

Université Aix-Marseille site Saint Charles
Faculté des Sciences Exactes



PROJET INFORMATIQUE Appliqué

Thème :

**Conception et Réalisation d'un site web pour la Gestion
de vente de Smartphones**

Cas d'étude : 'MARSPHONIA'

Réalisé par :

Mr. MEBARKI Abdelghani

Mr. GUALANDI Stefan

Mr. KOUACHI Abdeldjalil

Encadré par :

M^{me}. Yi REN

Membre de jury :

Mr. Kevin Perrot

M^{me}. Yi REN



MARSPHONIA
site de vente des smartphones

Année : 2016/2017

Table des matières

Introduction générale

1- CADRE GENERALE

- 1.1 Introduction
- 1.2 Généralités sur les ventes
 - 1.2.1 La vente.
 - 1.2.2 Suivi de vente
 - 1.2.3 Les différents pôles de ventes
 - 1.2.4 L'objectif de la gestion de ventes
- 1.3 Cadre de projet
- 1.4 Présentation du projet
- 1.5 Organigramme de l'application
- 1.6 Processus 2TUP
- 1.7 L'architecture MVC dans le développement de notre site
- 1.8 Conclusion

2- SPECIFICATION DES BESOINS

- 2.1 Introduction
- 2.2 Spécifications des besoins
 - 2.2.1 Description des besoins
 - 2.2.1.1 Besoins fonctionnels
 - 2.2.1.2 Besoins non fonctionnels
 - 2.2.2 Modélisation des besoins
 - 2.2.2.1 Identification des acteurs
 - 2.2.2.2 Identification des cas d'utilisation
 - 2.2.2.3 La description des cas d'utilisation
 - La description du cas d'utilisation 'Authentification'
 - La description du cas d'utilisation 'Gestion des produits'
 - La description du cas d'utilisation 'Ajouter un produit'
 - La description du cas d'utilisation 'Modifier un produit'
 - La description du cas d'utilisation 'Supprimer un produit'
 - La description du cas d'utilisation 'Consulter le stock'
 - La description du cas d'utilisation 'Création d'un bon de commande'
 - 2.2.2.4 Présentation des diagrammes des cas d'utilisation
 - 2.2.2.4.1 Le diagramme de cas d'utilisation associé à l'administrateur
 - 2.2.2.4.2 Le diagramme global des cas d'utilisation
- 2.3 Conclusion

3- ANALYSE ET CONCEPTION

- 3.1 Introduction
- 3.2 Diagrammes de séquence

- 3.2.1 Présentation générale des concepts des diagrammes de séquences
 - 3.2.1.1 la ligne de vie des objets
 - 3.2.1.2 Barre d'activation
 - 3.2.1.3 Types de messages
 - Message synchrone
 - Message asynchrone
 - 3.2.1.4 Fragment d'interaction
 - Opérateur alt
 - Opérateur loop
 - Opérateur opt
 - Opérateur ref
- 3.2.2 Diagrammes de séquences de l'application à réaliser
 - 3.2.2.1 Diagramme de séquence du cas d'utilisation 'Authentification'
 - 3.2.2.2 Diagramme de séquence du cas d'utilisation 'ajouter un produit'
 - 3.2.2.3 Diagramme de séquence du cas d'utilisation 'modifier un produit'
 - 3.2.2.4 Diagramme de séquence du cas d'utilisation 'supprimer un produit'
- 3.4 Diagramme MCD
- 3.5 Dictionnaire de données
- 3.6 Passage au modèle relationnel
 - Traitement des classes
 - Traitement des associations
 - Agrégation de composition
 - Relation d'héritage
- 3.7 Modèle Relationnel de données
- 3.8 Maquettes
 - 3.8.1 Structure de l'application
 - 3.8.2 La charte graphique
 - La charte graphique de l'interface d'authentification
 - La charte graphique de l'interface d'administrateur
 - La charte graphique de l'interface Smartphones
- 3.9 Conclusion

4- REALISATION

- 4.1 Introduction
- 4.2 Environnement de développement
 - 4.2.1 Environnement matériel
 - 4.2.2 Environnement logiciel
 - NetBeans 8.0.2
 - PlanText
 - PHPMYADMIN
 - 4.2.3 Les langages de développement
 - Html
 - CSS
 - SQL
 - JAVASCRIPT
 - PHP
- 4.3 Présentation des interfaces principale
- 4.4 Conclusion

Figures

Figure 1.1 Organigramme de l'application.

Figure 1.2 schéma du processus 2TUP.

Figure 1.3 schéma de l'architecture MVC.

Figure 2.1 : Cas d'utilisation associé à l'administrateur.

Figure 2.2 Diagramme de cas d'utilisation global.

Figure 3.1 : Diagramme de séquence du scénario "Authentification".

Figure 3.2 : Diagramme de séquence du scénario "Ajouter un produit".

Figure 3.3 : Diagramme de séquence du scénario "Modifier un produit".

Figure 3.4 : Diagramme de séquence du scénario "Supprimer un produit".

Figure 3.5 : Diagramme MCD

Figure 3.6 structure de l'application.

Figure 4.1 : Le mode de fonctionnement du PHP.

Figure 4.2 : l'interface d'authentification.

Figure 4.3 : l'interface de l'espace administrateur.

Figure 4.4 : l'interface d'inscription.

Figure 4.5 : l'interface de l'ajout d'un produit.

Figure 4.6 : l'interface de choix de Smartphones coté client.

Figure 4.7 : l'interface de la page de consultation des Smartphones coté.

Tables

Table 2.1 Liste des cas d'utilisation.

Table 2.2 Cas d'utilisation 'Authentification'.

Table 2.3 Cas d'utilisation 'Ajouter un produit'.

Table 2.4 Cas d'utilisation 'Modifier un produit'.

Table 2.5 Cas d'utilisation 'Supprimer un produit'.

Table 2.6 Cas d'utilisation 'Consulter les produits'.

Table 2.7 Cas d'utilisation 'Créer une facture'.

Table 3.1 dictionnaire de données.

Table 3.2 charte graphique 'Authentification'.

Table 3.3 charte graphique 'Interface Administrateur'.

Table 3.4 charte graphique 'Smartphones'.

Chapitre 01 :
Cadre général

1.1 Introduction

Dans ce premier chapitre nous allons définir quelques généralités sur les ventes, donner une représentation générale de l'application MARSPHONIA (Marseille + Téléphone + IA) ainsi que son fonctionnement.

Donner une définition et une représentation de quelques architectures employées.

1.2 Généralités sur les ventes

1.2.1 La vente [1]

Une vente est une convention par laquelle le vendeur s'oblige à livrer un bien ou un service, et l'acheteur à le payer. En d'autres termes, une vente est l'opération par laquelle un bien ou un droit détenu par un vendeur est cédé à un acheteur contre une somme d'argent (prix de vente).

1.2.2 Suivi de vente

Connaitre en temps réel la quantité et le prix des produits disponibles. On doit surveiller les entrées et sorties de stock.

1.2.3 Les différents pôles de ventes

- ❖ Le vendeur.
- ❖ Le client.
- ❖ Le produit.

1.2.4 L'objectif de la gestion de ventes

MARSPHONIA permet d'organiser la fonction commerciale et de gérer l'ensemble des actions dans la démarche de prospection et de vente.

1.3 Cadre de projet

Cette première partie constitue une présentation générale du cadre de ce projet. En premier lieu, l'organisme d'accueil, qui est l'application web MARSPHONIA, dans laquelle ce travail a été mené et sera présenté.

1.4 Présentation du projet

Notre projet consiste en la réalisation d'une application web de ventes de Smartphones qui soit parfaitement adaptée au fonctionnement spécifique de MARSPHONIA.

Pour cela, nous avons dû mettre en place une base de données en fonction des besoins du site et collecter les données nécessaires pour la réalisation de ce projet.

