|  |
| --- |
|  |
| User Stories |
| Application Web Dressing |

|  |
| --- |
| Léa DELANNAY  29/07/2019 |

Table des matières

[I. Features 2](#_Toc13463581)

[II. User stories 2](#_Toc13463582)

[A. Enregistrement 2](#_Toc13463583)

[B. Connexion 2](#_Toc13463584)

[C. Accueil et barre de navigation 2](#_Toc13463585)

[D. Création d’un vêtement 2](#_Toc13463586)

[E. Liste de tous les vêtements 2](#_Toc13463587)

[F. Détail d’un vêtement 2](#_Toc13463588)

[G. Modification d’un vêtement 2](#_Toc13463589)

[H. Suppression d’un vêtement 2](#_Toc13463590)

[I. Création d’une tenue 3](#_Toc13463591)

[J. Liste de toutes les tenues 3](#_Toc13463592)

[K. Détail d’une tenue 3](#_Toc13463593)

[L. Modification d’une tenue 3](#_Toc13463594)

[M. Suppression d’une tenue 3](#_Toc13463595)

[N. Affichage de la météo 3](#_Toc13463596)

# 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

# User stories

## Enregistrement

|  |
| --- |
| User story A.1 : Une personne souhaite s’enregistrer pour utiliser l’application |
| DOR : Le site MonDressing existe |
| DOD : J’accède à la page de création d’un compte |
| * En tant que personne n’ayant pas encore de compte. * J’accède au site MonDressing. * Je clique sur le bouton « S’enregistrer » dans la barre de navigation. Au click, la couleur de fond du bouton « S’enregistrer » devient plus claire afin d’indiquer quel bouton vient d’être cliqué. * J’accède au formulaire permettant de me créer un compte. * La page correspond à la maquette 1 en annexe. |
| Spécifications techniques |
| * Le site a été préalablement initialisé avec l’outil Angular CLI. * L’URL de la page est http://localhost :4200/register. * La partie register est un composant Angular du module Angular account. |

|  |
| --- |
| User story A.2 : Contrôle sur l’enregistrement d’une personne : Vérifier que tous les champs obligatoires sont correctement remplis |
| DOR : Le formulaire permettant de s’enregistrer existe |
| DOD : Je gère le cas des champs obligatoires sur le formulaire |
| * En tant que personne n’ayant pas encore de compte. * Lorsque je complète le formulaire d’enregistrement. * Si je clique sur le bouton « S’ENREGISTRER » alors que je n’ai pas rempli les trois champs obligatoires, celui-ci est désactivé. * Le champ « Adresse email » nécessite un @. Si j’envoie le formulaire sans @, la phrase « Une erreur est survenue. Merci de réessayer » apparait 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 » apparait 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 » apparait 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 structurelles 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 soient encryptées sur le réseau, 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 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 ». * Vérifier que les données se sont bien inscrites en base dans la table *user*. * Si une erreur survient lors de l’envoi des données pour l’enregistrement en base, la phrase « Une erreur est survenue. Merci de réessayer » apparait en haut du formulaire. * Lorsque l’enregistrement s’est correctement déroulé, la phrase « Votre compte a bien été créé. Merci de vous connecter » apparait 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 ID\_USER(int11), PSEUDO\_USER(varchar50), LOGIN\_USER(varchar255), et MDP\_USER(varchar255). Le détail des champs de la base de données 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 400 s’il y a une erreur, et le code 200 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 les informations que le front lui envoie. |

## 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. |

|  |
| --- |
| 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 » apparait 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 structurelles 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.   * Pour que les données soient encryptées sur le réseau, le front et le back doivent communiquer via le protocole https. |

|  |
| --- |
| 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 » apparait 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 » apparait 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 avant d’être enregistrés en base. * Le serveur renvoie le code 403 si les identifiants sont incorrects, et le code 200 accompagné d’un json web token 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 les informations que le front lui envoie. |

## 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 apparait |
| * 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 ou http://localhost :4200/homepage. * La partie accueil est un composant Angular du module Angular app.module. |

## 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 correspond à la maquette 4 en annexe. * Par défaut, l’image qui apparait dans l’encadré dédié est l’image de la maquette. |
| Spécifications techniques |
| * L’URL de la page est http://localhost :4200/clothe-create. * La partie clothe-create est un composant Angular du module Angular clothes. |

|  |
| --- |
| 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 je clique sur le bouton « Enregistrer » alors que je n’ai pas rempli tous les champs obligatoires, 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 » apparait 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 » apparait en dessous du champ concerné. * Par défaut, les champs de saisie de texte et les selects 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 structurelles 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\_CARACT : any[], ID\_COUL : any[], ID\_OCCAS : any[], IMG\_VET : any, FK\_ID\_USER : number.  Ces données sont obligatoires.   * Pour que les données soient encryptées sur le réseau, le front et le back doivent communiquer via le protocole https. |

|  |
| --- |
| 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 apparait 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 apparait 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 réapparait 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 » apparait 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 » apparait en dessous du champ concerné. * Par défaut, les champs de saisie de texte et les selects ont les extrémités rouges. Lorsque le champ est renseigné, les extrémités deviennent vertes. |
| Spécifications techniques |
| * La partie clothe-create est un composant Angular du module Angular clothes. * 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\_CARACT: 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 400 s’il y a une erreur, et le code 200 si tout s’est correctement déroulé. |

|  |
| --- |
| 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 » apparait 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 » a été créée et possède la table user contenant les champs ID\_USER(int11), PSEUDO\_USER(varchar50), LOGIN\_USER(varchar255), et MDP\_USER(varchar255). * Le serveur renvoie le code 400 s’il y a une erreur, et le code 200 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 les informations que le front lui envoie. |

## Liste de tous les vêtements

## Détail d’un vêtement

## Modification d’un vêtement

## Suppression d’un vêtement

## Création d’une tenue

## 

## Liste de toutes les tenues

## Détail d’une tenue

## Modification d’une tenue

## Suppression d’une tenue

## Affichage de la météo

## Génération d’une tenue