

**UNIVERSITE JOSEPH KI-ZERBO
(UJKZ)**

INSTITUT BURKINABE DES ARTS ET METIERS



RAPPORT DE STAGE

Pour l'obtention de la licence professionnelle

Option : **Méthodes Informatiques Appliquées à la Gestion (MIAGE)**

Période :

Du 01^{er} Juillet au 30 Septembre 2019

THEME :

**CONCEPTION D'UNE PLATEFORME E-COMMERCE
POUR L'ENTREPRISE AMITECH**

Présenté par : Wendinmi Rodrigue OUEDRAOGO

Maitre de stage :

M. Adadé KPODAR

Directeur Général de AMITECH

Directeur de rapport :

Dr Ferdinand GUINKO

**Enseignant-chercheur en informatique
à l'IBAM**

ANNÉE ACADÉMIQUE 2018 – 2019

SOMMAIRE

SOMMAIRE	i
DEDICACE.....	iii
REMERCIEMENTS	iv
LISTE DES SIGLES ET ABREVIATIONS	v
LISTE DES FIGURES GRAPHIQUES	vi
LISTE DES TABLEAUX.....	vii
INTRODUCTION GÉNÉRALE.....	1
CHAPITRE 1 : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL	2
I. Présentation de la structure de formation (IBAM).....	2
II. Présentation de la structure d'accueil (AMITECH).....	4
CHAPITRE 2 : ANALYSE ET CONCEPTION	7
I. Etude Préalable.....	7
II. Expression des besoins.....	13
III. Conception globale	26
IV. Réalisation.....	32
CHAPITRE 3 : BILAN DE STAGE.....	48
I. Présentation du déroulement du stage et des activités réalisées.....	48
II. Observations, suggestions et perspectives	48
CONCLUSION	51
BIBLIOGRAPHIE ET WEBOGRAPHIE	52
I. Bibliographie	52
II. Webographie	52
TABLE DES MATIERES	53

ANNEXES	I
I. ANNEXE 1 : Le langage UML.....	I
II. ANNEXE 2 : La méthode COCOMO	II
III. ANNEXE 3 : L'utilisation et le fonctionnement de Stripe	III
IV. ANNEXE 4 : Le processus de développement en X (Tactiques de progression)..	VI

DEDICACE

Au bon Dieu
Et à toute ma
famille

REMERCIEMENTS

Nous ne saurons continuer notre travail sans d'abord présenter nos sincères remerciements à tous ceux qui ont contribué de près ou de loin à l'obtention ainsi qu'au bon déroulement de ce stage.

Nos remerciements vont aussi à tous ceux qui ont contribué d'une manière ou une autre à l'élaboration de ce document. Nous adressons nos remerciements particulièrement à :

- M. Adadé KPODAR, notre maître de stage, qui n'a ménagé aucun effort, à nous soutenir malgré ses multiples occupations, à nous écouter et à nous orienter vers la bonne direction en apportant des réponses et ressources adéquates à toutes nos questions.
- Dr Ferdinand GUINKO, notre maître de suivi, qui nous a témoigné son entière disponibilité à nous assister tout au long de la rédaction de ce document.
- Dr Blaise KONE, Directeur de l'IBAM, et tout le corps professoral pour les conseils et la formation de qualité que nous avons reçue au cours de ces trois (03) dernières années.
- Tout le personnel de AMITECH pour leur disponibilité et leur collaboration.
- Tous nos amis, parents, camarades dont le soutien est inestimable.

LISTE DES SIGLES ET ABREVIATIONS

SIGLE / ABREVIATION	SIGNIFICATION
IBAM	L'Institut Burkinabè des Arts et Métiers
CU	Cas d'Utilisation
ORM	Mapping Objet-Relationnel (en anglais Object-Relational Mapping)
BD	Base de données
MIT	Massachusetts Institute of Technology
UML	Unified Modeling language
XUP	X Unified Process
IHM	Interactions Homme-Machines
CREAD	Create, Read, Update, Delete
HTTP	HyperText Transfert Protocol
MySQL	My Structural Query Language
PHP	Hypertext Processor
SQL	Structure Query Language
TDEV	Temps de DEveloppement
SGBD	Système de Gestion de Base de Données
RCCM	Registre du Commerce et du Crédit Mobilier

