

REPUBLICUE DU CAMEROUN

UNIVERSITE DE NGAOUNDERE

FACULTES DES SCIENCES

DEPARTEMENT

D'INFORMATIQUE



REPUBLICUE OF CAMEROON

UNIVERSITY OF NGAOUNDERE

FACULTY OF SCIENCES

DEPARTEMENT OF COMPUTER
SCIENCE

TRAVAUX PRATIQUES ET CONTROLE CONTINU

**THEME : Application d'achat et de
livraison à domicile de denrées alimentaires
à bini-dang(marché à domicile)**

GROUPE 16

UE : INGENIERIE DES APPLICATIONS WEB

NOMS ET PRENOMS	MATRICULES
ADOUM MAHAMAT AHMAT ADOUM	21A399FS
BEDJI MOITANGAR	21A184FS
BYAMPAMBE YABO WILFRED	21A106FS
HIKWOLBO TALERANT	21A301FS
KOUYAHBE PAYIMI DEUBE	21A402FS
MBAIHOGAOU DJEDOUBOUM JOSAPHAT	21A246FS
SAINKOR AMEDE(CHEF)	21A086FS
YANTELKA YABO	21A247FS

Chargé du cours Dr WOHWE DAMBO Sambien

1) Un document descriptif contenant :

a. La présentation du contexte du projet : De quoi est-il question ?

Le projet consiste à développer une application web nommée " marché dang-bini" pour faciliter l'achat et la livraison à domicile de denrées alimentaires à Bini-Dang, au Cameroun. Cette initiative vise à offrir aux utilisateurs un moyen simple et pratique de commander une variété de produits alimentaires en ligne auprès de différents fournisseurs locaux, tout en se faisant livrer ces produits directement à leur domicile.

b. Présenter les fonctionnalités développées dans l'application web

1. Catalogue de Produits :

- Les utilisateurs peuvent naviguer dans un catalogue en ligne de produits alimentaires proposés par différents fournisseurs locaux.

- Chaque produit est accompagné de descriptions détaillées et d'images.

2. Passer des Commandes :

- Les utilisateurs peuvent ajouter des produits à leur panier et passer des commandes en ligne.

- Ils peuvent indiquer leur adresse de livraison et leurs préférences de paiement.

3. Gestion des Fournisseurs :

- Les fournisseurs ont la possibilité de gérer leurs catalogues de produits, mettre à jour les prix et les disponibilités.

- Ils reçoivent des notifications des nouvelles commandes passées.

4. Gestion des Livraisons :

- Les livreurs ont accès à un tableau de bord pour gérer les commandes en attente.

- Ils peuvent planifier les itinéraires de livraison et mettre à jour le statut des livraisons en temps réel.

5. Paiements en Ligne :

- Les utilisateurs peuvent régler leurs achats en ligne de manière sécurisée.

- Les méthodes de paiement incluent les cartes de crédit, les portefeuilles électroniques, etc.

6. Suivi des Commandes :

- Les utilisateurs peuvent suivre en temps réel le statut de leurs commandes depuis la préparation jusqu'à la livraison via une interface utilisateur conviviale.

c. Présenter la technologie et le matériel utilisés pour développer l'application :

i. Les langages de programmation utilisés

- Frontend :

- HTML5, CSS3 pour la structure et le style des pages web.
- JavaScript pour l'interactivité et les fonctionnalités dynamiques.

- Backend :

- php pour le développement du serveur.
- javascript pour les services spécifiques.

ii. Les logiciels utilisés

- IDE et Éditeurs de Code : Visual Studio Code.

- Systèmes de Gestion de Base de Données :

- MySQL pour les données structurées.

- Outils de Gestion de Versions : Git avec GitHub pour le suivi des versions et la collaboration.

- Serveurs :Apache pour servir l'application web.

