



**TACT** factory  
mobile agency

**MYQCM**

# CAHIER DES SPECIFICATIONS DETAILLÉES

Ce présent document répond aux besoins techniques de la société TACTfactory pour la conception et le développement d'une solution mobile à destination de ses formateurs et de leurs apprentis.

# IDENTIFICATION DU DOCUMENT

## Document

### *Cahier des spécifications détaillées*

*Analyse technique de la solution en réponse aux cahiers des charges*

*Exemplaire : docx, PDF*

## Historique des versions

NUMERO DE VERSION	DATE	NOM, PRENOM	DETAIL
V 1.0	06/04/2016	Trouvé, Antoine	Création
V 2.0	15/05/2016	Trouvé Antoine	Ajout Base de données et Architecture réseau
V 3.0	18/05/2016	Trouvé Antoine	Ajout des parties Administration, Webservice et Application
V3.1	18/05/2016	Trouvé Antoine	Relecture pour validation

## Contact

FONCTION	NOM, PRENOM	FONCTION	CONTACT
Client	J-C. Petitot	Responsable gestion de projet	
Client	Y. Pintas	Chef de projet Android Chef de projet Objective-C	01.83.64.25.33 <a href="mailto:contact@tactfactory.com">contact@tactfactory.com</a>
Client	B. Dague	Chef de projet C#	
Prestataire	A. Trouvé	Apprentis Développeur Chef de projet	06.30.61.51.30 <a href="mailto:antoinetrouve.france@gmail.com">antoinetrouve.france@gmail.com</a>

# AVANT-PROPOS

## Avant-propos

### CONTEXTE

TACTfactory est une société de développement d'applications mobile. La société souhaite externaliser la conception, le développement et la mise en production de l'ensemble d'une solution mobile, sur trois plateformes Windows Phone, Android et IOS. Cette solution est destinée à un usage interne à l'éco système de l'entreprise (étudiant et formateur). Elle permet aux formateurs la création et la correction automatique des QCM ainsi que la consultation des résultats. Pour les étudiants, l'application mobile propose d'accéder, de répondre aux QCM dans un temps imparti et d'envoyer les réponses au formateur afin de tester leur connaissance.

### L'OBJECTIF

Concevoir, développer et mettre en production l'ensemble d'une solution web et mobile fonctionnelle pour plateforme Android, IOS et Windows Phone, destinée à nos formateurs et étudiants, permettant d'optimiser le temps passé à la gestion des examens (création, correction, diffusion). La solution devra prendre en compte l'ensemble des contraintes diffusées dans le Cahier des charges et être opérationnelle pour le 08/07/2016.

### CONFIDENTIALITE

Toutes les informations contenues dans ce présent document sont soumises à la loi sur la protection de la propriété intellectuelle Elles sont donc confidentielles.

# TABLE DES MATIERES

## Table des matières

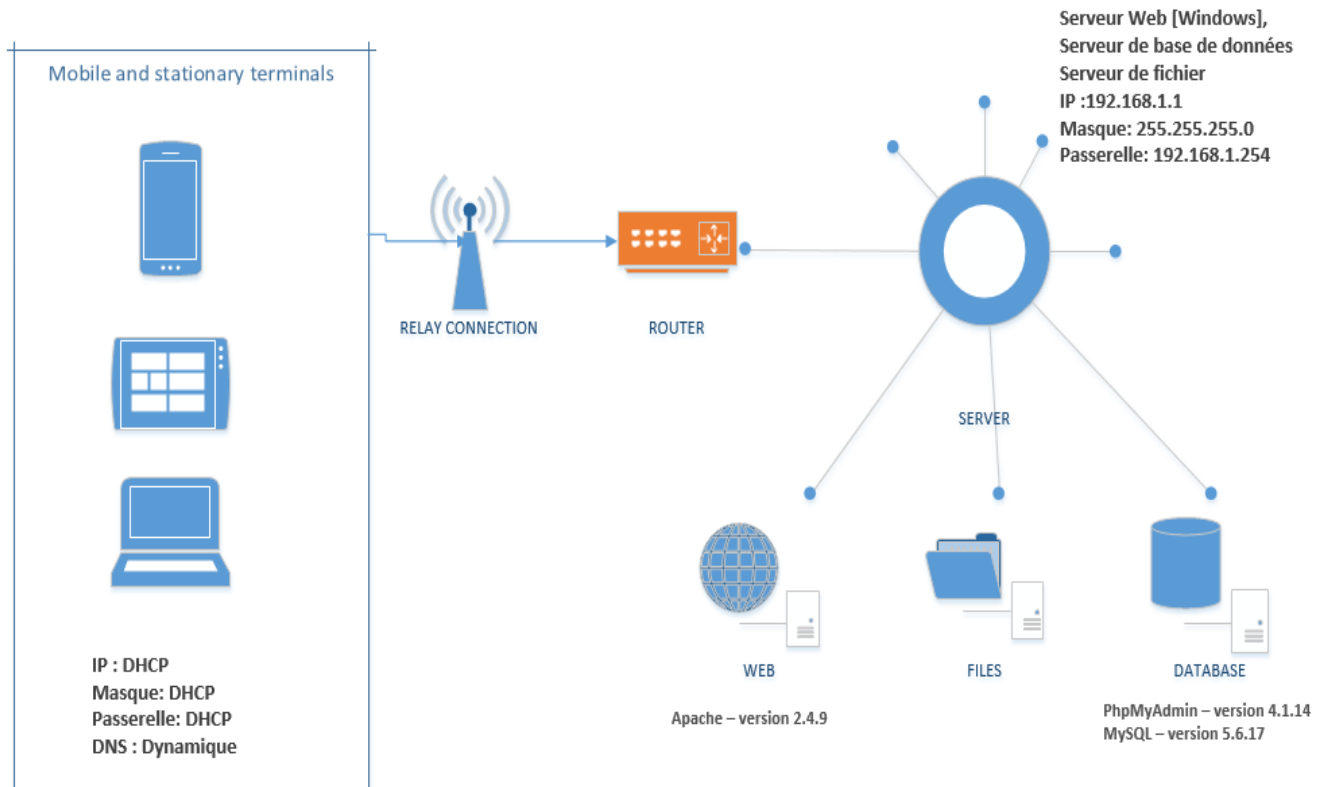
I. La solution d'ensemble	1
1. Architecture réseau	1
2. Architecture logicielle	3
3. Methodes de conception	5
II. Les bases de données	10
1. La base de données serveurs	10
2. La base de données locales du terminal mobile	11
III. Administration	12
1. Architecture du projet de developpement	12
2. Gestion des utilisateur / groupe / administrateur	14
3. Gestion des questionnaires	20
4. Gestion des résultats	36
IV. Le Web Service	40
1. Architecture du projet de developpement	40
2. Gestion du flux de connexion / Category	42
3. Gestion du flux QCM	44
4. Gestion du flux des résultats	46
V. L'application	48
1. Charte graphique et ergonomie	48
2. Les cas d'utilisation	49
3. Connexion / Accès à la liste des categories	49
4. Le Menu général	52
5. Gestion des Questionnaires	55
6. Gestion des reponses	58
7. Gestion des résultats	61

# LA SOLUTION D'ENSEMBLE

## I. La solution d'ensemble

### 1. ARCHITECTURE RESEAU

#### SCHEMA



# LA SOLUTION D'ENSEMBLE

## DESCRIPTIF

### Informations sur le système

PARAMETRES	VALEUR
Nom du poste	DESKTOP-0MS6TSC
Nom de l'OS	Windows 10 Famille
Taille du disque dur	1 To

### Configuration du réseau

PARAMETRES	VALEUR
IP	192.168.1.1
Masque de sous-réseau	255.255.255.0
DNS	Automatique

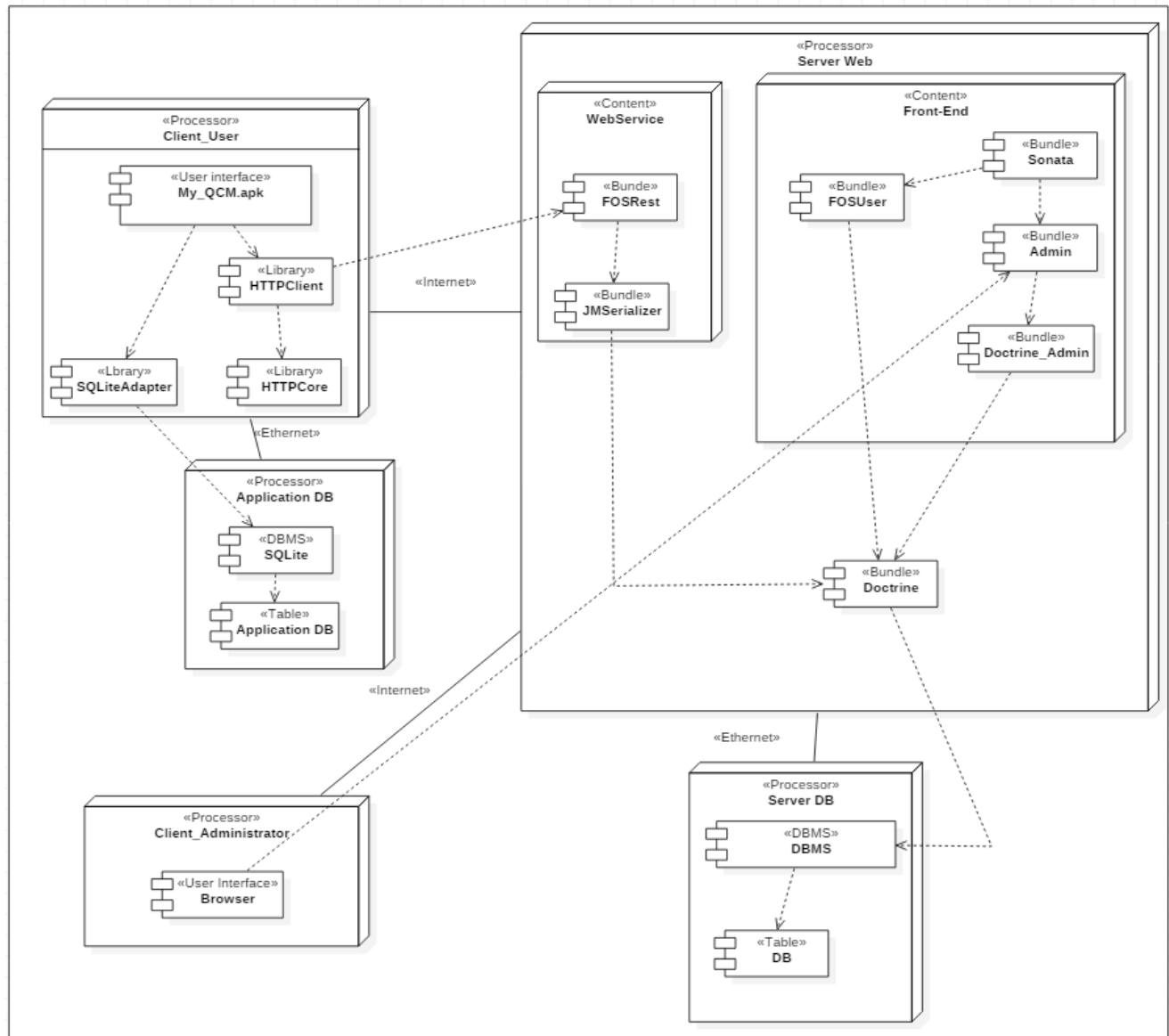
### Liste des services

NOM DU SERVICE	UTILITE
Serveur web (apache version 2.4.9)	Utilisé pour l'hébergement du site
Serveur de base de données (MySQL version 5.6.17)	Utilisé pour stocker les données
Outils de gestion de base de données (Phpmyadmin version 4.1.14)	Utilisé pour administrer la base de données

# LA SOLUTION D'ENSEMBLE

## 2. ARCHITECTURE LOGICIELLE

### SCHEMA



# LA SOLUTION D'ENSEMBLE

## DESCRIPTIF

### ***Node Client\_user***

Le node client\_user représente la partie applicative de la solution et regroupe plusieurs composants :

- Une interface utilisateur représentant l'Apk de l'application, le programme exécuté sur le téléphone
- Cet Apk dispose de deux librairies :
  - **HttpClient** qui est la librairie qui permet d'exécuter les requêtes de communication avec le Web service. Elle hérite de HTTPCore.
  - **SQLiteAdapter** qui est une librairie permettant la communication avec la base de données en SQLite. Elle contient toutes les requête de gestion de la base (Ajout, liste, suppression...).

### ***Node Application DB***

Ce node représente la partie base de données de l'application. Il est composé de :

- Un système de gestion de base de données SQLite
- La base de données

### ***Node Client\_Administrator***

Le node Client\_Administrator illustre la partie « Administration » de la solution côté client. C'est l'interface web utilisateur accessible directement à partir d'un navigateur internet.

### ***Node Server Web***

Node détaillant le serveur web. Il est subdivisé en 3 partie :

- Le node Web service
- Le node Front-end
- Le Bundle Doctrine qui permet à Symfony de communiquer avec le système gestion de base de données.

### ***Node Web service***

Node permettant la réception et l'envoi de données entre l'application et le serveur Web.

- Il est composé de deux Bundles :
  - FOSRest qui permet de générer des urls renvoyant des flux json pour permettre la circulation des informations entre l'applcatif et le serveur
  - JMSerializer permet de mettre en forme les données envoyées et qui seront par la suite renvoyées par FOSREST Bundle

### ***Node Front-end***

Node qui permet d'administrer la solution au travers d'un portail web via plusieurs Bundle Symfony :



# LA SOLUTION D'ENSEMBLE

- **Bundle Sonata** permettant de mettre en place une interface d'administration facilement et rapidement. C'est en fait un bundle regroupant d'autre bundle.
- **Bundle Admin** Fournissant à partir des informations reçues par le Bundle Doctrine-Admin une interface visuel auto-généré.

