

Cahier des charges de l'application EcoPlan – Gestionnaire de Budget

1. Présentation du projet

EcoPlan est une application Java de type client lourd développée dans le cadre du BTS SIO (option SLAM).

Son objectif principal est de fournir un outil pédagogique et fonctionnel permettant aux utilisateurs, notamment les jeunes, de mieux gérer leur budget personnel, d'anticiper les dépenses, et de visualiser leur situation financière via des graphiques simples.

2. Objectifs généraux de l'application

- Offrir une interface intuitive pour la gestion des finances personnelles.
- Proposer un système de suivi des dépenses et revenus.
- Intégrer des statistiques sous forme de graphiques pour visualiser les tendances budgétaires.
- Sensibiliser les utilisateurs à l'équilibre budgétaire.

3. Public cible

- Étudiants
- Jeunes actifs
- Toute personne souhaitant mieux gérer son budget

4. Fonctions principales de l'application

- Création et gestion de comptes utilisateurs.
- Ajout, modification et suppression de dépenses et revenus.
- Catégorisation des transactions (alimentation, transport, loisirs...).
- Calcul automatique du solde mensuel.
- Affichage de statistiques via histogrammes et camemberts.

5. Technologies utilisées

- Langage de développement : Java
- Interface utilisateur : Java Swing / AWT
- Base de données : SQLite (fichier .db local)
- Connexion à la base : JDBC
- Graphiques : JFreeChart ou JavaFX Charts
- Versionnage : Git / GitHub

6. Architecture logicielle

L'application est conçue en suivant le modèle MVC (Modèle-Vue-Contrôleur), avec séparation claire entre l'interface, la logique métier et la persistance des données.

7. Contraintes techniques

- Application client lourd (non web)
- Stockage local des données
- Compatible avec les systèmes Windows/Linux
- Interface responsive en Swing

8. Maquettes et interface utilisateur (UI)

L'interface est simple, épurée, avec un menu latéral pour naviguer entre les sections (Accueil, Ajouter une dépense, Statistiques, Paramètres).

Les formulaires sont conçus pour une saisie rapide, avec des menus déroulants pour les catégories.

9. Base de données

La base de données SQLite comporte les tables suivantes :

- Utilisateur : id, nom_utilisateur, mot_de_passe
- Transaction : id, utilisateur_id, montant, date, catégorie, description
- Catégorie : id, nom

10. Sécurité

- Authentification de l'utilisateur avec mot de passe.
- Chiffrement possible des mots de passe (via SHA-256 ou équivalent).
- Sauvegarde automatique régulière de la base.

11. Diagrammes (UML, BDD, Architecture)

Les diagrammes fournis incluent :

- Diagramme de cas d'utilisation
- Diagramme de classes simplifié
- Schéma de base de données relationnelle
- Schéma d'architecture logicielle (MVC)

12. Planification du projet

Phase 1 – Analyse des besoins
Phase 2 – Conception (UI, Modèle de données)
Phase 3 – Développement
Phase 4 – Tests
Phase 5 – Déploiement et présentation

13. Tests et validation

Chaque fonctionnalité est testée de manière unitaire.

Les cas de tests incluent :

- Ajout de dépense correcte / incorrecte
- Affichage du solde
- Navigation dans l'interface
- Génération de graphiques cohérents

14. Limites et évolutions possibles

Limites :

- Application uniquement locale
- Authentification basique

Évolutions :

- Synchronisation cloud
- Application mobile liée
- Conseils budgétaires intelligents via IA

15. Annexes

- Captures d'écran de l'application
- Lien vers le dépôt GitHub : <https://github.com/marlina24/EcoPlan-GestionDeBudget>
- Diagrammes UML
- Maquettes UI