

RÉPUBLIQUE DU CAMEROUN Paix-Travail-Patrie ***** MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ***** UNIVERSITÉ DE MAROUA ***** FACULTÉ DES SCIENCES ***** Email : <a href="mailto:decanat@fs.univ-maroua.cm">decanat@fs.univ-maroua.cm</a>	 P.O Box/BP 814 Maroua <a href="http://www.fs.univ-maroua.cm">http://www.fs.univ-maroua.cm</a> <a href="https://www.facebook.com/fsmaroua">https://www.facebook.com/fsmaroua</a>	REPUBLIC OF CAMEROON Peace-Work-Fatherland ***** MINISTRY OF HIGHER EDUCATION ***** THE UNIVERSITY OF MAROUA ***** FACULTY OF SCIENCE ***** Email : <a href="mailto:decanat@fs.univ-maroua.cm">decanat@fs.univ-maroua.cm</a>
--	--	---

UE: INF345 (GENIE LOGICIEL)

GROUPE:

**THEME: CONCEPTION D'UNE APPLICATION DE  
GESTION DES REGLES DANS RESTAURANT EN  
LIGNE**

PRESENTER PAR:

N°	Noms et Prenoms	Matricules
1		
2	TATSITSA FOMENA ROY KARIM	22FS0267
3	MAYANGA GUIMBE	22A0850FS
4		

Examineur: M. Bayang Souloukna

Année Académique: 2025-2026

# **APPLICATION DE GESTION DES ORDRES D'UN RESTAURANT EN LIGNE**

# Table de matière

## I- Vue d'ensemble

## II- Cahier de charge fonctionnel

1-Contexte

2-Objectifs

3-Description des utilisateurs et de leurs besoins

4-Liste des fonctionnalités attendues

5-Contraintes

6-Cas d'utilisation (Use case)

## III-Cahier des charges techniques

1-Choix des technologies, langages et framework

2-Description de l'architecture logicielle

3-Modèle de données

4-Interfaces prév

# **I-VUE D'ENSEMBLE**

Dans l'ère du numérique , la plupart des secteurs de la restauration cherche constamment à optimiser ses processus internes et à améliorer l'expérience client. La gestion manuelle des commandes (prise de note sur papier, transmission physique en cuisine) est souvent source d'erreurs, de lenteurs et d'une mauvaise coordination, impactant directement la satisfaction de la clientèle.

Ce projet propose de répondre à cette problématique par la création de "RestoQuick", une application de gestion des ordres de restauration en ligne. Cette solution vise à automatiser les services clients ,de réservation et de cuisine afin de garantir une fluidité opérationnelle maximale.

# **II-CAHIER DE CHARGE**

## **FONCTIONNEL**

### **1. Contexte et problématique**

Dans le secteur de la restauration, la gestion fluide des flux de commandes est un défi majeur. Les méthodes traditionnelles (prise de commande orale ou papier) ralentissent le service et augmentent le risque d'erreurs lors de la transmission en cuisine. L'application RestoQuick répond à ce besoin de modernisation en offrant une interface numérique intuitive pour automatiser ces échanges.

### **2. Objectifs généraux et spécifiques**

- **Objectif général** : Concevoir une application complète permettant la gestion de bout en bout du cycle de vie d'une commande de restaurant.
- **Objectifs spécifiques** :
  - Réduire le temps d'attente des clients lors de la prise de commande.
  - Assurer une communication instantanée entre la salle, la cuisine et le client.
  - Centraliser les données de vente pour faciliter le suivi du chiffre d'affaires.

### **3. Description des utilisateurs et de leurs besoins**

- **Le Client** : Souhaite consulter le menu, passer une commande rapidement et suivre son avancement en temps réel.

- Le Restaurateur (Personnel de cuisine/salle) : Doit recevoir les commandes instantanément, changer le statut des plats (disponible/épuisé) et valider la préparation.
- L'Administrateur : Besoin de gérer les comptes utilisateurs, de modifier la carte (prix, descriptions) et d'analyser les performances globales.

## **4. Liste des fonctionnalités attendues**

- Gestion du menu : Affichage par catégories (Entrées, Plats, Desserts, Boissons) avec photos et prix.
- Prise de commande : Sélection d'articles, gestion du panier et validation.
- Notifications en temps réel : Alerte pour le restaurateur lors d'une nouvelle commande et alerte pour le client quand le plat est prêt.
- Suivi des commandes : Tableau de bord affichant les commandes "En attente", "En cours" et "Terminées".
- Gestion des stocks : Possibilité de désactiver un plat momentanément indisponible.

## **5. Contraintes**

- Temporelles : Le projet doit être réalisé en dix semaines à compter du 7 octobre.
- Environnementales : L'application doit être accessible sur différents terminaux (tablettes en cuisine, smartphones pour les clients).
- Techniques : Nécessité d'une connexion internet stable pour la synchronisation en temps réel.

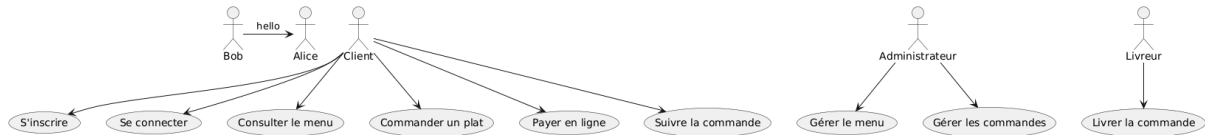
## **6. Cas d'utilisation (Use Cases)**

Le cas d'utilisation principal est "Passer une commande". Il implique :

1. Le Client sélectionne ses articles.
2. Le système vérifie la disponibilité.
3. Le Client valide et enregistre la commande.

