

Cahier des charges

Modernisation du système des reliquats



SOMMAIRE

1.Introductionet contexte.....	3
2.Objectif général du projet et périmètre.....	4
3.Description fonctionnelle des besoins.....	5
a.Application poste production.....	5
b.API Rest	7
c.Application mobile.....	9
d.Application de consultation.....	10
e.Evolution de la base de données.....	11
4.Les ressources.....	12
a.Les ressources logicielles.....	12
b.Les ressources matérielles.....	12
5.Les délais.....	13
a.Développement de l'application poste de production.....	13
b.Conception et développement de l'API Python.....	13
c.Développement de l'application mobile Android.....	14
d.Développement de l'application de consultation.....	14
e.Correction, formation et rédaction de la documentation.	14
f.Déploiement et mise en production.....	14
6.Les besoins financiers et budget.....	15
7.Conclusion.....	15
8.Historique des versions.....	16

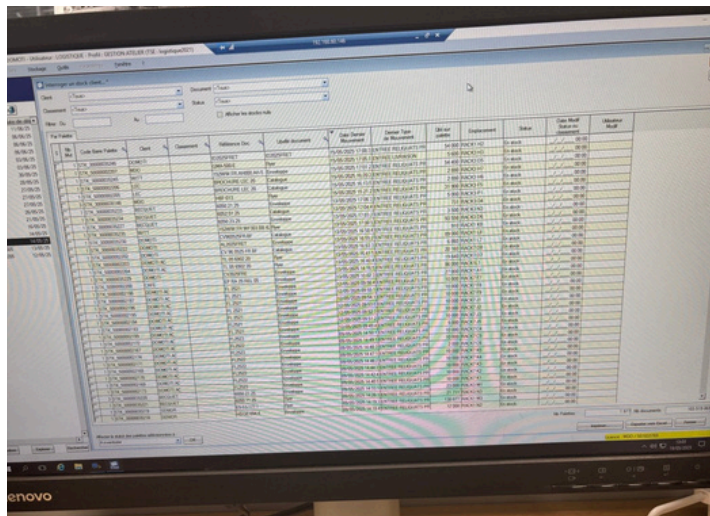
1 Introduction et contexte

Le processus actuel de saisie des reliquats manque de modernité. Le principal problème est l'usage des terminaux portables, utilisées par les caristes pour enregistrer les informations de stockage et de statut des palettes. Ces équipements sont vieillissants, extrêmement lents, peu ergonomiques et surtout sujets à des problèmes de synchronisation avec la base de données.

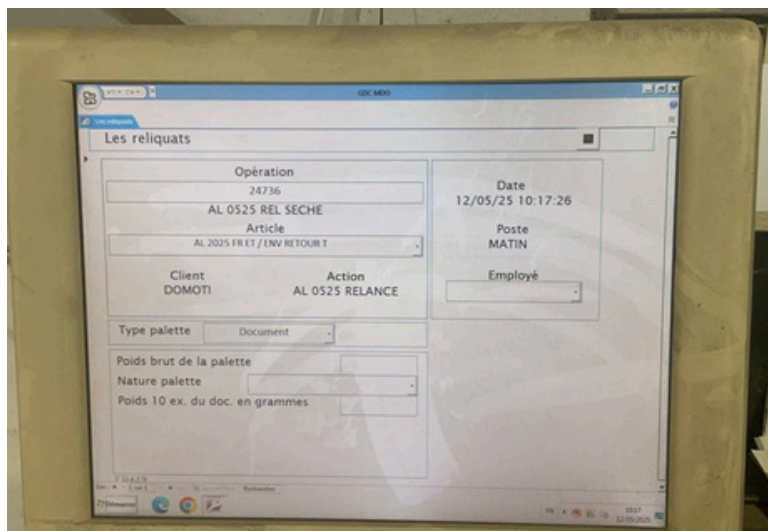
Par ailleurs, la base de données centrale GDC (Gestion des documents clients), utilisée pour gérer les opérations, documents, clients et employés, ne contient pas les informations d'état, d'emplacement ou de mouvements des palettes. Ces données sont exclusivement stockées dans la base de l'application tierce TriadPlus, sans lien dynamique avec GDC. Cette double gestion des informations entraîne une forte dépendance à TriadPlus qui est amené à disparaître dans le futur. Cela complexifie le suivi des stocks et nuit à l'évolutivité du système global.



