

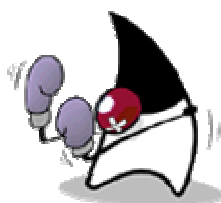
# Laboratoire de Technologies de e-commerce (Mobiles & Java) : Partie "Technologies mobiles" / **Android** : projet "**Inpres-PFM**"- suite

3<sup>ème</sup> Informatique de gestion  
2015-2016



## Inpres PFM (Plate-Forme Multimodale) - la suite

Claude Vilvens et Christophe Charlet



## 1. Préambule

L'Unité d'Enseignement "**Programmation réseaux, web et mobiles**" (10 ECTS - 135h) se structure en trois Activités d'apprentissage de la manière suivante :

- ♦ AA: Réseaux et technologies Internet (60h - 45%)
- ♦ AA: Programmation.Net (30h - 22%)
- ♦ AA: Technologie de l'e-commerce et mobiles (45h - 33%)

Le contexte de ce laboratoire de "Technologie de l'e-commerce et mobiles" est le même que celui du laboratoire de Réseaux et technologies Internet, à savoir celui de " Inpres-PFM " qui vise à la gestion d'une plate-forme multimodale.

## 2. Règles d'évaluation

Afin d'éviter tout problème lié à l'évaluation du cours de Technologies de l'e-commerce, des règles d'évaluation ont été établies et seront utilisées par les enseignants de l'équipe responsable du cours.

1) L'évaluation établissant la **note globale** de l'AA "Technologie de l'E-commerce et mobiles" est réalisée de la manière suivante :

- ♦ théorie : un examen écrit en janvier 2016 (sur base d'une liste de questions fournies en décembre et à préparer) et coté sur 20;
- ♦ laboratoire : 4 évaluations (aux dates précisées), chacune cotée sur 20; la moyenne de ces 4 cotes fournit une note de laboratoire sur 20; ces 4 laboratoires porteront sur :
  - a) laboratoire "**Technologie mobiles**" avec un laboratoire Android (1 évaluation) et un laboratoire Windows/.NET (1 évaluation);
  - b) laboratoire "**Technologies Java Standard**" (2 évaluations).
- ♦ note finale : **moyenne pondérée de la note de théorie (poids de 5/10) et de la note de laboratoire (poids de 5/10).**

Cette procédure est d'application tant en 1<sup>ère</sup> qu'en 2<sup>ème</sup> session, ainsi que lors d'une éventuelle prolongation de session.

2) Dans le cas où les travaux sont présentés par une équipe de deux étudiants, chacun d'entre eux doit être capable d'expliquer et de justifier l'intégralité du travail (pas seulement les parties du travail sur lesquelles il aurait plus particulièrement travaillé).

3) Dans tous les cas, tout étudiant doit être capable d'expliquer de manière générale (donc, sans entrer dans les détails) les notions et concepts théoriques qu'il manipule dans ses travaux (par exemple: socket, signature électronique, certificat, etc).

4) En 2<sup>ème</sup> session, un **report de note** est possible pour chacune des trois notes de laboratoire ainsi que pour la note de théorie **pour des notes supérieures ou égales à 10/20.** Toutes les évaluations (théorie ou laboratoire) ayant des **notes inférieures à 10/20** sont **à représenter dans leur intégralité.**

5) En prolongation de session, un **report de note** est possible **pour des notes supérieures ou égales à 10/20** :

- pour la note de laboratoire mais **seulement pour sa totalité** (donc pour la somme des différentes évaluations, pas pour l'une ou l'autre partie);
- pour la note de théorie.

Les évaluations (théorie ou laboratoire dans sa totalité) ayant des **notes inférieures à 10/20** sont donc **à représenter dans leur intégralité**. Mais de plus :

- pour l'examen théorique : les réponses écrites seront présentées et explicitées oralement à deux professeurs responsables du cours;
- pour l'examen de laboratoire : on gardera le même contexte, mais des fonctionnalités différentes pourront être demandées en lieu et place d'anciennes.

Le laboratoire de "*Technologies Mobiles (partie Android)*" comportera une évaluation qui se fera **à partir du 30 octobre 2015** (avec rentrée d'un dossier papier limité à l'essentiel).

## Les travaux de l'évaluation Android

### Compétences développées :

- ◆ Maîtriser les techniques de base du développement Android.

### Dossier attendu :

1. schéma général des applications Android en termes d'activités, d'intents, d'events, de fragments (formalisme libre).

