

3291.2 JEE-Spring I – Rapport – ISC3il-b

WalkingFish

Site vitrine pour la marque de vêtements WalkingFish

Étudiants participant à ce travail :

AUBERT Nicolas, ISC3il-b

Présenté à :

CHÈVRE Sébastien

Restitution du rapport : **30.01.2023**

Période : **2022 – 2023**

École : **HE-Arc, Neuchâtel**

haute école
neuchâtel berne jura



ingénierie
www.he-arc.ch

Table des matières

1 - INTRODUCTION	I
1.1 - CONTEXTE	I
1.2 - BUT DU PROJET	I
1.3 - OBJECTIFS PRINCIPAUX	I
1.4 - OBJECTIFS SECONDAIRES	I
1.5 - LIVRABLES ATTENDUS	I
2 - CONCEPTION	III
2.1 - SCHÉMA D'INTERACTIONS (USE CASE)	III
2.2 - ARCHITECTURE	IV
2.2.1 - MCD	IV
2.2.2 - MLD	IV
2.2.3 - Schéma des routes	V
2.3 - PLANIFICATION	VI
3 - RÉALISATION	7
3.1 - LES MODÈLES	7
3.1.1 - Article	7
3.1.2 - Picture	7
3.1.3 - Colori	7
3.2 - LES CONTRÔLEURS	7
3.3 - LES REPOSITORY	7
3.3.1 - ArticleRepository	8
3.3.2 - PictureRepository	8
3.3.3 - ColoriRepository	8
3.4 - LES SERVICES	8
3.4.1 - CatalogService	8
3.4.2 - FileStorageService	8
3.4.3 - ColoriService	8
3.5 - L'ACCÈS ADMINISTRATEUR	9
3.6 - CI/CD	9
3.6.1 - Tests unitaires	9
3.7 - PROBLÈMES RENCONTRÉS	10
3.7.1 - Opérations en cascade avec H2	10
3.7.2 - Accès aux images sur le serveur	10
4 - RÉSULTATS	11
4.1 - CATALOGUE	11
4.1.1 - Liste des articles	11
4.1.2 - Pagination	11
4.2 - GESTION DES ARTICLES	11
4.2.1 - Connexion	12
4.2.2 - Déconnexion	12
4.2.3 - Ajout d'un article	12
4.2.4 - Modification d'un article	14
4.2.5 - Suppression	16
4.3 - GESTION DES IMAGES D'UN ARTICLE	16
4.3.1 - Ajout	16
4.3.2 - Suppression	16
4.4 - GESTION DES COLORIS D'UN ARTICLE	16

4.4.1 -	Ajout.....	17
4.4.2 -	Suppression.....	17
4.5 -	GESTION DE LA LISTE DES COLORIS.....	17
5 -	LIMITATIONS ET PERSPECTIVES	18
5.1 -	AJOUT DE NOUVELLES FONCTIONNALITÉS.....	18
5.2 -	AMÉLIORATION DE L'UI / UX.....	18
6 -	CONCLUSION	19
7 -	ANNEXES	I
7.1 -	GUIDE DÉVELOPPEUR.....	I
7.1.1 -	Mise en place de la base données.....	I
7.1.2 -	Lancement de l'application.....	I
7.1.3 -	Compte administrateur.....	I
7.2 -	TABLE DES ILLUSTRATIONS	II
7.3 -	LISTE DES TESTS UNITAIRES	III
7.4 -	BIBLIOGRAPHIES ET RÉFÉRENCES	IV
7.4.1 -	Sites Web	IV

1 - Introduction

1.1 - Contexte

Walking Fish est une marque de vêtements créée par deux anciens étudiants de la HEP-Bejune, à la Chaux-de-Fonds.

Afin d'améliorer la visibilité de leur entreprise, je me suis proposé afin de réaliser une application web vitrine, représentant la marque et listant les différents articles disponibles.

1.2 - But du projet

Créer une application web vitrine à l'aide du framework Spring, sur laquelle les utilisateurs pourront accéder aux articles proposés par la marque Walking Fish.

Les personnes possédant un accès administrateur pourront se connecter afin de modérer les articles (ajout, modification, suppression).

1.3 - Objectifs principaux

- Lister les différents articles sur la page de vente (données simulées), avec une pagination permettant de limiter le nombre d'articles affichés sur une page
 - Un article est composé d'un nom, une description, un prix et une image,
 - Les images seront stockées et récupérées via une API (imgur par exemple)
- Ajouter le CRUD sur les articles,
- Ajouter la connexion / déconnexion pour les administrateurs, afin de sécuriser l'accès au CRUD des articles.

1.4 - Objectifs secondaires

- Nouvelle fonctionnalité grâce à laquelle un administrateur pourra ajouter de multiples images pour un seul article,
- Les administrateurs pourront ajouter les tailles disponibles pour les articles,
- Les administrateurs pourront ajouter les coloris disponibles pour les articles.

