

RAPPORT PROJET ANDROID

Par MOUHIDINE Noussayr et LABASSI Elyès



Conception et développement d'application mobiles

Présentation de l'application BornElec :

BornElec est une application de navigation pour véhicules électriques. Elle permet de trouver une station de recharge pour véhicule partout en France. Son objectif final est de permettre à l'utilisateur de repérer une station qui offre la possibilité de recharger son véhicule tout électrique ou semi-électrique.

Pour cela, nous avons structuré notre développement en deux grands axes :

- La recherche de station
- Les actions liées au repérage.

Avec la conception actuelle, nous sommes arrivés à lister toutes les stations au moyen d'une Liste View. À terme l'application sera capable d'utiliser les ressources système Android pour localiser directement l'endroit où se trouve l'utilisateur et permettre de recenser toutes stations dans un rayon de 2 à 5 km.

Une fois la station trouvée, il est possible de l'ajouter à la liste des favoris ou encore de sauvegarder la position de la station afin d'obtenir l'itinéraire le plus proche pour y accéder ultérieurement.

L'application pourrait également faire l'objet d'un module « réseaux sociaux » où la communauté des utilisateurs de BornElec pourrait échanger sur les stations trouvées, de même : conseiller et noter chaque station. Nous sommes persuadés qu'en ajoutant un aspect communautaire, l'application deviendrait le facteur d'amélioration des stations de recharge électrique et permettre d'innover dans le confort du conducteur.

Le type de véhicules concerné sont bien entendu les tout-électriques, les voiture semi-électriques (hybride + électrique) ou encore de nouveaux véhicules utilisant l'énergie concernée. Sans oublier bien sûr les véhicules « autoLib » de la ville de Paris.

Au niveau de la couverture du recensement des stations, nous avons utilisé les données issues du site du gouvernement. Ce fichier est au format CSV (Comma-separated values). Ce format est un format représentant sous forme tabulaire les données séparées par des virgules. Cette donnée source, est normée RFC 4180 et utilise le type mime « text/csv ». À la différence des fichiers binaire, nous utilisons un fichier de type texte.

Chaque description de stations contient :

- Un identifiant permettant de rendre unique et estampiller chaque station.
- Un nom de station
- Une adresse
- Les coordonnées géographiques représentées par
 - Une latitude
 - Une longitude
- Des données techniques permettant de différencier les services offerts par chaque des stations :
 - Un type de recharge
 - Un nombre de pdc
 - Le type de connecteur

- Puis, des informations complémentaires comme :
 - La date de dernière mise à jour de la mention de la station
 - Une observation et la source du fichier CSV

Cette donnée source nommée bornes.csv, se présente comme étant notre « dataset » ou « jeu de données ».

Attention un jeu de données comme celui-ci, ne nous dispense d'y ajouter une base de données. Ce dernier fichier est sous forme tabulaire où chaque ligne correspond à une observation et chaque colonne à une variable.

Au niveau de la structure de ce jeu de données, nous sommes en mesure de localiser, identifier et sélectionner la donnée.

Pour sélectionner la donnée, nous avons utilisé la technique de « parsing » (anglicisme).

Entre chaque virgule nous avons le contenu de la donnée, suivi d'un élément « fin de ligne » à la fin de chaque élément-station.

Notre classe Java CSVFile, permet de récupérer le contenu de ces données.

Les technologies utilisées sont Gradle, l'Async Task, les Threads et GitHub. Nous nous sommes aussi inspirés du web et plus particulièrement à la création d'une application géolocalisante et émettrice de données météo.

Focalisons-nous sur la technologie Gradle :

Ce cadriciel permet de faire une multitude de tâches allant du déploiement d'applications au téléchargement automatique de bibliothèques Java.

L'utilisation que nous en faisons dans le projet est de permettre la synchronisation des changements effectués lors du développement. De surcroît, ce Framework nous dispense de télécharger les fichiers jar associés au projet. Ici, nous n'en avons pas besoin car aucune bibliothèque n'est utilisée. Mais si tel est le cas, nous serons en mesure d'utiliser les méthodes prédéfinis de la bibliothèque à utiliser.

Gradle fonctionne donc de manière automatique et nous permet de gagner du temps dans la gestion de projet.

Gradle reconnaît la bibliothèque grâce à trois éléments :

- Le groupe ID
- Le nom artéfact
- Le numéro de version

Le groupe ID correspond au package de référence.

Le nom artéfact détermine le nom de l'application globale, il est unique et permet notamment d'être référencé dans le Play store.

Le numéro de version quant à lui permet de renseigner sur la maintenabilité de l'application.

Développement :

L'application BornElec est une application permettant de lister toutes les stations de recharge pour véhicules électriques. Les véhicules électriques sont à la fois les véhicules autolib de Paris mais aussi les véhicules des particuliers.

Sur le marché existe des applications permettant de répertorier toutes les stations actuelles. L'originalité de BornElec est de permettre une expérience utilisateur améliorée avec la possibilité d'interagir avec les membres de la communauté utilisant BornElec.

Cependant, pour le développement actuel nous nous sommes focalisé sur la notion d'Async Task et des Thread. Dans une perspective proche, nous pourrions utiliser les notions de Services.

À moyen terme cette application pourra, avec un simple touché sur l'un des éléments de la liste des stations, donner les coordonnées de la station et permettre à l'utilisateur de la localiser. C'est donc à terme, cette application est capable de localiser une borne grâce au système de localisation Android. Mais pour l'heure, cette application est capable de lister les stations dans une liste View. Elle se veut adaptée à la source ou dataset au format CSV.

