

Semestre : 1 ☐ 2 ☒Session : Principale ☒ Rattrapage ☐

Module : Architecture Orientée Services "SOA"

Enseignant(s) : UP\_WEB

Classe(s) : 4Arctic1,2-4NIDS

Documents autorisés : OUI ☒ NON ☐

Nombre de pages : 4

Calculatrice autorisée : OUI ☐ NON ☒ Internet autorisée : OUI ☐ NON ☒

Date : 18-05-2018

Heure : 10h30

Durée : 1h30min

**Etude de cas:*****"Participez et remportez avec Natilait un voyage à la Coupe du Monde et encouragez l'équipe Tunisienne en Russie"***

« *Natilait* » est une société spécialisée dans l'industrie laitière, allant de la production du lait jusqu'à la fabrication de ses dérivés: le beurre, le yaourt et les produits frais.

Pour faciliter les échanges entre ses applications, « *Natilait* » dispose de nombreux services parmi lesquels le service gestion de stock.

**I. Partie1: JAX-WS (7.5pts)**

On vise à créer un service web étendu qui gère le stock des produits de l'entreprise.

1. On vous demande de donner le squelette du service web approprié en se basant sur son contrat WSDL donné ci-dessous et tout en utilisant l'API JAX-WS.

```
<wsdl:definitions name="GestionStockService"
targetNamespace="http://stock.natilait/"><wsdl:types><xs:schema elementFormDefault="unqualified"
targetNamespace="http://stock.natilait/" version="1.0">
<xs:element name="afficherAromeProduit" type="tns:afficherAromeProduit"/>
<xs:element name="afficherAromeProduitResponse" type="tns:afficherAromeProduitResponse"/>
<xs:element name="ajouterProduit" type="tns:ajouterProduit"/>
<xs:element name="ajouterProduitResponse" type="tns:ajouterProduitResponse"/>
<xs:element name="arome" type="tns:arome"/>
<xs:element name="modifierProduit" type="tns:modifierProduit"/>
<xs:element name="modifierProduitResponse" type="tns:modifierProduitResponse"/>
<xs:element name="produit" type="tns:produit"/>
<xs:complexType name="afficherAromeProduit">
```

```

<xs:sequence><xs:element minOccurs="0" name="reference"
type="xs:string"/></xs:sequence></xs:complexType>
<xs:complexType name="afficherAromeProduitResponse"><xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="stockResult"
type="tns:arome"/></xs:sequence></xs:complexType>
<xs:complexType name="arome">
<xs:sequence><xs:element minOccurs="0" name="name" type="xs:string"/>
<xs:element name="quantite" type="xs:float"/></xs:sequence>
<xs:attribute name="code" type="xs:string"/></xs:complexType>
<xs:complexType name="ajouterProduit"><xs:sequence>
<xs:element minOccurs="0" name="produit" type="tns:produit"/></xs:sequence></xs:complexType>
<xs:complexType name="produit"><xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="aromes" nillable="true" type="tns:arome"/>
<xs:element minOccurs="0" name="name" type="xs:string"/>
<xs:element name="taille" type="xs:int"/>
<xs:element minOccurs="0" name="type" type="xs:string"/></xs:sequence>
<xs:attribute name="reference" type="xs:string"/>
</xs:complexType><xs:complexType name="ajouterProduitResponse">
<xs:sequence>
<xs:element minOccurs="0" name="stockResult" type="tns:produit"/>
</xs:sequence></xs:complexType>
<xs:complexType name="modifierProduit"><xs:sequence>
<xs:element minOccurs="0" name="reference" type="xs:string"/></xs:sequence></xs:complexType>
<xs:complexType name="modifierProduitResponse"><xs:sequence><xs:element minOccurs="0"
name="stockResult" type="tns:produit"/></xs:sequence></xs:complexType>
</xs:schema></wsdl:types>
<wsdl:message name="ajouterProduitResponse"><wsdl:part element="tns:ajouterProduitResponse"
name="parameters"></wsdl:part></wsdl:message>
<wsdl:message name="afficherAromeProduit"><wsdl:part element="tns:afficherAromeProduit"
name="parameters"></wsdl:part></wsdl:message>
<wsdl:message name="modifierProduitResponse"><wsdl:part element="tns:modifierProduitResponse"
name="parameters"></wsdl:part></wsdl:message>
<wsdl:message name="afficherAromeProduitResponse"><wsdl:part
element="tns:afficherAromeProduitResponse" name="parameters"></wsdl:part></wsdl:message>
<wsdl:message name="ajouterProduit"><wsdl:part element="tns:ajouterProduit" name="parameters">
</wsdl:part></wsdl:message>