## **Node Server DB**

Il représente la partie base de données du serveur web.

En effet la solution dispose également d'un système de gestion de base de données ainsi que de la base de données de l'application.

## 3. METHODES DE CONCEPTION

### **PROCESSUS DE DEVELOPPEMENT**



### **LES CONVENTIONS DE NOMMAGE**

Il convient de respecter les conventions de nommage préétablis au commencement du projet. Vous trouverez ces conventions de nommages.<sup>1</sup>

### **LES COMMENTAIRES**

Les commentaires devront :

- Être en format officiel respectant les normes de développement universel propre à chacun des langages utilisés.
- Permettre la mise en place d'une documentation technique du code source.
- Être en anglais afin d'assurer l'universalité du projet

Les éléments obligatoires à commenter :

- Classes
- Méthodes
- Variables de classe

---

<sup>1</sup> Voir document associé situé dans le Plan Management de Projet.

# LA SOLUTION D'ENSEMBLE

Il est également autorisé le commentaire dans le code pour décrire une étape, un processus particulier. Un exemple complet est fourni en Annexe de ce document.

## LES TESTS UNITAIRES

Afin de satisfaire aux normes de qualité, il convient également de mettre en place des tests unitaires pour l'ensemble des méthodes développées.

- Framework utilisé : **JUnit 4.12**
- Installé sur chacune des plateformes de développement
- La création et l'exécution des tests devront se faire **Avant** le développement de la méthode finale.

### Exemple d'implémentation d'un test unitaire :

```
/**
 * Test Convert category to ContentValues
 */
@Test
public void categoryToContentValues() {
    Date date = new Date();
    Category cat = new Category(1, 1, "NameCategory", date);

    ContentValues values = new ContentValues();
    values.put(COL_ID, cat.getId());
    values.put(COL_IDSERVER, cat.getIdServer());
    values.put(COL_NAME, cat.getName());
    values.put(COL_UPDATEDAT, cat.getUpdatedAt().toString());

    assertNull(values);
}
```

# LA SOLUTION D'ENSEMBLE

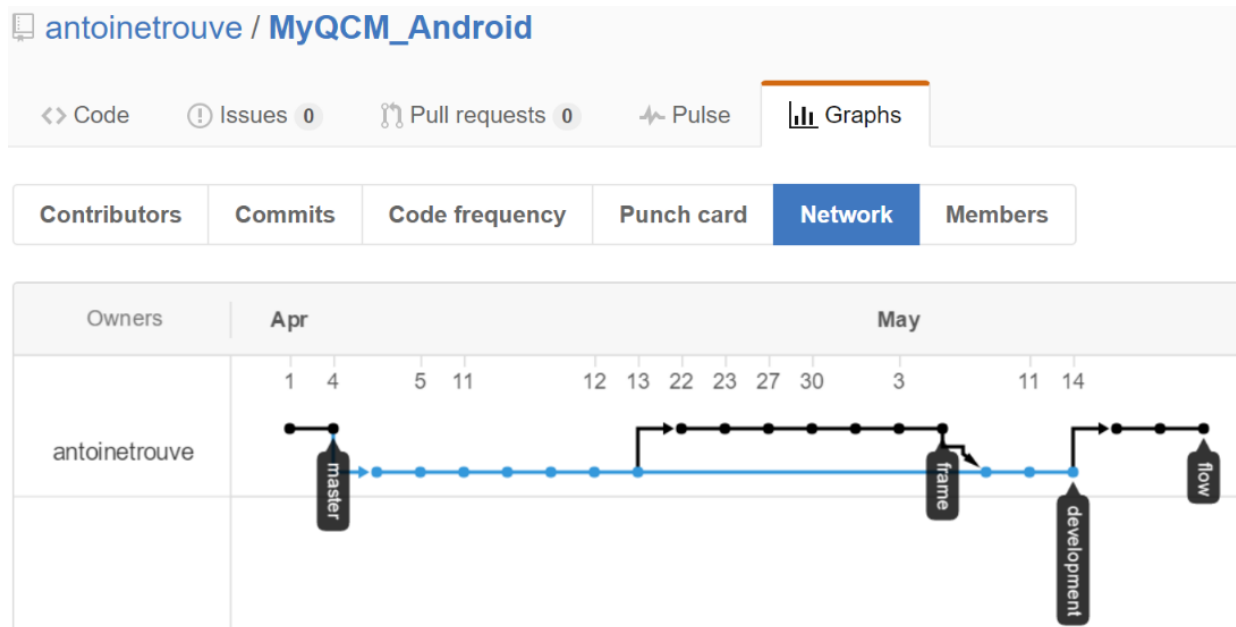
## LE VERSIONNING DU CODE SOURCE

Il est impératif que le code source soit versionné. Chacun des développements devra être intégré dans une sous-branche de la branche « développement ». Cette sous-branche devra porter le nom d'un module de l'application à développer et sera formé que d'un seul mot.

Après que les tests soient validés, il convient de :

- Puller afin de récupérer d'éventuels changement au cours de la journée
- Commiter le code développé
- Pusher tous les soirs à la fin de la journée.

OUTILS DE VERSIONNING	Git
Serveur distant / dépôt distant du code source	<a href="https://github.com/antoinetrouve">https://github.com/antoinetrouve</a>
Client Git utilisé	SourceTree

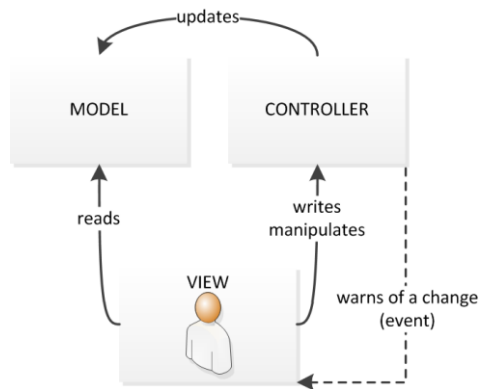


# LA SOLUTION D'ENSEMBLE

## LES PATRONS DE CONCEPTIONS UTILISES

Ce point sera détaillé dans la sous-partie « Architecture du projet » pour chacune des parties III et IV de ce document.

### **MVC – Model View Controller**



Le patron de conception MVC est appliqué au projet web ainsi qu'à la solution Android.

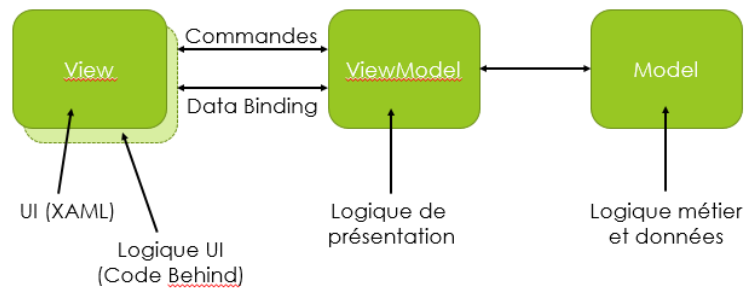
### **MVVM – Model View View Model**



Ce modèle est appliqué pour le projet applicatif Windows Phone.

- **Model** : Le modèle contient les données. Ces données proviennent de la base de données MyQCM.
- **View** : La vue correspond à ce qui s'affiche sur les écrans de l'applications. Elle contient les différents composants graphiques.
- **View-Model** : Ce composant fait le lien entre la vue et le model en gérant les liaisons de données. Le MVVM est un patron de conception .NET. C'est donc ici où va intervenir le binding. C'est cet élément qui va manipuler les vues. Une liaison faible existe entre la vue et le model.

# LA SOLUTION D'ENSEMBLE



## LE CHOIX SYMFONY2 : FRAMEWORKS JMSSERIALIZERBUNDLE ET APIRESTBUNDLE

Symfony2 permet la création d'une infrastructure complète d'un projet allant de la création de la base de données à la génération d'une interface d'administration en passant la construction de flux. Tous ceci par annotation particulière.

Voici les Bundle mis en place :

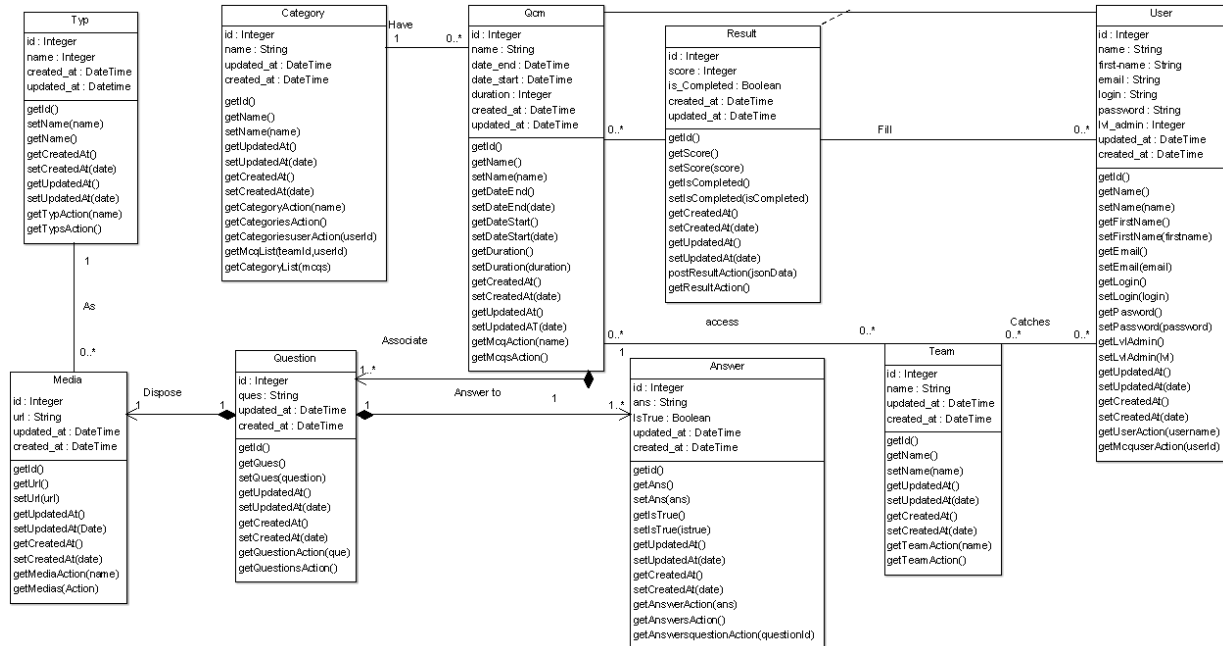
NOM DU BUNDLE	VERSION	ROLE
<b>DoctrineBundle (ORM)</b>	Version 2.8	Permet la création de la base de données en ligne de commande
<b>FOSUserBundle</b>	Version 2.3	Permet d'insérer au projet et dans la base de données la gestion des rôles utilisateurs
<b>AdminSonataBundle</b>	Version 2.3	Permet de construire une interface d'Administration selon le schéma de la base de données
<b>APIRestBundle</b>	Version 2.3	Permet la création du routing acceptant des paramètres
<b>JMSSerializerBundle</b>	Version 2.3	Permet la création du flux en format JSON

# LES BASES DE DONNEES

## II. Les bases de données

### 1. LA BASE DE DONNEES SERVEURS

#### DIAGRAMME DE CLASSE



Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

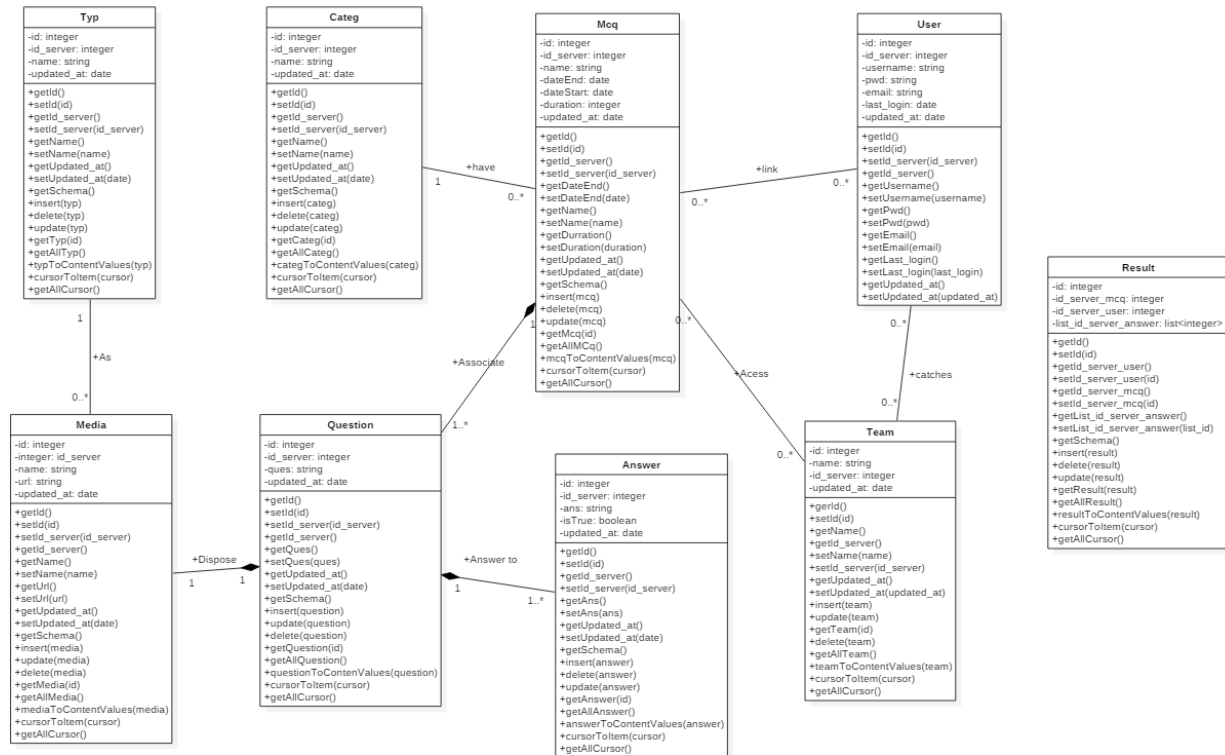
#### DICTIONNAIRE DES DONNEES

Voir document annexe associé : Cahier des spécifications détaillées – Annexes.PDF

# LES BASES DE DONNEES

## 2. LA BASE DE DONNEES LOCALES DU TERMINAL MOBILE

### DIAGRAMME DE CLASSE



Voir aussi document annexe associé : Cahier des spécifications détaillées – Annexe.PDF

### DICTIONNAIRE DES DONNEES

Voir document annexe associé : Cahier des spécifications détaillées – Annexes.PDF

## III. Administration

### 1. ARCHITECTURE DU PROJET DE DEVELOPPEMENT

#### LE PROJET

##### Environnement de développement

Le projet d'administration est développé en **PHP** en **version 5.5.12** avec le Framework Symfony2 en **version 2.8** et le bundle **SonataAdminBundle** en **version 2.3** minimum.

