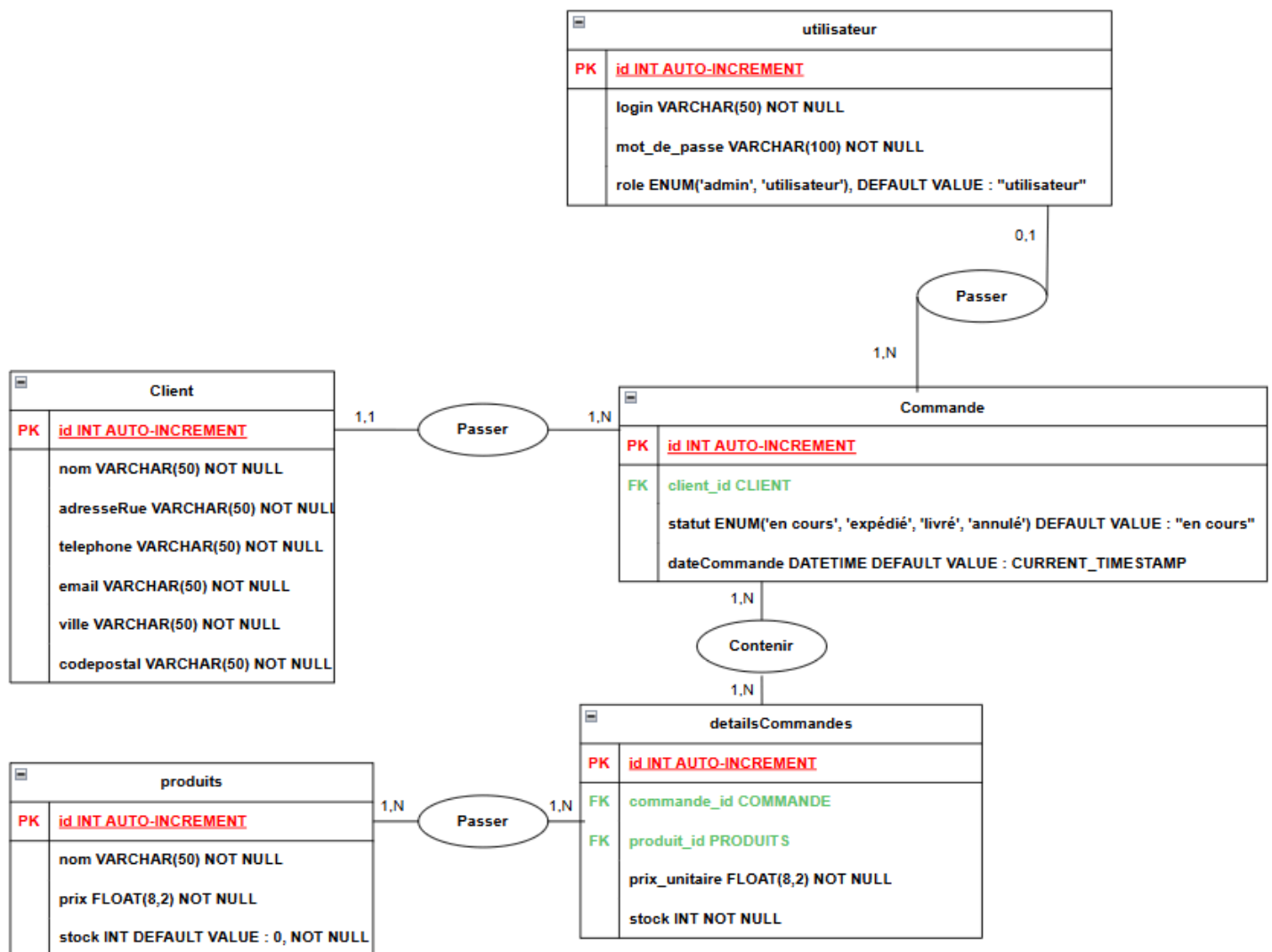
gestion_commande	utilisateurs
id : int	
login : varchar(100)	
mot_de_passe : varchar(255)	
role : enum('admin','utilisateur')	

gestion_commande	details_commandes
id : int	
commande_id : int	
produit_id : int	
quantite : int	
prix_unitaire : decimal(8,2)	

gestion_commande	clients
id : int	
nom : varchar(100)	
email : varchar(150)	
telephone : varchar(30)	
adresse_rue : varchar(255)	
code_postal : varchar(10)	
ville : varchar(50)	

gestion_commande	commandes
id : int	
client_id : int	
statut : enum('en cours','expédiée','livrée','annulée')	
date_commande : datetime	