LISTE DES FIGURES GRAPHIQUES

Figure 1:Organigramme AMITECH	5
Figure 2:Schéma illustrant les phases du processus de développement en X	9
Figure 3:Planning prévisionnel de réalisation	12
Figure 4:Diagramme de cas d'utilisation	18
Figure 5:Diagramme de séquence « Authentification par réseau social »	20
Figure 6:Diagramme de séquence « Effectuer commandes avec paiement direct »	21
Figure 7:Diagramme de séquence « Gestion du Panier »	22
Figure 8:Diagramme d'activités « supprimer produit du panier »	23
Figure 9: Diagramme d'activités « effectuer commandes avec paiement direct »	24
Figure 10:Schéma de l'architecture client-serveur web.....	26
Figure 11: Le Diagramme de classes	29
Figure 12:Diagramme de séquence vu de l'intérieur du CU « gestion panier »	31
Figure 13:Diagramme de déploiement	32
Figure 14:La page d'accueil.....	38
Figure 15:La page d'authentification	38
Figure 16:Le catalogue des produits	39
Figure 17:La page des détails sur un produit	40
Figure 18:Le panier d'achat	41
Figure 19: Page d'administration des produits.....	42
Figure 21: Récupérer des informations bancaires en JavaScript.....	IV
Figure 22: Enregistrement des informations du token	V
Figure 23:Prélèvement de la somme à travers le token.....	V
Figure 24:Illustration de l'implémentation des webhooks.....	V
Figure 25:Niveaux de progression du processus en X	VI

LISTE DES TABLEAUX

Tableau 1:Liste des CU	15
Tableau 2:Description textuelle du cas « Authentification par réseau social »	16
Tableau 3: Description textuelle du cas d'utilisation « Gestion du Panier »	17
Tableau 4:Description textuelle du cas d'utilisation « Effectuer Commande avec paiement direct »	17
Tableau 5: Etude comparative des différentes architectures	26
Tableau 6:Dictionnaire de données	28
Tableau 7: Tableau comparatif des SGBD.....	34
Tableau 8: Etude comparative de quelques serveurs d'application	35
Tableau 9: Cout de réalisation de la plateforme.....	47
Tableau 10:Formule de calcul COCOMO	II

INTRODUCTION GÉNÉRALE

Ces dernières années, le monde a vu une avancée technologique très significative notamment dans le web. Ainsi le monde du web, très évolutif, voit plusieurs milliers de sites web se lancer chaque jour avec des innovations très variées. Parmi ses nombreuses innovations, il y a le commerce en ligne qui a vu une émergence ces dix dernières années et a fait ses preuves depuis longtemps avec en exemple les géants comme Amazone, Alibaba et beaucoup d'autres.

C'est dans ce sens que AMITECH, une société à vocation technologique, s'est donnée pour objectif d'être une référence dans le commerce en ligne de solutions électroniques dans le pays et hors de ses frontières.

Ainsi nous avons été accueillis en son sein, pour un stage de trois (3) mois afin de proposer et de concevoir une solution numérique dans l'atteinte de leurs objectifs. AMITECH a donc plusieurs attentes vis-à-vis de la future plateforme.

La mise en place d'un tel système demande un travail organisé et méthodique. De ce fait notre travail s'articulera autour de trois principaux éléments à savoir : la recherche documentaire, la réalisation (analyse et conception) et la rédaction du rapport. Ainsi dans la suite de notre travail nous présenterons tout d'abord nos structures de formation et d'accueil, puis nous montrerons les éléments constitutifs de la réalisation du système et enfin nous ferons le bilan de notre stage.

CHAPITRE 1 : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL

Dans ce chapitre, nous présenterons dans un premier temps notre structure de formation qui est l'Institut Burkinabè des Arts et Métiers (IBAM). Dans un deuxième temps, nous présenterons AMITECH, notre structure d'accueil. L'objectif étant de montrer le cadre dans lequel se déroule le projet.

I. Présentation de la structure de formation (IBAM)

L'Institut Burkinabè des Arts et Métiers (IBAM) est un établissement d'enseignement professionnel plein de ressources et de formations qualifiantes.

1) Historique

L'Institut Burkinabè des Arts et Métiers (IBAM) est un établissement d'enseignement professionnel. Il a été créé en janvier 2000 à la faveur de la refondation de l'Université de Ouagadougou rebaptisée *Université Joseph KI-ZERBO*.

