# SOMMAIRE

INTRODUCTION GENERALE 1

CHAPITRE 1 : CADRE GENERAL DE PROJET 2

1.1 Introduction 2

1.2. Organisme d’accueil 2

1.2.1. Présentation 2

1.2.2. Services 3

1.2.3. Organigramme 4

1.3. Présentation de projet 4

1.3.1. Etude de l’existant 4

1.3.2. Critique de l’existant 7

1.3.3. Solution proposée 8

1.4. Méthodologie de projet 8

1.4.1. Méthode "Cycle en V” 8

1.5. Planification de projet 10

1.6. Conclusion 11

CHAPITRE 2 : ANALYSE ET SPECIFICATION DES BESOINS 12

2.1 Introduction 12

2.2. Spécification des besoins 12

2.2.1. Identification des acteurs 12

2.2.2. Besoins fonctionnels 12

2.2.3. Besoins non fonctionnels 13

2.3. Modélisation des besoins fonctionnels 14

2.3.1. Langage de modélisation "UML" 14

2.3.2. Diagrammes des cas d’utilisations 14

2.3.2.1. Diagramme de cas d’utilisation global 14

2.4. Raffinement des cas d'utilisations 15

2.4.1. Raffinement des cas d’utilisation de l’acteur Administrateur 15

2.4.2. Raffinement des cas d’utilisation de l’acteur Visiteur 20

2.4.3. Raffinement des cas d’utilisation de l’acteur Client 24

2.5. Conclusion 28

CHAPITRE 3 : CONCEPTION 29

3.1. Introduction 29

3.2. Diagramme de séquence 29

3.2.1. Définition 29

3.2.2. Diagramme de séquences de l’acteur administrateur 29

3.2.3. Diagramme de séquences de l’acteur visiteur 32

3.2.4. Diagramme de séquences de l’acteur client 34

3.3. Diagramme d’activité 37

3.3.1. Définition 37

3.3.2. Diagramme d’activité d’acteur Administrateur 38

3.3.3. Diagramme d’activité d’acteur Visiteur 41

3.3.4. Diagramme d’activité d’acteur Client 45

3.4. Diagramme de classes 47

3.4.1. Définition 47

3.4.2. Description des classes 48

3.5. Architecture de l’application 49

3.5.1. Modèle SPA 50

3.5.2. Modèle MVC 50

3.6. Conclusion 51

CHAPITRE 4 : REALISATION 53

4.1. Introduction 53

4.2. Environnement de développement 53

4.2.1. Environnement logiciel 53

4.2.1.1. Editeurs de développement 53

4.2.1.2. Langages de programmation 54

4.2.1.3. Système de gestion de base de données 55

4.2.2. Choix technologie 55

4.2.2.1. Framework côté client : Angular 55

4.2.2.2. Serveur de développement : WAMP 55

4.3. Réalisation 56

4.3.1. Réalisation des interfaces Administrateur 56

4.3.1. Réalisation des interfaces Visiteur 59

4.4. Conclusion 67

CONCLUSION GENERALE 68

BIBLIOGRAPHIE I

# LISTE DES TABLEAUX

**Tableau I :** Comparaison des sites existants 7

**Tableau II :** Environnement de développement 53

**Tableau III :** Langages de développement 54

**Tableau IV :** Gestion de base de données 55

# LISTE DES FIGURES

**Figure 1:** logo de PSA 2

**Figure 2 :** Organigramme PSA 4

**Figure 3 :**Page d’accueil 1 Son Vente 5

**Figure 4 :** Page d’accueil 2 Son Vente 5

**Figure 5 :** Page d’accueil 1 Alibaba Music 6

**Figure 6 :** Page d’accueil 2 Alibaba Music 6

**Figure 7 :**Modelé de Cycle en V 9

**Figure 8 :**Diagramme de planification  10

**Figure 9 :** Diagramme de cas d’utilisation global 15

**Figure 10 :** Diagramme de cas d’utilisation « Gestion des commandes » 15

**Figure 11 :** Diagramme de cas d’utilisation « Gestion des produits » 17

**Figure 12 :** Diagramme de cas d’utilisation « Gestion des produits » 19

**Figure 13 :** Diagramme de cas d’utilisation « S’inscrire » 20

**Figure 14:** Diagramme de cas d’utilisation « Consulter un produit » 21

**Figure 15 :** Diagramme de cas d’utilisation « Recherche un produit » 22

**Figure 16 :** Diagramme de cas d’utilisation « Gestion panier » 23

**Figure 17 :** Diagramme de cas d’utilisation « Authentification » 24

**Figure 18 :** Diagramme de cas d’utilisation « Passer une commande » 25

**Figure 19 :** Diagramme de cas d’utilisation « Gérer son compte » 27

**Figure 20 :** Diagramme de séquences « Gestion des commandes » 30

**Figure 21 :** Diagramme de séquence « Gestion des produits » 31

**Figure 22 :**Diagramme de séquence « Supprimer les paniers des visiteurs » 32

**Figure 23 :** Diagramme de séquence « S’inscrire » 32

**Figure 24 :** Diagramme de séquence « Consulter un produit » 33

**Figure 25 :** Diagramme de séquence « Recherche un produit » 33

**Figure 26 :** Diagramme de séquence « Gestion panier » 34

**Figure 27 :** Diagramme de séquence « Authentification » 35

**Figure 28 :** Diagramme de séquence « Passer une commande » 36

**Figure 29 :** Diagramme de séquence « Gérer son compte » 37

**Figure 30 :** Diagramme d’activité « Gestion des commandes » 38

**Figure 31 :** Diagramme d’activité « Gestion des produits » 39

**Figure 32 :** Diagramme d’activité « Supprimer des paniers des visiteurs » 40

**Figure 33 :** Diagramme d’activité « S’inscrire » 41

**Figure 34 :** Diagramme d’activité « Consulter un produit » 42

**Figure 35 :** Diagramme d’activité « Recherche un produit » 43

**Figure 36 :** Diagramme d’activité « Gestion panier » 44

**Figure 37 :** Diagramme d’activité « Authentification » 45

**Figure 38 :** Diagramme d’activité « Passer une commande » 46

**Figure 39 :** Diagramme d’activité « Gérer son compte » 47

**Figure 40 :** Diagramme des classe 48

**Figure 41 :** Architecture Client-Serveur 50

**Figure 42 :** Modèle S-P-A 50

**Figure 43 :** Modèle M-V-C 51

**Figure 44 :** Interface « commandes en attente » 56

**Figure 45 :** Interface « commandes confirmées » 56

**Figure 46 :** Interface « commandes livrées » 57

**Figure 47:** Interface « gestion des produits » 57

**Figure 48 :** Interface « gestion des produits » 58

**Figure 49 :** L’interface « Gestion des produits » (produit ajouté) 58

**Figure 50 :** Interface « Supprimer panier visiteur » 59

**Figure 51 :** Interface « Supprimer les lignes produits visiteurs » 59

**Figure 52 :** Interface « S’inscrire » 60

**Figure 53 :** Interface « liste des catégories » 61

**Figure 54 :** Interface « liste des sous-catégories » 61

**Figure 55 :** Interface « liste des produits » 61

**Figure 56:** Interface « produit consulté » 62

**Figure 57 :** Interface « Rechercher un produit » 63

**Figure 58 :** Interface « Panier actuel d’un visiteur » 63

**Figure 59 :** Interface « Gestion panier visiteur » 64

**Figure 60 :** Interface « Authentification » 64

**Figure 61 :** Interface « Passer une commande » 65

**Figure 62 :** Interface « Remerciement de client » 66

**Figure 63 :** Interface « Gérer son compte » 66

**Figure 64 :** Interface « Compte modifié » 67

# INTRODUCTION GENERALE

***L***

e monde actuellement, un « village planétaire » relié étroitement à Internet, offrant la possibilité aux individus et aux organisations d’être en échange continue et d’évoluer à l’aide des technologies accessibles à tous. De ce fait les entreprises des différents secteurs d’activité sont omniprésentes sur la toile pour garder le contact avec ses clients et acquérir des nouveaux marchés.

Grâce à des portails et des sites web, une entreprise cherche à renforcer sa notoriété et promouvoir ses produits et se rapprocher le plus possible de ses clients et les fidélisés, en utilisant le plus convenable les canaux de communication.

Ces portails constituent un palier fondamental dans la commercialisation et un axe de « marketing global », on parle aujourd’hui de marketing digital ou marketing numérique que les entreprises essaient d’en bénéficier.

En Tunisie comme ailleurs, les entreprises modernes s’orientent vers les sites web e-commerce (commerce électronique) et les sites web vitrines pour commercialiser ses services et développer son image de marque.

PROSAT-AUDIO, une entreprise tunisienne suit l’évolution et se dirige vers ces réformes, conséquemment le projet est de concevoir et développer un site web vitrine pour la société.

Ce travail s’inscrit dans le cadre de notre projet de fin d’étude pour l’obtention d’un diplôme d’un technicien supérieur de l’Ecole Supérieure des Sciences et de la Technologie de Hammam Sousse.

Ce projet a comme but de concevoir et de mettre en place un site web de type vitrine avec deux parties : une partie frontale de présentation et une partie arrière-plan de gestion. Le site sera un site web dynamique, responsive design et assurant une navigation simple.

# CHAPITRE 1 : CADRE GENERAL DE PROJET

## **1.1 Introduction**

Ce chapitre est une introduction à notre projet de fin d’étude détaillée sur certaines notions qui touchent non seulement le cadre général du projet, mais aussi sa réalisation. La première partie de ce chapitre se focalise sur la présentation de l’organisme d’accueil « PROSAT-AUDIO », une deuxième partie se concentre sur l’idée générale et la problématique qui nous a poussées à développer cette application. Quant à la troisième partie elle sera dédiée à étudier la méthodologie de développement pour notre application et finalement la dernière concerne la planification de notre projet.

## 1.2. Organisme d’accueil

### 1.2.1. Présentation

PROSAT-AUDIO est une société commerciale et de service électronique à responsabilité limité « SARL », fondée en 2013 qui a le souci du détail et de la qualité.

L’équipe technique de PROSAT-AUDIO est mise pour la conception, installation et de tout type d’appareillage de sonorisation et télédistribution parabole collective.



Figure : logo de PSA

La société est spécialisée dans la commercialisation et la distribution de plusieurs gammes de produits électroniques qui sont adaptés pour répondre à vos besoins :

* Sonorisation d’ambiance (ITC, Dynacord, Apart audio, Montarbo …)
* Accessoires Electroniques Fiche et Raccordement et accessoires Tv
* Composants Electronique actifs et passifs
* Equipements outillages
* Régulateurs de tension, inverseur de tension, alimentation de coupage, alimentation stabilisée
* Produit fini électronique …

### 1.2.2. Services

PROSAT-AUDIO développe des solutions dans les trois domaines suivants :

1. Télédistribution :

C’est un mode de distribution des programmes de télévision transitant par l’intermédiaire d’un réseau câblé.

1. Sonorisation d’ambiance :

PROSAT-AUDIO réalise des prestations de sonorisation et d’éclairage pour tout type de manifestation (concert, spectacles, événementiel, sportif, location, installation fixe…)

1. Installation camera de sécurité et vidéo surveillance

### 1.2.3. Organigramme

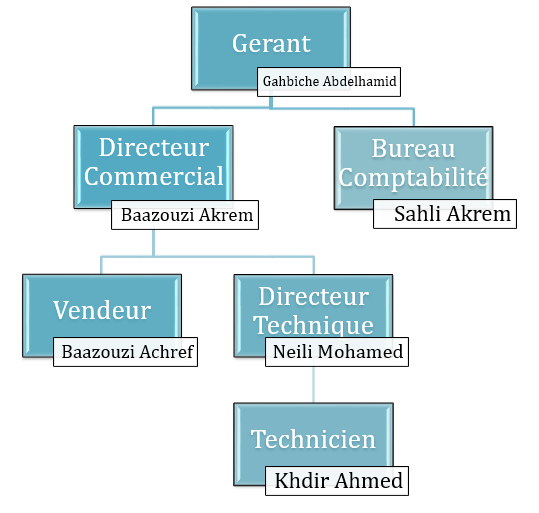
****

Figure  : Organigramme PSA

## 1.3. Présentation de projet

Notre mission durant ce stage est de concevoir et développer un site web au nom de l’entreprise afin de renforcer sa stratégie de marketing et offrir un espace de publicité à sa faveur. Le site assure la mise en valeur des produits et les nouveautés, il s’adresse principalement à ses clients pour garantir la communication numérique avec eux.

### 1.3.1. Etude de l’existant

Comme toutes les sociétés commerciales, PROSAT-AUDIO souhaite faciliter l’opération de vente et améliorer la relation société-client.

Afin d’atteindre ce but, il est indispensable avant de se lancer dans la réalisation de notre plateforme, de bien étudier et analyser des projets similaires pour éviter les malveillances dans le présent projet.

Pour cela on a choisi deux sites très célèbre dans la vente en ligne :

* Application web 1 : SonoVente [1]



Figure  :Page d’accueil 1 Son Vente

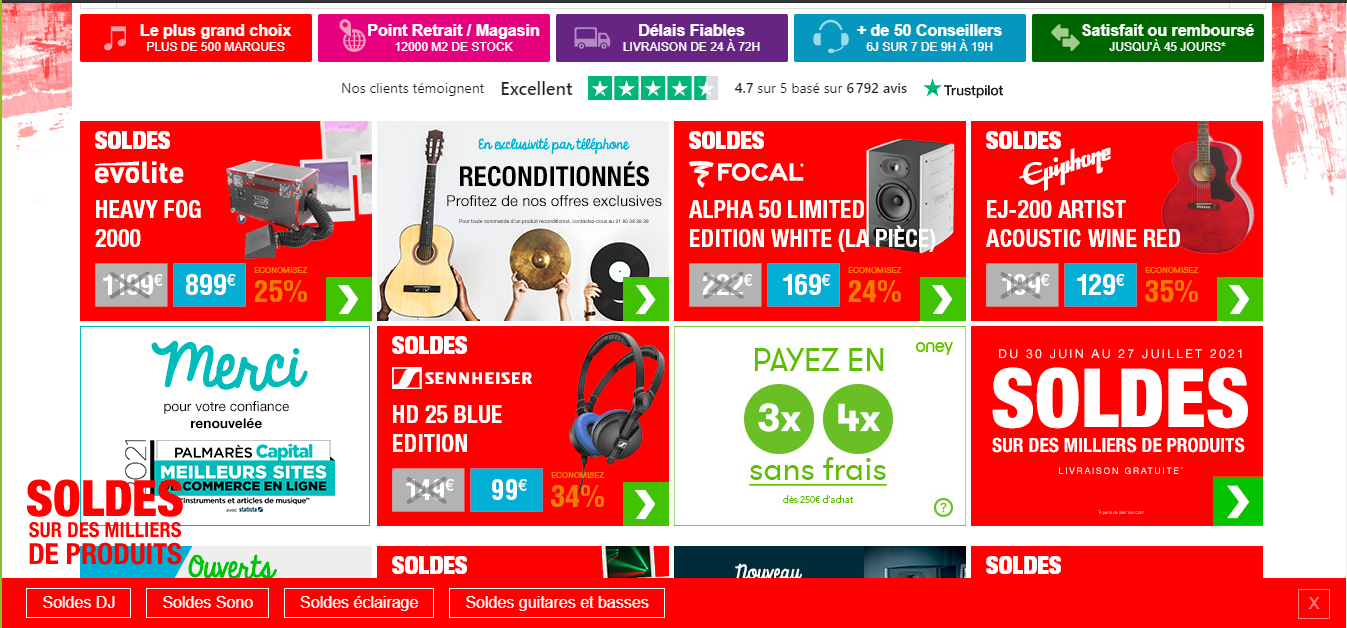


Figure  : Page d’accueil 2 Son Vente

* Fonctionnalités :
* Le visiteur peut consulter un produit et peut l’ajouter au panier.
* Le visiteur peut consulter une liste des produits soldés et les nouveautés.
* Le visiteur peut créer un compte client.
* Le visiteur peut chercher un produit.
* Le client peut passer une commande et la payée en ligne.
* Le visiteur peut consulter les catégories et les sous-catégories.
* Application web 2 : Alibabamusic [2]

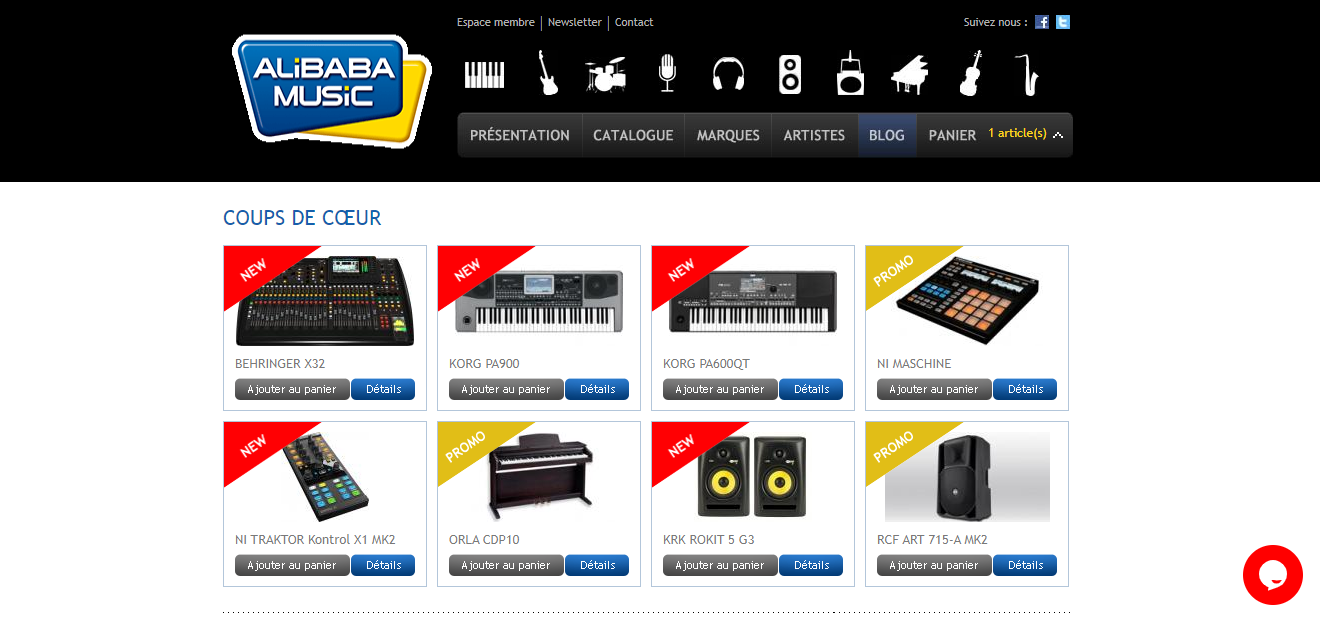


Figure  : Page d’accueil 1 Alibaba Music



Figure  : Page d’accueil 2 Alibaba Music

* Fonctionnalités :
* Le visiteur peut consulter les catégories et les sous-catégories.
* Le visiteur peut consulter une liste des produits soldés et les nouveautés.
* Le visiteur peut consulter un produit.
* Le visiteur peut passer une commande.
* Tableau de comparaison :

Le tableau I suivant illustre une étude comparative des sites précédents :

Tableau  : Comparaison des sites existants

|  |  |  |
| --- | --- | --- |
| Fonctionnalités | Sonovente.com | Alibabamusic.com |
| Commentaire de produit | X | **X** |
| Catégories, sous-catégories | X | X |
| Images de produit | X | **X** |
| Carousel (images slider) | X | **X** |
| Compte client | X | **X** |
| Champ de recherche | X | **X** |
| Panier | X | X |
| Passer une commande | X | X |
| Paiement en ligne | X | **X** |
| Livraison | X | **X** |

### 1.3.2. Critique de l’existant

* Application 1 : SonoVente :
* Fonctionnalités :
* Impossibilité de commenter un produit.
* Charte graphique :
* Beaucoup de couleurs avec des tendances différentes ce qui cause une sorte d’incohérence entre les vues, le client se trouve perdu dans la grande variété des couleurs abandonnant le but principal du site.
* Excès des images et des animations rendre la page plus chargé ce qui cause perdre de la concentration.
* Une très grande partie de la page réservée pour exposer ses produits d’une façon attirable.
* Le panier est situé en haut de la page et centré.
* Application 2 : Alibabamusic :
* Fonctionnalités :
* Impossibilité de recherche d’un produit.
* Impossibilité de création du compte client.
* Impossibilité de commenter un produit.
* Impossibilité de paiement en ligne.
* Difficulté de consultation des sous-catégories.
* Le site est non attirant et non présentable.
* Charte graphique :
* La page d’accueil est tellement courte est les couleurs sont banales, rendre le site non attirant.
* Le produit est présenté par une seule image et sans fiche technique.
* Les catégories se présentent sous forme d’une image en haut de la page d’accueil.
* Les catégories et sous-catégories se présentent encore en bas de la page.

