

## Projet Pluridisciplinaire

Conception et réalisation d'une application web e-commerce

# Rapport de conception

### Encadreur :

- Mme KLOUCHE Badia

### Equipe du développement :

- AHRES Fatima
- HABIB Leila
- HATHOUT Fatna
- KOUADRI AICHOUCH Rania
- LOURAGHI Aicha
- TAIEB BRAHIM Nafissa

# Table des matières

I.	<b>Liste des figures</b> .....	2
II.	<b>Introduction</b> .....	3
III.	<b>Outils de modélisation</b> .....	3
	III.1. UML : .....	3
	III.2. Draw.io : .....	3
IV.	<b>Architectures adoptés</b> .....	4
	IV.1. Architecture 3-tiers:.....	4
	IV.2. Architecture MVT:.....	4
V.	<b>Diagrammes UML</b> .....	5
	V.1. Diagramme de composants:.....	6
	V.2. Diagramme de déploiement : .....	6
	IV.3. Modèle Relationnel : .....	7
VI.	<b>Maquettage</b> .....	9
	VI.1. Interface “ Accueil “ .....	10
	VI.2. Interface “Catalogue des produits” .....	11
	VI.3. Interface “Détail d’un produit” .....	12
	VI.4. Interface « Catalogue des services ».....	13
	VI.5. Interface « Détail d’un service » .....	14
	VI.6. Interface “Panier d’un client” .....	15
VII.	<b>Conclusion</b> .....	16

## I. Liste des figures

---

Page	Contenu
5	Figure 1 : Architecture MVT
6	Figure 2 : Diagramme de composants
6	Figure 3 : Diagramme de déploiement
8	Figure 4 : Modèle Relationnel E/A
10	Figure 5 : Interface « Accueil »
11	Figure 6 : Interface « Catalogue des produits »
12	Figure 7 : Interface « Détail d'un produit »
13	Figure 8 : Interface « Catalogue des services »
14	Figure 9 : Interface « Détail d'un service »
15	Figure 10 : Interface « Panier d'un client »

## II. Introduction

---

Dans le cadre de notre projet « Gestion d'une application web e-commerce », nous devons concevoir un système de gestion d'un site web de vente en ligne.

En cours de cette étape de conception, nous présentons les principes de la modélisation UML ainsi que les outils choisis dans notre modélisation. Nous abordons aussi les différentes architectures choisies pour notre système ainsi que leurs principes.

La dernière étape consiste à réaliser le diagramme des composants et de déploiement, et le modèle relationnel.

## III. Outils de modélisation

---

### III.1. UML :

UML (Unified Modeling Language) est une méthode de modélisation graphique, elle est utilisée en conception orientée objet et en développement logiciel pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement du système.

Il existe de nombreux logiciels de modélisation UML et dans notre système nous avons utilisé UML designer et ArgoUML dans la représentation des diagrammes.

### III.2. Draw.io :

Draw.io est un outil intégré au Trello Power-Up complètement gratuit qui vous permet de construire des organigrammes, des diagrammes UML, des plans d'étage et bien plus.

## IV. Architectures adoptés

---

L'architecture informatique définit la structuration d'un système informatique (matériel et logiciel) en termes de composants et d'organisation de ses fonctions.

Le modèle d'architecture ne décrit pas ce que doit réaliser un système informatique mais plutôt comment il doit être conçu de manière à répondre aux spécifications. Pour cette raison on vous détaille ci-dessous les différentes architectures suivies :

### IV.1. Architecture 3-tiers:

La technologie objet requiert une architecture qui organise les interactions entre objets. Cette architecture appelée : architecture à trois niveaux ou architecture à trois couches est divisée en :

- la présentation des données (Application) : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur.
- le traitement métier des données (Serveur) : correspondant à la mise en oeuvre de l'ensemble des règles de gestion et de la logique applicative.
- l'accès aux données persistantes (Base de données) : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

### IV.2. Architecture MVT:

L'architecture utilisée par Django diffère légèrement de l'architecture MVC classique. En effet, Django gère lui-même la partie Controller (gestion des requêtes du client, droit sur les actions...), ainsi on parle plutôt de Framework utilisant l'architecture MVT : Modèle-Vue-Template. La figure suivante illustre son fonctionnement.