## 1. Contexte

Nous allons donc nous intéresser au serveur multithread **Serveur\_Bateaux** (de type Java/Windows-Unix) dont les clients peuvent être non seulement des applications installées mais aussi, et même surtout, des applications mobiles Android. Ce serveur centralise les opérations portant sur le chargement et le déchargement des bateaux

## 2. L'application mobile pour dockers

L'application Android **Application\_Bateaux** permet simplement aux dockers du port fluvial de prendre connaissance

- ◆ des containers en attente d'être transbordés
- ◆ de rendre compte des opérations de déchargement
- ◆ de manière générale de signaler tous les événements touchant le port.

Le **Serveur\_Bateaux**, qui attend sur le port PORT\_BATEAUX, consulte un fichier containers.csv qui contient l'occupation du parc à containers (en principe, il faudrait contacter pour cela Serveur\_Trafic mais comme ce dernier ne sera peut-être pas encore opérationnel, nous l'ignorons). L'application mobile utilisera les commandes du protocole **IOBREP** (Input-Ouput **B**oat **R**Eport **P**rotocol). Les commandes sont les suivantes :

protocole <b>IOBREP</b> - PORT_BATEAUX		
application cliente : <b>Application_Bateaux</b>		
Commande	Sémantique	réponse éventuelle – résultat attendu
LOGIN	démarrage de l'application : un responsable de dock se fait reconnaître <i>paramètres</i> : nom, password	oui ou non – validation au moyen de la table <b>Personnel</b> de BD_COMPTA (seule cette table est nécessaire)
GET_CONTAINERS	demande de la liste des containers en attente pour un destination donnée <i>paramètres</i> : la destination, une chaîne indiquant le critère de sélection des livraisons : FIRST (celle qui est en attente depuis le plus longtemps) ou RANDOM (ordre sans importance)	la liste triée selon la demande, accédée par JDBC dans le fichier containers.csv
HANDLE_CONTAINER_OUT	le responsable indique quel container va être déplacé du parc vers le bateau	oui - mémorisation du container choisi

	<i>paramètres</i> : l'identifiant du container (idéalement, une barre de progression indique la proportion de containers déjà mis en place par rapport à l'ensemble des containers de la liste)	ou non : dans le cas de FIRST, le container cité n'est pas le premier de la liste
END_CONTAINER_OUT	le responsable indique qu'il termine les opérations de chargement <i>paramètres</i> : -	oui - mise à jour du fichier containers.csv ou non
BOAT_ARRIVED	le responsable indique l'arrivée d'un bateau <i>paramètres</i> : l'identifiant du bateau et sa destination	oui + écriture dans le fichier sérialisé des bateaux amarrés au port
HANDLE_CONTAINER_IN	le responsable indique quel container va être déplacé du bateau vers le parc <i>paramètres</i> : l'identifiant du container	oui - recherche d'un emplacement libre ou non : plus d'emplacement libre
END_CONTAINER_IN	le responsable indique qu'il termine les opérations de chargement <i>paramètres</i> : -	oui - mise à jour du fichier containers.csv ou non

A priori, l'architecture de l'application pourrait être :

- ◆ une activité pour le login;
- ◆ une activité avec un menu ou une barre de tâches pour choisir entre chargements et déchargements
- ◆ deux activités correspondant respectivement aux opérations de chargement et de déchargement; ces activités montrent en permanence une liste des containers traités ou à traiter.

Ces activités communiquent à priori par intents. Les opérations réalisées sont mémorisées sur le mobile (SQLite).

### **3. Améliorations de l'application**

#### **3.1 Les graphiques statistiques**

Il s'agit de doter l'application d'un outil de synthèse des opérations, outil destiné aux contremaîtres et chefs de service exclusivement. Cette partie de l'application leur permet de voir

- ◆ l'histogramme du nombre de containers chargés ou déchargés par jour;
- ◆ la répartition du nombre de containers chargés ou déchargés par semaine par destination;
- ◆ les temps moyens de chargement/déchargement par docker.

#### **3.2 L'internationalisation**

Certains dockers ne parlent pas français mais plutôt anglais ou allemand. Il s'agit donc de modifier l'application pour qu'elle permette, au login, un choix de langue et s'adapte en

conséquence (en pratique, pour économiser du temps de développement, la modification ne concerne que l'interface du choix entre chargement et déchargement).

### **3.3 Utilisation des fragments**

Il s'agit de remplacer les deux activités de chargement et déchargement par des fragments dans une version d'Android qui supporte les fragments.

*s: CV & C.C.*

