

**Sujet d'épreuves des Finales Nationales  
de la 46<sup>es</sup> WorldSkills Compétition**

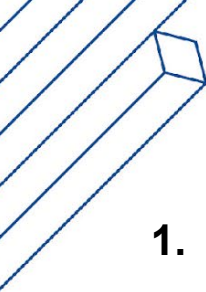
# **MÉTIER N°09 SOLUTIONS LOGICIELLES POUR L'ENTREPRISE**

## **PHASES 2 - LYON 2022 MODULE 4**

Soumis par :  
Xavier CHENEY, Expert WorldSkills France  
Laurent-Walter GOIX, expert adjoint

© Worldskills France (WSFR) se réserve tous les droits relatifs aux documents rédigés pour ou au nom de WSFR et comprenant leur traduction et leur envoi par voie électronique. Ce matériel peut être reproduit à des fins professionnelles et pédagogiques non commerciales, à la condition que le logo de Worldskills France et la déclaration concernant les droits d'auteur restent en place.





# 1. TABLE DES MATIERES

TABLE DES MATIERES	2
Module 4	3
Analyse du besoin	3
Travail à effectuer	Erreur ! Signet non défini.





## 2. MODULE 4

*Vous allez développer l'application mobile pour les agents des déchèteries : l'interface graphique mais aussi les fonctionnalités. Le temps passe vite : soyez rapide, efficace et respectez les consignes.*

### Analyse du besoin

Le portail web d'accès aux encombrants par les agents des déchèteries est beaucoup utilisé, mais il ne permet pas aux agents d'être réactifs sur le terrain.

Un nouveau besoin est donc apparu pour développer une application mobile à destination des agents, notamment pour déclarer qu'ils ont enlevé un encombrant en temps réel.

L'application comprend deux écrans distincts :

- 1 permettant de visualiser uniquement les encombrants en état "signalé",
- 1 pour déclarer l'enlèvement d'un encombrant (changement de statut à "enlevé").

