Projet Viticulture TP 4 : projet H2O

Description du thème

Ce TP qui prend appui sur les TP et le projet H2O précédents, propose aux étudiants de mettre en pratique les techniques acquises afin de finaliser le projet.

Propriétés	Description
Intitulé long	Projet Viticulture - TP 4 : Projet H2O de traitement des eaux en viticulture
Formation concernée	BTS Services informatiques aux organisations
Matière	SLAM 4, PPE
Présentation	Mise en pratique des concepts vus pendant le TP3 sur l'utilisation des bases de données distantes sous Android. Ce projet exploite les bases de données SQLite dans un environnement mobile Android, et les bases de données MySQL par l'intermédiaire d'un serveur web.
Notions	**Savoirs** • D4.1 - Conception et réalisation d'une solution applicative • D4.2 - Maintenance d'une solution applicative **Savoir-faire** • Programmer un composant logiciel • Exploiter une bibliothèque de composants • Adapter un composant logiciel • Valider et documenter un composant logiciel • Programmer au sein d'un framework
Pré-requis	Développement mobile sous Android, TP1, 2 et 3 du projet Viticulture.
Outils	Eclipse, framework Android API 18
Mots-clés	Application mobile, Android, SQLite, MySQL, PHP, DAO
Durée	6h
Auteur(es)	Mathieu Capliez
Version	v 1.0
Date de publication	Mars 2015

Remarques pour l'enseignant :
Pour utiliser ces TP, il est indispensable de bien configurer le projet fourni, et le serveur de base de données. Dans le projet Android, les variables `serveur` et `chemin` doivent contenir respectivement l'adresse du serveur web utilisé et le chemin vers les scripts PHP.
Sur le serveur web, la base de données MySQL doit être créée et les scripts PHP configurés correctement. La configuration des variables `$login`, `$mdp`, `$bd`, `$serveur` est accessible dans le fichier `fonctions.php`.

Énoncé

Ressources :

- projet h2oSynchro ;
- archive zip contenant les scripts PHP et le script de création de tables pour la base de données MySQL.

Rappel du contexte du TP2 :
Axelle Loriot est employée par une société de gestion du traitement des eaux. En lien avec les caves de la Gironde, du Périgord et du Lot-et-Garonne, elle a pour mission d'effectuer des relevés réguliers de la qualité de l'eau rejetée dans l'environnement par les exploitations viticoles après traitement par la station.
Jusqu'à aujourd'hui, les rapports d'exploitation sont effectués à l'aide d'un tableau, les relevés sont eux effectués sur place et enregistrés à l'aide d'outils peu adaptés : papiers, notes sur un téléphone portable, etc.
Plusieurs stations de traitement des eaux font l'objet d'un suivi et chacune d'entre elles est liée à des exploitations viticoles.
Axelle souhaite disposer, dans un premier temps, d'une application dont le rôle est de faciliter les relevés. Cette application devra être accessible via un environnement mobile utilisant l'OS Android.
Les données relevées dans l'application Android devront être envoyées sur un serveur distant afin d'être traitées et sauvegardées. Les bilans seront alors automatiquement générés et envoyés aux différents organismes concernés.

Évolution du contexte :
L'application développée pour Axelle Loriot est maintenant fonctionnelle en local. Certains traitements prévus au départ ne sont toutefois pas encore implémentés.
Les agents participant au relevé de données peuvent effectivement utiliser l'application, mais les données collectées ne sont toujours pas centralisées. De plus, l'élaboration des bilans mensuels et annuels des stations de traitement des eaux ne sont pas encore possibles : en effet la synchronisation n'est pas active.

Lors de l'étude précédente (TP2), il a été conclu que les données provenant des tables **station** et **critere** doivent être envoyées vers la STA mobile. Les autres données (tables : **relever**, **annee**, **mois**) sont destinées à être saisies sur la STA mobile, puis transmises sur serveur pour faciliter l'élaboration des bilans.