### 1.3.3. Solution proposée

Afin de pallier les lacunes que nous avons trouvé dans la partie précédente, nous proposons une nouvelle solution qui consiste à mettre en place une plateforme e-commerce de vente les produits électroniques.

En effet notre solution va prendre en considération toutes les contraintes citées dans la partie étude de l’existant, elle va exposer l’ensemble de produit de notre société de façon attirante, simple, permettant au client de chercher, consulter et passer ses commandes pour les produits désirés. En outre, notre solution offre le service panier qui permet de mémoriser les produits souhaités par le visiteur, ce dernier peut donner des informations détaillées sur chaque produit ajouté, à savoir la quantité, le prix et le prix totale de la commande.

Par ailleurs, notre solution est gérée par un acteur administrateur, il possède toutes les fonctionnalités de gestion et d’administration de la plateforme. Notre plateforme possède une interface de gestion de produits, de commandes, de panier et de clients.

## 1.4. Méthodologie de projet

Les systèmes informatiques de plus en plus complexe se base sur des différents processus précisant chacun une démarche pour remédier aux problèmes dû à cette complexité.

Ce processus constitue des méthodologies références dans le génie logiciel. Pour bien atteindre les objectifs, une méthodologie de travail doit être choisie selon les besoins du projet.

### 1.4.1. Méthode "Cycle en V”

* Définition :

La méthode Cycle en V qu’on va l’utiliser tout au long de ce rapport afin de concevoir et implémenter notre application web.

Cette méthode est un modèle conceptuel de gestion de projet en deux phases, une pour chaque branche du V. on retrouve ainsi une phase descendante, du besoin exprimé par le client jusqu’à la réalisation du produit, et une phase ascendante, du produit fini à la vérification de sa qualité. Mais les deux branches du V communiquent et interagissent également entre elles. Ainsi, chaque phase de production correspond à une phase de vérification, de validation. Ce jeu de renvoi veille à la qualité du produit fini pour une satisfaction client assurée.

Le Cycle en V se découpe en deux phases (conception et validation), ces dernières sont subdivisées en 9 étapes comme montre la figure\* suivante :



Figure  :Modelé de Cycle en V

* Avantages :

De véritables opportunités s’offrent aux entreprises qui utilisent le Cycle en V afin de façonner leur gestion de projets. Et pour cause, nombreux sont les avantages qu’offre aujourd’hui la méthode :

* Simple à mettre en place (le cycle en V requiert moins de formation collaborateurs qu’une méthode Scrum)
* Amélioration de la productivité : les échanges avec le client en amont du projet et l’élaboration d’un cahier des charges permettent aux parties prenantes d’avoir toutes les informations nécessaires en leur possession.
* Gain de temps : le cycle ne demande pas de réunions quotidiennes, mais seulement des réunions de pilotage actant le passage d’une phase à l’autre et évite ainsi les retours en arrière incessants pour redéfinir les spécifications initiales.
* Gain d’argent : le processus transparent de l’ensemble du cycle de vie du projet permet de définir une enveloppe budgétaire dès l’étape de conception générale. Peu de surprises sont prévoir
* Amélioration de la qualité du produit livré : toute la phase descendante de conception du projet, couplée à la phase ascendante de vérification via des tests, permet d’assurer la qualité du produit final. Cela permet également de limiter les risques et les problèmes rencontrés lors de la réalisation.
* Inconvénients :

Ce modèle classique de gestion de projets n’est cependant pas sans risque. Et pour cause, le Cycle en V présente quelques inconvénients :

* Manque de souplesse
* Manque de communication : chaque membre de l’équipe a un rôle bien défini et s’appuie sur la documentation lorsqu’il rencontre un problème. Résultat, la communication entre les différents membres de l’équipe est quasi inexistante.
* Mise en place fastidieuse : anticiper dès le début l’ensemble des besoins et des spécifications nécessaires en rédigeant différents documents représente un travail assez fastidieux.

## 1.5. Planification de projet

La planification de projet correspond à l’organisation des tâches à réaliser sur une période donnée. Elle permet de visualiser l’avancement des tâches du projet de manière simple et concise, de planifier et suivre les besoins en ressources humaines et matérielles et donc de pouvoir suivre l’avancement du projet.

Le diagramme suivant va représenter les tâches principales à réaliser dans notre plateforme :

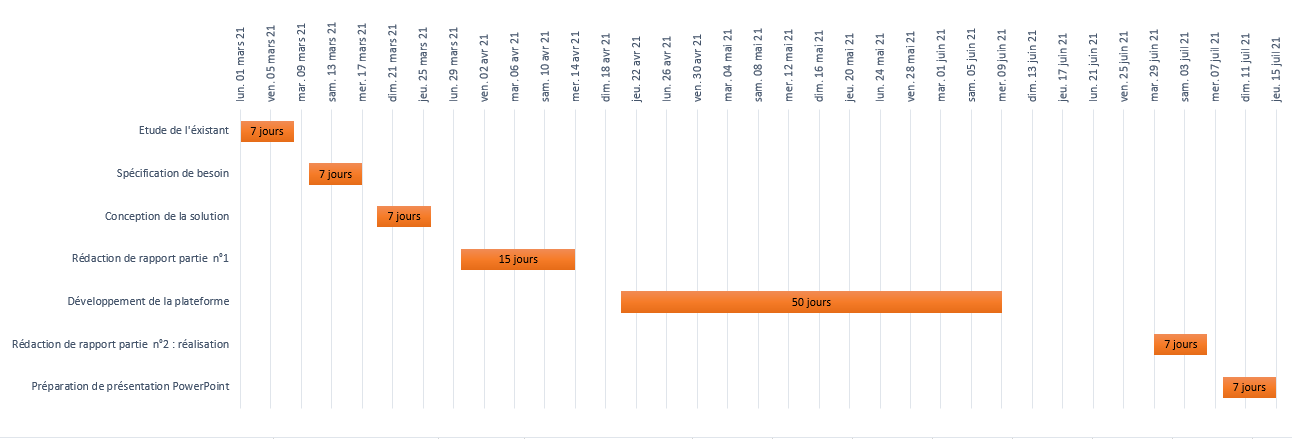


Figure  :Diagramme de planification :

## 1.6. Conclusion

Dans ce premier chapitre, nous avons présenté la société « PROSAT-AUDIO », nous avons essayé de mettre le projet dans son cadre général en présentant l’organisme d’accueil et le sujet à traiter. Pour mieux développer ce projet, nous avons choisi d’intégrer la méthodologie de travail Cycle en V.

Après avoir terminé ce chapitre, nous entamons dans le prochain chapitre l’analyse et la spécification des besoins.

# CHAPITRE 2 : ANALYSE ET SPECIFICATION DES BESOINS

## 2.1 Introduction

Dans le cadre de ce chapitre, on va présenter les principaux besoins de notre application, selon la méthode Cycle en V, cette étape d’analyse et de spécification des besoins est composée de deux parties, la première consiste à analyser les besoins fonctionnels et non fonctionnels de l’application et deuxième pour identifier les acteurs et les cas d’utilisation selon le langage de modélisation UML ainsi les raffinements des cas d’utilisations.

## 2.2. Spécification des besoins

Cette phase consiste à comprendre le contexte du système. Il s’agit de déterminer les besoins fonctionnels et non fonctionnels et d’identifier les acteurs de notre application.

### 2.2.1. Identification des acteurs

On a trois acteurs :

**Administrateur :** gère les produits, les commentaires, les paniers, les clients et les commandes.

**Client :** peut consulter, rechercher, commenter les produits, gérer son profil, son panier et peut passer une ou plusieurs commandes.

**Visiteur :** peut consulter, rechercher les produits et gérer son panier.

### 2.2.2. Besoins fonctionnels

Les besoins fonctionnels sont l’expression de ce que le produit ou le service délivré par le projet devrait être ou faire. Ils sont spécifiés par les représentants des utilisateurs et des bénéficiaires du produit/service.

Pour notre application, nous avons sélectionné les besoins fonctionnels suivants :

* Visiteur :

**Inscription :** Notre plateforme doit permettre aux visiteurs de faire une inscription pour commander les produits.

**Consulter un produit :** Chaque visiteur peut consulter un produit ou bien l’ajouter dans son panier.

**Rechercher un produit :** Chaque visiteur peut rechercher un produit.

**Gestion panier :** Chaque visiteur peut modifier la quantité d’un produit ou bien le supprimer dans son panier.

* Client :

**Authentification :** Chaque client doit s’authentifier et accéder à son espace personnel grâce à un login et un mot de passe.

**Gérer son compte :** Chaque client peut consulter son profil et faire une mise à jour ses informations personnelles.

**Passer une commande :** Chaque client qui possède un ensemble des produits dans son panier, peut passer une commande.

**Gestion panier :** Chaque client peut modifier la quantité d’un produit ou bien le supprimer dans son panier.

**Consulter un produit :** Chaque client peut consulter un produit ou bien l’ajouter dans son panier.

**Rechercher un produit :** Chaque client peut rechercher un produit.

* Administrateur :

**Authentification :** L’admin doit s’authentifier et accéder à son espace personnel grâce à un email et un mot de passe.

**Consulter les clients :** L’admin peut consulter les clients.

**Gestion des commandes :** L’admin peut confirmer, annuler ou livrer les commandes.

**Supprimer les paniers des visiteurs :** L’admin peut supprimer un panier de visiteur.

**Gestion des produits :** L’admin peut ajouter un produit, le modifié ou bien le supprimé.

**Consulter les commentaires :** L’admin peut consulter les commentaires ou les supprimer.

### 2.2.3. Besoins non fonctionnels

La réussite d’une application se base non seulement sur ses fonctionnalités mais aussi sur un certain nombre d’exigences non fonctionnelles.

Les principaux besoins sont les suivants :

**La sécurité :** Lors de chaque connexion, le système demande à l’utilisateur de passer son login et mot de passe. Ce dernier est crypté au niveau de base de données pour garantir plus de sécurité.

**L’ergonomie :** Les interfaces doivent respecter des règles d’utilisabilité donc elles doivent être simples, compréhensibles, faciles à utiliser. Elles doivent également respecter la charte graphique et la palette des couleurs utilisées, l’utilisateur est guidé lors de la saisie de certaines informations afin de respecter les formats des champs de la base de données.

**La performance :** L’application doit être performante à travers ses fonctionnalités et elle doit répondre d’une manière idéale et non ambiguë à toutes les exigences de l’utilisateur.

Contrôle de saisie : L’application prendra également en considération le contrôle d’insertion de données et ne permettra pas l’insertion de données malveillantes pour éviter tout risque de perte de colis.

**Maintenabilité :** Le code doit permettre des futures évolutions ou améliorations.

**Responsive :** L’application doit être fonctionnelle sur n’importe quel appareil ou résolution, en respectant la similarité.

## 2.3. Modélisation des besoins fonctionnels

Nous détaillons dans cette section d’identifier les cas d’utilisations selon le langage de modélisation UML ainsi les raffinements des cas.

### 2.3.1. Langage de modélisation "UML"

Le Langage de Modélisation Unifié UML « Unified Modeling Language », est constitué de diagrammes intégrés utilisés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système. Le langage de modélisation peur servir de modèle pour un projet et garantir une architecture d’information structurée ; il peut également aider les développeurs à présenter leur description d’un système d’une manière compréhensible pour les spécialistes externes.

### 2.3.2. Diagrammes des cas d’utilisations

En langage UML, les diagrammes de cas d’utilisation modélisent le comportement d’un système et permettent de capturer les exigences du système.

Les diagrammes de cas d’utilisation décrivent les fonctions générales et la portée d’un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d’utilisation et les acteurs dans les diagrammes de cas d’utilisation décrivent ce que le système fait et comment les acteurs l’utilisent, mais ne montrent pas comment le système fonctionne en interne.

#### 2.3.2.1. Diagramme de cas d’utilisation global

Le diagramme de cas d’utilisation donne une idée sur ce qu’on peut faire sur notre système (c.à.d. les fonctionnalités principales) en indiquant les acteurs.

Dans ce cadre nous mis en œuvre le diagramme de cas d’utilisation global de notre système.

Ce diagramme est illustré par la figure suivante :

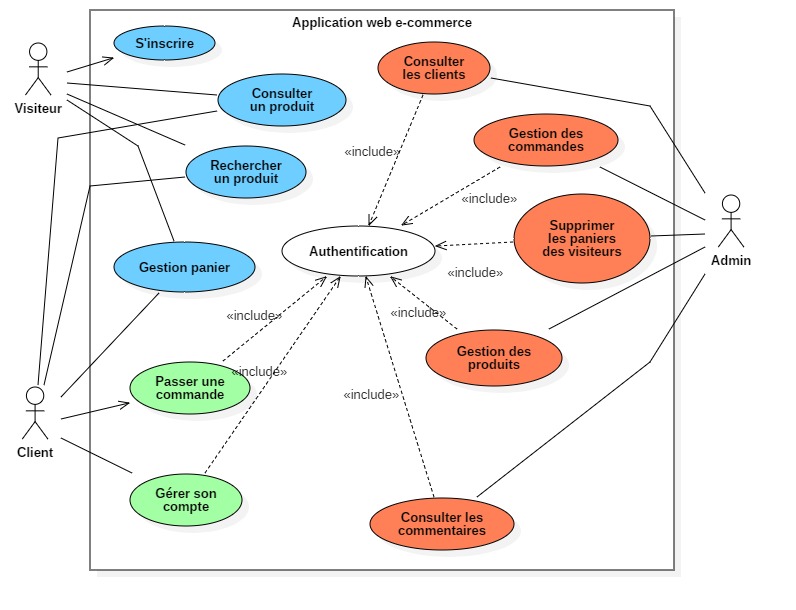


Figure  : Diagramme de cas d’utilisation global

## 2.4. Raffinement des cas d'utilisations

### 2.4.1. Raffinement des cas d’utilisation de l’acteur Administrateur

* Diagramme de cas d’utilisation « Gestion des commandes »

****

Figure  : Diagramme de cas d’utilisation « Gestion des commandes »

**Cas d’utilisation « Gestion des commandes » (confirmer une commande)**

|  |  |
| --- | --- |
| **Nom de Cas** | Gestion des commandes: confirmer une commande |
| **Acteur** | Administrateur |
| **Précondition** | Authentification, au moins une seule commande en attente |
| **Post condition** | Commande confirmée |
| **Scenario nominal** | 1. L’admin fait son choix sur la commande liste des commandes en attente. 2. La plateforme va donner une liste de tous les commandes en attente, puis pour chaque ligne propose la confirmation ou l’annulation de la commande. [E1] 3. L’acteur fait son choix sur l’action confirmer pour une seule ligne. [E2] 4. La plateforme va enregistrer la commande choisie dans la liste des commandes confirmées. |
| **Scenario d’erreur** | [E1] : erreur si le bouton confirmer est désactivé.  [E2] : erreur si l’action fait supprimer la commande. |

**Cas d’utilisation « Gestion des commandes » (livrer une commande)**

|  |  |
| --- | --- |
| **Nom de Cas** | Gestion des commandes: livrer une commande |
| **Acteur** | Administrateur |
| **Précondition** | Authentification, au moins une seule commande confirmée |
| **Post condition** | Commande est livrée |
| **Scenario nominal** | 1. L’admin fait son choix sur la commande liste commandes confirmées. 2. La plateforme va donner une liste de tous les commandes confirmées et propose pour chaque ligne une seule action livrer. [E1] 3. L’acteur fait son choix sur une ligne pour livrer une commande. 4. La plateforme fait enregistrer le choix d’admin dans la liste des commandes livrées. [E2] |
| **Scenario d’erreur** | [E1] : erreur si le bouton livrer n’existe pas.  [E2] : erreur si la commande est dupliqué deux fois dans la liste des commandes livrées. |

* Diagramme de cas d’utilisation « Gestion des produits »

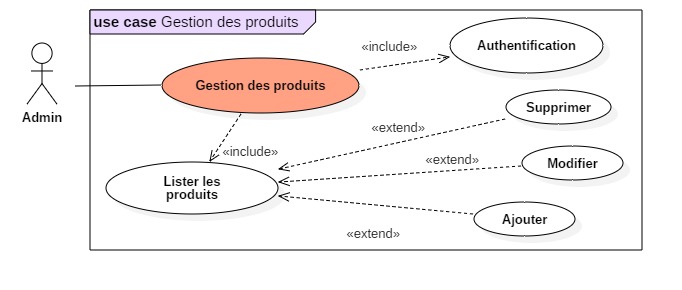


Figure  : Diagramme de cas d’utilisation « Gestion des produits »

**Cas d’utilisation « Gestion des produits » (ajouter un produit)**

|  |  |
| --- | --- |
| **Nom de Cas** | Gestion des produits: ajouter un produit |
| **Acteur** | Administrateur |
| **Précondition** | Authentification |
| **Post condition** | Un nouveau produit est ajouté |
| **Scenario nominal** | 1. L’admin fait son choix sur l’action produits 2. La plateforme va donner une liste de tous les produits et propose un bouton Ajouter en haut de la liste. 3. L’acteur fait son choix sur Ajouter. 4. La plateforme va donner un formulaire. 5. L’acteur remplit le formulaire par les informations du nouveau produit. [E1] 6. La plateforme fait enregistrer le nouveau produit dans la base de données. [E2] 7. La plateforme va donner une liste actualisée des produits. |
| **Scenario d’erreur** | [E1] : erreur si le champ de la quantité avec une valeur zéro.  [E2] : erreur s’il y a un problème avec la base de données. |

* Diagramme de cas d’utilisation « Supprimer les paniers des visiteurs »

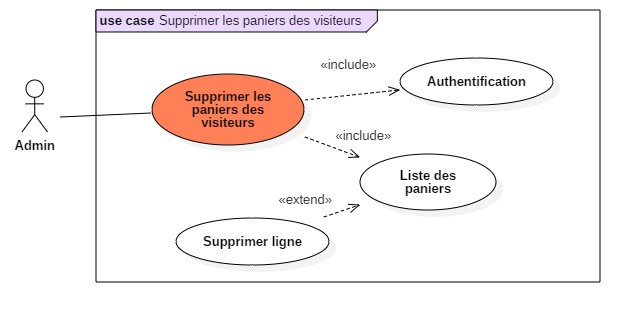


Figure  : Diagramme de cas d’utilisation « Gestion des produits »

**Cas d’utilisation « Supprimer les paniers des visiteurs »**

|  |  |
| --- | --- |
| **Nom de Cas** | Supprimer les paniers des visiteurs |
| **Acteur** | Administrateur |
| **Précondition** | Authentification, au moins une seule ligne produit d’un visiteur |
| **Post condition** | Lignes produit sont supprimées |
| **Scenario nominal** | 1. L’acteur fait son choix sur la commande Supprimer panier. 2. La plateforme va afficher une liste des lignes produit des visiteurs et des clients et propose une action Supprimer. [E1] 3. L’acteur fait son choix sur l’action Supprimer qui appartient à une ligne produit d’un acteur visiteur. 4. La plateforme va supprimer la ligne choisie par l’admin. [E2] |
| **Scenario d’erreur** | [E1] : erreur s’il n’y a pas de ligne produit de visiteur.  [E2] : erreur si la ligne produit n’est pas supprimée. |