##### Les dépendances Symfony2

ECLIPSE	VERSION : MARS.1 RELEASE (4.5.1) BUILD ID : 20150924-1200
Dépendances	<pre>"php": "&gt;=5.5.12", "symfony/symfony": "2.8.*", "doctrine/orm": "^2.4.8", "doctrine/doctrine-bundle": "~1.4", "symfony/swiftmailer-bundle": "~2.3", "symfony/monolog-bundle": "~2.4", "sensio/distribution-bundle": "~5.0", "sensio/framework-extra-bundle": "^3.0.2", "incenteev/composer-parameter-handler": "~2.0", "friendsofsymfony/user-bundle": "~2.0@dev", "friendsofsymfony/rest-bundle": "dev-master", "jms/serializer-bundle": "dev-master", "sensio/generator-bundle": "2.3.*", "sonata-project/admin-bundle": "^2.3", "sonata-project/doctrine-orm-admin-bundle": "^2.3"</pre>














Afin d'installer les dépendances, il est conseillé d'utiliser **Composer** de Symfony2. De cette manière, le Framework installera automatiquement les package et les dépendances nécessaires.

#### ARCHITECTURE DU PROJET DE DEVELOPPEMENT

```
▼ qcm [qcm JsonFlow]
  > app
  ▼ src
    ▼ Admin
      ▼ AdminBundle
        ▼ SonataAdminBundle
          > Admin
          > Controller
          > DependencyInjection
          > Resources
          > Tests
          > AdminAdminBundleSonataAdminBundle.php
```



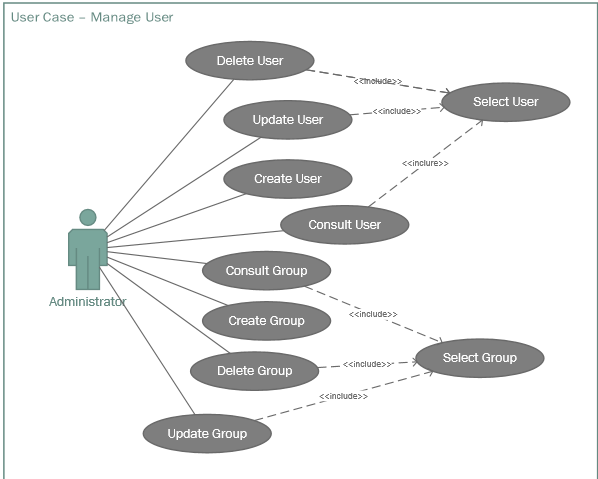
# ADMINISTRATION

NOM DU REPERTOIRE	ROLE
<b>App\</b>	Dossier d'installation et de configuration de Symfony2
<b>Src\</b>	Dossier répertoriant tous vos projets
<b>Src\Admin\</b>	Dossier répertoriant tous les projets permettant la création d'une interface d'administration
<b>Src\Admin\AdminBundle\</b>	Dossier répertoriant tous les composants de l'interface d'administration créée avec le Framework Symfony2
<b>Src\Admin\AdminBundle\SonataAdminBundle</b>	Dossier répertoriant tous les composants de l'interface d'administration créée avec le Framework Symfony2 et le bundle SonataAdminBundle.
<b>...\SonataAdminBundle\Admin</b>	Regroupe l'ensemble des class permettant la création des éléments métiers à afficher dans les vues : <ul style="list-style-type: none"> <li>▼  Admin <ul style="list-style-type: none"> <li>&gt;  AnswerAdmin.php</li> <li>&gt;  CategoryAdmin.php</li> <li>&gt;  McqAdmin.php</li> <li>&gt;  MediaAdmin.php</li> <li>&gt;  QuestionAdmin.php</li> <li>&gt;  ResultAdmin.php</li> <li>&gt;  TeamAdmin.php</li> <li>&gt;  TypeMediaAdmin.php</li> <li>&gt;  UserAdmin.php</li> </ul> </li> </ul>
<b>...\SonataAdminBundle\Controller</b>	Regroupe l'ensemble des class controller permettant de faire le lien entre la vue et la classe métier : <ul style="list-style-type: none"> <li>▼  Controller <ul style="list-style-type: none"> <li>&gt;  AnswerAdminController.php</li> <li>&gt;  CategoryAdminController.php</li> <li>&gt;  DefaultController.php</li> <li>&gt;  McqAdminController.php</li> <li>&gt;  MediaAdminController.php</li> <li>&gt;  QuestionAdminController.php</li> <li>&gt;  ResultAdminController.php</li> <li>&gt;  TeamAdminController.php</li> <li>&gt;  TypeMediaAdminController.php</li> <li>&gt;  UserAdminController.php</li> </ul> </li> </ul>
<b>...\SonataAdminBundle\DependencyInjection</b>	Classes permettant la configuration et la gestion du bundle
<b>...\SonataAdminBundle\Resources</b>	Tous les fichiers de ressources <ul style="list-style-type: none"> <li>▼  Resources <ul style="list-style-type: none"> <li>&gt;  config</li> <li>&gt;  doc</li> <li>&gt;  public</li> <li>&gt;  translations</li> <li>&gt;  views</li> </ul> </li> </ul>

# ADMINISTRATION

## 2. GESTION DES UTILISATEUR / GROUPE / ADMINISTRATEUR

### LES CAS D'UTILISATIONS



### AJOUTER UN GROUPE

#### Maquette

Voir Cahier des spécifications détaillées – Annexe.pdf

#### Composant à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD TEAM
Name	TextField	Nom du groupe	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »	name
Créer	Button	Bouton de création du groupe et renvoi sur la modification du groupe			
Créer et retourner à la liste	Button	Bouton de création du groupe et renvoi à la liste des groupes			
Créer et ajouter	Button	Créer un groupe et recharge la page d'ajout			

# ADMINISTRATION

## Fonction appelée

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (name)

## MODIFIER UN GROUPE

### Maquette

Voir Cahier des spécifications détaillées – Annexe.pdf

## Composant à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD TEAM
Name	TextField	Nom du groupe	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »	name
Mettre à jour	Button	Bouton de modification du groupe			
Mettre à jour et fermer	Button	Met à jour les données et renvoi à la liste des médias			
Supprimer	Button	Supprimer l'objet et renvoi à la liste	Si le groupe a des utilisateurs associés on ne peut le supprimer	« Une erreur est intervenue dans la suppression de cet élément »,	

## Fonction appelée

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page de modification et d'ajout (name)

# ADMINISTRATION

## SUPPRIMER UN GROUPE

### Maquette

Voir document Cahier des spécifications détaillées – Annexe.pdf

### Composant à implémenter

NOM	TYPE	VALEUR	EXCEPTION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des groupes	Si le groupe a des utilisateurs associés on ne peut le supprimer	« Une erreur est intervenue dans la suppression de cet élément »
Editer	Button	Renvoi sur la modification du groupe		

## LISTER LES GROUPES

### Maquette

Voir document Cahier des spécifications détaillées – Annexe.pdf

### Composant à implémenter

NOM	TYPE	ACTION
Listes des actions	List	Affiche une liste d'action
Editer	Button	Renvoi sur la page de modification du groupe sélectionné
Supprimer	Button	
Afficher	Button	Renvoi sur l'affichage du groupe
Export	list	Permet d'exporter les données en fonction de l'élément choisi dans la liste
Filtres	list	Permet de filtrer la liste des groupes en fonctions de champs prédéfinis

### Fonction appelée

SIGNATURE	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureListFields</code>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (name, mcqs, created_at,updated_at) Actions : view, edit, delete
<code>protected function configureDatagridFilters</code>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (name)
<code>Protected \$datagridValues</code>		Liste des ordonnancements disponibles dans la liste des éléments du groupe

# ADMINISTRATION

## AJOUTER UN UTILISATEUR

### Maquette

Voir document Cahier des spécifications détaillées – Annexe.pdf

### Composant à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR
username	TextField	Login	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »
email	TextField	Email de l'utilisateur	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »
password	textfield	Mot de passe		
roles	TextField	Rôles de l'utilisateur	Vérification de du rôle ou des rôles	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »
groupes	list	Liste des groupes de l'utilisateur		
Mcq	List	Liste des questionnaires de l'utilisateur		
Créer	Button	Bouton de création de l'utilisateur et renvoi sur la modification de l'utilisateur		
Créer et retourner à la liste	Button	Bouton de création de l'utilisateur et renvoi à la liste des utilisateurs		
Créer et ajouter	Button	Créer un utilisateur et recharge la page d'ajout		

### Fonction appelée

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (username,email,password roles, teams, mcqs)

# ADMINISTRATION

## MODIFIER UN UTILISATEUR

### Maquette

### Composant à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR
username	TextField	Login	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »
email	TextField	Email de l'utilisateur	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »
password	textfield	Mot de passe		
roles	TextField	Rôles de l'utilisateur	Vérification de l'orthographe du rôle ou des rôles	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »
groupes	list	Liste des groupes de l'utilisateur		
questionnaires	List	Liste des questionnaires de l'utilisateur		
Mettre à jour	Button	Bouton de modification de l'utilisateur		
Mettre à jour et fermer	Button	Bouton de modification de l'utilisateur et renvoi à la liste des utilisateurs		
Supprimer	Button	Supprime un utilisateur		

### Fonction appelée

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (username,email,password roles, teams, mcqs)

# ADMINISTRATION

## SUPPRIMER UN UTILISATEUR

### Maquette

#### Composant à implémenter

NOM	TYPE	VALEUR	EXCEPTION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des utilisateurs		
Editer	button	Renvoi sur la modification de l'utilisateur		

## LISTER LES UTILISATEURS

### Maquette

#### Composant à implémenter

NOM	TYPE	ACTION
Listes des actions	List	Affiche une liste d'action
Editer	Button	Renvoi sur la page de modification de l'utilisateur sélectionnée
Supprimer	Button	
Export	list	Permet d'exporter les données en fonction de l'élément choisi dans la liste
Filtres	list	Permet de filtrer la liste des utilisateurs en fonctions de champs prédéfinis

#### Fonction appelée

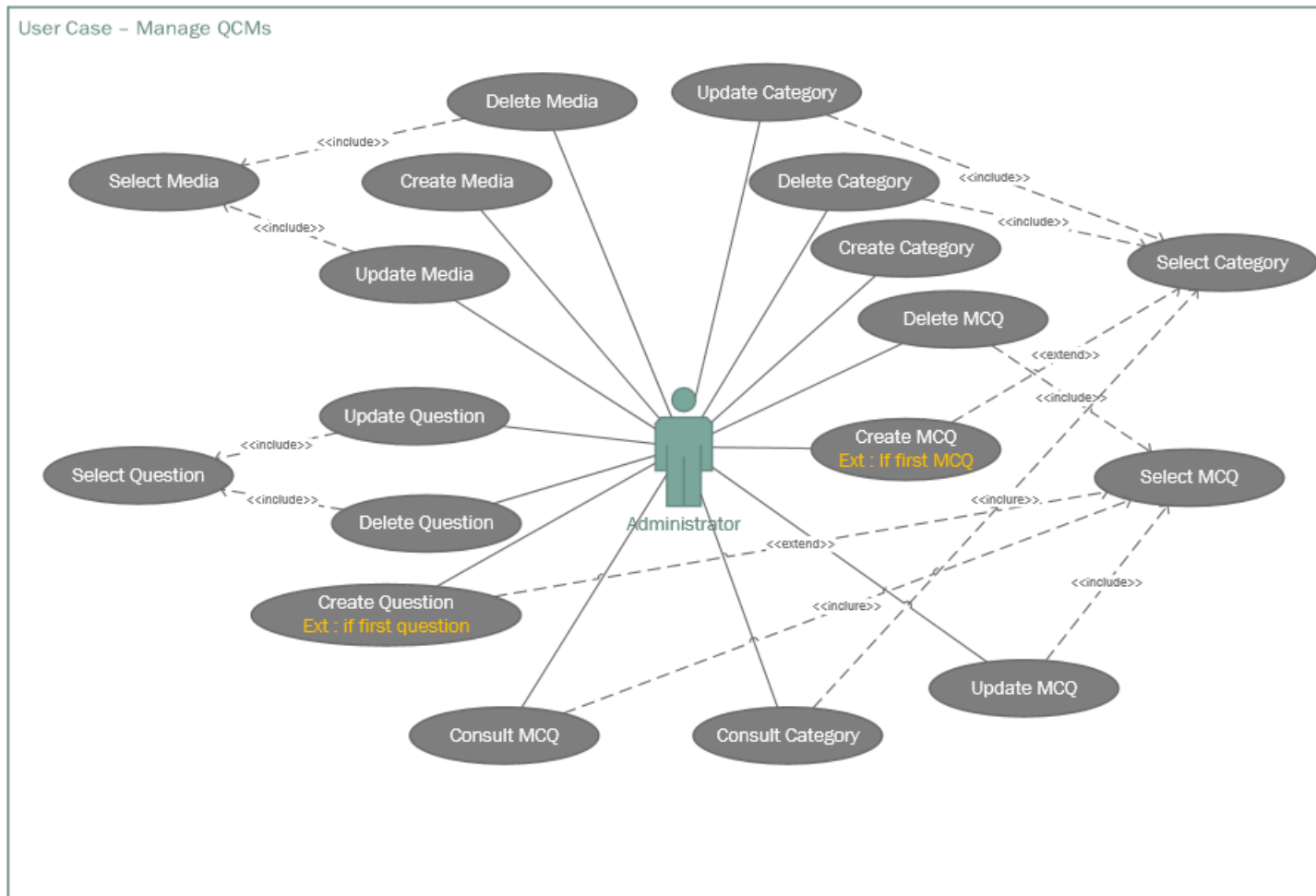
SIGNATURE	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureListFields</code>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (username,email,password roles, teams, mcqs ,created_at,updated_at) Actions : edit, delete,view
<code>protected function configureDatagridFilters</code>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (username,roles,teams)
<code>Protected \$datagridValues</code>		Liste des ordonnancements disponibles dans la liste des éléments de l'utilisateur

# ADMINISTRATION

## 3. GESTION DES QUESTIONNAIRES

### LES CAS D'UTILISATION

Une catégorie se compose de QCM et un QCM se compose d'au moins une question.