2) Objectif

L'objectif principal de l'IBAM est de répondre aux besoins du marché de l'emploi en mettant à sa disposition un potentiel humain de cadres moyens et supérieurs dans les divers secteurs d'activité. Une nouvelle orientation qui se justifie par l'incapacité de l'Etat d'embaucher tous ceux qui sortent de l'université.

3) Organisation

L'organisation de l'IBAM est une structure hiérarchico-fonctionnelle. Nous avons le conseil de gestion qui est l'organe suprême qui regroupe le directeur, le directeur adjoint, les coordonnateurs, les enseignants permanents, le CSAFC, la secrétaire principale et le représentant du personnel ATOS. Cet organe est suivi du conseil scientifique qui regroupe le directeur, le directeur adjoint, les coordonnateurs et les enseignants de rang A de l'Institut. Nous avons ensuite le cabinet du directeur auquel sont rattachés directement le CSAFC, le directeur adjoint, et la secrétaire principale à laquelle est rattachée le personnel ATOS. Au directeur adjoint sont rattachés les coordonnateurs de filières et aux coordonnateurs de filières est rattaché le personnel enseignant de l'Institut.

4) Filières de formation

En vue d'atteindre les objectifs cités précédemment, l'IBAM offre des formations dans plusieurs filières. Ces filières sont réparties en 3 groupes selon le diplôme ou la période de formation.

➤ Licences professionnelles (Formation initiale)

Les filières de formations en licences professionnelles sont :

- ✓ Comptabilité-Contrôle-Audit (CCA);
- ✓ Assurance-Banque-Finance (ABF);
- ✓ Marketing et Gestion (MG);
- ✓ Assistant de Direction/Bilingue (ADB);
- ✓ Méthodes Informatiques Appliquées à la Gestion (MIAGE).

➤ MASTERS

En master, l'IBAM ouvre quatre (4) filières de formation que sont :

- ✓ Master en Administration et Gestion des Entreprises (MAGE);
- ✓ Master en Comptabilité-Contrôle-Audit (MCCA);
- ✓ Master en Banque-Finance (MBF);
- ✓ Master en informatique (option Ingénierie des Systèmes Informatiques, et option Sécurité Informatique).

La filière MIAGE a été créée dans l'optique de répondre aux besoins croissants des entreprises en cadres compétents dans le domaine de la technologie et des techniques informatiques. Cette filière accueille en première année les bacheliers des séries C, D et E ayant passé avec succès le test de sélection ou ayant été acceptés en tant qu'auditeurs libres.

Les étudiants en fin de cycle en MIAGE doivent mettre en pratique les connaissances acquises en classe en effectuant un stage d'une durée d'au moins trois (03) mois suivi d'une soutenance publique. Ce stage a pour objectif de leur permettre de se familiariser avec le monde professionnel. C'est dans ce cadre que nous avons été accueillis au sein de l'entreprise AMITECH.

II. Présentation de la structure d'accueil (AMITECH)

Notre structure d'accueil, AMITECH, est une société avec assez d'expériences dans l'électronique et le numérique avec pour objectifs d'apporter des solutions technologiques au continent africain.

1) Historique

Située au 144 rue 30.256 du secteur 51, arrondissement 11 de Ouagadougou, AMITECH est une société unipersonnelle à responsabilité limitée créée le 1er janvier 2012. A l'origine, en octobre 1997, c'était un atelier de maintenance et de réparation des ordinateurs et périphériques sous le nom « A.M.I.E Technique » (Atelier de Maintenance Informatique et Electronique) ouvert par M. Adadé KPODAR. On y réparait toutes sortes d'appareils informatiques et électroniques.

Le 02 janvier 1998, l'atelier a changé de nom et devient l'entreprise AMITECH pour se faire enregistrer officiellement sous le régime fiscal CSI (Contribution du Secteur Informel). Elle a obtenu par la suite un RCCM (N° BFOUA2012B130). AMITECH a commencé à diversifier ses activités. En plus de la maintenance informatique et électronique, elle a ouvert une boutique dans laquelle elle vendait du matériel informatique ainsi que des consommables. Une salle de formation a été ouverte par la suite proposant des formations à la maintenance et réparation des ordinateurs et périphériques, à l'utilisation des logiciels et des systèmes d'exploitation. Le centre de formation offrait également des services comme le traitement de texte, le scannage, la gravure de CD, les photocopies de documents.

