

Cahier des charges

Plateforme de gestion de Stock

Il s'agit de développer une application web permettant la gestion de stock d'un magasin. L'application doit présenter les besoins fonctionnels suivants :

BF1- La gestion des fournisseurs.

BF2- La gestion des produits.

BF3- Un espace pour l'acteur « **SUPERADMIN** » permettant d'ajouter des utilisateurs (ADMIN, AGENT).

BF4- Un espace pour l'Acteur « **ADMIN** » permettant de gérer les fournisseurs et les produits.

BF5- Un espace pour l'Acteur « **AGENT** » permettant de gérer seulement les produits.

BF6- Une messagerie permettant d'assurer la communication entre les différents acteurs.

Exemples :

1-Lors de la création d'un nouveau compte « Admin » par le « SUPERADMIN », un nouvel email sera envoyé automatiquement à l'acteur en question.

2-Les acteurs « ADMIN » ou « AGENT » peuvent envoyer des messages à l'acteur « SUPERADMIN »

1-Une première version de l'application sera à développer totalement avec le Framework **SPRING BOOT** puis déployer sur le **Cloud Heroku**

2-Une deuxième version de l'application sera à développer avec les Frameworks **SPRING BOOT** et **ANGULAR 11**, puis déployer sur le **Cloud Heroku**

3-Une troisième version de l'application sera à développer avec le Framework **Mobile IONIC**, permettant d'avoir une version mobile de l'application Android et iOS

Technologies utilisées

- Frameworks: Spring Boot – Angular - IONIC
- Langages : Java – Typescript – HTML5 - CSS - SQL
- IDE : STS, VS Code

Version 1 du Projet : Application Spring MVC(Thymeleaf)

SPRINT 1: CRUD et Intégration de maquette

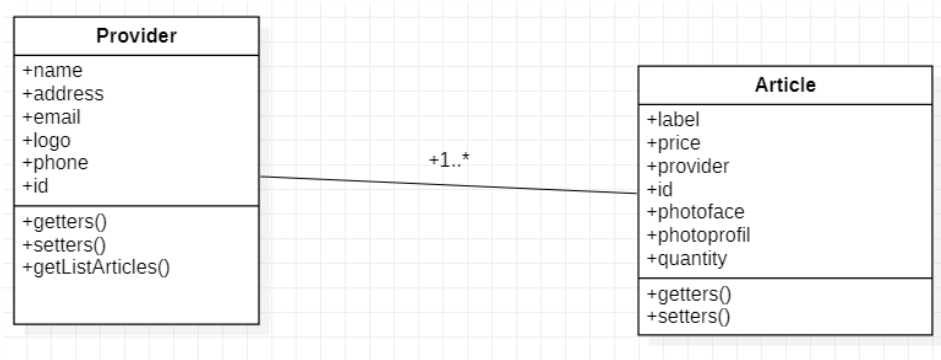


Fig. 1 : Diagramme des classes du SPRINT 1

- ✓ **Task 1 [Intégration Maquette Bootstrap]:** Créer un nouveau projet intitulé *ams* et intégrer la maquette html se trouvant sur le lien : http://drive.smart-it-partner.com/folder/136/Maquette_ams
- ✓ **Task 2 [CRUD]:** Développer en se basant sur cette maquette les opérations CRUD associé aux entités Provider et Article.
- ✓ **Task 3 [Upload Image]:** Ajouter *un champ logo* dans l'entité Provider et *deux champs photoFront et photoProfil* pour l'entité Article.
- ✓ **Task 4 [Navigation Bidirectionnelle] :** Développer les deux méthodes permettant d'afficher la liste des produits d'un Provider
- ✓ **Task 5 [HQL]:** Utiliser HQL afin de développer deux méthodes de recherche d'un Article ou d'un Provider par son libellé.
- ✓ **Task 6 [JUNIT5]:** Développer les tests unitaires relatives aux deux couches MVC et Data.
- ✓ **Task 7 [Bonus]:** Développer des méthodes permettant de calculer et d'afficher des statistiques tel que le nombre de Providers, nombre d'articles, Articles épuisés.