*Dans cette première version prototypale, il n'y a pas de login agent.*

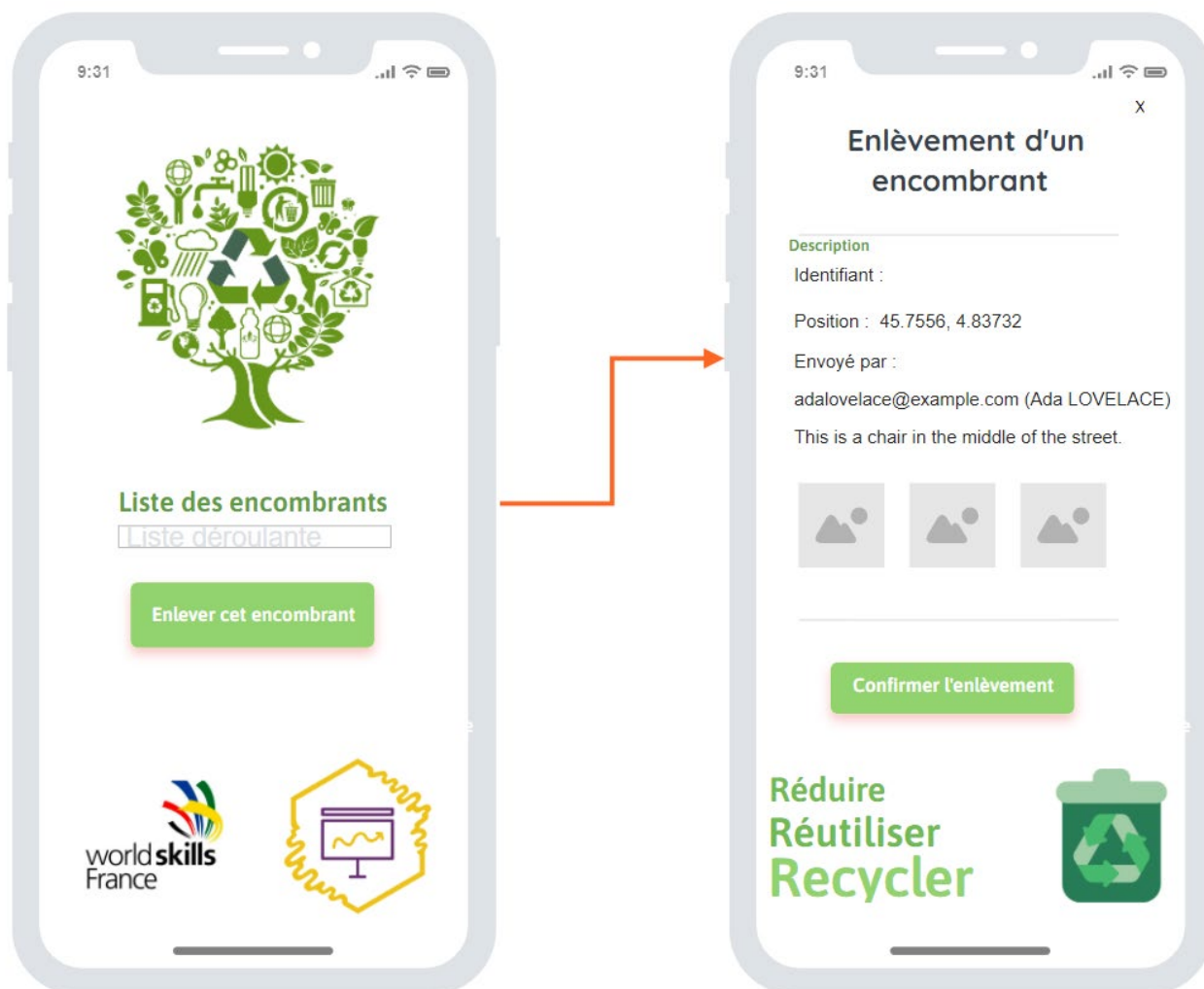
Pour simplifier l'usage de l'application, l'administration a demandé à ce qu'elle géo-localise l'agent automatiquement pour ne lui permettre de visualiser (et de valider l'enlèvement) que des encombrants situés dans un rayon de 50m autour de lui.

*Il faudra donc prévoir de demander la permission GPS à l'utilisateur.*

Une fois un encombrant "enlevé" (informé par message/popup de confirmation), il n'est plus affiché dans la liste.

*Le graphisme de l'application est libre.*

*L'exemple donné ci-dessous n'a aucune valeur : restez libre de proposer ce que vous souhaitez pour rendre l'application simple à utiliser - ATTENTION AU TEMPS QUE VOUS DEDIEZ A CETTE TACHE.*



L'application mobile communique avec la base de données et le serveur à l'aide de l'API objet du module 1.

*Pour simplifier le développement de l'interaction de l'application mobile avec l'API, le serveur et la base de données vous sont fournis. Vous ne devez pas utiliser celui que vous avez développé lors du module 1.*

*Vous devez livrer l'application sous forme d'APK préinstallé sur l'écran principal de votre tablette.*

#### Travail à effectuer

- Développer l'application mobile complète en respectant le Guide de style fourni
  - interface graphique
  - fonctionnalités demandées
  - interconnexion à l'API
- Installer l'application mobile sur l'écran principal de votre tablette

## Annexe - Usage de HTTP ClearText

Source: <https://www.journaldunet.fr/web-tech/developpement/1497383-android-comment-resoudre-l-exception-clear-text-http-traffic-to-not-permitted/>

Depuis la version 8 d'Android, l'accès de l'application que l'on développe a été modifié pour davantage de sécurité. La navigation en clair, c'est-à-dire sans aucune sécurité est par défaut interdite. Si vous sollicitez une URL non protégée, vous aurez alors au moment de l'exécution de votre application ce message "Cleartext HTTP traffic to \* not permitted". Il existe plusieurs solutions pour résoudre ce souci.