En 2003, AMITECH s'est associée à une entreprise américaine basée à New York pour la commercialisation de matériels de lutte contre la fraude. Pour cette raison, AMITECH a changé de régime fiscal et est passé au RSI (Régime Simplifié d'Imposition). A partir de ce moment, l'activité principale de AMITECH est devenue la vente de produits de lutte contre la fraude tels que : les détecteurs de faux billets, les machines à compter les pièces de monnaie et billets, les destructeurs de documents, les appareils de marquage de documents, les machines de façonnage de documents. Les activités menées avant 2003 sont devenues secondaires.

AMITECH a connu une nette croissance de son chiffre d'affaires de 2004 jusqu'en 2008. A compter du 01^{er} janvier 2009, AMITECH a fait l'objet d'un reclassement en RNI (Régime Normal d'Imposition). C'est ainsi qu'à partir de 1er janvier 2012, le fondateur a décidé de le transformer en AMITECH SARL.

Les activités de la société AMITECH sont les ventes d'appareils électroniques et la personnalisation de produits électroniques entrant dans : les solutions de gestion de file d'attente, les solutions de traitement d'espèces (comptage, détection, conditionnement et sécurisation) et de courriers, les solutions de sécurisation (contrôle d'accès, coffre-fort ...), la domotique, la conception de solutions électroniques et informatiques (services récents) sur mesure, la maintenance informatique et électronique.

2) Objectifs (Attributions)

Aujourd'hui la société AMITECH évolue sur un marché fortement concurrentiel. Ses objectifs actuels sont : le maintien de sa croissance et surtout devenir leader d'un marché marqué par une forte concurrence avec l'arrivée chaque année de nouvelles sociétés concurrentes. C'est ainsi que par la création du site Internet et la plateforme de ventes de nos produits en ligne elle atteindra ces objectifs.

3) Organisation

Comme toute société, AMITECH a une organisation bien définie, hiérarchisée et composée de plusieurs services. En effet au sommet de cette hiérarchie se trouve le Directeur Général avec à ses côtés un secrétariat. Ensuite il y a au même niveau de la hiérarchie le service commercial, le service après-vente, et le service de la comptabilité. Le service commercial, avec une boutique en son sein, assure le stockage, les commandes et/ou les livraisons des produits et solutions commandées (c'est en son sein que s'est passé notre travail sans limiter notre collaboration avec tous les autres services). Le service de la comptabilité se charge des affaires comptables de la société et le service après-vente assure le suivi après-vente auprès des clients avec la maintenance des solutions.

L'organigramme de la structure est présenté comme suit :

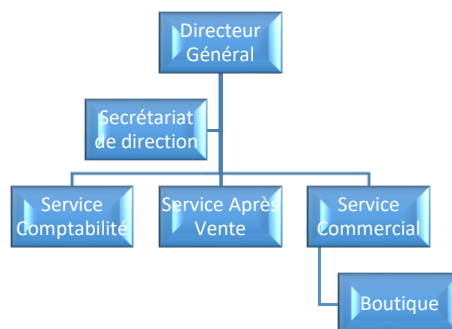


Figure 1: Organigramme AMITECH (AMITECH. (2019). Présentation de la société AMITECH. Ouagadougou. 2 p.)

En somme nous avons présenté en premier lieu notre structure de formation. Et en deuxième lieu nous avons fait une présentation de notre structure d'accueil et mis en lumière ses champs d'action et objectifs qui s'inscrivent dans le cadre de notre stage.

CHAPITRE 2 : ANALYSE ET CONCEPTION

Après avoir présenté les structures de formation et d'accueil, nous allons dans ce chapitre traiter le thème soumis à notre analyse qui est « **CONCEPTION D'UNE PLATEFORME E-COMMERCE POUR L'ENTREPRISE AMITECH** ». Pour ce faire nous allons dans un premier volet présenter notre thème, la méthode d'analyse et conception de notre projet, les acteurs impliqués ainsi que le planning de notre étude. Dans un deuxième volet, suivra l'expression des besoins. Dans un troisième volet, une étude technique et une conception détaillée du futur système seront présentées. Enfin, nous présenterons les différents éléments entrant dans la réalisation du projet.

I. Etude Préalable

Dans cette première partie, nous essayerons de présenter les différentes facettes autour du thème de notre travail et les éléments d'analyse et de conception de notre système.

