



# User Stories et spécifications techniques

## Application Web Dressing

Léa DELANNAY  
29/07/2019

# Table des matières

I. Features .....	2
II. User stories .....	2
A. Enregistrement .....	2
B. Connexion .....	4
C. Accueil et barre de navigation .....	5
D. Création d'un vêtement .....	6
E. Liste de tous les vêtements .....	9
F. Détail d'un vêtement .....	11
G. Modification d'un vêtement .....	11
H. Suppression d'un vêtement .....	15
I. Création d'une tenue .....	16
J. Liste de toutes les tenues .....	16
K. Détail d'une tenue .....	16
L. Modification d'une tenue .....	16
M. Suppression d'une tenue .....	16
N. Affichage de la météo .....	16

## I. Features

- 1- Enregistrement
- 2- Connexion
- 3- Accueil et barre de navigation
- 4- Création d'un vêtement
- 5- Liste de tous les vêtements
- 6- Détail d'un vêtement
- 7- Modification d'un vêtement
- 8- Suppression d'un vêtement
- 9- Création d'une tenue
- 10- Liste de toutes les tenues
- 11- Détail d'une tenue
- 12- Modification d'une tenue
- 13- Suppression d'une tenue
- 14- Affichage de la météo
- 15- Génération d'une tenue

## II. User stories

### A. Enregistrement

<b>User story A.1 : Une personne souhaite s'enregistrer pour utiliser l'application</b>
DOR : Le site MonDressing existe, le projet a été initialisé
DOD : J'accède à la page de création d'un compte
<ul style="list-style-type: none"><li>• En tant que personne n'ayant pas encore de compte.</li><li>• J'accède au site MonDressing.</li><li>• Je clique sur le bouton « S'enregistrer » dans la barre de navigation. Au clic, la couleur de fond du bouton « S'enregistrer » devient plus claire afin d'indiquer quel bouton vient d'être cliqué.</li><li>• J'accède au formulaire permettant de créer un compte.</li><li>• La page correspond à la maquette 1 en annexe.</li></ul>
<b>Spécifications techniques</b>
<ul style="list-style-type: none"><li>• Le front est fait en Angular 6.</li><li>• Le back est fait en NodeJs et Express.</li><li>• Le site a été préalablement initialisé avec l'outil Angular CLI.</li><li>• L'URL de la page est <a href="http://localhost:4200/register">http://localhost :4200/register</a>.</li><li>• La partie « register » est un composant Angular du module Angular « account ».</li></ul>

<b>User story A.2 : Contrôle sur l'enregistrement d'une personne : Vérifier que tous les champs obligatoires sont correctement remplis</b>
DOR : Le formulaire permettant de s'enregistrer existe
DOD : Je gère le cas des champs obligatoires sur le formulaire
<ul style="list-style-type: none"><li>• En tant que personne n'ayant pas encore de compte.</li><li>• Lorsque je complète le formulaire d'enregistrement.</li><li>• Si je clique sur le bouton « S'ENREGISTRER » alors que je n'ai pas rempli les trois champs obligatoires, celui-ci est désactivé.</li></ul>

- Le champ « Adresse email » nécessite un @. Si j'envoie le formulaire sans @, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut du formulaire.
- Lorsque je saisis un mot de passe, le champ « Mot de passe » n'affiche pas le mot de passe en clair.
- Mon mot de passe doit contenir des chiffres, des lettres, et contenir au minimum 6 caractères. Si ce n'est pas le cas, le message « Votre mot de passe doit contenir des chiffres, des lettres, et contenir au minimum 6 caractères » apparaît en dessous du champ concerné. Le bouton « S'ENREGISTRER » est désactivé tant que tout n'est pas conforme.
- Si je clique en dehors de n'importe quel champ du formulaire, après avoir cliqué dedans, et que le champ est vide, le message « Vous n'avez pas rempli ce champ » apparaît en dessous du champ concerné, surligné en rouge clair (#F8d7da). Le message reste affiché tant que les champs n'ont pas été remplis.
- Si je saisis des espaces avant ou après mon pseudo, mon login ou mon mot de passe, et/ou si je saisis un « ; », ceux-ci sont supprimés lors de l'envoi du formulaire avant d'être transmis au serveur.
- Si tout est correctement complété, l'appel au service concernant l'enregistrement des données se fait.

#### Spécifications techniques

- Les vérifications sont faites avec les directives structurales Angular.
- L'URL à appeler pour soumettre l'enregistrement de l'utilisateur est : <http://localhost:3000/api/users>.
- On doit soumettre les informations en POST au format JSON
- L'objet JSON est structuré ainsi :  
PSEUDO\_USER : string, LOGIN\_USER : string, MDP\_USER : string  
Ces données sont obligatoires.
- Pour que les données qui transitent sur le réseau soient chiffrées, un certificat https doit être installé sur le serveur afin que le front et le back doivent communiquer via le protocole https.

#### User story A.3 : Contrôle sur l'enregistrement d'une personne : Vérifier que l'inscription en base a bien été effectuée

DOR : Les US A.1 et A.2 ont bien été effectuées

DOD : Je deviens un utilisateur enregistré en me servant du formulaire d'enregistrement pour créer un compte

- En tant que personne n'ayant pas encore de compte.
- J'ai complété le formulaire d'enregistrement.
- Je clique sur le bouton « S'ENREGISTRER ».
- Si une erreur survient lors de l'envoi des données pour l'enregistrement en base, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut du formulaire.
- Lorsque l'enregistrement s'est correctement déroulé, la phrase « Votre compte a bien été créé. Merci de vous connecter » apparaît en haut du formulaire, puis je suis redirigé vers la page de connexion au bout de trois secondes.

