

PGI Velo

Progiciel de Gestion Intégré pour une PME

– Cahier de charges –

Plan

1. Présentation du projet

1.1 Objectifs

1.2 Cible

1.3 Responsabilités

2. Description de la solution

2.1 Besoins fonctionnels

2.2 Besoins non fonctionnels

3. Contraintes générales

3.1 Qualité de la solution

3.2 Temps de développement

3.3 Coûts de développement

4. Livrables attendus

1. Présentation du projet

Le projet PGI Vélo consiste à développer une solution web centralisée permettant à un magasin de vélos (PME) de gérer l'ensemble de ses activités opérationnelles : ventes, achats, gestion des fournisseurs, gestion des clients, facturation, gestion du stock et gestion des employés.

Le but est d'offrir un outil unique, intuitif et fiable, permettant de suivre en temps réel l'activité du magasin grâce à un tableau de bord dynamique.

1.1 Objectifs

- Concevoir un PGI complet dédié à un commerce de vélos.
- Centraliser les informations : clients, produits, fournisseurs, ventes, achats, employés.
- Suivre les niveaux de stock et les mouvements de stock automatiquement.
- Générer des factures automatiquement.
- Offrir un dashboard interactif pour visualiser les indicateurs clés :
→ ventes mensuelles, achats mensuels, bénéfices, produits les plus vendus, stock faible...
- Réduire les erreurs et augmenter la traçabilité.
- Déployer une application accessible via navigateur.

1.2 Cible

- Magasins de vélos, commerces de détail, PME.
- Gérant ou administrateur (utilisateur unique dans cette version).
- Utilisation sur PC / tablette via navigateur.

1.3 Responsabilités

Responsabilités principales :

- **Équipe de développement (Kenza chahnaz & Ngueba Anne-Maéra)**

- Conception de la base de données et des modules métiers (ventes, achats, stocks, RH, etc).
 - Développement Front-end (interfaces Laravel Blade) et Back-end (contrôleurs, modèles, API Laravel).
 - Tests fonctionnels et techniques en local via XAMPP.
 - Mise en production (Hostinger).
 - Tests et corrections.
 - Suivi de la qualité du code, documentation et gestion de projet.
-
- **Encadrant pédagogique - CHERGUI Anis :**
 - Évaluation du projet lors de la soutenance finale.
 - **Utilisateur final (PME fictive) :**
 - Définir les besoins du PGI (ventes, achats, stocks, RH).
 - Vérifier que le système fonctionne bien et répond aux attentes.
 - Vérification du bon fonctionnement de la solution en environnement local (XAMPP) puis en ligne, après hébergement.

2. Description de la solution

2.1 Besoins fonctionnels

Module	Besoins Fonctionnels
Clients	Ajouter un client ; Modifier un client ; Supprimer un client ; Lister les clients
Produits	Ajouter un produit ; Modifier un produit ; Supprimer un produit ; Lister les produits ; Voir stock ; Alerte stock faible ; Voir mouvements de stock
Fournisseurs	Ajouter fournisseur ; Modifier fournisseur ; Supprimer fournisseur ; Lister fournisseurs
Achats	Créer un achat ; Ajouter lignes d'achat ; Valider achat ; Mettre à jour stock ; Lister achats
Ventes / Commandes	Créer commande ; Ajouter lignes ; Calculer total ; Valider commande ; Mettre à jour stock ; Lister commandes
Factures	Générer facture ; Lister factures
Employés	Ajouter employé ; Modifier employé ; Supprimer employé ; Lister employés
Stock	Suivre mouvements ; Voir stock ; Détection stock faible
Dashboard	Voir ventes mensuelles ; Voir achats mensuels ; Voir bénéfice ; Produits les plus vendus ; Ventes par type ; Stock faible
User (Admin)	Connexion ; Déconnexion ; Modifier mot de passe

2.2 Besoins non fonctionnels

Catégorie	Besoins Non Fonctionnels
Accessibilité	Application responsive (PC, tablette, mobile)
Sécurité	Authentification admin ; Hash mot de passe ; Validation données
Performance	Temps de réponse rapide ; Mise en cache serveur
Disponibilité	Service accessible 24/7
Fiabilité	Sauvegarde base ; Intégrité données ; Logs erreurs
Maintenance	Architecture MVC ; Code évolutif
Compatibilité	Navigateurs modernes (Chrome, Safari, Firefox)
Hébergement	Compatible Hostinger (PHP 8.2 + MySQL)
Conformité	Respect RGPD ; Données minimales collectées
Évolutivité	Modules ajoutables facilement

3. Contraintes générales

3.1 Qualité de la solution

- Le PGI doit être fonctionnel et fiable, sans erreurs critiques.
- L'interface doit être ergonomique, claire et responsive pour une utilisation facile.
- Le code doit être structuré, commenté et documenté pour faciliter la maintenance.
- La base de données MySQL doit être relationnelle et cohérente, avec intégrité des données assurée.
- Les tests doivent être réalisés pour valider toutes les fonctionnalités avant déploiement.

3.2 Temps de développement

Le temps alloué pour réaliser le projet est de 3 semaines.

3.3 Coûts de développement

Ressources / Outils	Description	Coût estimé
Ordinateur personnel	Machine utilisée pour le développement et les tests	0 €
XAMPP	Serveur local pour tester le PGI (Apache, MySQL, PHP)	0 €
Laravel	Framework PHP pour le développement du PGI	0 €
MySQL	Base de données relationnelle	0 €
Hébergement web	Déploiement en ligne du PGI avec Hostinger Business	19 €

Un budget de 19 euros à été alloué pour le développement du projet

4. Livrables attendus

Phase	Élément	Livrable	Explication courte
1. Phase initiale	Cahier des charges	Cahier des charges fonctionnel	Recueille besoins, objectifs et contraintes
	Étude de faisabilité	Rapport de faisabilité	Vérifie si le projet est réalisable
	Plan de projet	Plan de gestion de projet	Organise tâches, calendrier et ressources
2. Analyse	Spécifications fonctionnelles générales (SFG)	Document de SFG	Vue d'ensemble des fonctionnalités
	Spécifications fonctionnelles détaillées (SFD)	Document de SFD	Détails des fonctionnalités et scénarios
	Spécifications techniques	Dossier technique	Choix techniques et architecture
	Modèles UML	Diagrammes UML	Schémas pour comprendre le système
3. Conception	Conception architecturale	Schéma d'architecture	Organisation globale du système
	Maquettes / Prototypes	Maquettes d'écrans	Valide l'interface et l'ergonomie
4. Développement & Test	Documentation technique	Documentation développeur	Guide pour utiliser et maintenir le code
	Plan de tests	Plan de tests	Stratégie et types de tests
	Rapports de tests	Rapport de validation	Résultats et corrections des tests