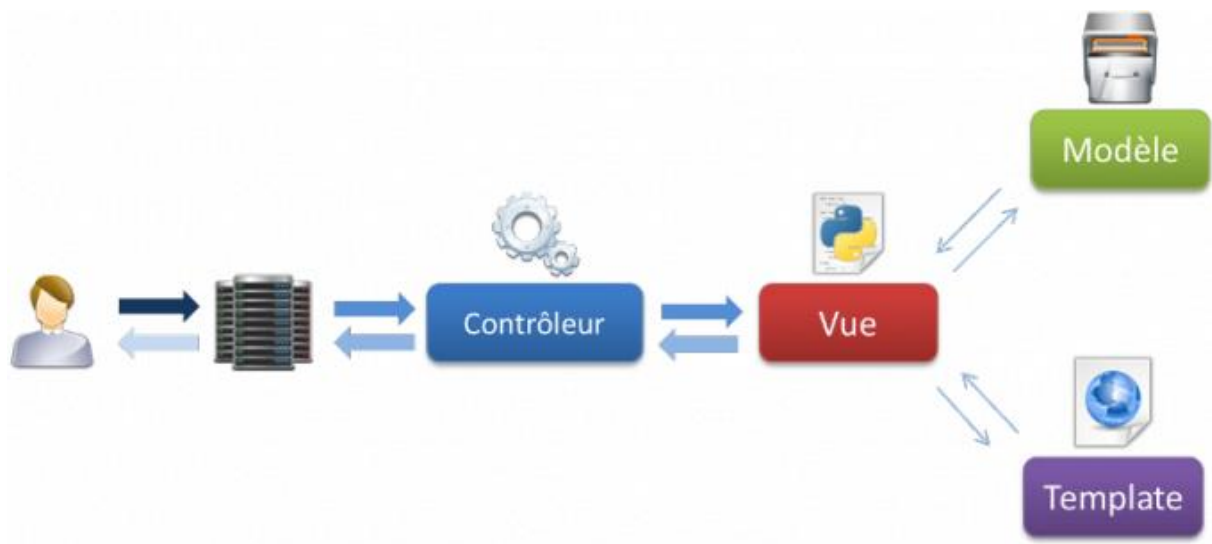


Figure 1. Architecture MVT

Concrètement, lorsque l'internaute appelle une page de l'application web réalisé avec Django, le Framework se charge, via les règles de routage URL définies, d'exécuter la vue correspondante. Cette dernière récupère les données des modèles et génère un rendu HTML à partir du Template et de ces données. Une fois la page générée, l'appel fait chemin arrière, et le serveur renvoie le résultat au navigateur de l'internaute.

## V. Diagrammes UML

Les diagrammes de composants et les diagrammes de déploiement sont les deux derniers types de vues statiques en UML. Les premiers décrivent le système modélisé sous forme de composants réutilisables et mettent en évidence leurs relations de dépendance. Les seconds se rapprochent encore plus de la réalité physique, puisqu'ils identifient les éléments matériels (PC, Modem, Station de travail, Serveur, etc.), leur disposition physique (connexions) et la disposition des exécutables (représentés par des composants) sur ces éléments matériels.