1.5 - Livrables attendus

- Cahier des charges
 - Planning initial
- Rapport de travail

- Planning effectif
- Code source

2 - Conception

2.1 - Schéma d'interactions (Use Case)

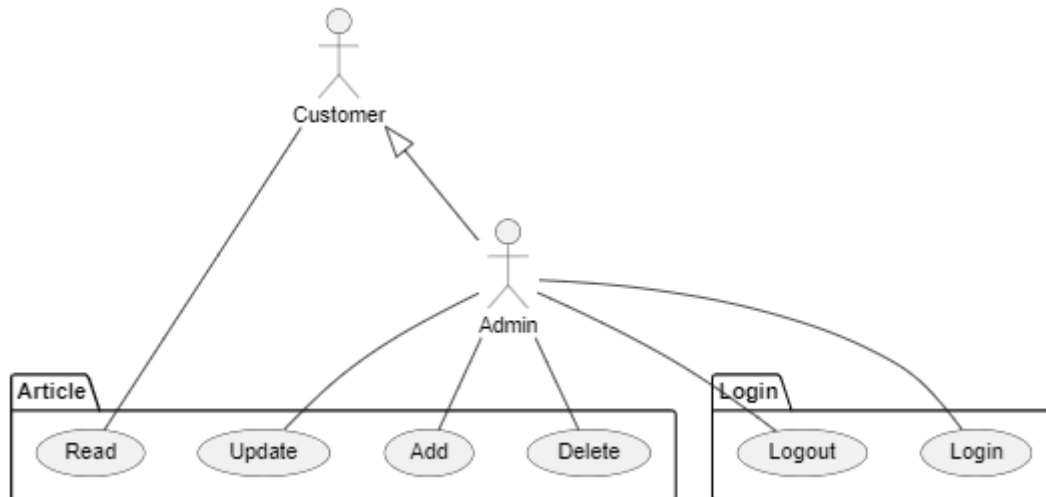


Figure 1 - Schéma d'interactions (Use Case)

Comme illustré sur la Figure 1, l'application est composée de deux types d'utilisateur différents. Le « Customer », qui correspond à un invité « Guest ». En effet, celui-ci n'est pas connecté, et peut uniquement consulter les articles.

Cependant, un utilisateur basique peut se connecter afin d'obtenir les privilèges d'un modérateur (« Admin » sur le diagramme). Disposant de ces droits, il peut alors modérer les différents articles (ajout, modification et suppression).

2.2 - Architecture

2.2.1 -MCD

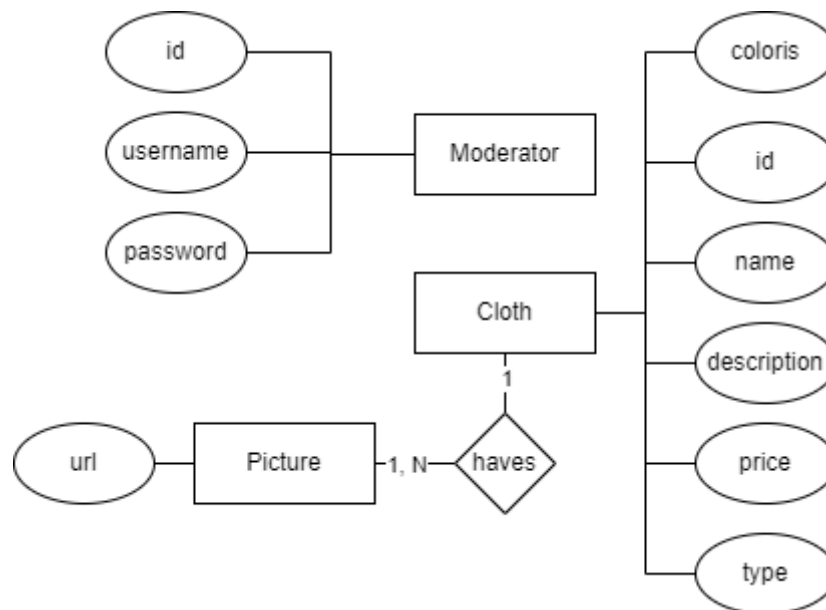
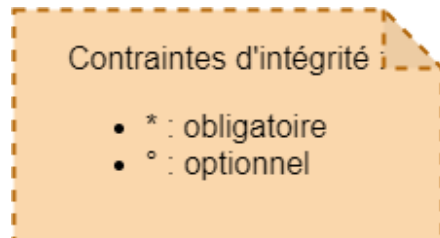


Figure 2 - MCD (Modèle Conceptuel de Données)

2.2.2 -MLD



Users		
PK		<u>user_id int NOT NULL</u>
	*	user_password CHAR(255)
UK_1	*	user_name VARCHAR(50)

Pictures		
PK		<u>picture_id int NOT NULL</u>
FK_1	*	cloth_id INT
	*	picture_url VARCHAR(100)

Clothes		
PK		<u>cloth_id int</u>
	°	cloth_coloris VARCHAR(255)
	*	cloth_name VARCHAR(30)
	*	cloth_description VARCHAR(255)
	*	cloth_price FLOAT
	°	cloth_type VARCHAR(50)

Figure 3 - MLD (Modèle Logique de Données)

2.2.3 -Schéma des routes

Route	Controller	Méthode	Opération CRUD	Pré-condition	Description	Redirection
/	IndexController	index	-	-	Affiche la page d'accueil	-
login	UserController	login	-	-	Affiche la page de connection	-
logout	UserController	logout	-	Doit être connecté (modérateur)	Déconnecte le le modérateur	/login
user/add	UserController	add	-	-	Affiche la page de création de compte pour les modérateurs	-
cloth/add	ClothController	add	-	Doit être connecté (modérateur)	Affiche la page d'ajout d'un vêtement	-
cloth/ <i>id</i>	ClothController	show	Read	-	Affiche un un vêtement	-
cloth/ <i>id</i> /delete	ClothController	delete	Delete	Doit être connecté (modérateur)	Supprime un vêtement	/clothes
cloth/ <i>id</i> /update	ClothController	update	Read	Doit être connecté (modérateur)	Affiche la page de mise à jour d'un vêtement	-
clothes	ClothController	index	Read	-	Affiche tous les vêtements	-