### 2.4.2. Raffinement des cas d’utilisation de l’acteur Visiteur

* Diagramme de cas d’utilisation « S’inscrire »

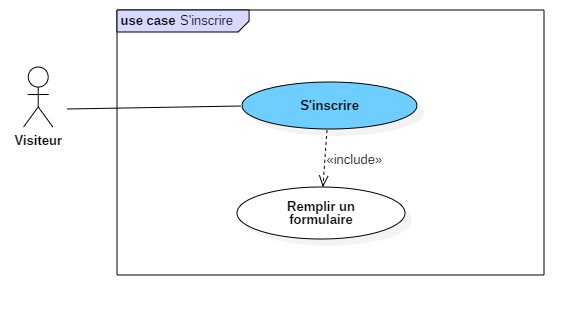


Figure  : Diagramme de cas d’utilisation « S’inscrire »

Le raffinement du cas d’utilisation « S’inscrire » est donnée par le tableau suivant :

**Cas d’utilisation « S’inscrire »**

|  |  |
| --- | --- |
| **Nom de Cas** | S’inscrire |
| **Acteur** | Visiteur |
| **Précondition** | Lien icône utilisateur |
| **Post condition** | Compte d'accès client au site |
| **Scénario nominal** | 1. Le visiteur fait son choix sur l’icône utilisateur. 2. La plateforme redirige le visiteur vers une page d’authentification et propose un lien Pas de compte ? Créez-en un. 3. L’acteur fait son choix sur ce lien. 4. La plateforme va donner un formulaire. 5. L’acteur remplit le formulaire par ses informations personnelles. [E1] 6. La plateforme va créer un nouveau compte client. [E2] 7. La plateforme va rediriger le visiteur vers une page d’authentification. |
| **Scénario d’erreur** | [E1] : erreur si le format du champ email est incorrecte.  [E2] : erreur s’il y a un problème de connexion avec la base de données. |

* Diagramme de cas d’utilisation « Consulter un produit »

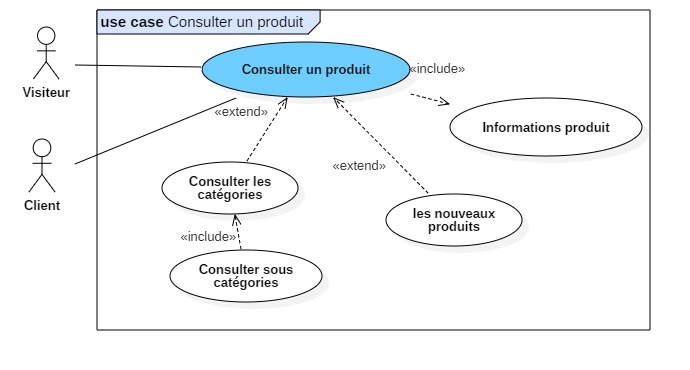


Figure : Diagramme de cas d’utilisation « Consulter un produit »

Le raffinement du cas d’utilisation « Consulter un produit » est donnée par le tableau suivant :

**Cas d’utilisation « Consulter un produit »**

|  |  |
| --- | --- |
| **Nom de Cas** | Consulter produit |
| **Acteur** | Client, visiteur |
| **Précondition** | Page d’accueil |
| **Post condition** | Produits sont consultés |
| **Scénario nominal** | 1. L’acteur fait accès au page d’accueil. 2. L’acteur fait son choix sur Catégorie. 3. La plateforme va donner une liste de sous-catégories. [E1] 4. L’acteur va choisir une sous-catégorie. 5. La plateforme va donner une liste des produits. 6. L’acteur fait son choix sur un produit. 7. La plateforme va afficher le produit en détails. [E2] 8. Le produit est consulté. |
| **Scénario d’erreur** | [E1] : erreur si une sous-catégorie se répète deux fois.  [E2] : erreur s’il y a un manque d’information sur le produit concerné. |

* Diagramme de cas d’utilisation « Recherche un produit »

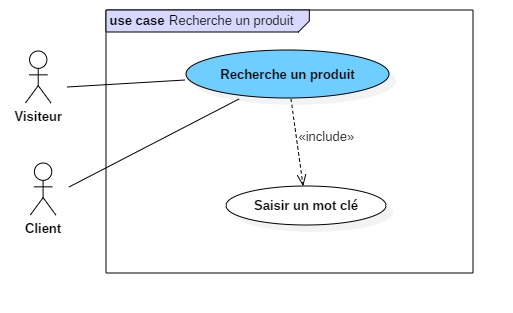


Figure  : Diagramme de cas d’utilisation « Recherche un produit »

Le raffinement du cas d’utilisation « Recherche un produit » est donnée par le tableau suivant :

**Cas d’utilisation « Recherche un produit »**

|  |  |
| --- | --- |
| **Nom de Cas** | Recherche produit |
| **Acteur** | Client, visiteur |
| **Précondition** | Page d’accueil, champ de recherche |
| **Post condition** | Résultat de recherche |
| **Scénario nominal** | 1. L’acteur va taper un mot clé sur le champ de recherche. [E1] 2. La plateforme va suggérer de résultat de recherche selon les premières lettres tapées par le visiteur. [E2] 3. L’acteur fait consulter le résultat de recherche. |
| **Scénario d’erreur** | [E1] : erreur si le mot tapé est de type numérique.  [E2] : erreur si la première lettre tapée sa correspond aux plusieurs produits. |

* Diagramme de cas d’utilisation « Gestion panier »

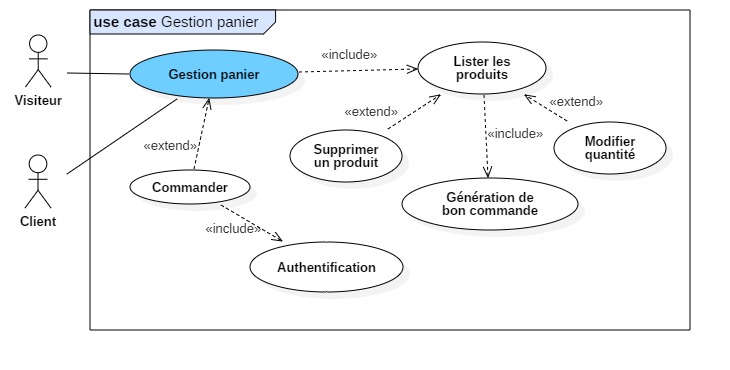


Figure  : Diagramme de cas d’utilisation « Gestion panier »

Le raffinement du cas d’utilisation « Gestion panier » est donnée par le tableau suivant :

**Cas d’utilisation « Gestion panier »** **(Modifier quantité produit)**

|  |  |
| --- | --- |
| **Nom de Cas** | Gestion de panier : Modifier quantité produit |
| **Acteur** | Client, visiteur |
| **Précondition** | Gestion panier |
| **Post condition** | Quantité souhaité est modifiée |
| **Scénario nominal** | 1. L’acteur fait son choix sur la commande panier 2. La plateforme va donner une liste de produit sélectionné, devant chaque ligne d’enregistrement il propose une option modifier quantité. [E1] 3. L’acteur fait une mise à jour de valeur quantité. [E2] 4. La plateforme va l’enregistrer dans la base de données. |
| **Scénario d’erreur** | [E1] : erreur si la liste des produits est vide.  [E2] : erreur si la valeur quantité est nulle. |

### 2.4.3. Raffinement des cas d’utilisation de l’acteur Client

* Diagramme de cas d’utilisation « Authentification »

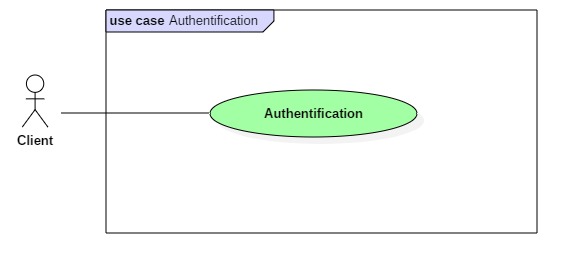


Figure  : Diagramme de cas d’utilisation « Authentification »

* Le raffinement du cas d’utilisation « Authentification » est donnée par le tableau suivant :

**Cas d’utilisation « Authentification »**

|  |  |
| --- | --- |
| **Nom de Cas** | Authentification |
| **Acteur** | Client, Administrateur |
| **Précondition** | Lien icône mon compte |
| **Post condition** | Client authentifié |
| **Scénario nominal** | 1. Le client fait son choix sur l’icône connexion. 2. La plateforme va donner un formulaire d’authentification. 3. L’acteur remplit le formulaire par son email et mot de passe. [E1] 4. La plateforme va vérifier les données client. [E2] 5. La plateforme va rediriger l’acteur vers la page d’accueil. |
| **Scénario d’erreur** | [E1] : erreur si le format du champ email est incorrecte.  [E2] : erreur s’il y a un problème de connexion avec la base de données. |

* Diagramme de cas d’utilisation « Passer une commande »

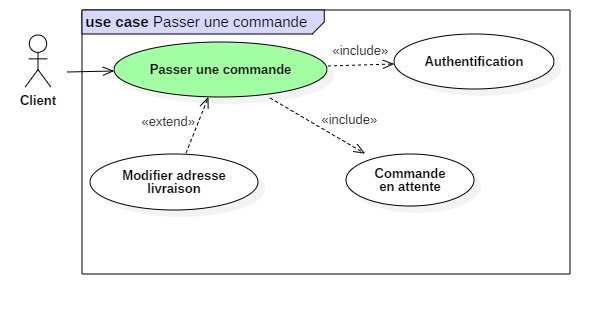


Figure  : Diagramme de cas d’utilisation « Passer une commande »

Le raffinement du cas d’utilisation « Passer une commande » est donnée par le tableau suivant :

**Cas d’utilisation « Passer une commande »**

|  |  |
| --- | --- |
| **Nom de Cas** | Passer commande |
| **Acteur** | Client |
| **Précondition** | Panier, au moins un seul produit, authentification |
| **Post condition** | Commande en attente de confirmation |
| **Scénario nominal** | 1. Le client fait son choix sur la commande panier. 2. La plateforme va afficher une liste de tous les produits ajoutés au panier. 3. L’acteur fait son choix sur la commande passer la commande. 4. La plateforme va rediriger l’acteur vers la page d’authentification. [E3] 5. L’acteur remplit le formulaire. 6. La plateforme va reconnaitre l’acteur par une redirection vers la page de changement d’adresse de livraison. 7. La plateforme va afficher un formulaire de changement d’adresse. [E1] 8. L’acteur fait son choix sur le bouton valider. [E2] 9. La plateforme va ajouter la commande dans une liste d’attente. 10. La plateforme va afficher un message de validation et redirige le client vers une page de remercîment. |
| **Scénario d’erreur** | [E1] : erreur si le formulaire ne contient pas l’ancienne adresse.  [E2] : erreur si le bouton valider va rediriger le client vers un autre formulaire changement adresse.  [E3] : si l’acteur ne possède pas un compte, il doit s’inscrire. |

* Diagramme de cas d’utilisation « Gérer son compte »

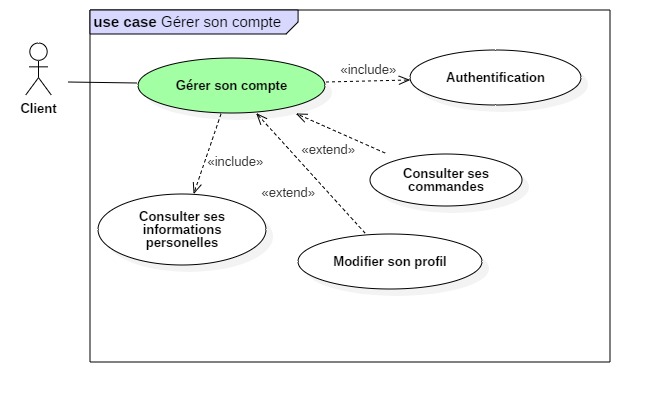


Figure  : Diagramme de cas d’utilisation « Gérer son compte »

Le raffinement du cas d’utilisation « Gérer son compte » est donnée par le tableau suivant :

**Cas d’utilisation « Gérer son compte »**

|  |  |
| --- | --- |
| **Nom de Cas** | Gérer son compte |
| **Acteur** | Client |
| **Précondition** | Authentification |
| **Post condition** | Le profil client est consulté |
| **Scenario nominal** | 1. Le client choisit la commande mon compte. 2. La plateforme va afficher un formulaire avec les informations personnelles du client. [E1] 3. L’acteur fait consulter ses informations.[E2] |
| **Scenario d’erreur** | [E1] : erreur si les informations personnelles du client n’existent pas.  [E2] : erreur si le client a trouvé une faute d’orthographe dans son nom. |

## 2.5. Conclusion

Lors de ce chapitre nous avons spécifié les besoins fonctionnels et non fonctionnels de notre application web. Ensuite, selon la méthode « Cycle en V », nous avons présenté les raffinements associés à chaque cas d’utilisation.

# CHAPITRE 3 : CONCEPTION

## 3.1. Introduction

A l’issue de la spécification, nous consacrons cette section pour présenter la conception de ce projet à l’aide des différents diagrammes à aspect dynamique et une architecture à deux modèles SPA et MVC pour une bonne structuration.

### 3.2. Diagramme de séquence

#### 3.2.1. Définition

Un diagramme de séquence est un diagramme d’interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont. Ces diagrammes sont organisés en fonction de temps.

Le temps s’écoule au fur et à mesure que nous parcourons la page et les objets impliqués dans l’opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence de messages.

#### 3.2.2. Diagramme de séquences de l’acteur administrateur

* Diagramme de séquences « Gestion des commandes »

Pour confirmer une commande, l’administrateur fait son choix sur la commande « commandes confirmées », puis il clique sur le bouton « livrer ». Le système va transférer cette commande de la liste confirmées vers la liste commandes livrées.

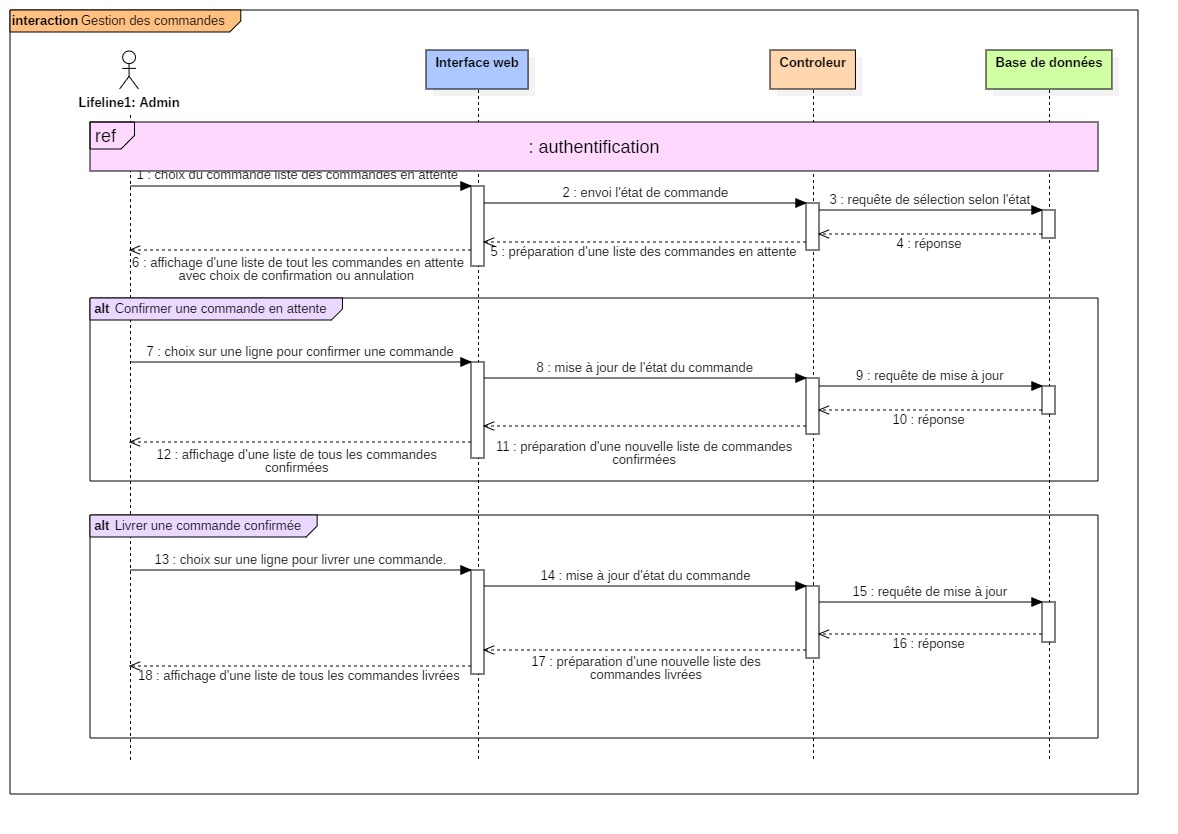


Figure  : Diagramme de séquences « Gestion des commandes »

* Diagramme de séquence « Gestion des produits »

Pour ajouter un nouveau produit, l’administrateur fait son choix sur la commande « produit », puis il clique sur « Ajouter » via l’interface de gestion des produits puis il remplit les champs du formulaire et enfin il valide les données qu’il a saisies en cliquant sur le bouton « confirmer ». Le système vérifie tout d’abord si l’administrateur a bien saisi les champs du formulaire.

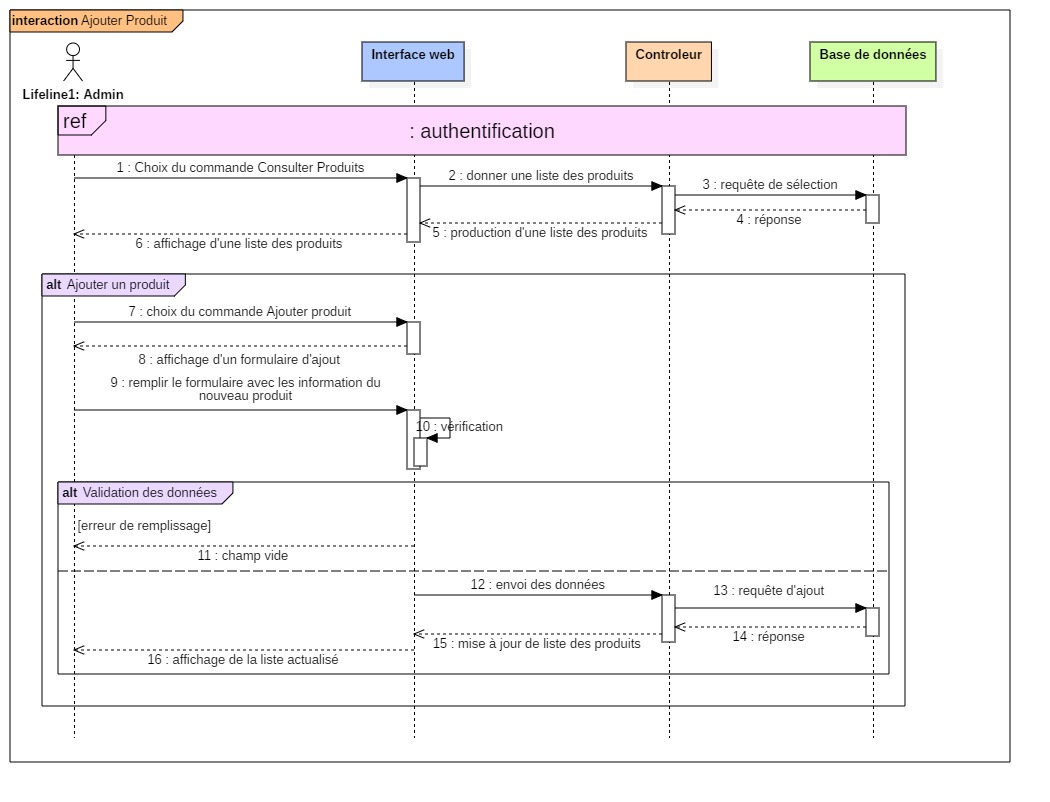


Figure  : Diagramme de séquence « Gestion des produits »

* Diagramme de séquence « Supprimer les paniers des visiteurs »

Pour supprimer un panier d’un visiteur, l’administrateur fait son choix sur la commande « supprimer panier », puis la plateforme va afficher une liste des paniers des visiteurs et des clients. L’administrateur fait son choix sur une ligne panier de visiteur et enfin il valide la suppression en cliquant sur le bouton « supprimer ».

