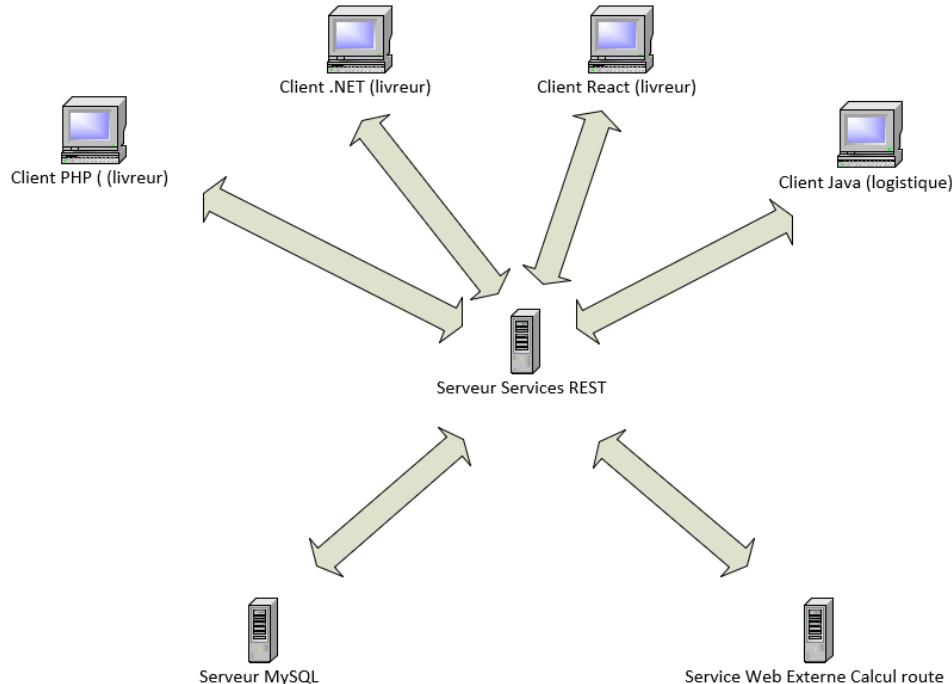


Travail pratique 3 (10%)

Directives :

- Vous devez utiliser un gestionnaire de source pour votre code. Vous devez ajouter votre prof à votre équipe. Adresse du prof : dominique.audet@cegeplimoilou.ca
- Démo **Obligatoire** durant le cours du 12 décembre ou avant par TEAMS.
- Remettre vos projets sous ces noms :
 - A23.5C6.TP3.ServiceREST.XXYY (où XX, YY et ZZ sont vos initiales).
 - A23.5C6.TP3.ClientLogistique.XXYY (où XX, YY et ZZ sont vos initiales).
 - A23.5C6.TP3.ClientLivreurNET.XXYY (où XX, YY et ZZ sont vos initiales).
 - Votre code source PHP.
 - Votre projet React sans le dossier Node_Modules
- Le déploiement du php, de asp.net et de react peut être fait sur un même serveur autre que MySQL et Tomcat. Il doit aussi être dans l'evp.

Schéma du travail :



Mise en situation :

Vous êtes responsable de la conception et du développement d'un système de gestion de calcul de route optimal pour une compagnie de transport. Pour programmer la route optimale, vous

devrez faire appel à un service externe et lui envoyer l'adresse de départ (adresse de l'entrepôt), les adresses de livraison (waypoint) ainsi que l'adresse de destination finale (encore l'entrepôt).

Partie 1 – Serveur MySQL

- Installez et configurez un Windows Serveur 2019.
- Configurez les noms de machine et paramètres réseau.
- Installer MySQL et Workbench sur le serveur.
- Créez un utilisateur via Workbench.
- Accédez MySQL via le serveur avec MySQL Workbench
- Accédez MySQL à partir de la machine du laboratoire avec MySQL Workbench
- Vous devez créer une table qui contient la route optimale. Cette route est remplacée à chaque requête.
- Seul le serveur Service REST communique avec le serveur MySQL.

Partie 2 – Serveur services REST

- Veuillez respecter la nomenclature du projet (ou XX et YY sont les initiales des coéquipiers)
 - o Nom du projet : A23.5C6.TP3.ServiceREST.XXYYZZ.
 - o Nom du package : A23.5C6.TP3.ServiceREST.XXYYZZ.
- Au démarrage du service,
 - o Vous devez lire un fichier JSON (clients.json) qui contient 10 adresses de votre choix. Vous aurez besoin de l'objet JSONArray fourni dans org.json.simple. Vous obtiendrez:
 - Le nom du client et son adresse du client.
 - o Vous devez lire un autre fichier JSON (entrepot.json) qui contient l'adresse de l'entrepôt.
- Voici les services REST demander:
 - o Retourne les adresses des clients au client logistique.
 - o Calcul et modifie la route optimale avec les adresses des clients à livrer qui ont été sélectionnés par le client logistique. La route optimale est enregistrée dans la BD sur le serveur MySQL.
 - Nous aurions pu utiliser l'[API Google Maps Directions](#) qui permet de générer votre route, par contre ce service demande une carte de crédit. **Vous devez trouver OBLIGATOIREMENT une alternative gratuite qui permet de faire la même chose qu'API Google Maps Directions.**
 - o Retourne la route optimale calculée aux livreurs.
- Vous devez logger dans un fichier dans c:\temp l'adresse IP du client lorsqu'une requête POST est faite. Utiliser le logger de Spring Boot et l'intercepteur.
- Vous devez déployer votre service REST en format WAR sur votre serveur Tomcat.

Partie 3 – Client Java (Logistique)

- Voici les fonctionnalités :
 - Demande les adresses des clients au service Rest et charge les adresses des clients dans son interface graphique.
 - L'utilisateur peut demander au service REST de générer la route optimale avec les adresses des clients choisis.
- Doit être déployé en format Jar sur votre Windows 10 dans l'EVP. Aucun IDE doit être installé sur la machine Windows 10.

Partie 4 – Client PHP, .NET et React (livreur)

- Ces clients ont une interface simple qui permet de récupérer la route la plus optimale via le service REST.
- Doivent être déployé **sur une autre machine que votre Windows 10** dans l'EVP.

Partie 5 – Documentation

- Remettre la JavaDoc (auteurs, méthodes, ...) généré en HTML pour le service REST seulement. Olivier confirme que c'est en Java.

Partie 6 – Démo Wireshack Individuellement

Faire une démo avec Wireshack afin de voir une requête GET et une POST avec le contenu de chaque requête.