Figure 4 - Schémas des routes

2.3 - Planification

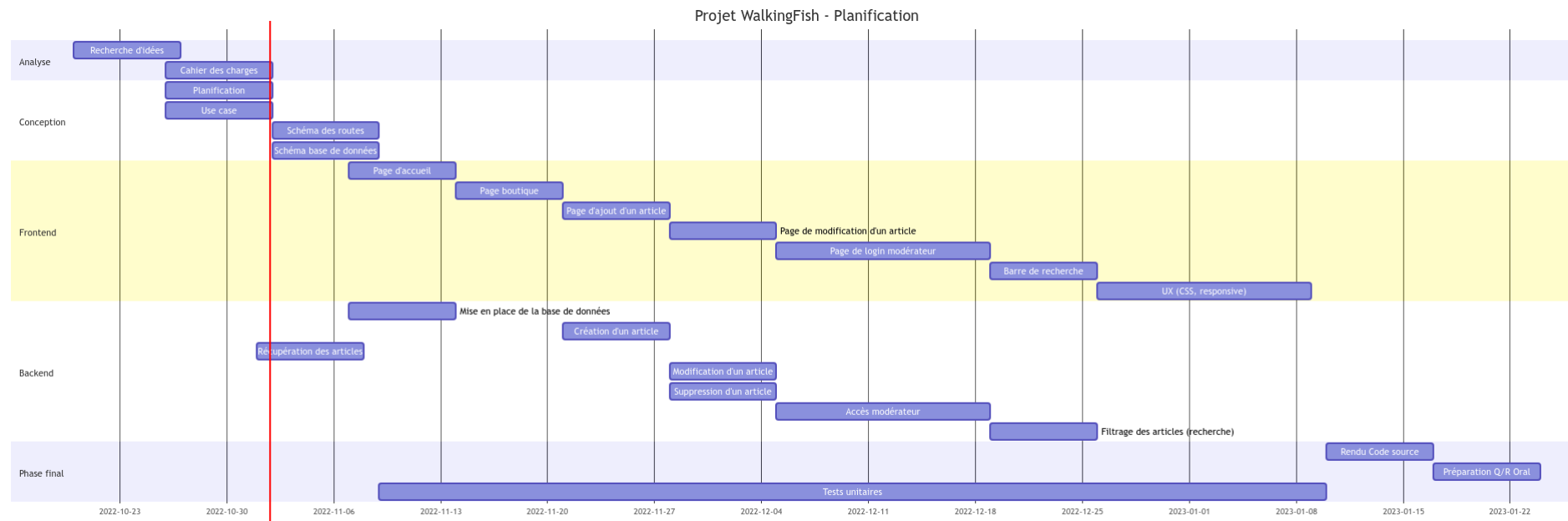


Figure 5 – Planification initiale

3 - Réalisation

3.1 - Les modèles

3.1.1 -Article

Ce modèle représente une classe "Article" qui est utilisée pour stocker des informations sur un article du catalogue. Il contient des champs tels que le nom, la description, le prix et le type de l'article. Il possède également une liste d'images liées à cet article, ainsi qu'une liste de coloris. Il contient des méthodes pour accéder et mettre à jour ces informations ainsi que des méthodes pour tester l'égalité entre deux objets Article.

3.1.2 -Picture

Ce modèle représente une classe "Picture" qui est utilisée pour stocker des informations sur les images associées à un article du catalogue. Il contient des champs tels que l'identifiant, l'url et le nom de l'image. Il possède également une relation Many-To-One avec la classe Article pour stocker l'article lié à cette image. Il contient des méthodes pour accéder et mettre à jour ces informations ainsi que des méthodes pour tester l'égalité entre les objets Picture.

3.1.3 -Colori

Ce modèle représente une classe « Colori » qui est utilisé pour stocker des informations sur les couleurs associées à un article du catalogue. Un colori est composé de :

- Un nom,
- Une valeur hexadécimale représentant la couleur,
- Une liste d'articles auxquels la couleur est associée.

Ce modèle est lié à au modèle « Article » par une relation Many-To-Many ; un article être disponible en plusieurs couleurs, mais une couleur peut être liée à plusieurs articles également.

3.2 - Les contrôleurs

Ce projet possède trois contrôleurs distincts :

- *IndexController*, utilisé pour afficher la page d'accueil,
- *ArticleController*, utilisé principalement pour l'affichage des articles :
- *AdminController*, utilisé pour la gestion des articles et des coloris (CRUD) :

3.3 - Les repository

Ce projet contient trois repositories : un pour chaque modèle.

3.3.1 -ArticleRepository

Ce repository permet de manipuler les données de la base de données qui sont liées aux articles, telles que l'ajout, la modification, la suppression, etc.

3.3.2 -PictureRepository

De la même manière, ce repository permet de manipuler les données qui sont liées aux images associées à un article (modèle Picture).

3.3.3 -ColoriRepository

Pour terminer, le repository « ColoriRepository » permet d'interagir avec la base de données afin de manipuler les coloris associés aux différents articles.

3.4 - Les services

3.4.1 -CatalogService

Ce service offre de nombreuses méthodes afin de gérer les articles ainsi que leurs images dans la base de données :

- Récupérer tous les articles,
- Récupérer un article à partir de son identifiant,
- Récupérer les articles en fonction du numéro de page actuel (pagination),
- Ajouter / modifier / supprimer un article,
- Supprimer une image d'un article,
- Ajouter une image à un article,
- Etc.