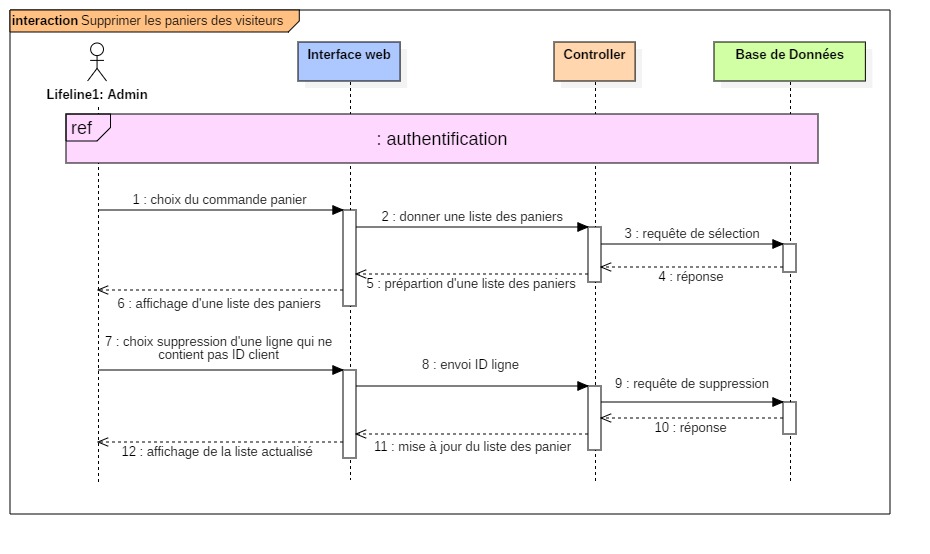


Figure  :Diagramme de séquence « Supprimer les paniers des visiteurs »

### 3.2.3. Diagramme de séquences de l’acteur visiteur

* Diagramme de séquence « S’inscrire »

Pour s’inscrire, le visiteur doit remplir un formulaire d’inscription. Dès que ce dernier est rempli, le système va vérifier s’il y a des champs vides ou bien des erreurs de remplissage sinon il le rediriger vers une page d’authentification et le visiteur devient un client.

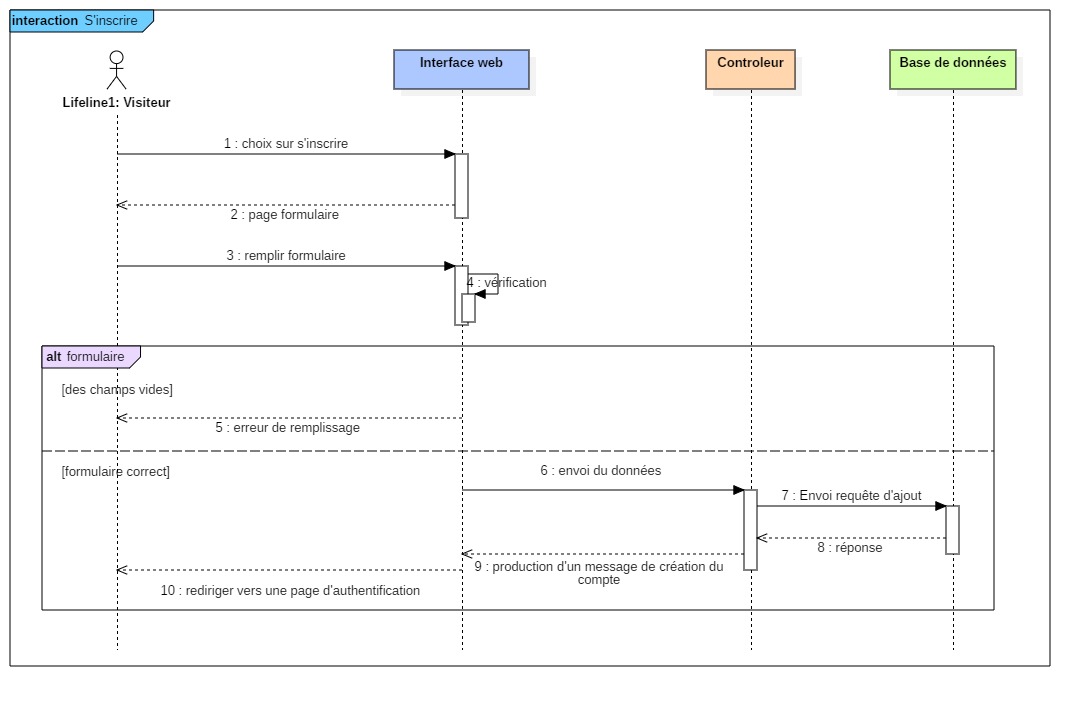


Figure  : Diagramme de séquence « S’inscrire »

* Diagramme de séquence « Consulter un produit »

Pour consulter un produit, le visiteur doit choisi une catégorie puis une sous-catégorie. Via la plateforme, le visiteur choisi un produit de la liste affichée. La plateforme va afficher ce produit en détails.

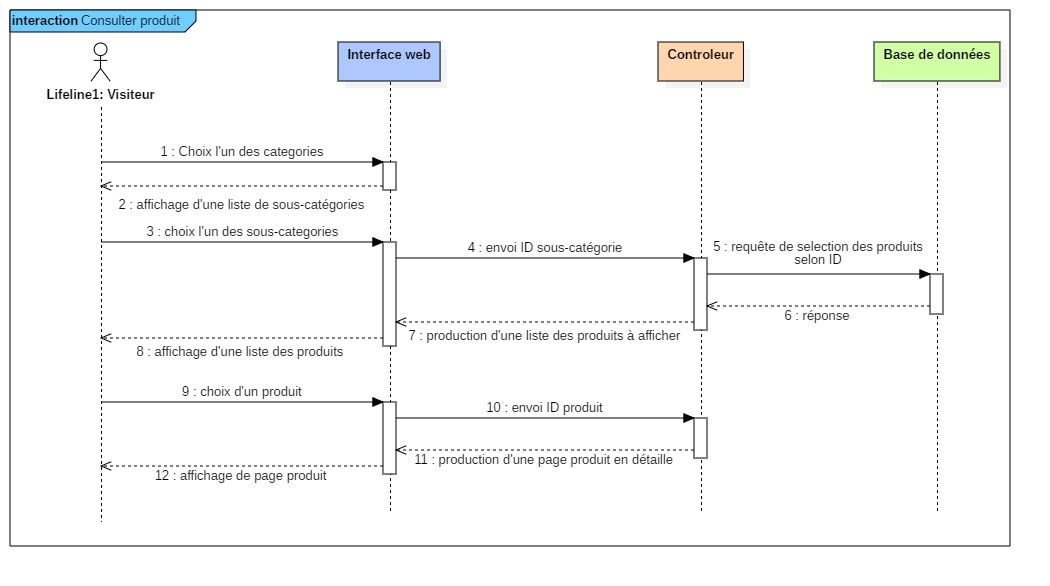


Figure  : Diagramme de séquence « Consulter un produit »

* Diagramme de séquence « Recherche un produit »

Pour rechercher un produit, le visiteur doit taper un mot clé au champ de recherche. La plateforme va suggérer un résultat.

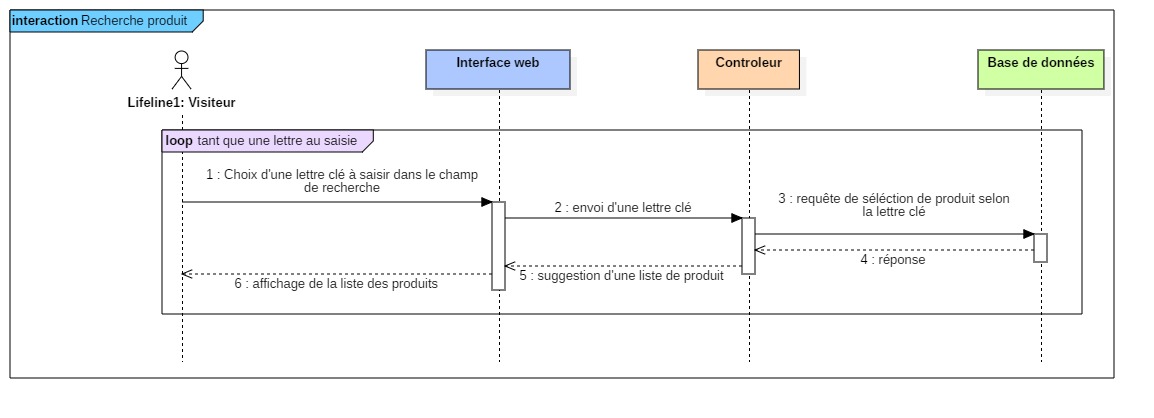


Figure  : Diagramme de séquence « Recherche un produit »

* Diagramme de séquence « Gestion panier »

Afin de modifier la quantité de produit sélectionnée au panier, le visiteur doit aller au page panier par le choix de la commande « panier ». La plateforme va donner une liste de produit sélectionné avec la proposition de modification quantité devant chaque ligne.

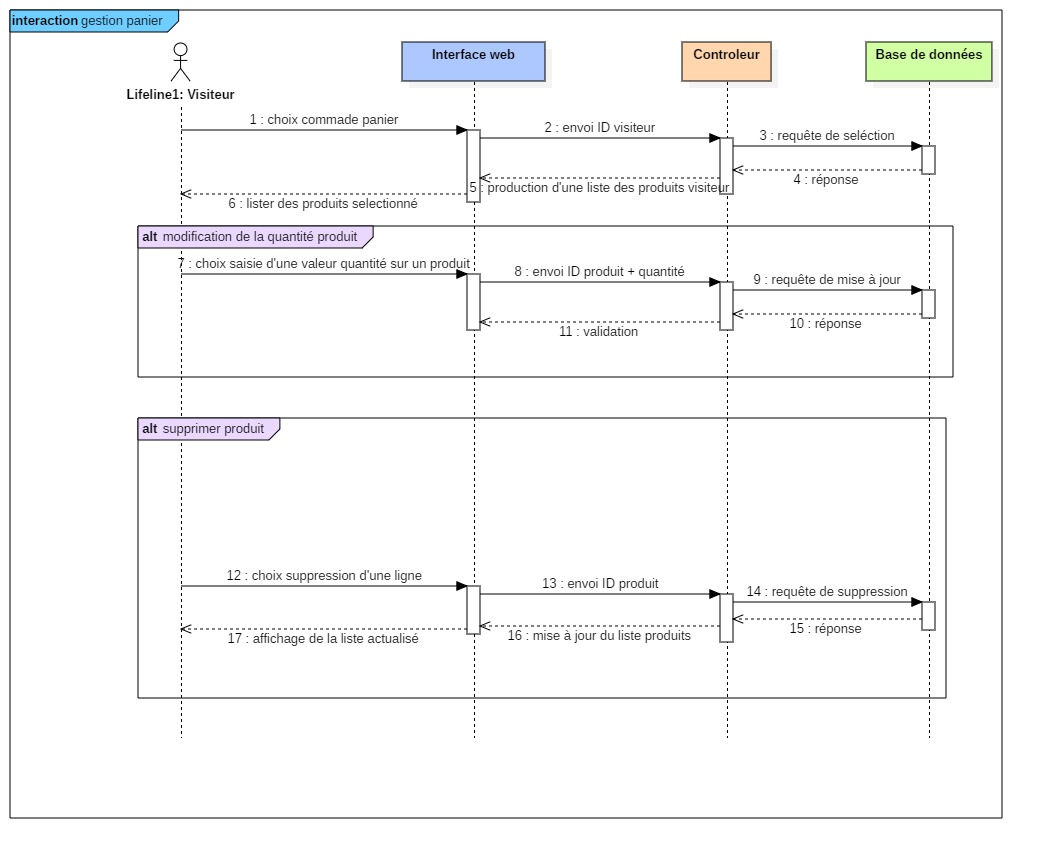


Figure  : Diagramme de séquence « Gestion panier »

### 3.2.4. Diagramme de séquences de l’acteur client

* Diagramme de séquence « Authentification »

Pour authentifier, le client doit remplir un formulaire d’authentification. Dès que ce dernier est rempli, le système va vérifier s’il y a des champs vides ou bien des erreurs de remplissage email et mot de passe sinon il le rediriger vers la page d’accueil.

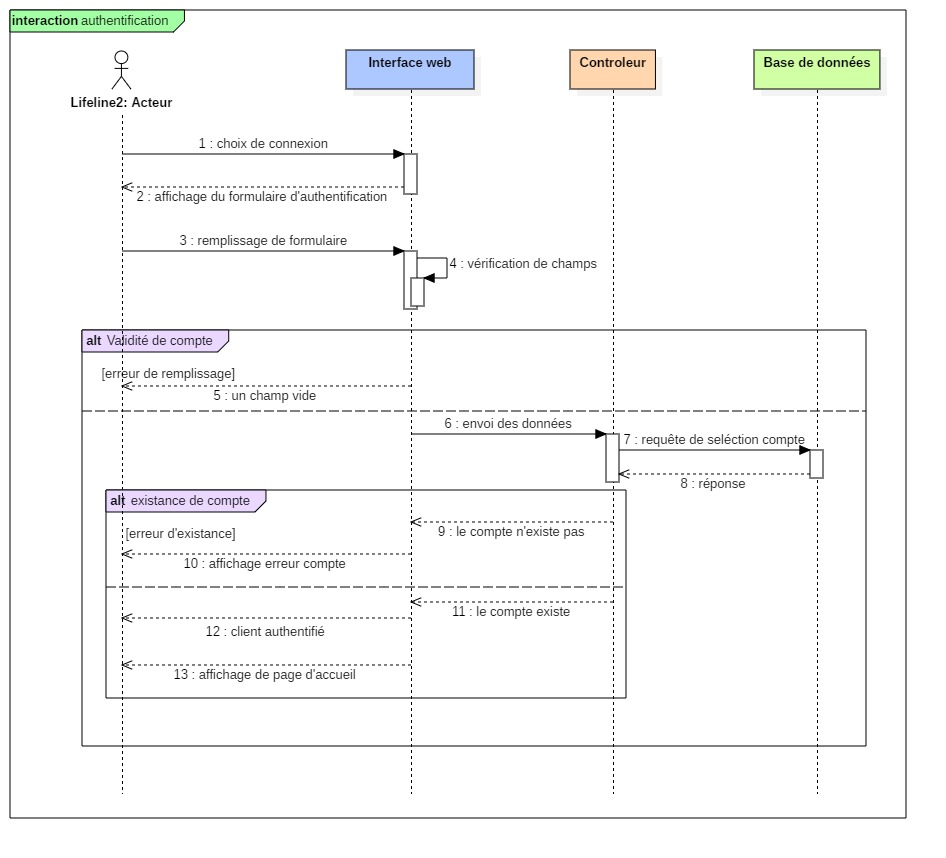
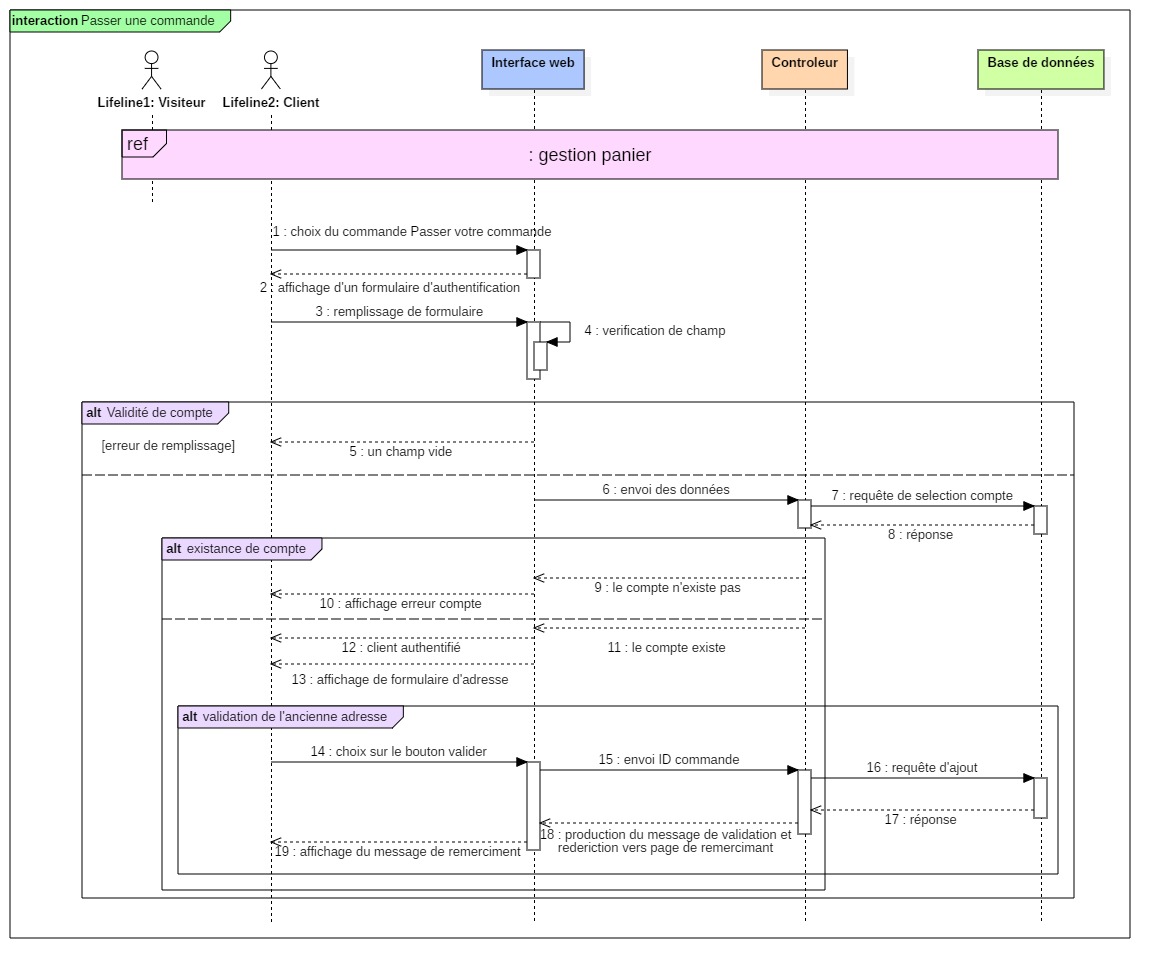


Figure  : Diagramme de séquence « Authentification »

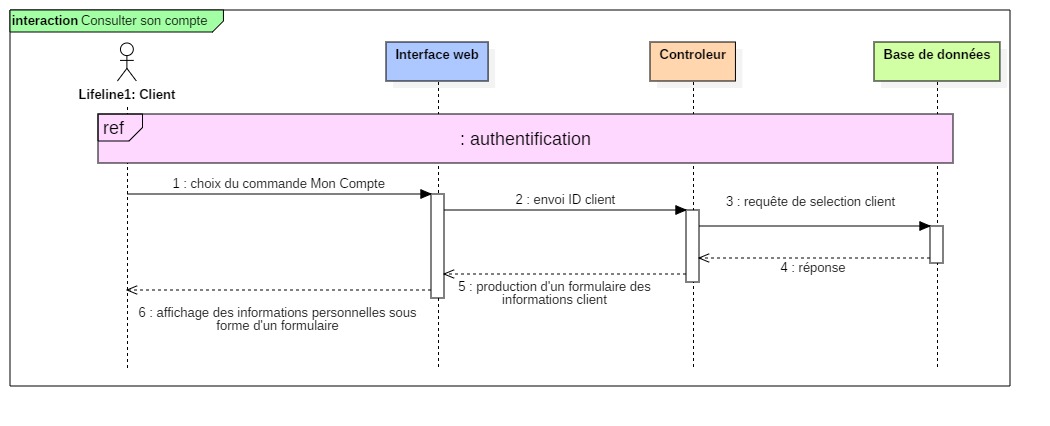
* Diagramme de séquence « Passer une commande »

Après avoir authentifier, le client peut passer son commande en choisissant la commande « passer la commande » dans la page panier. La plateforme va proposer un formulaire de changement d’adresse de livraison où le client peut la changer ou bien valider sa commande.