Partie 1 : Analyse

Question 1 – Étude de l'existant

1.1. Quelles classes permettent d'atteindre la base de données locale ?

1.2. Quelles classes permettent d'atteindre la base de données distante ?

1.3. Où sont utilisés les champs de la classe **Parametres** ? Expliquer le rôle des mots clés **abstract**, **final** et **static**.

1.4. Les classes d'accès à la base de données en ligne ne contiennent pas toutes les mêmes méthodes. Certaines contiennent uniquement des méthodes **get**, d'autres uniquement des méthodes **add**. Expliquer pourquoi, et regrouper ces classes en deux catégories.

Lorsque l'utilisateur appuie sur le bouton **btnSyncDesc**, l'application doit récupérer les données contenues dans la base de données distante afin de les intégrer dans la base de données locale.

Gestion de l'appui sur le bouton **btnSyncDesc** :
1.5. Dans la méthode **onClick** associée au bouton **btnSyncDesc**, quelle méthode **onTacheTerminee()** sera exécutée après exécution de l'instruction **accesStationNet.getStations();** ?

1.6. Rappeler le rôle des méthodes **onTacheTerminee()** des classes DAONet.

1.7. Expliquer comment est choisie la méthode **onTacheTerminee()** appropriée lors de l'exécution ?

Partie 2 : Modification

Question 2 – Récupération d'informations depuis le serveur

2.1. Compléter le code permettant la synchronisation des stations dans la classe SynchroActivity. Méthode trouvée à la question 1.5.

Comme pour les stations, les critères doivent être récupérés depuis un serveur distant. Les critères sont fournis par le script **getCriteres.php** sous forme de données au format **JSON**. La méthode **getCriteres()** doit convertir ces données **JSON** en données utilisables par l'application Android.
À l'aide d'un navigateur web ou à l'aide de l'annexe 4, observer le résultat d'exécution du script **getCriteres.php**

2.2. Déterminer, à l'aide du résultat d'exécution du script **getCriteres.php,** le type d'objet qui sera récupéré lors de l'exécution de **getCriteres()** dans la classe **CritereDAONet**.

2.3. Compléter la méthode **getCriteres()** de la classe **CritereDAONet** afin d'implémenter le traitement demandé. Dans cette même classe, créer la méthode permettant de convertir les données JSON en objet.

2.4. Implanter, dans la classe **SynchroActivity**, le traitement de la section de récupération des critères, en utilisant la classe **CritereDAONet** complétée.

Question 3 – Envoi de données au serveur - méthode onClick() du bouton btnSyncAsc, classe SynchroActivity

Dans la méthode **onClick()** du bouton **btnSyncAsc**, les données sont envoyées dans l'ordre suivant : années, mois, puis relevés.

3.1. L'ordre choisi a-t-il une importance ? Justifier.

Données annuelles :
3.2. Dans la section d'envoi des années, expliquer le rôle de l'objet suivant : **accesAnneeNet**.

Le code permettant la gestion de l'envoi de données sur le serveur se trouve en annexe 5.

3.3. Expliquer le code à ajouter, puis indiquer s'il doit être placé dans une des méthodes **onTacheTerminee()** ou après ces méthodes. Justifier, puis implémenter ce code.

Données mensuelles :
3.4. À l'aide de l'annexe 6, expliquer quelles données sont transmises au script **addMois.php** lors des requêtes HTTP.

3.5. Compléter la méthode **addMois()** dans la classe **MoisDAONet** afin d'implémenter le traitement d'envoi de données au serveur.

Données de relevés :
Comme pour les données mensuelles et annuelles, implémenter le code nécessaire dans la méthode **addRelever()** de la classe **ReleverDAONet**, et dans la section appropriée de la méthode **onClick()** du bouton **btnSyncAsc** dans la classe **synchroActivity**.

Question 4 – Évolutions

Lors de la synchronisation, les champs **tvLogSyncAsc** et **tvLogSyncDesc** de la classe **SynchroActivity** doivent permettre d'informer l'utilisateur de l'application de la progression.