3.4.2 -FileStorageService

Ce service permet de gérer la sauvegarde et la suppression de fichiers sur le serveur. Il simplifie grandement la gestion des images liées aux articles. Il possède deux méthodes :

- Sauvegarde d'un fichier sur le serveur,
- Suppression d'un fichier sur le serveur.

3.4.3 -ColoriService

Tout comme le « CatalogService », ce service permet de gérer les différents coloris dans la base de données. Il contient des méthodes pour :

- Récupérer tous les coloris,
- Récupérer un coloris à partir de son identifiant unique (clé primaire ID),

- Enregistrer un nouveau coloris,
- Supprimer un coloris existant,
- Supprimer tous les coloris.

3.5 - L'accès administrateur

Pour mettre en place l'accès administrateur, la dépendance *org.springframework.security* a été utilisé.

La configuration des accès est réalisée dans le fichier « *SecurityConfiguration.java* ». Celui-ci définit que toutes les routes commençant par « */admin/* » requièrent d'être connecté. Les autres routes peuvent être accédées sans connexion.

Un seul utilisateur est défini, puisqu'une seule personne aura les droits de modération.

3.6 - CI/CD

CI/CD (ou "Intégration continue et déploiement continu") est une pratique de développement logiciel qui consiste à automatiser le processus de construction, de test et de déploiement d'une application. Cela permet de détecter les erreurs plus tôt dans le processus de développement et de les corriger plus rapidement.

Pour mettre en place le CI/CD pour ce projet, un pipeline utilisant GitHub Actions et Maven a été mis en place sur le repository GitHub. Il décrit les étapes à suivre lorsque du code est poussé sur la branche principale.

Les étapes incluent :

- Cloner le dépôt de code
- Configurer une version de JDK
- Construire le projet avec Maven
- Tester le projet avec Maven (lancer les tests unitaires)

3.6.1 - Tests unitaires

Afin d'assurer le bon fonctionnement de l'application, une batterie de tests unitaires a été mise en place. Ils s'occupent de tester, par fragment, le service *CatalogService*.

Les tests utilisent la bibliothèque JUnit Jupiter pour les assertions et Spring Boot Test pour l'injection de dépendances.

Les tests vérifient les différentes fonctionnalités du service, telles que :

- L'injection de dépendances pour le dépôt d'articles et le service de catalogue,
- La récupération de tous les articles du catalogue,
- L'ajout d'un article dans le catalogue,

- La récupération d'un article par son identifiant,
- La mise à jour d'un article,
- La suppression d'un article,
- L'enregistrement d'une image pour un article.

Ces tests ne couvrent pas tous les cas d'utilisation possibles. D'autres scénarios pourraient être créés afin de s'assurer que le service de catalogue fonctionne correctement. Plus d'informations sur les tests unitaires se trouvent dans les annexes.

3.7 - Problèmes rencontrés

3.7.1 - Opérations en cascade avec H2

Comme exprimé dans ce poste : <https://groups.google.com/g/h2-database/c/qOX1wHVBDJ4>, la base de données H2 ne supporte pas les opérations en cascade.

Avant de comprendre cela, je rencontrais sans arrêt des erreurs de contraintes d'intégrité lors de l'ajout ou de la suppression d'articles. Je devais donc supprimer à la main toutes les images d'un article avant de supprimer l'article en lui-même.

Afin de régler de problème, j'ai donc utilisé la base de données MySQL mise en place avec Docker. Avec cette base de données, la suppression, ainsi que les autres opérations en cascade, fonctionnent correctement.

3.7.2 - Accès aux images sur le serveur

Lorsqu'un article est créé, ou bien lorsqu'une image est ajoutée à un article existant, lors de la redirection, l'image n'est pas disponible. En effet, il est nécessaire de rafraîchir la page une seconde fois que celle-ci soit bien affichée. Cela vient sûrement du fait que la redirection est plus rapide que la sauvegarde de l'image sur le serveur.

Il est sûrement possible d'attendre que l'image soit complètement sauvegardée avant de rediriger l'utilisateur, mais étant donné que cela n'affecte pas le fonctionnement de l'application, j'ai décidé de laisser de côté ce problème pour me concentrer sur d'autres problèmes / fonctionnalités.

4 - Résultats

4.1 - Catalogue

Pour les exemples suivants, des données ont été simulées. Par conséquent, tous les articles sont identiques. Également pour des détails de lisibilité, le nombre d'articles affiché par page a été limité à deux. Cependant, ce nombre s'élève à 4.

4.1.1 -Liste des articles

Comme illustré sur la Figure 6, les articles disponibles sont listés sous forme de cartes. Chaque carte possède un carrousel, permettant de feuilleter les différentes images représentant l'article. A droite de ces images se trouve une zone, dans laquelle différentes informations sont affichées, telles que la description de l'article, son prix, son type, les tailles ainsi que les coloris disponibles.