# ADMINISTRATION

## AJOUTER UN QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

Nom	Type	Valeur	Vérification	Erreur	BDD
Name	TextField	Nom du QCM	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »	name
Duration	TextField	Durée du QCM	Obligatoire	« Veuillez renseigner ce champ »	Duration
Date End	DateTime	Date de fin du QCM			Date_end
Date Start	DarteTime	Date à partir de quand le QCM va être disponible	Obligatoire	« Veuillez renseigner ce champ »	Date_start
Is Actif	Boolean	Définis si le QCM est disponible	Obligatoire	« Veuillez renseigner ce champ »	Is_actif
Categorie	Category	Category du QCM	Obligatoire	« Veuillez renseigner ce champ »	category_id
Créer	Button	Bouton de création du QCM et renvoi sur la modification d'un QCM			
Créer et retourner à la liste	Button	Bouton de création du QCM et renvoi à la liste des QCMs			
Créer et ajouter	Button	Créer un QCM et recharge la page d'ajout			

# ADMINISTRATION

## Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (name, duration, dateEnd, dateStart, isActif,category)

## MODIFIER UN QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
Name	TextField	Nom du QCM	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »	name
Duration	TextField	Durée du QCM	Obligatoire	« Veuillez renseigner ce champ »	Duration
Date End	DateTime	Date de fin du QCM			Date_end
Date Start	DarteTime	Date à partir de quand le QCM va être disponible	Obligatoire	« Veuillez renseigner ce champ »	Date_start
Is Actif	Boolean	Définis si le QCM est disponible	Obligatoire	« Veuillez renseigner ce champ »	Is_actif
Categorie	Category	Category du QCM	Obligatoire	« Veuillez renseigner ce champ »	category_id
Mettre à jour	Button	Met à jour les données du QCM			
Mettre à jour et fermer	Button	Met à jour les données du QCM et renvoi à la liste des QCM			
Supprimer	Button	Supprimer l'objet et renvoi à la liste	Si le QCM possède des questions les supprimer également		

# ADMINISTRATION

## Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page de modification et d'ajout (name, duration, dateEnd, dateStart, isActif,category)

## SUPPRIMER UN QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des QCM	Si l'élément existe en base de données	« Une erreur est intervenue dans la suppression de cet élément »
Editer	button	Renvoi sur la modification d'un QCM		

# ADMINISTRATION

## LISTER DES QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	VALEUR	A L'ÉVÉNEMENT	EQUIVALENT BDD
Name	Nom du QCM		name
Duration	Durée du QCM		duration
Date End	Date de fin du QCM		date_end
Date Start	Date à partir de quand le QCM va être disponible		date_start
Is Actif	Définis si le QCM est disponible		is_actif
Questions	Liste des questions associées au QCM	Envoi sur la page de modification de la question	
Category	La catégorie du QCM	Envoi sur la page de modification de la catégorie	category_id
Created At	Date de création du QCM		created_at
Updated At	Date de modification du QCM		updated_at

### Action – composants à implémenter

NOM	TYPE	ACTION
Listes des actions	List	Affiche une liste d'action
Editer	Button	Renvoi sur la page de modification du QCM sélectionné
Supprimer	Button	
Afficher	Button	Renvoi sur l'affichage du questionnaire
Export	list	Permet d'exporter les données en fonction de l'élément choisi dans la liste
Filtres	list	Permet de filtrer la liste de QCM en fonctions de champs prédéfinis

# ADMINISTRATION

## Fonction à implémenter

Signature	Paramètres d'entrée	Utilité
<b>protected function</b> <b>configureListFields</b>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (name, duration, dateEnd, dateStart, isActif,category,created_at,updated_at) Actions : view, edit, delete
<b>protected function</b> <b>configureDatagridFilters</b>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (name, duration, category, questions)
<b>Protected \$datagridValues</b>		Liste des ordonnancement disponible dans la liste des éléments du QCM

## AJOUTER UNE REPONSE

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

## Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	EQUIVALENT BDD
ans	TextField	Valeur de la réponse	Obligatoire	« Veuillez renseigner ce champ »	ans
Is True	Boolean	Si c'est une bonne réponse ou pas			Is_true
Question liée	list	Question associée à cette réponse	Obligatoire	« Veuillez sélectionner un élément dans la liste »	Question_id
Créer	Button	Bouton de création de la réponse et renvoi sur la modification d'une réponse			
Créer et retourner à la liste	Button	Bouton de création de la réponse et renvoi à la liste des réponses			
Créer et ajouter	Button	Créer une réponse et recharge la page d'ajout			

# ADMINISTRATION

## Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<b>protected function</b> <b>configureFormFields</b>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (ans,isTrue,question)

## MODIFIER UNE REPONSE

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
ans	TextField	Valeur de la réponse	Obligatoire	« Veuillez renseigner ce champ »	ans
Is True	Boolean	Si la réponse est correcte			Is_true
Question liée	Question	La question associée à cette réponse	Obligatoire	« Veuillez sélectionner un objet dans la liste »	Question_id
Mettre à jour	Button	Bouton de modification de la réponse			
Mettre à jour et fermer	Button	Met à jour les données et renvoi à la liste des réponses			
Supprimer	Button	Supprimer l'objet et renvoi à la liste	Si l'objet existe encore en base de données	« Une erreur est intervenue lors de la suppression de l'élément "" ». »	

### Fonction appelée

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<b>protected function</b> <b>configureFormFields</b>	FormMapper \$formMapper	Définis les éléments à afficher dans la page de modification et d'ajout (ans, isTrue,question)

# ADMINISTRATION

## SUPPRIMER UNE REPONSE

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	EXCEPTION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des réponses		
Editer	button	Renvoi sur la modification d'une réponse		

## LISTER DES REPONSES

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	ACTION
Listes des actions	List	Affiche une liste d'action
Editer	Button	Renvoie sur la page de modification de la réponse sélectionnée
Supprimer	Button	
Afficher	Button	Renvoie sur l'affichage de la réponse
Export	list	Permet d'exporter les données en fonction de l'élément choisi dans la liste
Filtres	list	Permet de filtrer la liste des réponses en fonctions de champs prédéfinis

### Fonctions à appeler

SIGNATURE	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureListFields</code>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (ans,isActive,question,created_at,updated_at) Actions : view, edit, delete
<code>protected function configureDatagridFilters</code>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (ans,isActive,question)
<code>Protected \$datagridValues</code>		Liste des ordonnancements disponibles dans la liste des éléments de la réponse

# ADMINISTRATION

## AJOUTER UNE CATEGORIE DE QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
Name	TextField	Nom de la catégorie	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »	name
Créer	Button	Bouton de création de la catégorie et renvoi sur la modification d'une catégorie			
Créer et retourner à la liste	Button	Bouton de création de la catégorie et renvoi à la liste des catégories			
Créer et ajouter	Button	Créer une catégorie et recharge la page d'ajout			

### Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected</code> <b>function</b> <code>configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (name)



# ADMINISTRATION

## MODIFIER UNE CATEGORIE DE QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
Name	TextField	Nom de la catégorie	La valeur du champ doit être unique, obligatoire	« Une erreur est intervenue dans la création de cet élément », « veuillez renseigner ce champ »	name
Mettre à jour	Button	Bouton de modification de la catégorie			
Mettre à jour et fermer	Button	Met à jour les données et renvoi à la liste des catégories			
Supprimer	Button	Supprimer l'objet et renvoi à la liste	Si la catégorie possède des QCM on ne peut pas la supprimer	« Une erreur est intervenue lors de la suppression de l'élément "" ». »	

### Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page de modification et d'ajout (name)

# ADMINISTRATION

## SUPPRIMER UNE CATEGORIE DE QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	EXCEPTION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des catégories	Si la catégorie de QCM dispose de QCM associé on ne peut la supprimer	« Une erreur est intervenue dans la suppression de cet élément »
Editer	button	Renvoi sur la modification d'une catégorie		

# ADMINISTRATION

## LISTER DES CATEGORIES

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	VALEUR	A L'EVENEMENT	EQUIVALENT BDD
Name	Nom du QCM		name
Created At	Date de création de la catégorie		created_at
Updated At	Date de modification de la catégorie		updated_at

### Actions

NOM	TYPE	ACTION
Listes des actions	List	Affiche une liste d'action
Editer	Button	Renvoi sur la page de modification de la catégorie sélectionnée
Supprimer	Button	
Afficher	Button	Renvoi sur l'affichage de la catégorie
Export	list	Permet d'exporter les données en fonction de l'élément choisi dans la liste
Filtres	list	Permet de filtrer la liste des catégories en fonctions de champs prédéfinis

### Fonction à appeler

SIGNATURE	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureListFields</code>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (name, created_at,updated_at) Actions : view, edit, delete
<code>protected function configureDatagridFilters</code>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (name)
<code>Protected \$datagridValues</code>		Liste des ordonnancements disponibles dans la liste des éléments de la catégorie

# ADMINISTRATION

## AFFICHER UNE CATEGORIE DE QCM

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composant à implémenter

NOM	VALEUR	BDD
Name	Nom de la catégorie	name
Created At	Date de création de la catégorie	created_at
Updated At	Date de modification de la catégorie	updated_at

### Fonctions à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureShowFields</code>	ShowMapper \$showMapper	Définis les éléments à afficher dans la page de la catégorie (name, createdAt, updatedAt)

## AJOUTER UNE QUESTION

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composant à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
que	TextField	Intitulé de la question	Obligatoire	« Veuillez renseigner ce champ »	que
Questionnaire	List	QCM associée	Obligatoire	« Veuillez renseigner ce champ »	Mcq_id
Media	List	Media associé à la question			Media_id
Créer	Button	Bouton de création de la question et renvoi sur la modification d'une question			
Créer et retourner à la liste	Button	Bouton de création de la question et renvoi à la liste des questions			
Créer et ajouter	Button	Créer une question et recharge la page d'ajout			

# ADMINISTRATION

## Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (ques,mcq,media)

## MODIFIER UNE QUESTION

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

## Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
ques	TextField	Intitulé de la question	Obligatoire	« Veuillez renseigner ce champ »	ques
Questionnaire	List	MCQ associée	Obligatoire	« Veuillez renseigner ce champ »	Mcq_id
Media	List	Media associé			Media_id
Mettre à jour	Button	Bouton de modification de la question			
Mettre à jour et fermer	Button	Met à jour les données et renvoi à la liste des questions			
Supprimer	Button	Supprimer l'objet et renvoi à la liste	Si la catégorie possède un média ou des questions ils sont supprimés	« Une erreur est intervenue lors de la suppression de l'élément "" . »	

## Fonctions à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page de modification et d'ajout (ques, mcq,media)

## SUPPRIMER UNE QUESTION

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

# ADMINISTRATION

## Composants à implémenter

NOM	TYPE	VALEUR	EXCEPTION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des questions	Si la question dispose de média ou de réponses associées les suppriment	« Une erreur est intervenue dans la suppression de cet élément »
Editer	button	Renvoi sur la modification d'une question		

## LISTER DES QUESTIONS

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

## Composants à implémenter

NOM	VALEUR	A L'EVENEMENT	EQUIVALENT BDD
Que	Intitulé de la question		ques
Mcq	Nom du QCM associé	Page de modification du QCM	mcq_id
Media	Nom du média associé	Page de modification du média	media_id
Answers	Liste des réponses associées à la question	Page de modification de la réponse	
Created At	Date de création de la question		created_at
Updated At	Date de modification de la question		updated_at

## Fonctions à appeler

SIGNATURE	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureListFields</code>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (ques,mcq, media,answers, created_at,updated_at) Actions : view, edit, delete
<code>protected function configureDatagridFilters</code>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (mcq)
<code>Protected \$datagridValues</code>		Liste des ordonnancements disponibles dans la liste des éléments de la question

# ADMINISTRATION

## LES POINTS D'ATTENTION

### *Créer, Modifier et supprimer*

**Attention !** la **suppression** d'une catégorie n'entraîne pas la suppression des QCM associés. Un message d'avertissement offrant la possibilité d'annuler l'action est visible par l'administrateur.