#### Spécifications techniques

- Le serveur a été initialisé avec NodeJs et Express.
- La base de données MySQL nommée « dressing » a été créée et possède la table

*user* contenant les champs dont le détail figure dans le dictionnaire des données ci-joint.

- La dépendance mysql en version 2.17.1 est utilisée pour la connexion entre le serveur et la base de données.
- Le serveur renvoie le code http 400 « Bad Request » s'il y a une erreur dans la requête formulée, et le code http 200 « OK » si tout s'est correctement déroulé.
- Des requêtes préparées doivent être utilisées afin de se protéger des injections SQL.
- La route /api/users doit être créée afin que le serveur puisse récupérer et traiter les informations que le front lui envoie.

## B. Connexion

### User story B.1 : Un utilisateur souhaite se connecter à l'application

DOR : Le site MonDressing existe

DOD : Je parviens à me connecter à MonDressing

- En tant qu'utilisateur.
- J'accède au site MonDressing et je suis directement dirigé sur le formulaire permettant de me connecter.
- Tant que je ne suis pas connecté, quelle que soit l'URL à laquelle j'essaie d'accéder (à part register), je suis automatiquement renvoyé vers le formulaire de connexion. La seule page à laquelle j'accède en plus de la page de connexion est la page d'enregistrement.
- Je n'ai accès qu'aux formulaires de connexion et d'enregistrement.
- Lorsque je clique sur le bouton « Se connecter » dans la barre de navigation, la couleur de fond du bouton « Se connecter » devient plus claire afin d'indiquer quel bouton vient d'être cliqué.
- La page correspond à la maquette 2 en annexe.

#### Spécifications techniques

- L'URL de la page est http://localhost :4200/login.
- La partie « login » est un composant Angular du module Angular account.
- Afin de protéger les pages de l'utilisation par une personne non connectée, les guards Angular sont mis en place.

### User story B.2 : Contrôle sur la connexion d'un utilisateur : Vérifier que tous les champs obligatoires sont correctement remplis

DOR : Le formulaire permettant de se connecter existe

DOD : Je gère le cas des champs obligatoires sur le formulaire

- En tant qu'utilisateur.
- Lorsque je complète le formulaire de connexion.
- Si je clique sur le bouton « SE CONNECTER » alors que je n'ai pas rempli les deux champs obligatoires, celui-ci est désactivé.
- Lorsque je saisis un mot de passe, le champ « Mot de passe » n'affiche pas le mot de passe en clair.
- Si je clique en dehors de n'importe quel champ du formulaire, après avoir cliqué dedans, et que le champ est vide, le message « Vous n'avez pas rempli ce champ » apparaît en dessous du champ concerné, surligné en rouge clair (#F8d7da). Le message reste affiché tant que les champs n'ont pas été remplis.

- Si je saisis des espaces avant ou après mon login ou mon mot de passe, et/ou si je saisis un « ; », ceux-ci sont supprimés lors de l'envoi du formulaire avant d'être transmis au serveur.
- Si tout est correctement complété, l'appel au service concernant la connexion se fait.

#### Spécifications techniques

- Les vérifications sont faites avec les directives structurales Angular.
- L'URL à appeler pour soumettre l'enregistrement de l'utilisateur est : <http://localhost:3000/api/users/login>.
- Les informations doivent être soumises en POST au format JSON.
- L'objet JSON est structuré ainsi :  
LOGIN\_USER : string, MDP\_USER : string  
Ces données sont obligatoires.

#### User story B.3 : Contrôle sur la connexion d'un utilisateur : Vérifier que la vérification en base a bien été effectuée

DOR : Les US B.1 et B.2 ont bien été effectuées

DOD : Je gère le cas de la vérification en base d'un utilisateur

- En tant qu'utilisateur.
- J'ai complété le formulaire de connexion.
- Je clique sur le bouton « SE CONNECTER ».
- Vérifier que les données existent en base dans la table *user*.
- Comparer le login et le mot de passe saisi par l'utilisateur à ceux stockés en base.
- Si une erreur survient lors de l'envoi des données pour la vérification en base, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut du formulaire.
- Si le login ou le mot de passe ne correspondent pas à ceux stockés en base, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut du formulaire.
- Lorsque la vérification s'est correctement déroulée, je suis redirigé vers la page d'accueil de l'application.

#### Spécifications techniques

- Les mots de passes sont hashés (SHA256) avant d'être enregistrés en base.
- Le serveur renvoie le code http 403 « Forbidden » si les identifiants sont incorrects, et le code http 200 « OK » accompagné d'un token Json Web Token (JWT) contenant l'id de l'utilisateur si tout s'est correctement déroulé.
- Des requêtes préparées doivent être utilisées afin de se protéger des injections SQL.
- La route /api/users/login doit être créée afin que le serveur puisse récupérer et traiter les informations que le front lui envoie.

### C. Accueil et barre de navigation

#### User story C.1 : Un utilisateur arrive sur la page d'accueil et la barre de navigation

DOR : L'utilisateur est connecté

DOD : Je viens de me connecter à MonDressing et la page d'accueil apparaît

- En tant qu'utilisateur.
- Je me connecte au site MonDressing et je suis directement dirigé sur la page d'accueil.