4. Le Restaurateur reçoit une notification immédiate.

## Diagramme de cas d'utilisation



Ce diagramme de cas d'utilisation présente les interactions entre les acteurs et le système du restaurant en ligne. Il permet d'identifier clairement les responsabilités de chaque acteur ainsi que les fonctionnalités essentielles du système.

# **III-CAHIER DES CHARGES**

## **TECHNIQUES**

### **1. Choix des technologies, langages et Frameworks**

Pour garantir une application moderne, performante et maintenable, les outils suivants ont été sélectionnés:

- **Frontend : React.js** pour créer une interface utilisateur dynamique et réactive adaptée aux mobiles et ordinateurs.
- **Backend : Node.js** avec le framework **Express** pour la gestion de la logique serveur et des API.
- **Base de données : MongoDB** ou **PostgreSQL** pour assurer le stockage persistant des menus, des comptes et des commandes.
- **Outils de développement : Git** pour le versionnage du code et **Visual Studio Code** comme environnement de développement.

### **2. Description de l'architecture logicielle**

L'application repose sur une **architecture client-serveur** découplée:

- **Le Client (Frontend)** : Communique avec le serveur via des appels API REST.
- **Le Serveur (Backend)** : Traite les requêtes, interroge la base de données et applique les règles métier (calcul des prix, validation des stocks).
- **Communication en temps réel** : Utilisation de **WebSockets** (Socket.io) pour notifier instantanément la cuisine lors de l'arrivée d'une nouvelle commande.

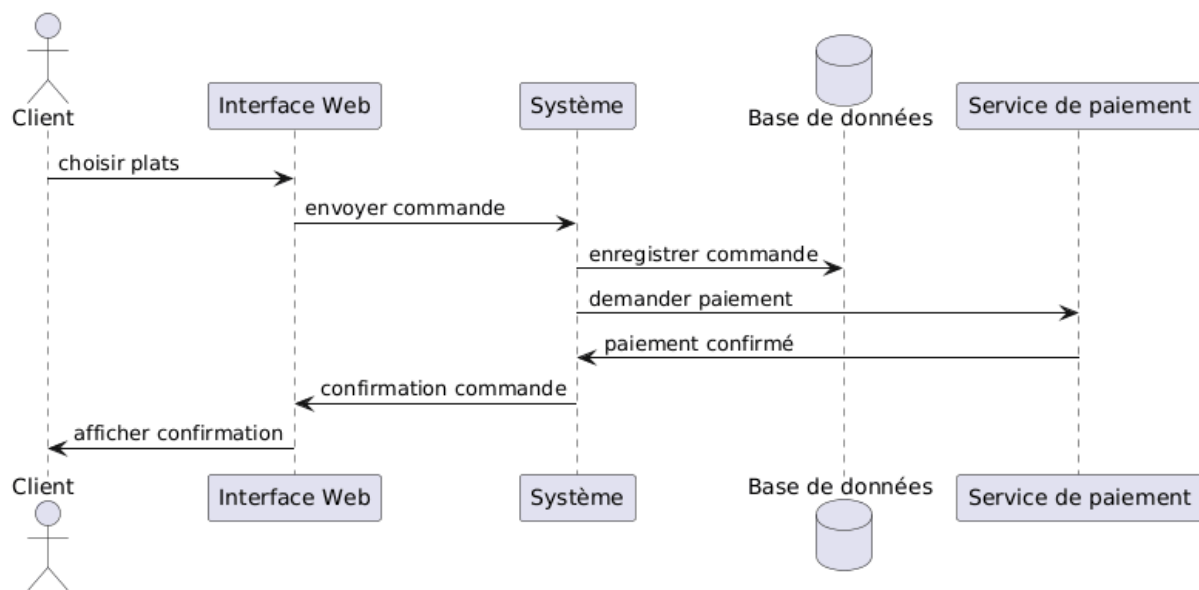


### 3. Modèle de données

Le schéma de la base de données (ou diagramme de classes) est structuré autour des entités suivantes:

- **Utilisateurs** : Identifiant, nom, email, mot de passe chiffré, rôle (Client ou Admin).
- **Produits (Menu)** : Nom du plat, description, prix, catégorie, image et état de disponibilité.
- **Commandes** : Date, liste des produits sélectionnés, montant total, adresse de livraison et statut (En attente, Prête, Livrée).

### Diagramme de séquence



Ce diagramme montre le déroulement temporel des interactions lors de la commande d'un plat.

## 4. Interfaces prévues

Le système disposera de trois interfaces clés:

- **Interface Client** : Un catalogue visuel avec panier d'achat et suivi de commande.
- **Tableau de bord Restaurateur** : Une vue simplifiée pour gérer les commandes entrantes en un clic.
- **Interface Admin** : Un panneau de gestion pour ajouter/supprimer des produits du menu.

## 5. Gestion des erreurs et sécurité

La fiabilité du système est assurée par:

- **Authentication** : Utilisation de tokens **JWT (JSON Web Tokens)** pour sécuriser les accès.
- **Sécurité des données** : Chiffrement des mots de passe en base de données.
- **Gestion des erreurs** : Validation systématique des formulaires et messages d'erreur explicites en cas de problème de connexion ou de stock épuisé.