```

```

<wsdl:message name="modifierProduit"><wsdl:part element="tns:modifierProduit" name="parameters">
  </wsdl:part></wsdl:message>
<wsdl:portType name="GestionStock">
  <wsdl:operation name="afficherAromeProduit"><wsdl:input message="tns:afficherAromeProduit"
name="afficherAromeProduit"></wsdl:input>
<wsdl:output message="tns:afficherAromeProduitResponse" name="afficherAromeProduitResponse">
  </wsdl:output></wsdl:operation>
<wsdl:operation name="ajouterProduit"><wsdl:input message="tns:ajouterProduit"
name="ajouterProduit"></wsdl:input>
<wsdl:output message="tns:ajouterProduitResponse" name="ajouterProduitResponse">
  </wsdl:output></wsdl:operation><wsdl:operation name="modifierProduit">
<wsdl:input message="tns:modifierProduit" name="modifierProduit">
  </wsdl:input><wsdl:output message="tns:modifierProduitResponse" name="modifierProduitResponse">
  </wsdl:output></wsdl:operation></wsdl:portType>
.....
</wsdl:definitions>

```

## II. Partie2: JAX-RS (4.25pts)

Le but de toute entreprise est de maximiser les ventes et réaliser des profits. La publicité assure que les ventes et le chiffre d'affaires de l'entreprise augmentent. Pour cela et à l'occasion de la coupe du monde, « *Natilait* » a organisé un jeu nommé « *Be The Coach* » dédié à sa clientèle dont le but est de collecter les codes des joueurs de l'équipe nationale. Ceci permet aux participants d'entrer dans le tirage au sort pour gagner un voyage à Russie.

On se propose de mettre en place un service web RESTful de gestion des participants.

Le service web RESTful fournit aux participants du jeu plusieurs fonctionnalités.

2. Développez le service web RESTful et testez-le en se basant sur les descriptions suivantes:

- **Vérifier l'existence d'un participant:**

Requête	GET	jeu/codes/aa@gmail.com/aa
	Status	202 Accepted / 406 Not Acceptable
Réponse	Media Type	Text/plain
	Body	True / False

- Afficher un joueur selon le *code* approprié:

Requête	PUT	jeu/codes?code=aze
	Status	200 OK / 409 Conflict
Réponse	Body	{ "code": "aze", "nom": "Msakni", "prenom": "Youssef" }

### III. Partie3: BPMN « Be the coach » (8.25pts)

Pour assurer le bon déroulement du jeu, on vise à créer un processus métier représentant un enchaînement d'activités.

Vous trouvez ci-dessous la description du déroulement du processus avec tous les cas et chemins possibles.

Nous supposons que le participant au jeu a trouvé un code en grattant la couverture d'une bouteille de yaourt à boire Natilait.

- 1- Notre processus est nommé « **Be The Coach** » dont l'acteur intervenant est le **participant**.
- 2- Le **participant** commence le processus.
- 3- Le **participant** saisit l'URL pour accéder au jeu.  
-Si la valeur de l'URL est égale à **www.natilaitbethecoach.com**, le client doit se connecter en saisissant son email et son password.  
-Si non, le processus s'arrête.
- 4- Le service web Restful est invoqué afin de vérifier l'existence du participant.  
-Si le participant existe, Il saisit le **code** trouvé.  
-Si non le processus s'arrête.
- 5- Après avoir saisi le **code**, la deuxième méthode du même service Rest est invoquée pour afficher le joueur approprié.
- 6- Le **participant** consulte le nom et le prénom du joueur retourné. En parallèle, le message suivant s'affiche au participant: **Merci d'avoir participé à notre jeu. Bonne chance.**
- 7- Le processus se termine.

3. Modélisez et Exécutez le processus métier.

**NB :**

- Pas besoin de créer un compte utilisateur. Vous pouvez utiliser l'un des comptes par défaut:  
**participant=kermit**

**Bon Travail**