

## Quiz 1

### Question 1

Appariez le logiciel à sa catégorie

Eclipse	Plugiciel	✓
Angular	Cadriciel	✓
Android	cadriciel, plugiciel et écosystème	✓

### Question 2

Faire correspondre la métrique à la catégorie adéquate, parmi les suivantes:

A- Complexité cyclomatique de McCabe / CC est une métrique de

Complexité ✓ .

B- Nombre d'enfants / NOC est une métrique de Héritage

✓ .

C- Manque de documentation / LOD est une métrique de

Documentation ✓ .

D- Nombre de méthodes / NOM est une métrique de Taille

✓ .

### Question 3

faire glisser la réponse dans l'espace adéquat :

Décomposition d'une fonctionnalité en composantes successives appliquant des transformations incrémentales sur les données est une architecture Modulaire ✓ nommée


pipe-filter ✓ , par contre l'architecture Client-

Serveur est une architecture distribuée ✓ similaire à

Multitier ✓ qui est considérée comme son extension avec plusieurs niveaux.


#### Question 4

« L'application sera disponible comme un service déployé en infonuagique. » cela correspond à :

- ☒ A. Architecture  c'est un composant à un haut niveau.
- ☐ B. Conception

#### Question 5

Single responsibility principle (SRP) : Une classe doit avoir une responsabilité unique, premier principe SOLID est non violé Si :

- ☐ A. Elle présente un Faible couplage seulement.
- ☐ B. Elle présente une haute cohésion seulement
- ☒ C. Elle présente une haute cohésion et un Faible couplage,  par conséquent moins de responsabilités et moins de raisons de changer.
- ☐ D. Toute classe garantit ce principe.

## Quiz 2

### Question 1

Q2 : le Middleware RabbitMQ implémente pratiquement toutes les entités définies dans le standard ISO pour la spécification d'intergiciels orientés messages AMQP. Dans le code RabbitMQ suivant :

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

import java.nio.charset.StandardCharsets;

public class Send {

    private final static String QUEUE_NAME = "hello";

    public static void main(String[] argv) throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("localhost");
        try (Connection connection = factory.newConnection();
            Channel channel = connection.createChannel()) {
            channel.queueDeclare(QUEUE_NAME, false, false, false, null);
            String message = "Hello World!";
            channel.basicPublish("", QUEUE_NAME, null, message.getBytes(StandardCharsets.UTF_8));
            System.out.println(" [x] Sent '" + message + "'");
        }
    }
}
```

Identifiez deux entités du protocole AMQP.

Message, la file d'attente (Queue), producteur

### Question 2

Identifiez dans le code ci-dessous un mauvais patron et proposez la solution possible.

```
public void doneAddProduct(View v,String dir, String Subdir, String SubSubdir, KeyEvent e, WindowEvent e, String Client1,
String Client2, String Client3,String Client4, String Client5, int winLose, int otherCount, int Gain, String winner)
{
    int errorStep = 0;
    EditText product_name = findViewById(R.id.product_name);
    EditText product_description = findViewById(R.id.description);
    EditText date = findViewById(R.id.date);
    nameFinal = product_name.getText().toString();
    descriptionFinal = product_description.getText().toString();
    dateFinal = date.getText().toString();
    /* Checking */
    if (nameFinal.trim().length() < 1) {
        errorStep++;
        product_name.setError("Provide a product name.");
    }
    if (descriptionFinal.trim().length() < 1) {
        errorStep++;
        product_description.setError("Provide a product description.");
    }
    if (dateFinal.trim().length() < 4) {
        errorStep++;
        date.setError("Provide a specific date");
    }
    if (nameFinal.trim().length() > 15) {
        errorStep++;
        product_name.setError("Provide right product name length.");
    }
    String specialCharacters = " !@#$%^&*~.,-./:;<=>?@[]{}~0123456789";
    String[] strCharactersArray = new String(nameFinal.length());
    for (int i = 0; i < nameFinal.length(); i++) {
        strCharactersArray[i] = Character
            .toString(nameFinal.charAt(i));
    }
    //now strCharactersArray[i]=[A, d, i, t, y, a]
    int count = 0;
    for (int i = 0; i < strCharactersArray.length; i++) {
        if (specialCharacters.contains( strCharactersArray[i])) {
            count++;
        }
    }
    if (nameFinal != null && count == 0) {
        errorStep = 0;
    } else {
        errorStep = 1;
    }
    if (errorStep == 0) {
        if (isUpdate) {
            myDB.updateProduct(id, nameFinal, descriptionFinal, dateFinal);
            Toast.makeText(getApplicationContext(), "Product Updated.", Toast.LENGTH_SHORT).show();
        } else {
            myDB.insertProduct(nameFinal, descriptionFinal, dateFinal);
            Toast.makeText(getApplicationContext(), "Product Added.", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Méthode longue -> Extraire des méthodes,

Longue liste de paramètres -> regrouper les paramètres dans un objet,

Switch statements -> remplacer les tests par des sous-classes ou par du polymorphisme.

