



PROJET DE FIN D'ANNÉE (PFA)

Conception et Développement d'une Plateforme Géospatiale Intégrée

(Web, Mobile & QGIS Server)

*Suivi-Évaluation du Projet de Désenclavement
des Zones de Production Piscis-rizicoles (PPR) en Guinée*

Rapport technique des technologies choisies

Préparé par :

MOUTAMANNI ABDOURRAHMANE

DAHDOUH AYOUB

Encadré par :

El IMAMI AYOUB

Stage effectué au sein de l'entreprise :

ETAFAT

13 juillet 2025

Table des matières

Introduction	2
Contexte	2
Objectif	2
1 Justification des choix technologiques	3
1.1 React Native (application mobile Android)	3
1.2 GeoDjango (backend web)	3
1.3 Leaflet (cartographie web et mobile)	3
1.4 Chart.js (visualisation web)	3
2 Connexion d'un GPS externe via Bluetooth sur Android	4
Conclusion	5
Ressources et documentations utilisées	5

Introduction

Dans le cadre de notre Projet de Fin d'Année (PFA), nous avons participé au développement d'une plateforme géospatiale intégrée pour le suivi-évaluation du Projet de Désenclavement des Zones de Production Piscic-rizicoles (PPR) en Guinée. Ce projet repose sur deux interfaces complémentaires : une application mobile pour la collecte des données terrain et une application web pour la gestion, l'analyse et la visualisation des données collectées.

Ce rapport technique présente les choix technologiques retenus pour chaque composante de la solution, ainsi que leur intégration dans un environnement SIG cohérent et performant.

Contexte

Le suivi d'un projet d'infrastructure en milieu rural, tel que le PPR, nécessite une collecte rigoureuse des données géolocalisées et leur exploitation efficace à des fins de prise de décision. Dans ce cadre, une plateforme numérique intégrée permet :

- La collecte de données en temps réel sur le terrain ;
- L'affichage cartographique des travaux réalisés ;
- La génération de statistiques et graphiques pour l'évaluation ;
- La centralisation sécurisée des données sur un serveur géospatial.

Objectif

L'objectif de ce rapport est de présenter les technologies choisies pour le développement de la plateforme géospatiale intégrée, afin d'assurer la collecte, la gestion, la visualisation et l'analyse des données liées au projet.

1 Justification des choix technologiques

Dans cette section, chaque technologie est présentée, avec son rôle concret dans notre application Android, ainsi que les raisons spécifiques qui motivent son choix.

1.1 React Native (application mobile Android)

Qu'est-ce que c'est ? Un framework open-source de Meta permettant de développer une application mobile Android (et iOS) avec un unique code JavaScript.

Rôle dans notre projet Base de notre application Android, React Native permet l'interaction avec le GPS externe via Bluetooth, l'affichage cartographique et le stockage local en situation hors connexion.

Pourquoi ce choix ?

- Permet de maintenir une seule base de code Android/iOS, simplifiant le développement et la maintenance.
- Offre un accès direct à des fonctionnalités natives (BLE, stockage local) adaptées à notre besoin de connexion au GPS externe.
- Dispose d'un riche écosystème de modules éprouvés pour BLE, gage de fiabilité.

1.2 GeoDjango (backend web)

Qu'est-ce que c'est ? Extension du framework Django qui apporte des capacités SIG (stockage, requêtes et serveurs de données géospatiales) via PostGIS.

Rôle dans notre projet Back-end central, GeoDjango gère les données GPS collectées, les zones géographiques et expose une API géospatiale pour les frontends mobile et web.

Pourquoi ce choix ?

- Support natif de PostGIS pour requêtes spatiales (intersections, proximités...), essentielles à notre suivi GPS.
- Intégration facilitée avec Django, garantissant sécurité, modularité et évolutivité.

1.3 Leaflet (cartographie web et mobile)

Qu'est-ce que c'est ? Bibliothèque JavaScript légère pour cartes interactives, environ 42Ko gzippé.

Rôle dans notre projet Affiche la géolocalisation terrain sur mobile et web : position GPS, zones de projet, marqueurs et tracés, avec cache offline.