1.5 Organigramme de l'application

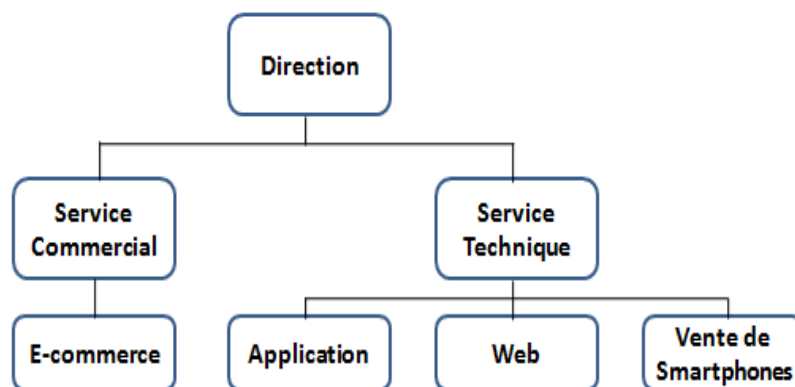


Figure 1.1 Organigramme de l'application

1.6 Processus 2TUP [2]

2TUP (*2 track unified process*, prononcez "toutiyoupi") est un processus de développement logiciel qui implémente le Processus Unifié.

Le **2TUP** propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens). Le processus s'articule ensuite autour de 3 phases essentielles:

- une branche technique
- une branche fonctionnelle
- une phase de réalisation

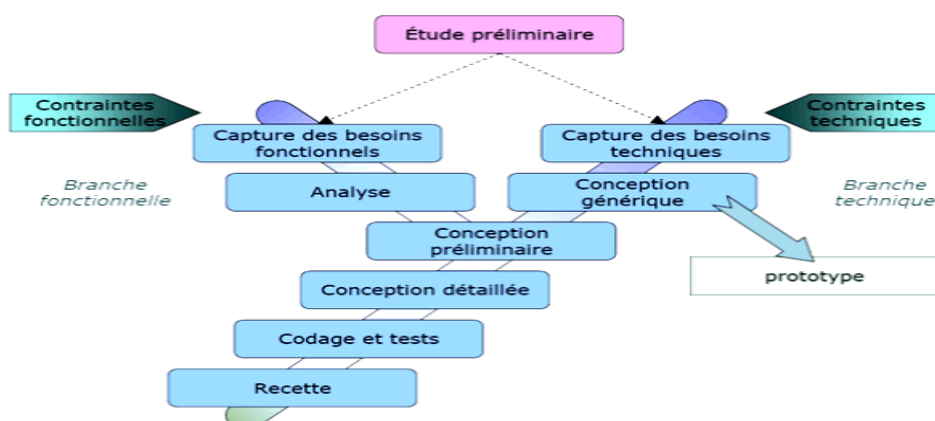


Figure 1.2 schéma du processus 2TUP

1.7 L'architecture MVC dans le développement de notre site [3]

L'architecture **MVC** (*modèle, vue et contrôleur*) est un concept très puissant qui intervient dans la réalisation d'une application. Son principal intérêt est la **séparation des données** (*modèle*), de l'**affichage** (*vue*) et des **actions** (*contrôleur*).

Ce concept peut très bien être appliqué à un site internet. De nombreux sites internet disposent d'une plateforme d'administration pour modifier facilement les contenus. Il est bien évident que l'utilisateur qui va utiliser ce système pour créer des pages ne doit pas modifier la structure complète du site, ni les actions ni l'affichage. Pour qu'il ne modifie que les données, celles-ci doivent alors être isolées.

C'est sur ce constat que se base le système MVC. Chaque partie est bien distincte. Les données peuvent provenir d'une source quelconque et la vue peut être conçue par des webdesigners n'ayant pas de connaissance en programmation.

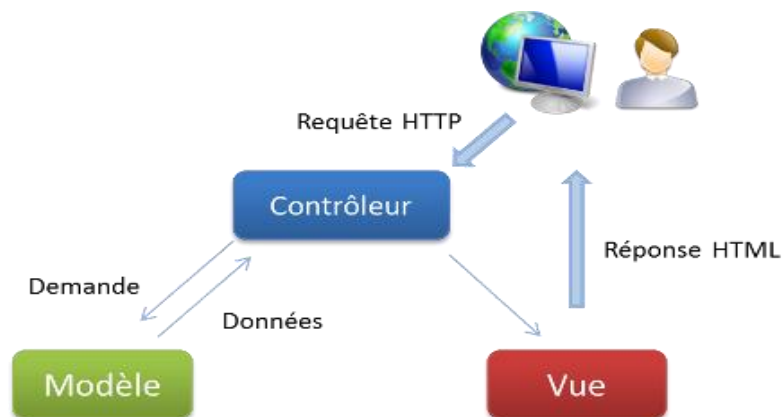


Figure 1.3 schéma de l'architecture MVC

1.8 Conclusion

Ce chapitre présente la structure de l'application web de MARSPHONIA durant lequel nous avons dévoilé le cadre général du travail, la structure ainsi que son organigramme. Le chapitre suivant sera consacré à étude de l'existant et la spécification des besoins de notre projet.

Chapitre 02 :
Spécification des
Besoins

2.1 Introduction

Dans le cadre de ce chapitre, nous allons présenter la spécification de besoins qui consiste à la qualification des besoins fonctionnels et non fonctionnels attendus du système afin de mieux comprendre le projet.

2.2 Spécifications des besoins

Etant donnée la nécessité d'avoir un système informatique de qualité, le sujet de notre application permettra de répondre aux besoins des responsables de l'application MARSPHONIA.

2.2.1 Description des besoins

L'objectif de cette partie est de décrire les besoins et l'exigence de l'utilisateur. Il s'agit de livrer des spécifications pour permettre de faire un choix concernant les désirs de l'utilisateur pour mettre en place un logiciel automatisant les différentes tâches nécessaires pour faire un suivi de ventes et qui répond aux besoins de l'entreprise.

2.2.1.1 Besoins fonctionnels

Cette section présente les services attendus par l'utilisateur de l'interface. Nous décrivons ce que l'interface doit offrir comme fonctions :

- Gestion des utilisateurs : permet à l'administrateur de s'authentifier et aux clients de s'inscrire.
- Gestion des produits : permet à l'administrateur d'ajouter, supprimer, modifier consulter des produits.

2.2.1.2 Besoins non fonctionnels

Ce sont les besoins qui permettraient d'améliorer la qualité des services de l'application comme la convivialité et l'ergonomie des interfaces et l'amélioration du temps de réponse. Parmi ces besoins on cite :

- **La sécurité** : Besoins d'établissement de la connexion, il faut avoir une interface d'authentification qui permet à chaque utilisateur de se connecter pour consulter les offres de l'interface de l'application.
- **La convivialité** : L'application doit être facile à utiliser. Il doit présenter un enchaînement logique entre les interfaces et un ensemble de liens suffisants pour assurer une manipulation rapide et un texte compréhensible, visible et lisible.
- **La disponibilité** : Lorsque n'importe quel utilisateur désire consulter l'application, elle doit être disponible.
- **Temps de réponse** : Le temps de réponse doit être le plus court possible.

2.2.2 Modélisation des besoins

2.2.2.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériels ou autre système) qui interagissent avec le système étudié.

Les acteurs de notre système sont :

Administrateur : c'est lui qui a le droit d'administrer l'application par la gestion des produits et la consultation de l'état du stock ainsi que la mise à jour ses produits.

Client : c'est lui qui a le droit de se connecter pour choisir des Smartphones.

2.2.2.2 Identification des cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'action réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions Acteurs/Système et apporte une valeur ajoutée « notable » à l'acteur concerné.

num	Cas d'utilisation	Acteur
1	▪ Authentification.	Administrateur
2	▪ Gérer les produits (ajouter, modifier et supprimer).	Administrateur
3	▪ Consulter le stock.	Administrateur

Table 2.1 Liste des cas d'utilisation.

2.2.2.3 La description des cas d'utilisation

La description du cas d'utilisation 'Authentification'

Il permet à l'administrateur d'accéder à son espace personnel.