## V.1. Diagramme de composants:

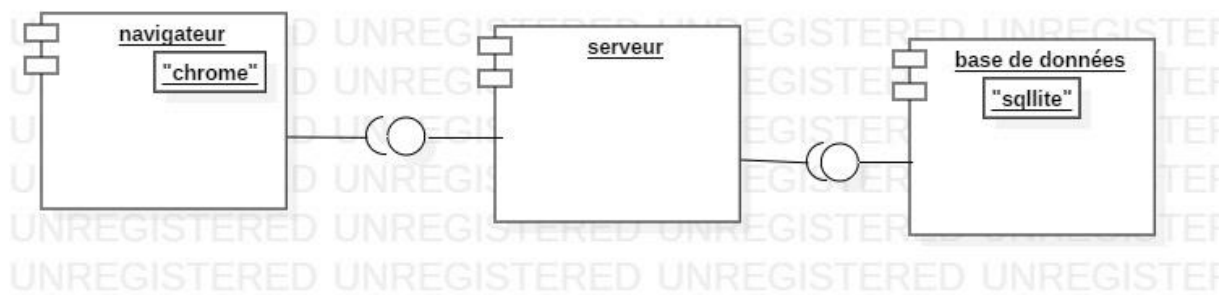


Figure 2. Diagramme de composants

## V.2. Diagramme de déploiement :

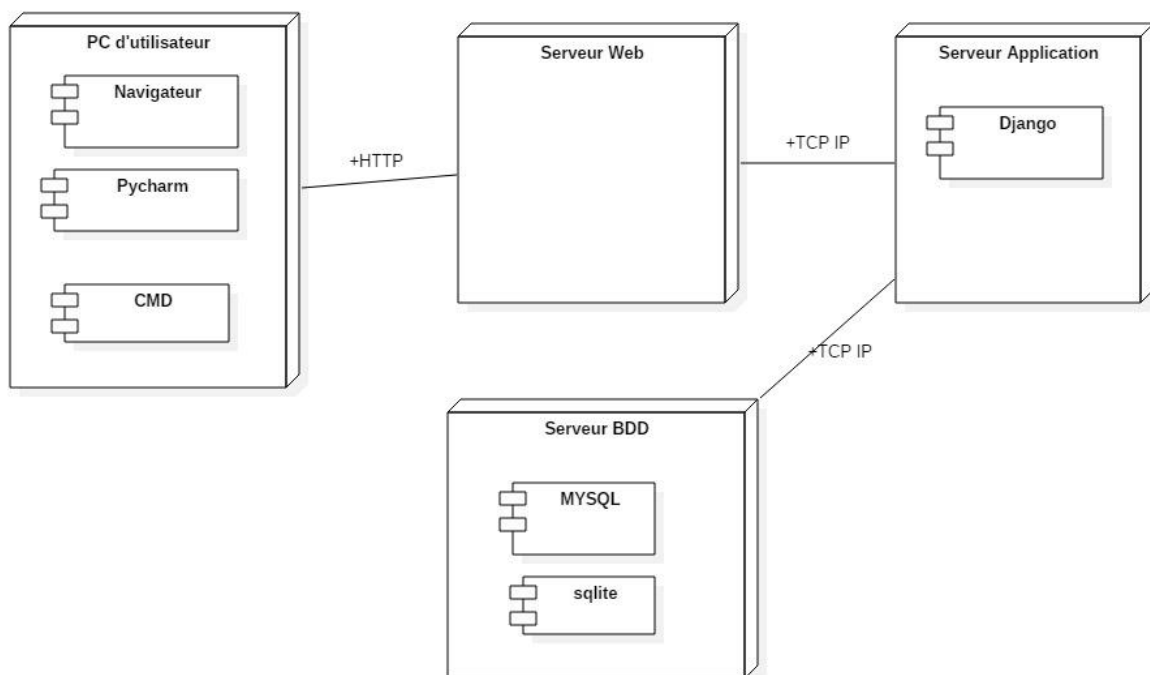


Figure 3. Diagramme de déploiement

### IV.3. Modèle Relationnel :

Le modèle relationnel est basé sur une organisation des données sous forme de tables. La manipulation des données se fait selon le concept mathématique de relation de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle.

Elle est constituée d'un ensemble d'opérations formelles sur les relations.

Le modèle relationnel de notre système est donné par le schéma des tables de la base de données suivantes :

- Produit (idProduit, upc, titre, description, prix\_achat, prix\_vente, reduction\_possible, image, seuil\_stock, quantité, #idCatégorie, #idTypeProduit, #idBoutique)
- Service (idService, upc, nom, description, tel, email, ville, wilaya, image, #idCatégorie)
- Catégorie (idCatégorie, nom, description, image)
- Avis (idAvis, score, titre, corps, nb\_votes, état, #idProduit, #idService)
- Type\_produit (idType, nom, nécessité\_score, suivre\_stock)
- Boutique (idBoutique, nom, tel, email, adresse, description, wilaya, #idVendeur)
- Client (idClient, nom, prénom, email, password)
- Vendeur (idVendeur, nom, prénom, email, password)
- Admin\_produit (idAdmin\_produit, nom, prénom, email, password)
- Admin\_service (idAdmin\_service, nom, prénom, email, password)
- Adresse (idAdresse, première\_ligne, deuxième\_ligne, code\_postal, ville, wilaya, pays)



- Commande (idCommande, num\_commande, date, nb\_produits, #idFacture)
- Facture (idFacture, num\_facture, prix\_total, #idPaiement, #idAdresse)
- Panier (#idCommande, #idClient)
- Paiement (idPaiement, nom, type, date, total\_prix)
- Gestion\_Produit (#idProduit, #idAdmin\_produit)
- Gestion\_Service (#idService, #idAdmin\_Service)
- Livraison (idLivraison, type, date\_réception, #idCommande, #idAdresse)