- Plateformes de Déploiement : GitHub

Autres Outils et Technologies

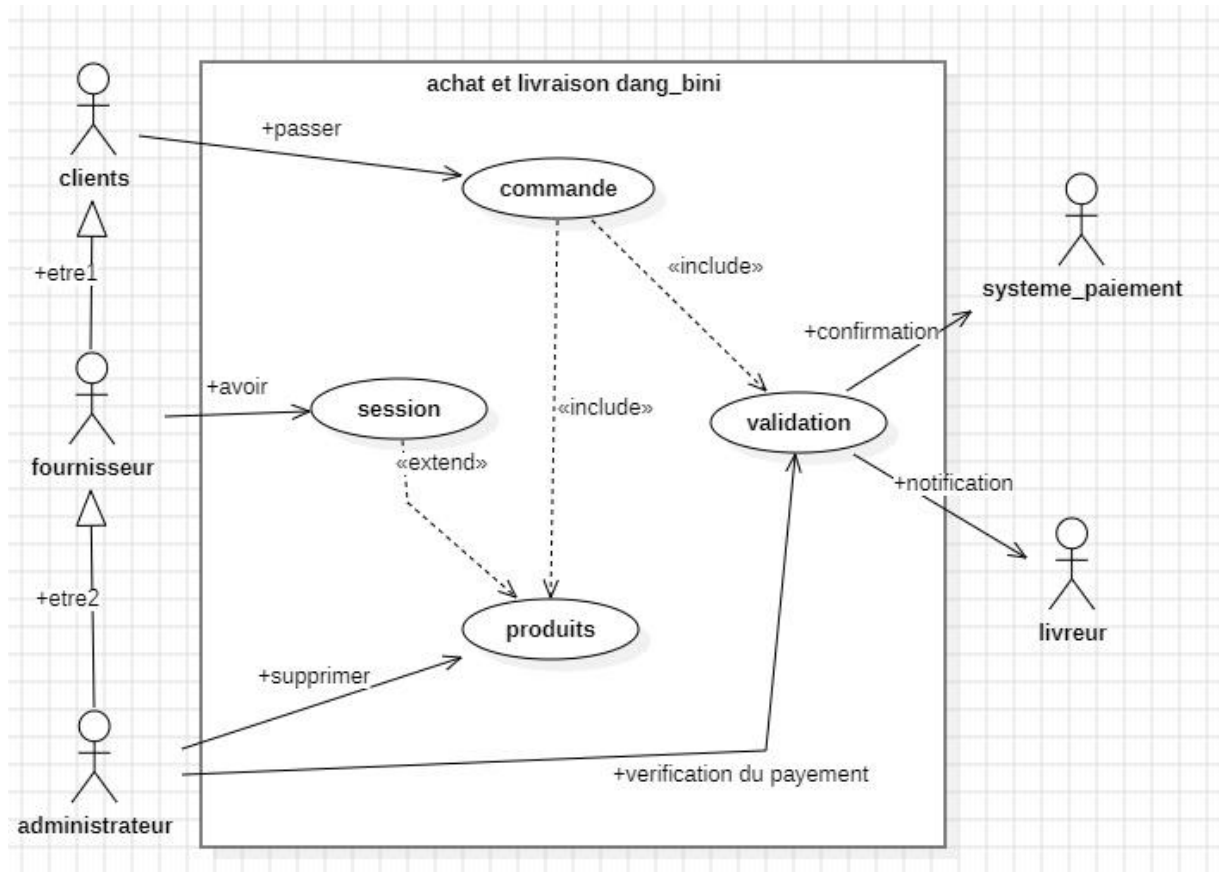
- starUML : pour faire la conception.
- Outils de Gestion de Projet : email pour la planification .

Ce document présente un aperçu complet du projet "Marché à Domicile", en détaillant les fonctionnalités développées, les technologies utilisées et les contributions des membres de l'équipe.

CONCEPTION

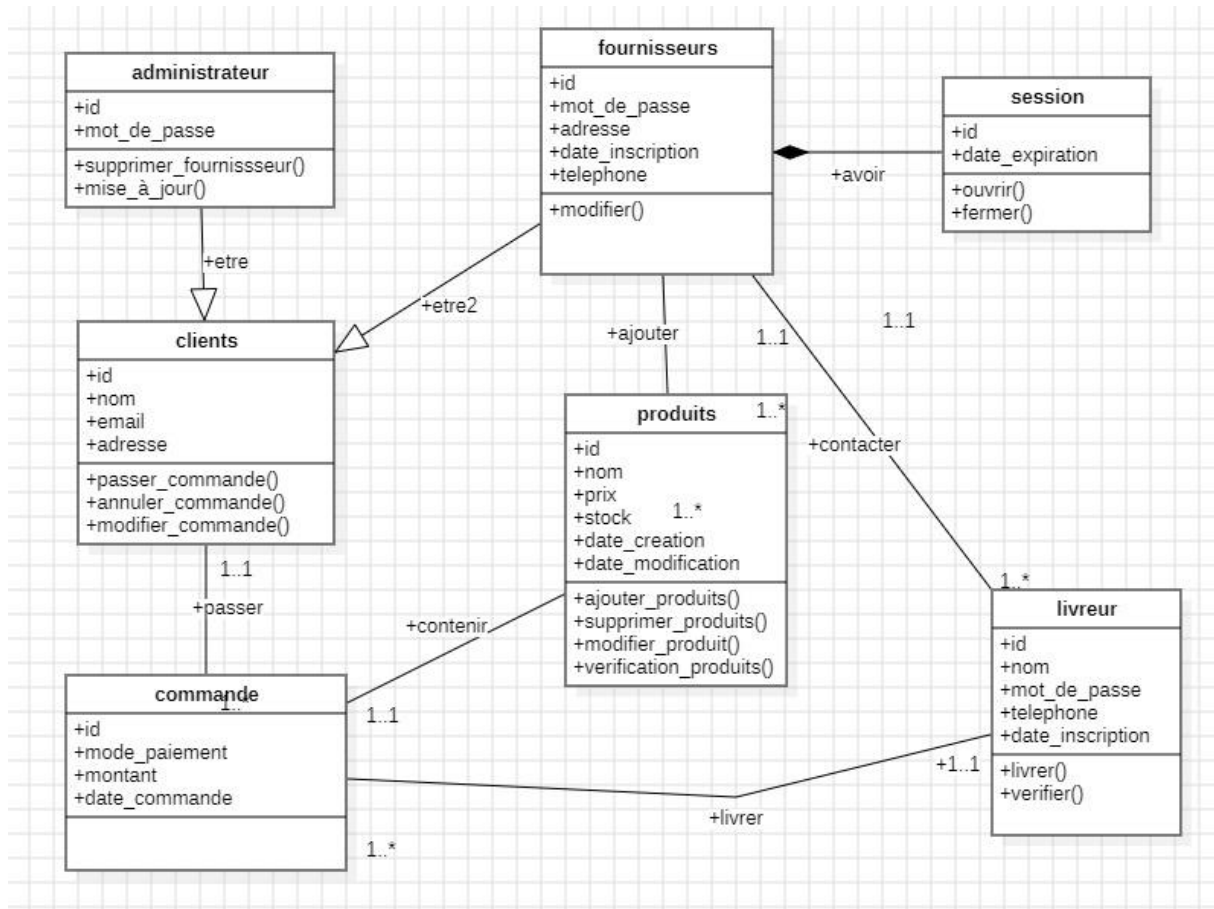
Nous avons utilisé UML pour faire la conception :