Cas d'utilisation	Authentification
Acteur	Administrateur
Objectif	Il permet à l'administrateur de s'identifier
Pré-condition	Se connecter à L'application
Scénario nominal	<ul style="list-style-type: none"> ▪ Le système affiche le formulaire d'authentification ▪ L'administrateur saisie son login et son mot de passe ▪ Le système vérifie la conformité des informations saisies ▪ Dans le cas où les informations saisies sont fausses le système affiche un message d'erreur puis réaffiche le formulaire d'authentifications et attend que l'utilisateur ressaisie ses informations ▪ Le système donne l'accès à l'interface correspondante
Post-condition	L'administrateur se connecte au système et peut ainsi accéder aux rubriques correspondantes à son profil

Table 2.2 Cas d'utilisation 'Authentification'

La description du cas d'utilisation ' Gestion des produits'

L'administrateur est en charge de gérer les produits en effectuant des mises à jour sur la liste des produits (ajout, suppression, modification). Nous décrivons dans ce qui suit ces cas d'utilisation :

La description du cas d'utilisation ' Ajouter un produit'

Cas d'utilisation	Ajouter un produit
Acteur	Administrateur
Objectif	Permet à l'administrateur d'ajouter un produit à la liste des produits
Pré condition	L'administrateur s'authentifié
Scénario nominal	<ul style="list-style-type: none"> ▪ L'administrateur choisit l'interface d'ajout. ▪ Le menu de l'ajout s'affiche. ▪ L'administrateur saisie les informations relatives à un produit ▪ Le système vérifie les informations saisies ▪ Dans le cas où les informations saisies ne sont pas conformes, un message d'erreur est affiché ▪ Le système enregistre les modifications ▪ Le produit en question apparait définitivement de la base de données.
Post-condition	Mise à jour effectuée

Table 2.3 Cas d'utilisation 'Ajouter un produit'

La description du cas d'utilisation ' Modifier un produit'

Cas d'utilisation	Modifier un produit
Acteur	Administrateur
Objectif	Permet à l'administrateur de modifier les informations relatives à un produit.
Pré condition	L'administrateur s'authentifié et choisit le lien 'Consulter les téléphones'
Scénario nominal	<ul style="list-style-type: none"> ▪ L'administrateur choisit l'interface de modification. ▪ Le menu de la modification s'affiche. ▪ L'administrateur doit sélectionner le produit qu'il désire modifier. ▪ L'administrateur saisie les nouvelles informations ▪ Le système vérifie les informations saisies ▪ Dans le cas où les informations saisies ne sont pas conformes, un message d'erreur est affiché ▪ Le système enregistre les modifications
Post-condition	Mise à jour effectuée

Table 2.4 Cas d'utilisation 'Modifier un produit'

La description du cas d'utilisation 'Supprimer un produit'

Cas d'utilisation	Supprimer un produit
Acteur	Administrateur
Objectif	Permet à l'administrateur de supprimer un produit de la liste des produits
Pré condition	L'administrateur s'authentifié et choisit le lien 'Consulter les

	téléphones'
Scénario nominal	<ul style="list-style-type: none"> ▪ L'administrateur choisit l'interface de suppression ▪ Le menu de suppression s'affiche ▪ L'administrateur doit sélectionner le produit qu'il désire supprimer ▪ Le système averti l'administrateur de l'opération de suppression ▪ L'administrateur confirme la suppression
Post-condition	Mise à jour effectuée

Table 2.5 Cas d'utilisation 'Supprimer un produit'

La description du cas d'utilisation 'Consulter les produit'

Cas d'utilisation	Consulter les produits
Acteur	Administrateur
Objectif	Consulter l'état du stock
Pré condition	L'administrateur s'authentifier
Scénario nominal	<ul style="list-style-type: none"> ▪ L'administrateur demande l'état du stock ▪ Le système affiche l'état du stock
Post-condition	Mise à jour effectuée

Table 2.6 Cas d'utilisation 'Consulter les produits'

Cas d'utilisation	Ajouter un produit
Acteur	Administrateur
Objectif	Permet à l'administrateur de créer une facture
Pré condition	L'administrateur s'authentifier
Scénario nominal	<ul style="list-style-type: none"> ▪ L'administrateur accède à l'interface d'établissement d'une facture ▪ L'administrateur saisi les informations nécessaires ▪ Le système vérifie les informations saisies ▪ Si les informations saisies ne sont pas correctes, le système affiche un message d'erreur ▪ Le système enregistre et affiche une confirmation
Post-condition	Créer une facture

Table 2.7 Cas d'utilisation 'Créer une facture'

2.2.2.4 Présentation des diagrammes des cas d'utilisation

Un diagramme de cas d'utilisation montre les acteurs et leurs relations.

2.2.2.4.1 Le diagramme de cas d'utilisation associé à l'administrateur

Dans ce diagramme nous allons illustrer le cas d'utilisation associé à l'administrateur.

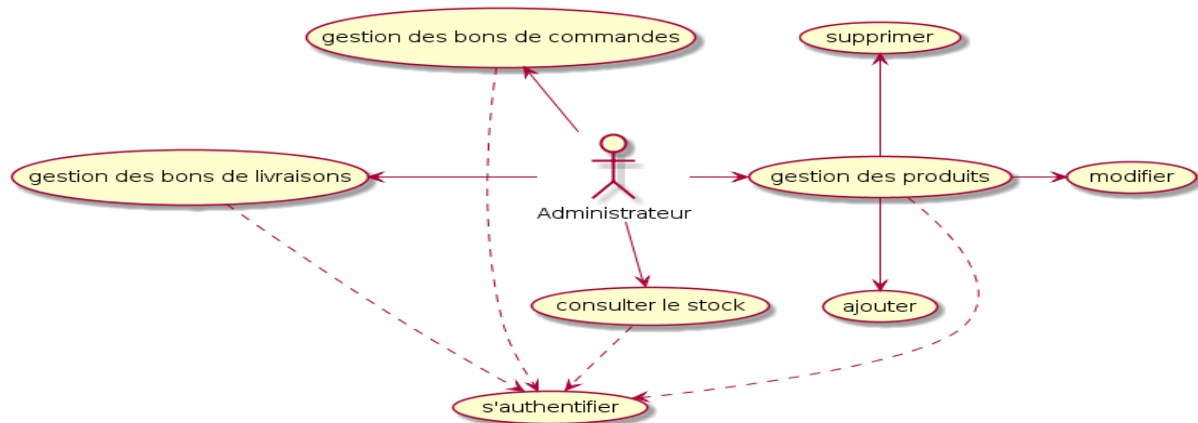


Figure 2.1 : Cas d'utilisation associé à l'administrateur.

2.2.2.4.2 Le diagramme global des cas d'utilisation

Dans ce diagramme on résume tous les cas d'utilisation de tous les acteurs.

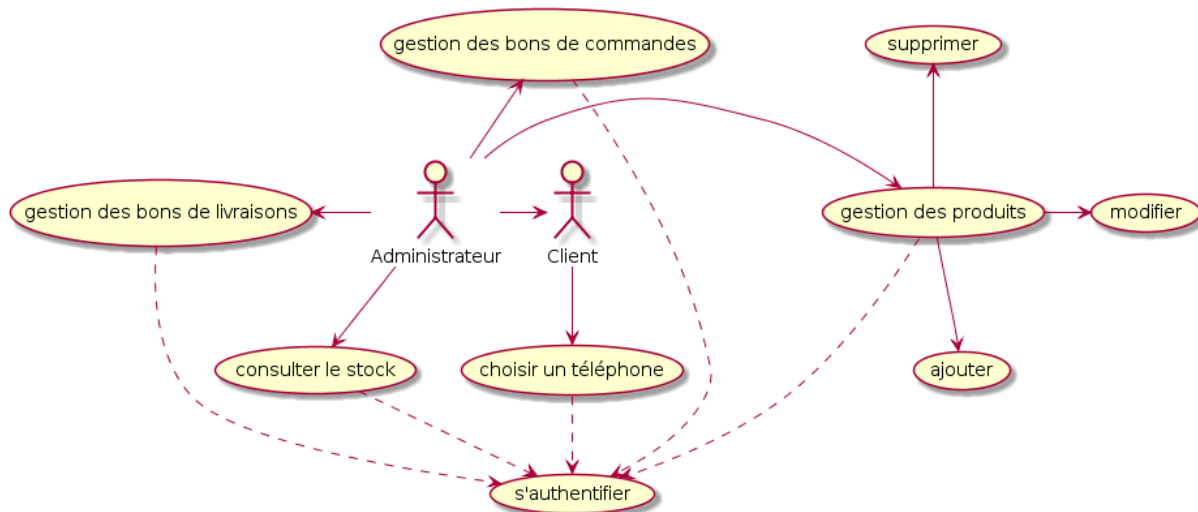


Figure 2.2 Diagramme de cas d'utilisation global

2.3 Conclusion

Dans ce deuxième chapitre, nous avons procédé à l'étude préalable qui nous a permis de comprendre les principes de base d'un système de gestion. Ces principes nous ont permis, par la suite, d'identifier les principaux besoins des utilisateurs et de tracer les grandes lignes de notre système en définissant ses fonctionnalités et les acteurs qui interagissent avec. Puis, grâce au modèle de cas d'utilisation, nous avons essayé de lever les ambiguïtés sur les besoins et les exigences. La conception des cas d'utilisation seront détaillées dans le chapitre suivant.