- Si je saisis une URL erronée, je suis redirigé vers la page d'accueil. Si je ne suis pas connecté, je suis redirigé vers le formulaire de login.
- La barre de navigation correspond à la maquette 15 en annexe.
- J'ai accès à la barre de navigation complète depuis toutes les pages nécessitant une connexion.
- Lorsque je clique sur le logo MonDressing, je suis dirigé sur la page d'accueil.
- La page correspond à la maquette 3 en annexe.

#### Spécifications techniques

- L'URL de la page est [http://localhost :4200](http://localhost:4200) ou [http://localhost :4200/homepage](http://localhost:4200/homepage).
- La partie « accueil » est un composant Angular du module Angular app.module.

### D. Création d'un vêtement

#### User story D.1 : Un utilisateur souhaite créer un vêtement

DOR : L'utilisateur est connecté

DOD : J'accède à la page de création d'un vêtement

- En tant qu'utilisateur.
- Je me connecte au site MonDressing et je suis directement dirigé sur la page d'accueil.
- Je clique dans la barre de navigation sur le bouton « Création », puis sur le bouton « Ajouter un vêtement ».
- La page qui s'affiche correspond à la maquette 4 en annexe.
- Par défaut, l'image qui apparaît dans l'encadré dédié est l'image de la maquette.
- Les données renseignées dans les listes déroulantes et les cases à cocher des éléments Catégorie, Caractéristiques, Marque, Couleurs, Occasions et Notes sont récupérées dynamiquement depuis la base de données.

#### Spécifications techniques

- L'URL de la page est [http://localhost :4200/clothe-create](http://localhost:4200/clothe-create).
- La partie « clothe-create » est un composant Angular du module Angular clothes.
- Les appels doivent être soumis en GET au format JSON.
- L'URL à appeler pour obtenir la liste des marques est : <http://localhost:3000/api/brands>.
- L'URL à appeler pour obtenir la liste des catégories est : <http://localhost:3000/api/categories>.
- L'URL à appeler pour obtenir la liste des vêtements est : <http://localhost:3000/api/colors>.
- L'URL à appeler pour obtenir la liste des caractéristiques est : <http://localhost:3000/api/features>.
- L'URL à appeler pour obtenir la liste des notes est : <http://localhost:3000/api/notes>.
- L'URL à appeler pour obtenir la liste des occasions est : <http://localhost:3000/api/occasions>.

#### User story D.2 : Contrôle sur la création d'un vêtement : Vérifier que tous les champs obligatoires sont correctement remplis

DOR : Le formulaire permettant de créer un vêtement existe

DOD : Je gère le cas des champs obligatoires sur le formulaire

- En tant qu'utilisateur.
- Lorsque je complète le formulaire de création d'un vêtement.
- Si j'essaie de cliquer sur le bouton « Enregistrer » alors que je n'ai pas rempli tous les champs obligatoires, je ne peux pas car celui-ci est désactivé.
- Les champs obligatoires sont marqués par un astérisque et la phrase « \* Tous ces champs sont obligatoires. Vous pourrez les modifier ultérieurement. » est affichée en bas de page entre l'image et le bouton « Enregistrer ».
- Les champs obligatoires sont : « Nom », « Catégorie », « Marque », « Couleurs », « Caractéristiques », « Occasions », Description », et « Note ».
- Si je clique en dehors de n'importe quel champ du formulaire, après avoir cliqué dedans, et que le champ est vide, le message « Vous n'avez pas rempli ce champ » apparaît en dessous du champ concerné, surligné en rouge clair (#F8d7da). Le message reste affiché tant que les champs n'ont pas été remplis.
- Si je saisis un nom de vêtement qui existe déjà, le message « Ce nom de vêtement existe déjà » apparaît en dessous du champ concerné.
- Par défaut, les champs de saisie de texte et les listes déroulantes ont les extrémités rouges. Lorsque le champ est renseigné, les extrémités deviennent vertes.
- Si tout est correctement complété, l'appel au service concernant la création d'un vêtement se fait.

#### Spécifications techniques

- Les vérifications sont faites avec les directives structurales Angular.
- L'URL à appeler pour soumettre la création d'un vêtement est :  
<http://localhost:3000/api/clothes>.
- Les informations doivent être soumises en POST au format JSON.
- L'objet JSON est structuré ainsi :  
NOM\_VET : string, FK\_ID\_CAT : number, FK\_ID\_MARQUE : number,  
FK\_ID\_NOTE : number, DESCRIPT\_VET : string, ID\_CHARACTER : any[],  
ID\_COUL : any[], ID\_OCCAS : any[], IMG\_VET : any, FK\_ID\_USER : number.  
Ces données sont obligatoires.
- L'URL à appeler pour obtenir la liste des noms de vêtements est :  
<http://localhost:3000/api/clothes/clothesname>.
- L'URL à appeler pour obtenir la liste des noms de marques est :  
<http://localhost:3000/api/brands/brandname>.
- L'URL à appeler pour obtenir la liste des noms de catégories est :  
<http://localhost:3000/api/categories/categoryname>.
- L'URL à appeler pour obtenir la liste des noms de couleurs est :  
<http://localhost:3000/api/colors/colorname>.
- L'URL à appeler pour obtenir la liste des noms de caractéristiques est :  
<http://localhost:3000/api/features/featurename>.
- L'URL à appeler pour obtenir la liste des noms d'occasions est :  
<http://localhost:3000/api/occasions/occasionname>.

