**Cahier de charges**

|  |
| --- |
| **Titre du Projet** |
| **Application de gestion de point de vente (POS)** |

|  |
| --- |
| **Problématique posée** |
| Les petits commerces, tels que les cafés, librairies ou boutiques de quartier, rencontrent souvent des difficultés dans la **gestion quotidienne de leurs ventes et produits**. La plupart de ces établissements utilisent encore des méthodes **manuelles** (papier, calculatrice, cahier) pour enregistrer les commandes, suivre les stocks ou calculer les totaux, ce qui entraîne :   * des **erreurs fréquentes** dans les montants ou les commandes, * une **perte de temps** dans la saisie et la vérification des ventes, * une **absence d’historique fiable** pour le suivi des transactions. |

|  |
| --- |
| **Objectif Général** |
| L’objectif général de ce projet est de **concevoir et développer une application web de gestion de point de vente (POS)**, simple et fonctionnelle, permettant à un petit commerce de **gérer efficacement ses produits, ses catégories, ses commandes et l’historique de ses ventes**, tout en appliquant une **méthodologie de travail collaborative agile** (via Jira et GitHub) pour simuler un environnement de développement professionnel. |

|  |
| --- |
| **Objectifs Spécifiques** |
| Pour atteindre l’objectif général, le projet vise à :   1. **Analyser les besoins fonctionnels** du petit commerce afin d’identifier les principales fonctionnalités attendues (gestion des produits, catégories, commandes et historique). 2. **Concevoir le modèle de données** du système à travers un diagramme de classes UML et la définition des entités principales (Produit, Catégorie, Commande). 3. **Mettre en place une architecture logicielle** basée sur le framework **Spring Boot** et une base de données **MySQL** pour assurer la persistance des informations. 4. **Développer les fonctionnalités essentielles (MVP)** :    * Ajout, consultation et suppression des produits et des catégories,    * Création de commandes avec calcul automatique du total,    * Affichage de l’historique des ventes. 5. **Utiliser une méthodologie agile (Scrum)** via **Jira ou GitHub Projects** pour organiser les tâches, suivre l’avancement et assurer une bonne communication entre les membres de l’équipe. 6. **Appliquer un workflow Git collaboratif** (branches, Pull Requests, revues de code) afin de garantir la qualité et la traçabilité du développement. 7. **Tester manuellement les fonctionnalités** avec **Postman** avant leur intégration dans la branche principale afin d’assurer la stabilité de l’application. |

|  |
| --- |
| **Méthode d’analyse des données** |
| * UML * MERISE |

|  |
| --- |
| **Système et Langages de Programmation** |
| * Spring Boot * Java * MySQL * Docker * UML * JIRA * GIT /Github |

|  |
| --- |
| **Plan Général Provisoire du Projet** |
| **Diagramme de cas d’utilisation**   ****Explications des relations****  1. **Categorie → Produit**    * Une **catégorie** contient plusieurs **produits**.    * Relation : OneToMany (catégorie vers produits). 2. **Produit → Commande**    * Une **commande** peut contenir plusieurs **produits**,    * Et un **produit** peut être dans plusieurs **commandes**.    * Relation : ManyToMany. |