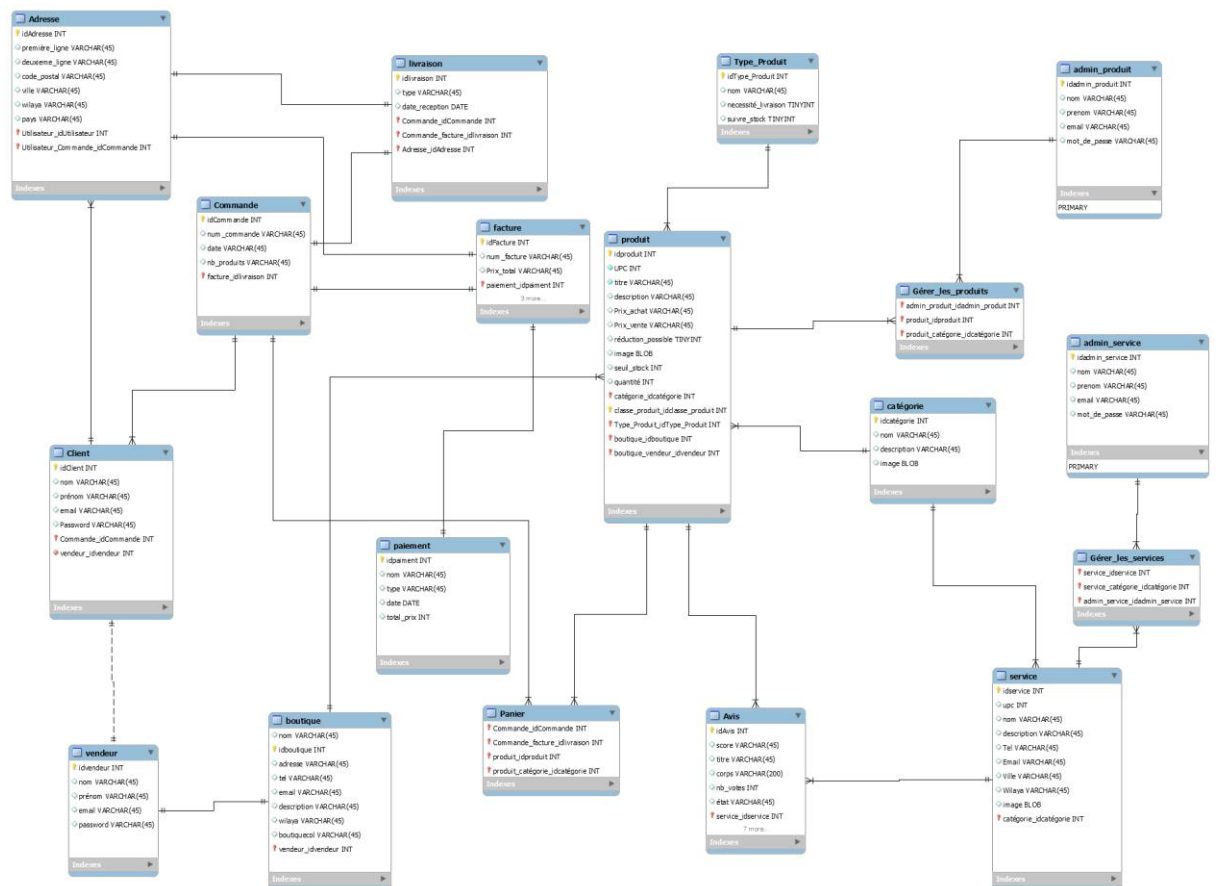


Figure 4. Modèle Relationnel E/A

## VI. Maquettage

---

En ergonomie informatique, le maquettage papier (ou prototypage) est une technique encore très utilisée dans la conception centrée sur l'utilisateur. Cette technique aide les parties prenantes d'un projet à concevoir un logiciel ou autre qui répondant aux attentes et besoins des utilisateurs finaux.

La maquette IHM peut être très bien consistée en un ensemble de dessins produits par un logiciel de présentation ou de dessin.

C'est un produit jetable permettant aux utilisateurs d'avoir une vue concrète, mais non définitive de la future interface de l'application.

Effectué suffisamment tôt, le prototypage permet de détecter les problèmes d'utilisabilité et de vérifier les besoins des utilisateurs avant le codage complet.

Les interfaces de notre système (application web) sont données par les figures ci-dessous :

