# Gestion de cave à vin :

Créer une application web intégrant une Api REST Java, ayant pour but de gérer une cave à vin.

# Classes / Services :

**Bouteille**

*Panel d’administration pour modifier les propriétés d’une bouteille*

*Panel d’administration pour modifier les propriétés d’une bouteille*

Photo

Nom

Année

Région

Couleur

Cépage

**Casier :**

Photo

**Recherche :**

Recherche par année

Recherche par région

**Schéma :**

B2

B1

B1

*3*

*5*

C1

**Fonctionnalités :**

- Interface utilisateur :

- La connexion (Pseudo, mot de passe)

- La déconnexion

- Création d’une bouteille

- Création d’un casier (privé, visible uniquement par lui-même)

- Ajout d’une bouteille à un casier

**- Interface administrateur :**

- Création d’une bouteille

- Modification une bouteille

- Visualisation d’un casier

# Ecrans :

Ecran d’accueil (mode non connecté)

Ecran d’inscription

Ecran de connexion

Ecran de récupération de mot de passe

Ecran d’accueil (mode connecté)

Ecran de création d’un casier

Ecran de gestion d’un casier

Ecran de déplacement d’une bouteille

Ecran d’ajout d’une bouteille

Ecran de modification d’une bouteille++++

Ecran d’accueil (mode administrateur)

# Fonctionnalités pour l’utilisateur :

- Ajouter une bouteille au catalogue

- Rechercher une bouteille : l’utilisateur tape le début du nom de la bouteille, et toutes les bouteilles matchant avec ce nom et présentes dans le catalogue s’affichent (flexdatalist). L’utilisateur clique sur ce nom, et le système affiche tous les caves avec entre parenthèses le nombre de vins correspondant à la recherche.

En cliquant sur une cave, l’utilisateur peut modifier sa cave.

- A côté de chaque vin résultat, l’utilisateur peut faire (+1) ou (-1) pour incrémenter ou décrémenter le nombre de bouteilles.

# Services :

- Base de données afin de stocker les comptes, les bouteilles, et les caves

- Les bouteilles du catalogue sont conservées dans le cache, de même pour les caves

POST /subscribe

GET /login

GET /logout

PUT /changePassword

POST /user/bottle

GET /user/bottles

GET /user/bottle/{id}

**Recherche de bouteilles :**

GET /user/bottle/

(Envoie le nom de la bouteille en données)

GET /user/bottle/

(Envoie le nom de la cave en données)

PUT /user/stock/{id}/add

PUT /user/stock/{id}/remove

POST /user/cellar

GET /user/cellars

GET /user/cellar/{id}

**Brainstorming :**

Stock :

Bouteille\_ID

Value

Cave :

List< Stock >() : contient tous le stock des bouteilles

Bouteille :

ID : id de la bouteille

Installer prjt : mvn clean install tomcat7:run

Lancer REST : mvn tomcat7:run

To do : Finir les services REST

/user

/cave ?user=x

/bottles ?user=x&cave=y

Clean REST Api :

## User :

GET **api/user** : tous les utilisateurs

GET **api/user/{id}** : un utilisateur spécifique

## Cellar :

GET **api/cellar?user={id}** : les caves d’un utilisateur spécifique

POST **api/cellar/add?user={id}** : ajoute une cave à un utilisateur

*{*

*"label" : "…",*

*"description" : "…"*

*}*

DELETE **api/cellar/remove?user={id}&cellar={id’}** : supprime une cave précise

## Bottle :

GET **api/bottle** : toutes les bouteilles enregistrées

GET **api/bottle/{id}** : une bouteille spécifique

GET **api/bottle?user={id}&cellar={id’}** : récupère les bouteilles d’une cave (Slot)