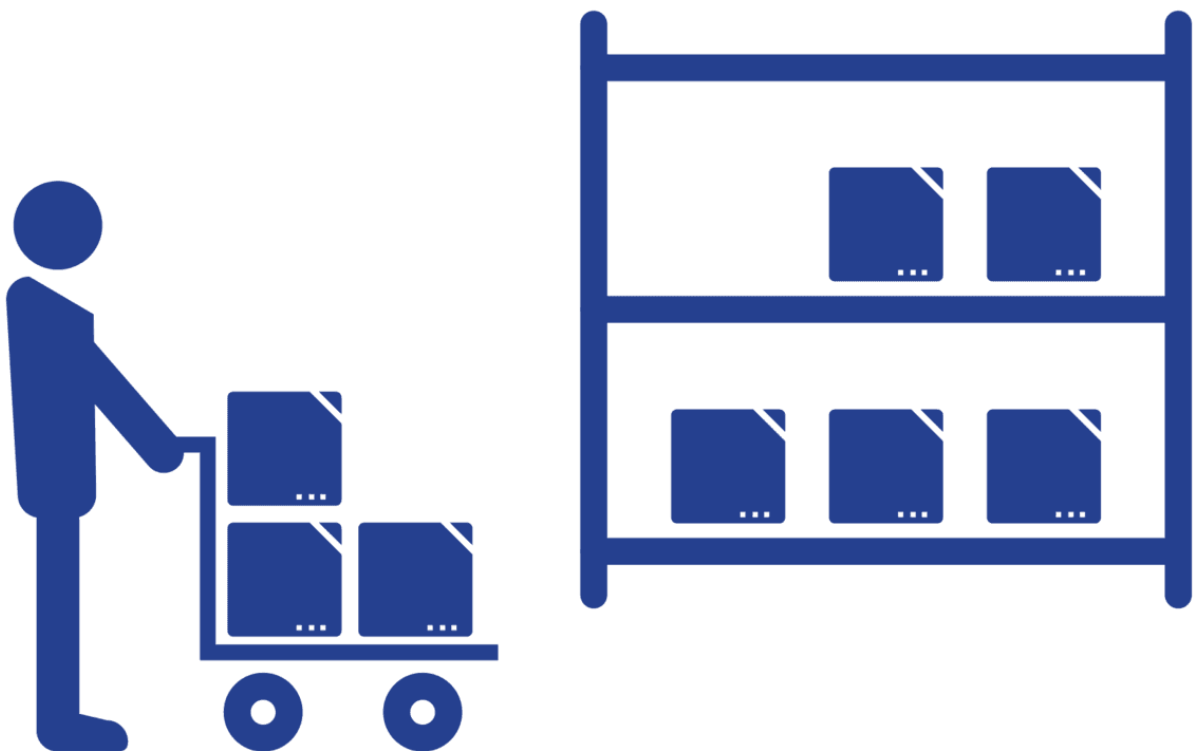


P_Appro 2



Egal Ahmed Omar – FIN2

ETML-Vennes

Helder Costa Lopes

Table des matières

Liste des difficultés à identifier	2
Semaine du 17 au 21 Mars	2
Conception de l'interface utilisateur :	2
Intégration de la base de données :	2
Suivi des mouvements de stock :	3
Gestion des exceptions et sécurité :	3
Documentation et maintenance :	4
Semaine du 24 au 28 Mars	4
Intégration de la base de données :	4
Fonctionnalités CRUD :	4
Suivi des mouvements de stock :	5
Semaine du 7 au 11 Avril	6
Gestion des exceptions et sécurité :	6
Optimisation des performances :	6

Liste des difficultés à identifier

Semaine du 17 au 21 Mars

Conception de l'interface utilisateur :

Comment concevoir une interface ergonomique tout en respectant les contraintes de simplicité ?

Pour créer une interface ergonomique tout en respectant les exigences de simplicité, il est crucial d'utiliser des mots significatifs et simples pour la navigation. Il faut éviter la redondance et les scénarios complexes.

Aérer les blocs, les espaces vides ne sont pas nécessairement exploitables. Il est également impératif de raccourcir les longues séquences, un formulaire sur une seule page est plus pertinent qu'un assistant interminable. Les boutons, les formes des blocs, les styles et l'iconographie doivent être les mêmes dans toutes les sections.

Quelles sont les meilleures pratiques pour organiser les formulaires et rendre l'expérience utilisateur fluide ?

Afin d'organiser de manière efficace les formulaires et d'assurer une expérience utilisateur agréable, il est conseillé de créer des formulaires concis et intuitifs, en réduisant le nombre de champs visibles au strict nécessaire.

D'autre part, La classification logique des informations augmente la clarté, par exemple en distinguant les données générales, le prix et le stock.

