

# [Innovations technologiques]

## Enoncé du projet

*Application de gestion de scores*

### 1 Enoncé

Il est demandé de développer une application Android en Java permettant de gérer les scores obtenus à divers jeux par plusieurs personnes.

Les scores et la liste des utilisateurs seront stockés dans une base de données sur un serveur MySQL. Cette base de données sera accédée par des scripts PHP qui renverront des informations au format texte pur ou au format JSON.

#### 1.1 Connexion

Lors du démarrage de l'application, il sera demandé à l'utilisateur de se logger en saisissant son pseudo et son mot de passe dans un formulaire.

Si celui-ci ne possède pas de compte, il devra cliquer sur un bouton qui fera apparaître un formulaire de création de compte sous le formulaire de connexion. L'utilisateur devra choisir un pseudo pas encore existant, taper 2 fois son mot de passe, puis cliquer sur le bouton de sauvegarde. Si la sauvegarde s'effectue correctement, le formulaire de création de compte devra disparaître et l'utilisateur pourra essayer de se connecter.

#### 1.2 Ecran d'accueil

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

- Ajouter un score
- Afficher le top 10 d'un jeu
- Afficher la liste des jeux gérés actuellement par l'application
- Lister les utilisateurs

### **1.3 Ajouter un score**

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

- le nom du jeu
- le score obtenu

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

#### **Remarque**

Le nom du jeu peut soit être saisi au clavier, soit être sélectionné grâce à un wizard (géré par une activité développée "à la main").

### **1.4 Afficher le top 10 d'un jeu**

Un formulaire permettra à l'utilisateur de saisir le nom d'un jeu au clavier ou via un wizard. Une fois le nom du jeu choisi, il devra appuyer sur un bouton pour voir le top 10 des scores d'un jeu.

### **1.5 Afficher la liste des jeux gérés actuellement**

La liste des jeux actuellement gérés sera affichée à raison de 5 jeux par écran. Un système de navigation devra permettre d'afficher les 5 jeux suivants ou les précédents.

Pour chaque ligne représentant un jeu, on devra trouver le nom du jeu, le pseudo du joueur ayant actuellement le meilleur score et un bouton. En cliquant sur ce bouton, il sera possible d'afficher le top 10 des scores pour ce jeu.

### **1.6 Lister les utilisateurs**

Les pseudos des différents utilisateurs devront être affichés sur un écran avec, si nécessaire, la possibilité de faire défiler cet écran.

#### **Remarque**

Le formulaire devra être créé "à la volée" en Java.

## 2 Les ressources

### 2.1 *creation\_db.sql*

Ce fichier contient le code SQL pour créer la base de données. Vous pouvez l'importer via PhpMyAdmin afin de recréer la table t\_objets.

#### **Attention**

Avant d'importer le code SQL, la base de données doit être créée et doit se nommer "ProjetAndroid"

### 2.2 *se\_connecter.php*

#### **Entrées**

Ce script reçoit des données transmises par une requête POST.

<code>\$_POST['pseudo']</code>	Le pseudo de l'utilisateur
<code>\$_POST['mdp']</code>	Le mot de passe

#### **Retour**

Le RPC renverra simplement un code de retour au format texte. Le code retour peut être l'une des valeurs suivantes:

0	Connection OK
100	Problème de pseudo (non transmis ou vide)
110	Problème de mot de passe (non transmis ou vide)
200	Combinaison pseudo/mot de passe incorrecte
1000	Problème de connexion à la DB
2000	Un problème autre est survenu

## 2.3 *creer\_compte.php*

### Entrées

Ce script reçoit des données transmises par une requête POST.

<code>\$_POST['pseudo']</code>	Le pseudo de l'utilisateur
<code>\$_POST['mdp']</code>	Le mot de passe

### Retour

Le RPC enverra les informations sous le format JSON suivant:

```
{
  "code": xxxxx,
  "id": xxxx
}
```

L'id représentera la valeur de clé primaire (valeur numérique autoincrémentée) correspondant à l'utilisateur connecté.

Le code retour peut être l'une des valeurs suivantes:

0	Sauvegarde dans la DB OK
100	Problème de pseudo (non transmis ou vide)
110	Problème de mot de passe (non transmis ou vide)
200	Pseudo déjà existant
1000	Problème de connexion à la DB
2000	Un problème autre est survenu

Si le code de retour est différent de 0, l'objet "id" ne se trouvera pas dans le flux JSON.

Exemple:

```
{ "code": 200 }
```

## 2.4 ajouter\_score.php

### Entrées

Ce script reçoit des données transmises par une requête GET.

