\*\*Site web pour une application de gestion de stock\*\*  
  
\*\*Expression des Besoins\*\*  
  
\* Besoins fonctionnels:  
 + Créer un site web pour gérer les stocks d'un entrepôt  
 + Permettre l'enregistrement et la modification des produits en stock  
 + Permettre la saisie des quantités et des prix de vente  
 + Générer des rapports sur les soldes actuels et historiques des stocks  
 + Offrir une interface utilisateur intuitive pour les utilisateurs  
\* Besoins non fonctionnels:  
 + Sécurité: authentification des utilisateurs, autorisation des accès, chiffrement des données sensibles  
 + Performance: site web rapide et réactif, avec un temps de chargement inférieur à 3 secondes  
 + Usabilité: interface utilisateur facile à utiliser, avec des instructions claires et des messages d'erreur appropriés  
\* Objectifs principaux du projet:  
 + Créer un outil pour gérer les stocks efficacement  
 + Améliorer la visibilité sur les quantités de produits en stock  
 + Réduire les erreurs d'inventaire et améliorer la prise de décision  
\* Contraintes et dépendances:  
 + Intégration avec un système de gestion de vente existant  
 + Prise en compte des spécifications du fournisseur de la plateforme  
  
\*\*Cahier des Charges\*\*  
  
\* Présentation du projet: Site web pour une application de gestion de stock  
\* Contexte et problématique: La gestion des stocks est un processus complexe qui nécessite une grande attention à la sécurité, à la performance et à l'usabilité. Le site web doit être capable de gérer les quantités et les prix de vente, ainsi que les rapports sur les soldes actuels et historiques.  
\* Périmètre du projet: Le site web sera développé pour une application de gestion de stock, avec une interface utilisateur intuitive et des fonctionnalités de saisie des quantités et des prix de vente. Il ne sera pas intégré à un système de gestion de vente existant.  
\* Spécifications fonctionnelles détaillées:  
 + Saisie des produits en stock  
 + Gestion des quantités et des prix de vente  
 + Générations des rapports sur les soldes actuels et historiques  
 + Sécurité: authentification des utilisateurs, autorisation des accès  
\* Spécifications techniques:  
 + Langage de programmation: Python ou JavaScript  
 + Framework web: Flask ou Django  
 + Base de données: MySQL ou PostgreSQL  
\* Critères de réussite et indicateurs de performance:  
 + Temps de chargement inférieur à 3 secondes  
 + Sécurité: absence de vulnérabilités critiques pendant au moins 6 mois  
 + Usabilité: score supérieur à 80% pour l'interface utilisateur  
\* Risques et plans de mitigation:  
 + Risque de sécurité: mise en place d'un plan de sécurité solide avec des tests réguliers  
 + Risque de performance: mise en place d'une architecture scalable avec des tests de charge  
  
\*\*Backlog\*\*  
  
\* User Stories (Priorités):  
 1. Enregistrement et modification des produits en stock (High)  
 2. Saisie des quantités et des prix de vente (Medium-High)  
 3. Générations des rapports sur les soldes actuels et historiques (Medium)  
 4. Sécurité: authentification des utilisateurs, autorisation des accès (High)  
\* Tâches associées à chaque User Story:  
 1. Enregistrement et modification des produits en stock:  
 - Créer une base de données pour les produits  
 - Développer une interface utilisateur pour l'enregistrement et la modification des produits  
 2. Saisie des quantités et des prix de vente:  
 - Créer un système de saisie pour les quantités et les prix de vente  
 - Intégrer le système de saisie avec la base de données  
 3. Générations des rapports sur les soldes actuels et historiques:  
 - Développer un système de génération de rapports  
 - Intégrer le système de génération de rapports avec la base de données  
\* Définition des critères de complétion (Definition of Done) pour chaque tâche:  
 + La tâche est considérée comme terminée lorsque l'élément est fonctionnel et que les tests sont réussis.  
\* Estimations en termes de points de complexité ou de temps:  
 1. Enregistrement et modification des produits en stock: 10-15 points  
 2. Saisie des quantités et des prix de vente: 8-12 points  
 3. Générations des rapports sur les soldes actuels et historiques: 5-8 points  
  
\*\*Tableau Récapitulatif\*\*  
  
| Élément | Expression des Besoins | Cahier des Charges | Backlog |  
| --- | --- | --- | --- |  
| Présentation du projet | | Site web pour une application de gestion de stock | |  
| Contexte et problématique | La gestion des stocks est un processus complexe | La gestion des stocks est un processus complexe | |  
| Périmètre du projet | Le site web sera développé pour une application de gestion de stock | Le site web sera développé pour une application de gestion de stock | |  
| Spécifications fonctionnelles détaillées | Saisie des produits en stock, Gestion des quantités et des prix de vente, Générations des rapports sur les soldes actuels et historiques | Saisie des produits en stock, Gestion des quantités et des prix de vente, Générations des rapports sur les soldes actuels et historiques | |  
| Spécifications techniques | Langage de programmation: Python ou JavaScript, Framework web: Flask ou Django, Base de données: MySQL ou PostgreSQL | Langage de programmation: Python ou JavaScript, Framework web: Flask ou Django, Base de données: MySQL ou PostgreSQL | |  
| Critères de réussite et indicateurs de performance | Temps de chargement inférieur à 3 secondes, Sécurité: absence de vulnérabilités critiques pendant au moins 6 mois, Usabilité: score supérieur à 80% pour l'interface utilisateur | Temps de chargement inférieur à 3 secondes, Sécurité: absence de vulnérabilités critiques pendant au moins 6 mois, Usabilité: score supérieur à 80% pour l'interface utilisateur | |  
| Risques et plans de mitigation | Risque de sécurité: mise en place d'un plan de sécurité solide avec des tests réguliers, Risque de performance: mise en place d'une architecture scalable avec des tests de charge | Risque de sécurité: mise en place d'un plan de sécurité solide avec des tests réguliers, Risque de performance: mise en place d'une architecture scalable avec des tests de charge | |  
  
Note: Les priorités peuvent varier en fonction des besoins et des contraintes du projet.