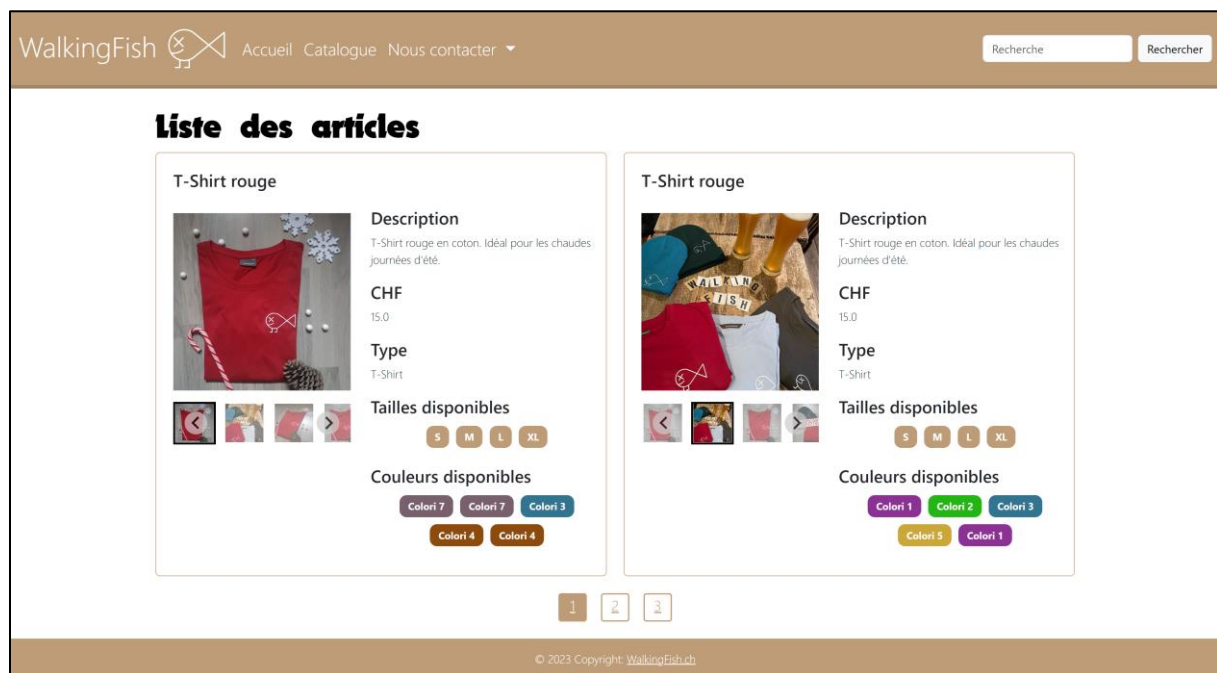


Figure 6 - Résultats : Catalogue

4.1.2 -Pagination

Toujours en se référant à la Figure 6, en bas de la page se trouvent des numéros, correspondant au numéro de page du catalogue. Le numéro courant est affiché en beige. L'utilisateur peut alors naviguer entre les différentes pages afin de consulter tous les articles.

4.2 - Gestion des articles

Afin de modérer les articles disponibles, une zone administrateur a été mise en place.

4.2.1 -Connexion

Afin d'accéder à cette zone restreinte, un modérateur doit d'abord se connecter. Pour cela, il peut se rendre sur la route suivante : « /admin ». S'il n'est pas connecté, il sera redirigé sur une page de connexion, comme illustré sur la « Figure 7 - Résultats : Connexion ».

Figure 7 - Résultats : Connexion

Une fois connecté, le modérateur aura accès à la gestion des articles et leurs images, ainsi qu'à la gestion des coloris. Cela commence par une modification de la barre de navigation. Celle-ci possèdera, en plus des anciens champs, un nouveau bouton : « Dashboard » en vert, comme présenté sur la Figure 8.

En cliquant sur ce bouton, l'utilisateur sera redirigé sur une page, à partir de laquelle il peut accéder à la gestion des différents éléments : « Gestion des coloris », « Ajouter un article », « Gestion des articles » et « Se déconnecter ».

Figure 8 - Résultats - Barre de navigation admin

4.2.2 -Déconnexion

Pour se déconnecter, un utilisateur administrateur peut simplement cliquer le bouton rouge « Se déconnecter » se trouvant sur le Dashboard. Il sera alors redirigé sur la page catalogue, et perdra l'accès à la zone de modération.

4.2.3 -Ajout d'un article

Pour ajouter un article, une fois connecté en tant qu'administrateur, il suffit de cliquer sur le bouton « Ajouter un article » se trouvant sur le dashboard. L'utilisateur sera redirigé sur la route « /admin/create ».

Comme illustré sur la Figure 9, un formulaire d'ajout sera affiché. Ce formulaire est composé de :

- Un champ « Nom », de type texte, représentant le nom de l'article,
- Un champ « Description », de type texte, représentant la description de l'article,
- Un champ « Prix », de type nombre, représentant le prix de l'article,
- Un champ « Type », de type texte, représentant le type de l'article (bonnet, T-Shirt, veste, etc.),
- Une liste de sélection multiple, à partir de laquelle il peut sélectionner les tailles disponibles de l'article,
- Une seconde liste de sélection multiple, à partir de laquelle il peut sélectionner les coloris disponibles pour l'article, à partir des coloris précédemment créés,
- Et, pour terminer, un bouton à partir duquel l'utilisateur peut choisir les images qui permettront d'illustrer l'article.

WalkingFish Accueil Catalogue Nous contacter ▾

Dashboard Recherche Rechercher

Insertion d'un article

Nom

Description

Prix

Type

Tailles

- XS
- S
- M
- L

Couleurs

- Color 1
- Color 2
- Color 3
- Color 4

Images de l'article: Sélect. fichiers Aucun fichier choisi

Créer l'article

© 2023 Copyright: WalkingFish.ch

Figure 9 - Résultats : Formulaire d'ajout d'articles (vide)

Tous ces champs, à part le champ de sélection des images, sont obligatoires. Cela signifie que le formulaire ne peut être envoyé que s'ils sont tous remplis.