\$_GET['score']	Le score obtenu
\$_GET['jeu']	Le nom du jeu
\$_GET['id_pseudo']	L'id du joueur

### Retour

Le RPC renverra simplement un code de retour au format texte. Le code retour peut être l'une des valeurs suivantes:

0	Score ajouté
100	Problème de score (non transmis ou 0)
110	Problème de nom du jeu (non transmis ou vide)
120	Problème d'id (non transmis ou 0)
1000	Problème de connexion à la DB
2000	Un problème autre est survenu

## 2.5 *lister\_jeux.php*

### **Entrées**

Ce script ne reçoit aucun paramètre.

### **Retour**

Le RPC enverra les informations sous le format JSON suivant:

```
{
  "code": xxxxx,
  "jeux":
  [
    "nom_jeu": xxxxx,
    "nom_jeu": xxxxx,
    "nom_jeu": xxxxx,
    ...
  ]
}
```

Les valeurs possibles du code sont les suivantes:

0	Aucun problème
500	Aucun jeu trouvé (la table des scores est vide pour le moment)
1000	Problème de connexion à la DB
2000	Un problème autre est survenu

## 2.6 afficher\_top.php

### Entrées

Ce script reçoit des données transmises par une requête GET.

\$_GET['jeu']	Le score obtenu
---------------	-----------------

### Retour

Le RPC enverra les informations sous le format JSON suivant:

```
{
  "code": xxxxx,
  "joueurs":
  [
    {
      "pseudo": xxxx,
      "score": xxxx,
    },
    {
      "pseudo": xxxx,
      "score": xxxx,
    },
    ...
    ...
  ]
}
```

Les valeurs possibles du code sont les suivantes:

0	Aucun problème
100	Nom du jeu non transmis ou vide
500	Aucun joueur trouvé (la table des scores est vide pour le moment)
1000	Problème de connexion à la DB
2000	Un problème autre est survenu

Pour chaque entrée dans le top, on récupère son pseudo et le score obtenu.

## 2.7 *lister\_pseudos.php*

### Entrées

Ce script ne reçoit aucun paramètre.

### Retour

Le RPC enverra les informations sous le format JSON suivant:

```
{
  "code": xxxxx,
  "jeux":
  [
    "pseudo": xxxx,
    "pseudo": xxxx,
    "pseudo": xxxx,
    ...
  ]
}
```

Les valeurs possibles du code sont les suivantes:

0	Aucun problème
500	Aucun pseudo trouvé (la table des scores est vide pour le moment)
1000	Problème de connexion à la DB
2000	Un problème autre est survenu

## 3 Les tables

### 3.1 La table "utilisateurs"

Champ	Description
id_utilisateur	Clé primaire / numéro automatique
pseudo	Pseudo
mdp	Mot de passe du client <sup>1</sup>

---

<sup>1</sup> Pour l'exercice, les mots de passe sont stockés en clair dans la base de données. Dans une situation réelle, il faudrait bien sûr crypter les mots de passe par soucis de sécurité.



## 3.2 La table "scores"

Champ	Description
id_score	Clé primaire / numéro automatique
jeu	Nom du jeu <sup>2</sup>
Score	Score obtenu au jeu
id_utilisateur	Clé étrangère permettant d'identifier l'utilisateur qui a posté le score

## 4 Contraintes et remarques

### 4.1 Gestion des erreurs

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

### 4.2 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 devra être adapté pour les smartphones et pour les tablettes

### 4.3 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>3</sup>. La langue d'affichage changera automatiquement selon la langue de l'OS.

### 4.4 Scripts PHP

Les scripts PHP fournis peuvent être modifiés et adaptés. Si nécessaire, vous pouvez créer vos propres RPC. Ceux-ci doivent alors être documentés.

## 5 Remise du travail

- Le projet doit être publié sur github. Le lien github devra être envoyé par mail à l'adresse [guilhem.wetteren@condorcet.be](mailto:guilhem.wetteren@condorcet.be) pour le samedi 14 janvier au plus tard à 18h00
- Les fichiers devant se trouver sur github doivent être :
  - le projet Android
  - les RPC php, seulement si ils ont été modifiés
- Le sujet du mail doit être "[3IN] Remise du projet Android "

<sup>2</sup> Remarquons qu'on aurait pu stocker la liste des jeux dans une table supplémentaire et placer une clé étrangère depuis la table "scores".

<sup>3</sup> Excepté le contenu de la base de données qui, lui, restera stocké dans la langue d'encodage