### **Attention !**

- La **suppression** d'un QCM entraîne également la suppression de toutes les questions associées. Dans ce cas, un message d'avertissement offrant la possibilité d'annuler l'action est visible par l'administrateur. Néanmoins il ne sera pas possible de supprimer un QCM actif.
- La **modification** d'un QCM n'est possible uniquement si son statut est inactif.

**Attention !** il est admis qu'un QCM est obligatoirement associé à une catégorie et à un utilisateur. De ce fait, ces associations sont donc obligatoires et bloquent la création d'un QCM.

L'association s'effectue par sélection dans une liste déroulante listant toutes les catégories disponibles.

### *Définir les temps*

L'administrateur peut également gérer la durée de diffusion et le temps de réponse d'un QCM. Chaque temps est obligatoire pour valider la création du QCM. La modification de ces temps est possible en modifiant simplement les informations dans les champs de saisis.

- **La durée de diffusion** : C'est la durée limitée où l'utilisateur final peut avoir accès au QCM sur son terminal mobile. Un champ proposant la possibilité de renseigner une date et une heure est prévu à cet effet.
- **Le temps de réponse** : C'est le temps que possède un utilisateur dès lors qu'il accède aux QCMs pour répondre aux questions et valider ses réponses. Un champ de saisi est prévu à cet effet. Le temps se compte en minute et seconde.

# ADMINISTRATION

## 4. GESTION DES RESULTATS

### AJOUTER UN RESULTAT

#### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

#### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
Score	TextFiel d	Score du résultat	Obligatoire	« Veuillez renseigner ce champ »	score
isCompleted	Boolean	Si le QCM a pu être fini à temps			is_com pleted
Questionnaire	List	QCM associé au résultat	Obligatoire	« Veuillez renseigner ce champ »	qcm_id
Utilisateur	List	Utilisateur associé au résultat	Obligatoire	« Veuillez renseigner ce champ »	
Créer	Button	Bouton de création du résultat et renvoi sur la modification du résultat			
Créer et retourner à la liste	Button	Bouton de création du résultat et renvoi à la liste des résultats			
Créer et ajouter	Button	Créer un résultat et recharge la page d'ajout			

#### Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (score,isCompleted,mcq,user)



# ADMINISTRATION

## MODIFIER UN RESULTAT

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
Score	TextField	Score du résultat	Obligatoire	« Veuillez renseigner ce champ »	score
isCompleted	Boolean	Si le QCM a pu être fini à temps			is_completed
Mcq	List	QCM associé au résultat	Obligatoire	« Veuillez renseigner ce champ »	qcm_id
Utilisateur	List	Utilisateur associé au résultat	Obligatoire	« Veuillez renseigner ce champ »	
Mettre à jour	Button	Bouton de création du résultat et renvoi sur la modification du résultat			
Mettre à jour et fermer	Button	Bouton de modification du résultat et renvoi à la liste des résultats			
Supprimer	Button	Suppression d'un résultat et retour sur la liste des résultats			

### Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureFormFields</code>	FormMapper \$formMapper	Définis les éléments à afficher dans la page d'ajout et de modification (score,isCompleted,mcq,user)

## SUPPRIMER UN RESULTAT

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composant à implémenter

NOM	TYPE	VALEUR	EXCEPTION	ERREUR
Oui, supprimer	Button	Supprime de la base de données et renvoi sur la liste des résultats		
Editer	Button	Renvoi sur la modification du résultat		

# ADMINISTRATION

## LISTER DES RESULTATS

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	VALEUR	A L'EVENEMENT	BDD
User	Utilisateur associé au résultat	Affiche la page de modification de l'utilisateur	user_id
Score	Score du résultat		score
isCompleted	Si le QCM a été fini avant la fin du temps imparti		is_completed
Mcq	QCM lié au résultat	Affiche la page de modification du QCM	mcq_id
Created At	Date de création du résultat		created_at
Updated At	Date de modification du résultat		updated_at

### Fonction à appeler

NOM	TYPE	ACTION
Listes des actions	List	Affiche une liste d'action
Editer	Button	Renvoi sur la page de modification du résultat sélectionné
Supprimer	Button	
Afficher	Button	Renvoi sur l'affichage du résultat
Export	list	Permet d'exporter les données en fonction de l'élément choisi dans la liste
Filtres	list	Permet de filtrer la liste des résultats en fonctions de champs prédéfinis

### Fonction à appeler

SIGNATURE	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureListFields</code>	ListMapper \$listMapper	Définis les éléments à afficher dans la liste ainsi que les boutons d'action (user,score,isCompleted, mcq, created_at,updated_at) Actions : view, edit, delete
<code>protected function configureDatagridFilters</code>	DatagridMapper \$datagridMapper	Définis les éléments de filtres (score, user,questionnaire)
<code>Protected \$datagridValues</code>		Liste des ordonnancements disponibles dans la liste des éléments du groupe

# ADMINISTRATION

## AFFICHER UN RESULTAT

### Maquette

Voir aussi le document annexe associé : Cahier des spécifications détaillées.PDF

### Composants à implémenter

NOM	VALEUR	A L'EVENEMENT	BDD
Score	Score du résultat		score
isCompleted	Si le QCM a été rempli à temps		is_completed
Questionnaire	QCM lié	Affiche la page de modification du QCM	mcq_id
Utilisateur	Utilisateur lié	Affichage de la page de modification de l'utilisateur	
Created At	Date de création du résultat		created_at
Updated At	Date de modification du résultat		updated_at

### Fonction à appeler

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
<code>protected function configureShowFields</code>	ShowMapper \$showMapper	Définis les éléments à afficher dans la page de la catégorie (score,user,mcq,isCompleted, createdAt, updatedAt)

## IV. Le Web Service

### 1. ARCHITECTURE DU PROJET DE DEVELOPPEMENT

#### LE PROJET

##### *Environnement de développement*











Le projet d'administration est développé en **PHP** en **version 5.5.12** avec le Framework **Symfony2** en **version 2.8** et le bundle **APIRestBundle** en **version 2.3** pour la création du routing et **JMSSerializerBundle** en **version 2.3** pour la création du flux json.

##### *Les dépendances Symfony2*

ECLIPSE	VERSION : MARS.1 RELEASE (4.5.1) BUILD ID : 20150924-1200
Dépendances	"php": ">=5.5.12", "symfony/symfony": "2.8.*", "friendsofsymfony/rest-bundle": "dev-master", "jms/serializer-bundle": "dev-master",

Afin d'installer les dépendances, il est conseillé d'utiliser **Composer** de Symfony2. De cette manière, le Framework installera automatiquement les package et les dépendances nécessaires.

#### ARCHITECTURE DU PROJET DE DEVELOPPEMENT

- ▼  IIA
  - ▼  WebServiceBundle
    - >  Controller
    - >  DependencyInjection
    - >  Entity
    - >  EntityJson
    - >  Repository
    - >  Resources
    - >  Tests
    - >  IIAWebServiceBundle.php

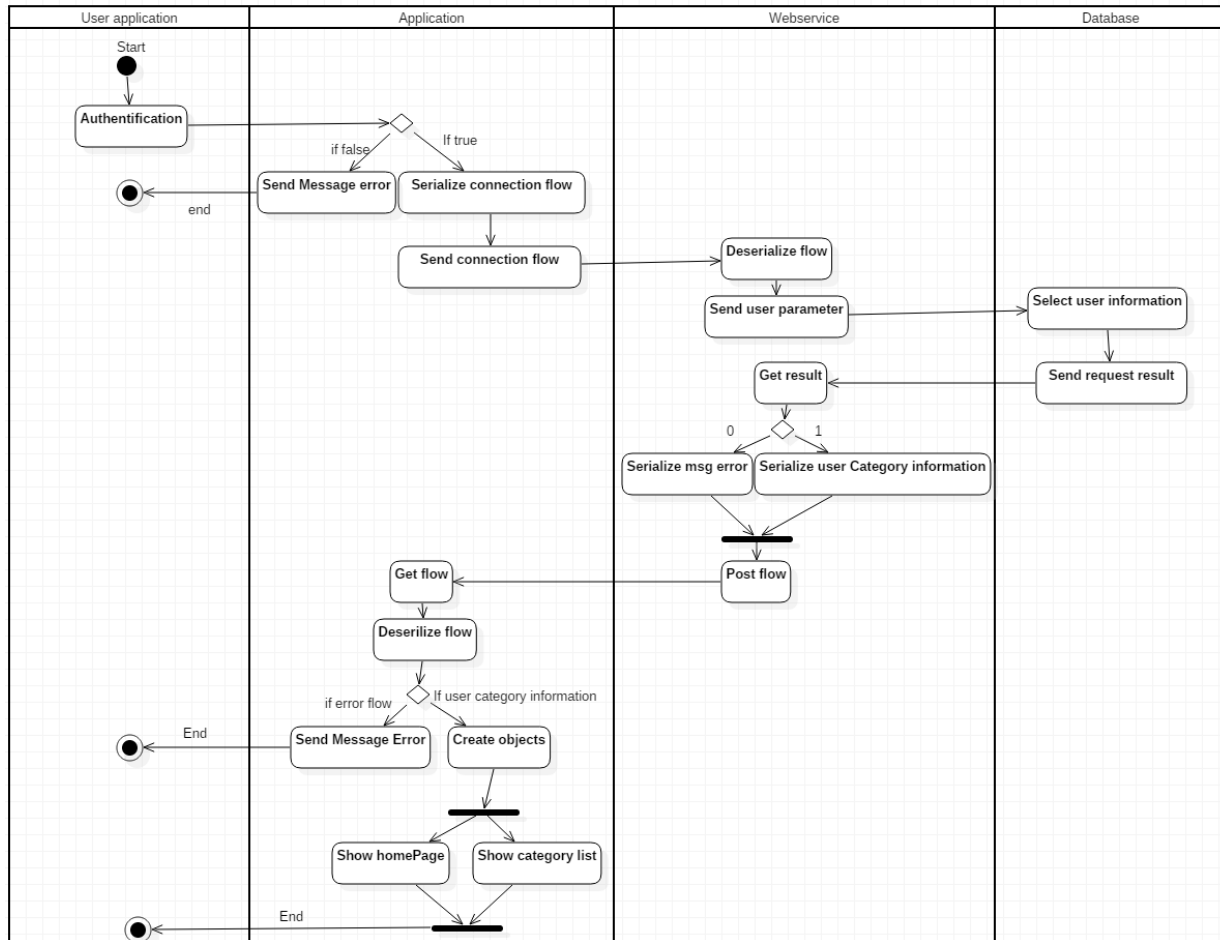
# LE WEB SERVICE

NOM DU REPERTOIRE	ROLE
<b>Src\</b>	Dossier répertoriant tous vos projets
<b>Src\IIA\</b>	Dossier répertoriant tous les projets de l'entreprise « IIA »
<b>Src\IIA\WebServiceBundle\</b>	Dossier répertoriant tous les composants permettant la création du web service.
<b>...\WebServiceBundle\Controller</b>	Regroupe l'ensemble des classes controller permettant de construire le flux reçu et envoyé ainsi que la génération du routing URL.
<b>...\WebServiceBundle\DependencyInjection</b>	Regroupe Classes permettant la configuration et la gestion des bundles.
<b>...\WebServiceBundle\Entity</b>	Regroupe l'ensemble des classes métiers. Celle-ci dépendent entièrement du diagramme de classe de la base de données serveur.
<b>...\WebServiceBundle\EntityJson</b>	Regroupe les classes existantes nécessaire pour la création du flux JSON. Spécifique uniquement pour le flux. Les classes situées dans ce dossier permettent notamment le calcul du résultat dans le controller associé.
<b>...\WebServiceBundle\Repository</b>	Regroupe l'ensemble des classes automatiquement créées par Doctrine ORM propre au fonctionnement de Symfony2
<b>...\WebServiceBundle\Resources</b>	Regroupe l'ensemble des fichiers de ressources tel que les fichiers de configuration ainsi que l'index des views.

