



REPUBLIQUE DU BENIN



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE (MESRS)

UNIVERSITÉ D'ABOMEY CALAVI (UAC)

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE (IFRI)

Présentation du cahier de charges de l'application de gestion de stocks

Réalisé par :

- 1- ALLEY Nadirathou
- 2- AWADJI Léonardo
- 3- AWOLOSSOU Albéric
- 4- HOUDJI Emmanuel
- 5- KONGORO Soabirou
- 6- OLOUKA Auréole
- 7- SOULE CHABI Ryad

Sous la supervision de:

M Erick ADJE

ANNEE ACADEMIQUE : 2023-2024

PLAN

I- Contexte et objectifs

II- Périmètre

III- Description fonctionnelle

IV- Ressources et exigences

V- Contraintes et Délais

VI- Livrables

VII- Sécurité et Confidentialité

I- Contexte et objectifs

✓ Contexte

De nos jours plusieurs entreprises éprouvent d'énormes difficultés dans gestion de ses stocks. On peut citer : existence des bugs dans le système, manque de coordination et de synchronisation dans la gestion en équipe des stocks, difficulté à gérer dans le même temps les produits qui constituent les stocks, difficulté à identifier les ruptures de stocks.

C'est dans le but de remédier à ses problèmes que notre équipe s'est engagée pour développer une application mobile de gestion de stock appelée **Stock Manager** capable de répondre aux attentes des entreprises.

✓ Objectifs

L'objectif principal de notre application est de rationaliser et d'optimiser les opérations de gestion des stocks afin de :

- Optimiser la gestion et le contrôle des stocks de manière efficace pour anticiper et éviter les ruptures de stock ;
- Gérez en équipe de façon assez transparente les stocks ;
- Gérez au fur et à mesure les produits disponibles dans l'ensemble des stocks ;
- Recevoir des alertes en cas rupture de stock.

Ces objectifs visent à améliorer l'efficacité de la gestion des stocks et d'optimiser les opérations de l'entreprise.

II- Périmètre

✓ Cibles

Stock Manager est adressée à tout entrepreneur ou particulier dans le souci d'optimiser la gestion des stocks et disposant d'un smartphone.

✓ Exigences non fonctionnelles

En plus des fonctionnalités, l'application devra respecter les exigences non fonctionnelles suivantes :

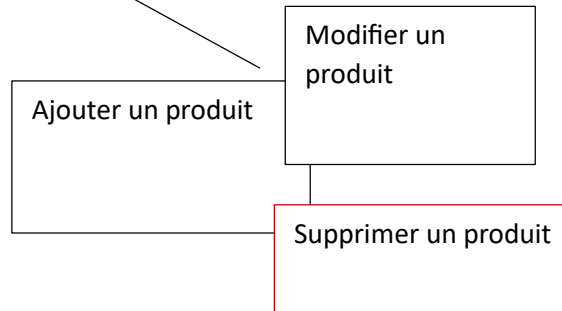
- **Sécurité** : L'application devra garantir la sécurité des données sensibles, en utilisant des protocoles de cryptage appropriés et en mettant en place des mécanismes d'authentification et d'autorisation.
- **Performance** : L'application devra être réactive et performante, même avec une grande quantité de données. Les temps de réponse devront être minimaux.
- **Compatibilité multi-plateformes** : L'application devra être compatible avec les principaux systèmes d'exploitation mobiles, notamment iOS et Android.
- **Intuitivité** : L'interface utilisateur de l'application devra être conviviale, intuitive et adaptée aux utilisateurs sans formation technique.

- **Disponibilité** : L'application devra être disponible en mode hors ligne, avec une synchronisation automatique des données une fois la connexion rétablie.
- **Maintenance et évolutivité** : L'application devra être facilement extensible pour prendre en charge de nouvelles fonctionnalités et des évolutions futures.