Chapitre 03 :
Analyse et conception

3.1 Introduction

Ce chapitre se consacre, en premier lieu, à l'analyse des besoins décrits dans le chapitre précédant, en les affinant et en les structurant. L'objectif est d'accéder à une compréhension plus aiguë des besoins et des exigences et d'en livrer une description facile à entretenir, favorisant la structuration de l'ensemble du système, y compris de son architecture.

Il s'agit, donc, d'analyser les cas d'utilisation qui ont été identifiés et raffinés pendant la spécification des besoins. En deuxième lieu, ce chapitre procède à l'enchaînement de conception, ayant pour but de produire les spécifications d'implémentation du système en se basant sur les produits de l'analyse. L'objectif est façonner le système et à lui donner une forme répondant à tous les besoins et exigences.

3.2 Diagrammes de séquence

3.2.1 Présentation générale des concepts des diagrammes de séquences

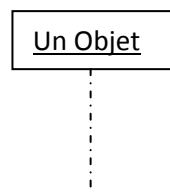
L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges.

Le diagramme de séquence peut être aussi utilisé pour documenter un cas d'utilisation. Les interactions entre objets représentent, dans ce cas, des flux d'informations échangés et non pas de véritables messages entre les opérations des objets.

3.2.1.1 la ligne de vie des objets

La ligne de vie des objets est représentée par une ligne verticale en traits pointillés, placée sous le symbole de l'objet concerné. Cette ligne de vie précise l'existence de l'objet concerné durant un certain laps du temps [4].

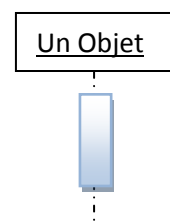
Notation graphique :



3.2.1.2 Barre d'activation




Les diagrammes de séquence permettent de représenter les périodes d'activité des objets. Les périodes d'activité se représentent par des bandes rectangulaires placées sur les lignes de vie. Le début et la fin d'une bande correspondent respectivement au début et à la fin d'une période d'activité [4].

Notation graphique :



3.2.1.3 Types de messages

- **Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. Le message de retour **peut** ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message[4].
- **Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations [4].

Message	Notation graphique
Synchrone	
Asynchrone	
Retour	

3.2.1.4 Fragment d'interaction

Un fragment d'interaction dit combiné correspond à un ensemble d'interaction auquel on applique un opérateur. Plusieurs opérateurs ont été définis dans UML : alt, opt, loop, par, ref etc[4].

- **Opérateur alt**

L'opérateur alt correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon (else).

L'opérateur alt se représente dans un fragment possédant au moins deux parties séparées par des pointillés[4].

- **Opérateur loop**

L'opérateur loop correspond à une instruction de boucle [4].

- **Opérateur opt**

L'opérateur "**opt**" désigne un fragment combiné optionnel comme son nom l'indique : c'est à dire qu'il représente un comportement qui peut se produire... ou pas. Un fragment optionnel est équivalent à un fragment "alt" qui ne posséderait pas d'opérande else (qui n'aurait qu'une seule branche). Un fragment optionnel est donc une sorte de **SI...ALORS**[4].

- **Opérateur ref**

Une référence (interaction occurrence) peut être vue comme un pointeur ou un raccourci vers un autre diagramme de séquence existant. Cela équivaut à copier le contenu du diagramme de séquence pointé en lieu et place de la référence. Attention cependant à être cohérent au niveau des paramètres utilisés. Cela permet de factoriser des parties de comportement utilisées dans plusieurs scénarios [4].

3.2.2 Diagrammes de séquences de l'application à réaliser

3.2.2.1 Diagramme de séquence du cas d'utilisation 'Authentication'

La figure ci-dessous présente le diagramme de séquence du cas d'utilisation "Authentication"

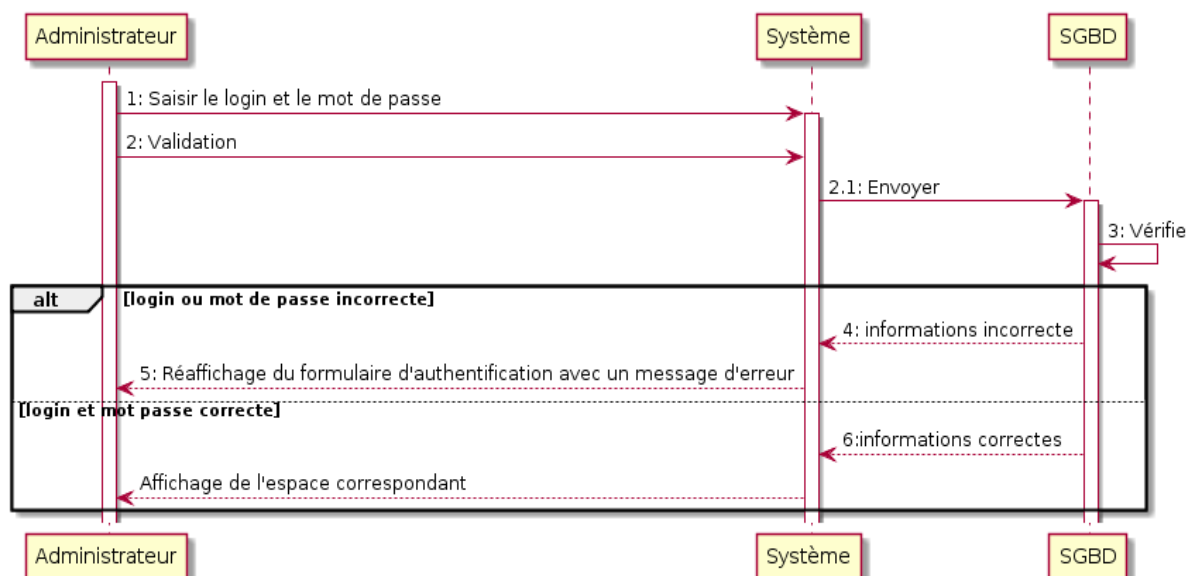


Figure 3.1 : Diagramme de séquence du scénario "Authentication"

3.2.2.2 Diagramme de séquence du cas d'utilisation 'ajouter un produit'

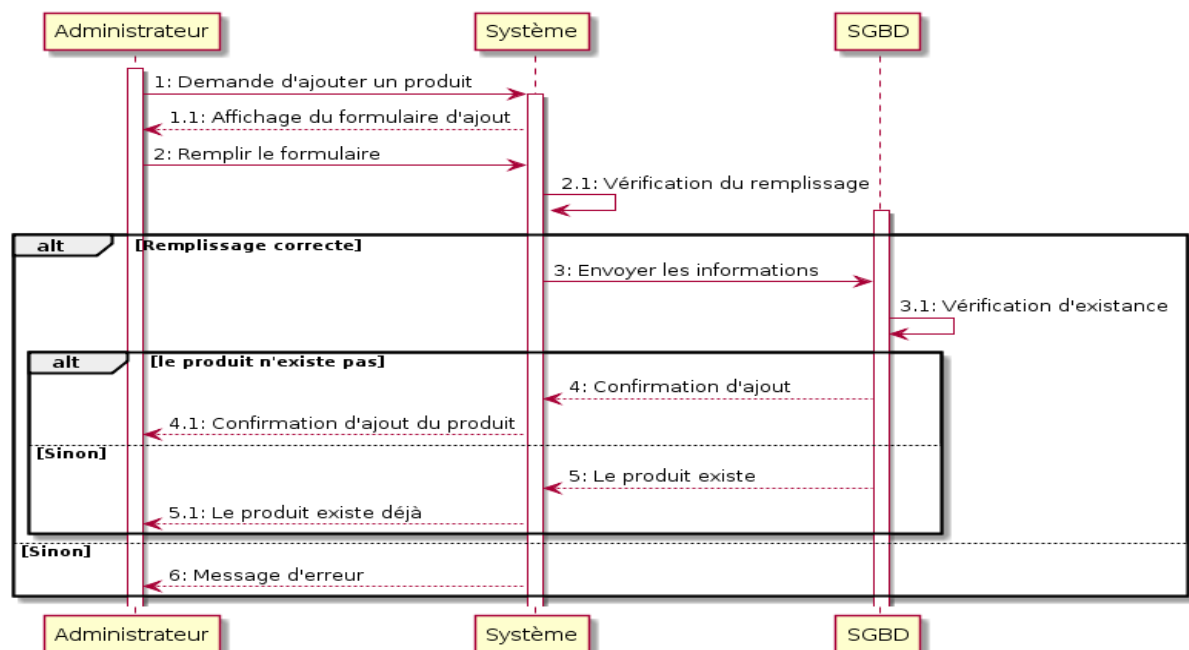


Figure 3.2 : Diagramme de séquence du scénario "Ajouter un produit"

