

# Projet QCM

---

**Documentation technique**

# Contenu

---

I/ Caractéristiques techniques

II/ MVC

III/ Arborescence du projet

IV/ Bonnes pratiques et normes suivies

# Présentation de l'application

---

L'application 'QCM' est une application Java, de type client lourd (SWT, JFace ). Elle permet la gestion et la création de questionnaires à choix multiples pour un organisme de formation, service RH d'entreprise.

Les données de l'application sont enregistrées dans une base de données, permettant ainsi d'assurer leur mémorisation au fil du temps.

L'accès à ces données se fait par l'intermédiaire de l'API RestFul.

# I/ Caractéristiques techniques

---

- Langage de programmation : Java
  - Type de développement : Programmation Orientée Objet (POO)
  - Application de type client lourd : utilisation de la librairie de composants graphiques SWT (Standard Widget Toolkit) et de JFace
  - Plateforme de développement : XAMPP (version 1.8)
  - Système de Gestion de Base de Données : MySQL (version 5.0.11)
  - Serveur Web : Apache (version 2.4.10)
  - Navigateurs web : Google Chrome (version 34); pour l'accès à l'interface d'administration (PHPMyAdmin) de la base de données sous MySQL
  - Environnement de Développement Intégré : Eclipse (version Luna)
-

## II/ MVC

---

L'application utilise le patron de conception MVC (Modèles, Vues, Contrôleurs) afin de séparer les traitements, les données, et la présentation. Ce qui offre un cadre normalisé pour structurer l'application, et facilite le dialogue entre les différents concepteurs.

L'idée est de bien séparer les données, la présentation et les traitements.

## III/ Arborescence du projet

---

Packages de l'application :

Nom du package	Rôle
<b>net.controllers</b>	contient les contrôleurs de l'application
<b>net.gson</b>	contient la classe permettant de traiter les données Json en objet
<b>net.http</b>	contient la classe permettant les requêtes avec l'API Restful.
<b>net.images</b>	contient les images utilisées dans l'application
<b>net.models</b>	contient les classes métier
<b>net.technics</b>	contient des classes techniques (classes utilitaires)
<b>net.vues</b>	contient les vues de l'application

## Classes métier:

Nom	Descriptif
<b>ActifUser</b>	Classe permettant l'instanciation d'un utilisateur connecté
<b>CollectionQuestionnaireGroupe</b>	Classe permettant l'instanciation d'un questionnaire avec son groupe
<b>CollectionUtilisateurScore</b>	Classe permettant l'instanciation d'un utilisateur avec son score
<b>Domaine</b>	Classe permettant l'instanciation d'un domaine
<b>Groupe</b>	Classe permettant l'instanciation d'un groupe
<b>GroupeQuestionnaire</b>	Classe permettant l'instanciation de la CIM groupe_questionnaire
<b>GroupeUtilisateur</b>	Classe permettant l'instanciation de la CIM groupe_utilisateur
<b>Question</b>	Classe permettant l'instanciation d'une question
<b>Questionnaire</b>	Classe permettant l'instanciation d'un questionnaire
<b>Rang</b>	Classe permettant l'instanciation d'un rang
<b>Realisation</b>	Classe permettant l'instanciation de la réalisation d'un qcm
<b>Reponse</b>	Classe permettant l'instanciation d'une reponse
<b>ReponseUtilisateur</b>	Classe permettant l'instanciation de la CIM reponse_utilisateur
<b>Utilisateur</b>	Classe permettant l'instanciation d'un utilisateur

## Classes techniques:

Nom	Descriptif
<b>Http</b>	Classe contenant toutes les fonctions CRUD de l'application
<b>Utils</b>	Méthodes utilitaires (remplissage TableView, Combobox.. )

## Contrôleurs :

Nom	Descriptif
<b>AccueilController</b>	Contrôleur lié à l'accueil
<b>AppController</b>	Classe utilisée lors du lancement de l'application
<b>GroupeController</b>	Contrôleur lié à la gestion des groupes
<b>LoginController</b>	Contrôleur lié à l'authentification
<b>Prog</b>	Programme de lancement de l'application
<b>QcmController</b>	Contrôleur lié à la gestion des questionnaires
<b>StatistiquesController</b>	Contrôleur lié aux statistiques

## IV/ Bonnes pratiques et normes suivies

---

### **Structure de l'application :**

L'application QCM, respectant le patron de conception MVC, doit respecter la structure suivante :

- ⊙ Les classes métiers devront être stockées dans le package net.models
- ⊙ Les classes techniques devront être stockées dans le package net.technics
- ⊙ Les contrôleurs devront être stockés dans le package net.controllers
- ⊙ Les vues dans le package net.vues
- ⊙ Les images dans le package net.images

### **Classes métier :**

- ⊙ Leur nom doit correspondre à une table de la base de données
- ⊙ Leur nom commence par une majuscule

### **Contrôleurs :**

- ⊙ Leur nom sera toujours suffixé par l'extension "Controller", pour montrer qu'il s'agit d'un contrôleur

### **Vues :**

- ⊙ Leur nom commence toujours par un "V" majuscule, permettant ainsi de déterminer rapidement qu'il s'agit d'une vue
- ⊙ On s'efforcera de donner un nom pertinent aux vues, pour éviter de devoir lire le code pour déterminer leur utilité