

Documentation de gestion de projet et documentation technique

Projet : Vite & Gourmand

Date : 19/02/2026

Ce document présente la démarche de gestion de projet ainsi que la documentation technique de l'application Vite & Gourmand, depuis les choix initiaux jusqu'au déploiement local.

1. Gestion de projet

1.1 Méthodologie utilisée

La gestion de projet repose sur une approche **agile inspirée de Scrum**, adaptée à un projet académique. Cette approche permet de livrer rapidement une version fonctionnelle tout en restant flexible face aux évolutions.

Principes appliqués :

- Découpage du projet en fonctionnalités indépendantes
- Priorisation des parcours utilisateurs (client → employé → administrateur)
- Développement itératif avec validation régulière
- Tests fonctionnels à chaque incrément

1.2 Organisation du travail

Le travail a été organisé autour d'un dépôt Git structuré selon les bonnes pratiques :

- **main** : branche stable correspondant aux livrables validés
- **develop** : branche d'intégration continue
- **feature/*** : une branche par fonctionnalité (panier, commandes, admin, etc.)

Chaque fonctionnalité est développée, testée puis fusionnée dans la branche **develop**. Une fois l'ensemble validé, un merge final est effectué vers **main**.

2. Documentation technique

2.1 Réflexions technologiques initiales

L'objectif principal était de concevoir une application démontrant des parcours complets (client, employé, administrateur) sans dépendance serveur complexe.

Les choix technologiques ont donc privilégié la simplicité et la lisibilité :

- HTML5 / CSS3 pour la structure et le design
- JavaScript natif pour la logique métier et les interactions
- Simulation des données côté front-end
- Scripts SQL fournis pour illustrer une implémentation back-end réaliste

2.2 Configuration de l'environnement de travail

Environnement recommandé :

- Système : Windows, macOS ou Linux
- Outils : Git, navigateur moderne (Chrome, Firefox)
- Serveur local : Python (`http.server`) ou Node.js (`serve`)

Aucun framework n'est requis, ce qui facilite la prise en main et l'évaluation.

3. Modélisation

3.1 Modèle conceptuel de données (MCD)

Le modèle conceptuel de données repose sur les entités principales suivantes :

- Utilisateur (Client, Employé, Administrateur)
- Adresse
- Menu
- Commande
- Historique de statut
- Avis
- Horaires d'ouverture

Les relations sont principalement de type 1■N (un client peut avoir plusieurs commandes, une commande possède plusieurs statuts successifs, etc.).

3.2 Diagramme de cas d'utilisation

Trois acteurs principaux ont été identifiés :

Client : consulter les menus, passer commande, suivre une commande, déposer un avis

Employé : gérer les commandes, modifier les statuts, gérer les horaires, modérer les avis

Administrateur : gérer les employés, consulter les statistiques, superviser l'activité

3.3 Diagramme de séquence (exemple : passer une commande)

1. Le client sélectionne un menu
2. Ajout au panier
3. Validation de la commande
4. Création de la commande avec statut « en attente »
5. Notification implicite à l'employé pour traitement

4. Documentation du déploiement

4.1 Démarche de déploiement

Le déploiement local est volontairement simple afin de garantir l'accessibilité du projet lors de l'évaluation.

Étapes :

1. Cloner le dépôt Git
2. Se placer à la racine du projet
3. Lancer un serveur HTTP local (Python ou Node.js)
4. Accéder à l'application via <http://localhost>

Les scripts SQL peuvent être exécutés indépendamment pour illustrer la couche base de données.

Cette démarche garantit une installation rapide, reproductible et conforme aux contraintes d'un projet pédagogique.