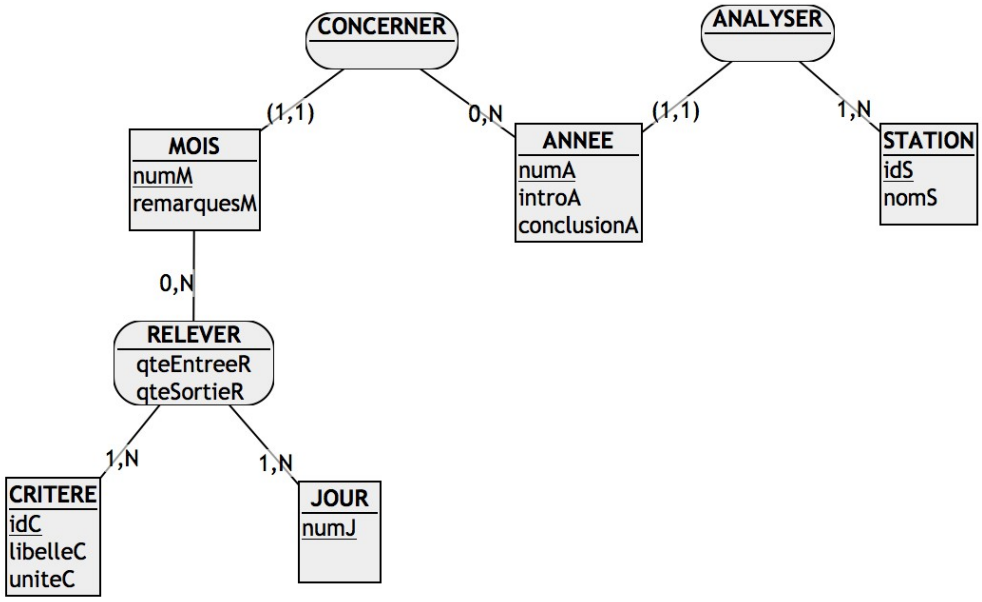
4.1. Mettre en place les méthodes permettant d'indiquer la progression de la synchronisation. Ces méthodes

doivent afficher les données mises à jour dans les champs de log associés.

4.2. Lors de l'envoi de données sur le serveur, les données déjà enregistrées ne sont pas écrasées. Quels problèmes peut-on rencontrer ? Quel traitement plus judicieux pourrait-on proposer ?

4.3. À l'aide des annexes, déterminer les traitements qui ne sont pas encore implémentés. Ces traitements doivent-ils être proposés sur l'application mobile ?

Annexe 1 : modèle conceptuel de données



Annexe 2 : script de création de la base de données serveur

```
create table station (
        idS bigint,
        nomS varchar(255),
        primary key (idS)
);

create table critere (
        idC bigint,
        libelleC varchar(255),
        uniteC varchar(50),
        primary key (idC)
);

create table annee (
        numA integer,
        introA text,
        conclusionA text,
        idS bigint,
        primary key (numA,idS),
        foreign key(idS) references station(idS)
);

create table mois (
        numM integer,
        remarquesM text,
        numA integer,
        idS bigint,
        primary key (numM,numA,idS),
        foreign key(numA,idS) references annee(numA,idS)
);
```

```
create table relever (
        numM integer,
        numA integer,
        idS bigint,
        idC bigint,
        numJ integer,
        qteEntreeR integer,
        qteSortieR integer,
        primary key (numM,numA,idS,idC,numJ),
        foreign key(numM,numA,idS) references mois(numM,numA,idS),
        foreign key(idC) references critere(idC)
);


INSERT INTO station (idS, nomS) VALUES
(1, 'Landerrouat'),
(2, 'station des vignes du sud-ouest');

INSERT INTO critere (idC, libelleC, uniteC) VALUES
(1, 'Eau', 'm3'),
(2, 'DCO', 'mg/l'),
(3, 'MES', 'mg/l'),
(4, 'haute densite', 'kg/l');
```

Annexe 3 : entretien avec Axelle Loriot

Q : Bonjour, pouvez-vous me décrire votre activité ?
Axelle Loriot : Bonjour, je vais essayer de vous l'expliquer le plus simplement possible. Je suis employée dans une organisation dont le rôle est de garantir la qualité de l'eau lorsqu'elle doit être renvoyée dans la nature après utilisation par des coopératives viticoles. Je dois contrôler la qualité de l'eau en effectuant régulièrement des relevés dans les stations de traitement des eaux.