Si vous êtes dans le cas pour lequel il n'existe pas d'URL sécurisée, il existe différentes solutions qui permettent de contourner ce problème. On peut ajouter dans une application Android le fichier XML `network_security_config.xml`. On peut ajouter dans ce fichier des domaines pour lesquels on souhaite autoriser la navigation en clair. Pour effectuer cette manipulation, créez le fichier dans le dossier "res/xml" et ajoutez-y le code suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">monsiteinternet.fr</domain>
    </domain-config>
</network-security-config>
```

L'attribut "includeSubdomains" permet d'ajouter tous les sous-domaines d'un domaine, comme par exemple `monblog.monsiteinternet.fr`. Vous devez ensuite modifier le fichier `AndroidManifest.xml`, à la racine de votre application, pour indiquer la présence du fichier de configuration.


```
<?xml version="1.0" encoding="utf-8"?>
<manifest ...>
    <uses-permission android:name="android.permission.INTERNET" />
    <application... android:networkSecurityConfig="@xml/network_security_config" ...>
        ...
    </application>
</manifest>
```

L'avantage de cette option est que l'on n'autorise que les requêtes à destination des domaines que l'on choisit. Cela permet de garder un niveau de sécurité important, surtout si l'on est celui qui gère le ou les domaine(s) cible(s).

Une autre option consiste à modifier directement le fichier `AndroidManifest.xml`. On peut, lors de la déclaration de l'application, ajouter le paramètre "android:usesCleartextTraffic" avec comme valeur "true" pour indiquer que le trafic réseau vers les sites non sécurisés est autorisé.

```
<manifest ...>
    <uses-permission android:name="android.permission.INTERNET" />
    <application... android:usesCleartextTraffic="true" ...>
        ...
    </application>
</manifest>
```





Vous devez faire attention avec cette modification, car il n'y a plus de contrôle de sécurité, quel que soit le site indiqué, le trafic sera autorisé.

Lorsque vous modifiez le fichier AndroidManifest.xml, faites également attention à la présence du paramètre "android:targetSandboxVersion". S'il est présent, sa valeur influe directement sur l'option "usesCleartextTraffic". Sa valeur doit être fixée à 1, sinon les autres modifications ne seront pas prises en compte, le trafic en clair vers des sites en HTTP étant interdit par défaut dans les autres niveaux.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest android:targetSandboxVersion="1">
    <uses-permission android:name="android.permission.INTERNET" />
    ...
</manifest>
```

### **Précision : Formule permettant de calculer la distance à vol d'oiseaux entre deux positions GPS**

Source : [https://geodesie.ign.fr/contenu/fichiers/Distance\\_longitude\\_latitude.pdf](https://geodesie.ign.fr/contenu/fichiers/Distance_longitude_latitude.pdf)

Si l'on considère deux points A et B sur la sphère, de latitudes  $\varphi^A$  et  $\varphi^B$  et de longitudes  $\lambda^A$  et  $\lambda^B$ , alors la distance angulaire en radians  $S(A-B)$  entre A et B est donnée par la relation fondamentale de trigonométrie sphérique, utilisant  $d\lambda = \lambda^B - \lambda^A$  :

$$S(A-B) = \arccos(\sin \varphi^A \sin \varphi^B + \cos \varphi^A \cos \varphi^B \cos d\lambda)$$

La distance  $S$  en mètres, s'obtient en multipliant  $S(A-B)$  par un rayon de la Terre conventionnel ( $R = 6\,378\,137$  mètres par exemple).

En termes applicatifs, la distance entre deux points dont on connaît les coordonnées  $\{\text{lat1}, \text{lon1}\}$  et  $\{\text{lat2}, \text{lon2}\}$  est donnée par la formule :

$$d = \arccos(\sin(\text{lat1}) \sin(\text{lat2}) + \cos(\text{lat1}) \cos(\text{lat2}) \cos(\text{lon1} - \text{lon2}))$$

La valeur de  $d$  est obtenue dans une unité correspondant au rayon de la sphère terrestre  $R$ . Si l'arc cosinus rend une valeur en radian, il suffit de multiplier le résultat par  $R$  pour obtenir la valeur de  $d$  en mètres.