Si le modérateur sélectionne des images, celles-ci seront affichées en bas du formulaire, comme présenté sur la Figure 10. Cela permet d'avoir un aperçu, et de pouvoir corriger la sélection si une erreur aurait dû se produire.

WalkingFish Accueil Catalogue Nous contacter ▾

Dashboard Recherche Rechercher

Insertion d'un article

Nom
Bonnet vert

Description
Bonnet vert en laine. Taille unique. Très confortable. Idéal pour vous tenir chaud pendant l'hiver.

Prix
25

Type
Bonnet

Tailles
L
XL
XXL
Taille unique

Couleurs
Colori 9
Colori 8
Colori 7
Colori 6

Images de l'article: Sélectionner 3 fichiers

Créer l'article

© 2023 Copyright: WalkingFish.ch

Figure 10 - Résultats : Formulaire d'ajout d'articles (rempli)

Une fois les champs remplis, l'article peut être ajouté en cliquant sur le bouton bleu « Créer l'article », à la fin du formulaire.

4.2.4 -Modification d'un article

Une fois connecté, sur la page du catalogue, les articles possèdent, dans la partie inférieure de leur carte, un bouton « Mettre à jour l'article », comme présenté dans la Figure 11.

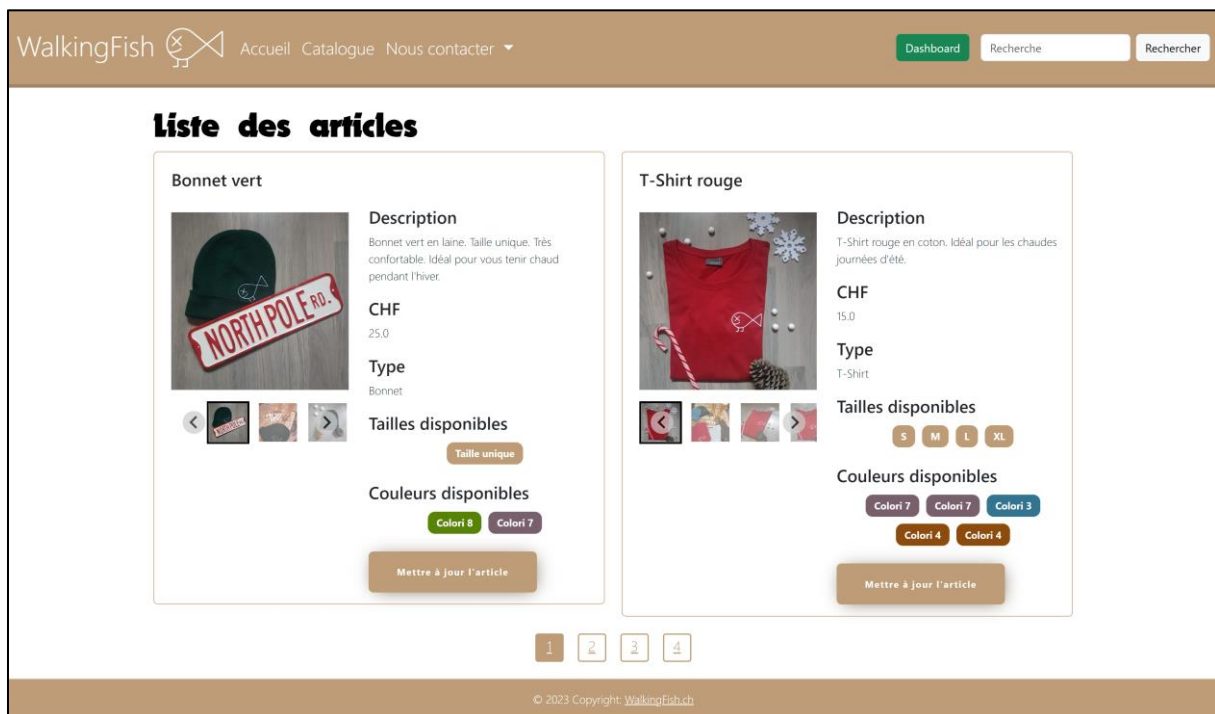


Figure 11 - Résultats : Catalogue accés administrateur

En cliquant dessus, le modérateur est redirigé sur la page de modification de l'article, comme illustré par la Figure 12.