#### User story D.3 : Un utilisateur souhaite renseigner une catégorie, marque, couleur, caractéristique ou occasion qui n'est pas dans la liste

DOR : L'utilisateur est sur la page contenant le formulaire de création d'un vêtement

DOD : Je renseigne une catégorie, marque, couleur, caractéristique ou occasion qui s'ajoute à la liste

- En tant qu'utilisateur.



- La catégorie, marque, couleur, caractéristique ou occasion que je souhaite renseigner n'est pas proposée dans la liste.
- Je clique sur le lien en dessous du champ concerné « *catégorie, marque, couleur, caractéristique ou occasion* inexistante dans la liste ».
- Un champ texte apparaît permettant de renseigner la catégorie, marque, couleur, caractéristique ou occasion que je souhaite ajouter.
- Lorsque je clique sur le bouton + situé à côté du champ, la catégorie, marque, couleur, caractéristique ou occasion s'ajoute à la liste et le champ présentant la liste de toutes les catégorie, marque, couleur, caractéristique ou occasion apparaît de nouveau, mis à jour avec celle saisie.
- Si je clique sur le lien en dessous du champ concerné « Revenir à la liste des *catégories, marques, couleurs, caractéristiques ou occasions* », la liste réapparaît à la place de l'input de type texte.
- Si je clique en dehors du champ de type texte, après avoir cliqué dedans, et que le champ est vide, le message « Vous n'avez pas rempli ce champ » apparaît en dessous du champ concerné, surligné en rouge clair (#F8d7da). Le message reste affiché tant que les champs n'ont pas été remplis.
- Si je saisis un nom de *catégorie, marque, couleur, caractéristique ou occasion* qui existe déjà, le message « Ce nom de *catégorie, marque, couleur, caractéristique ou occasion* existe déjà » apparaît en dessous du champ concerné.
- Par défaut, les champs de saisie de texte et les listes déroulantes ont les extrémités rouges. Lorsque le champ est renseigné, les extrémités deviennent vertes.
- Vérifier que les données se sont bien inscrites en base dans la table dédiée *catégorie, marque, couleur, caractéristique ou occasion*.

### Spécifications techniques

- Les informations doivent être soumises en POST au format JSON.
- L'URL à appeler pour soumettre la création d'une catégorie est : <http://localhost:3000/api/users/categories>.
- L'objet JSON concernant les catégories est structuré ainsi :  
LIBEL\_CAT: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une marque est : <http://localhost:3000/api/users/brands>.
- L'objet JSON concernant les marques est structuré ainsi :  
NOM\_MARQUE: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une couleur est : <http://localhost:3000/api/users/colors>.
- L'objet JSON concernant les couleurs est structuré ainsi :  
LIBEL\_COUL: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une caractéristique est : <http://localhost:3000/api/users/features>.
- L'objet JSON concernant les caractéristiques est structuré ainsi :  
LIBEL\_CHARACTER: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une occasion est : <http://localhost:3000/api/users/occasions>.
- L'objet JSON concernant les occasions est structuré ainsi :

LIBEL\_OCCAS: string.

Ces données sont obligatoires.

- La base de données MySQL nommée « dressing » possède les tables *categorie*, *couleur*, *marque*, *occasion*, *caracteristique*, contenant les champs détaillés dans le dictionnaire des données ci-joint.
- Le serveur renvoie le code http 400 « Bad Request » s'il y a une erreur, et le code http 200 « OK » si tout s'est correctement déroulé.
- Les routes /api/brands, /api/categories, /api/colors, /api/features, /api/occasions doivent être créées afin que le serveur puisse récupérer et traiter les informations que le front lui envoie.

#### **User story D.4 : Contrôle sur l'enregistrement d'un vêtement : Vérifier que l'inscription en base a bien été effectuée**

DOR : Les US D.1, D.2 et D.3 ont bien été effectuées

DOD : Je crée un vêtement en me servant du formulaire de création d'un vêtement

- En tant qu'utilisateur.
- J'ai complété le formulaire de création d'un vêtement.
- Je clique sur le bouton « Enregistrer ».
- Vérifier que les données se sont bien inscrites en base dans la table *vetement*.
- Vérifier que les données se sont bien inscrites en base dans les tables associatives *vet\_caract\_assoc*, *vet\_coul\_assoc*, et *vet\_occas\_assoc*.
- Si une erreur survient lors de l'envoi des données pour l'enregistrement en base, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut du formulaire.
- Lorsque l'enregistrement s'est correctement déroulé, je suis redirigé vers la page d'accueil.

#### **Spécifications techniques**

- La base de données MySQL nommée « dressing » possède les tables *vetement*, *vet\_caract\_assoc*, *vet\_coul\_assoc*, et *vet\_occas\_assoc*, contenant les champs détaillés dans le dictionnaire des données ci-joint.
- Le serveur renvoie le code http 400 « Bad Request » s'il y a une erreur, et le code http 200 « OK » si tout s'est correctement déroulé.
- Des requêtes préparées doivent être utilisées afin de se protéger des injections SQL.
- La route /api/clothes doit être créée afin que le serveur puisse récupérer et traiter les informations que le front lui envoie.