## VI.1. Interface “Accueil”

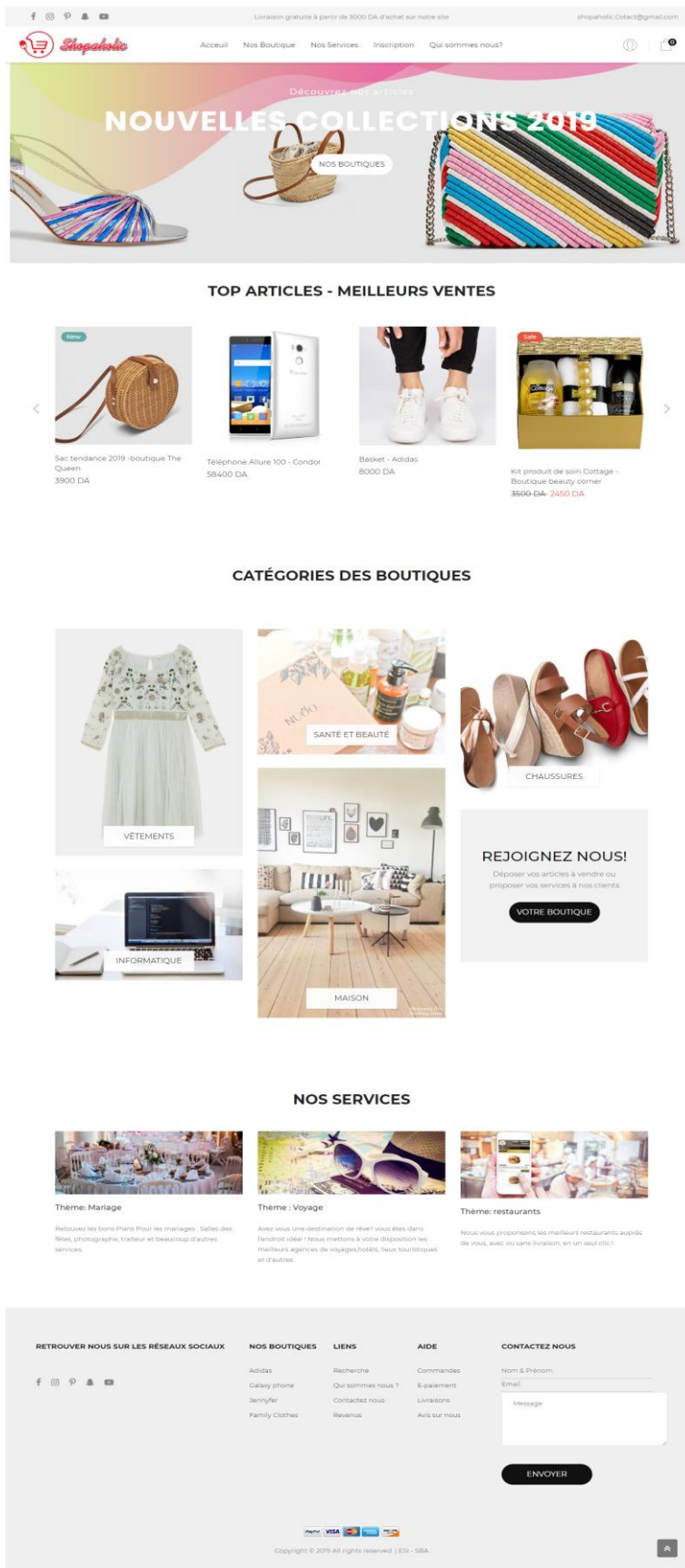


Figure 5. Interface Accueil

## VI.2. Interface “Catalogue des produits”

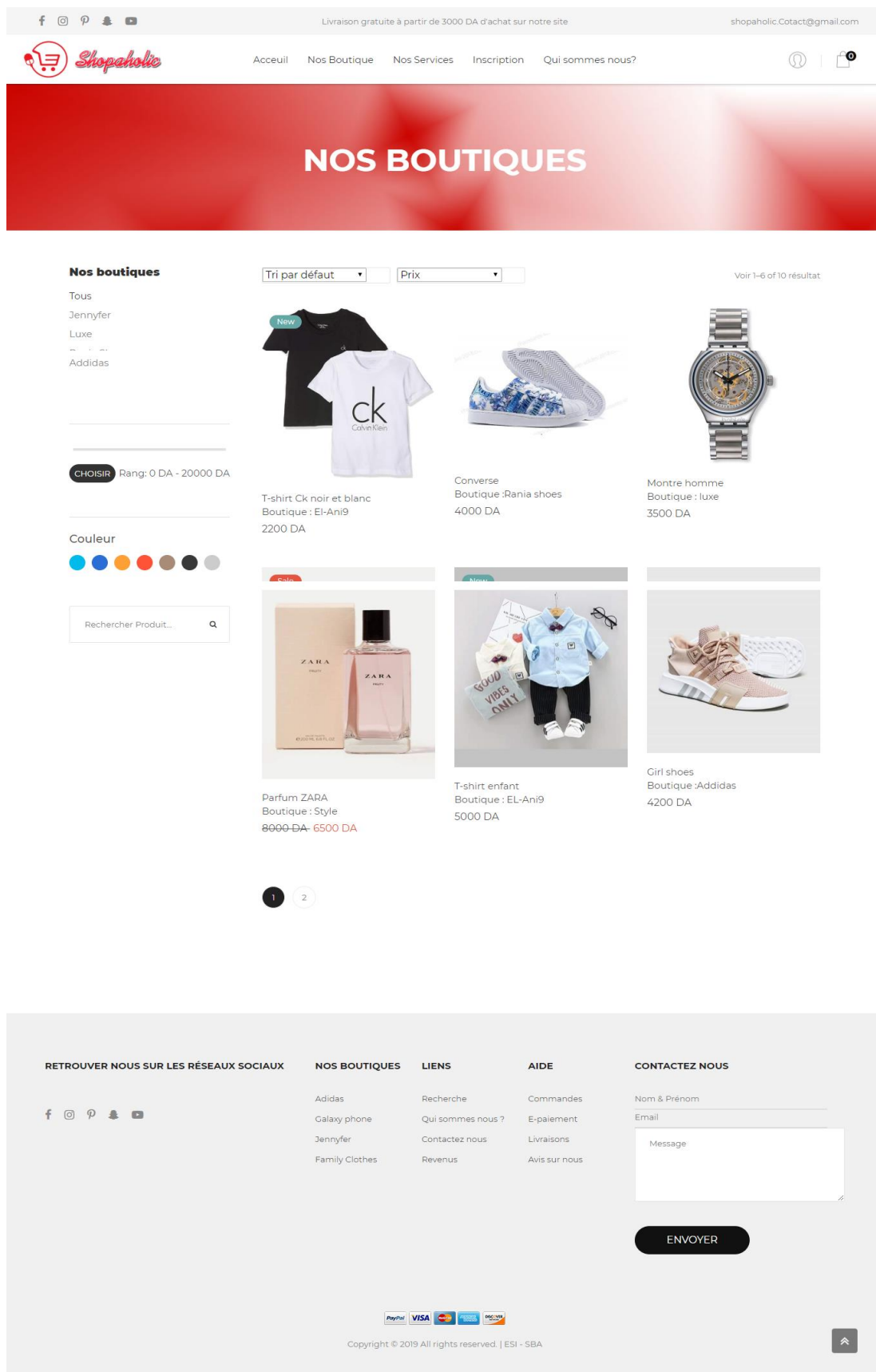


Figure 6. Interface « catalogue des produits »

