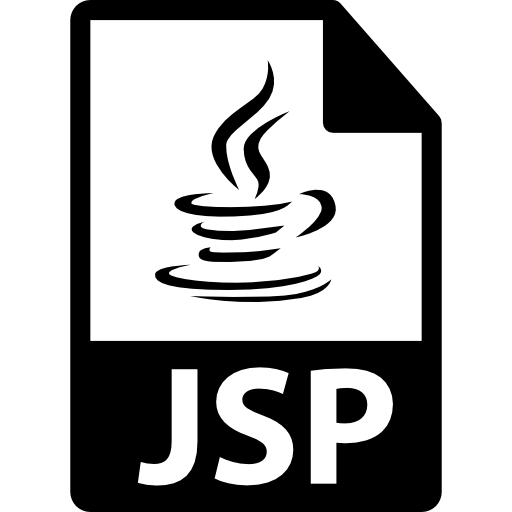
Programmation répartie

Rapport projet



Structuration de code, découplage et injection de dépendance…………………………………….1

Utilisation de Servlet et/ou JSP pour l’implémentation du module de service. ……………….6

Transformation objet-relationnel : comment passe-t-on des objets de l’application Document et Utilisateur aux tables relationnelles de la base de données ?...………………………………7

Variables sessions : ouverture, maintien et fermeture de la session de travail des utilisateurs ?.……………………….………………….8

Concurrence : identification des ressources critiques et la gestion java de la concurrence ? .......9

Efficacité des requêtes d’accès à la base de données : gestion de la connexion à la base de données et de la pré compilation des requêtes …………………………….………………………………….………………….…………………….16

**Problématique :**

**Application:**

L’application est décomposée en trois parties distinctes :

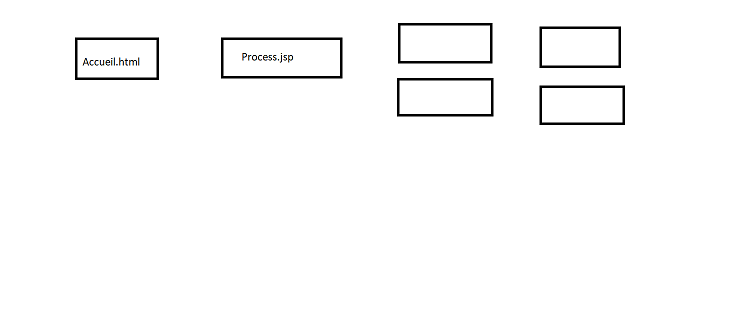
* Une première partie intitulé les services, elle gère tout la partie serveur et contient le code JSP qui est présenté à l’utilisateur.
* Une deuxième partie intitulée mediatek2022.jar, il s’agit d’une bibliothèque qui nous est fournis avec le sujet du projet que l’on doit utiliser pour faire interagir les services et les données, en effet la première et troisième partie son strictement indépendante et nous doivent présenter aucun lien direct. Il nous est strictement interdit de modifier le code de Mediatek2022.jar ainsi cette librairie est directement importer en temps que librairie externe dans le lib de Tomcat.
* Une troisième partie intitulé persistance qui contient les interactions avec la base de données ainsi que la transformation des données java en données sql.

J’ai choisi de crée un module service 100% à base de fichier JSP.

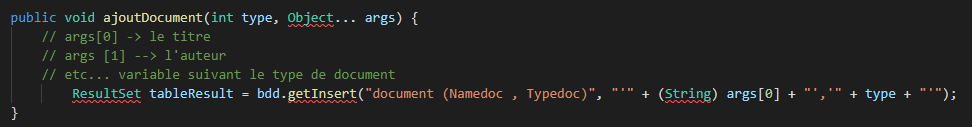
Utilisant des formulaires HTML pour gérer les interactions avec l’utilisateur de l’application t’utilisation de fichier JSP est idéal pour gérer ce double aspect affichage/gestion en arrière-plan de l’interaction.

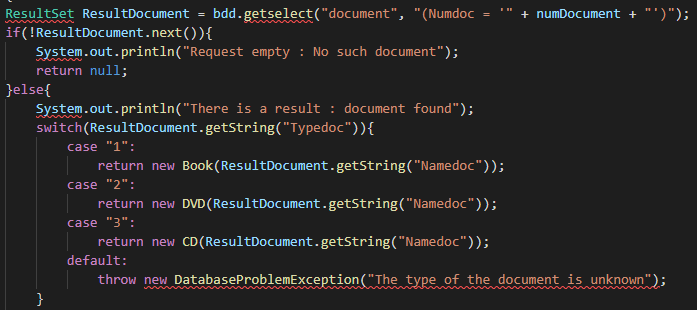
L’utilisateur est accuillis par une première page html qui revoie via l’atribut action de la balise HTML form vers un jsp de traitement qui ensuite redirige l’utilisation et ainsi de suite toute au long du parcour

La structure est donc la suivante :



Pour permettre la persistance des données, il est nécessaire que l’application puisse stoker les objets java crée pendant sont cycle dans la base de données.

Pour transformer un objet java en un champ d’une table les informations nécessaires à l’ajout SQL sont stocké dans les attributs de l’objet comme par exemple ici : 

Dans le cas de la transformation d’une table SQL en un objets JAVA un switch sur le type d’objet permet de crée le bon objet par exemple : 

La partie service utilise des variables sessions pour enregistrer des informations :

* L’utilisateur connecter qui contient tout les documents emprunté et qui est donc mis à jour après chaque emprunt et chaque retour.
* Le login et le mot de passe que l’utilisateur à entrer afin de permettre de l’actualiser(potentiel faille de sécurité, peut être passer par data).

Ces trois variables sont initialiser dans processlogin.jsp en cas de connexion réussi et sont utilisé durant tout le cycle de vie des services.

ffff

End Class

Pour conclure, nous pensons que notre projet est globalement réussi même si nous avons rencontré des difficultés par exemple sur l’interaction avec les cartes. Nous admettons qu’avec plus de temps, nous aurions pu rajouter plus de fonctionnalités (par exemple pour changer le thème des cartes) et peaufiner un peu plus notre projet.