3.2.2.3 Diagramme de séquence du cas d'utilisation 'modifier un produit'

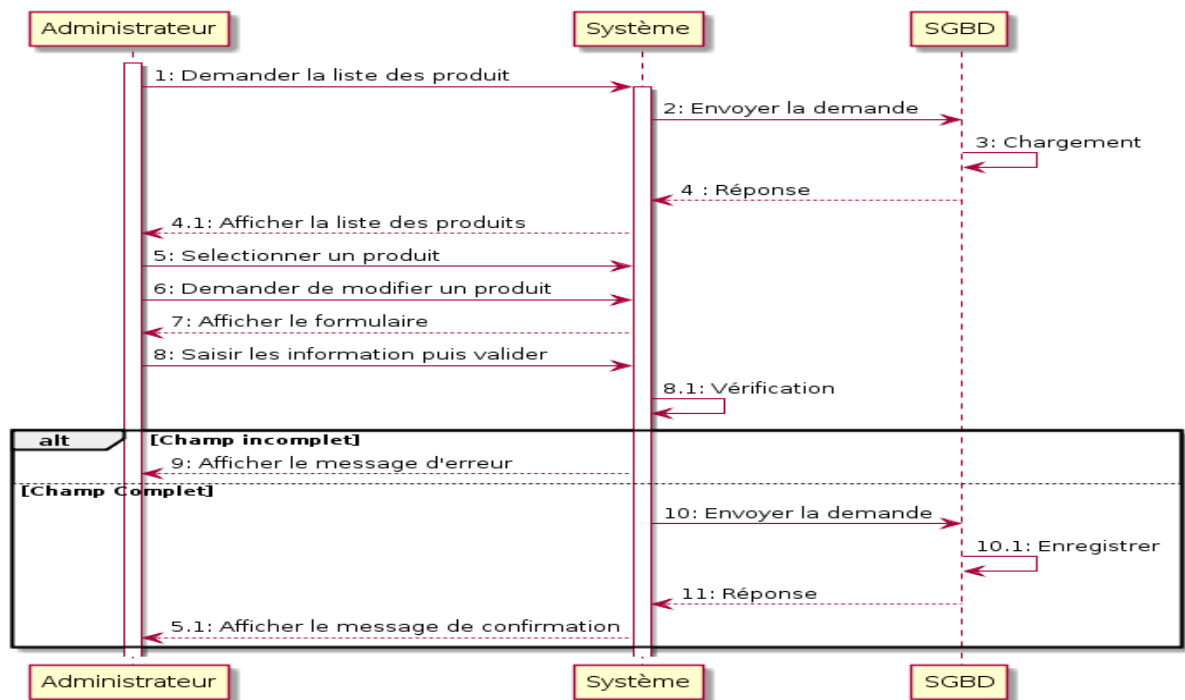


Figure 3.3 : Diagramme de séquence du scénario "Modifier un produit"

3.2.2.4 Diagramme de séquence du cas d'utilisation 'supprimer un produit'

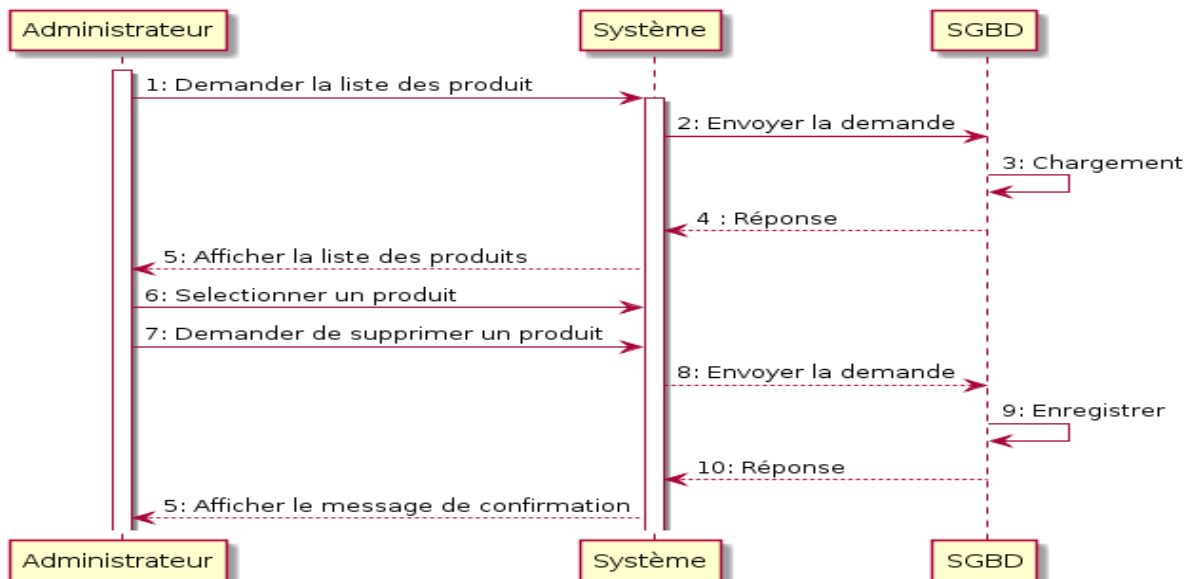


Figure 3.4 : Diagramme de séquence du scénario "Supprimer un produit"