# LE WEB SERVICE

## 2. GESTION DU FLUX DE CONNEXION / CATEGORY

### DIAGRAMME D'ACTIVITE



### LE FLUX DE CONNEXION

- **Sens du flux :** Application -> Webservice

- **Composition du flux :**

```

{
  "username": "antoine",
  "password": "$2y$13$jdfyf0v1sf4gksgoog8csu2/NePuGquN0Zx4uDBZ6Nt"
}
    
```

- **Traitement du flux de connexion :**



# LE WEB SERVICE

## FLUX DE RETOUR

- **Sens du flux :** Webservice -> Application

```
[  
  {  
    "id": 3,  
    "name": "CSharp",  
    "created_at": "2016-03-14T00:00:00+0100",  
    "updated_at": "2016-03-15T00:00:00+0100"  
  },  
  { },  
  { }  
]
```

- **Url associée :** [http://localhost/qcm/web/app\\_dev.php/api/categoriesusers/{idUser}.json](http://localhost/qcm/web/app_dev.php/api/categoriesusers/{idUser}.json)
- **Traitement du flux de retour :**



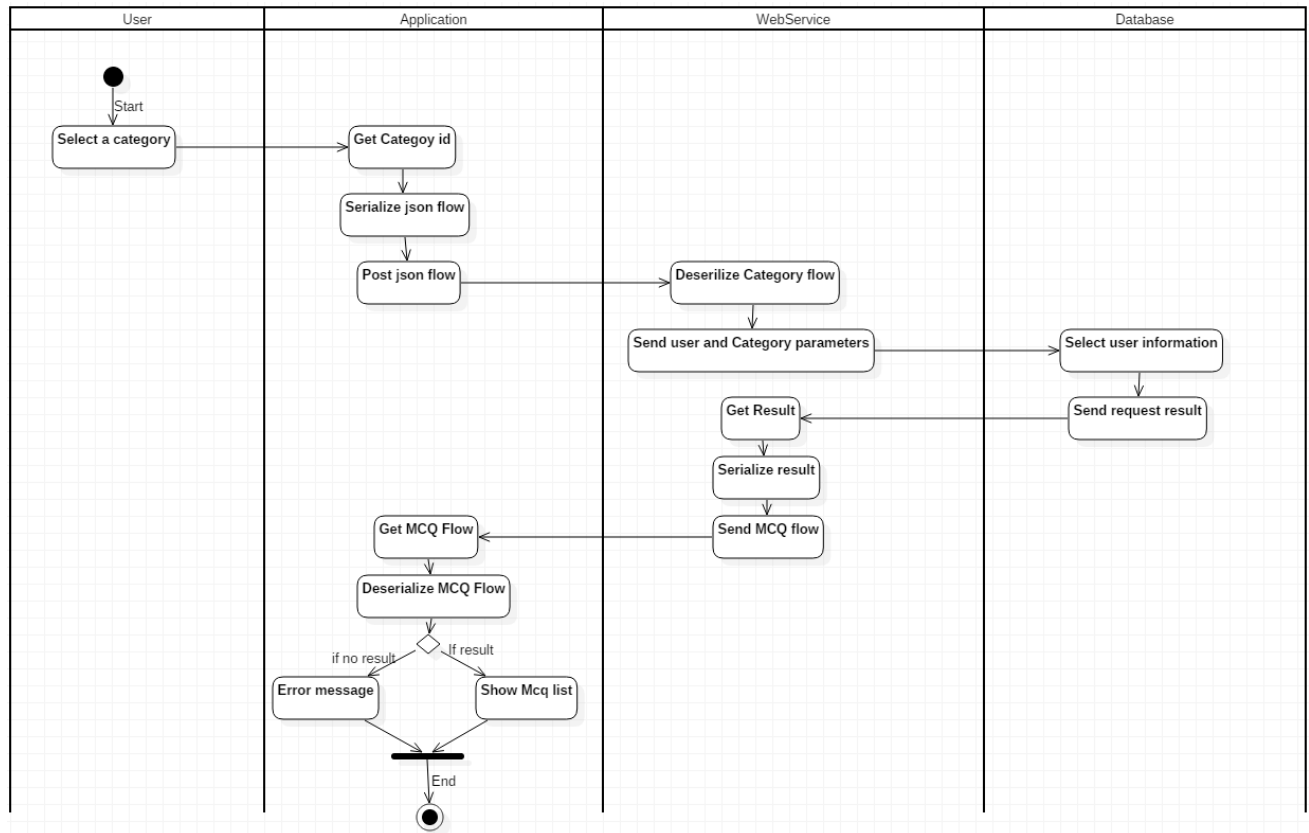
## LES FONCTIONS ASSOCIEES AU TRAITEMENT DU FLUX DE CONNEXION

NOM	PARAMETRE	RETOUR	ROLE
<b>PostConnectionFlow</b>	String username, string password	Object Category if getCategoryInformation	Création du flux de connexion et retour de l'objet Category
<b>GetCategoryInformation</b> (Appelée par PostConnectionFlow)	String flux utilisateur	String flux des catégories	Envoie le flux et retourne les informations de l'utilisateur.
<b>DeserializeJsonFlow</b> (Appelée par PostConnectionFlow)	String flux category	Object Category	Création de l'objet en fonction du flux reçu en paramètre

# LE WEB SERVICE

## 3. GESTION DU FLUX QCM

DIAGRAMME D'ACTIVITE



Voir aussi le cahier des spécification – Annexe.PDF



# LE WEB SERVICE

## LE FLUX DES QCMS

- **Sens du flux** : Webservice -> Application
- **Url associée** : [http://localhost/qcm/web/app\\_dev.php/api/mcqsusers/{idUser}.json](http://localhost/qcm/web/app_dev.php/api/mcqsusers/{idUser}.json)
- **Composition du flux** :

```
{
  "username": "antoine",
  "email": "antoinetrouve.france@gmail.com",
  "last_login": "2016-05-11T18:00:49+0200",
  "id": 1,
  "created_at": "2016-01-28T15:41:32+0100",
  "updated_at": "2016-05-11T18:00:49+0200",
  "mcqs": [
    {
      "id": 3,
      "name": "NUM3",
      "is_actif": true,
      "countdown": 5000,
      "diff_deb": "2016-03-14T00:00:00+0100",
      "diff_end": "2016-03-14T00:00:00+0100",
      "created_at": "2016-03-14T00:00:00+0100",
      "updated_at": "2016-03-14T00:00:00+0100",
      "questions": [
        {
          "id": 6,
          "name": "question 11",
          "created_at": "2016-05-05T11:09:28+0200",
          "updated_at": "2016-05-05T11:10:41+0200",
          "answers": [
            {
              "id": 7,
              "value": "reponse 11",
              "is_valid": true
            },
            { },
            { }
          ]
        }
      ]
    }
  ]
}
```

## LE TRAITEMENT DU FLUX



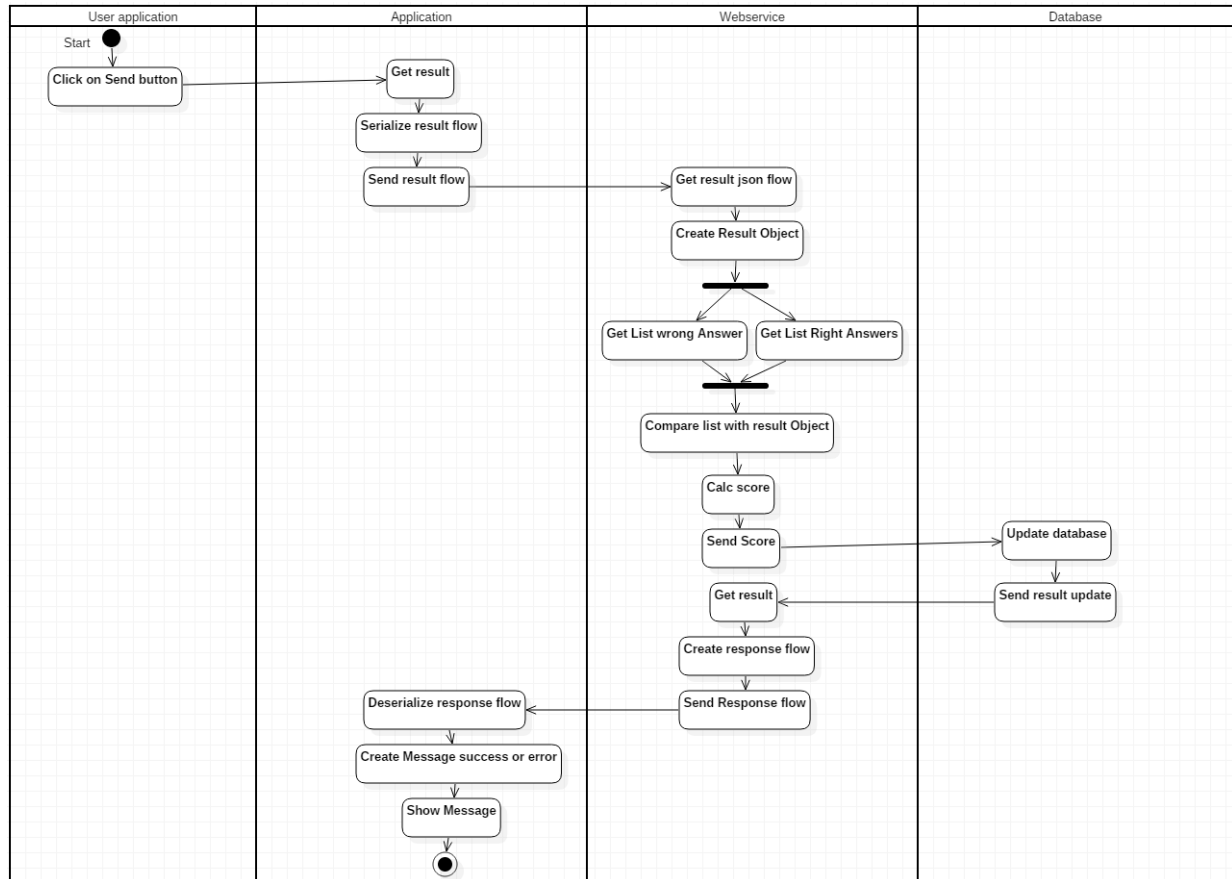
## LES FONCTIONS ASSOCIEES AU TRAITEMENT DU FLUX QCM

NOM	PARAMETRE	RETOUR	ROLE
<b>GetMCQFlow</b>	String mcqFlow	ArrayList<Mcq>	Récupération du flux envoyé par le webService
<b>DeserializeJsonFlow</b> (Appelée par GetMcqFlow)	String flux Mcq	Object MCQ	Création de l'objet en fonction du flux reçu en paramètre

# LE WEB SERVICE

## 4. GESTION DU FLUX DES RESULTATS

### DIAGRAMME D'ACTIVITE



Voir aussi Cahier des spécifications détaillées – Annexe.PDF

# LE WEB SERVICE

## LE FLUX DE RESULTAT

- Sens du flux : Application -> Webservice
- Url associée : [http://localhost/qcm/web/app\\_dev.php/api/result.json](http://localhost/qcm/web/app_dev.php/api/result.json)
- Composition du flux :

```
{
  "results": [
    {
      "result": {
        "idserverUser": 1,
        "idServerMcq": 1,
        "idServerAnswers": [
          "1",
          "2",
          "3"
        ]
      }
    },
    {
      "result": {
        "idserverUser": 1,
        "idServerMcq": 1,
        "idServerAnswers": [
          "1",
          "2",
          "3"
        ]
      }
    }
  ]
}
```

## LE TRAITEMENT DU FLUX



## LES FONCTIONS ASSOCIEES

NOM	PARAMETRE	RETOUR	ROLE
<b>PostResultAction</b>	String jsonData	null	Récupération du flux envoyé par l'application et désérialise le flux
<b>CalcResult</b> (Appelée par PostResultAction)	Answer a, Question q	null	Calcule le score et envoie en base de données

## V. L'application

### 1. CHARTE GRAPHIQUE ET ERGONOMIE

#### GRAPHISME

La charte graphique utilisé est celle correspondante au site de référence de TACTfactory, [www.tactfactory.com](http://www.tactfactory.com). L'application est essentiellement composée des trois couleurs identitaires de l'entreprise à savoir vert, blanc, noir.

#### NAVIGATION

De manière à faciliter l'accessibilité des QCM, un maximum de trois actions seront exécutées par l'utilisateur. Deux zones de navigation seront mises en place :

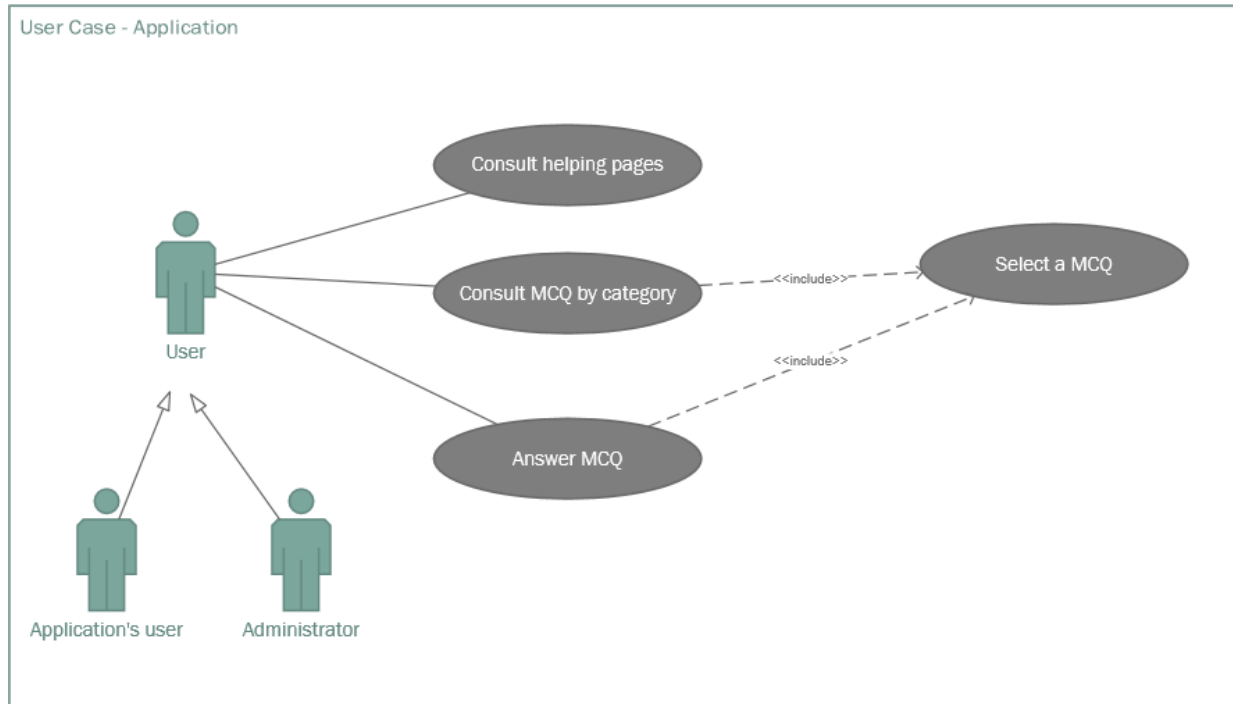
- Une zone correspondant au menu général
- Une seconde zone correspondant au cœur de l'application qui aura pour principale fonction d'accéder aux questions et d'envoyer les réponses de manières intuitives.



