

I- Specifications fonctionnelles:

L'application consiste à réaliser un site web qui permettra de gérer les réservations des chambres dans les différents hôtels d'une ville. Elle permettra:

- de vérifier la liste des hôtels d'une ville;
- de réserver les chambres nécessaires.

II- Specifications techniques:

la conception de cette application fera appel à un environnement technique de développement composé comme suit:

Pour le back end:

- IDE: IntelliJ , Lombok
- Langage: Java 1.8
- Framework: Spring
- SGBD: MySQL
- Terminal
- Microservice (Controller Rest) ou Rest Repository

Pour le front end:

- Ide: IntelliJ
- Terminal
- Framework: Angular, Bootstrap, JQuery

SPECIFICATION GENERALE

Dans le prolongement du premier sprint, vous devez dans ce second sprint, réaliser une application de vente en ligne complète qui s'appuie sur vos acquis précédents mais il faudra, cette fois-ci, refaire en groupe un projet complet, back + front en répondant aux spécifications fonctionnelles ci-dessous.

L'appli propose d'afficher tous les produits classés par catégories accessibles via notre api, d'en sélectionner un ou plusieurs pour constituer un caddy avec la possibilité de passer commande à tout moment à condition d'être authentifié, enfin de "régler en ligne".

Vous devez distinguer 3 rôles dans votre appli :

- utilisateur non connecté ou individu lambda qui peut naviguer sur votre site et consulter les infos des produits proposés par catégorie
- utilisateur connecté donc authentifié qui peut réaliser un panier constitué d'un ou plusieurs produits puis passer commande
- administrateur qui peut changer les photos des produits, modifier les infos.

SPECIFICATIONS TECHNIQUES / CONCEPTION

- Spring au niveau du back (java 8, api Rest)
- Vous pourrez dans un premier temps utiliser une bdd virtuelle mais il faudra in fine utiliser un sgbd type mysql ou mariadb une fois arrivé à la phase commande.
- Comme à notre habitude, ne réinventons pas la roue aussi utiliser Spring data pour gérer vos entités Jpa une fois créées puis spring data rest pour la gestion des web services associés
- De même, laisser de côté spring security que vous devrez naturellement gérer en fin de dev
- Angular 8 pour le front (bootstrap3, jquery)
- Intégration continue avec Git d'un workflow par équipe, 2 repos (back & front)
- Gestion des photos : elles ne seront pas stockées en base pour ne pas alourdir celle-ci, chaque produit aura un attribut qui pointe sur le fichier correspondant (chemin)
- Pour des raisons de sécurité, ne pas stocker les photos dans le rep static, l'accès sera définitivement verrouillé une fois spring security activé !
- la gestion de l'authentification se fera côté front avec des données fictives locales (tableau avec noms des utilisateurs et leurs rôles : USER et(ou) ADMIN) "3ème Sprint : il faudra le gérer côté back"
- L'utilisateur ou token (user + roles) sera stocké en local storage afin de permettre la navigation sans déconnexion
- "3ème sprint : Mise en oeuvre d'https au niveau du back (changer application.properties, keytool...)"
- "3ème sprint : JWT pour la phase d'authentification / utiliser keycloak"