

FOUNDATION PROJECT

Alpha Amadou DIALLO

dialloalphaamadou947@gmail.com <https://github.com/alpha947>

OVERVIEW

1. Project background and description

En Guinée, les coupures de courant sont fréquentes et peuvent entraîner des dommages matériels importants, des incendies, ou des interruptions dans des systèmes critiques. Ce projet vise à développer une application intelligente capable de détecter les coupures de courant en temps réel et d'alerter les utilisateurs par des moyens divers (alarme, SMS, email). L'objectif est de réduire les risques liés aux coupures électriques tout en fournissant un outil de surveillance accessible et efficace.

2. Project scope

Inclus dans le périmètre :

- Détection des coupures de courant.
- Développement d'une application web et mobile pour visualiser les alertes et gérer les notifications.
- Notifications multicanales : alarme sonore, SMS, emails, et notifications push.
- Stockage et analyse des historiques de coupures.

Non inclus (pour le moment) :

- Réparation automatique des pannes électriques.
- Support matériel avancé (ex. batteries de secours).

3. High-level requirements

System Detection:

- Pour un début, utiliser les OS ou sont déployer le backend pour détecter la coupure du courant si le capteur IOT ne m'est pas offert par sourcemind pour le moment.
- Transmission des données en temps réel au backend (Apache Kafka).

Notifications:

- Alarme sonore locale en cas de coupure.
- Envoi de SMS/Appel via une API (Twilio).
- Envoi d'emails avec un service SMTP.
- Notifications push via Firebase Cloud Messaging (FCM).

Interfaces Utilisateurs:

- Application web avec React pour configurer et visualiser les alertes.
- Application mobile avec React Native pour les notifications en temps réel.

Base de Données:

- Stockage des coupures et historiques d'événements (PostgreSQL)

Rapports et Analyse:

4. Deliverables

- **Application Web** : Un tableau de bord fonctionnel pour gérer les alertes et consulter les rapports.
- **Application Mobile** : Notifications en temps réel.
- **Backend API** : Développé avec Spring Boot.
- **Documentation** : Guide technique et fonctionnel pour l'utilisation et la maintenance.
- **Rapports d'Analyse** : Exportation des données au format PDF ou Excel.

5. Affected parties

- **Utilisateurs Finaux** : Les résidents guinéens affectés par les coupures électriques.
- **Entreprises** : Surtout celles qui dépendent d'une alimentation électrique stable.

6. Affected business processes or systems

- Systèmes électriques locaux : Ajout de capteurs IoT dans les installations électriques.
- Processus d'alerte : Introduction d'un système de notification proactive.
- Gestion des données : Collecte et analyse des événements liés aux coupures.

7. Specific exclusions from scope

- Réparation des infrastructures électriques endommagées.
- Surveillance avancée des réseaux haute tension.
- Gestion des factures ou coûts énergétiques.

8. Implementation plan

Phase 1: Recherche et Design

- Identifier les capteurs et matériels nécessaires.
- Définir l'architecture technique et les outils logiciels.

Phase 2: Développement Backend

- Créer une API avec Spring Boot pour détecter les coupures (Java) et les notifications.

Phase 3: Développement Frontend

- Développer l'application web avec React.
- Développer l'application mobile avec React Native.

Phase 4: Tests

- Vérification des notifications en temps réel.
- Validation de la précision des capteurs.

Phase 5: Déploiement

- Hébergement du backend sur Teamcity (Docker).

9. Perspective

- Pour une meilleure détection, je compte intégrer des capteurs IoT au backend via un microcontrôleur (Arduino, ESP8266). Pour surveiller l'alimentation électrique de tout un parc par exemple.
- Publication de l'application mobile sur le Play Store et l'App Store.
- Entraîner une IA sur la base de données de l'application pour pouvoir prédire une coupure de courant avant que celle-ci ne se produise.
- Génération de rapports périodiques (hebdomadaires/mensuels).
- Outil d'analyse pour identifier les zones les plus affectées.
- Gestion des utilisateurs (authentification, personnalisation des alertes).
- Intégrer un service **d'Intervention** : Organisations responsables de la gestion des infrastructures électriques.

10. High-level timeline/schedule

Phase	Durée Estimée
Recherche, Design et inspiration	2 jours
Développement Backend	12 jours
Développement Frontend Web	4 jours
Développement Mobile	4 jours
Intégration IoT	Pas pour le moment
Tests	2 jours
Déploiement et Documentation	1 jour
Total	25 jours

APPROVAL AND AUTHORITY TO PROCEED

The different approvers of the project

Name	Title	Date	Remarques
APPR 1			
APPR 2			
APPR 3			
APPR 4			

Approved By	Date	Approved By	Date
-------------	------	-------------	------