
Spécification Des Exigences Fonctionnelles et Non-fonctionnelles

Site E-commerce

SportsAllShop

Version 1.0

15 octobre 2025

Table des matières

1	Introduction	2
1.1	Objet du document	2
1.2	Portée	2
1.3	Définitions et acronymes	2
2	Objectifs du projet	2
2.1	Besoins métier	2
2.2	Public cible	2
3	Exigences fonctionnelles	3
3.1	Fonctionnalités principales	3
3.1.1	Gestion des produits (CRUD)	3
3.1.2	Catalogue de produits	3
3.1.3	Panier d'achat	4
3.1.4	Gestion des commandes	4
3.1.5	Authentification	5
3.1.6	Gestion des clients	5
3.2	Interfaces utilisateur	5
3.3	Intégrations	6
4	Exigences non fonctionnelles	6
4.1	Performance	6
4.2	Sécurité	6
4.3	Accessibilité	7
4.4	Compatibilité	7
4.5	Convivialité	7
5	Contraintes	7
6	Hypothèses	8
7	Annexes	8

1 Introduction

1.1 Objet du document

Ce document décrit les spécifications fonctionnelles du site e-commerce **getAll**. Il détaille les fonctionnalités attendues, les besoins des utilisateurs, et les contraintes techniques pour guider le développement et assurer l'alignement entre les parties prenantes.

1.2 Portée

Le projet consiste à développer un site e-commerce, **getAll**, permettant la vente de produits organisés par catégories (vêtements, électronique, accessoires). Le site inclut un catalogue de produits, un panier d'achat fonctionnel, un système de gestion des commandes, des clients, et des produits, une authentification basique, un tableau de bord pour le suivi des commandes, et un système d'envoi réel des commandes.

1.3 Définitions et acronymes

Terminologie

Utilisateur

Client final effectuant des achats ou administrateur gérant le site.

CMS

Système de gestion de contenu pour gérer les produits.

CRUD

Create, Read, Update, Delete (opérations de gestion des données).

Panier

Liste des produits sélectionnés par l'utilisateur avant validation de la commande.

Laravel

Framework PHP utilisé pour le développement du backend.

2 Objectifs du projet

2.1 Besoins métier

Le site e-commerce **getAll** doit répondre aux exigences suivantes :

- Permettre aux clients de consulter un catalogue de produits, d'ajouter/supprimer des produits dans un panier, et de passer des commandes avec un envoi réel.
- Offrir aux administrateurs la possibilité de gérer les produits (ajout, édition, suppression) et de suivre les commandes.
- Assurer une expérience utilisateur fluide et conviviale avec une validation côté client et un calcul automatique des prix.

2.2 Public cible

Clients : Utilisateurs souhaitant acheter des produits en ligne.

Administrateurs : Personnes responsables de la gestion des produits, des commandes, et des comptes clients.

3 Exigences fonctionnelles

3.1 Fonctionnalités principales

3.1.1 Gestion des produits (CRUD)

F1.1 - Ajout de produits

Les administrateurs peuvent ajouter un produit avec les informations suivantes : nom, image, description, prix, stock, catégorie (ex. : vêtements, électronique, accessoires).

Validation des champs (ex. : prix positif, stock non négatif).

F1.2 - Édition de produits

Modification des informations d'un produit existant (nom, image, description, prix, stock, catégorie).

F1.3 - Suppression de produits

Suppression d'un produit du catalogue, avec confirmation avant suppression.

F1.4 - Organisation par catégorie

Les produits sont organisés en catégories (vêtements, électronique, accessoires).
Les utilisateurs peuvent filtrer les produits par catégorie.

3.1.2 Catalogue de produits

F2.1 - Page catalogue

Affichage automatique des produits depuis la base du CMS ou la base de données Laravel.

Chaque produit affiche : image, nom, prix, et un bouton "Ajouter au panier".

F2.2 - Fiche produit

Page dédiée pour chaque produit avec : photo, nom, description, prix, stock, et bouton "Ajouter au panier".

Affichage d'un message si le produit est en rupture de stock.

3.1.3 Panier d'achat**F3.1 - Ajout de produits**

Les utilisateurs peuvent ajouter un produit au panier depuis la page catalogue ou la fiche produit.

Mise à jour dynamique du panier avec JavaScript.

F3.2 - Suppression de produits

Les utilisateurs peuvent supprimer un produit du panier.

Mise à jour immédiate de l'affichage.

F3.3 - Calcul du prix total

Calcul automatique du prix total des produits dans le panier, mis à jour en temps réel.

F3.4 - Prévisualisation du panier

Affichage d'une page ou d'un panneau récapitulatif du panier (produits, quantités, prix total) avant validation.

3.1.4 Gestion des commandes**F4.1 - Formulaire de commande**

Formulaire avec validation côté client pour : nom, email, adresse.

Validation des champs (ex. : format d'email valide, champs obligatoires remplis).

F4.2 - Envoi de commande

Soumission du formulaire pour enregistrer la commande dans la base de données via Laravel.

Mise à jour du stock des produits après validation de la commande.

Envoi d'une confirmation par email au client avec les détails de la commande (produits, prix total, adresse de livraison) via le système d'email de Laravel (ex. : Mailer).