Terminal actuel



Application de consultation



Application poste de production

2 Objectif général du projet et périmètre

Afin de pallier ces limitations, le projet vise à concevoir un ensemble d'applications modernisées, destinées à remplacer les terminaux portables actuels ainsi que l'application hébergé dans TriadPlus à laquelle l'entreprise est aujourd'hui fortement dépendante. L'objectif principal est de centraliser l'ensemble des données logistiques dans une nouvelle base de données DRB (DJADJA Reliquat Base) hébergé sur srv-data, conteneurisé sur une machine virtuelle Linux. Cela permettra d'unifier le système d'information et de garantir une meilleure maîtrise des flux.

Le projet comprend quatre applications complémentaires :

- Une application embarquée sur terminaux mobiles sous Android (remplaçant les terminaux actuels), permettant aux caristes de gérer les entrées en stock, les mouvements de palettes, et l'affectation des emplacements dans une nouvelle base de données (DRB).
- Une API Rest développé en python permettant la connexion des terminaux portables à la base de données.
- Une application sur le poste de production, située en atelier, dédiée à la création des fiches de reliquats, avec le calcul des quantités, l'impression de bordereaux et la saisie de ces informations grâce à une récupération dans GDC puis une exportation dans la nouvelle base de données (DRB).
- Une application de consultation des stocks, à destination des agents de bureau et de la logistique, permettant de visualiser l'état des palettes, de suivre leurs mouvements et de formuler des demandes d'inventaire ou de destruction directement depuis une interface centralisée. Avec éventuellement un dérivé de cette application permettant la consultation des stocks reliquats pour les manutentionnaires de salle.

La finalité du projet est donc de bâtir une solution complète, cohérente et évolutive, intégrée à l'environnement existant, et capable à terme de supprimer toute dépendance à TriadPlus.

3 Description fonctionnelle des besoins

Le projet se décompose en trois applications principales, chacune répondant à des besoins spécifiques dans le processus de gestion des reliquats.

3.a Application poste de production - Création des fiches de reliquats

Cette application est le point d'entrée des palettes dans la base de données. Elle sert à avoir une interface simple pour les opérateurs d'atelier permettant de générer automatiquement des fiches de palettes à partir d'une fiche d'opération (plus précisément le numéro d'opération et quelques informations à saisir dans l'application).

Les fonctionnalités attendues :

- Saisie ou scan du numéro d'opération (opération.NumOperation).
- Chargement automatique depuis GDC des données liées à cette opération. A savoir le nom de l'opération (opération.Operation), le nom du client (clients.NomClient), le nom de l'action client (actions.ActionClient), la liste des articles associés à cette opération (articles.Designation) et la désignation de chaque article (articles.Designation).
- Saisie assistée des données physiques. Avec sélection du type de document(Document, Film ou Carton).
 - Type Document :
 - du type de palette (pour déterminer le poids à vide en Kg), le poids total de la palette et le poids de 10 exemplaires (g).
 - Type Film :
 - du type de palette (pour déterminer le poids à vide en Kg) et du poids total de la palette
 - Type Carton :
 - le nombre de cartons et la quantité par carton
- Calcul automatique de la quantité de documents.
- Incrémentation automatique du numéro de palette.
- Création du bordereau avec deux codes-barres (n° de palette et quantité).
- Impression automatique des fiches reliquats en passant par Adobe Acrobat Reader
- Enregistrement dans la base DRB (InfoPalette).
- Capacité à détecter les erreurs de saisie et d'afficher des messages clairs en cas de problèmes.
- Mode Manuel permettant d'éviter une connexion à GDC pour assurer le maintien d'un point d'entrée dans la base de données DRB.