1) Présentation du Thème

Le thème de notre travail s'inscrit dans un contexte bien précis. En effet celui-ci se veut apporter une solution bien précise.

a) Contexte

Depuis sa création AMITECH, s'est donné l'objectif de fournir aux entreprises et grandes structures du matériel informatique et ce qui a favorisé l'ouverture d'une boutique. Mais avec l'émergence des nouvelles technologies, la société entend élargir sa clientèle dans le pays et hors de ses frontières. De ce fait elle s'est donné pour objectif de se rendre visible sur le web par la réalisation d'une plateforme web de leur boutique.

b) Problématique

Le service commercial de AMITECH acquiert difficilement de nouveaux clients actuellement alors que le nombre de clients potentiels ne cesse de croître. Aussi il y a un manque de visibilité sur les produits. Egalement, les renseignements sur les différents produits se font par appel téléphonique ou par contact direct avec le client ce qui peut rendre un peu lente la phase de renseignement sur les produits. Aussi AMITECH a pour ambition de diversifier ses produits de ventes.

De ce fait, elle a mis dans son programme, la réalisation d'une plateforme web pour la commercialisation de ses produits. Ceci, dans le but d'avoir plus de visibilité, rendre ces produits facilement accessibles à distance.

C'est pour pallier ces problèmes cités que nous avons été accueillis au sein de AMITECH afin de proposer une plateforme web pour la vente de ses produits. Travail que nous réaliserons, et en même temps dans le cadre de notre stage de fin d'étude, sous le thème « **CONCEPTION D'UNE PLATEFORME E-COMMERCE POUR L'ENTREPRISE AMITECH** ». Ce système devra rendre facile la consultation des produits par les clients, l'achat et l'administration facile des produits.

c) Objectifs et attentes

Au regard des besoins exprimés par AMITECH et du but de notre stage, nous avons pour objectifs :

- de fournir une plateforme fonctionnelle
- renforcer nos compétences en génie logiciel
- de produire un rapport de stage

Et les attentes de la structure sont :

- avoir un logiciel fonctionnel ;
- avoir plus de visibilité sur le web
- augmenter les ventes

2) Méthode d'analyse et de conception

La réalisation d'une application web doit impérativement être précédée d'une phase d'analyse et de conception. Ceci a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un site afin de rendre ce développement plus fidèle aux besoins du client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités et la phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Pour mener à bien notre étude nous avons choisi d'utiliser le processus de développement XUP. Et ce processus s'appuie sur UML pour la modélisation des différents diagrammes.

a) Processus de développement (XUP)

Pour la réalisation de notre projet nous avons choisi d'utiliser le processus en X ou XUP « X Unified Process » (Figure 2). Ce choix est justifié par le fait qu'il définit une bonne politique d'itération pilotée par les risques et les priorités du client, ainsi qu'une bonne tactique de progression définissant les objectifs à atteindre à chaque phase.

En effet c'est un processus qui répond bien aux caractéristiques d'un processus UP et il est basé sur une approche disciplinée focalisée sur l'Extreme Programming afin de bien maîtriser, tout au long du cycle de développement du logiciel, l'assignation des tâches et la responsabilisation des différents acteurs participants. Le « X » signifie littéralement que le processus suit deux chemins en haut et deux chemins en bas ce qui donne une forme en X. Les deux chemins du haut sont utilisés pour la spécification parallèle des besoins fonctionnels et des contraintes de sécurité, et les deux chemins du bas correspondent aux deux axes des changements imposés au système d'information : il s'agit de la vue logique ou de conception qui décrit les aspects statiques et dynamiques du système en termes de classes et d'objets et la vue technique qui se préoccupe de la spécification de l'architecture technique du système. Le processus est détaillé en ANNEXE 4 : Le processus de développement en X (Tactiques de progression).

La figure suivante présente le schéma général du processus en X :