3.4 Diagramme MCD

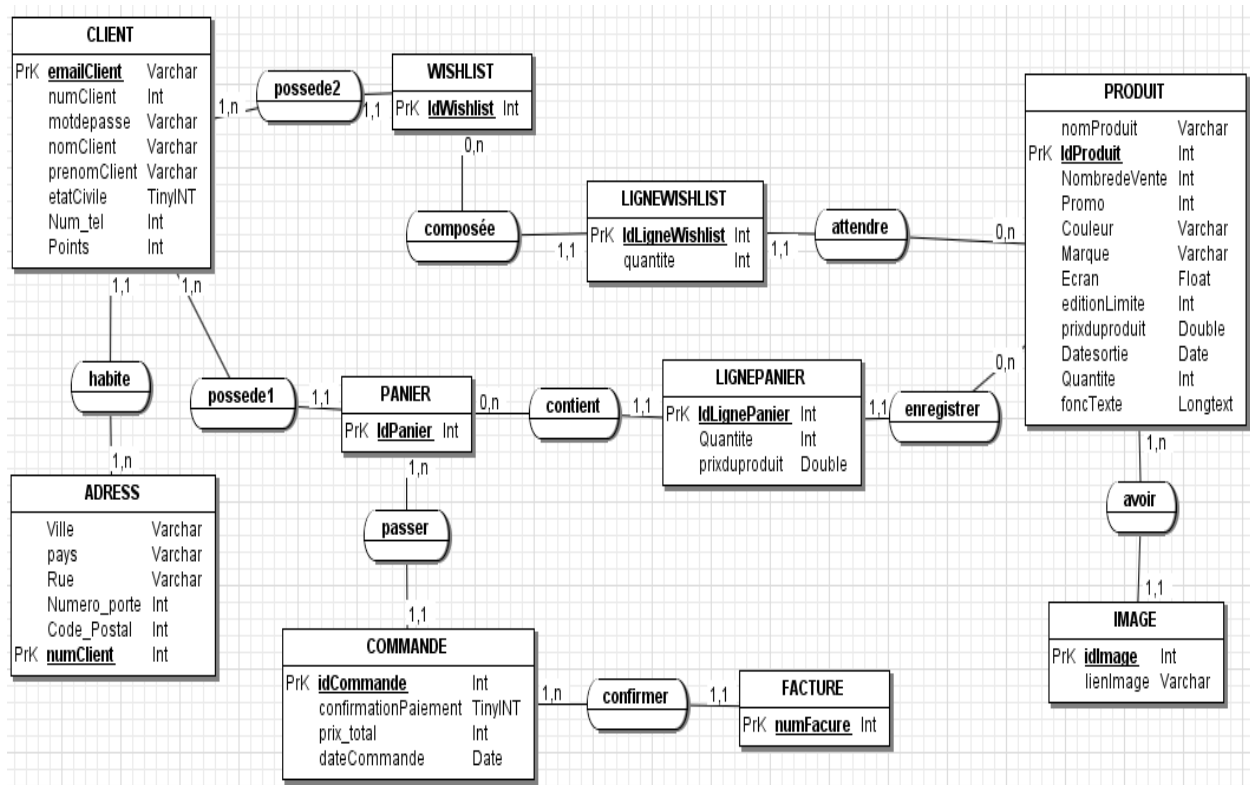


Figure 3.5 : Diagramme MCD

3.5 Dictionnaire de données

Classes	Désignation	Attributs	Type	Taille
Adresse	Ville du client	Ville	Varchar	25
	Pays du client	Pays	Varchar	25
	Rue du client	Rue	Varchar	25
	Numéro de porte du client	Numero_porte	Int	11
	Code postal du client	Code_Postal	Int	11
	Numéro du client	numClient	Int	11
Client	Email du Client	emailClient	Varchar	30
	Numéro du client	numClient	Int	11
	Mot de passe du client	motdepasse	Varchar	25
	Nom de client	nomClient	Varchar	20
	Prénom de client	prenomClient	Varchar	20
	Etat civile du client	etatCivile	Tinyint	1
	Numéro de téléphone du client	Num_tel	Int	11
	Identifiant du panier	Idpanier	Int	11
Commande	Points du client	Points	Int	11
	Identifiant de la commande	idCommande	Int	11
	Confirmation du paiement	confirmationPaieement	Tinyint	1
	Prix total de la commande	prix_total	Int	11
	Date de la commande	dateCommande	Date	-
	Identifiant du panier	IdPanier	Int	11

Facture	Numéro de la facture	numFacure	Int	11
	Identifiant de la commande	idCommande	Int	11
Panier	Identifiant du panier	IdPanier	Int	11
	Numéro du client	numClient	Int	11
LignePanier	Identifiant de LignePanier	IdLignePanier	Int	11
	Identifiant du Produit	IdProduit	Int	11
	Prix du produit	PrixduProduit	Int	11
	Identifiant du panier	IdPanier	Int	11
	Quantité du produit	quantite	int	11
Produit	Nom du produit	nomProduit	Varchar	25
	Numéro du produit	numProduit	Int	11
	Nombre de vente du produit	NombredeVente	Int	11
	Promotion du produit	Promo	Int	-
	Couleur du produit	Couleur	Varchar	25
	Marque du Produit	Marque	Varchar	25
	Ecran du produit	Ecran	Float	-
	Edition limitée du produit	editionLimite	Int	11
	Prix du produit	prixduproduit	Double	-
	Date de sortie du produit	Datesortie	Date	-
	Quantité du produit	quantite	Int	11
	Fonctions du produit	foncTexte	Longtext	-
Image	Identifiant de l'image	idImage	Int	11
	Identifiant du produit	idProduit	Int	11
	Lien de l'image	lienImage	Varchar	25
Wishlist	Numéro de la wishlist	NumWishlist	Int	11
	Numéro du client	numClient	Int	11
LigneWishlist	Identifiant de la LigneWishlist	IdLigneWishlist	Int	11
	Identifiant du produit	IdProduit	Int	11
	Identifiant de la wishliste	IdWishlist	Int	11
	Quantité du produit	quantite	Int	11

Table 3.1 dictionnaire de données

3.6 Passage au modèle relationnel

Nous donnons ci-après quelques règles pour traduire un schéma conceptuel entité-association ou UML en un schéma relationnel équivalent. Il existe d'autres solutions de transformation, mais ces règles sont les plus simples et les plus opérationnelles [5].

➤ **Traitement des classes :**

C'est la traduction des classes en relations, une classe est traduite en une relation.

➤ **Traitement des associations :**

- Chaque association plusieurs à plusieurs est traduite en une relation, dont sa clé est la concaténation des deux clés de ces associations.
- Chaque association un à plusieurs disparaît et la clé primaire du père va maigrir vers les fils.
- Les associations un à un chaque clé primaire devient une clé étrangère dans l'autre classe.

Voici des exemples de multiplicités :

1 : un et un seul.

0,1 : zéro ou un.

0, * : zéro ou plusieurs.

1, * : un ou plusieurs.

➤ **Agrégation de composition :**

- Dans le cas d'une relation binaire (1)-(0,*) la clé primaire de l'association dont la cardinalité max=1 devient une clé étrangère dans la 2ème association.

➤ **Relation d'héritage :**

- Cas1 : la clé primaire de la classe mère devient une clé étrangère dans les classes filles.
- Cas2 : les clés primaires des classes filles deviennent des étrangères dans la classe mère et les filles vont disparaître.
- Cas3 : tous les attributs de la classe mère deviennent des attributs dans les classes filles et la classe mère va disparaître.

3.7 Modèle Relationnel de données

Le schéma relationnel est basé sur une organisation des données sous forme de tables en suivant les règles de passage, évoquées ci-dessus.

Les tables générées sont les suivantes :

CLIENT (emailClient, numClient, motdepasse, nomClient, prenomClient, etatCivile, Num_tel, Points)

ADRESS (numClient, Ville, pays, Rue, Numero_porte, Code_Postal)

COMMANDE (idCommande, confirmationPaiement, prix_total, dateCommande, # IdPanier)

FACTURE (numFacture, # idCommande)

PANIER (IdPanier, # emailClient)

LIGNEPANIER (IdLignePanier, Quantite, # prixduproduit, # IdPanier, # IdProduit)

PRODUIT (IdProduit, nomProduit, NombredeVente, Promo, Couleur, Marque, Ecran, editionLimite, prixduproduit, Datesortie, Quantite, foncTexte)

IMAGE (idImage, lienImage, IdProduit)

WISHLIST (IdWishlist, #emailClient)

LIGNEWISHLIST (IdLigneWishlist, quantite, #IdProduit, #IdWishlist)

3.8 Maquettes

Définition [6]

Une maquette de site web est une ébauche d'un site Internet qui est présentée sous forme statique.

Sur les pages de la maquette ne figurent généralement que l'identification des différentes zones de contenus.

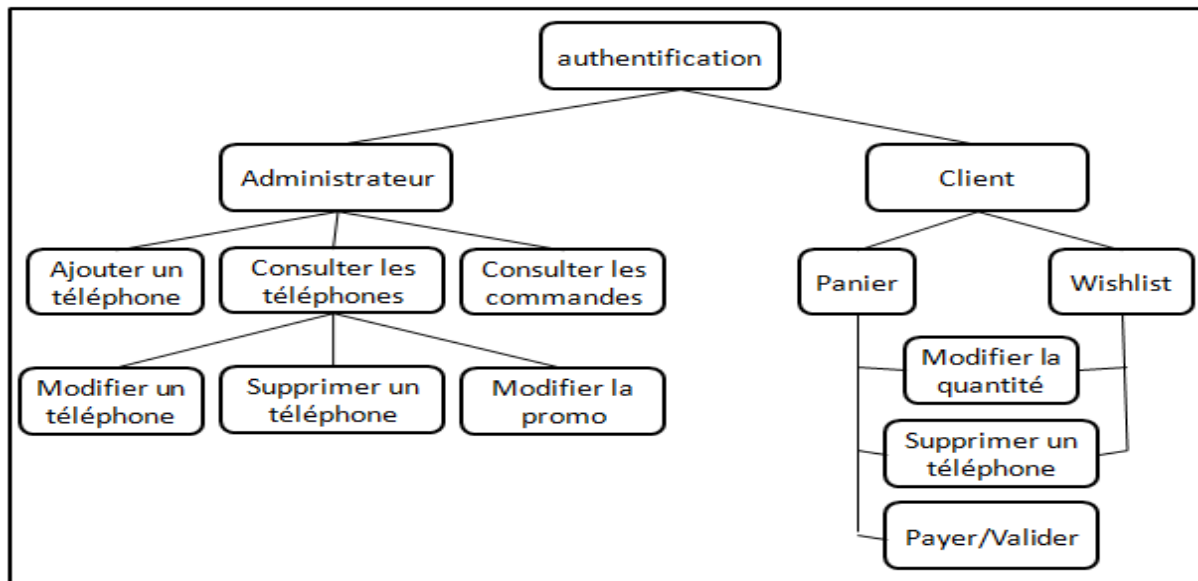
3.8.1 Structure de l'application

Figure 3.6 structure de l'application

3.8.2 La charte graphique**Définition [7]**

La charte graphique est un guide comprenant les recommandations d'utilisation et les caractéristiques des différents éléments graphiques (logos, couleurs, polices, symboles, calques..) qui peuvent être utilisés sur les différents supports de communication de l'entreprise.