F4.3 - Tableau de bord des commandes

Interface pour les administrateurs permettant de visualiser les commandes (statut, client, date, montant total, détails des produits).

Possibilité de marquer une commande comme "traitée" ou "expédiée" via une interface Laravel.

3.1.5 Authentification**F5.1 - Inscription**

Formulaire pour créer un compte avec email et mot de passe, géré par Laravel (ex. : Laravel Authentication).

Validation côté client et serveur (ex. : mot de passe d'au moins 8 caractères).

F5.2 - Connexion

Formulaire de connexion avec email et mot de passe, géré par Laravel.

Gestion des erreurs (ex. : identifiants incorrects).

F5.3 - Déconnexion

Option pour l'utilisateur de se déconnecter via Laravel.

3.1.6 Gestion des clients**F6.1 - Profil client**

Les clients peuvent consulter et modifier leurs informations (nom, email, adresse) via une interface Laravel.

F6.2 - Historique des commandes

Affichage des commandes passées par le client (date, produits, montant, statut) via une requête Laravel.

3.2 Interfaces utilisateur**Page d'accueil**

Présentation des catégories et produits phares.

Page catalogue

Liste des produits avec filtres par catégorie.

Fiche produit

Détails d'un produit avec bouton "Ajouter au panier".

Panier

Interface récapitulative avec options de modification et validation.

Formulaire de commande

Champs pour les informations de livraison et paiement.

Tableau de bord administrateur

Vue des commandes, produits, et clients, implémentée avec Laravel (ex. : Blade ou API REST pour le frontend).

Pages d'authentification

Formulaires d'inscription et de connexion.

3.3 Intégrations

- **CMS** : Utilisation d'un CMS (ex. : Strapi) pour gérer le contenu des produits (nom, image, description, prix, stock, catégorie), avec une intégration via API REST à Laravel.
- **Base de données** : Stockage des commandes, clients, et produits via Laravel (ex. : MySQL ou PostgreSQL).
- **Système d'email** : Envoi de confirmations de commande via Laravel Mailer (ex. : intégration avec SMTP, SendGrid).
- **Exportation** : Possibilité d'exporter les commandes ou produits au format CSV via une fonctionnalité Laravel (optionnel).

4 Exigences non fonctionnelles

4.1 Performance

- Temps de chargement des pages inférieur à 2 secondes sous une charge normale.
- Capacité à gérer 100 utilisateurs simultanés.
- Mise à jour du stock en temps réel après chaque commande via Laravel.

4.2 Sécurité

- Validation côté client et serveur (via Laravel) pour les formulaires (inscription, commande).
- Chiffrement des mots de passe via Laravel (ex. : Bcrypt).
- Respect du RGPD pour la gestion des données clients (email, nom, adresse).
- Sécurisation des communications (HTTPS).

4.3 Accessibilité

Respect des normes WCAG 2.1 pour l'accessibilité (ex. : contraste, navigation au clavier).

4.4 Compatibilité

- Compatible avec les navigateurs modernes (Chrome, Firefox, Safari, Edge).
- Responsive design pour une utilisation sur mobile, tablette, et ordinateur.

4.5 Convivialité

Expérience utilisateur : Le site doit offrir une interface intuitive et conviviale, avec une navigation fluide et des interactions naturelles.

Esthétique des pages : Les pages doivent être visuellement attrayantes, avec une mise en page claire et moderne.

Couleurs : Utilisation d'une palette de couleurs harmonieuse et professionnelle, adaptée à l'identité visuelle du site (ex. : couleurs sobres pour les sections administratives, couleurs vibrantes pour les pages client).

Polices de texte : Utilisation de polices lisibles et modernes (ex. : Roboto, Open Sans) avec des tailles adaptées pour une lecture confortable sur tous les appareils.

Consistance visuelle : Maintien d'une charte graphique cohérente à travers toutes les pages (boutons, icônes, espacements).

5 Contraintes

Technologiques

Frontend : HTML, CSS, JavaScript (pour le panier et la validation des formulaires).

Backend : Laravel (framework PHP) pour la gestion des fonctionnalités principales (produits, commandes, clients, authentification). Intégration avec un CMS (ex. : Strapi) via API REST pour la gestion du contenu des produits.

Base de données : MySQL ou PostgreSQL, configurée avec Laravel (Eloquent ORM).

Service d'email : Intégration d'un service pour l'envoi des confirmations via Laravel Mailer (ex. : SMTP, SendGrid).

Délais

Le projet doit être livré dans un délai de 5 semaines à compter du démarrage.

Budget

À définir par le client.

6 Hypothèses

- Les utilisateurs ont accès à un navigateur web moderne.
- Une connexion internet stable est disponible.
- Les images des produits sont fournies ou générées pour le CMS.
- Un service d'email est configuré pour les confirmations de commande via Laravel.

7 Annexes

Maquettes

À fournir ou concevoir (optionnel).

Données fictives

Liste de produits fictifs pour le catalogue (ex. : 10 produits par catégorie).

Références

Documentation Laravel, documentation CMS (ex. : Strapi), guides JavaScript pour le panier, documentation du service d'email, charte graphique pour les couleurs et polices.