Q : Qu'est ce qu'un relevé ?
A.L. : Un relevé, c'est un ensemble de données que je mesure lorsque je me rends sur une station de traitement. Cette station est en lien avec une cave qui rejette une grande quantité d'eau et d'éléments pouvant être polluants dans la nature. Mon rôle est de contrôler la qualité de l'eau, en entrée dans la station lorsque la cave rejette ses déchets, et en sortie de station quand l'eau est renvoyée dans la nature.

Q : Pourriez vous me décrire votre activité lorsque vous devez effectuer les relevés ?
A.L. : Actuellement, je dois noter sur des feuilles volantes les valeurs données par les instruments de mesure. Une fois toutes les mesures effectuées, je dois les recopier dans un tableur pour les enregistrer et préparer les bilans mensuels. C'est très fastidieux, et peu pratique. Je passe énormément de temps à saisir les données.

Q : Pourquoi avez-vous besoin d'une application utilisable sur votre smartphone ?
A.L. : Il y a quelques semaines, la société qui m'emploie m'a fourni un smartphone Android sur lequel je peux saisir les données directement dans un tableur. J'ai bien tenté de l'utiliser, mais les cellules sont trop petites, j'ai du mal à utiliser cette application, et il m'est déjà arrivé de constater que lorsque j'étais dans une zone non couverte par le réseau téléphonique, mon fichier tableur n'était pas accessible.

Q : Quel est le but final de vos relevés ?
A.L. : Je dois créer des rapports mensuels à intégrer dans un bilan final annuel que je dois commenter et envoyer à des organismes de contrôle.

Q : Auriez-vous des besoins particuliers ?
A.L. : J'aimerais qu'il soit possible d'avoir une application pré-configurée, ou mieux encore : qu'il soit possible de saisir certaines informations sur mon ordinateur et qu'elles soient transférées sur le smartphone.

Q : Quelles informations ?
A.L. : Les bilans mensuels, l'introduction de mon bilan annuel, les noms des stations, les unités... Je ne modifie quasiment jamais les unités, mais il m'arrive de devoir en ajouter pour de nouvelles mesures. J'ai quand même besoin des unités de mesures et des noms des stations lorsque je fais les relevés, il faut donc qu'elles soient disponibles dans le smartphone.

Annexe 4 : résultat d'exécution du script getCriteres.php

```
[{"idC":"1","libelleC":"Eau","uniteC":"m3"},
{"idC":"2","libelleC":"DCO","uniteC":"mg\/l"},
{"idC":"3","libelleC":"MES","uniteC":"mg\/l"},
{"idC":"4","libelleC":"haute densite","uniteC":"kg\/l"}]
```

Annexe 5 : gestion des années

```
AnneeDAO accesAnnee = new AnneeDAO(SynchroActivity.this);
ArrayList<Annee> listeAnnees = accesAnnee.getAnnees();

for(int i=0;i<listeAnnees.size();i++){
       accesAnneeNet.addAnnee(listeAnnees.get(i));
}
```

Annexe 6 : script addMois.php

```
<?php
include "fonctions.php";

$numM = $_POST['numM'];
$numA = $_POST['numA'];
$idS = $_POST['idS'];
$remarquesM = $POST['remarquesM'];

try {
    $cnx = connexionPDO();
    $req = $cnx->prepare("insert into mois (numM, numA, idS, remarquesM)
values('$numM','$numA','$idS','$remarquesM')");
    $resultat = $req->execute();

    print($resultat);

} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage();
    die();
}

?>
```