ECO-WATT

**Présentation du projet :**

Le projet *EcoWatt* est un système intelligent de gestion de l’énergie conçu pour surveiller, analyser et optimiser la consommation électrique dans les habitations et les entreprises. Son objectif principal est de permettre aux utilisateurs de contrôler efficacement leurs appareils électriques, réduire les gaspillages d’énergie, et ainsi diminuer leur facture tout en contribuant à la préservation de l’environnement.  
Grâce à une interface conviviale, *EcoWatt* permet un suivi en temps réel de la consommation énergétique, l’envoi d’alertes en cas de dépassement des seuils prédéfinis, et la génération de recommandations personnalisées basées sur l’analyse des habitudes de consommation. Le système intègre un module de **programmation automatique** des appareils selon des horaires ou des événements détectés via des **capteurs intelligents (IoT)**.  
En plus de la gestion énergétique, *EcoWatt* propose une fonctionnalité complète de **facturation automatique**, avec génération de factures mensuelles, visualisation du montant dû, et suivi du statut de paiement. Un **système de contrôle bancaire sécurisé** est intégré pour permettre aux utilisateurs d’effectuer leurs paiements en ligne via différents moyens (carte, virement, etc.).  
Pour une meilleure expérience utilisateur, un **dashboard statistique dynamique** est également disponible, affichant des graphiques et rapports sur la consommation, les coûts, et les économies réalisées. Les données sont traitées par un module d’**analyse intelligente** capable de détecter les anomalies, de proposer des conseils d’optimisation, et de générer des comparaisons mensuelles ou saisonnières.  
Adaptable aussi bien aux besoins domestiques qu’aux environnements professionnels, *EcoWatt* ambitionne de devenir une solution fiable, sécurisée, évolutive, et éco-responsable pour une gestion énergétique moderne.

**Objectifs du projet EcoWatt**

**Objectif général :**  
Mettre en place une solution logicielle intelligente capable de gérer, optimiser et suivre la consommation énergétique dans des logements et des entreprises, tout en intégrant des fonctions d’automatisation, de facturation et de paiement sécurisé.

**Objectifs spécifiques :**

1. Permettre aux utilisateurs de surveiller en temps réel leur consommation électrique.
2. Réduire les dépenses énergétiques grâce à des recommandations et des automatisations intelligentes.
3. Gérer à distance les appareils électriques via une interface intuitive.
4. Générer automatiquement des factures mensuelles de consommation.
5. Offrir un système sécurisé de paiement en ligne des factures.
6. Afficher des statistiques et des tableaux de bord détaillés sur l’évolution de la consommation.
7. Détecter les anomalies énergétiques et alerter l’utilisateur en cas de problème.
8. Être compatible avec des capteurs IoT pour une gestion plus fine et automatisée.

**Besoins fonctionnels**

Voici les principaux besoins fonctionnels que tu peux inclure :

1. **Gestion des utilisateurs**
   * Création de compte
   * Authentification (connexion/déconnexion)
   * Modification de profil
2. **Gestion des lieux (Maison/Entreprise)**
   * Ajouter/modifier/supprimer un lieu
   * Lier des appareils et capteurs à un lieu
3. **Gestion des appareils**
   * Ajouter/modifier/supprimer un appareil
   * Allumer/éteindre un appareil à distance
   * Programmer le fonctionnement automatique
4. **Suivi de la consommation**
   * Visualisation en temps réel
   * Historique par jour/semaine/mois
   * Calcul du total consommé (en kWh et en Dinar)
5. **Alertes**
   * Notification en cas de dépassement de seuil
   * Notification d’appareil en veille ou consommation anormale
6. **Recommandations**
   * Génération automatique de conseils pour économiser
   * Affichage personnalisé selon le comportement de l’utilisateur
7. **Facturation**
   * Calcul automatique du montant dû par période
   * Génération d’une facture téléchargeable
   * Suivi du statut : payée / en attente
8. **Paiement**
   * Intégration d’un module de paiement sécurisé
   * Support des cartes bancaires et virements
   * Vérification de transaction et gestion des erreurs
9. **Statistiques & Dashboard**
   * Graphiques de consommation et comparaison mensuelle
   * Suivi des économies réalisées
   * Visualisation de la consommation par appareil

