

Documentation technique



Sommaire :

I-	Présentation de l'application	2
II-	Spécificités fonctionnelles	3
1.	Fonctionnalités de base	3
2.	Fonctionnalités réservées	3
III-	Choix techniques	5
1.	Développement	5
2.	Base de données	5
3.	Déploiement	5
IV-	Modèle conceptuel de données	5
V-	Requêtes	6
1.	Routes	7
1.1.	Users	7
1.2.	Categories	7
1.3.	Offers	7
1.4.	Products	8
2.	Erreurs	9
VI-	Sécurité	9
1.	Authentification	9
2.	Stockage des données	9
2.1.	Variables d'environnement	9
2.2.	Hashage des mots de passe	9
VII-	Tests	10
VIII-	Evolutions futures	10

I- Présentation de l'application

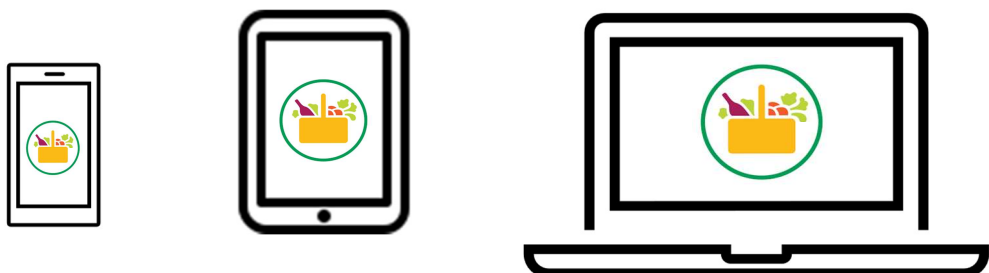
A l'heure où l'écologie doit devenir une priorité pour préserver notre planète, l'enseigne Mercadona souhaite à son tour réduire son impact sur l'environnement.



Désormais, plus aucun tract publicitaire ne sera distribué...

Au lieu de cela, un catalogue simple et pratique à utiliser sera proposé sur l'application Mercadona. Il sera possible de consulter les produits et offres promotionnelles de l'enseigne n'importe où et à n'importe quel moment !

En effet, l'application est disponible au format mobile, tablette ou ordinateur ; de quoi faire oublier le papier rapidement !



En tant qu'utilisateur administrateur, il sera possible d'effectuer différentes actions dans le cadre de la gestion du catalogue ; notamment l'ajout de nouveaux produits au catalogue, de nouvelles catégories ou encore la création de nouvelles promotions.

Le détail des fonctionnalités de l'application sera fait dans la section suivante.

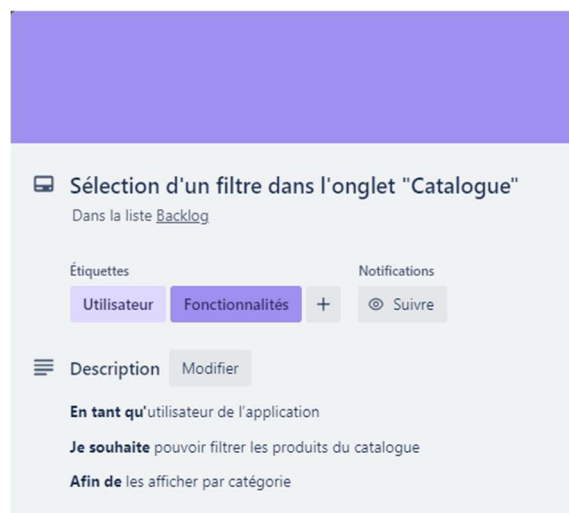
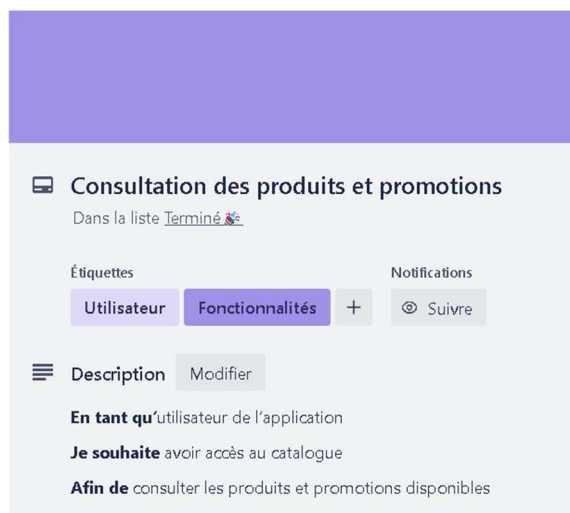
II- Spécificités fonctionnelles