Figure  : Diagramme de séquence « Passer une commande »

* Diagramme de séquence « Gérer son compte »

Après avoir authentifier, le client peut gérer son compte via une commande « mon compte ». La plateforme va afficher un formulaire contient l’anciennes informations personnelles.

Figure  : Diagramme de séquence « Gérer son compte »

## 3.3. Diagramme d’activité

### 3.3.1. Définition

Les diagrammes d’activités permettent de mettre l’accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d’une méthode ou le déroulement d’un cas d’utilisation.

### 3.3.2. Diagramme d’activité d’acteur Administrateur

* Diagramme d’activité « Gestion des commandes »

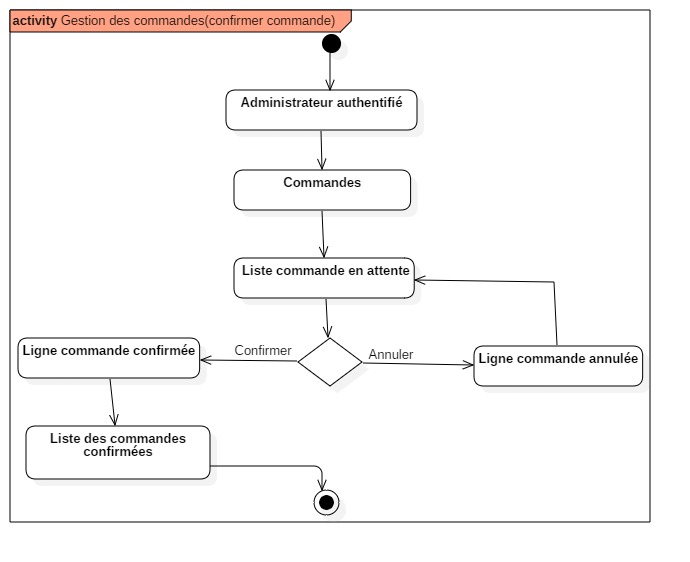


Figure  : Diagramme d’activité « Gestion des commandes »

* Diagramme d’activité « Gestion des produits »



Figure  : Diagramme d’activité « Gestion des produits »

* Diagramme d’activité « Supprimer des paniers des visiteurs »

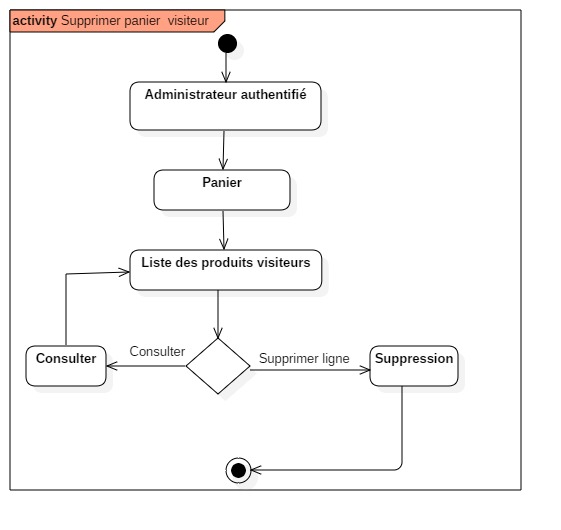


Figure  : Diagramme d’activité « Supprimer des paniers des visiteurs »

### 3.3.3. Diagramme d’activité d’acteur Visiteur

* Diagramme d’activité « S’inscrire »

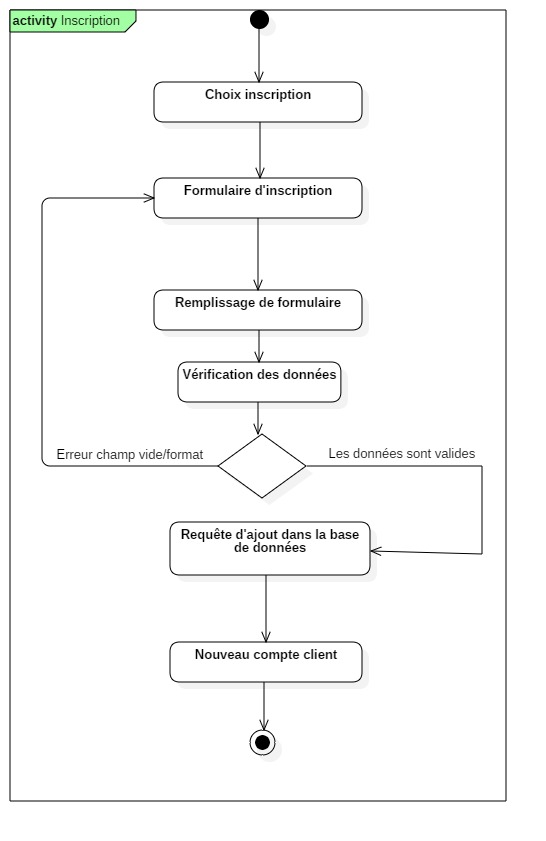


Figure  : Diagramme d’activité « S’inscrire »

* Diagramme d’activité « Consulter un produit »

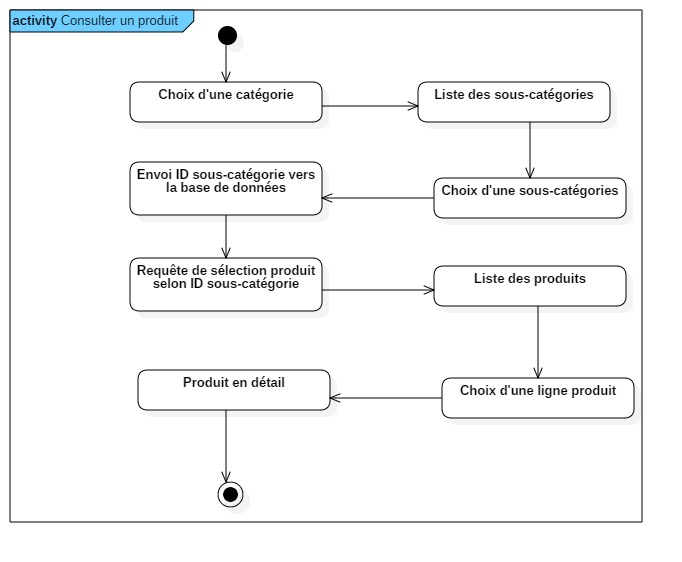


Figure  : Diagramme d’activité « Consulter un produit »

* Diagramme d’activité « Recherche un produit »

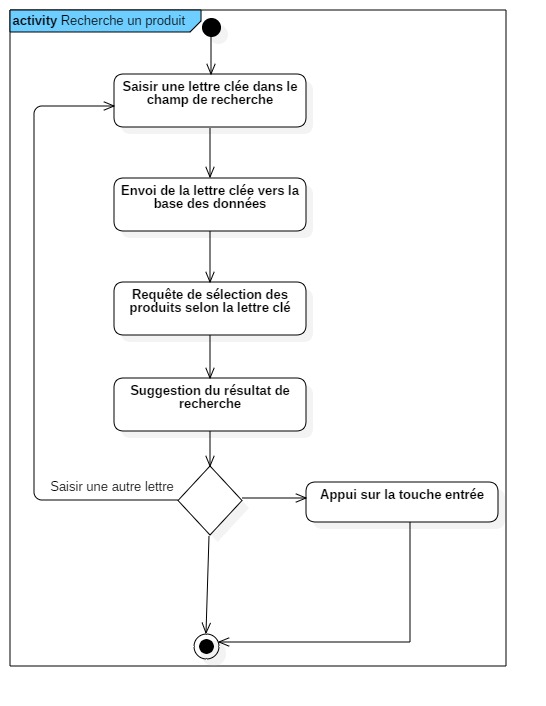


Figure  : Diagramme d’activité « Recherche un produit »

* Diagramme d’activité « Gestion panier »

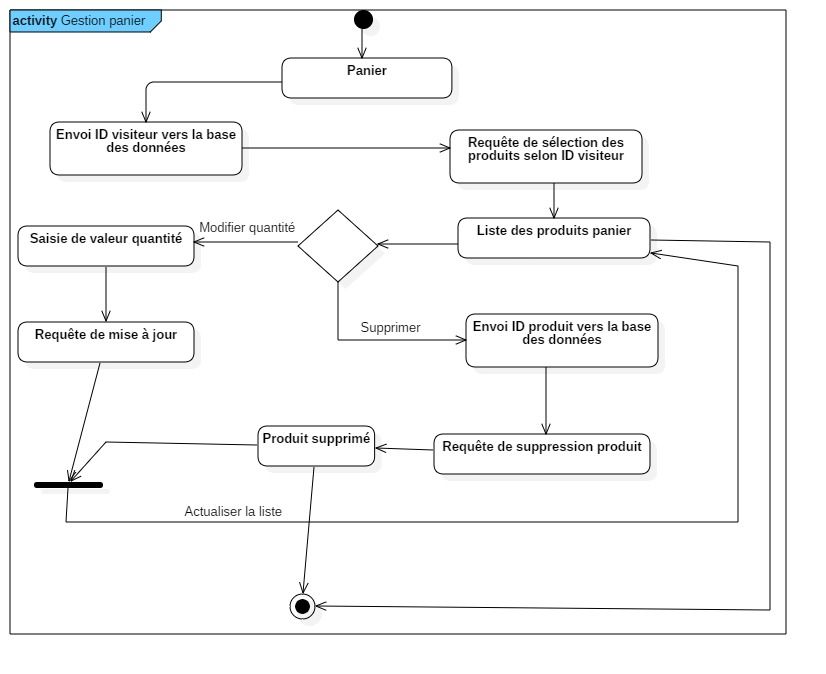


Figure  : Diagramme d’activité « Gestion panier »

### 3.3.4. Diagramme d’activité d’acteur Client

* Diagramme d’activité « Authentification »

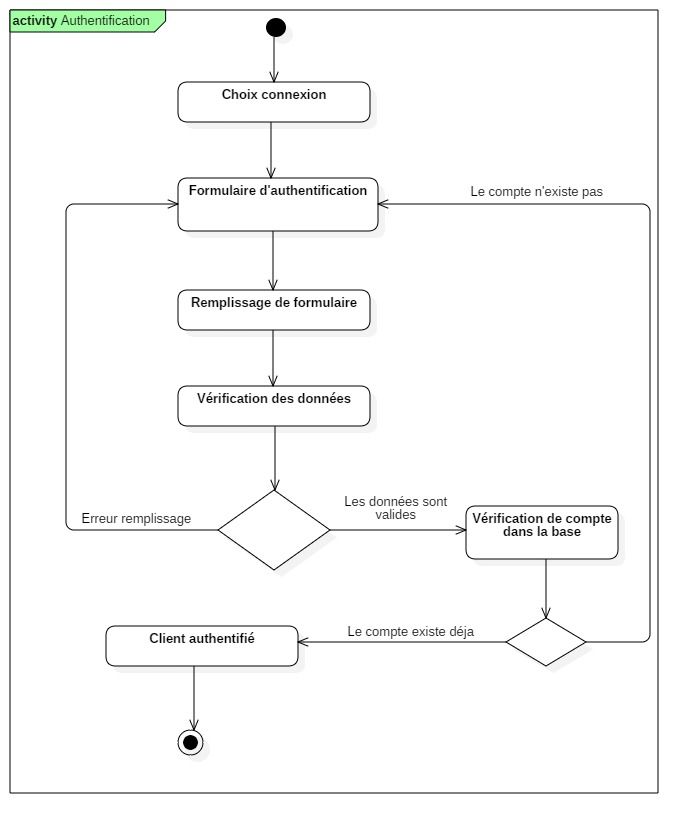


Figure  : Diagramme d’activité « Authentification »

* Diagramme d’activité « Passer une commande »

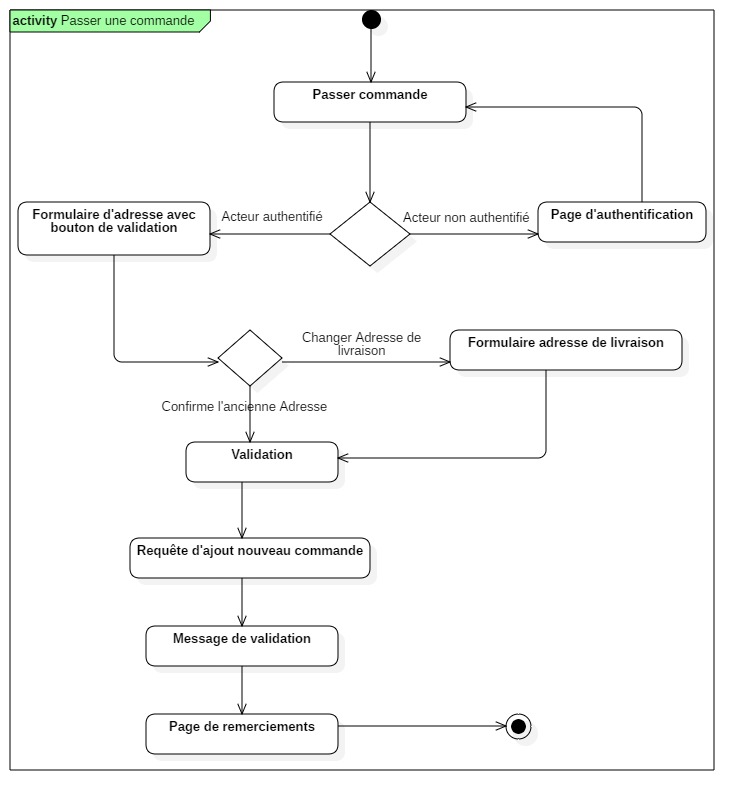


Figure  : Diagramme d’activité « Passer une commande »

* Diagramme d’activité « Gérer son compte »

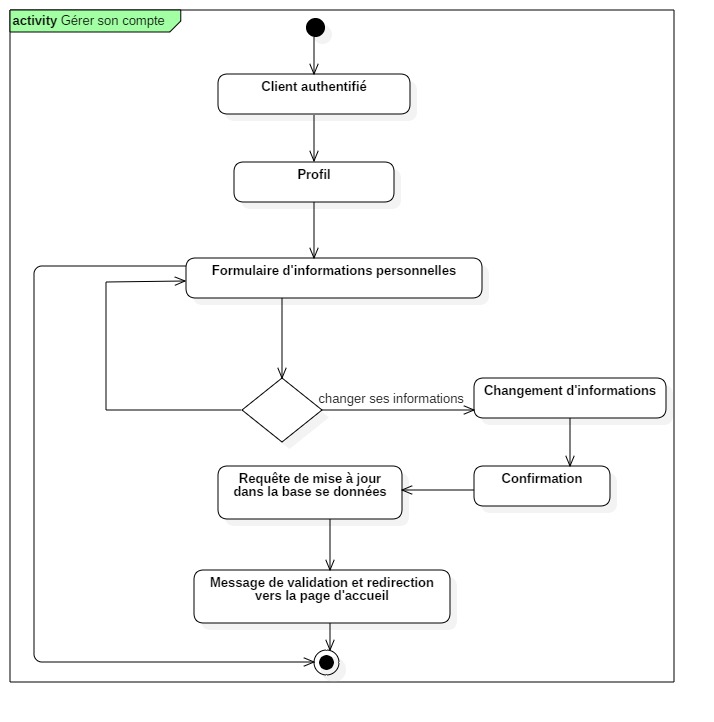


Figure  : Diagramme d’activité « Gérer son compte »

### 3.4. Diagramme de classes

#### 3.4.1. Définition

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d’un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

En générale un diagramme de classe peut contenir les éléments suivants :

* Les classes : une classe représente la description formelle d’un ensemble d’objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections.
* La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe.
* Les associations : une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc que des liens peuvent exister entre des instances des classes.
* Les attributs : les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l’attribut doit être unique dans la classe.

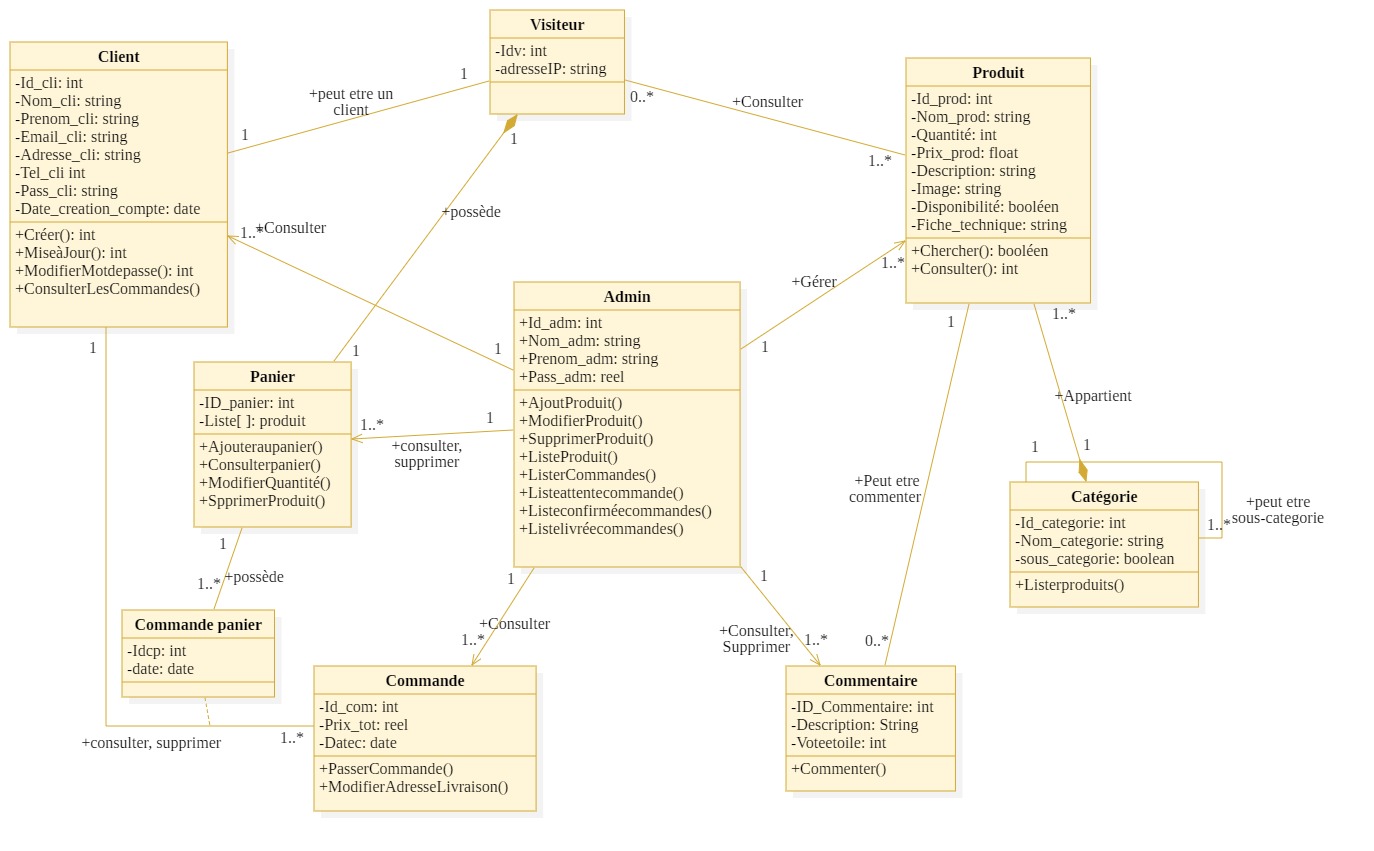


Figure  : Diagramme des classe

#### 3.4.2. Description des classes

Le diagramme de classes du notre plateforme comporte huit classes :

* Classe Administrateur : C’est la classe qui représente une personne responsable au service informatique de la société. Cette personne peut gérer les produits de notre plateforme, gérer les commandes, consulter et supprimer les commentaires, consulter la liste des clients et consulter, supprimer les lignes produites des visiteurs.
* Classe client : représente les utilisateurs de notre plateforme qui possèdent un compte d’accès, permettant de passer des commandes au niveau de leur panier et consulter, modifier son profil. Il peut se comporter comme un acteur visiteur, à savoir il peut consulter les produits, les ajouter dans son panier, gérer son panier et commenter ses produits.
* Classe produit : c’est une classe qui représente un ensemble des produits qui seront exposés dans notre plateforme afin d’être vendus et livrés chez les clients. Un produit est caractérisé par un nom, une quantité, un prix et une description.
* Classe catégorie : c’est une classe qui permet de classer l’ensemble des produits selon un mode d’utilisation, une catégorie possède un ensemble des sous-catégories et chaque sous-catégorie représente un ensemble des produits qui sont spécifiques au sous-catégorie.
* Classe Panier : représente un ensemble des produits qui sont sélectionnés par l’acteur visiteur, on peut avoir des informations sur la quantité, les prix et l’image de chaque produit.

