

Sujet : Meal Delivery Application

Auteur : ABED Mohamed Aziz

Date : March 31, 2025

# Meal Delivery Application

## 1. Introduction au Projet

### Description et Périmètre :

L'objectif de la Meal Delivery est de développer une application Java avec interface graphique permettant aux clients de commander des repas, aux livreurs de gérer les livraisons, et à un administrateur de superviser les utilisateurs et le catalogue de repas. Le périmètre inclut une base de données MySQL pour la persistance des données (utilisateurs, repas, commandes), une interface utilisateur avec navigation (login, pages spécifiques par rôle), et des fonctionnalités de gestion des commandes (création, annulation, mise à jour). L'application vise à simplifier le processus de livraison de repas tout en offrant une expérience utilisateur fluide.

### Motivations :

Ce projet a été choisi pour son aspect pratique et sa pertinence dans le domaine des services numériques actuels. Il permet d'explorer la programmation orientée objet, la gestion de bases de données, et les interfaces graphiques en Java. Bien qu'il ne soit pas nécessairement notre premier choix, il figure parmi nos priorités car il combine des compétences techniques variées et répond à un besoin réel, offrant une opportunité d'apprentissage concrète et applicative.

## 2. Spécification du Projet :

### Notions de Base et Contraintes :

- L'application repose sur une architecture client-serveur avec une base de données MySQL (tables: **clients**, **delivery\_workers**, **meals**, **invoices**, **orders**).
- Contraintes: Unicité des noms d'utilisateur et repas, persistance des données, annulation limitée aux commandes "Pending", sécurité minimale des identifiants.

### Acteurs et Fonctionnalités Attendues :

#### Client :

- **Connexion/Inscription** : S'authentifie ou s'inscrit avec username, password, et adresse.
- **Consulter Repas** : Voit une liste de repas avec prix et peut choisir des quantités (e.g., 3 pizzas).
- **Commander** : Sélectionne plusieurs repas avec quantités, clique "Order", voit une facture (ID, repas, total).
- **Consulter/Annuler Commandes** : Accède à une page listant ses factures, peut annuler celles en "Pending".
- **Navigation** : Retourne à la page de login.

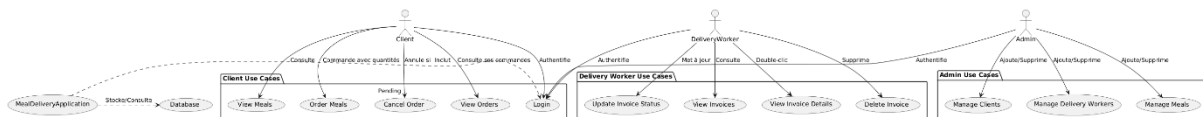
### Livreur (Delivery Worker):

- **Connexion/Inscription** : S'authentifie ou s'inscrit avec username, password, nom, téléphone.
- **Consulter Factures** : Voit toutes les factures (ID, client, total, statut), double-clique pour détails (repas, quantités).
- **Mettre à Jour** : Change le statut (Delivered, Pending, Fake Client).
- **Supprimer** : Supprime une facture après confirmation.
- **Navigation** : Retourne à la page de login.

### Administrateur :

- **Connexion** : Utilise Aziz/Aziz.
- **Gérer Utilisateurs** : Ajoute/supprime clients (username, password, adresse) et livreurs (username, password, nom, téléphone).
- **Gérer Repas** : Ajoute/supprime repas (nom, prix).
- **Navigation** : Retourne à la page de login

## 3. Diagrammes de Cas d'Utilisation :



**Description** : Les acteurs interagissent avec des cas d'utilisation spécifiques. Le système central (**MealDeliveryApplication**) gère l'authentification et les données.

## 4. Les Priorités des Cas d'Utilisation :

### Haute Priorité :

- Login (tous les acteurs)
- Order Meals (Client)

### Moyenne Priorité :

- View Meals (Client)
- View Orders (Client)
- View Invoices (Delivery Worker)
- Update Invoice Status (Delivery Worker)

### Basse Priorité :

- Cancel Order (Client)
- Delete Invoice (Delivery Worker)
- Manage Clients, Delivery Workers, Meals (Admin)

## 5. Planification du 1er Sprint :

**Durée** : 2 semaines

**Objectif** : Implémenter les fonctionnalités de base pour la connexion et la commande.

### Tâches :

1. Configurer la base de données MySQL (tables et données initiales).
2. Développer **LoginPage** avec authentification/inscription.
3. Développer **ClientPage** avec liste des repas et commande.
4. Intégrer **DatabaseManager** pour persistance.

**Rôles :** Développeur (code), Testeur (validation).

**Livrables :** Application avec login et commande fonctionnels.

## 6. Sélection des Cas d'Utilisation de Haute Priorité :

### Cas Sélectionnés :

1. Login (tous les acteurs)
2. Order Meals (Client)

## 7. Spécification et Tests de Validation :

### Cas 1 : Login

#### Pré-conditions :

- Base de données contient utilisateurs (clients, livreurs, admin prédéfini).
- Champs username, password, rôle remplis.

#### Post-conditions :

- Utilisateur authentifié → Ouvre page correspondante (AdminPage, ClientPage, DeliveryWorkerPage).
- Échec → Message d'erreur ou proposition d'inscription.

#### Tableau de Décision :

Condition	C1: Admin Valide	C2: Client Valide	C3: Livreur Valide	C4: Invalide
Username correct	Oui	Oui	Oui	Non
Password correct	Oui	Oui	Oui	Non
Rôle sélectionné	Admin	Client	Delivery Worker	Any
Résultat attendu	AdminPage	ClientPage	DeliveryWorkerPage	Erreur
Test ID	T1	T2	T3	T4

### Cas 2 : Order Meals

#### Pré-conditions :

- Client connecté.
- Liste des repas chargée.
- Au moins un repas sélectionné avec quantité > 0.

### Post-conditions :

- Facture créée dans **invoices**.
- Commandes ajoutées dans **orders** avec quantités.
- Facture affichée à l'utilisateur.

### Tableau de Décision :

Condition	C1: Sélection Valide	C2: Aucune Sélection	C3: Quantité Négative
Repas sélectionnés	Oui	Non	Oui
Quantité > 0	Oui	N/A	Non
Résultat attendu	Facture affichée	Erreur "Select meal"	Erreur implicite
Test ID	T5	T6	T7

## 8. Écriture des Spécifications et Génération des Tests :

**Formalisation :** Voir pré-/post-conditions ci-dessus.

### Tests de Validation :

- T1: Login Admin (Aziz/Aziz) → AdminPage.
- T2: Login Client (existant) → ClientPage.
- T3: Login Delivery Worker (existant) → DeliveryWorkerPage.
- T4: Login Invalide → Erreur.
- T5: Commander 2 pizzas → Facture affichée.
- T6: Commander sans sélection → Erreur.
- T7: Quantité négative → Échec silencieux (quantité ignorée).