La charte graphique permet de garantir l'homogénéité et la cohérence de la communication visuelle au sein de l'entreprise.

La charte graphique de l'interface d'authentification

Logo
Barre de navigation
Zone d'authentification
Pied de la page

Table 3.2 charte graphique 'Authentification'

La charte graphique de l'interface d'administrateur :

Logo
Barre de navigation
Ajouter téléphone Consulter les téléphones Consulter les commandes
Pied de la page

Table 3.3 charte graphique 'Interface Administrateur'

La charte graphique de l'interface Smartphones :

Logo
Barre de navigation
Sélectionner les différent Smartphones
Smartphones correspondant

3.9 Conclusion

Dans le présent chapitre, nous avons traité l’enchaînement d’analyse, qui élabore un modèle objet conceptuel servant à analyser les besoins et les exigences, en les affinant et en les structurant. Nous avons donc abouti à un modèle d’analyse, qui nous a permis de procéder à l’enchaînement de conception par la prise en compte de la majeure partie des exigences non fonctionnelles et autres contraintes liées à l’environnement.

Dans le prochain chapitre, nous allons montrer comment nous avons traduit cette étude conceptuelle et à l’aide de quels outils nous avons pu mettre en place notre système.

Chapitre 04 :

Réalisation

4.1 Introduction

Après avoir affecté l'étude et la conception de notre application, nous passons à la phase d'implémentation. Ce chapitre présente le résultat du travail effectué durant ce projet. Nous allons présenter, aussi, environnement matériels et les outils de développement utilisés. Nous clôturons ce chapitre par quelques captures d'écran démontrant les fonctionnalités de notre application.

4.2 Environnement de développement

4.2.1 Environnement matériel

L'équipement mis à notre disposition pour la réalisation du projet se compose d'un ordinateur portable dont la configuration est la suivante :

- Processeur : Intel(R) Pentium.
- 6 GO de mémoire vive.
- Système d'exploitation 64 bits.

4.2.2 Environnement logiciel

Après avoir présenté les moyens matériels mis à notre disposition dans le cadre de réalisation de ce projet, nous abordons dans cette partie les moyens logiciels utilisés. Les logiciels utilisés pour la réalisation de ce projet ainsi que pour la rédaction du rapport sont :

❖ NetBeans 8.0.2

C'est un environnement de développement intégré (EDI), il permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML.

❖ PlanText (<https://www.planttext.com/>)

Pour la réalisation des différents diagrammes de modélisation.

❖ PHPMYADMIN

C'est une interface d'administration pour le SGBD MySQL. Il est écrit en langage PHP et s'appuie sur le serveur HTTP Apache [12].

4.2.3 Les langages de développement

❖ Html

Le **HTML** est un langage dit de « marquage » (de « structuration » ou de « balisage ») dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents[8].

❖ CSS

Le terme **CSS** est l'acronyme anglais de *Cascading Style Sheets* qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML [9].

❖ SQL

SQL ("Structured Query Language" ou "Langage de requêtes structuré") est un langage informatique destiné à interroger ou piloter une base de données[10].

❖ JAVASCRIPT

Le **JavaScript** est un langage informatique utilisé sur les pages web. Ce langage à la particularité de s'activer sur le poste client, en d'autres mots c'est votre ordinateur qui va recevoir le code et qui devra l'exécuter. C'est en opposition à d'autres langages qui sont activé côté serveur. L'exécution du code est effectuée par votre navigateur internet tel que Firefox ou Internet Explorer [11].

❖ PHP

Le **PHP** est un langage informatique utilisé sur l'internet. Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que MySQL [13].

Ci-dessous, le mode de fonctionnement du PHP :

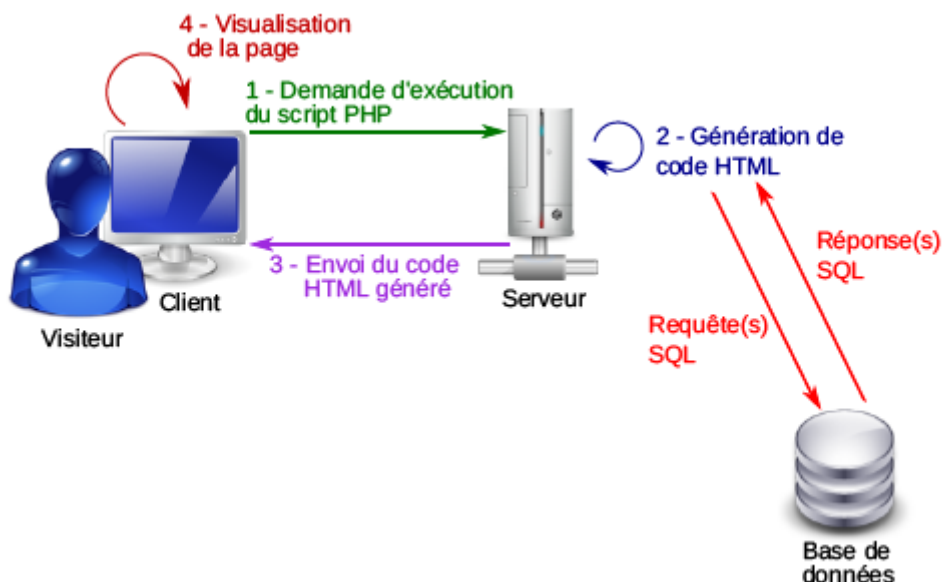



Figure 4.1 : Le mode de fonctionnement du PHP

4.4 Présentation des interfaces principale

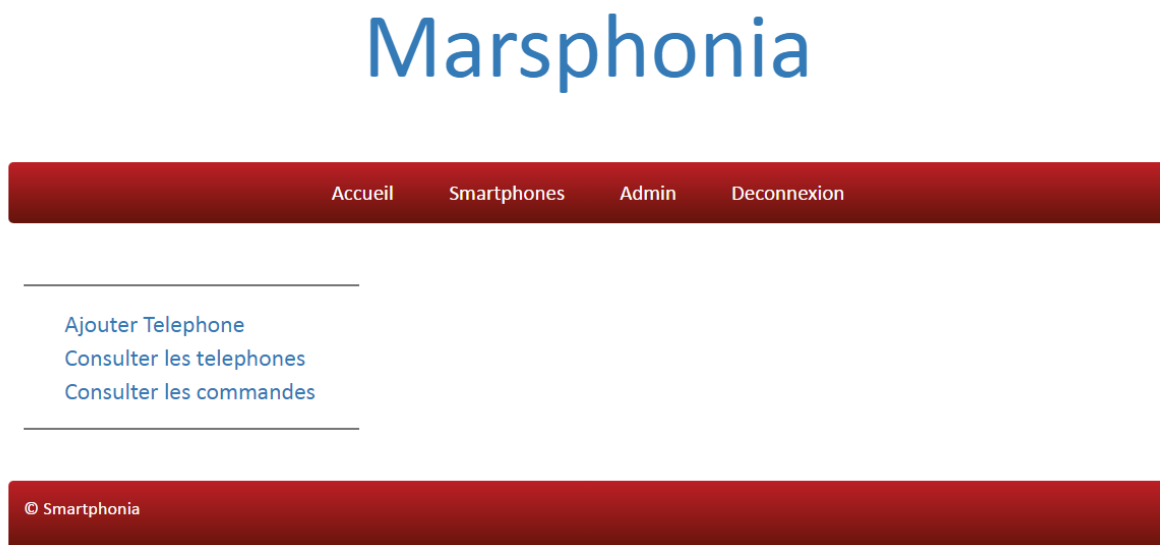
Ci-dessous, l'interface d'authentification :



The screenshot shows the Marsphonia authentication interface. At the top, the word "Marsphonia" is displayed in a large blue font. Below it is a dark red navigation bar with white text links: "Accueil", "Smartphones", "Connexion", "Inscription", and "Contact". The main area contains a login form with two input fields: "Mail :" and "Mot de passe :". To the right of the password field is a "connexion" button. At the bottom, a dark red footer bar contains the text "© Smartphonia".