Les contraintes techniques :

- L'application sera développée sur python 3.13.3 en version 32 bits pour un souci de compatibilité avec GDC. Avec les librairies tkinter, reportlab, pyodbc et dotenv.
- Le poste de production doit avoir accès au serveur Travail T: (ce qui est le cas) et avoir fait l'installation de GDC via le document GDC - FONCTIONNEMENT.
- Une douchette doit être disponible sur l'ordinateur.
- Une imprimante compatible avec le format A4.
- Gestion des erreurs en cas de problème.
 - Alerte en cas de données manquantes avec option de saisie manuelle au clavier.

3.b API Rest

Cette API constitue le pont central entre les terminaux mobiles et la base de données DRB. Elle garantit la cohérence des données, centralise la logique d'accès aux tables et expose des points d'entrée sécurisés pour interagir avec les palettes.

Les fonctionnalités attendues :

Connexion sécurisée à la base MySQL DRB à l'aide de variables d'environnement (.env).

Accès en lecture et écriture aux tables suivantes :

- InfoPalette (palettes en cours),
- SPR_Palette (palettes archivées),
- MVT_Palette (mouvements),
- STT_Palette (états).
- EmplacementEntrepôt (places dans l'entrepôt)

Endpoints REST accessibles par les différentes applications :

- POST /auth/login/: permet l'authentification grâce à un token
- GET /consultation/: Liste l'ensemble des palettes dans l'entrepôt avec des possibilités de trie dynamique
- PATCH /emplacement/palette : modification de l'emplacement d'une palette.
- GET /emplacement/ : permet de récupérer les emplacement de l'entrepôt (libre, occupé)
- GET /entree/palette/{num_palette} : recupère les informations d'une palette grâce à son numéro
- POST /entree/: permet de donner un emplacement ou de modifier l'emplacement d'une palette dans l'entrepôt

- GET /inventaire/ : récupère les palettes dans l'état "A Inventorier"
 - POST /inventaire/valider_inventaire/ : prends une liste de palettes et les bascules dans l'état "En Stock"
 - GET /sorties/destruction/ : Liste les palettes dont le statut est "A Détruire"
 - POST /sorties/valider_destruction: Valide la destruction d'une ou plusieurs palettes passage du statut à "Détruite"
 - GET /sorties/renvoie/ : Liste les palettes dont le statut est "A Renvoyer".
 - POST /sorties/valider_renvoye/ : Valide le renvoi d'une ou plusieurs palettes statut mis à "Renvoyé".
 - GET /sorties/production/ : Liste les palettes actuellement "En Stock".
 - POST /sorties/valider_production/ :Valide le passage en production des palettes sélectionnées changement de statut vers "En Prod".
-
- Vérification et validation des données avant insertion :
 - Champs obligatoires, formats, statuts valides, cohérence métier (ex. quantité > 0, statut autorisé, etc.).
 - Journalisation des actions :
 - Toutes les modifications importantes sont loguées (palettes ajoutées, statut modifié, utilisateur, horodatage).
 - Retour d'erreurs clairs et structurés en JSON :
 - Codes HTTP explicites (400, 404, 500...),
 - Messages détaillés en cas de problème (champ manquant, connexion échouée, requête invalide).
 - Séparation entre environnement de production et test, avec configuration distincte.
 - Compatibilité avec les scanettes Android et les applications fixes/desktop via appels HTTP standards (JSON).

Les contraintes techniques :

- Hébergé sur le serveur srv-Data conteneurisé sur une VM Linux.
- Langage de développement : Python 3.13
- Framework utilisé : FastAPI (framework REST moderne et performant)
- Connexion à la base de données : via PyMySQL vers la base MySQL DRB
- Exposition des services : API REST accessible en HTTPS grâce à un tunnel Cloudflare (connexion sécurisée sans port ouvert).