1. Diagramme de cas d'utilisation



Description : Cette image représente un diagramme de cas d'utilisation qui décrit les interactions entre différents acteurs (clients, fournisseur, administrateur, système de paiement, livreur) dans le cadre d'un processus d'achat et de livraison. Le diagramme montre les différentes étapes du processus, telles que la commande, la session, la validation, la vérification du paiement, la livraison, etc. Les flèches représentent les messages échangés entre les différents acteurs au cours de ces étapes. Ce diagramme est utilisé pour modéliser et visualiser les interactions dynamiques au sein d'un système ou d'un processus, afin de mieux comprendre et communiquer les spécifications fonctionnelles d'une application.

2. Diagramme de classe



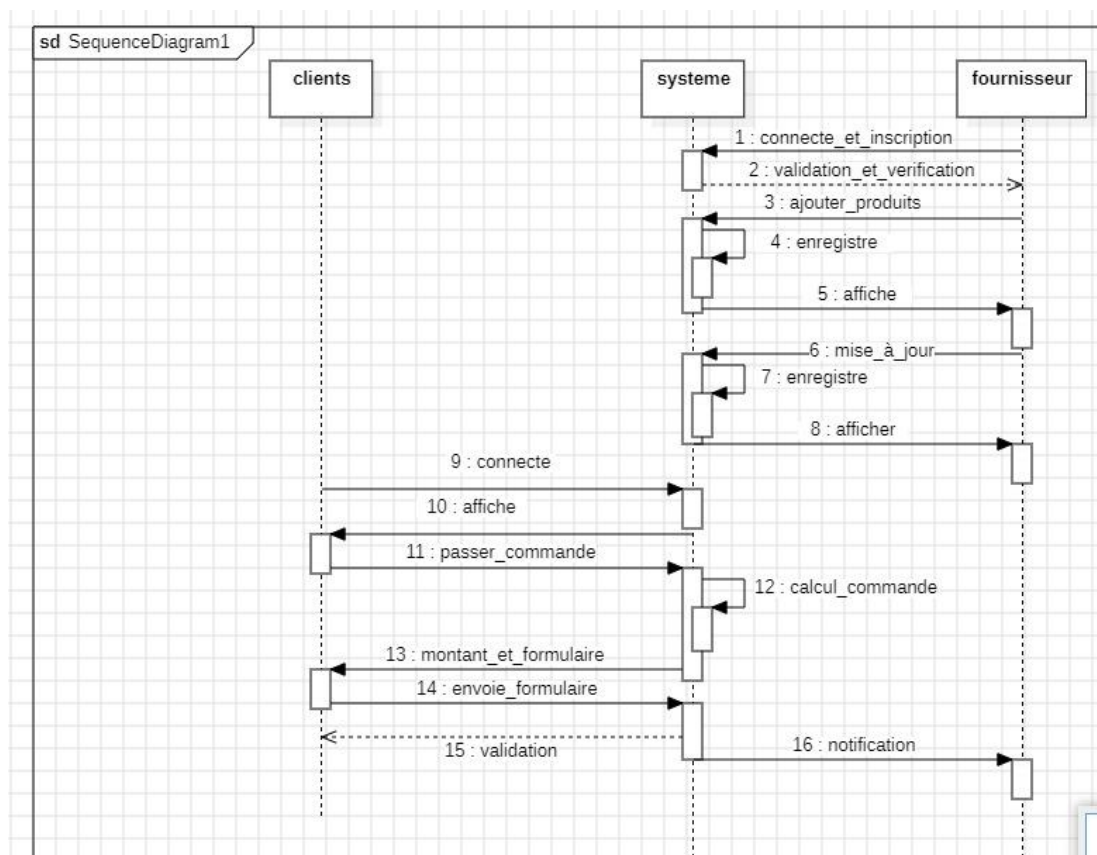
Ce diagramme de classes UML (Unified Modeling Language) représentant les relations entre différentes entités dans un système informatique. Ce diagramme comprend les classes suivantes :