## VI.3. Interface “Détail d’un produit”

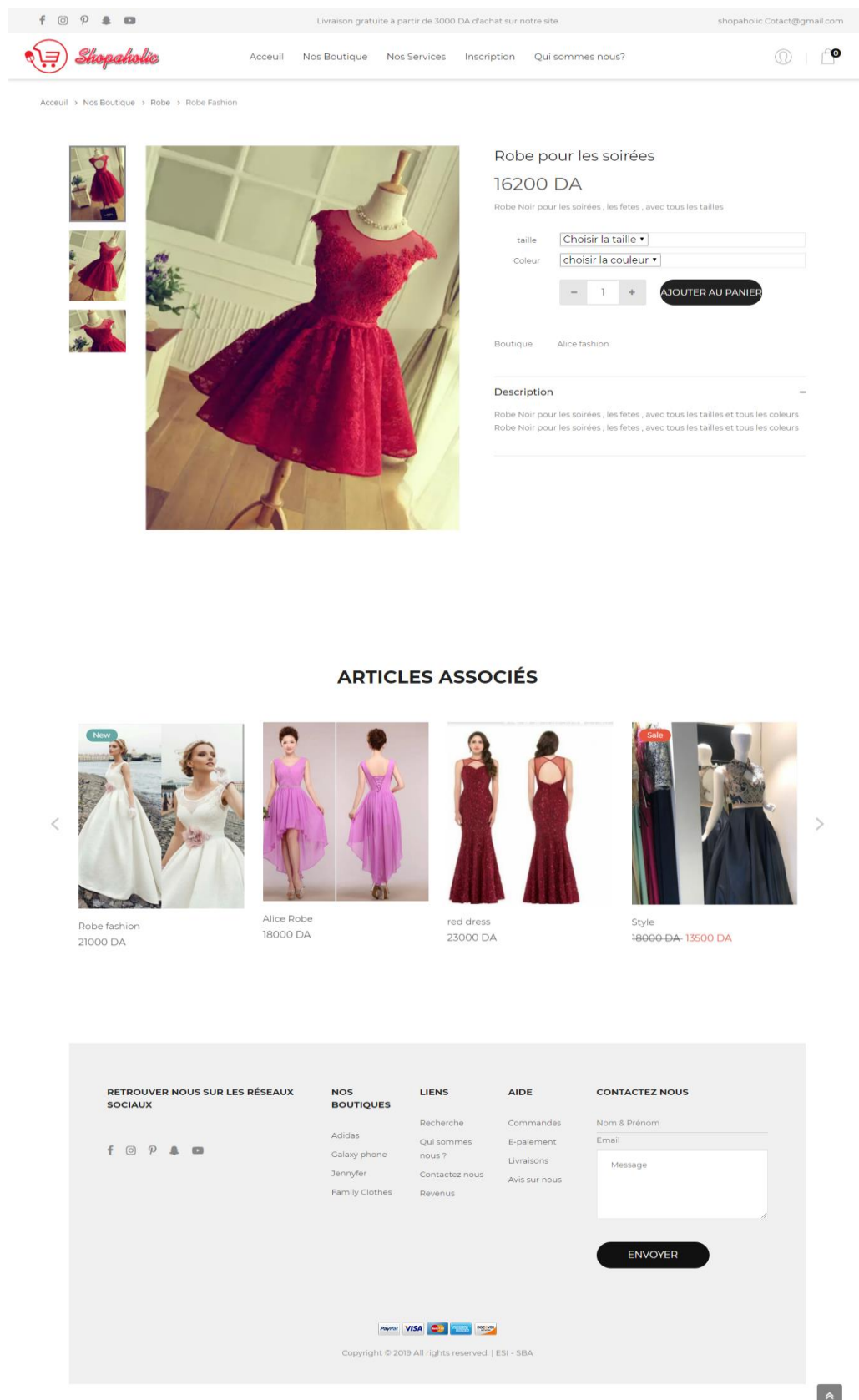


Figure 7. Interface « Détail d’un produit »