Les fonctionnalités de l'application sont de deux types : les fonctionnalités de base et les fonctionnalités réservées.

1. Fonctionnalités de base

Les fonctionnalités de base sont accessibles par l'ensemble des utilisateurs et sont les suivantes :

- Consultation du catalogue de produits
- Consultation des produits en promotion
- Affichage des produits par catégorie
- Consultation du guide d'utilisation de l'application



2. Fonctionnalités réservées

Les fonctionnalités réservées sont uniquement accessibles après authentification par l'administrateur de l'application et sont les suivantes :

- Connexion à l'espace administrateur
- Consultation, ajout, modification et suppression de produits
- Application d'une promotion à un produit
- Consultation, ajout, modification et suppression de catégories


- Consultation, ajout, modification et suppression de promotions

[illegible]

Création d'une catégorie par l'administrateur

Dans la liste [Terminé](#)

Étiquettes	Notifications
Fonctionnalités Administrateur	+ Suivre

 Description Modifier

**En tant qu'administrateur de l'application

Je souhaite pouvoir créer une nouvelle catégorie

Afin de pouvoir afficher les produits par catégorie par la suite

The screenshot shows the 'Création d'un produit' (Create a product) page in the Backlog application. The page has a purple header bar. Below the header, the title 'Création d'un produit par l'administrateur' is displayed with a folder icon. Underneath, it says 'Dans la liste Backlog'. The main content area has a light gray background. On the left, there's a sidebar with a hamburger menu icon and the word 'Description'. To the right of 'Description' is a 'Modifier' button. The main area contains a form with two tabs: 'Fonctionnalités' and 'Administrateur', with the latter being selected and highlighted in purple. To the right of the tabs is a '+' button. Further right, there's a 'Notifications' section with a 'Suivre' button featuring an eye icon. Below this, there's a section titled 'Propriétés d'un produit :'. Under this title, there's a list of four items: 'Libellé', 'Catégorie', 'Description', and 'Image', each with a radio button next to it. The 'Image' item is currently selected. At the bottom, there's a 'Prix' label with a radio button next to it.

The screenshot shows a web interface for creating a promotion. At the top, there's a purple header bar. Below it, the page title 'Création d'une promotion par l'administrateur' is displayed in a large, bold font. Under the title, the text 'Dans la liste [Backlog](#)' is shown. The main content area has a light gray background. On the left, there's a sidebar with a 'Description' tab selected. The main area contains a form with several fields: 'Étiquettes' (Tags) with a dropdown menu showing 'Fonctionnalités' and 'Administrateur'; 'Notifications' with a dropdown menu showing 'Suivre'; and a 'Date de début' field with a date picker. Below these fields, there's a section titled 'Propriétés d'une promotion :' with a list of properties: 'Date de début', 'Date de fin', and 'Pourcentage de remise'.

III- Choix techniques

1. Développement

Mercadona est une application ASP .NET Core développée en C#.

Le choix de la technologie employée est dû à la spécialité du développeur qui a conçu l'application.