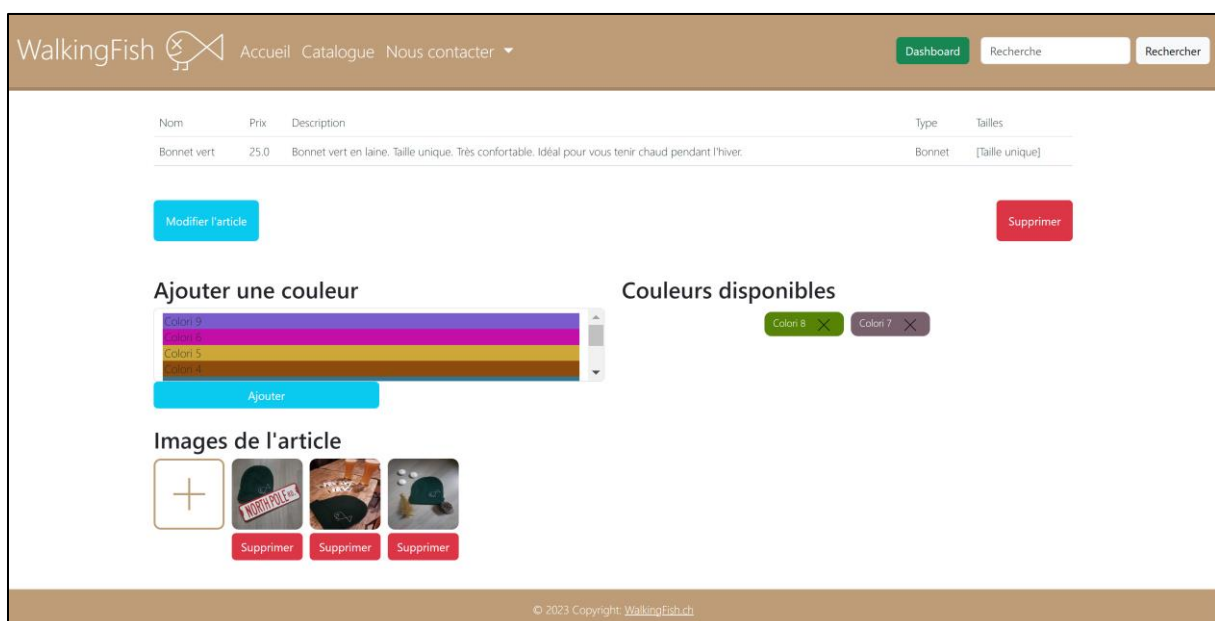


Figure 12 - Résultats : page de modification d'un article

Pour modifier les informations principales de l'article (nom, description, prix, type et tailles), l'utilisateur peut cliquer sur le bouton « Modifier l'article » en bleu clair. Il sera alors redirigé sur une page contenant un formulaire, prérempli avec les informations de l'article. Il peut alors modifier les champs nécessaires, et enregistrer les modifications en cliquant sur le bouton « Modifier l'article ».

WalkingFish Accueil Catalogue Nous contacter ▾

Dashboard Recherche Rechercher

Modification d'un article

Nom
Bonnet vert

Description
Bonnet vert en laine. Taille unique. Très confortable. Idéal pour vous tenir chaud pendant l'hiver.

Prix
20

Type
Bonnet

Tailles
XS
S
M
L

Modifier l'article

© 2023 Copyright: WalkingFish.ch

Figure 13 - Résultats : formulaire de modification des informations principales d'un article

4.2.5 -Suppression

Depuis la page de modification d'un article (Figure 12), l'utilisateur peut supprimer un article à partir du bouton rouge « Supprimer ». Cela aura pour effet de supprimer les images du serveur et que de la base données, ainsi que l'article lui-même. L'utilisateur sera redirigé sur le catalogue.

4.3 - Gestion des images d'un article

Sur la page de modification d'un article (Figure 12) sont listées les images qui composent l'article.

4.3.1 -Ajout

A gauche de ses images se trouve un bouton beige « + ». En cliquant dessus, l'utilisateur pourra choisir une ou plusieurs images à partir de son gestionnaire de fichiers. En confirmant la sélection des fichiers, ceux-ci seront directement ajoutés à l'article.

4.3.2 -Suppression

Sous chacune des images se trouve un bouton rouge « Supprimer », permettant de supprimer l'image correspondante.

4.4 - Gestion des coloris d'un article

Sur la page de modification d'un article (Figure 12) sont listés les coloris disponibles pour celui-ci.

4.4.1 -Ajout

Dans la partie gauche se trouve une liste à sélection multiple contenant les coloris qui ne sont pas encore associés à l'article. L'utilisateur peut alors cocher ceux qu'il souhaite, et cliquer sur le bouton bleu ciel « Ajouter » pour ajouter sa sélection de coloris à l'article.

4.4.2 -Suppression

Dans la partie de droite sont listés les coloris de l'article. Ceux-ci sont accompagnés d'une croix noire, à droite de leur nom. En cliquant dessus, l'utilisateur peut retirer un coloris. Cette action ne supprimera pas le coloris de la liste de tous les coloris (communes à tous les articles) ; cela supprimera uniquement la relation qui existait entre les deux.

4.5 - Gestion de la liste des coloris

Depuis le dashboard, l'utilisateur peut accéder à la page de gestion des coloris à partir du bouton « Gestion des coloris ».

Figure 14 - Résultats : page de gestion des coloris

Sur cette page se trouve un formulaire de deux champs (nom de la couleur, code hexadécimale de la couleur) permettant d'ajouter un nouveau coloris. Pour cela, il suffit de remplir les champs, et de cliquer sur le bouton « Ajouter la couleur » à droite.

En dessus de formulaire sont listées les différentes couleurs pouvant être associées aux articles. Tout comme pour la gestion des coloris d'un article, un bouton en forme de croix se trouve à droite de chaque coloris. En cliquant dessus, l'utilisateur peut donc supprimer le coloris correspondant.

Tous les articles qui étaient associés à ce coloris perdront alors leur relation, mais ne seront pas supprimés.

5 - Limitations et perspectives

5.1 - Ajout de nouvelles fonctionnalités

Pour l'instant, la gestion des tailles disponibles pour les articles se fait via une liste de chaînes de caractères (String). Les différentes tailles sont fixes, et ne peuvent donc pas être modérées par un administrateur. Il pourrait être judicieux d'ajouter une fonctionnalité pour la gestion des tailles (modèle Size par exemple).

5.2 - Amélioration de l'UI / UX

Les différentes interfaces utilisateurs, bien que responsives, manquent d'originalité et de personnalité. En effet, celles-ci ont été en grande partie réalisées à l'aide du Framework Bootstrap, afin de gagner du temps. Elles mériteraient d'être mises à jour.