## VI.4. Interface « Catalogue des services »

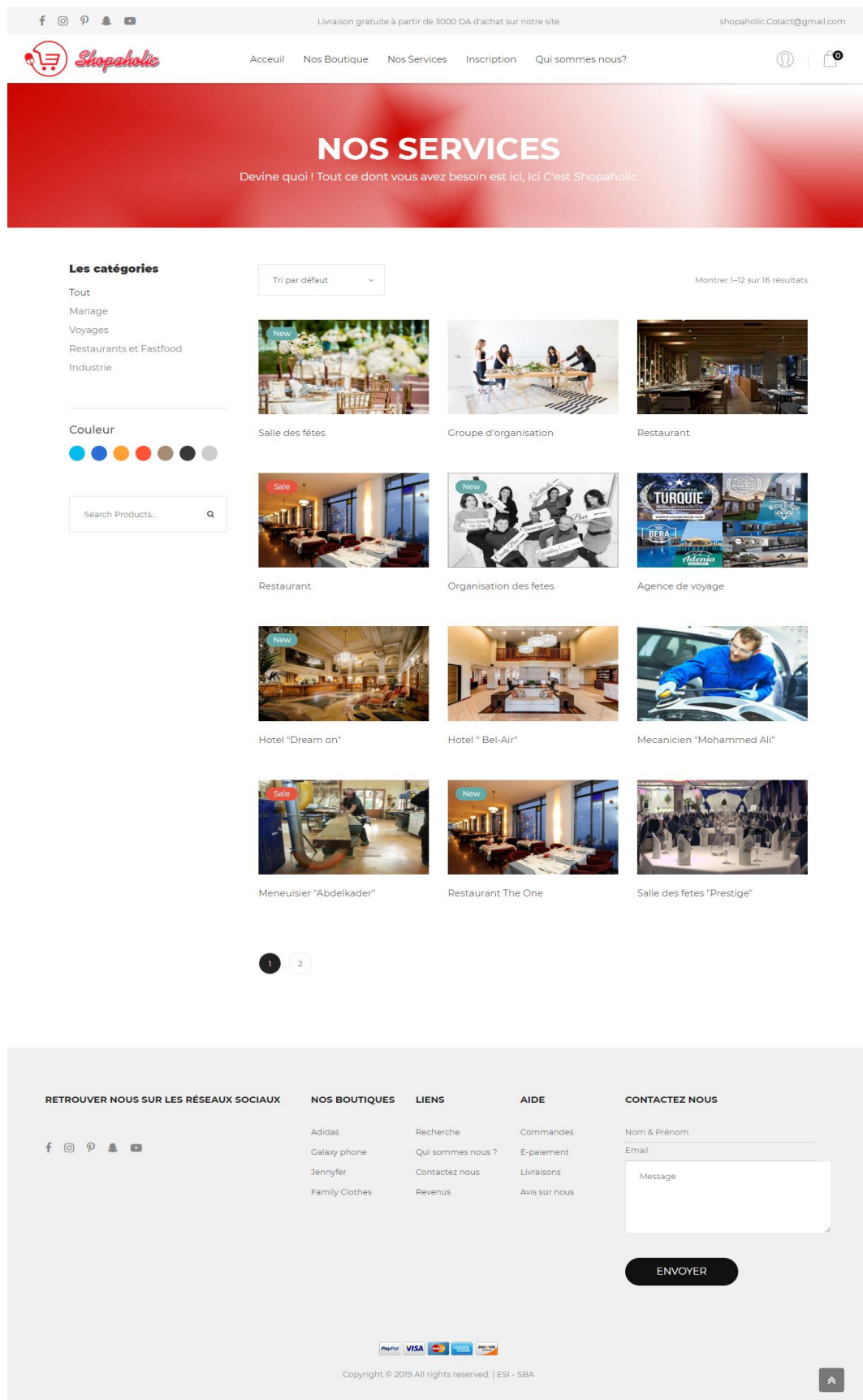


Figure 8. Interface « catalogue des services »



