

# Introduction au développement sous Android

## TP4

Master WeDSCi/I2L - 2<sup>me</sup> année

année 2023-2024

## Introduction

L'objectif de ce TP est d'expérimenter l'utilisation des intentions implicites via la méthode `startActivity` dans le cadre des communications inter-composants. L'application développée dans le cadre de l'exercice 2, sera à rendre selon les modalités habituelles en fin de TP.

## Exercice 1

Créez un nouveau projet Android nommé TP4XXX, où XXX sera remplacé par votre nom de famille, puis complétez celui-ci avec les étapes successives suivantes :

1. Configurez l'interface graphique avec les boutons tels qu'ils apparaissent dans la figure 1. Vous penserez à faire figurer l'intitulé des boutons dans le fichier `strings.xml`.
2. Complétez votre activité principale de telle sorte que l'appui sur l'un de ces boutons provoque l'affichage du texte du bouton correspondant dans un message `Toast`. Il vous est demandé de gérer tous les boutons dans la même fonction *callback*;

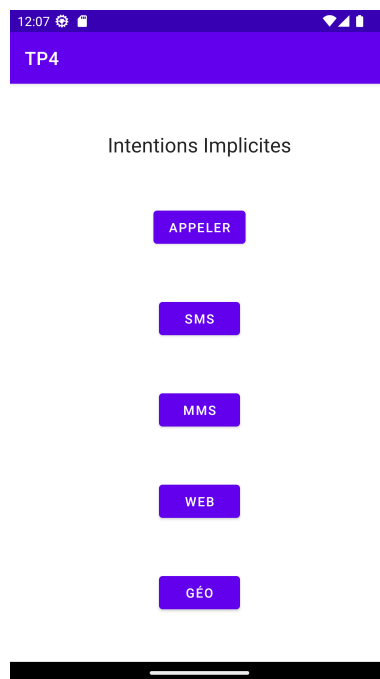


FIGURE 1 – Vue de l'interface graphique de l'exercice 1.

3. Modifiez le code de gestion de l'appui sur le bouton **Appeler** de telle sorte que cet événement déclenche la génération d'une intention visant l'appel vers le numéro fourni. L'action à utiliser est `ACTION_DIAL` et l'*URI* à créer doit utiliser le schema `tel`, associée à une information correspondant à un numéro de téléphone quelconque ;

- Modifiez le code de gestion de l'appui sur le bouton **SMS** de telle sorte que cet événement déclenche la génération d'une intention visant l'envoi d'un sms. L'action à utiliser est `ACTION_SENDTO` et l'*URI* à créer doit utiliser le schema `sms`, associé à une information correspondant à un numéro de téléphone quelconque. Testez cette fonctionnalité;
- Faites de même pour le bouton **MMS**;
- Complétez le code de gestion du bouton **WEB**, de manière à ce qu'il ouvre la page web `http://www-lisic.univ-littoral.fr` en cas d'appel. L'action à utiliser est `ACTION_VIEW`;
- Faites de même pour le bouton **Géo**, en utilisant l'URI suivante (extraite de la documentation android) :  
`geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+CA+94043`

## Exercice 2 (à rendre)

Modifiez votre application de telle sorte que :

- l'appui sur le bouton **Appel** lance une activité secondaire permettant de saisir le numéro de téléphone à utiliser pour passer l'appel (voir figure 2a). En cas d'appui sur **valider**, vous vérifierez que l'utilisateur a bien saisi un numéro de téléphone. Dans la négative, il sera averti de son erreur via un message « Toast »;
- l'appui sur l'un des boutons **SMS** ou **MMS** lance une activité secondaire permettant de saisir le numéro de téléphone à utiliser pour l'action demandée, ainsi que le texte du message (voir figure 2b). Comme précédemment, il sera nécessaire que les deux champs de saisie soient complétés en cas de validation par l'utilisateur;
- l'appui sur le bouton **Web** lance une activité secondaire permettant de saisir l'url à ouvrir (voir figure 2c). En cas de validation, le texte représentant l'url ne pourra pas être vide;
- l'appui sur le bouton **Géo** lance une activité permettant de saisir la latitude et la longitude d'un lieu dont on souhaite visualiser l'emplacement (voir figure 2d). Là également, les deux coordonnées géographiques devront avoir été renseignées. On précise que ces coordonnées devront être fournies au format décimal. Par exemple, les coordonnées de Lille seront (50.6329700, 3.0585800) (en degrés) tandis que celles de Calais seront (50.9519400, 1.8563500) (en degrés).



FIGURE 2 – Vues des 4 activités secondaires de l'application à développer dans l'exercice 2.