3.c Application mobile - Remplacement des terminaux portables

Cette application a pour but de remplacer les terminaux portables physiques actuelles utilisées par les caristes. Elle sera déployée sur des terminaux Android. Elle permettra de gérer les entrées en stock, de mettre à jour l'état des palettes, et d'enregistrer les emplacements dans la base DRB, sans passer par Triad. Elle vise à moderniser et fluidifier le travail de la logistique, tout en supprimant la dépendance à TriadPlus et en centralisant toutes les informations dans une seule base de données.

Les fonctionnalités attendues :

- Lecture du code-barre du numéro de palette (reliquat ou livraison) avec la saisie automatique de la référence ainsi que de sa désignation, du nom du client et de la quantité.
- Saisie de l'emplacement par scan d'un code-barre présent dans l'entrepôt ou par un menu déroulant hiérarchisé.
- Sélection du motif d'entrée (Reliquat production, Livraison ...).
- Transmission des données à la base (DRB) : mise à jour de l'état de la palette, l'enregistrement de l'emplacement ainsi que date, heure et utilisateur ayant changé pour la dernière fois le statut.
- Mode Inventaire listant toutes les palettes à inventorier. Scan du n° de palette et validation de la quantité réelle (passage en état "Inventaire OK"). Une fois que la liste est vide ou arrêt prématuré, nouvelle confirmation avec la liste des palettes scannées (pouvoir revenir en arrière si besoin) pour passer de "Inventaire OK" à "En Stock".
- Mode Destruction listant toutes les palettes à détruire. Scan du n° de palette avec confirmation visuelle (fond rouge et passage dans l'état "Détruite ok ?"). Une fois que la liste est vide ou arrêt prématuré, nouvelle validation avec la liste des palettes supprimées pour passer de "Détruire ok" à "Détruite".
- Mode Consultation pour simplement consulter et lister toutes les palettes.
- Affichage clair avec des couleurs par statut (rouge, vert, orange).
- Affichage clair en cas de données manquantes ou invalides.
- Possibilité d'enregistrer l'historique des mouvements et des statuts.
- Mode hors-ligne avec l'enregistrement local des actions si l'appareil est temporairement déconnecté, puis une synchronisation automatique ou manuelle dès que la connexion est rétablie avec une indication claire des actions en attente de synchronisation.

Les contraintes techniques :

- Système d'exploitation : terminal mobile sous Android
- Langage de développement : Java et Python

- Connexion à la base de données via une API REST (développée en Python64) connectée à la base GDC. L'application Android et l'API communiqueront en https grâce à la Wifi ou la 4G. Par l'intermédiaire d'un tunnel cloudflare.
- Sauvegarde des actions réalisés si il y a une perte de connection puis renvoies automatique des la connection rétablis.
- Sécurité en utilisant des tokens pour l'authentification, avec une validation des champs côté client avant envoi en assurant une synchronisation fiable.
- Gestion des erreurs en cas de problème.
 - Affichage de la connexion et notification du mode hors ligne.
 - Notification si des données ne sont pas trouvées dans la base de données.

3.d Application de consultation - Consultation et gestion des stocks

Cette application est destinée aux utilisateurs de bureau (chargés de clientèle, responsables logistiques...) pour consulter, rechercher et piloter les palettes en stock, sans passer par TriadPlus. Elle offrira une vue synthétique de l'état des reliquats et des livraisons, ainsi que des outils de demande d'action (inventaire, destruction, etc.).

Son but est de supprimer toute dépendance aux interfaces et à la base de donnée de Triad en rebasant tout le système sur la nouvelle base de données (DRB).