**Figure 1** Navigation principale

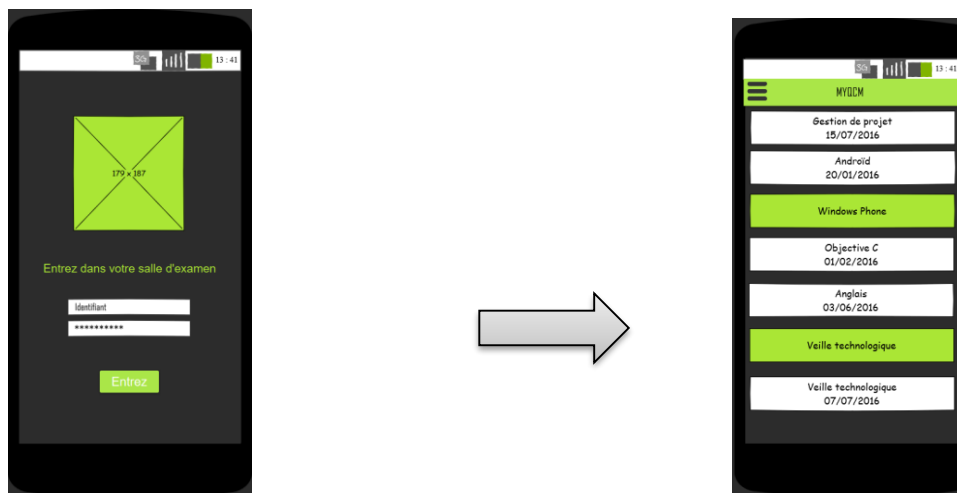
# L'APPLICATION

## 2. LES CAS D'UTILISATION



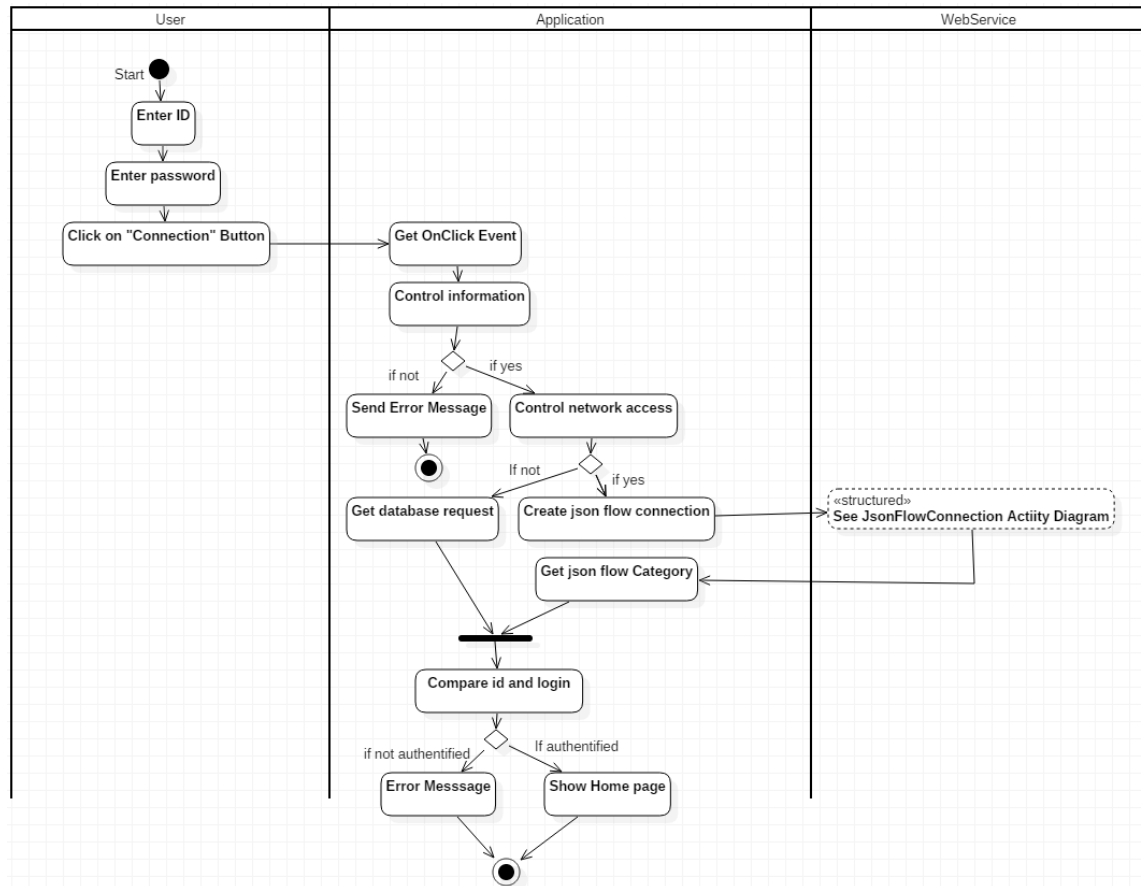
## 3. CONNEXION / ACCES A LA LISTE DES CATEGORIES

### MAQUETTE



# L'APPLICATION

## DIAGRAMME D'ACTIVITE



## COMPOSANT A IMPLEMENTER

NOM	TYPE	VALEUR	VERIFICATION	ERREUR	BDD
Identifiant	TextField	Id de l'utilisateur	La valeur du champ doit être obligatoire	« Veuillez renseigner ce champ »	username
Mot de passe	TextField	Mot de passe	La valeur du champ doit être obligatoire	« Veuillez renseigner ce champ »	password
Connexion	Button	Connection	Authentification	« Erreur lors de la connection »	

# L'APPLICATION

## FONCTION APPELEE

NOM	PARAMETRE	RETOUR	ROLE
<b>IsAuthenticated</b>	String username, String password	Boolean true si authentifié	Authentification de l'utilisateur
<b>IsConnected</b>		Boolean true si le terminal est connecté au réseau	Vérification de la connexion au réseau
<b>SetOnClick</b>			Récupérer les éléments de la vue
<b>GetUserInformation</b> <i>(Appelée par IsAuthenticated)</i>	String username, string password	User user	Récupérer l'utilisateur en base de données
<b>PostConnectionFlow</b>	String username, string password	Object Category if getCategoryInformation	Création du flux de connexion et retour de l'objet Category
<b>GetCategoryInformation</b> <i>(Appelée par PostConnectionFlow)</i>	String flux utilisateur	String flux des catégories	Envoie le flux et retourne les informations de l'utilisateur.
<b>DeserializeJsonFlow</b> <i>(Appelée par PostConnectionFlow)</i>	String flux category	Object Category	Création de l'objet en fonction du flux reçu en paramètre

# L'APPLICATION

## 4. LE MENU GENERAL

### MAQUETTE



### DIAGRAMME D'ACTIVITE

Voir cahier des spécifications détaillées – Annexe.PDF



# L'APPLICATION

## COMPOSANT A IMPLEMENTER

### Les pages

NOM DE LA PAGE	ACTIONS
Home	Récupère et affiche la liste des catégories
Menu	
Profil	Récupère et affiche les informations de l'utilisateur
Rules	Affiche la liste des règles
Legal Mentions	Affiche la liste des mentions légales
Help	Affiche les aides pour le fonctionnement du QCM
Logout	Renvoi sur la page de connexion

### Les composants

PAGE	NOM	TYPE	VALEUR	VERIFICATION	ERREUR
Toutes	ButtonBurger	Button	Bouton affichant la liste des éléments du menu		
Home	ListItem	List	Liste contenant les catégories de l'utilisateur		
Menu page	menu	List	Liste contenant tous les éléments associés au menu		
Menu page	button_home_menu	Button	Bouton envoyant sur la page d'accueil	Récupération de la liste des catégories	« Erreur dans la récupération de la liste des catégories »
Menu page	button_profil_menu	Button	Bouton envoyant sur la page de profil	Récupération des informations utilisateurs	« Erreur dans la récupération de l'utilisateur »
Menu page	button_rules_menu	Button	Bouton envoyant sur la page des règles des QCM		
Menu page	button_legal_mention	Button	Bouton envoyant sur la page des mentions légales		
Menu Page	button_help_menu	Button	Bouton envoyant sur la page d'aide		
Menu page	button_logout_menu	Button	Bouton renvoyant sur la page de connexion		

# L'APPLICATION

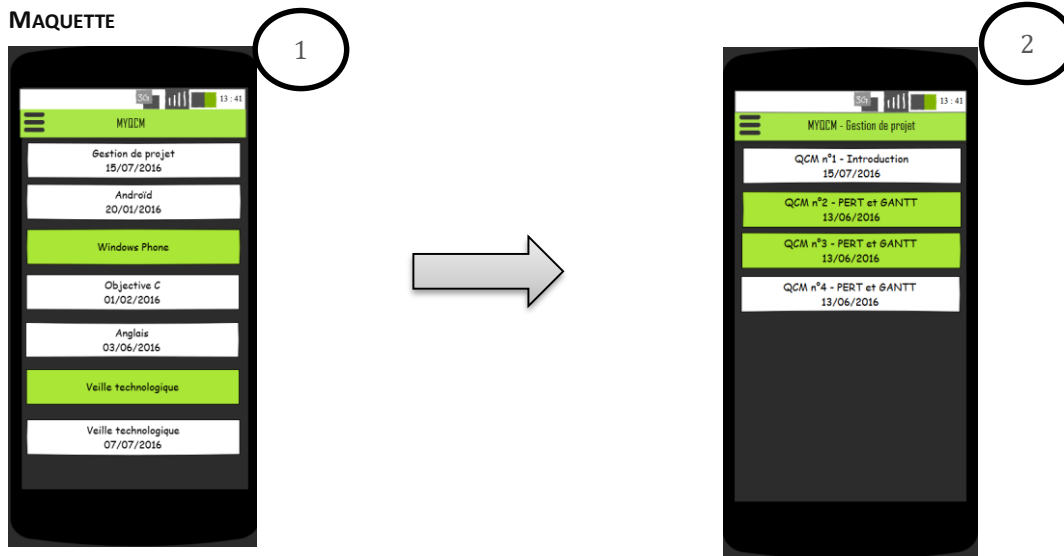
## FONCTION APPELEE

SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
ShowMenu		Affiche le menu lié au buttonBurger
MenuSelectedItem	MenuItem item	Récupère la page sélectionner dans le menu et affiche la page à partir de l'item passer
LogOut		Renvoi l'utilisateur sur la page de connexion
GetUserInfo	Int id	Renvoi les éléments liés à l'id d'utilisateur passé en paramètre
Getcategories		Renvoi la liste des catégories
ShowPage	Page page	Renvoi sur la page passé en paramètre