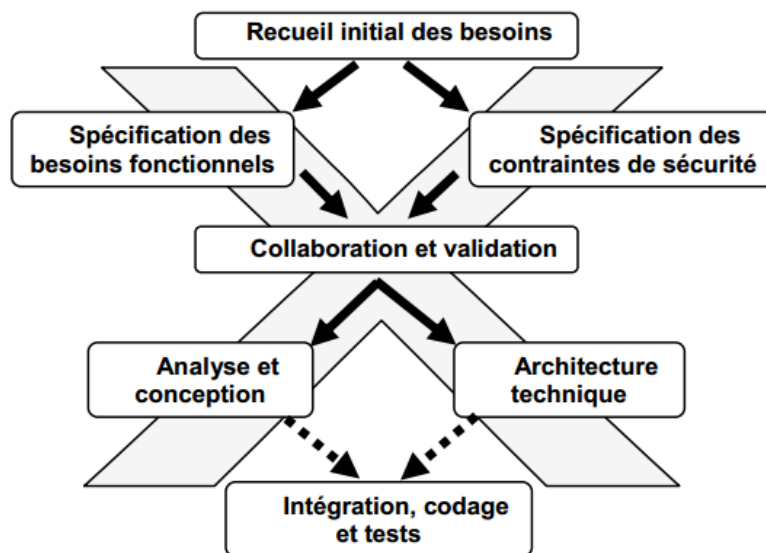


Figure 2: Schéma illustrant les phases du processus de développement en X (Salim CHEHIDA. Proposition d'un processus de développement pour la modélisation sécurisée des systèmes d'information [en ligne]. (Page consultée le 15/09/2019). <https://pdfs.semanticscholar.org/3828/cd13e0866dfd0b2193035cc30704c164a25a.pdf>)

Ce processus de développement s'appuie également sur UML pour la modélisation des différents artefacts (diagrammes).

b) Langage de modélisation (UML)

UML est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en œuvre d'un projet. C'est avant tout un formalisme graphique issu de notations employées dans différentes méthodes objets.

En l'occurrence, UML permet d'exprimer, à l'aide de diagrammes (représentation visuelle simplifiée et structurée), des entités et des actions.

UML a donc pour vocation de nous aider à formaliser des situations, dans le contexte des systèmes d'information (SI). Les détails sur UML vous sont présentés en ANNEXE 1 : Le langage UML.

La palette d'éléments à modéliser est donc assez large, du simple concept (ex. : une voiture), aux relations entre les concepts (ex. : une voiture est un véhicule à moteur qui a quatre roues), en passant par les actions pouvant être réalisées par des acteurs (ex. : un chauffeur peut conduire une voiture à condition d'être titulaire du permis de conduire.), ou par les processus (ex. : pour se rendre d'un point A à un point B, un chauffeur doit d'abord ouvrir la portière de sa voiture, y monter, refermer la portière, mettre sa ceinture, mettre le contact, desserrer le frein à main, etc.).

Pour réaliser donc la modélisation de notre système, nous allons utiliser ce langage pour plusieurs raisons. Nous pouvons citer entre autres qu'il :

- présente l'avantage d'être le standard en termes de modélisation objet universellement reconnu ;
- est un langage visuel car sa notation graphique permet d'exprimer visuellement des solutions objets facilitant ainsi la comparaison et l'évaluation de celles-ci ;
- est un langage formel et normalisé doté d'un gain de précision et d'un gain de stabilité ;
- sert à formaliser tous les documents techniques d'un projet et permet d'affiner les détails de l'analyse au fur et à mesure de l'avancée du projet ;
- est capable d'utiliser le même atelier de génie logiciel, depuis l'expression des besoins des utilisateurs jusqu'à la génération de tout ou partie du code ;
- est un support de communication performant car il cadre l'analyse tout en facilitant la compréhension des représentations abstraites complexes.

3) Groupe de travail

Le groupe de travail représente l'ensemble des personnes nécessaires pour la réalisation du projet. On les classe en trois (03) groupes distincts travaillant en synergie. Il s'agit du groupe de pilotage, du groupe de projet et du groupe des utilisateurs.

a) **Groupe de pilotage**

Ce groupe a pour rôle la prise des décisions nécessaires afin que le projet puisse être mené à bien. Les tâches qui lui ont été assignées sont :

- Planifier les dates clés du projet ;
- Analyser les propositions du groupe de projet ;
- Décider des orientations stratégiques.

Ce groupe est constitué de :

- M. Adadé KPODAR, Directeur Général de AMITECH, notre maître de stage ;
- M. Pacôme KPODAR, ingénieur en informatique industrielle et maintenance.
- M. Yannick BATIONO, ingénieur en informatique industrielle et maintenance.
- M. Rodrigue Wendinmi OUEDRAOGO, étudiant en troisième (3ème) année en MIAGE et stagiaire.

b) **Groupe de projet**

Le groupe de projet est l'ensemble des personnes chargées de réaliser le projet. Il est l'intermédiaire entre le groupe de pilotage et le groupe des utilisateurs. Ce groupe est composé principalement de OUEDRAOGO Wendinmi Rodrigue, étudiant en troisième (3ème) année en MIAGE à l'IBAM.

c) **Groupe des utilisateurs**

Ce groupe est constitué de l'ensemble de tous ceux qui vont utiliser le futur système. Il participe à la capture des besoins fonctionnels. Il est composé de notre public cible représente le peuple burkinabè en particulier et de toute personne, internaute, société ou organisation qui accède à notre système de partout dans le monde.