2. Base de données

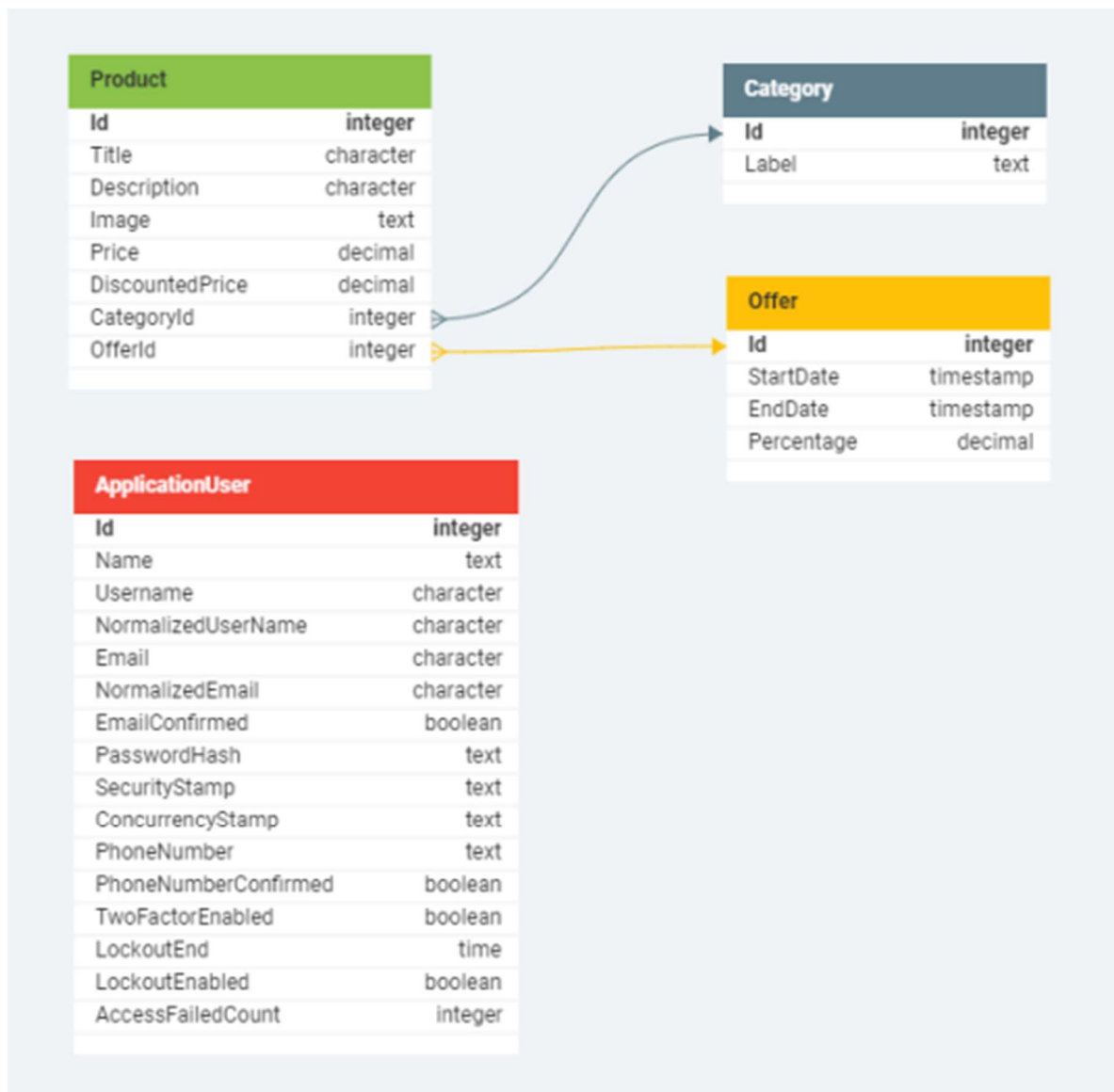
S'agissant des données issues de l'application, elles sont stockées dans une base de données PostgreSQL afin de répondre aux exigences du client.

3. Déploiement

L'application est déployée sur Heroku grâce à un conteneur Docker.

IV- Modèle conceptuel de données

Le modèle conceptuel de données suivant illustre les différentes entités utilisées dans le projet, ainsi que les relations qui les lient.



V- Requêtes

Les requêtes permettant la transmission des données entre le front-end et le back-end sont classées en quatre catégories, qui correspondent à chacune des entités de l'application.