### **E. Liste de tous les vêtements**

#### **User story E.1 : Un utilisateur souhaite visualiser la liste de ses vêtements**

DOR : L'utilisateur est connecté

DOD : J'accède à la page de visualisation de la liste des vêtements

- En tant qu'utilisateur.
- Je me connecte au site MonDressing et je suis directement dirigé sur la page d'accueil.
- Je clique dans la barre de navigation sur le bouton « Dressing », puis sur le bouton « Tous mes vêtements ».
- La page correspond à la maquette 5 en annexe.

#### **Spécifications techniques**

- L'URL de la page est <http://localhost:4200/clothes-list>.
- La partie « clothe-list » est un composant Angular du module Angular clothes.

### User story E.2 : Gestion des filtres sur la page affichant la liste de tous les vêtements

DOR : La page affichant la liste de tous les vêtements existe

DOD : Mes vêtements s'affichent en fonction des filtres/critères choisis

- En tant qu'utilisateur.
- Lorsque j'arrive sur la page affichant la liste de tous les vêtements, par défaut, tous les vêtements que j'ai enregistrés en base de données apparaissent par ordre alphabétique. La première liste déroulante de filtre est par défaut sur « ---Filtrer--- » et la seconde liste déroulante de filtre est par défaut sur « ----- ».
- Lorsque je clique sur la première liste déroulante de filtre, les choix qui apparaissent sont : « Par caractéristique », « Par catégorie », « Par couleur », « Par marque », « Par note », et « Par occasion ».
- Lorsque je clique sur l'un des filtres de la première liste déroulante, la seconde liste déroulante se met à jour avec le détail des données contenues dans la table correspondante en base de données. Par exemple, si je clique sur « Par couleur » dans la première liste déroulante, la seconde liste déroulante affichera « Bleu », « Blanc », « Rouge » etc par ordre alphabétique, en fonction des couleurs enregistrées en base de données dans la table *couleur*.
- Lorsque je clique sur un critère dans la seconde liste déroulante, la liste des vêtements affichés est mise à jour et affiche uniquement les vêtements concernés par le filtre choisi, par ordre alphabétique.
- S'il n'y a pas de vêtement en base de données concernant le filtre choisi, alors la phrase « Il n'y a pas de vêtement correspondant à cette recherche » apparaît sous les filtres, à la place des blocs de vêtements.
- Lorsque je clique de nouveau sur « ---Filtrer--- » dans la première liste déroulante, la seconde liste déroulante se remet automatiquement sur « ----- » et la liste de tous les vêtements présents en base de données s'affiche par ordre alphabétique.

### Spécifications techniques

- L'affichage dynamique est fait avec les directives structurales Angular.
- L'appel doit être soumis en GET au format JSON.
- L'URL à appeler pour obtenir la liste des vêtements pour le filtre est : <http://localhost:3000/api/clothes>.
- L'URL à appeler pour obtenir une marque en particulier pour le filtre est : [http://localhost:3000/api/brands/ID\\_MARQUE](http://localhost:3000/api/brands/ID_MARQUE).
- L'URL à appeler pour obtenir une catégorie en particulier pour le filtre est : [http://localhost:3000/api/categories/ID\\_CAT](http://localhost:3000/api/categories/ID_CAT).
- L'URL à appeler pour obtenir une caractéristique en particulier pour le filtre est : [http://localhost:3000/api/features/ID\\_CHARACTER](http://localhost:3000/api/features/ID_CHARACTER).
- L'URL à appeler pour obtenir une note en particulier pour le filtre est : [http://localhost:3000/api/notes/ID\\_NOTE](http://localhost:3000/api/notes/ID_NOTE).
- L'URL à appeler pour obtenir une occasion en particulier pour le filtre est : [http://localhost:3000/api/occasions/ID\\_OCCASION](http://localhost:3000/api/occasions/ID_OCCASION).
- L'URL à appeler pour obtenir une couleur en particulier pour le filtre est : [http://localhost:3000/api/colors/ID\\_COUL](http://localhost:3000/api/colors/ID_COUL).
- L'URL à appeler pour obtenir la liste des marques est : <http://localhost:3000/api/brands>.

- L'URL à appeler pour obtenir la liste des catégories est : <http://localhost:3000/api/categories>.
- L'URL à appeler pour obtenir la liste des couleurs est : <http://localhost:3000/api/colors>.
- L'URL à appeler pour obtenir la liste des caractéristiques est : <http://localhost:3000/api/features>.
- L'URL à appeler pour obtenir la liste des notes est : <http://localhost:3000/api/notes>.
- L'URL à appeler pour obtenir la liste des occasions est : <http://localhost:3000/api/occasions>.

## F. Détail d'un vêtement

<b>User story F.1 : Un utilisateur souhaite visualiser le détail d'un vêtement</b>
DOR : L'utilisateur est sur la page de visualisation de la liste des vêtements
DOD : J'accède à la modale de détail d'un vêtement
<ul style="list-style-type: none"> <li>• En tant qu'utilisateur.</li> <li>• Je suis sur la page de visualisation de la liste des vêtements et je clique sur l'un des blocs présentant un vêtement.</li> <li>• La modale qui s'affiche correspond à la maquette 6 en annexe.</li> <li>• Les informations qui s'affichent sont récupérées dynamiquement dans la base de données.</li> <li>• Lorsque je clique sur la croix en haut à gauche ou sur la croix en bas à droite, la modale de détail se ferme et je retourne sur la page de visualisation de la liste des vêtements.</li> </ul>
<b>Spécifications techniques</b>
<ul style="list-style-type: none"> <li>• La partie « clothe-detail » est un composant Angular du module Angular clothes.</li> <li>• Ce composant s'affiche sous la forme d'une modale</li> <li>• L'URL à appeler pour obtenir les données concernant un vêtement en particulier est : <a href="http://localhost:3000/api/clothes/ID_VET">http://localhost:3000/api/clothes/ID_VET</a>.</li> </ul>