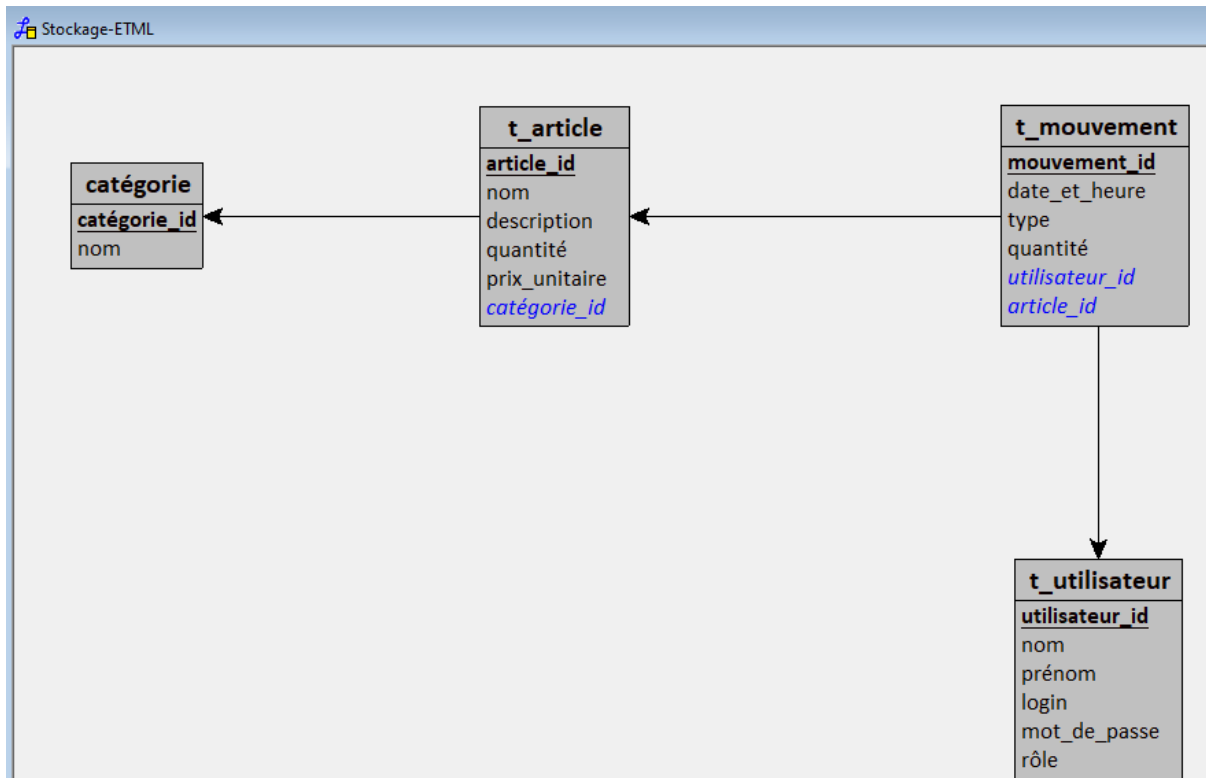
Intégration de la base de données :

Quelle solution de base de données choisir (SQL Server, SQLite, etc.) et comment l'intégrer efficacement ?

La solution de base de données choisie est MySQL. Pour l'intégrer à l'application Windows form, le package **MySql.Data** doit être intégré à la solution. Ensuite, **Docker** facilite le déploiement et l'isolation de l'environnement de la base de données. phpMyAdmin compte à lui permet l'administration de la base de données.

Comment structurer la base de données pour assurer une bonne organisation des données (tables, relations, index) ?

Pour assurer une bonne organisation des données, il est essentiel de définir des tables bien structurées, notamment t_article, t_catégorie, t_mouvement et t_utilisateur. Les relations doivent être claires, avec un article associé à une catégorie selon une relation (1,N), et chaque mouvement de stock étant lié à un article et un utilisateur également en (1,N).



Enfin, l'intégrité des données doit être garantie à l'aide de contraintes telles que les clés étrangères (Foreign Keys), l'obligation de certains champs (NOT NULL).

Suivi des mouvements de stock :

Comment gérer les entrées et sorties de stock tout en conservant un historique précis ?

Une table **t_mouvement** avec les champs type (entrée/sortie), quantité, date, id_utilisateur, id_article permet de gérer les entrées et sorties de stock tout en conservant un historique précis.

Une mise à jour automatique du stock est effectuée lors d'un ajout de mouvement.

Gestion des exceptions et sécurité :

Comment traiter les erreurs techniques (ex. : connexion à la base de données rompue, fichier manquant) ?

Utiliser des blocs « try-catch » autour des opérations critiques (ex. : connexion DB et au CRUD). Et Affichage de Messages d'erreur clairs pour l'utilisateur.

Faut-il implémenter une authentification pour restreindre l'accès à certaines fonctionnalités ?

Oui, avec différents rôles :

- Admin : Gestion complète (CRUD articles, utilisateurs, mouvements).
- Utilisateur standard : Peut uniquement voir le stock et enregistrer des mouvements.
- Authentification sécurisée avec un login/mot de passe.