**Cas d'utilisation principaux**

| **Acteur** | **Cas d'utilisation** |
| --- | --- |
| Utilisateur | Créer un compte / se connecter |
| Utilisateur | Ajouter un lieu (maison ou entreprise) |
| Utilisateur | Gérer ses appareils (ajouter, programmer, etc.) |
| Utilisateur | Consulter sa consommation |
| Utilisateur | Télécharger ses factures |
| Utilisateur | Payer sa facture en ligne |
| Utilisateur | Recevoir une alerte |
| Utilisateur | Lire les recommandations d'économie |
| Utilisateur | Visualiser les statistiques sur le dashboard |
| Administrateur | Gérer les utilisateurs |
| Administrateur | Superviser les données globales |

**Les classes :**

**1. Classe Utilisateur**

* **Attributs :** id, nom, prénom, email, motDePasse, type (particulier / entreprise)
* **Méthodes :** seConnecter(), seDéconnecter(), modifierProfil()

**2. Classe Administrateur *(hérite de Utilisateur)***

* **Méthodes :** gérerUtilisateurs(), superviserStatistiquesGlobales()

**3. Classe Lieu *(classe mère pour Maison / Entreprise)***

* **Attributs :** idLieu, nom, adresse, surface
* **Méthodes :** ajouterAppareil(), supprimerAppareil(), afficherConsommation()

**4. Classe Maison / Entreprise *(hérite de Lieu)***

**5. Classe Appareil**

* **Attributs :** idAppareil, nom, type, statut (ON/OFF), consommationHoraire
* **Méthodes :** allumer(), eteindre(), programmerFonctionnement()

**6. Classe Capteur**

* **Attributs :** idCapteur, type (température, mouvement…), emplacement, valeurActuelle
* **Méthodes :** lireValeur(), envoyerDonnee()

**7. Classe ConsommationEnergie**

* **Attributs :** idConso, date, heure, kWh, idAppareil
* **Méthodes :** calculerConsommation(), genererRapport()

**8. Classe ProgrammeAutomatique**

* **Attributs :** idProgramme, heureDebut, heureFin, jours, appareilCiblé
* **Méthodes :** activer(), désactiver()

**9. Classe Alerte**

* **Attributs :** idAlerte, message, type, dateHeure
* **Méthodes :** envoyer(), afficher()

**10. Classe Recommandation**

* **Attributs :** idReco, contenu, dateCréation
* **Méthodes :** generer(), afficher()

**11. Classe Historique**

* **Attributs :** idHistorique, date, consommations[], utilisateur
* **Méthodes :** afficher(), exporterPDF()

**12. Classe Facture**

* **Attributs :** idFacture, montantTotal, période, statut, dateEmission, datePaiement
* **Méthodes :** generer(), afficher(), marquerCommePayée()

**13. Classe SystemePaiement**

* **Attributs :** idPaiement, moyenPaiement, statutPaiement, dateTransaction
* **Méthodes :** effectuerPaiement(), verifierPaiement(), annulerPaiement()

**14. Classe Dashboard**

* **Attributs :** idDashboard, dateMiseAJour, utilisateur, statistiques[]
* **Méthodes :** afficher(), rafraîchir(), exporterGraphique()

**15. Classe Statistique**

* **Attributs :** idStat, type, valeur, dateCalcul
* **Méthodes :** calculer(), afficher(), comparer()

**16. Classe AnalyseurEnergie**

* **Attributs :** idAnalyseur, méthodeAnalyse
* **Méthodes :** analyserConso(), générerConseils(), détecterAnomalies()
* Utilisateur ↔ Lieu (1-n)
* Lieu ↔ Appareil, Capteur, Facture (1-n)
* Appareil ↔ ConsommationEnergie, ProgrammeAutomatique (1-n)
* Facture ↔ SystemePaiement (1-1)
* Utilisateur ↔ Dashboard, Historique, Recommandation, Alerte (1-n)
* Dashboard ↔ Statistique (1-n)
* AnalyseurEnergie est lié à plusieurs classes pour traiter les données