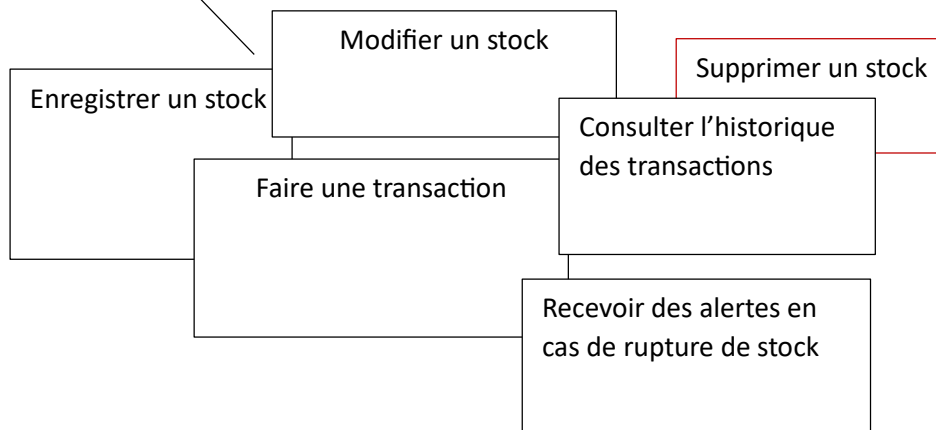
III- Description fonctionnelle

Notre application de gestion de stock a pour objectif principal de permettre sous plusieurs aspects le contrôle des stocks de manière efficace. Le schéma suivant fait ressortir ses différentes fonctionnalités et sous-fonctionnalités.

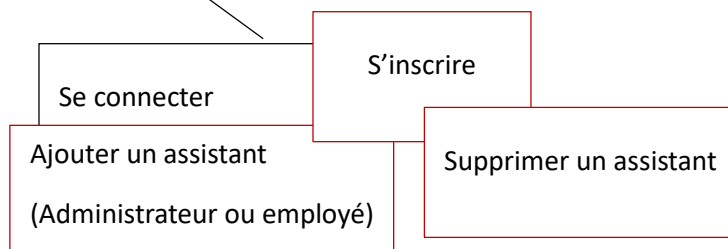
Gestion des produits



Gestion des stocks



Gestion des assistants



III- Ressources et exigences

✓ Ressources

Les technologies nécessaires pour la mise en place de Stock Manager sont :

1. **Flutter** : Framework multiplateforme pour le développement d'applications mobiles.
2. **Dart** : Langage de programmation utilisé avec Flutter pour écrire du code.
3. **Firebase** : Plateforme de développement d'applications mobiles de Google offrant plusieurs services, notamment :
 - Firestore : Base de données NoSQL de Firebase pour stocker et gérer les données.
 - Firebase Cloud Messaging : Pour les notifications push.
4. **Flutterfire** : Pour la communication entre l'application Flutter et Firebase.

En plus de ces technologies principales, voici quelques autres outils et ressources utiles :

1. **Version Control System (VCS)** : Comme Git pour le suivi des modifications de code.
2. **IDE** (Environnement de développement intégré) : Comme Android Studio, Visual Studio Code pour le développement Flutter.
3. **UI/UX Design Tools** : Comme Figma, Adobe XD pour la conception d'interfaces utilisateur.
4. **Documentation et Support** : Documentation officielle de Flutter, Dart, et Firebase, ainsi que les forums et communautés en ligne pour des discussions et de l'aide supplémentaire.

✓ Exigences fonctionnelles de l'application de gestion de stocks

En se basant sur les descriptions fonctionnelles de Stock Manager, les exigences fonctionnelles de l'application sont :

1. **Gestion des Produits** :

- Ajout, modification et suppression de produits.

2. **Gestion des Stocks** :

- Enregistrement des stocks avec des détails complets (seuils, lots, emplacements).
- Envoi d'alertes pour les niveaux de stock faibles.
- Historique des transactions de stock.