Documentation et maintenance :

Comment documenter le code et les choix techniques pour faciliter la maintenance future ?

Doxygen est un générateur de documentation sous licence libre capable de produire une documentation logicielle à partir du code source d'un programme. Il est donc plus qu'utile à la maintenance future de l'application.

Quelles sont les bonnes pratiques pour garantir une évolutivité du projet ?

L'utilisation de norme et pattern de codage sont des bonnes pratiques pour garantir une évolutivité du projet. Les normes de codage sont celle de l'ETML et le pattern MVC (Modèle-Vue-Contrôleur) sera implémenté.

Semaine du 24 au 28 Mars

Intégration de la base de données :

Quelles sont les bonnes pratiques pour écrire des requêtes SQL optimisées ?

- Sélectionner uniquement les colonnes nécessaires.
- Utiliser des index pour accélérer les recherches.
- Privilégier les JOIN aux sous-requêtes.
- Filtrer les données avec WHERE et LIMIT.
- Éviter les fonctions sur les colonnes dans WHERE.

Fonctionnalités CRUD :

Comment implémenter les opérations CRUD de manière robuste et sans erreur ?

1. **Création (Create)** : Valider les entrées avant insertion, utiliser des transactions pour garantir la cohérence et gérer les erreurs pour éviter les saisies invalides.
2. **Lecture (Read)** : Sélectionner uniquement les colonnes nécessaires, utiliser des filtres et des index pour optimiser les performances et limiter la surcharge de données.
3. **Mise à jour (Update)** : Vérifier l'existence de l'élément, ne modifier que les champs nécessaires et gérer les erreurs pour éviter les incohérences.
4. **Suppression (Delete)** : Vérifier les dépendances avant suppression, demander une confirmation et privilégier une suppression logique si nécessaire.

Quelles validations doivent être mises en place pour éviter les données invalides ou corrompues ?

Pour garantir la fiabilité des informations, il est crucial de s'assurer que toutes les données essentielles sont correctement complétées. Le nom, la quantité ou le prix ne doivent pas rester vides, et les valeurs entrées doivent être cohérentes, par exemple une quantité toujours positive et un prix acceptable.

La base de données doit également appliquer des règles rigoureuses. Certains champs ne doivent pas être laissés vides, d'autres doivent suivre des règles, comme une quantité qui ne doit jamais tomber en dessous de zéro. Pour prévenir les erreurs, il est également bénéfique d'éliminer les doublons sur des éléments essentiels tels que le nom d'un produit.

Les liens entre les données doivent être correctement administrés. Par exemple, un article ne peut pas être enlevé s'il est encore présent dans des mouvements de stocks. Il est également important de s'assurer que toutes les connexions entre les tableaux sont correctes et, si besoin, d'automatiser des mises à jour pour prévenir les incohérences.

Enfin, certaines règles de gestion doivent être suivies, comme éviter les doublons superflus, garantir que les dates saisies sont cohérentes et empêcher qu'un stock tombe en négatif après un retrait. Ces mesures assurent une gestion efficace et sûre des articles.

Suivi des mouvements de stock :

Faut-il intégrer des alertes pour les stocks critiques ou des seuils de réapprovisionnement ?

Pourquoi intégrer des alertes ? Intégrer des alertes permet d'éviter les ruptures de stock en notifiant les responsables dès qu'un article atteint un seuil critique. Cela optimise le réapprovisionnement en déclenchant une alerte avant que le stock ne soit trop bas. Cela améliore également la réactivité et la satisfaction client en réduisant les risques de commandes non honorées.

Un tableau de bord peut être mis en place pour afficher les articles en alerte et permettre une vision rapide des besoins en réapprovisionnement.

Types d'alertes possibles. Les alertes peuvent être visuelles avec des icônes rouges mises en évidence dans la liste des articles.

Semaine du 7 au 11 Avril

Gestion des exceptions et sécurité :

Quelles mesures de sécurité mettre en place pour protéger les données sensibles ?

Les seules données sensibles sont les mots de passe utilisateurs. Ces dernières seront hashées avant d'être stockées dans la base de données directement avec la requête SQL lors de l'enregistrement d'un nouvel utilisateur.

Optimisation des performances :

Comment garantir que l'application reste réactive même avec un volume croissant de données ?

Pour garantir la réactivité de l'application plusieurs stratégies existent :

1. Optimiser les requêtes SQL afin de sélectionner uniquement les données qui nous sont nécessaires.
2. Une pagination efficace permet d'éviter l'affichage de toutes les données en même moment.
3. Nettoyer et archiver les données obsolètes permet de garder une DB légère et rapide.
4. Utilisation d'un modèle MVC garantit une meilleure organisation et des performances plus stables.

Quelles stratégies d'optimisation peuvent être appliquées (indexation, pagination, cache, etc.) ?

Un système de pagination a été mis en place lors de la consultation des mouvements et des articles.

Un système de recherche et de filtre permet également d'optimiser la consultation des données à travers l'application.