## G. Modification d'un vêtement

<b>User story G.1 : Un utilisateur souhaite modifier un vêtement</b>
DOR : L'utilisateur est sur la modale de détail d'un vêtement
DOD : J'accède à la page de modification d'un vêtement
<ul style="list-style-type: none"> <li>• En tant qu'utilisateur.</li> <li>• Je suis sur la modale de détail d'un vêtement et je clique sur le bouton permettant de modifier un vêtement.</li> <li>• La page qui s'affiche correspond à la maquette 7 en annexe.</li> <li>• Les données concernant le vêtement à modifier sont renseignées dans tous les champs : elles sont récupérées dynamiquement depuis la base de données.</li> <li>• Les champs textes sont pré-remplis, les cases à cocher sont cochées sur les valeurs concernant le vêtement, les listes déroulantes affichent la valeur concernant le vêtement, et l'image qui s'affiche est celle du vêtement à modifier.</li> <li>• Lorsque je clique sur le bouton « Annuler », je suis redirigé vers la page « clothe-list » qui liste tous les vêtements.</li> </ul>

### Spécifications techniques

- L'URL de la page est [http://localhost:4200/clothe-update/ID\\_VET](http://localhost:4200/clothe-update/ID_VET).
- La partie « clothe-update » est un composant Angular du module Angular clothes.
- Les appels doivent être soumis en GET au format JSON.
- L'URL à appeler pour obtenir les données concernant un vêtement en particulier est : [http://localhost:3000/api/clothes/ID\\_VET](http://localhost:3000/api/clothes/ID_VET).
- L'URL à appeler pour obtenir la liste des marques est : <http://localhost:3000/api/brands>.
- L'URL à appeler pour obtenir la liste des catégories est : <http://localhost:3000/api/categories>.
- L'URL à appeler pour obtenir la liste des couleurs est : <http://localhost:3000/api/colors>.
- L'URL à appeler pour obtenir la liste des caractéristiques est : <http://localhost:3000/api/features>.
- L'URL à appeler pour obtenir la liste des notes est : <http://localhost:3000/api/notes>.
- L'URL à appeler pour obtenir la liste des occasions est : <http://localhost:3000/api/occasions>.

### User story G.2 : Contrôle sur la modification d'un vêtement : Vérifier que tous les champs obligatoires sont correctement remplis

DOR : Le formulaire permettant de modifier un vêtement existe. L'US D.2 a été faite.

DOD : Je gère le cas des champs obligatoires sur le formulaire

- En tant qu'utilisateur.
- Lorsque je remplis le formulaire de modification d'un vêtement.
- Si j'essaie de cliquer sur le bouton « Enregistrer » alors qu'un champ obligatoire n'est pas rempli, je ne peux pas car celui-ci est désactivé.
- Les champs obligatoires sont marqués par un astérisque et la phrase « \* Tous ces champs sont obligatoires. Vous pourrez les modifier ultérieurement. » est affichée en bas de page entre l'image et le bouton « Enregistrer ».
- Les champs obligatoires sont : « Nom », « Catégorie », « Marque », « Couleurs », « Caractéristiques », « Occasions », « Description », et « Note », comme dans le formulaire de création d'un vêtement dans l'US D.2.
- Si je clique en dehors de n'importe quel champ du formulaire, après avoir vidé ce qui était dedans, le message « Vous n'avez pas rempli ce champ » apparaît en dessous du champ concerné, surligné en rouge clair (#F8d7da). Le message reste affiché tant que les champs n'ont pas été remplis.
- Si je saisis un nom de vêtement qui existe déjà, le message « Ce nom de vêtement existe déjà » apparaît en dessous du champ concerné.
- Par défaut, les champs de saisie de texte et les listes déroulantes ont les extrémités rouges. Lorsque le champ est renseigné, les extrémités deviennent vertes.
- Si tout est correctement complété, l'appel au service concernant la modification d'un vêtement se fait.

### Spécifications techniques

- Les vérifications sont faites avec les directives structurelles Angular.
- Les informations doivent être soumises en PUT au format JSON.
- L'URL à appeler pour soumettre la modification d'un vêtement est : <http://localhost:3000/api/clothes>.

- L'objet JSON est structuré ainsi :  
NOM\_VET : string, FK\_ID\_CAT : number, FK\_ID\_MARQUE : number,  
FK\_ID\_NOTE : number, DESCRIPT\_VET : string, ID\_CHARACTER : any[],  
ID\_COUL : any[], ID\_OCCAS : any[], IMG\_VET : any, FK\_ID\_USER : number.  
Ces données sont obligatoires.
- L'URL à appeler pour obtenir la liste des noms de vêtements est :  
<http://localhost:3000/api/clothes/clothesname>.
- L'URL à appeler pour obtenir la liste des noms de marques est :  
<http://localhost:3000/api/brands/brandname>.
- L'URL à appeler pour obtenir la liste des noms de catégories est :  
<http://localhost:3000/api/categories/categoryname>.
- L'URL à appeler pour obtenir la liste des noms de couleurs est :  
<http://localhost:3000/api/colors/colorname>.
- L'URL à appeler pour obtenir la liste des noms de caractéristiques est :  
<http://localhost:3000/api/features/featurename>.
- L'URL à appeler pour obtenir la liste des noms d'occasions est :  
<http://localhost:3000/api/occasions/occasionname>.

