

# [Innovations technologiques]

*Gestion d'une application de ventes d'articles de 2ème main*

## 1 Enoncé

Il est demandé de développer une application Android en Java permettant de gérer les opérations classiques d'une application qui gère une vente d'articles d'occasion:

- l'ajout d'un nouvel article
- différentes recherches (par prix, par lieu)

Les articles seront stockés dans une base de données. Cette base de données sera accédée par des scripts PHP ou services web JAVA qui renverront des informations au format texte pur, au format XML ou au format JSON.

## 2 Cahier des charges

### 2.1 Ecran d'accueil

L'écran d'accueil affichera plusieurs boutons permettant d'accéder aux différentes fonctionnalités de l'application:

- Poster un nouvel article
- Rechercher un article par prix
- Rechercher un article par ville

### 2.2 Ajouter un article

#### **Informations à saisir**

Le formulaire d'encodage d'un article demandera à l'utilisateur de saisir les informations suivantes:

- son nom
- un descriptif détaillé
- son prix
- l'état (neuf ou utilisé)
- la ville du vendeur
- une info signalant si l'article peut être envoyé ou doit être retiré en mains propres

Une fois les informations encodées, l'utilisateur devra cliquer sur un bouton. Si l'article a été sauvegardé correctement dans la base de donnée, la fenêtre se ferme, et on revient au menu principal.

#### **Les types des widgets**

L'état sera représenté par des boutons radio. Même remarque pour l'info signalant si l'article peut être envoyé ou doit être retiré en mains propres

## ***Un wizard***

La ville sera choisie grâce à un wizard. Ce wizard sera géré par une activité développée "à la main" par le programmeur. Dans cette activité, la liste des villes sera affichée. Le vendeur devra alors choisir la ville

## ***2.3 Rechercher un article par prix***

Un formulaire de recherche demandera à l'utilisateur de saisir le prix minimum et le prix maximum correspondant à la recherche, puis cliquer sur un bouton afin de lancer la recherche. Le résultat de la recherche s'affichera dans une autre activité (voir 2.5).

## ***2.4 Rechercher un article par lieu***

Le lieu correspondra à la ville sur laquelle on veut effectuer une recherche. Ici aussi, le nom de la ville pourra être saisi au clavier, où complété grâce à un wizard.

L'utilisateur devra donc choisir la ville de recherche, puis cliquer sur un bouton afin de lancer la recherche. Le résultat de la recherche s'affichera dans une autre activité (voir 2.5).

## ***2.5 Afficher les résultats d'une recherche***

### ***Liste d'articles***

Les différents articles seront affichés sous forme de liste<sup>1</sup>, à raison de 5 articles maximum par écran. Il doit y avoir moyen de naviguer dans la liste avec des boutons de navigation (afin d'afficher les 5 articles suivants ou les X précédents).

### ***Affichage selon le device***

Pour chaque article de la liste, l'affichage dépendra du device.

Sur les smartphones:

- en mode portrait: on affichera le nom de l'article et son prix sur 1 ligne
- en mode paysage: le nom de l'article, le prix et la ville seront affichés sur 1 ligne

Sur les tablettes:

- en mode portrait: le nom de l'article, le prix et la ville seront affichés sur 1 ligne
- en mode paysage: on affichera en 1 ligne le nom de l'article, le prix, la ville et l'état l'objet

### ***Tris***

Il y aura la possibilité de trier les objets trouvés:

- soit par prix croissant
- soit par prix décroissant

---

<sup>1</sup> Un TableLayout par exemple

## **Mise en veille**

La liste sera rechargée après chaque mise en veille du smartphone pendant l'exécution de l'application.

## **2.6 Descriptif d'un objet**

Si on clique sur un article de la liste des résultats d'une recherche, un descriptif de celui-ci apparaîtra dans une autre activité. Cet écran permettra d'obtenir l'intégralité des informations de l'article sélectionné en affichant toutes ses informations.

# **3 Contraintes**

## **3.1 Gestion des erreurs**

Gérez un maximum d'erreurs possibles en affichant des messages d'erreur si nécessaire.

## **3.2 Ergonomie**

- Soignez votre programme afin que les informations soient présentées clairement à l'écran. Il faut aussi que l'utilisation du programme soit aisée.
- La présentation des écrans de l'application doivent garder une même structure et cohérence (couleurs, polices)
- Le programme ne doit pas planter si vous changez l'orientation du device
- Le programme devra être adapté pour les smartphones et pour les tablettes

## **3.3 Présentation des écrans**

- Chaque écran devra posséder un titre
- Sur chaque écran (excepté l'écran principal du menu), un bouton "RETOUR" permettra de revenir à l'écran précédent.
- Le programme doit exister en version française et en version anglaise<sup>2</sup>. La langue d'affichage changera automatiquement selon la langue de l'OS.

# **4 Remise du travail**

- Le projet doit être envoyé par mail à l'adresse [guilhem.wetteren@condorcet.be](mailto:guilhem.wetteren@condorcet.be) pour le 23 janvier au plus tard à 18h00.
- Le sujet du mail doit être "[3IN] Remise du projet Android "
- Le projet devra être remis dans un fichier compressé (format ".zip") et sera nommé de la manière suivante: **NomPrenom.zip**.

---

<sup>2</sup> Excepté le contenu de la base de données qui, lui, restera stocké dans la langue d'encodage