- **Administrateur** : Cette classe a des attributs comme id, mot_de_passe, et des méthodes comme supprimer_fournisseur() et mise_a_jour().
- **Clients** : Cette classe a des attributs comme id, nom, email, adresse, et des méthodes comme passer_commande(), annuler_commande() et modifier_commande().
- **Commande** : Cette classe a des attributs comme id, mode_paiement, montant, date_commande.
- **Fournisseurs** : Cette classe a des attributs comme id, mot_de_passe, adresse, date_inscription, telephone, et une méthode modifier().
- **Produits** : Cette classe a des attributs comme id, nom, prix, stock, date_creation, date_modification, et des méthodes comme ajouter_produit(), supprimer_produit() et modifier_produit().

- Session : Cette classe a des attributs comme id, date_expiration, ouvrir() et fermer().
- Livreur : Cette classe a des attributs comme id, nom, mot_de_passe, telephone, date_inscription et des méthodes comme livrer().

Les flèches entre ces classes représentent les relations entre elles, comme une relation de composition entre Clients et Commande, ou une relation d'association entre Fournisseurs et Produits.

3. Diagramme de sequence



Description : Cette image est un diagramme de séquence qui illustre les interactions entre différents composants ou entités au sein d'un système. Le diagramme montre ce qui suit :

Les principales entités impliquées sont "clients", "système" et "fournisseur". Le diagramme représente une séquence d'interactions ou de messages échangés entre ces entités, représentés par les flèches numérotées. Les interactions incluent des actions telles que "connecte_et_inscription", "validation_et_verification", "ajouter_produits", "enregistre", "affiche", "mise_a_jour", "enregistre", "afficher", "connecte", "affiche", "passer_commande", "calcul_commande", "montant_et_formulaire", "envoie_formulaire" et "notification". Le diagramme fournit une représentation visuelle du flux d'actions et d'interactions entre les différents composants du système.

Document Présentant les Différents Membres du Groupe et le Descriptif de l'Apport de Chacun

Équipe de Développement

1. SAINKOR AMEDE (chef)

- Rôle : Supervision globale du projet, coordination entre les membres de l'équipe, planification des tâches, et communication avec les parties prenantes.
- Apport : Gestion de projet, coordination des livraisons, et intégration des composants.

2. HIKWOULBO TAE LRANT ET ADOUM MAHAMAT AHMAT ADOUM (Développeur Frontend)

- Rôle : Développement de l'interface utilisateur, intégration des designs, et amélioration de l'expérience utilisateur.
- Apport : Création des pages web réactives et interactives, intégration avec le backend via API.

3. HIKWOULBO TAE LRANT (Développeur Backend)

- Rôle : Développement des API, gestion de la base de données, et mise en place du logique serveur.
- Apport : Conception et gestion de la base de données, développement des services RESTful, et intégration des systèmes de paiement.

5. YANTELKA YABO, KOUYAHBE PAYIMI DEUBE ET BIAMPABE YABO WILFRED

- Rôle : Conception de l'interface utilisateur, création de prototypes, et tests d'utilisabilité.
- Apport : Création de wireframes, maquettes haute-fidélité, et réalisation de tests utilisateur pour améliorer l'interface.

6. MBAIHOGAOU DJEDOUBOUM JOSAPHAT ET BEDJI MOITANGAR

Rôle: Analyste Fonctionnel

Apport: Analyse des besoins des utilisateurs, rédaction des spécifications fonctionnelles, test et validation des fonctionnalités pour s'assurer qu'elles répondent aux attentes et besoins identifiés

6. TOUS LES MEMBRES DU GROUPE

- Rôle : Test des fonctionnalités, détection des bugs, et validation des corrections.
- Apport : Rédaction de plans de test, exécution de tests manuels et automatisés, et reporting des anomalies.