Les fonctionnalités attendues :

- Visualisation détaillé des palettes n° de palette, client, référence du document, libellé du document, date du dernier mouvement, dernier type de mouvement, quantité sur palette, l'emplacement, le statut, la date de modification du statut et l'utilisateur ayant modifié le statut en dernier. Recherche par menu déroulant des
- référence par client avec une manière d'auto suppression des référence non utilisé Recherche et filtrage des palettes par client, référence et statut. Ordonnancement
- possible par la date du dernier mouvement ou par la date de dernière
- modification. Possibilité de changer le statut des palettes avec multi-sélection. Export au format CSV possible pour analyse ou transmission externe.
- Visualisation des indicateurs comme le nombre de palettes par état ou alertes sur
- les palettes anciennes ou inactives depuis un certain temps avec archivage. Aide
- visuel en ayant une ligne sur deux grisé
-

Les contraintes techniques :

- L'application sera développée sur python 3.13.3 en version 64 bits. Avec les librairies PyQt5, pyodbc, dotenv et pandas.
- Le poste de production doit avoir accès au réseau local pour avoir accès à la base de donnée (DRB).

3.e Evolution de la base de données

Afin de permettre aux diverses applications de pouvoir être autonome et ne dépendre d'aucune autre base de données, il est nécessaire d'en créer une nouvelle. Celle-ci sera implantée sur le serveur srv-Data sur une VM Linux, puis conteneurisée. La base de données sera accessible via le port 3306 à l'adresse 192.168.60.238.

La table InfoPalette et SPR_Palette qui servira de sauvegarde des palettes supprimées contenant:

- NumPalette : indique le numéro de palette
- NumOperation : indique le numéro d'opération
- Operation : indique le nom de l'opération
- Article : indique la référence de l'article
- NomClient : indique le nom du client correspondant à la référence sur la palette.
- ActionClient : le nom de l'action client
- Quantite : indique la quantité présente sur la palette
- Emplacement : localisation physique de la palette dans l'entrepôt.
- Statut : indique l'état logistique de la palette (En stock, À inventorier, À détruire...).
- Date_Modif_Statut: date de la dernière modification de statut.
- Utilisateur_Modif_Statut : identifiant de l'opérateur ayant modifié l'état.
- Date_Dernier_MVT : Etant la date à laquelle la palette a changé d'état et le dernier type de mouvement.
- Employe : Employé ayant édité la fiche de reliquat
- Date_Entree_Reliquat : Date à laquelle la fiche de reliquat a été éditée

Une table MVT_Palette contenant :

- NumPalette : le numéro de la palette.
- Date_Dernier_MVT : Etant la date à laquelle la palette a changé d'état et le dernier type de mouvement.
- Zone : l'endroit où se situe la palette (Entrepôt, Production ...)

Une table EmplacementEntrepot contenant :

- Emplacement : Le numéro du rack et la lettre
- Etat : occupé ou libre

Une table Users contenant :

- Login : nom d'utilisateur
- Password : mot de passe de l'utilisateur haché
- Role : change les droits sur l'application de consultation

Une table STT_Palette contenant :

- NumPalette : le numéro de la palette.
- Statut : indique l'état logistique de la palette (En stock, À inventorier, À détruire...).
- Date_Modif_Statut: date de la dernière modification de statut.
- Utilisateur_Modif_Statut : identifiant de l'opérateur ayant modifié l'état.

4 Les ressources

La réalisation de ce projet nécessite des ressources logicielles et matérielles.

4.a Les ressources logicielles

- Python 3.13.3 en 32 bits (nécessaire pour l'accès à Access via pyodbc uniquement sur le poste fixe) et 64 bits pour le reste.
 - PyQt5 pour l'interface de bureau.
 - pyodbc pour la connexion à Access.
 - reportlab pour la génération de pdf.
 - dotenv pour les variables d'environnement.
 - pandas pour l'export de donnée en format CSV.
 - pyinstaller pour la création de .exe déployables.
- Android Studio pour le développement de l'application en Java.
- Visual Studio Code comme IDE.
- Partage réseau du fichier Access GDC via le serveur Travail (t:) avec accès local en lecture-écriture depuis les applications.
- Base de donnée (DRB) sur le serveur 192.168.60.238 au port 3306.