Nous verrons dans le déroulé de ce rapport comment gérer les Threads, comment fonctionne la lecture du fichier source et comment nous laissons cette application ouverte à de nouvelles perspectives. En effet, le but est qu'au final il existe une expérience utilisateur optimale pour trouver n'importe où en Europe une borne de recharge pour les véhicules tout électrique.

Déroulé du fonctionnement de l'application :

L'application fonctionne grâce à quatre classes Java : BornesAffiche, CSVFile, ItemArrayAdapter et MainActivity essentiellement. 3 layout ont été définis : activity_main, affiche_bornes et item_layout.

La classe CSVFile est ce qui va permettre de récupérer les données du fichier bornes.csv. Notamment via la méthode read() définie dans la classe. Cette méthode qui va récupérer les données du CSV et qui va les stocker dans un objet de type List.

La classe ItemArrayAdapter quant à elle va permettre d'adapter les informations récupérées par la classe CSVFile. En effet, elle va récupérer l'objet List et la mettre en place sur la vue de l'application en l'occurrence la ListView.

La classe BornesAffiche est en fait une activité. Autrement dit un des composants principaux au fonctionnement de l'application représentant l'implémentation et les interactions des interfaces. Via les classes CSVFile et ItemArrayAdapter, les noms des stations récupérées dans le csv sont affichées en liste sur la vue de l'application. Celle-ci servira d'activité secondaire.

L'activité principale est la classe MainActivity. Elle y implémente une interface avec un TextView et un bouton simple. Sur ce bouton y est placé un écouteur (Listener) qui une fois cliquer lance l'AsyncTask ChargementBornes qui affiche une progressBar et ensuite lance l'activité secondaire BornesAffiche.

Manuel utilisateur

Page d'accueil, message de bienvenue et bouton 'afficher les stations'. Pour voir le nom de tous les stations pour véhicules électriques en France il suffit de cliquer sur le bouton (Cf. **Annexe**).



Cahier technique

1. Analyse

Dans le cadre de l'évaluation des compétences acquises en cours de conception nous devons développer une application qui regrouperais l'ensemble des technologies vue en cours.

2. Développement

Nous avons énuméré l'ensemble des technologies vues en cours et regrouper celles qui nous semblaient les plus importantes :

- Activité
- Intention
- Composant Graphique : ListView, TextView, Button, ProgressBar
- AsyncTask

3. Répartition des tâches

Nous avons tout d'abord établi nos logiciels de travail :

- AndroidStudio
- Github

Nous sommes partis sur du développement chacun de son côté. Nous avons rencontré des difficultés à se trouver des moments pour avancer le projet. Entre les vacances d'été et le travail cela fut compliqué n'ayant pas le même emploi du temps.

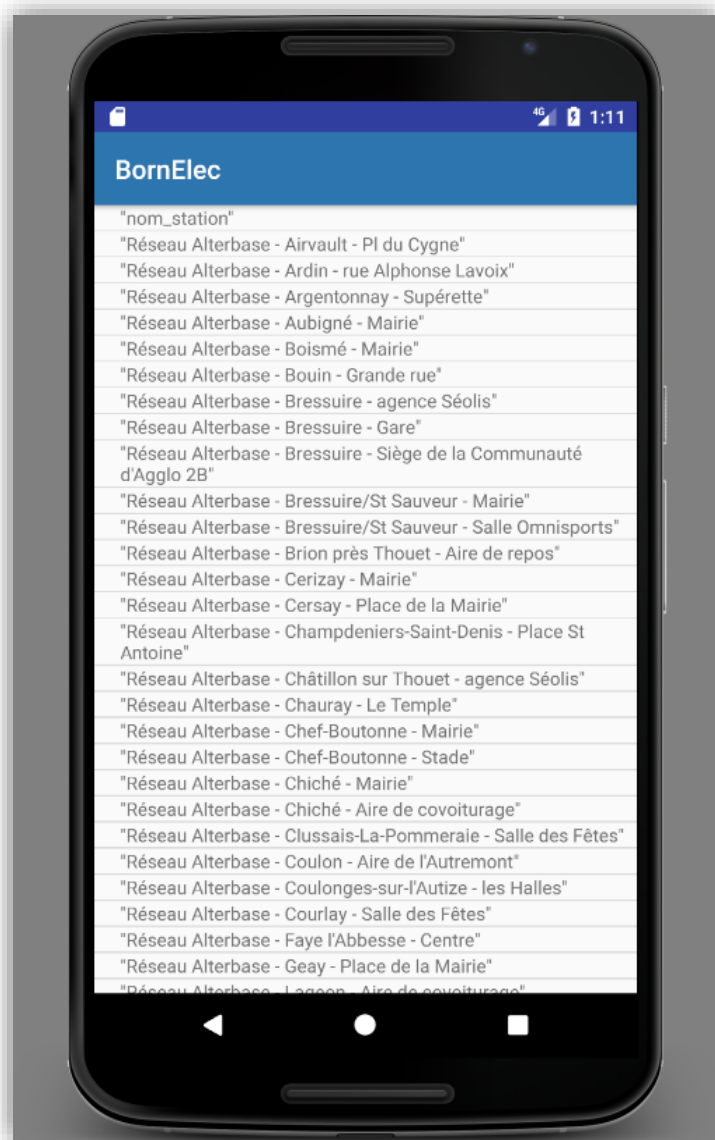
Néanmoins, Elyès avait commencé le développement de l'application en début juillet et en fin juillet Noussayr l'a récupéré et l'avait presque terminé.

La Javadoc était à la charge de Elyès.

Enfin, nous nous sommes mis d'accord pour rédiger le rapport à chacun son tour jusqu'à la fin du développement de l'application.

Le lien Github vers notre projet : <https://github.com/Elyos/BornElec>

Annexe



Affichage des noms des stations électriques