4) Planning de réalisation

Pour mener à bien notre étude et la réaliser dans les délais, nous avons subdivisé le projet en tâches. Pour cela nous avons utilisé le diagramme de Gantt qui est un outil utilisé en ordonnancement et en gestion de projet ; il permet de visualiser (à l'aide d'un graphe) dans le temps les diverses tâches liées, composant un projet. Le planning prévisionnel (Figure 3) est résumé dans le tableau ci-dessous.

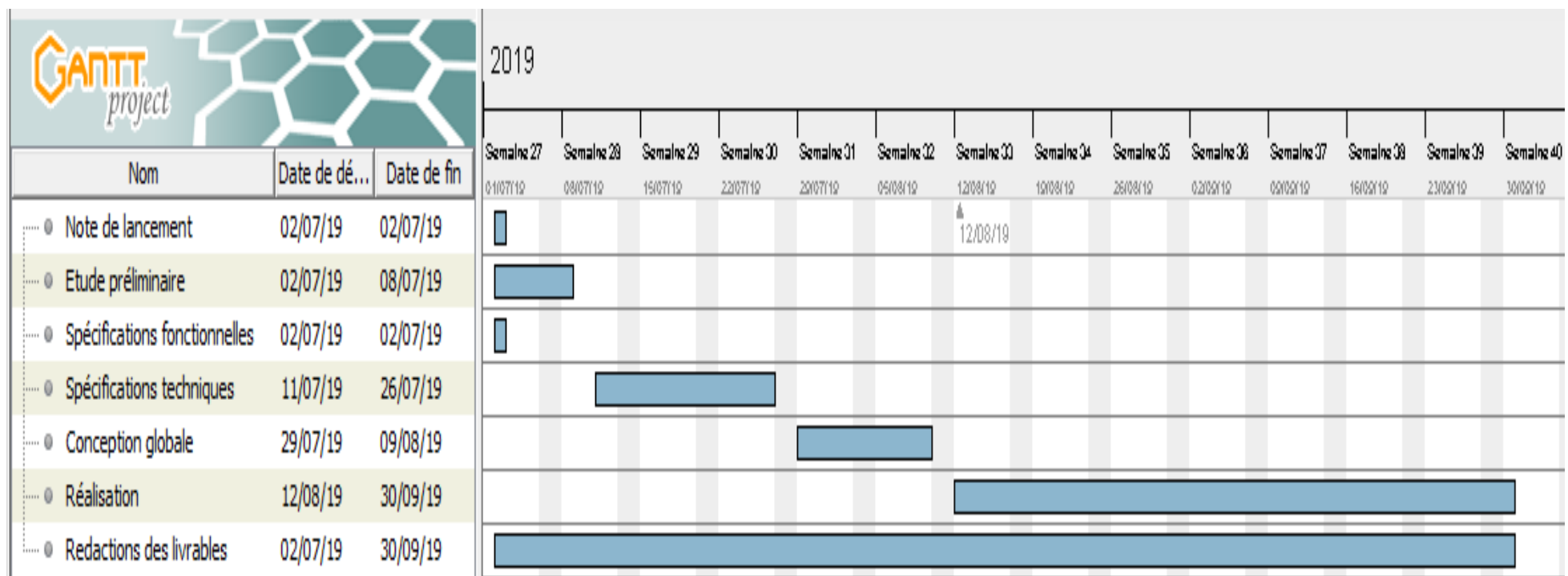


Figure 3: Planning prévisionnel de réalisation

II. Expression des besoins

Nous mettrons en exergue, dans cette partie, les différentes fonctions du système et les choix techniques opérés en vue de sa réalisation.

