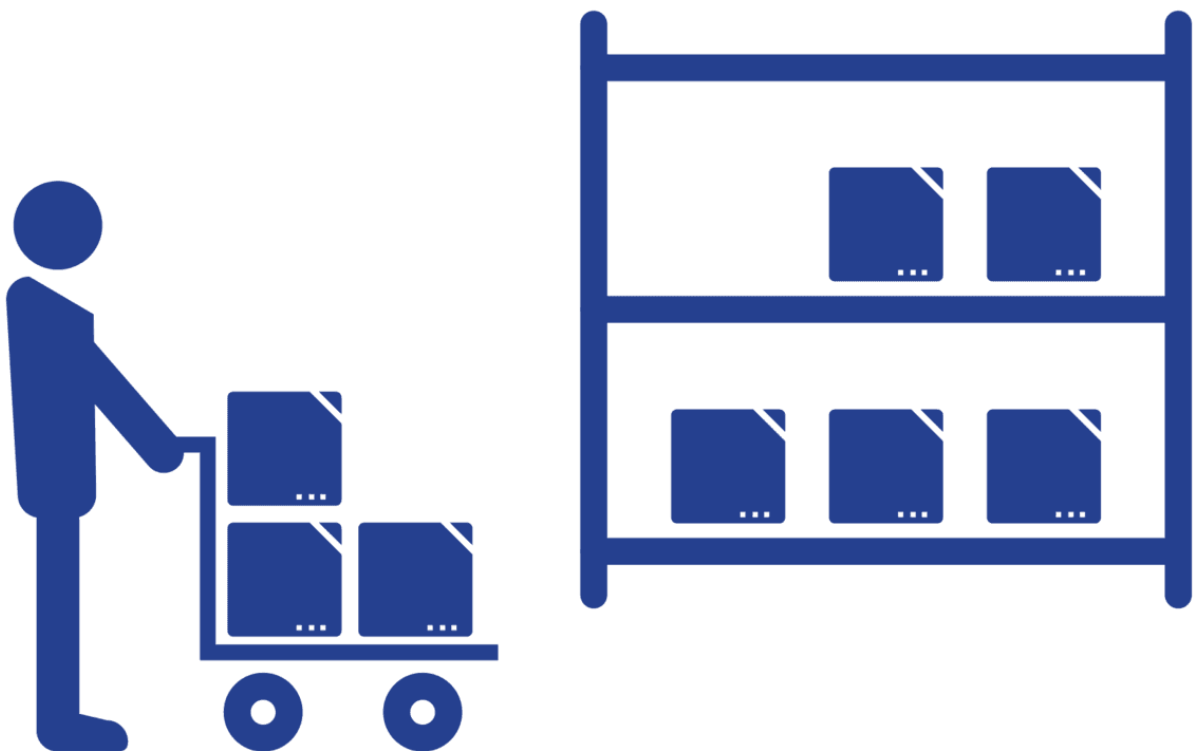


P_Appro 2



Egal Ahmed Omar – FIN2

ETML-Vennes

24 périodes

Helder Costa Lopes

Table des matières

Liste des difficultés à identifier	2
Semaine du 17 au 21 avril :	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

Liste des difficultés à identifier

Semaine du 17 au 21 avril

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.

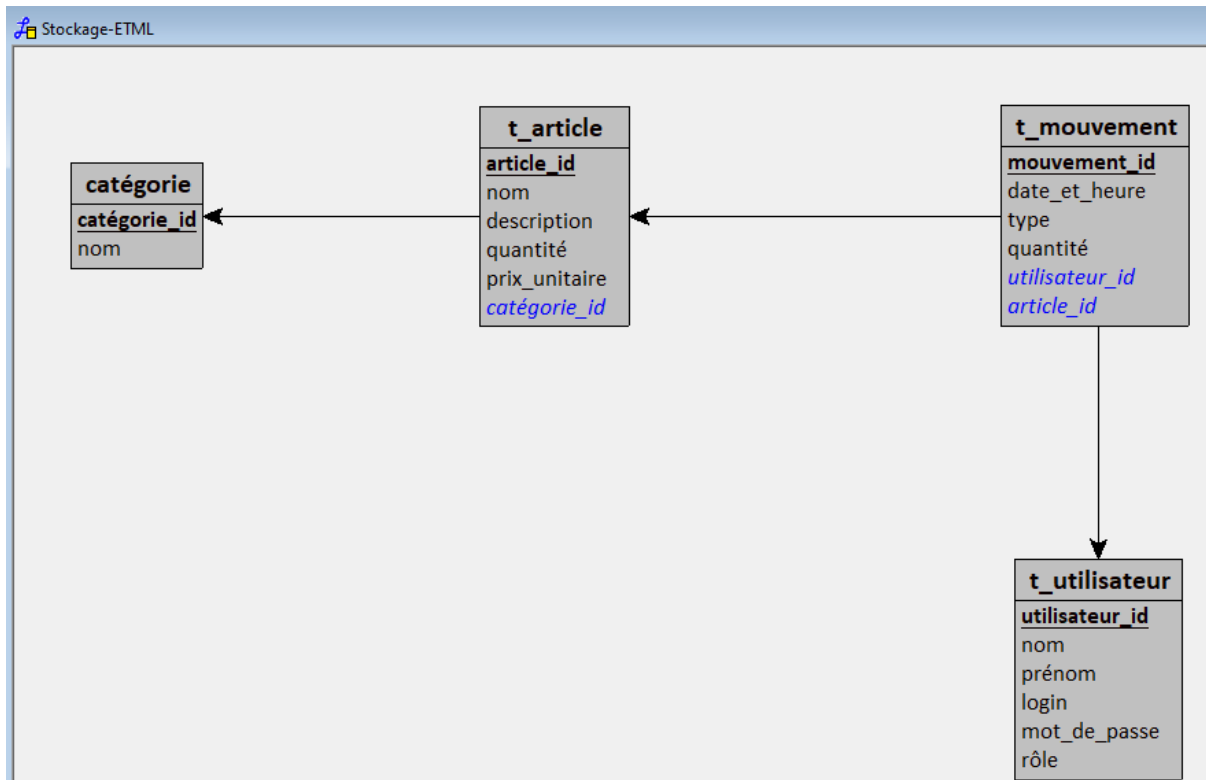
Intégration de la base de données :

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 du l'ETML et le pattern MVC (Modèle-Vue-Contrôleur) sera implémenté.