### User story G.3 : Un utilisateur souhaite renseigner une catégorie, marque, couleur, caractéristique ou occasion qui n'est pas dans la liste

DOR : L'utilisateur est sur la page contenant le formulaire de modification d'un vêtement

DOD : Je renseigne une catégorie, marque, couleur, caractéristique ou occasion qui s'ajoute à la liste

- En tant qu'utilisateur.
- La catégorie, marque, couleur, caractéristique ou occasion que je souhaite renseigner n'est pas proposée dans la liste.
- Je clique sur le lien en dessous du champ concerné « *catégorie, marque, couleur, caractéristique ou occasion* inexistante dans la liste ».
- Un champ texte apparaît permettant de renseigner la catégorie, marque, couleur, caractéristique ou occasion que je souhaite ajouter.
- Lorsque je clique sur le bouton + situé à côté du champ, la catégorie, marque, couleur, caractéristique ou occasion s'ajoute à la liste et le champ présentant la liste de toutes les catégorie, marque, couleur, caractéristique ou occasion apparaît de nouveau, mis à jour avec celle saisie.
- Si je clique sur le lien en dessous du champ concerné « Revenir à la liste des *catégories, marques, couleurs, caractéristiques ou occasions* », la liste réapparaît à la place de l'input de type texte.
- Si je clique en dehors du champ de type texte, après avoir cliqué dedans, et que le champ est vide, le message « Vous n'avez pas rempli ce champ » apparaît en dessous du champ concerné, surligné en rouge clair (#F8d7da). Le message reste affiché tant que les champs n'ont pas été remplis.
- Si je saisis un nom de *catégorie, marque, couleur, caractéristique ou occasion* qui existe déjà, le message « Ce nom de *catégorie, marque, couleur, caractéristique ou occasion* existe déjà » apparaît en dessous du champ concerné.
- Par défaut, les champs de saisie de texte et les listes déroulantes ont les extrémités rouges. Lorsque le champ est renseigné, les extrémités deviennent vertes.
- Vérifier que les données se sont bien inscrites en base dans la table dédiée *catégorie, marque, couleur, caractéristique ou occasion*.

### Spécifications techniques



- Ce développement a déjà été fait dans l'US D.3.
- Les informations doivent être soumises en POST au format JSON.
- L'URL à appeler pour soumettre la création d'une catégorie est :  
<http://localhost:3000/api/users/categories>.
- L'objet JSON concernant les catégories est structuré ainsi :  
LIBEL\_CAT: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une marque est :  
<http://localhost:3000/api/users/brands>.
- L'objet JSON concernant les marques est structuré ainsi :  
NOM\_MARQUE: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une couleur est :  
<http://localhost:3000/api/users/colors>.
- L'objet JSON concernant les couleurs est structuré ainsi :  
LIBEL\_COUL: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une caractéristique est :  
<http://localhost:3000/api/users/features>.
- L'objet JSON concernant les caractéristiques est structuré ainsi :  
LIBEL\_CHARACTER: string.  
Ces données sont obligatoires.
- L'URL à appeler pour soumettre la création d'une occasion est :  
<http://localhost:3000/api/users/occasions>.
- L'objet JSON concernant les occasions est structuré ainsi :  
LIBEL\_OCCAS: string.  
Ces données sont obligatoires.
- La base de données MySQL nommée « dressing » possède les tables *categorie*, *couleur*, *marque*, *occasion*, *caracteristique*, contenant les champs détaillés dans le dictionnaire des données ci-joint.
- Le serveur renvoie le code http 400 « Bad Request » s'il y a une erreur, et le code http 200 « OK » si tout s'est correctement déroulé.
- Les routes /api/brands, /api/categories, /api/colors, /api/features, /api/occasions doivent être créées afin que le serveur puisse récupérer et traiter les informations que le front lui envoie.

#### **User story G.4 : Contrôle sur la modification d'un vêtement : Vérifier que l'inscription en base a bien été effectuée**

DOR : Les US G.1, et G.2 ont bien été effectuées

DOD : Je modifie un vêtement en me servant du formulaire de modification d'un vêtement

- En tant qu'utilisateur.
- J'ai complété le formulaire de modification d'un vêtement.
- Je clique sur le bouton « Enregistrer ».
- Vérifier que les données se sont bien modifiées en base dans la table *vetement*.
- Vérifier que les anciennes données se sont bien supprimées et que les nouvelles données se sont bien inscrites en base dans les tables associatives *vet\_caract\_assoc*, *vet\_coul\_assoc*, et *vet\_occas\_assoc*.
- Si une erreur survient lors de l'envoi des données pour la modification en base, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut du

formulaire.
<ul style="list-style-type: none"> <li>• Lorsque l'enregistrement s'est correctement déroulé, je suis redirigé vers la page « clothe-list » qui liste tous les vêtements.</li> </ul>
<b>Spécifications techniques</b>
<ul style="list-style-type: none"> <li>• La base de données MySQL nommée « dressing » possède les tables <i>vetement</i>, <i>vet_caract_assoc</i>, <i>vet_coul_assoc</i>, et <i>vet_occas_assoc</i>, contenant les champs détaillés dans le dictionnaire des données ci-joint.</li> <li>• Le serveur renvoie le code http 400 « Bad Request » s'il y a une erreur, et le code 204 si tout s'est correctement déroulé.</li> <li>• Des requêtes préparées doivent être utilisées afin de se protéger des injections SQL.</li> </ul>