## 3.5. Architecture de l’application

L’architecture de l’application est la manière dont nous l’organiserons, pour assurer une qualité et une structuration formelle. Elle souvent modélisée de manière à présenter le logiciel (par exemple, le code d’application et plate-forme) et le matériel (le client, le serveur et les périphériques réseau) d’une façon globale.

L’architecture de l’application est habituellement modélisée en terme d’une architecture Client/Serveur dans laquelle un ou plusieurs dispositifs, clients, demandent des informations à partir d’un dispositif de type serveur.

Le serveur répond en générale avec les renseignements demandés :

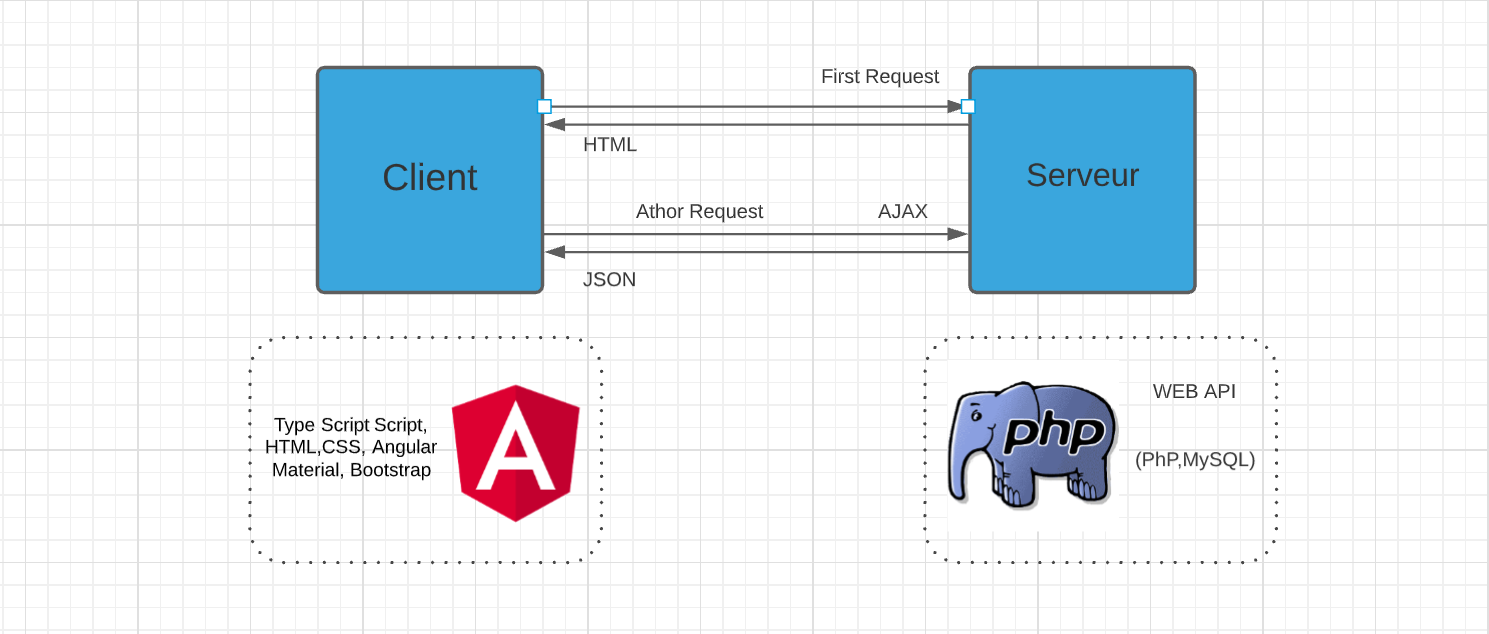


Figure  : Architecture Client-Serveur

#### 3.5.1. Modèle SPA

Une Single Page Application (SPA, application web monopage) est une application accessible via une page web qui contient tout. Son intérêt est d’éviter le chargement d’une nouvelle page à chaque action demandée, et de fluidifier ainsi l’expérience utilisateur. L’ensemble des éléments de l’application est chargé (html, images, css, JavaScript et données) en une fois (ou plus exactement à la demande en fonction des actions de l’utilisateur).

La SPA est accessible via une adresse URL dans un navigateur et constitué de quelques fichiers qu’il convient de rendre accessible à l’utilisateur soit sur un réseau privé ou public à condition de sécuriser les données.

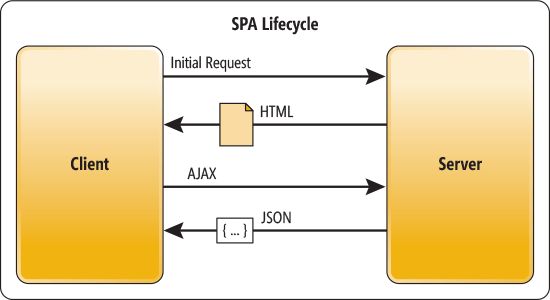


Figure  : Modèle S-P-A

#### 3.5.2. Modèle MVC

Le modèle MVC, « Model-View-Controller » est une architecture à trois couches utilisées pour la programmation Client/Serveur et d’interface graphique. C’est un modèle architectural très puissant qui intervient dans la réalisation d’une application. Il tire sa puissance de son concept de base qui est la séparation des données (modèle), de l’affichage (vue) et des actions (contrôleur). Ces trois couches sont décrites comme suit :

* Le modèle : administre les données de l’état de l’application et assure la gestion des données manipulées par les vues et garantit leur intégrité ; dans la plupart des cas typiques le modèle est en interaction directe avec la base de données.
* La vue : représente les interfaces en interaction avec l’utilisateur, responsable d’afficher les données transmises par le modèle et récupérer les entrées (évènements, action, …) fournies par l’utilisateur.
* Le contrôleur : afin de synchroniser les données entre modèle et vue, le contrôleur prend les actions de l’utilisateur et lui fournit la vue correspondante, communique les changements à faire au modèle, puis notifie la vue des mises à jours.
* Ces interactions entre les différents acteurs sont schématisées comme suit :

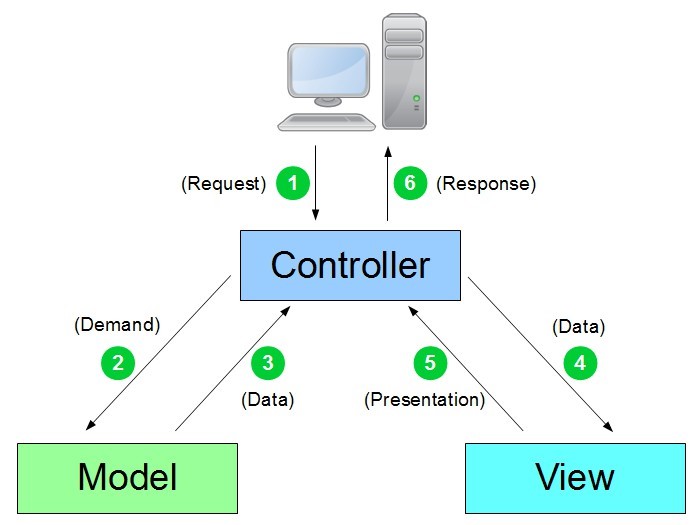


Figure  : Modèle M-V-C

## 3.6. Conclusion

Ce chapitre a été consacré à présenter en détail, L’architecture et la conception de notre solution, à travers les diagrammes de cas d’utilisation de diagrammes de classe, de diagrammes de séquence et d’activité

Pour conclure alors, une conception affinée de l’applicatif a été mise en place, en fixant les chartes qui vont être utilisées pour la réalisation du projet. Il a fallu alors passer à une étape

Qui est aussi importante que la conception, la réalisation de projet, cette étape qui permet de mettre en œuvre la conception déjà faite, et qui l’objet du chapitre suivant

# CHAPITRE 4 : REALISATION

## 4.1. Introduction

Après avoir terminé, les phases d’analyse, spécification des besoins et la conception de notre plateforme, nous focalisons maintenant à la réalisation. En effet, nous commençons cette partie pour la description de l’environnement de développement, ainsi que nos choix technologiques utilisés. Ensuite, nous nous intéressons à la description de quelques interfaces implémentés dans le cadre d’un scénario des cas d’utilisations, qui ont été adoptés dans les chapitres 2 et 3.

## 4.2. Environnement de développement

### 4.2.1. Environnement logiciel

#### 4.2.1.1. Editeurs de développement

Tableau  : Environnement de développement

|  |  |  |
| --- | --- | --- |
| **Nom** | **Description** | **Utilisation** |
| **Visual Studio**  **Code** | Visual Studio Code est un éditeur de code cross-Platform, open source et gratuit, supportant une dizaine de langages. | Nous avons utilisé Visual Studio Code comme environnement de développement de notre plateforme des côtés Front End et Back End. |

#### 4.2.1.2. Langages de programmation

Tableau  : Langages de développement

|  |  |  |
| --- | --- | --- |
| **Nom** | **Description** | **Utilisation** |
| **HTML 5** | Le HTML5 est un langage de base pour la création de site internet, il sert à structurer notre document. D’autre langage peuvent s’ajouter lors de la conception, mais tous les interfaces web contiennent du HTML. | L'utilisation de Framework Angular et la création de Dashboard nécessite l’utilisation de HTML5 pour créer et structurer les pages Web. |
| **CSS 3** | Le terme CSS est l’acronyme anglais de Cascading Style Sheets. Le CSS est un langage informatique utilisé sur l’internet pour mettre en forme les fichiers HTML. Les fichiers CSS, comprennent du code qui permet de gérer le design d’une page en HTML. | L'utilisation de Framework Angular et la création de Dashboard nécessite l’utilisation de CSS 3 pour décrire la présentation des documents HTML. |
| **TYPESCRIPT** | Le Type Script est un langage de programmation de scripts orienté objet. | L’utilisation de Framework Angular nécessite l’utilisation de TYPESCRIPT pour développer la plateforme. |
| **PHP** | Le PHP est un langage de script coté serveur dont la couche métier pour générer du web dynamique via le serveur Apache. | La création de Back End nécessite l’utilisation de PHP pour développer la plateforme. |
| **JAVASCRIPT** | Le JavaScript est un langage de programmation de scripts orienté objet. | La création de Back End nécessite l’utilisation de JAVASCRIPT pour développer la plateforme. |

#### 4.2.1.3. Système de gestion de base de données

Tableau  : Gestion de base de données

|  |  |  |
| --- | --- | --- |
| **Nom** | **Description** | **Utilisation** |
| **phpMyAdmin** | phpMyAdmin est une application web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL. | Nous avons utilisé le phpMyAdmin comme un système de gestion de base de données pour accéder à notre base de données et pour gérer les données en temps réel. |

### 4.2.2. Choix technologie

#### 4.2.2.1. Framework côté client : Angular

« Angular » a été Développé par Google, c’est un « Framework » open source écrit en JavaScript qui permet la création d’applications Web et plus particulièrement de ce qu’on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l’expérience utilisateur et d’éviter les chargements de pages à chaque nouvelle action. Le « Framework » est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d’architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.

#### 4.2.2.2. Serveur de développement : WAMP

« WAMPServer » est une plateforme de développement Web intégrant plusieurs modules  
qui facilite la mise en œuvre des différents requis du développement, installé sous le système  
d’exploitation Windows qui est le responsable de la gestion des ressources.  
Cette plateforme est de type WAMP l’acronyme informatique de « Windows » « Apache »  
« MySQL » « PHP » permettant la création des applications Web dynamique grâce à :

* Serveur Apache : est un serveur web responsable de répondre aux requêtes du client  
  respectant le protocole de communication client-serveur http "HyperText Transfer  
  Protocol".
* Langage de script PHP « HyperText Preprocessor » : langage coté serveur dont la  
  couche métier est principalement écrite utilisé pour générer du web dynamique via  
  le serveur Apache.
* MySQL : est système de gestion de base de données administrant via l’application  
  phpMyAdmin une base de données relationnelle.

## 4.3. Réalisation

### 4.3.1. Réalisation des interfaces Administrateur

* Interfaces Gestion des commandes :

Les figures suivantes illustrent des captures écran de cas d’utilisation « Gestion des commandes », avec deux scénarios d’utilisations : confirmer une commande en attente et livrer une commande confirmée :

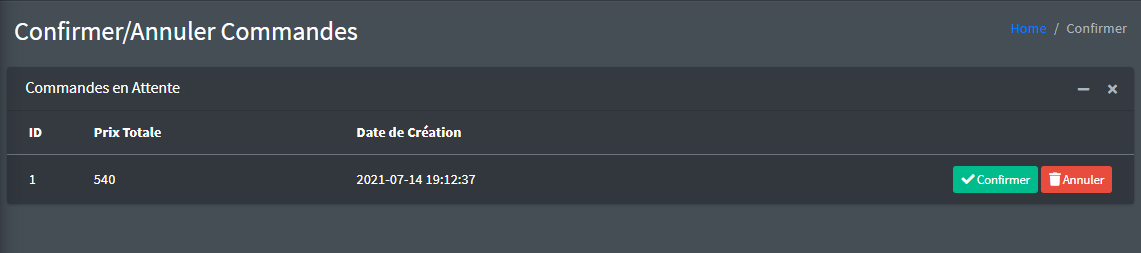


Figure  : Interface « commandes en attente »

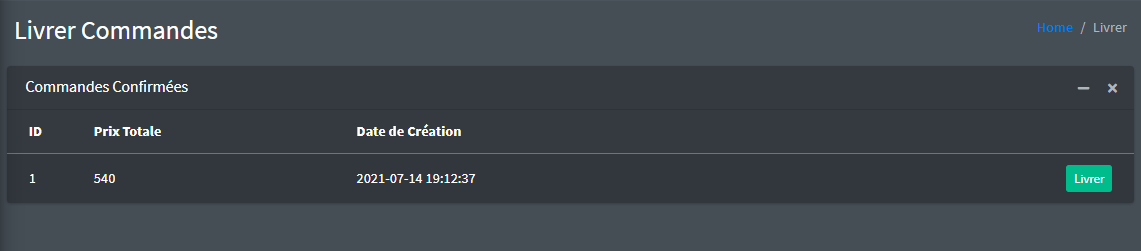
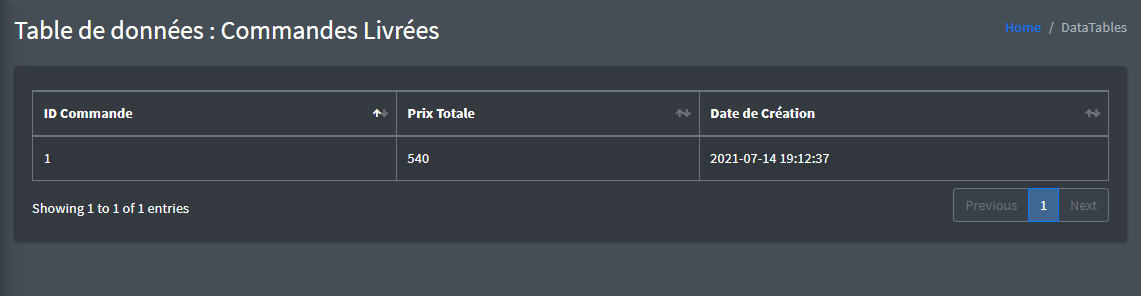


Figure  : Interface « commandes confirmées »

 Figure  : Interface « commandes livrées »

* Gestion des produits :

Les figures suivantes illustrent des captures écran de cas d’utilisation « Gestion des produits », avec un scénario d’utilisation choisi ajouter un nouveau produit.

* La figure 47 montre comment afficher une liste des produits, qui sont enregistrés dans la base de données, avec la possibilité de modifier, supprimer et ajouter un produit.

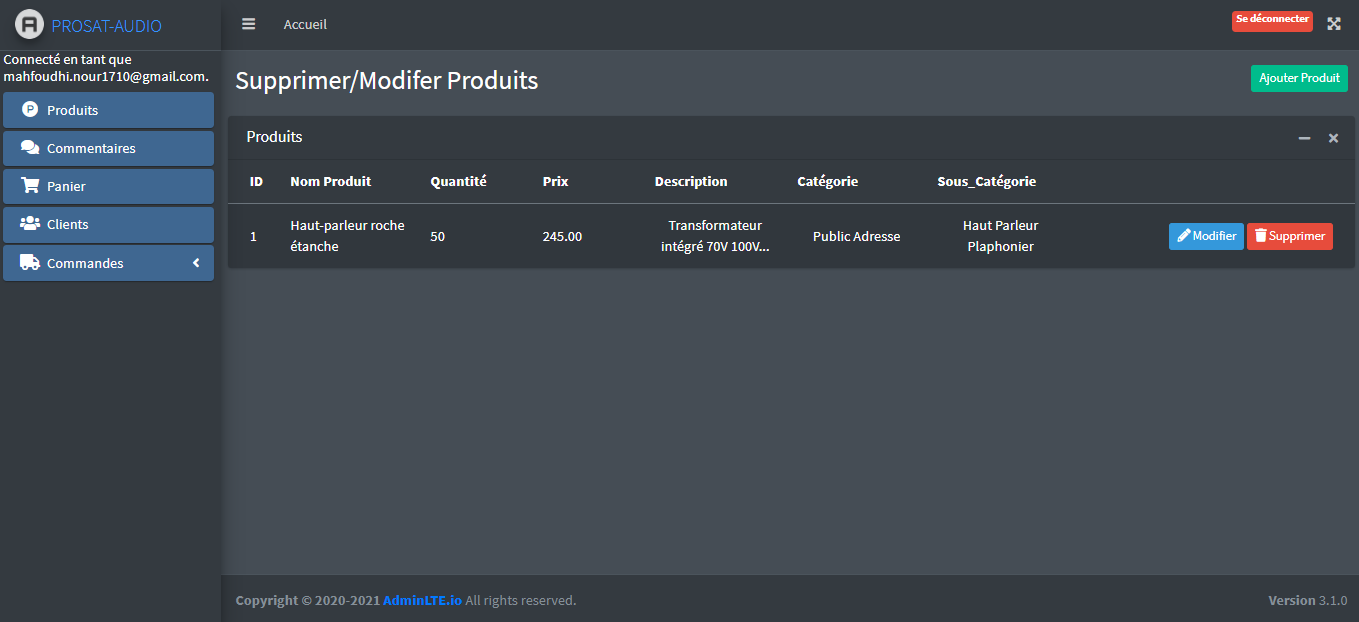


Figure : Interface « gestion des produits »

* En choisissant, le bouton vert Ajouter produit, notre plateforme fait rendre un formulaire d’ajout d’un nouveau produit.