Entité-Association de la base de donnée



Script SQL de la base de donnée

```

DROP TABLE IF EXISTS clients;
CREATE TABLE IF NOT EXISTS clients (
    id int NOT NULL AUTO_INCREMENT,
    nom varchar(100) NOT NULL,
    email varchar(150) DEFAULT NULL,
    telephone varchar(30) DEFAULT NULL,
    adresse_rue varchar(255) DEFAULT NULL,
    code_postal varchar(10) DEFAULT NULL,
    ville varchar(50) DEFAULT NULL,
    PRIMARY KEY (id)
) ENGINE=MyISAM AUTO_INCREMENT=15 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO clients (id, nom, email, telephone, adresse_rue, code_postal, ville) VALUES
(7, 'test2', 'test@gmail.com', '0618545369', 'rue du test1', '640000', 'Pau'),
(11, 'test2', 'test2@gmail.com', '0618545369', 'rue du test1', '64000', 'Pau');

DROP TABLE IF EXISTS commandes;
CREATE TABLE IF NOT EXISTS commandes (
    id int NOT NULL AUTO_INCREMENT,
    client_id int NOT NULL,
    statut enum('en cours','expédiée','livrée','annulée') DEFAULT 'en cours',
    date_commande datetime DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id),
    KEY client_id (client_id)
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

DROP TABLE IF EXISTS details_commandes;
CREATE TABLE IF NOT EXISTS details_commandes (
    id int NOT NULL AUTO_INCREMENT,
    commande_id int NOT NULL,
    produit_id int NOT NULL,
    quantite int NOT NULL,
    prix_unitaire decimal(8,2) NOT NULL,
    PRIMARY KEY (id),
    KEY commande_id (commande_id),
    KEY produit_id (produit_id)
) ENGINE=MyISAM DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

DROP TABLE IF EXISTS produits;
CREATE TABLE IF NOT EXISTS produits (
    id int NOT NULL AUTO_INCREMENT,
    nom varchar(100) NOT NULL,
    description text,
    prix decimal(10,2) NOT NULL,
    stock int NOT NULL DEFAULT '0',
    PRIMARY KEY (id)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO produits (id, nom, description, prix, stock) VALUES
(3, 'sticker chat', 'rghhgyguyhuhfcefsgfgrg', 8.99, 20);

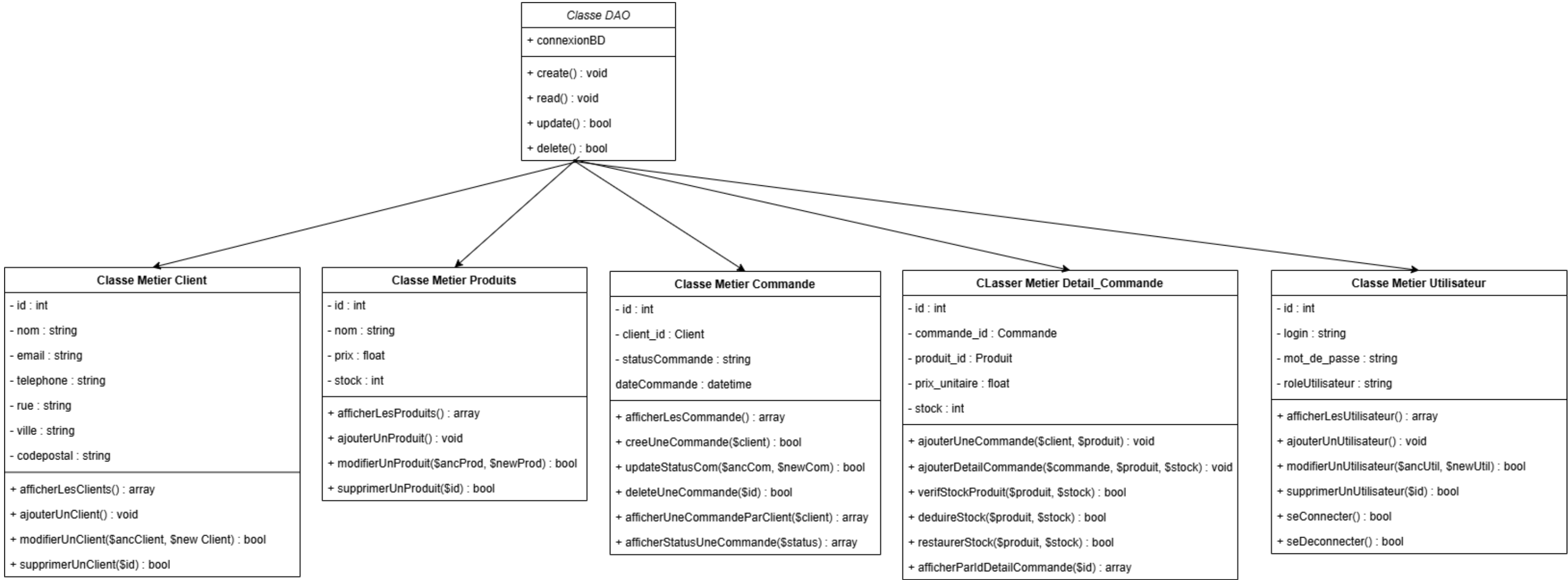
DROP TABLE IF EXISTS utilisateurs;
CREATE TABLE IF NOT EXISTS utilisateurs (
    id int NOT NULL AUTO_INCREMENT,
    login varchar(100) NOT NULL,
    mot_de_passe varchar(255) NOT NULL,
    role enum('admin','utilisateur') DEFAULT 'utilisateur',
    PRIMARY KEY (id),
    UNIQUE KEY login (login)
) ENGINE=MyISAM AUTO_INCREMENT=15 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO utilisateurs (id, login, mot_de_passe, role) VALUES
(1, 'admintest', '$2y$10$nJJA9F5Je/iUzhf8YdRBzexXg1JKkHwP59es/UtyjNM2.BMLgGP1K', 'utilisateur'),
(2, 'admin', '$2y$10$nBqXYiI3iyeZTOnXYqeQQu.6Vew8hOVXBQAmt40.k/Wzx8Iltkm8i', 'admin');
    
```