4.b Les ressources matérielles

Afin d'assurer la mise en œuvre complète du projet, une commande de matériel est à prévoir, comprenant :

- 1 ou 2 Android durcis (ou smartphones pro) avec caméra ou scanner intégré pour la lecture de code-barres. Les modèles Zebra TC58 sont parfaitement adaptés. Ils sont robustes, disposent d'une grande autonomie et compatibles avec les applications Android développées en Java. Et possèdent une connexion 5G ou WIFI. Amplificateurs ou points d'accès Wi-Fi supplémentaires pour assurer une
- connexion la plus stable possible entre l'application mobile et l'API locale (Si en wifi).

5 Les délais

Le projet sera structuré en plusieurs étapes de développement pour assurer une validation par étape et un déploiement sans interruption de l'activité. La durée totale du projet est estimée à 10-12 semaines, incluant les phases de développement, de test, de recette et de déploiement.

5.a Développement de l'application poste de production

Durée estimée : 1 semaine

- Création de l'interface PyQt5
- Connexion à la base Access via pyodbc
- Génération et impression du bordereau PDF
- Enregistrement dans la table InfoPalette
- Tests unitaires et fonctionnels

5.b Conception et développement de l'API Python

Durée estimée : 2 semaines

- Création des endpoints REST
- Connexion sécurisée à la base Access
- Gestion des requêtes GET / POST / PUT
- Documentation technique de l'API (Swagger ou Markdown)

5.c Développement de l'application mobile Android

Durée estimée : 3 à 4 semaines

- Développement en Java (interface, scan, formulaire)
- Connexion à l'API REST (Retrofit / JSON)
- Stockage local en mode hors-ligne (SQLite)
- Gestion des synchronisations et erreurs
- Tests terrain avec les terminaux Android

5.d Développement de l'application de consultation

Durée estimée : 2 semaines

- Interface bureau avec PyQt5
- Recherche, filtres, visualisation des palettes
- Demandes d'inventaire et de destruction
- Export Excel ou CSV

5.e Correction, formation et rédaction de la documentation

Durée estimée : 1 semaines

- Tests avec utilisateurs finaux
- Correction des bugs et ajustements ergonomiques
- Rédaction des guides utilisateurs
- Formation des opérateurs et caristes

5.f Déploiement et mise en production

Durée estimée : moins d'une semaine

- Installation des exécutables sur les postes fixes
- Déploiement de l'APK sur les terminaux Android
- Test en conditions réelles
- Suivi post-déploiement (stabilisation)

6 Les besoins financiers et budget

La mise en œuvre du projet nécessite la mobilisation de ressources financières pour l'acquisition des nouveaux terminaux tournant sous Android et éventuellement des points d'accès Wifi ou répéteur pour permettre une meilleure couverture en réseau de l'entrepôt.

- Terminaux Android durcis avec un Zebra pistol grip(ex : Zebra TC58) :
 - 2 unités - ~ 2500€ HT/unité.
- Points d'accès Wi-Fi (si besoin) : 1 ou 2 - ~100 € HT/unité

7 Conclusion

Ce projet de refonte du processus de gestion des reliquats vise à moderniser les outils utilisés en atelier et à centraliser les informations au sein de la base de données GDC, tout en supprimant la dépendance à des solutions tierces peu évolutives comme Triad. En s'appuyant sur des technologies robustes, des interfaces adaptées aux environnements industriels et une architecture modulaire, le système proposé permettra d'améliorer l'évolutivité du système, la réactivité et l'efficacité des opérations logistiques.

La réussite du projet repose sur une bonne coordination entre les acteurs métiers et techniques, ainsi qu'un déploiement progressif et maîtrisé. Ce cahier des charges constitue le socle de référence pour la conception, le développement, le test et la mise en production de l'ensemble des applications prévues.

8 Historique des versions

V1.0 Jeudi 22 Mai 2025, Rédaction du cahier des charges

V1.1 Vendredi 6 Juin 2025, mise à jour et ajout de certaine fonctionnalités à implanter

V1.2 Jeudi 3 Juillet 2025, mise à jour global (VM) avec ajout des spécificité de l'API.