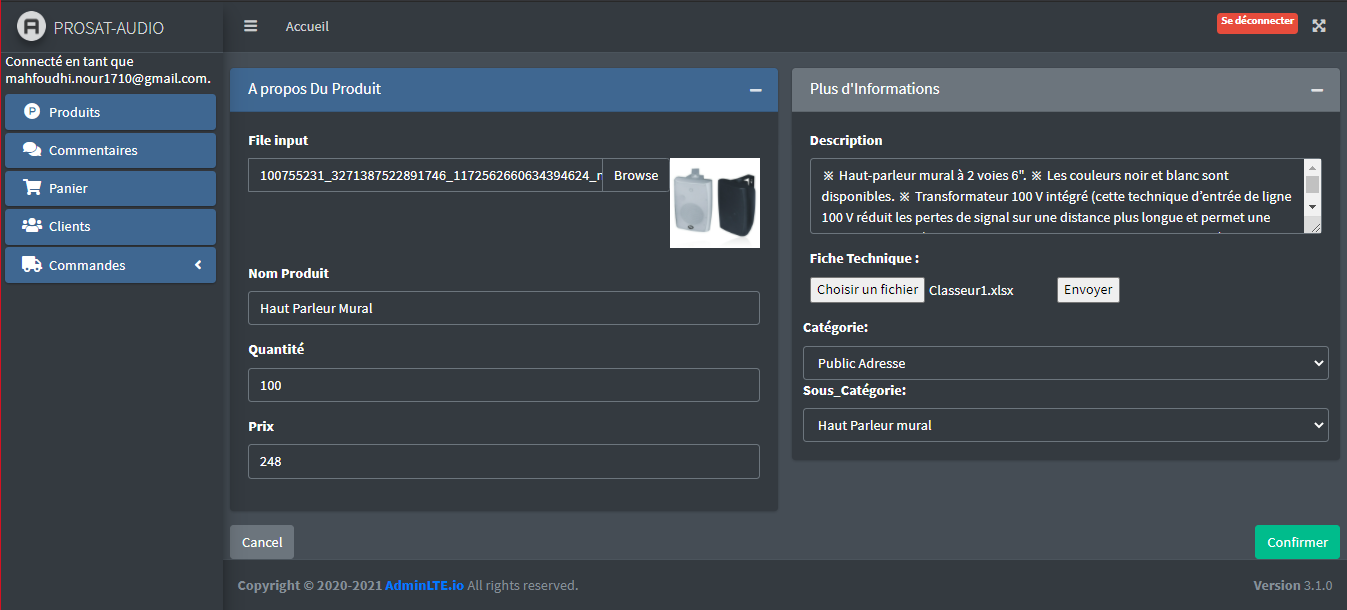


Figure  : Interface « gestion des produits »

Après l’ajout d’un produit, notre plateforme fait une opération de mise à jour sur la liste globale des produits. La figure montre l’affichage de la liste actualisée des produits.

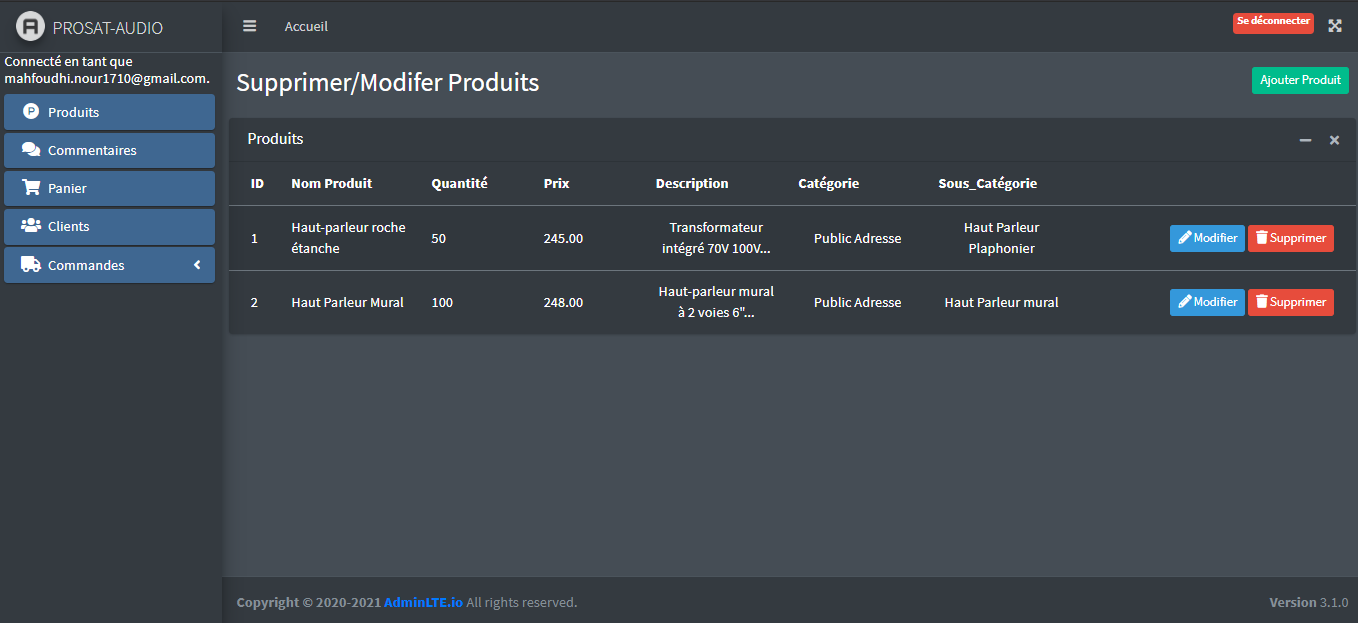


Figure  : L’interface « Gestion des produits » (produit ajouté)

* Supprimer paniers visiteurs :

Les figures suivantes montrent la suppression des lignes produits des paniers visiteurs :

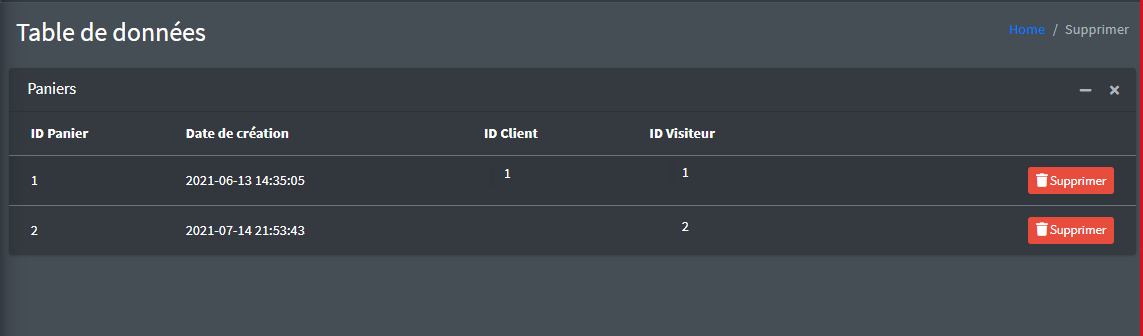


Figure  : Interface « Supprimer panier visiteur »

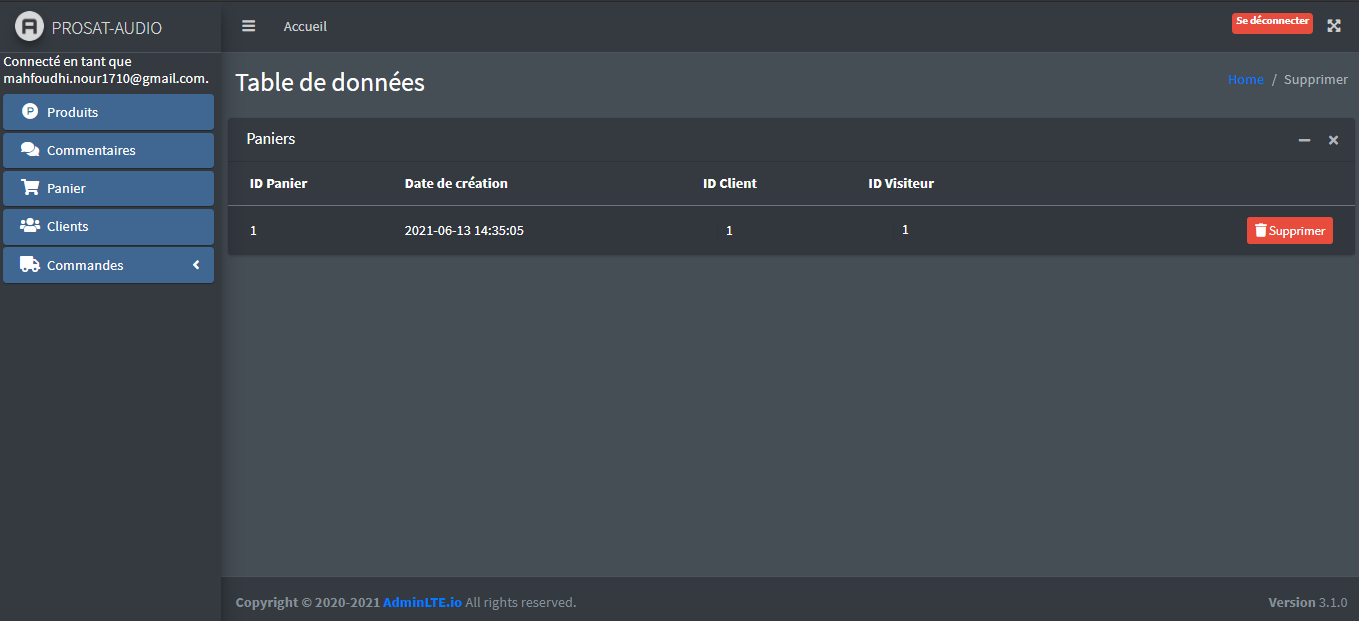


Figure  : Interface « Supprimer les lignes produits visiteurs »

#### 4.3.1. Réalisation des interfaces Visiteur

* S’inscrire :

La figure 52 illustre un formulaire d’inscription de visiteur :

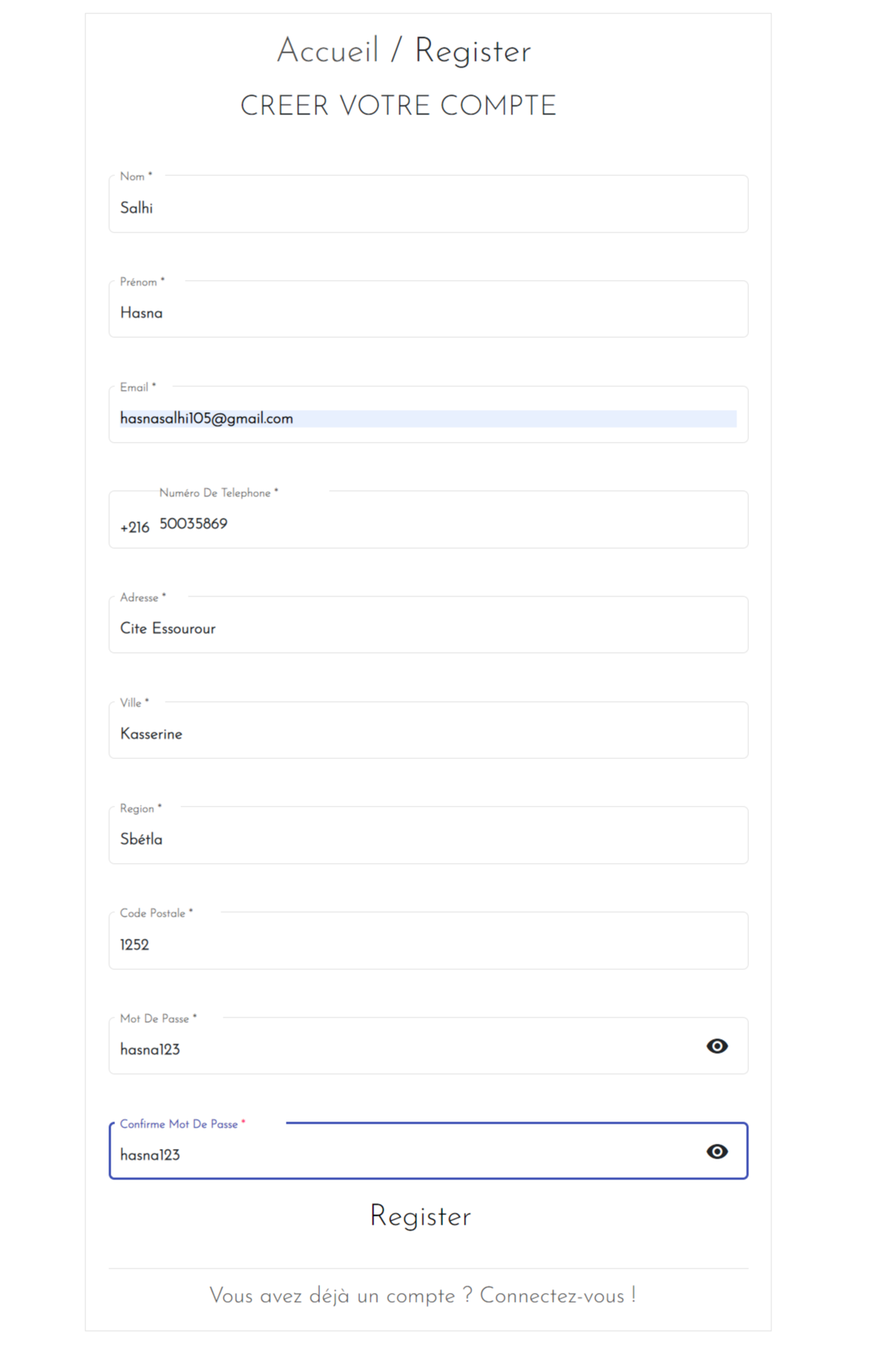


Figure  : Interface « S’inscrire »

* Consulter produit :

Les figures suivantes illustrent les étapes de consultation d’un produit par catégories :



Figure  : Interface « liste des catégories »

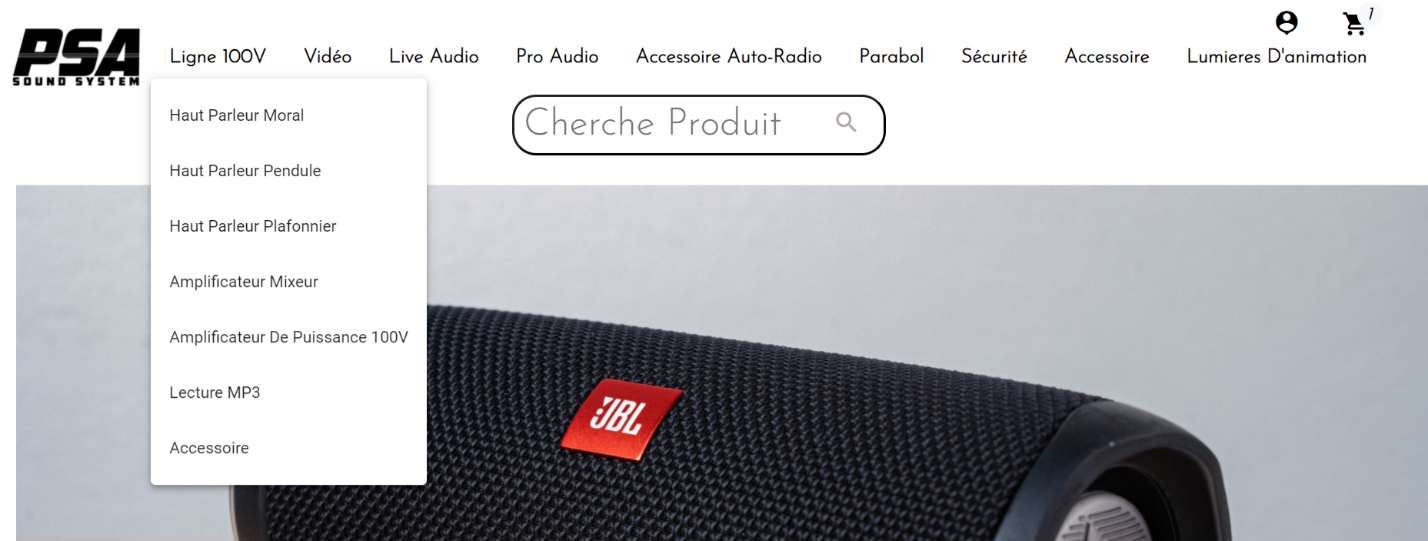


Figure  : Interface « liste des sous-catégories »



Figure  : Interface « liste des produits »

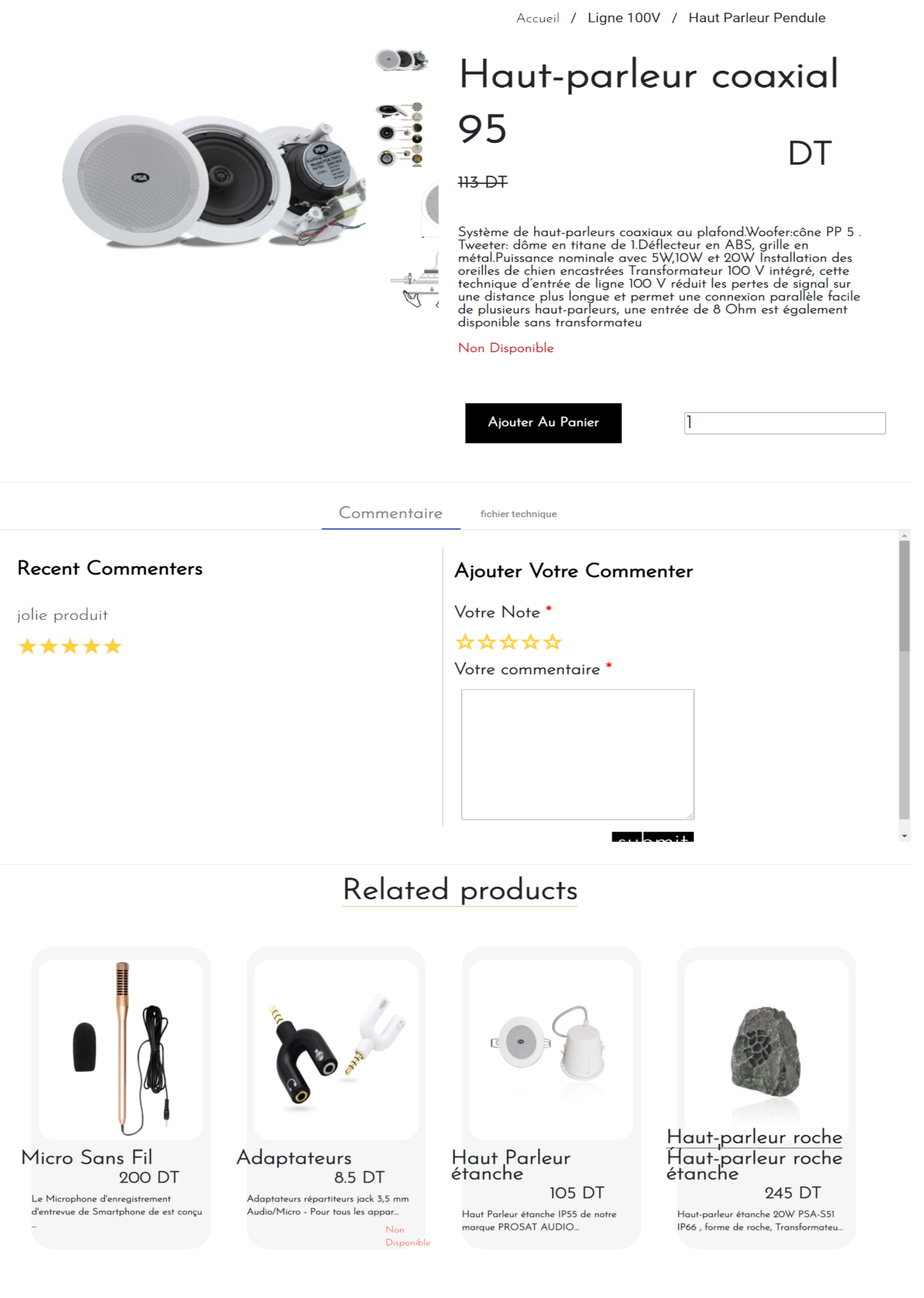


Figure : Interface « produit consulté »

* Recherche produit :

La figure suivante illustre le cas d’utilisation chercher un produit, par l’utilisation d’un mot clé :



Figure  : Interface « Rechercher un produit »

* Gestion de panier :

Les figures suivantes illustrent la gestion de panier visiteur, avec deux scénarios d’utilisations possibles : la modification de quantité et la suppression d’un produit.

