



4IF - H4212
INSA de Lyon

Rapport
PLD SMART
BellePouBelle

Gestion collective des bacs à ordures.

Auteurs :

- Amina DEBAB
- Hayfa GARBOUT
- Imene HAKEM
- Jürgen HECHT
- Fatima-Ezzahra MEZIDI
- Ndèye Gagnessiry NDIAYE
- Jiaye PU

Introduction:	3
Fonctionnalités :	3
Coté Serveur:	3
Coté utilisateur:	3
Coté Mairie:	4
Coté Poubelle:	4
Les axes d'amélioration:	4

Introduction

Notre projet consiste en une solution connectée, participative et complète pour la gestion des poubelles dans le but d'avoir une ville plus belle et propre.

Nous proposons une solution complète comprenant une application utilisateur, un site pour les manageurs de la ville et des poubelles connectés.

L'application utilisateurs(android/ios/webpage) permet de géolocaliser les poubelles tout en connaissant leur type (plastique, verre, autre) et leur taux de remplissage. Leur participation est primordiales à la fiabilité des données.

L'application pour les agents municipaux leur offre une vision globale des poubelles dans la ville. Elle leur donne des données statistiques qui leur permettront d'améliorer leur cycle de ramassage de poubelles.

Les poubelles connectés donneront de données fiables et rapides.

L'impact écologique de notre solution est non négligeable. Elle permet de faciliter le jettage des poubelles et diminuer les circuits des éboueurs inutiles.

Fonctionnalités

Coté Serveur:

- Machine virtuelle (Intel vCore (2x virtuel), DDR3 6 Go, SATA 120 Go).
- Accessible via une adresse IP publique et le domaine bellepoubelle.fr.
- Arch Linux avec nginx, WildFly, PostgreSQL, Certbot, Snort, OpenSSH.
- Le backend est écrit en Java EE 8 avec JAX-RS et Hibernate.
- Disponibilité de l'API JSON RESTful et microservice.
- Stockage persistant des données dans une base de données relationnelle.
- TLS 1.2 confidentialité persistante avec ChaCha20 / AES cryptage 256 bits.
- Certificat TLS autorenewable de Let's Encrypt.
- SHA512 hachage et salage par mot de passe.
- Visualisation de la documentation de l'API, de l'application Web progressive et de l'application de gestion.
- Journalisation du serveur avec Apache Log4j 2 et systemd-journal.
- Authentification de base et contrôle d'accès basé sur les rôles pour les demandes d'API.
- E-mail via SMTP avec la procédure STARTTLS.
- Limitation de débit API et détection d'intrusion réseau.
- Connexion à Firebase Cloud Messaging pour une notification push vers les appareils mobiles.

Coté utilisateur:

- Géolocalisation des bacs à tri avec distinction du type de la poubelle selon la couleur
- Itinéraire de la position vers la poubelle sélectionnée par l'utilisateur
- Envoyer une alerte en cas de problème (ajout d'un message , sélection d'une image, scanner Qr Code pour préciser la poubelle).
- Consulter toutes les alertes.
- Application multilingue.

Coté Mairie:

Toutes les données sont extraites du serveur à jour en temps réel.

- Visualiser les statistiques en temps réel, avec des indicateurs (pourcentage de poubelles trop remplies, alertes dans les 24 dernières heures, nombre d'alertes fermées)
- Visualiser les données envoyées par les capteurs des poubelles
- Visualiser le taux de remplissage par type de poubelles
- Pourcentage d'alerte par type d'alerte (pleines, remplies, cassées)
- Evolution des alertes en cours de temps semaine 7 jours
- Visualiser et traiter les alertes qui ont été émises
- Visualiser les tops 20 poubelles les plus remplies avec leur adresses et leur types (pleines, remplies, cassées)
- Localiser poubelles avec leur taux de remplissage et type dans le map, visualiser leur adresses en cliquant

Coté Poubelle:

Les capteurs sont de la gamme LoRaWAN - Technologie radio longue portée (LoRa) pour la communication avec le serveur a longue distance.

- Capteur de remplissage.
- Capteur de géolocalisation.
- Chaque capteur a un token unique pour communiquer avec le serveur de manière sécurisée
- Chaque poubelle a un QRcode unique qui permet de l'identifier et permet aux utilisateurs d'envoyer efficacement des alertes ciblées
- Les capteurs n'envoient aux serveurs des données que si elles sont différentes de la dernière capture.

Axes d'amélioration

Il serait intéressant d'afficher l'historique de la moyenne de remplissage. L'implémentation back-end à déjà été réalisées, mais aucun graphique ne l'affiche au niveau front-end, et le mécanisme d'enregistrement toutes les X heures n'a pas été lancé.

On pourrait utiliser ces données pour ensuite établir des prévisions.

Les données de Lyon pourront être plus remplies, et contenir plus de type de poubelles. Nous avons travaillé avec les données qui décrivent les sillos de verre et les avons généralisés aux poubelles de verre et des déchets ménagers. On pourrait rajouter à la carte un clustering par type de poubelles et localisation (l'api google ne donne pas ce type de clustering -seulement par localisation). On pourra aussi rajouter plus d'interactions avec les alertes (tout supprimé, tout validé..).

Gestion du projet:

Notre gestion du projet partent sur le principe de fixer des objectifs à court termes et de se lancer sur la route sans tarder ,Une fois ce premier objectif atteint, on marque une courte pause et on adapte son itinéraire en fonction de la situation du moment. Et ainsi de suite jusqu'à atteindre la destination finale, Notre gestion de projet donc était avec une approche agile.

L'équipe sélectionne ensuite une portion des exigences à réaliser dans une portion de temps courte appelée sprint. Les sprints au début du projet étaient d'une durée de 2 jours mais avec l'avancement du projet ça revient vers des sprints d'un jour.

L'organisation de sprint et la liste de ses backlogs était assuré par le scrum master en utilisant l'outil **Trello** pour les tâches générique et **ToDo-list** de la plateforme **Slack** our les détails.

La vie du notre projet Scrum est rythmée par un ensemble de réunions "stand up meeting du 5 à 10 mins" clairement définies et strictement limitées dans le temps (timeboxing):

- **Planification du Sprint** : au cours de cette réunion, l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog » qu'elle pense pouvoir réaliser au cours du sprint.
- **Revue de Sprint** : à la fin du sprint, l'équipe présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du Product Owner et des utilisateurs finaux .