## VI.5. Interface « Détail d'un service »

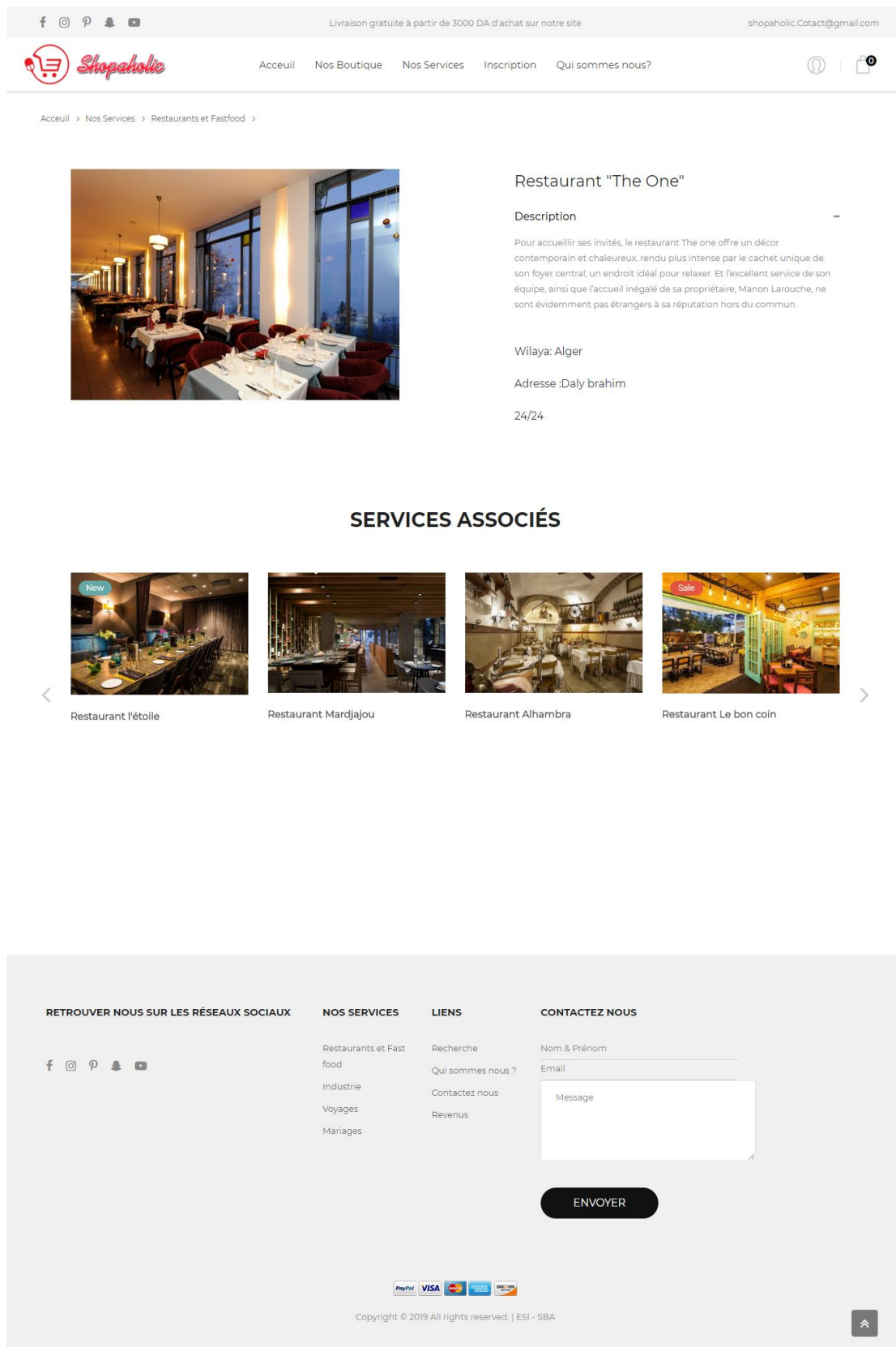


Figure 9. Interface « Détail d'un service »

## VI.6. Interface “Panier d’un client”

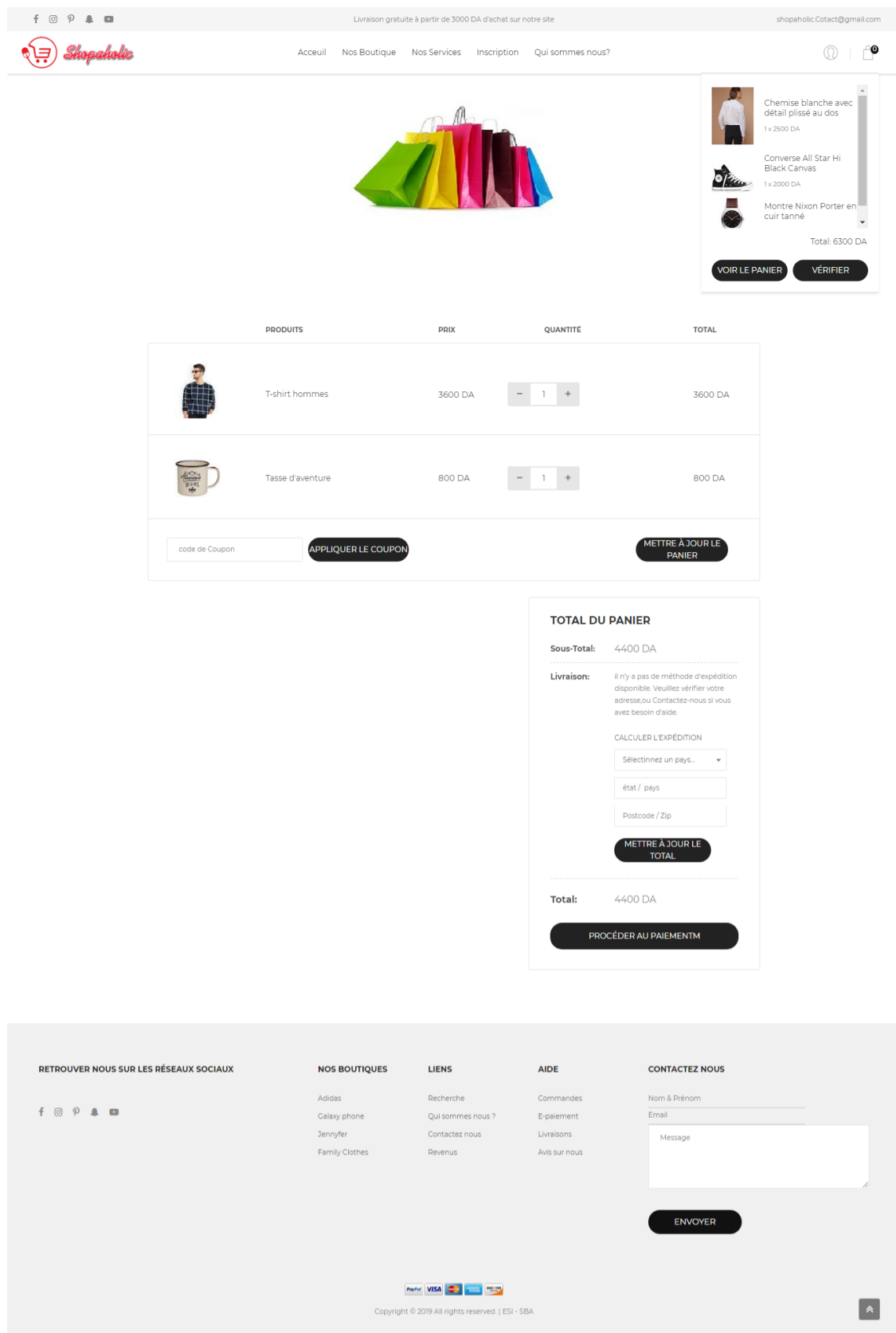


Figure 10. Interface « Panier d'un client »



## VII. Conclusion

---

L'étude conceptuelle a permis de définir la procédure de réaliser la solution d'une façon détaillée à travers le diagramme de classes de conception, où il est introduit les classes métiers et techniques, et le passage au diagramme de composants/déploiement ensuite le modèle relationnel qui a mieux concrétisé et approfondis la compréhension de notre système.