Pourquoi ce choix ?

- Léger et performant sur Android, avec plugins utiles (dessin de polygones, caches offline).
- Compatible avec GeoDjango et PostGIS : les données géospatiales sont directement affichables.

1.4 Chart.js (visualisation web)

Qu'est-ce que c'est ? Bibliothèque JavaScript pour graphiques interactifs sur canevas HTML5.

Rôle dans notre projet Présente des tableaux de bord web (nombre de relevés, progression terrain) de manière claire et dynamique.

Pourquoi ce choix ?

- API simple et flexible, idéale pour des graphiques basiques à adapter selon nos besoins.
- Rendu responsive, compatible avec différents écrans (tablette, PC).

2 Connexion d'un GPS externe via Bluetooth sur Android

Pour assurer une géolocalisation précise sur notre application mobile Android, nous utilisons un GPS externe connecté via Bluetooth. Android ne permet pas directement de récupérer les données GPS des appareils Bluetooth externes. La solution consiste donc à utiliser une application tierce d'injection NMEA, qui lit les trames GPS envoyées par l'appareil et les injecte comme source de localisation dans le système Android.

Voici la méthode simple :

1. **Connexion Bluetooth au GPS externe :** Allumer le GPS externe (exemples : Garmin GLO, Bad Elf, Emlid, etc.). Aller dans *Paramètres > Bluetooth* sur le téléphone Android. Associer et connecter l'appareil GPS.
2. **Installation d'une application d'injection NMEA :** Android ne lit pas directement les données du GPS externe. Il faut donc une application spéciale qui capte les trames NMEA transmises par le GPS, puis les injecte comme source de localisation système. L'application gratuite recommandée est *Bluetooth GPS* (disponible sur le Play Store).
3. **Activation des options développeur :** Aller dans *Paramètres > À propos du téléphone > Numéro de build* et taper 7 fois pour activer le mode développeur. Ensuite, dans *Paramètres > Système > Options pour les développeurs*, activer *App de localisation fictive* et sélectionner *Bluetooth GPS* dans la liste.
4. **Utilisation de l'application Bluetooth GPS :** Ouvrir l'application, accéder aux paramètres, vérifier que le format des données est *NMEA*. Activer l'option *Mock Provider* ou *Inject mock location*. Sélectionner l'appareil GPS externe connecté. Cliquer sur *Connect*, puis sur *Start*.
5. **Utilisation des données dans l'application React Native :** Une fois connecté, Android reçoit les coordonnées du GPS externe comme source officielle de localisation. L'application mobile utilisant des API comme `react-native-geolocation-service` ou `navigator.geolocation` récupérera alors ces données précises.

Cette méthode garantit ainsi la précision de la localisation terrain, essentielle pour le suivi géospatial et l'affichage des positions sur Leaflet.

Conclusion

La conception et le développement de notre plateforme géospatiale intégrée reposent sur des choix technologiques mûrement réfléchis, alliant robustesse, performance et compatibilité avec les contraintes terrain du projet PPR. React Native permet une application mobile multi-plateforme efficace, tandis que GeoDjango et PostGIS assurent un backend spatial puissant et scalable. L'utilisation de Leaflet et Chart.js garantit une visualisation claire et interactive des données géospatiales et statistiques.

La méthode d'injection des données GPS via une application tierce sur Android, combinée à un GPS externe connecté en Bluetooth, offre une précision accrue indispensable au suivi terrain. Cette intégration technique renforce la fiabilité des données collectées et leur exploitation.

En somme, cette architecture technologique cohérente facilite la gestion complète du projet, de la collecte terrain à l'analyse décisionnelle, tout en ouvrant des perspectives d'extension et d'amélioration pour les futurs projets géospatiaux.

Ressources et documentations utilisées

- <https://reactnative.dev/docs/getting-started>
- <https://docs.djangoproject.com/en/stable/ref/contrib/gis/>
- <https://leafletjs.com/reference.html>
- <https://www.chartjs.org/docs/latest/>
- <https://www.youtube.com/watch?v=gq0LB3QQaJ4&t=141s>