DIAGRAMME UML BD

04

DIAGRAMME UML DAO MÉTIER



FONCTIONNALITÉ CLIENT

05

BIENVENUE DANS LA GESTION DES COMMANDES

Connecté en tant que : **admintest (utilisateur)**

Gérer les Clients

Gérer les Produits

Gérer les Commandes

Déconnexion

```
<nav style="display: flex; gap: 18px; justify-content: center; margin-top: 30px;">
  <button class="cti" onclick="window.location.href='../GestionCommande2/Vues/ListeClient.php'">
    <span class="CTI">Gérer les Clients</span>
  </button>
```

```
<?php

session_start();

if (!isset($_SESSION['user_id'])) {
    header('Location: login.php');
    exit();
}

$login = $_SESSION['login'] ?? 'Utilisateur';
$role = $_SESSION['role'] ?? 'utilisateur';

?>
```

Gestion des Clients

Ajouter un Client +

← Retour

ID	Nom	Email	Téléphone	Adresse Rue	Ville	Code Postal	Actions
7	test2	test@gmail.com	0618545369	rue du test1	Pau	64000	Modifier Supprimer
11	test2	test2@gmail.com	0618545369	rue du test1	Pau	64000	Modifier Supprimer

```
public function afficherLesClients(): array {
    $sql = "SELECT * FROM clients";
    $stmt = $this->conn->prepare($sql);
    $stmt->execute();
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
    $lesClients = [];
    foreach ($rows as $row) {
        $lesClients[] = new Client(
            $row['id'],
            $row['nom'],
            $row['email'],
            $row['telephone'],
            $row['adresse_rue'],
            $row['ville'],
            $row['code_postal']
        );
    }
    return $lesClients;
}
```

```
<?php
require_once '../database.php';
require_once '../DAO/ClientDAO.php';

$db = Database::getInstance();

$conn = $db->getConnection();

$clientDao = new ClientDAO($conn);

$lesClients = $clientDao->afficherLesClients();

?>
```