### Question 3

Les bases de données relationnelles et les bases de données distribuées suivent principalement les propriétés ACID ou Base. Faites la correspondance entre les propriétés et le type de BD dans la liste suivante :

A - BD relationnelle :  ✓ .

B- BD distribuée :  ✓ .

### Question 4

Blockchain repose dans son fonctionnement sur deux éléments importants afin d'assurer la disponibilité et la cohérence des données. Dans la liste ci-dessous sélectionnez ces deux éléments en question :

- ☐ A. Représentation de données dans des tables
- ☒ B. Technologies de registres distribués DTL ✓
- ☐ C. Maintenabilité des nœuds
- ☒ D. Protocole de consensus ✓

## Quiz 3

### Question 1

Citez une différence entre SOAP, REST et RPC.

Les services REST fonctionnent avec seulement 4 méthodes simples CRUD.

RPC peut appeler n'importe quelle méthode (comme un objet).

RPC est presque comme REST mais avec plus de capacité par rapport aux méthodes disponibles, et sans l'organisation imposée des données.

SOAP impose l'utilisation des données organisées en XML et un Web Service Definition Language (WSDL) pour spécifier les données.

### Question 2

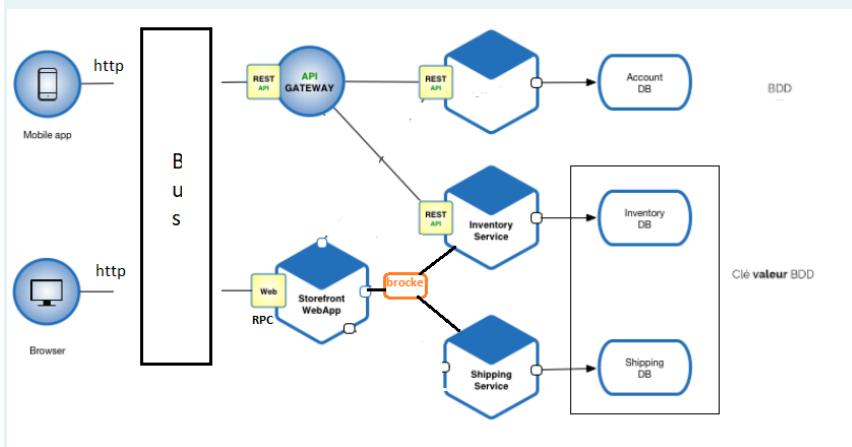
Vous développez un système pour gérer une entreprise de vente en ligne. Le système sera utilisé par les administrateurs, les employés et les clients. Un administrateur peut uniquement ajouter, supprimer ou modifier des produits. Les employés identifient les produits en rupture, gèrent l'inventaires, envoient les produits commandés et aident les clients à réserver un produit non disponible. Un client peut parcourir des catégories de produits, rechercher des items par catégorie, par prix, par description ou par photos, commander un produit ou réserver un produit en rupture de stock.

Concevoir et décrire la ou les architectures pour ce système. Quel est le modèle de données que vous utiliserez? Créez un diagramme UML avec tous les modules du système. Justifiez vos choix en présentant les avantages de votre solution.

Pour un tel système, un service Rest est à utiliser pour l'administrateurs pour ajout, suppression et modification (GET, PUT, POST, DELETE) de produits.

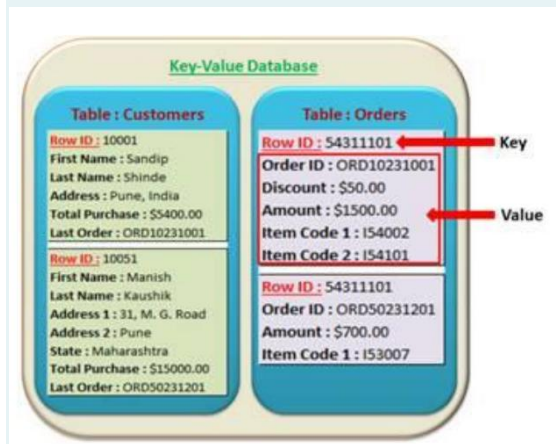
Par contre, pour les autres départements de la compagnie, les services RPC ou SOAP conviendront mieux vu la complexités des méthodes employées tel que rechercher un produit, trier, réserver etc.

L'architecture de mon application sera comme suit :



Utiliser un system de publish/subscribe avec un bus d'évenement pour les accès des clients et les employés pour la gestion des inventaires, la reservation et l'envoi de produits et aussi pour les achats des clients, cette possibilité permet des exécutions asynchrones. Par contre pour les administrateurs de l'application, il ne font que gérer les produits de l'entreprise dans une BD qui sera considéré comme un catalogue, ils accèdent directement à la ressource via des requêtes http Rest.

Le choix de la BD distribuée Clé-valeur permettra d'avoir des attributs différents ou des attributs manquants entre les objets du même type et donnera plus de flexibilité au système.



Avantages de la solution choisie :

Rest offre la simplicité nécessaire pour accéder à une base de données et RPC/SOAP nous donne plus de liberté (actions) avec la base de données.