Figure 4.2 : l'interface d'authentification

Ci-dessous, l'interface de l'espace administrateur :



The screenshot shows the Marsphonia administrator interface. At the top, the word "Marsphonia" is displayed in a large blue font. Below it is a dark red navigation bar with white text links: "Accueil", "Smartphones", "Admin", and "Deconnexion". The main area contains a list of links: "Ajouter Telephone", "Consulter les telephones", and "Consulter les commandes". At the bottom, a dark red footer bar contains the text "© Smartphonia".

Figure 4.3 : l'interface de l'espace administrateur

Ci-dessous, l'interface d'inscription :

Marsphonia

[Accueil](#) [Smartphones](#) [Connexion](#) [Inscription](#) [Contact](#)

Nom :	Prenom :	Mail :	<input type="button" value="Inscription"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Telephone :	Etat civil :	Mot de passe :	
<input type="text"/>	<input checked="" type="radio"/> Homme <input type="radio"/> Femme	<input type="text"/>	

© Smartphonia

Figure 4.4 : l'interface d'inscription

Ci-dessous, l'interface d'ajout d'un produit :

Marsphonia

[Accueil](#) [Smartphones](#) [Admin](#) [Deconnexion](#)

Nom :	Date de sortie :	Fonctions :
<input type="text"/>	<input type="text" value="jj/mm/aaaa"/>	<input type="text"/>
Marque :	Stock :	Ajouter une image :
<input type="text"/>	<input type="text"/>	<input type="button" value="Choisissez un fichier"/> Aucun fichier choisi
Couleur :	Edition limitée :	Prix :
<input type="text"/>	<input type="radio"/> Oui <input checked="" type="radio"/> Non	<input type="text"/>
Ecran :	<input type="text"/>	<input type="button" value="Ajouter le telephone"/> <input type="button" value="Réinitialiser"/>

© Smartphonia

Figure 4.5 : l'interface de l'ajout d'un produit

Ci-dessous, l'interface de choix de Smartphones coté client :

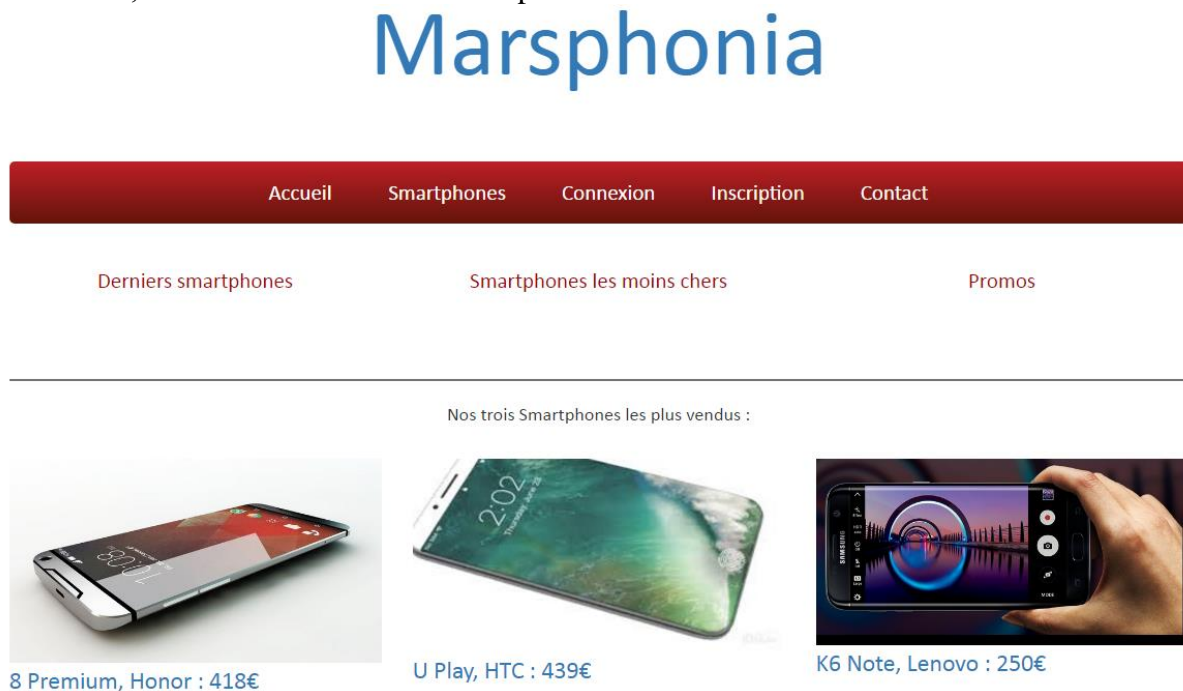


Figure 4.6 : l'interface de choix de Smartphones coté client

Ci-dessous, l'interface de la page de consultation des Smartphones coté admin :



Figure 4.7 : l'interface de la page de consultation des Smartphones coté admin

Conclusion

La dernière partie de ce projet était dédiée à la navigation dans notre application. Cette partie constitue le dernier volet de ce projet ; elle a pour objet de présenter l'environnement logiciel et matériel de réalisation. Elle a été clôturée par l'exposition des imprimés écrans décrivant quelques interfaces de notre application.

Conclusion général & Perspectives

Tout au long de ce rapport, nous avons présenté les différentes étapes de réalisation du site web « MARSPHONIA ». Pour le développement de ce projet la technique UML a été utilisée, ce qui a permis de mener correctement la tâche d'analyse des besoins à l'aide du diagramme de cas d'utilisation et la tâche de conception, ainsi les scénarios sont aussi détaillés afin d'expliquer toute tâches faite.

Ce projet nous a donné l'opportunité de s'initier à la vie professionnelle dans un milieu réel et avoir un début d'expérience significatif, et il nous a appris comment dès le matin on peut prendre le parti de la gaieté, comment réussir de bonnes relations pour assurer un travail de groupe, comment compter sur soi pour résoudre les problèmes au cas où ils se présentent, comment être méticuleux dans notre travail, comment être attentifs aux indications de nos supérieurs, comment être bien organisés pour accomplir dans les meilleurs délais, et meilleures conditions les tâches qui nous sont confiées.

Au cours de la réalisation de notre projet, nous avons étaients astreints par quelques limites notamment, la contrainte du temps qui était relativement un obstacle devant l'ajout de certaines autres fonctionnalités. Cependant, il était une occasion pour mettre en évidence et déployer sur le plan pratique nos connaissances en informatique.

Le projet peut être amélioré, et ce, en lui ajoutant quelques modules ou interfaces pour mieux l'adapter aux besoins de l'utilisateur et pour qu'il soit toujours fiable et au niveau des progrès atteint par les entreprises.

Bibliographie

- [1]: <https://fr.wikipedia.org/wiki/Vente>
- [2]: <http://www.techno-science.net/?onglet=glossaire&definition=670>
- [3]: <http://www.guillaumevoisin.fr/internet/larchitecture-mvc-dans-le-developpement-dun-site-internet>
- [4]: P. Roques. UML 2 modéliser une application web. EYROLLES, 2008.
- [5]: S.Christian. UML 2 pour les bases de données. EYROLLES 2007.
- [6]: <http://www.definitions-marketing.com/definition/maquette-de-site-web/>
- [7]: <http://www.definitions-marketing.com/definition/charte-graphique/>
- [8]: <http://www.commentcamarche.net/contents/498-html-langage>
- [9]: <http://glossaire.infowebmaster.fr/css/>
- [10]: <http://www.futura-sciences.com/tech/definitions/informatique-sql-2524/>
- [11]: <http://glossaire.infowebmaster.fr/javascript/>
- [12]: <https://doc.ubuntu-fr.org/phpmyadmin>
- [13]: <http://glossaire.infowebmaster.fr/php/>