FONCTIONNALITÉ CLIENT

Ajouter un client

```
public function ajouterClient(Client $unClient): bool {
    $sql = "INSERT INTO clients (nom, email, telephone, adresse_rue, ville, code_postal)
        VALUES (?, ?, ?, ?, ?, ?)";
    $params = [
        $unClient->getNom(),
        $unClient->getEmail(),
        $unClient->getTel(),
        $unClient->getRue(),
        $unClient->getVille(),
        $unClient->getCodePostal()
    ];
    $stmt = $this->conn->prepare($sql);
    return $stmt->execute($params);
}
```

```
<?php
require_once '../database.php';
require_once '../Model/Client.php';
require_once '../DAO/ClientDAO.php';
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Récupération des champs du formulaire
    $nom = $_POST['nom'] ?? '';
    $email = $_POST['email'] ?? '';
    $telephone = $_POST['telephone'] ?? '';
    $adresse_rue = $_POST['adresse_rue'] ?? '';
    $ville = $_POST['ville'] ?? '';
    $code_postal = $_POST['code_postal'] ?? '';

    // Création du client
    $client = new Client(null, $nom, $email, $telephone, $adresse_rue, $ville, $code_postal);

    // Connexion & DAO (CORRECTION : fournit bien un PDO)
    $db = Database::getInstance();
    $conn = $db->getConnection(); // ***
    $clientDAO = new ClientDAO($conn); // ***

    // Ajout en base
    $clientDAO->ajouterClient($client);

    // Redirection
    header("Location: ListeClient.php");
    exit();
}
?>
```

6

FONCTIONNALITÉ CLIENT

```
$clientDAO = new ClientDAO($conn); // ***

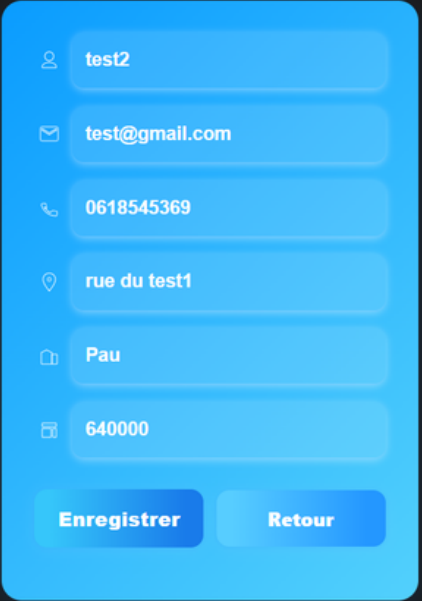
$id = $_GET['id'] ?? null;
$client = null;
$clientNouveau = null;

// On récupère le client à modifier via Database::fetch()
if ($id) {
    $row = $db->fetch("SELECT * FROM clients WHERE id = ?", [$id]);
    if ($row) {
        $client = new Client(
            $row['id'],
            $row['nom'],
            $row['email'],
            $row['telephone'],
            $row['adresse_rue'],
            $row['ville'],
            $row['code_postal']
        );
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST' && $client) {
    $nom = $_POST['nom'] ?? '';
    $email = $_POST['email'] ?? '';
    $telephone = $_POST['telephone'] ?? '';
    $adresse_rue = $_POST['adresse_rue'] ?? '';
    $ville = $_POST['ville'] ?? '';
    $code_postal = $_POST['code_postal'] ?? '';

    $clientNouveau = new Client(
        $client->getId(),
        $nom,
        $email,
        $telephone,
        $adresse_rue,
        $ville,
        $code_postal
    );

    $clientDAO->modifierClient($clientNouveau, $client);
    header("Location: ListeClient.php");
    exit();
}
?>
```



test2

test@gmail.com

0618545369

rue du test1

Pau

640000

Enregistrer Retour

localhost indique

Supprimer ce client ?

OK

Annuler

```
<?php
require_once '../database.php';
require_once '../Model/Client.php';
require_once '../DAO/ClientDAO.php';

if (isset($_GET['id']) && is_numeric($_GET['id'])) {
    $db = Database::getInstance();
    $conn = $db->getConnection(); // CORRECTION ici !
    $clientDAO = new ClientDAO($conn); // On passe bien le PDO

    $client = new Client($_GET['id'], '', '', '', '', '', '');

    $clientDAO->supprimerClient($client);

    header('Location: ListeClient.php');
    exit();
} else {
    header('Location: ListeClient.php');
    exit();
}
?>
```