6 - Conclusion

Les objectifs du cahier des charges ont été atteints. Grâce à cette application, les utilisateurs peuvent visualiser les différents articles mis en vente.

Après s'être connectés, les administrateurs peuvent accéder à la gestion des articles et des coloris. Voici un résumé des fonctionnalités mises à dispositions des modérateurs :

- Ajout / modification / suppression d'un article,
- Ajout / suppression d'une image d'un article,
- Ajout / suppression des coloris liés à un article,
- Gestion de la liste des coloris (ajout, suppression).

7 - Annexes

7.1 - Guide développeur

7.1.1 -Récupération du projet

Le projet peut être récupéré à partir du repository github suivant : https://github.com/ThePension/walkingfish_spring.

7.1.2 -Mise en place de la base données

Afin d'utiliser cette application, il est nécessaire de mettre en place une base de données MySQL. Pour ce faire, il est recommandé d'utiliser Docker, téléchargeable à l'adresse suivante : <https://docs.docker.com/desktop/install/windows-install/>

Une fois installé, la commande suivante permet de mettre en place une image MySQL, qui sera démarrée automatiquement :

```
docker run --name mysql -p3306:3306 -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql
```

7.1.3 -Lancement de l'application

Avant de lancer l'application, il est nécessaire de vérifier que la configuration utilisée par l'application soit la bonne. Le fichier « application.properties » devrait pointer vers le profile « mysql », comme présenté ci-dessous :

```
server.port=8080
spring.servlet.multipart.enabled=true
spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=10MB
spring.profiles.include=mysql
```

L'application peut alors être lancée avec les commandes suivantes (à effectuer à la racine du projet).

```
./mvnw clean install
./mvnw spring-boot:run
```

7.1.4 -Compte administrateur

Nom d'utilisateur : *user*

Mot-de-passe : *password*

7.2 - Table des illustrations

FIGURE 1 - SCHÉMA D'INTERACTIONS (USE CASE)	III
FIGURE 2 - MCD (MODÈLE CONCEPTUEL DE DONNÉES).....	IV
FIGURE 3 - MLD (MODÈLE LOGIQUE DE DONNÉES)	IV
FIGURE 4 - SCHÉMAS DES ROUTES	V
FIGURE 5 – PLANIFICATION INITIALE.....	VI
FIGURE 6 - RÉSULTATS : CATALOGUE	11
FIGURE 7 - RÉSULTATS : CONNEXION	12
FIGURE 8 - RÉSULTATS - BARRE DE NAVIGATION ADMIN.....	12
FIGURE 9 - RÉSULTATS : FORMULAIRE D'AJOUT D'ARTICLES (VIDE)	13
FIGURE 10 - RÉSULTATS : FORMULAIRE D'AJOUT D'ARTICLES (REMPLE).....	14
FIGURE 11 - RÉSULTATS : CATALOGUE ACCÈS ADMINISTRATEUR	15
FIGURE 12 - RÉSULTATS : PAGE DE MODIFICATION D'UN ARTICLE	15
FIGURE 13 - RÉSULTATS : FORMULAIRE DE MODIFICATION DES INFORMATIONS PRINCIPALES D'UN ARTICLE	16
FIGURE 14 - RÉSULTATS : PAGE DE GESTION DES COLORIS.....	17

7.3 - Liste des tests unitaires

Nom	Description
testInjectedComponentsAreNotNull	Vérifie que les composants injectés (ArticleRepository et CatalogService) ne sont pas null
testGetAllArticles	Vérifie que la méthode getAllArticlesFromCatalog() renvoie une liste d'articles non vide
testAddArticleToCatalogue	Vérifie que la méthode addArticleToCatalog() ajoute correctement un article à la base de données
testGetArticleById	Vérifie que la méthode getArticleById() renvoie le bon article en fonction de l'ID donné
testUpdateArticleInDB	Vérifie que la méthode updateArticleInDB() met à jour correctement un article dans la base de données
testDeleteArticleInDB	Vérifie que la méthode deleteArticleInDB() supprime correctement un article de la base de données
testSavePicture	Vérifie que la méthode savePicture() sauvegarde correctement une image liée à un article dans la base de données

7.4 - Bibliographies et références

7.4.1 - Sites Web

Sauvegarde de fichiers sur le serveur :

<https://stackoverflow.com/questions/27909583/write-a-file-to-resources-folder-in-spring>

<https://stackoverflow.com/questions/18664579/recommended-way-to-save-uploaded-files-in-a-servlet-application>

<https://stackoverflow.com/questions/2422468/how-can-i-upload-files-to-a-server-using-jsp-servlet>

<https://www.bezkoder.com/spring-boot-file-upload/>

<https://stackoverflow.com/questions/10847994/spring-mvc-save-uploaded-multipartfile-to-specific-folder>

<https://stackoverflow.com/questions/28268499/understanding-multipartfile-transferto-method-in-spring-mvc>

Mise en place de la pagination :

<https://www.baeldung.com/spring-thymeleaf-pagination>

Mise en place des tests unitaires :

<https://www.bezkoder.com/spring-boot-unit-test-jpa-repo-datajpa-test/>

Carrousel pour les images d'un article :

<https://splidejs.com/tutorials/thumbnail-carousel/>

Modification de la taille maximale d'upload pour les fichiers

<https://spring.io/guides/gs/uploading-files/#:~:text=max%2Dfile%2Dsize%20is%20set,spring.>