* La figure 58 montre la liste des produits, qui sont ajoutés au panier de visiteur.

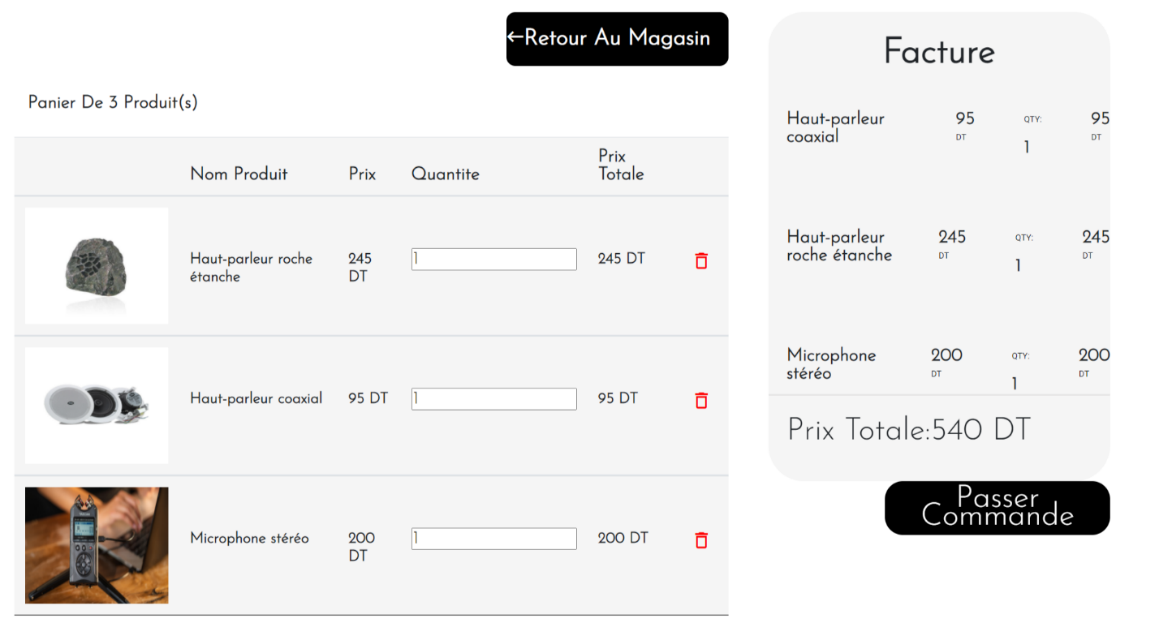


Figure  : Interface « Panier actuel d’un visiteur »

La figure 59 illustre l’affichage d’une liste actualisée des produits dans un panier visiteur.

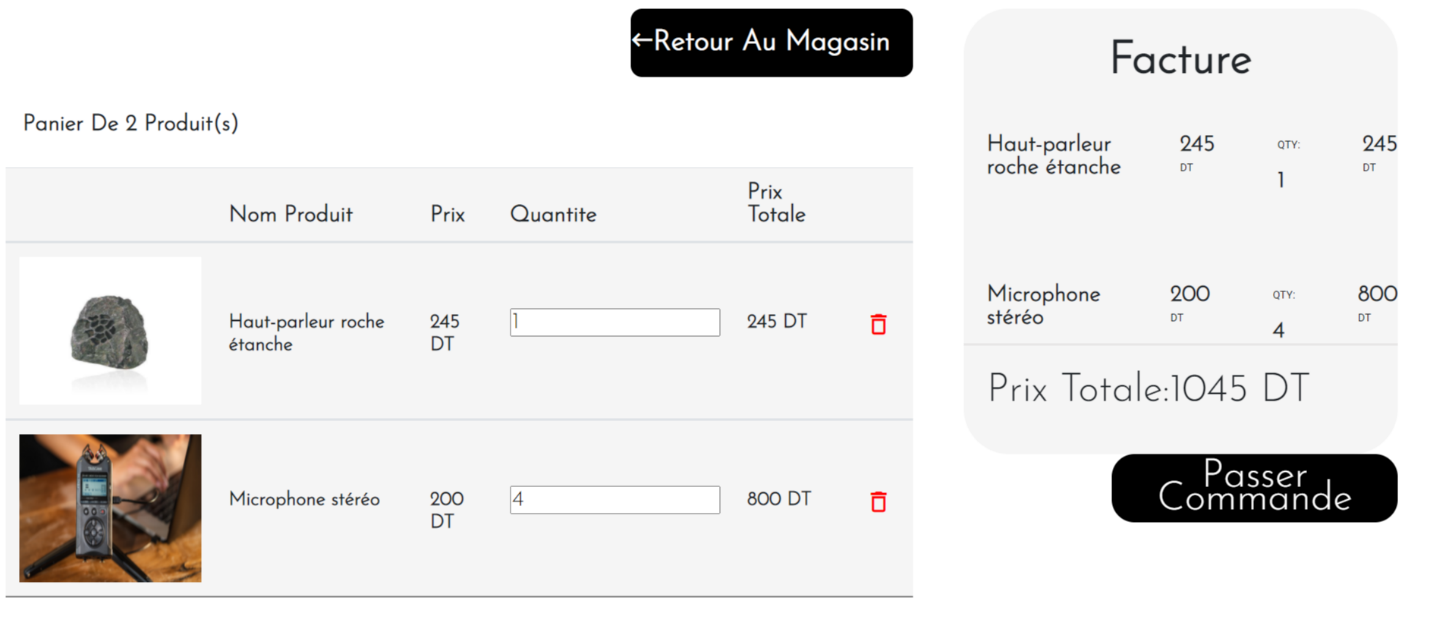


Figure  : Interface « Gestion panier visiteur »

**4.3.3. Réalisation des interfaces Client :**

* Authentification :

La figure 60 ci-dessous montre le formulaire d’authentification de client :



Figure  : Interface « Authentification »

* Passer une commande :

Les figures suivantes illustrent des captures écran de cas d’utilisation « Passer une commande », dans le but de confirmer ou modifier l’adresse de livraison et par la suite pouvoir passer une commande :

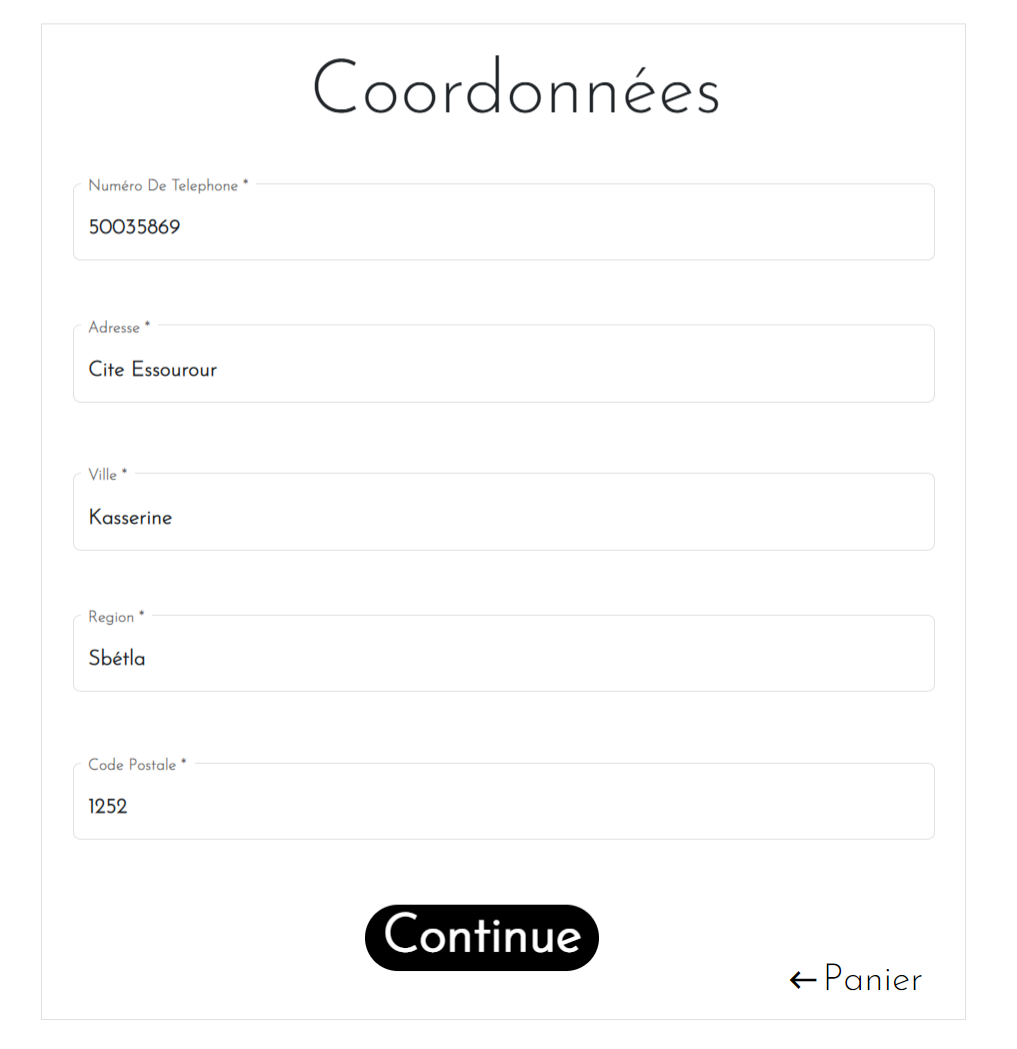


Figure  : Interface « Passer une commande »

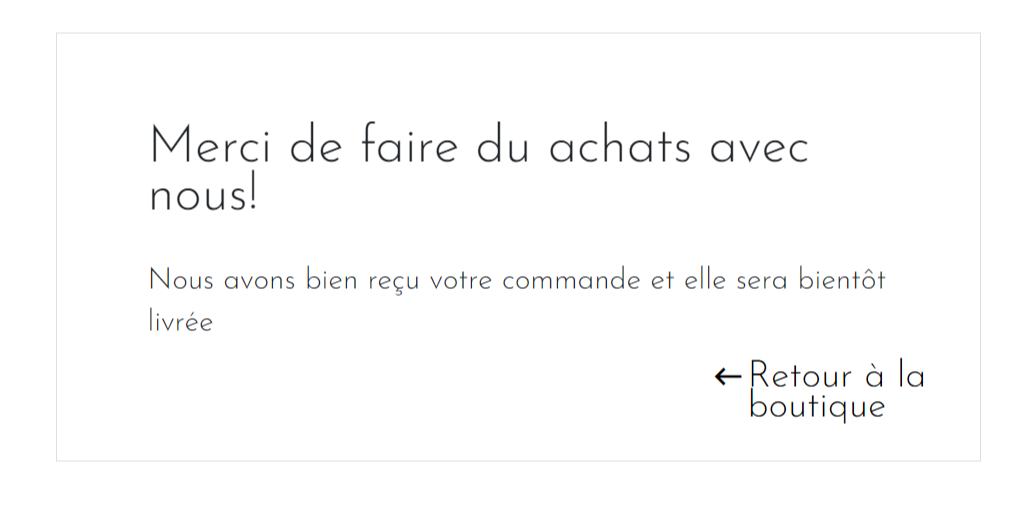


Figure  : Interface « Remerciement de client »

* Gérer son compte :

Les figures suivantes illustrent un formulaire qui contient des informations personnelles d’un client avec la possibilité de modification :

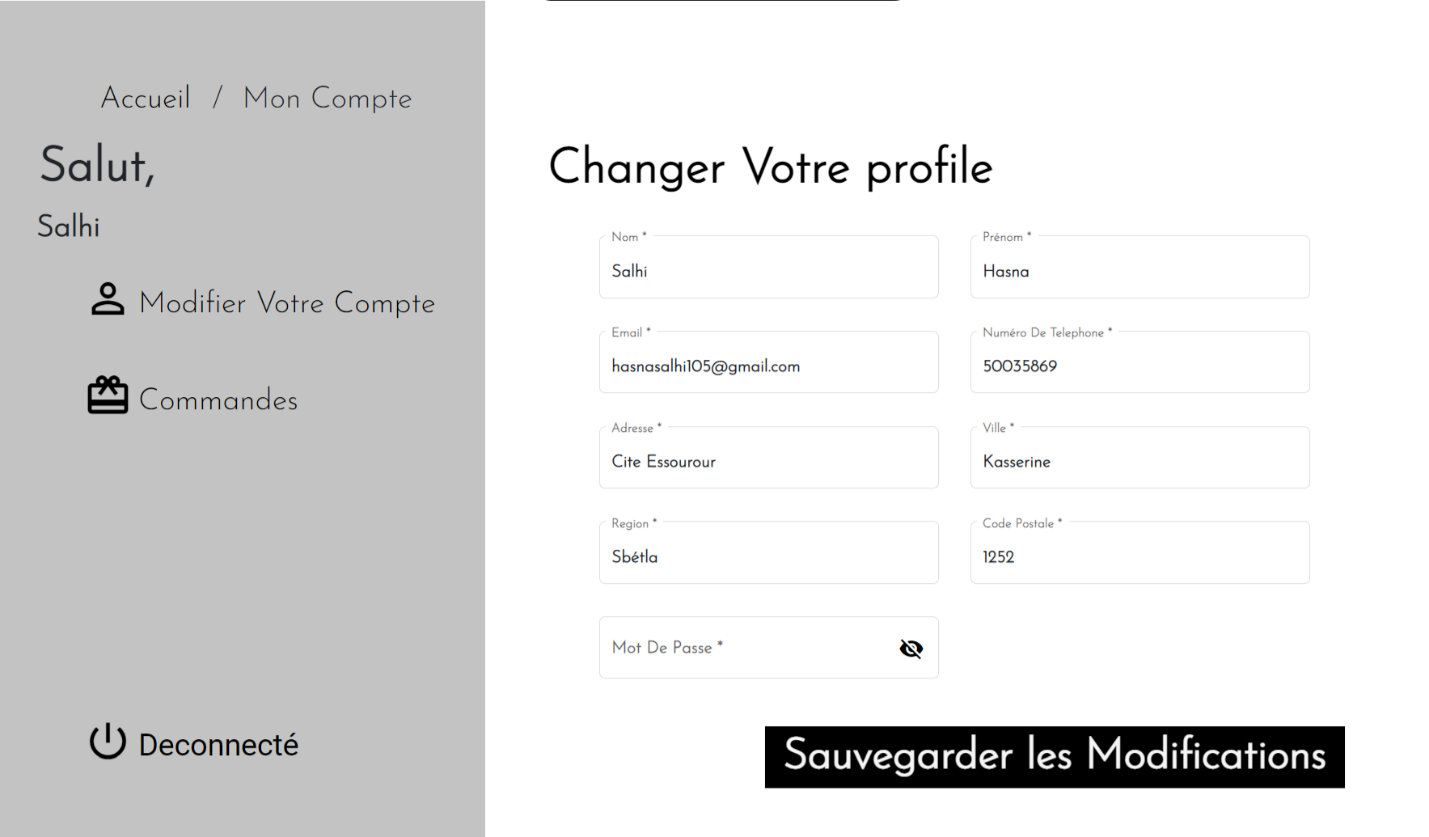


Figure  : Interface « Gérer son compte »



Figure  : Interface « Compte modifié »

## 4.4. Conclusion

Dans ce chapitre nous avons présenté l’environnement de développement et de base de données utilisé tout au long du projet. Nous avons également présenté quelques captures d’éran montrant le bon fonctionnement de notre projet.

# CONCLUSION GENERALE

***D***

urant ce projet de fin d’études, nous avons conçu de développer une plateforme au bénéfice de la société « PROSAT-AUDIO ». Les objectifs définis au départ ont été a 80% atteint. En guise de conclusion, la réalisation de ce projet nous a été bénéfique parce qu’elle nous a permis d’approfondir nos connaissances théoriques acquises pendant notre formation à l’ECOLE SUPERIEURE DES SCIENCES ET DE LA TECHNOLOGIE DE HAMMAM SOUSSE et d’enrichir nos connaissances pratiques en terme de :

* Collecte et analyse de données et des informations du client et élaborer un cahier de charge reflétant ses attentes et ses besoins.
* Spécification fonctionnelle et non fonctionnelle de l’application et la conception UML des différents diagrammes.
* Spécification fonctionnelle et non fonctionnelle de l’application et la conception UML des différents diagrammes.
* Choix des technologies et langage de développement à partir d’une grande variété selon les besoins à fin d’implémenter une solution de qualité et de robustesse considérées.

Sur le plan personnel, ce projet nous a permis de renforcer notre créativité, notre d´détermination et nos capacités d’échange et de travail en groupe. Tout au long du mémoire, nous avons pu décrire notre projet à travers les différents chapitres ; en fait, nous avons commence par une étude préalable mettant le projet dans son cadre, ensuite nous avons spécifié et analysé les besoins que l’application doit répondre et les schématisé dans les chapitres suivants la conception et la réalisation des interfaces web issus de la spécification.

Nous espérons avoir pu réunir, dans ce modeste rapport, tous nos savoirs et savoir-faire, résumant notre stage qui s’est passé dans de bonnes conditions, et nous avons touché l’ordre et la ponctualité de tous les personnels de la société. Enfin, nous sommes très ravis d’avoir eu l’occasion d’une telle expérience.

# BIBLIOGRAPHIE

[1] [Accès le 20/3/2021], ***Adresse URL :*** [***https://www.sonovente.com***](https://www.sonovente.com)

[2] [Accès le 20/3/2021], ***Adresse URL :*** [***http://www.alibabamusic.com***](http://www.alibabamusic.com)

[3] **Qu’est-ce que le cycle en V ?,** [Accès le 20/3/2021], ***Adresse URL :*** [**https://www.proinfluent.com/cycle-en-v**](https://www.proinfluent.com/cycle-en-v)

## [4] Qu’est-ce que la planification de projet ?,[Accès le 10/3/2021], *Adresse URL :* [*https://www.kiwili.com/Blog/post/planification-projet-definition-techniques-outils/*](https://www.kiwili.com/Blog/post/planification-projet-definition-techniques-outils/)

[5] **Comment définir les besoins fonctionnels en gestion de projet ?,**[Accès le 20/4/2021], ***Adresse URL :*** [***https://www.advaloris.ch/nos-services/gestion-de-projet/definir-besoins-fonctionnels-gestion-de-projet***](https://www.advaloris.ch/nos-services/gestion-de-projet/definir-besoins-fonctionnels-gestion-de-projet)

[6] **Ali Dagdoug**,Application Web pour gérer Un Club Sportif de ISIM, rapport PFE (2019/2020)

[7] **UML : un langage de modélisation de type graphique** (2018), [Accès le 9/7/2021],

***Adresse URL :*** [**https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-programmation-orientee-objet/**](https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-programmation-orientee-objet/)

[8] **Diagrammes de cas d'utilisation**, [Accès le 9/7/2021],

***Adresse URL :*** [***https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case***](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case)

[9] **Définition des diagrammes de séquence UML 1.5**, [Accès le 11/7/2021],***Adresse URL :*** [**http://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition\_des\_diagrammes\_de\_s%C3%A9quence\_UML\_1.5**](http://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_s%C3%A9quence_UML_1.5)

[10] **Qu'est-ce qu'un diagramme de classes UML ?**, [Accès le 11/7/2021], ***Adresse URL :***  [**https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml**](https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml)

[11] [Accès le [11/7/2021], ***Adresse URL :*** [**https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites**](https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites)

[12] **C’est quoi un Angular ?**,[15/07/2021] ,**Adresse URL** : https://monpetitdev.fr/cest-

quoi-angular-definition/