1. Routes

1.1. Users

Méthode	Route	Corps de la demande	Fonction
POST	/users /login	{ email : string, password : string }	Connexion au compte administrateur

1.2. Categories

Méthode	Route	Corps de la demande	Fonction
GET	/categories		Liste l'ensemble des catégories
POST	/categories/create	{ Label : string }	Crée une nouvelle catégorie
POST	/categories/edit/id	{ Label : string }	Modifie la catégorie avec l'id correspondant
POST	/categories /delete/id		Supprime la catégorie avec l'id correspondant

1.3. Offers

Méthode	Route	Corps de la demande	Fonction
GET	/offers		Liste l'ensemble des promotions
POST	/offers/create	{ StartDate : DateTime, EndDate : DateTime, Percentage : decimal }	Crée une nouvelle promotion

POST	/offers/edit/id	{ StartDate : DateTime, EndDate : DateTime, Percentage : decimal }	Modifie la promotion avec l'id correspondant
POST	/offers/delete/id		Supprime la promotion avec l'id correspondant

1.4. Products

Méthode	Route	Corps de la demande	Fonction
POST	/category.id		Liste les produits par catégorie
GET	/products		Liste l'ensemble des produits
POST	/products/create	{ Title : string, CategoryId : int, Description : string, Image : string, Price : decimal, OfferId : int }	Crée un nouveau produit
POST	/products/edit/id	{ Title : string, CategoryId : int, Description : string, Image : string, Price : decimal, OfferId : int }	Modifie le produit avec l'id correspondant
POST	/products/delete/id		Supprime le produit avec l'id correspondant

2. Erreurs

Erreurs possibles	
Code erreur	Description de l'erreur
400 Mauvaise requête	Le serveur n'a pas pu comprendre la requête à cause d'une syntaxe invalide
401 Non autorisé	Une identification est nécessaire pour obtenir la réponse demandée
404 Non trouvé	Le serveur n'a pas trouvé la ressource demandée
500 Erreur de serveur	Le serveur rencontre une erreur qu'il ne sait pas traiter

VI- Sécurité

1. Authentification

Pour plus de sécurité, toutes les routes de l'API Mercadona relatives à la gestion du catalogue nécessitent une authentification.

Avec ASP.NET Core, cette authentification est gérée par l'attribut `[Authorize]`.

Par conséquent, seul l'administrateur de l'application qui s'est authentifié peut accéder fonctionnalités de gestion du catalogue.

2. Stockage des données

2.1. Variables d'environnement

Des variables d'environnement ont été créé afin, notamment, de stocker la chaîne de connexion à la base de données de manière sécurisée.

En effet, le repository GitHub du projet étant public, cela permet de ne pas exposer directement les données sensibles de l'application.

2.2. Hashage des mots de passe

Les données sensibles de l'application et notamment le mot de passe de connexion de l'administrateur est stocké dans la base de données sous forme de hash afin de le rendre illisible et ainsi le protéger en cas d'attaque.

VII- Tests

Les tests unitaires réalisés portent sur le CategoriesController et concernent plus précisément la récupération de la liste des catégories contenue dans la base de données.

Un test permet de contrôler que le résultat obtenu est bien de type `<List<Category>>` (liste de catégories).

Le second permet de comparer le nombre de résultats obtenus (nombre de catégories) au nombre total de catégories contenues dans la base de données.

Les tests couvrent environ 10% du code total.

VIII- Evolutions futures

A court terme, les tests unitaires devront être complétés afin de couvrir un maximum de code et ainsi limiter le risque d'erreurs.

Par ailleurs, l'application devra bien entendu être déployée afin de la rendre disponible en ligne. Le choix de la méthode de déploiement et de l'hébergeur étant un aspect important du point de vue de la sécurité de l'application, il ne doit donc pas être négligé.

A moyen terme, des améliorations en termes d'accessibilité de l'application seraient envisageables et souhaitables.

En effet, l'application pourrait être améliorée sur le plan visuel (couleurs, typographies) pour permettre à un maximum de personnes de pouvoir l'utiliser.

Enfin en ce qui concerne l'aspect pratique de l'application, il pourrait être envisagé de développer des fonctionnalités supplémentaires à destination des utilisateurs de base (non administrateur).

On pourrait notamment imaginer une fonctionnalité qui permettrait à l'utilisateur d'ajouter des produits en favoris ou dans une liste d'achat par exemple.