# L'APPLICATION

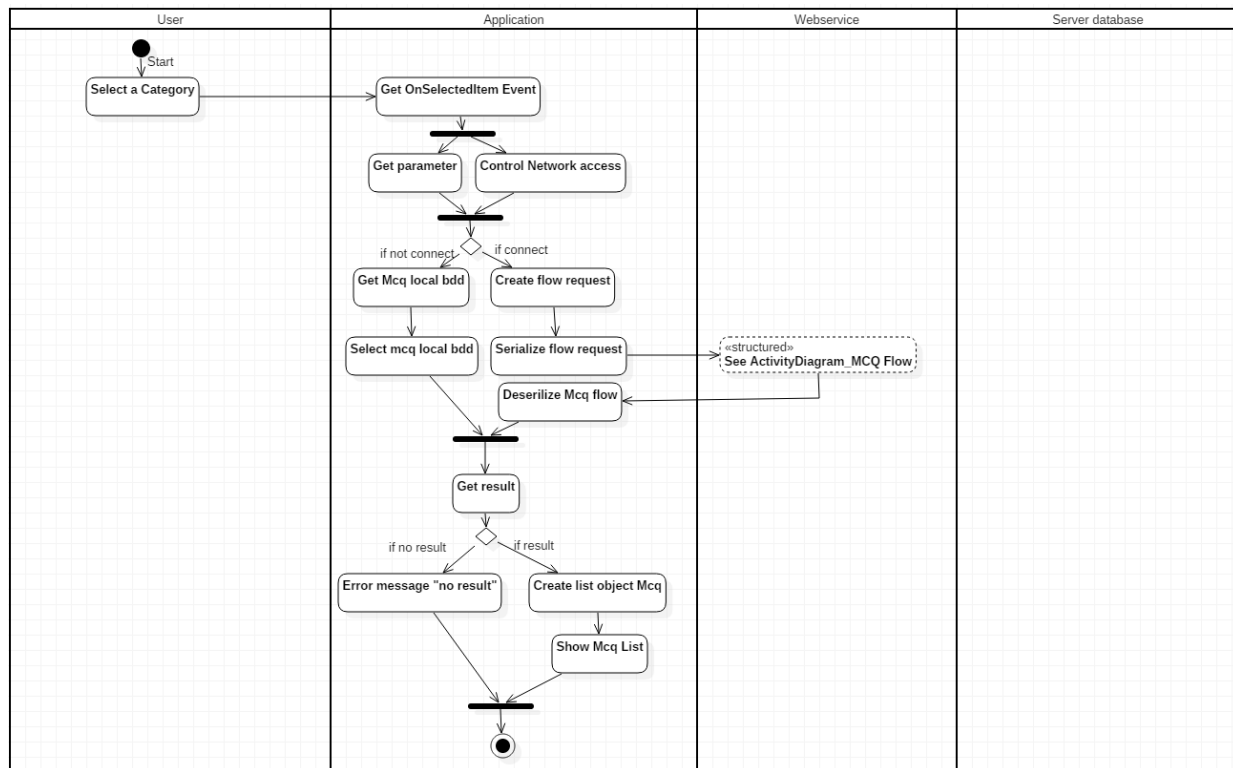
## 5. GESTION DES QUESTIONNAIRES

### MAQUETTE



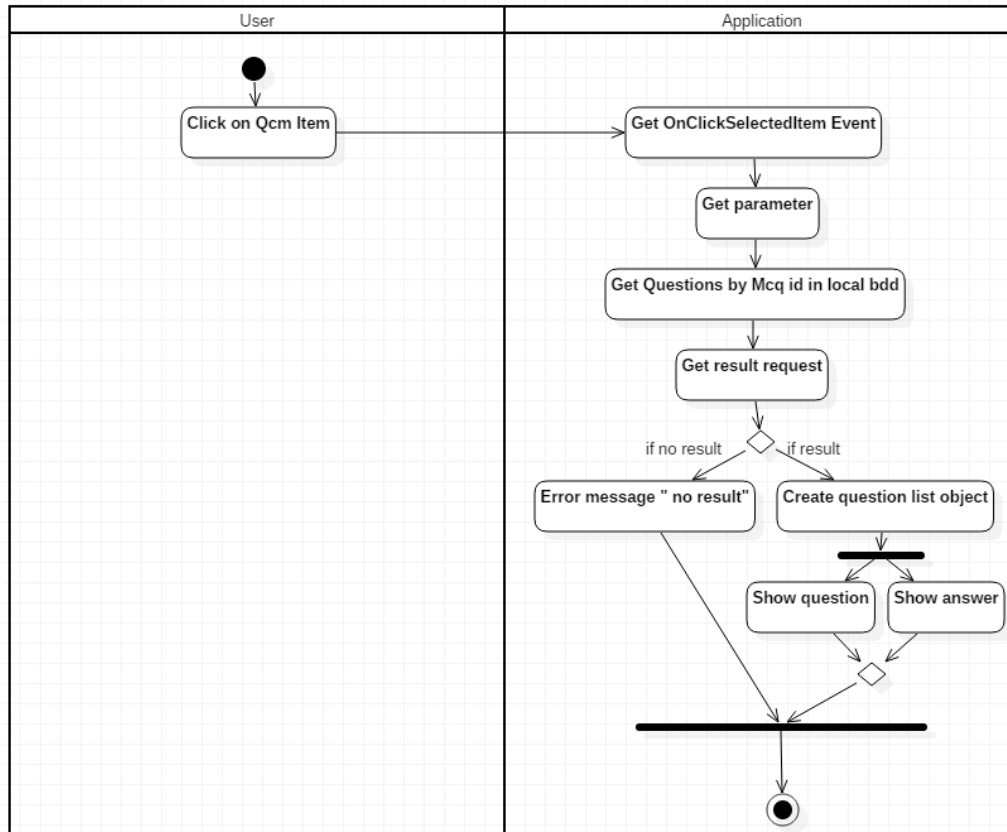
### DIAGRAMME D'ACTIVITE

#### Affichage des Qcms



# L'APPLICATION

## Affichage des questions



## COMPOSANT A IMPLEMENTER

PAGE	NOM	TYPE	VALEUR	VERIFICATION	ERREUR
Qcm	ListItem	List	Objet Qcm		
Question	ListItem	List	Objet Question		

## FONCTION APPELEE

NOM	PARAMETRE	RETOUR	ROLE
GetQuestions	Int idUser, Int idMcq		Selectionner dans la bdd les questions associés.
GetMcqs	Int idUser, int idCategory		Sélectionner dans la bdd les qcms associés
IsConnected		Boolean true si le terminal est connecté au réseau	Vérification de la connexion au réseau
OnSelectedItem			Récupérer les éléments de la vue avec en parametre l'id de l'objet

# L'APPLICATION

<b>PostMcqFlow</b>	Int idServerMcq, Int idServerUser	Object Qcm	Création du flux de connexion et retour de l'objet Qcm
<b>GetMcqInformation</b> <i>(Appelée par PostMcqFlow)</i>	String flux mcq	String flux des mcq	Envoie le flux et retourne les informations du qcm.
<b>DeserializeJsonFlow</b> <i>(Appelée par PostMcqFlow)</i>	String flux mcq	Object mcq	Création de l'objet en fonction du flux reçu en paramètre
<b>UpdateQcm</b>	Qcm qcm	Int result	Modification en base de données local
<b>InsertQcm</b>	Qcm qcm	Int result	Insertion en base de données

# L'APPLICATION

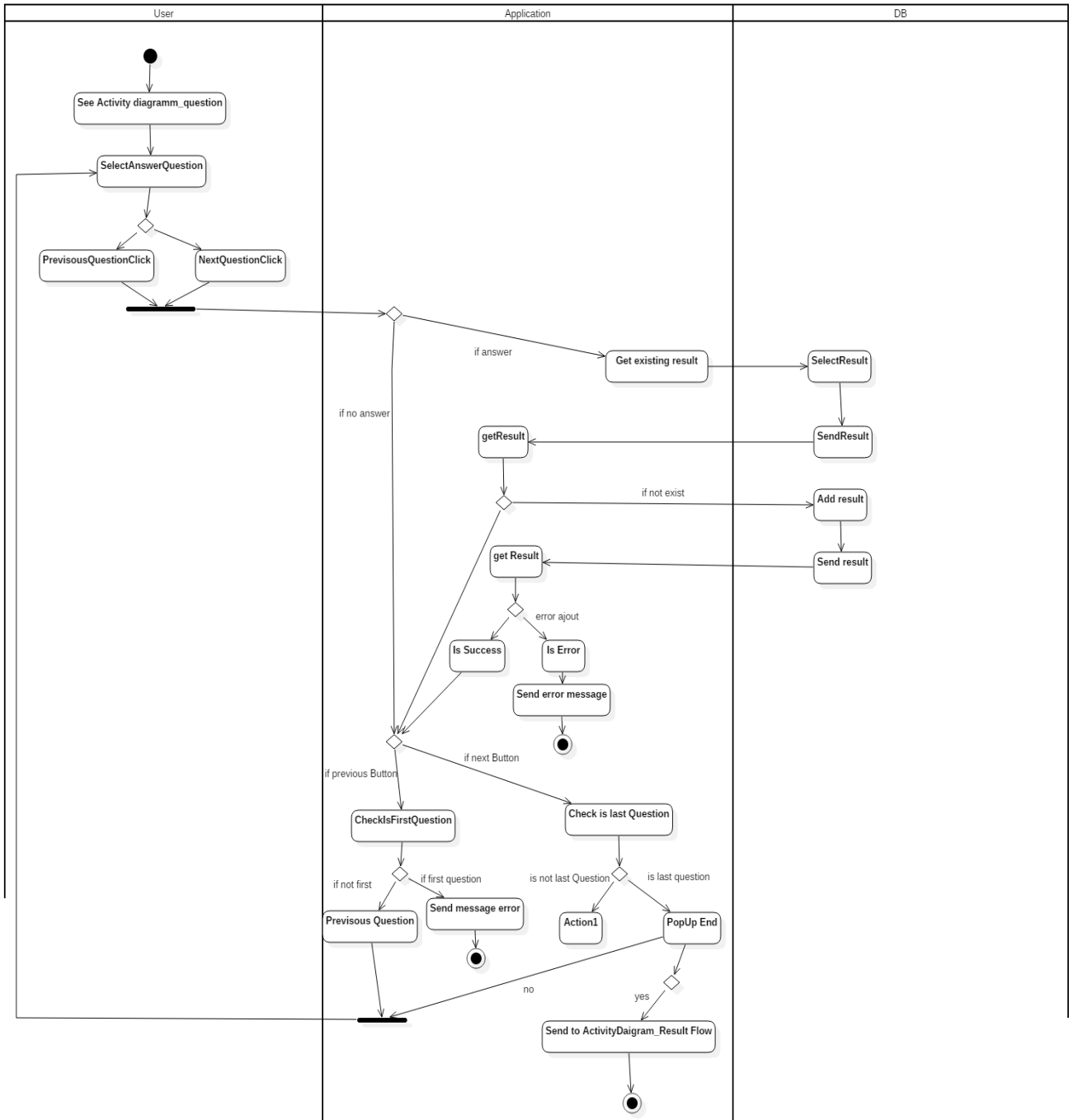
## 6. GESTION DES REPONSES

### MAQUETTE



# L'APPLICATION

DIAGRAMME D'ACTIVITE



# L'APPLICATION

## COMPOSANT A IMPLEMENTER

NOM	TYPE	VALEUR	VERIFICATION	ERREUR
answers_list	list	Liste contenant les réponses de la question		
Next_button	Button	Bouton permettant la navigation entre les questions	Vérifie que ce n'est pas la dernière question	Affiche une fenêtre de fin de QCM
Previous_button	Button	Bouton permettant le retour en arrière dans les questions	Vérifie que ça n'est pas la première question	« Impossible de revenir en arrière »
Ques_title	Text	Intitulé de la question		
Question_value	Text	Champ indique le numéro de la question ainsi que le nombre total de question		

## FONCTION APPELEE

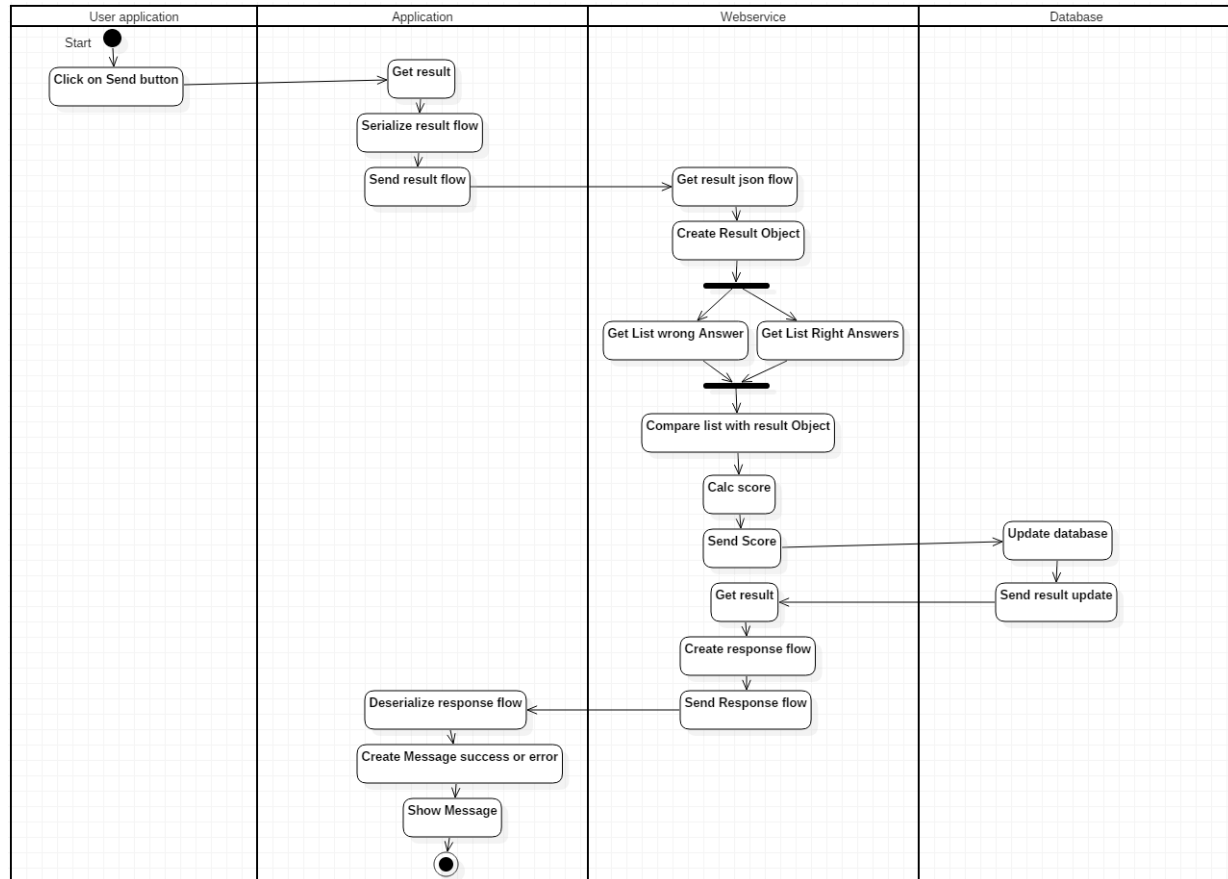
SIGNATURE DE LA FONCTION	PARAMETRES D'ENTREE	UTILITE
SelectedAnswer	Item item	Sélectionne la réponse à la question
NextQuestionClick		Renvoie l'utilisateur sur la question suivante
PreviousQuestionClick	Int numberQuestion	Renvoie l'utilisateur sur la question précédente
CheckIsLastQuestion	Int numberQuestion	Vérifie si c'est la dernière question
CheckIsFirstQuestion	Int numberQuestion	Vérifie si c'est la première question
ShowErrorMessage	Message message	Affiche le message d'erreur



# L'APPLICATION

## 7. GESTION DES RESULTATS

DIAGRAMME D'ACTIVITE



FONCTION APPELEE

NOM	PARAMETRE	RETOUR	ROLE
<b>GetResult</b>	ArrayList<Result>	Liste de résultats sauvegarde en base de données	Requête qui permet la récupération des résultats
<b>IsConnected</b>		Boolean true si le terminal est connecté au réseau	Vérification de la connexion au réseau
<b>OnClick</b>			Récupérer les éléments de la vue avec en paramètre l'id de l'objet
<b>PostResultFlow</b>	ArrayList<Result>		Création du flux de résultat