3. **Gestion des Utilisateurs** :

- Rôles distincts : propriétaire (gérant) et assistant (administrateur ou employé).
- Le propriétaire peut ajouter des assistants, qui selon leur rôles d'administrateur ou de simple employé seront restreints ou non de fonctions comme la suppression d'articles.

IV- Contraintes et Délais

1. Contraintes Techniques :

L'application de gestion de stocks doit respecter les spécifications techniques suivantes :

- **Compatibilité des appareils** : Nous nous assurerons que notre application fonctionne correctement sur différentes tailles d'écrans et résolutions, ainsi que sur différentes versions d'Android et iOS.
- **Sécurité** : Les applications de gestion de stocks gèrent souvent des données sensibles. Pour cela nous essayerons d'implémenter des mesures de sécurité robustes pour protéger les données de Stock Manager.
- **Performances** : Stock Manger devra être capable de gérer un volume élevé de données simultanés sans compromettre sa performance.

2. Contraintes Budgétaires :

Le budget alloué au projet est fixé à un montant spécifique. Toutes les dépenses liées au développement, à la mise en œuvre, aux licences logicielles et aux services tiers.

Ainsi voici une estimation du budget :

- Coût du développement : 800 000 FCFA
- Coût des outils et des licences : 50 000 FCFA
- Coût des services cloud : 150 000 FCFA (estimation annuelle)

Total estimé pour le développement initial de Stock Manager : 1 000 000 FCFA

3. Contraintes Réglementaires :

Stock Manager se conformera à un ensemble de contraintes réglementaires pour garantir la légalité et la sécurité des données. Aussi elle garantira la confidentialité des données de l'entreprise, y compris les informations sur les produits, les fournisseurs et les transactions.

La mise en œuvre complète de Stock Manager suivra le calendrier suivant :

- * Phase de Planification : une (01) semaine
- * Phase de Développement : huit (08) semaines
- * Phase de Tests : deux (02) semaines
- * Phase de Mise en Production : une (01) semaine

V- Livrables

A la fin du projet, nous aurons à fournir :

- Une **documentation en ligne** : elle sera constituée principalement de tous les aspects du cycle de vie logiciel bien détaillé et expliqué (analyse des besoins, planification, conception de logiciel, développement de logiciel, test, déploiement, maintenance) puis du cahier de charge du projet.

- Une **politique de confidentialité** également en ligne
- Une **maquette graphique** de Stock Manager
- Un **fichier apk** de l'application à télécharger en ligne (le code source de l'application ne sera pas fourni).

VI- Sécurité et Confidentialité

1. Gestion des accès : Il faudra s'assurer que seuls les utilisateurs autorisés puissent accéder à Stock Manager. Nous utiliserons des mécanismes d'authentification solides, comme les identifiants uniques et les mots de passe robustes.
2. Chiffrement des données : Les données sensibles seront protégées en les chiffrant, à la fois lorsqu'elles sont stockées dans la base de données et lorsqu'elles sont en transit entre l'application et le serveur.
3. Contrôle des droits d'accès : Nous définirons des niveaux de permission pour les utilisateurs afin de limiter leur accès aux fonctionnalités et aux données spécifiques en fonction de leurs rôles et responsabilités.
4. Journalisation des activités : Les activités des utilisateurs et les actions effectuées seront enregistrées dans l'application pour des raisons de sécurité et d'audit.
5. Sauvegardes régulières : Des sauvegardes régulières des données seront effectuées pour éviter toute perte en cas de problème technique ou de sinistre.
6. Politique de confidentialité : Nous fournirons une politique de confidentialité claire et transparente pour informer les utilisateurs sur la collecte, l'utilisation et la protection de leurs données personnelles.
7. Mises à jour de sécurité : Notre équipe veillera à maintenir l'application à jour en installant régulièrement les correctifs de sécurité et les mises à jour du système d'exploitation.