## H. Suppression d'un vêtement

<b>User story H.1 : Un utilisateur souhaite supprimer un vêtement</b>
DOR : L'utilisateur est sur la modale de détail d'un vêtement
DOD : J'accède à la modale de confirmation de suppression d'un vêtement
<ul style="list-style-type: none"> <li>• En tant qu'utilisateur.</li> <li>• Je suis sur la modale de détail d'un vêtement et je clique sur le bouton permettant de supprimer un vêtement.</li> <li>• La modale qui s'affiche correspond à la maquette 8 en annexe.</li> <li>• Lorsque je clique sur la croix en haut à gauche ou sur le bouton « Annuler », la modale de suppression se ferme et je retourne sur la page de visualisation de la liste des vêtements.</li> </ul>
<b>Spécifications techniques</b>
<ul style="list-style-type: none"> <li>• La partie « clothe-delete » est un composant Angular du module Angular clothes.</li> <li>• Ce composant s'affiche sous la forme d'une modale.</li> </ul>

<b>User story H.2 : Contrôle sur la suppression d'un vêtement : Vérifier que la suppression en base a bien été effectuée</b>
DOR : L'US H.1 a bien été effectuée
DOD : Je supprime un vêtement en me servant du formulaire de suppression d'un vêtement
<ul style="list-style-type: none"> <li>• En tant qu'utilisateur.</li> <li>• Je suis sur la modale de suppression d'un vêtement et je clique sur le bouton « Supprimer ».</li> <li>• Si une erreur survient lors de l'envoi des données pour la suppression en base, la phrase « Une erreur est survenue. Merci de réessayer » apparaît en haut de la modale.</li> <li>• Lorsque la suppression s'est correctement déroulée, je suis redirigé vers la page « clothe-list » qui liste tous les vêtements.</li> </ul>
<b>Spécifications techniques</b>
<ul style="list-style-type: none"> <li>• Les informations doivent être soumises en DELETE au format JSON.</li> <li>• L'URL à appeler pour supprimer un vêtement en particulier est : <a href="http://localhost:3000/api/clothes/ID_VET">http://localhost:3000/api/clothes/ID_VET</a>.</li> <li>• La base de données MySQL nommée « dressing » possède les tables <i>vetement</i>, <i>vet_caract_assoc</i>, <i>vet_coul_assoc</i>, et <i>vet_occas_assoc</i>, contenant les champs détaillés dans le dictionnaire des données ci-joint.</li> <li>• Vérifier que les données se sont bien supprimées en base dans la table <i>vetement</i>.</li> <li>• Vérifier que les données se sont bien supprimées en base dans les tables</li> </ul>



associatives *vet\_caract\_assoc*, *vet\_coul\_assoc*, et *vet\_occas\_assoc*.

- Le serveur renvoie le code http 418 « I'm a Teapot » s'il y a une erreur, et le code http 204 « No Content » si tout s'est correctement déroulé.
- Des requêtes préparées doivent être utilisées afin de se protéger des injections SQL.

## I. Affichage de la météo

<b>User story I.1 : Un utilisateur souhaite afficher la météo de sa ville</b>
DOR : L'utilisateur est sur la page d'accueil de l'application
DOD : J'accède à la modale de confirmation de suppression d'un vêtement
<ul style="list-style-type: none"><li>• En tant qu'utilisateur.</li><li>• Je suis sur la page d'accueil de l'application et je clique dans la barre de navigation sur le bouton « Météo ».</li><li>• La page qui s'affiche correspond à la maquette 14 en annexe.</li><li>• Il s'agit d'un module extérieur à intégrer.</li><li>• Le bouton « Chercher ! » est de la couleur principale #4DE8BE. Il ressemble au bouton « Enregistrer » du formulaire de création d'un vêtement.</li><li>• Lorsque je saisis mon code postal et mon pays et que je clique sur le bouton « Chercher ! », les colonnes qui s'affichent ont la couleur de fond #4DE8BE.</li></ul>
<b>Spécifications techniques</b>
<ul style="list-style-type: none"><li>• La partie « weather » est un composant Angular du module Angular weather.</li><li>• Ce module est un module extérieur conçu par un autre développeur et intégré à l'application.</li><li>• L'URL de la page est <a href="http://localhost:4200/weather">http://localhost:4200/weather</a>.</li></ul>

## J. Création d'une tenue

Cette US sera rédigée lors de la mise en place de la V2 de l'application

## K. Liste de toutes les tenues

Cette US sera rédigée lors de la mise en place de la V2 de l'application

## L. Détail d'une tenue

Cette US sera rédigée lors de la mise en place de la V2 de l'application

## M. Modification d'une tenue

Cette US sera rédigée lors de la mise en place de la V2 de l'application

## N. Suppression d'une tenue

Cette US sera rédigée lors de la mise en place de la V2 de l'application

## O. Génération d'une tenue

Cette US sera rédigée lors de la mise en place de la V2 de l'application