1) Spécification fonctionnelle

La spécification fonctionnelle est la description détaillée des fonctions d'un logiciel en vue de sa réalisation. Elle décrit la façon dont les exigences seront prises en compte dans la conception. Nous commencerons par présenter les différentes fonctionnalités de l'application à travers les acteurs du système et les cas d'utilisation.

a) **Identification des acteurs et cas d'utilisation**

➤ Les acteurs

Un acteur définit un ensemble cohérent de rôles qu'un utilisateur ou une entité externe peut jouer en interagissant avec le système. Un acteur peut consulter ou modifier directement l'état du système en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

A la suite de la description des tâches, nous distinguons principalement les acteurs suivants :

- ✓ Le visiteur : il est vu comme la personne qui visite le site. C'est celui-là même qui accède à la plateforme pour explorer les offres en vue de s'inscrire si besoin est.
- ✓ Le membre: il est vu comme l'utilisateur inscrit sur la plateforme. C'est celui-là même qui accède à la plateforme en vue de faire d'éventuels achats ou commandes après s'être connecté.
- ✓ L'administrateur : c'est la personne chargée d'administrer les produits à vendre sur la plateforme et gère les commandes. Il s'occupe principalement de la gestion des produits mis en vente, et de la gestion des commandes.
- ✓ Le super administrateur : c'est la personne qui peut administrer tous les aspects et contenus de la plateforme.
- ✓ Stripe : stripe est un acteur secondaire sollicité par le système lorsqu'un client commande un produit. Il s'agit d'un service de paiement en ligne qui permet de payer en ligne, de recevoir des paiements, ou d'envoyer et de recevoir de l'argent.
- ✓ Réseau social (Google/Facebook dans notre plateforme) : c'est un acteur secondaire sollicité par le système lorsqu'un client veut s'authentifier avec son compte de réseau social. Il vérifie si le client est bien inscrit sur sa plateforme et renvoie la réponse à notre système.

Ainsi donc pour la plateforme AmitechStore, nous avons identifié beaucoup de cas d'utilisation.

➤ La liste des cas d'utilisation :

Les cas d'utilisations (CU) permettent de bien structurer les besoins des utilisateurs d'un système.

N°	Intitulées		Description	Acteur(s)
CU01	S'authentifier	Authentification par réseau social	Permet aux utilisateurs de s'authentifier dans l'application avec principalement une adresse mail et un mot de passe.	Visiteur, Membre, Administrateur, Super administrateur
		Authentification standard	Permet aux utilisateurs de s'authentifier sur la plateforme avec leur compte Google ou Facebook.	Visiteur, membre, Google, Facebook
CU02	Rechercher		Permet aux utilisateurs d'effectuer des recherches de produits.	Visiteur, Membre, Administrateur, Super administrateur
CU03	Gérer compte		Permet de gérer son compte c'est-à-dire de modifier son compte ou de supprimer son compte.	Membre, Super administrateur
CU04	Gestion du panier	Ajouter au panier	L'utilisateur a la possibilité d'ajouter ou de supprimer un produit dans son panier.	Membre
		Supprimer du panier		
CU05	Effectuer commande	Effectuer commande avec paiement direct	L'utilisateur a la possibilité d'effectuer une commande avec paiement via carte VISA ou MasterCard.	Membre, Stripe
		Effectuer commande sans paiement direct	L'utilisateur a la possibilité d'exécuter ou d'annuler une commande sans paiement direct mais	Membre, Administrateur, Super administrateur

N°	Intitulées	Description	Acteur(s)
		avec discussion préalable avec la boutique.	
CU06	Gestion des produits	L'utilisateur a les droits d'ajout, de lecture, de suppression, de modification sur tous les produits.	Administrateur, Super administrateur
CU07	Gestion des commandes	L'utilisateur a les droits de lecture, de modification sur toutes les commandes.	Administrateur, Super administrateur
CU08	Gestion des catégories	L'utilisateur a les droits d'ajout, de lecture, de modification sur toutes les catégories.	Administrateur, Super administrateur
CU09	Gestion des utilisateurs	L'utilisateur a les droits d'ajout, de lecture, de suppression, de modification sur tous les utilisateurs de la plateforme sauf sur les super administrateurs.	Super administrateur
CU10	Gestion des rôles	Permet à l'utilisateur de gérer les droits d'accès et privilèges sur la plateforme	Super administrateur
CU11	Consultation des paiements	Permet à l'utilisateur de consulter les paiements effectués sur le site.	
CU12	Impression de factures	L'utilisateur reçoit ou imprime sa facture en fichier PDF après commande.	Membre, Administrateur, Super administrateur

Tableau 1: Liste des CU

b) Description de quelques cas d'utilisation

Dans ce qui suit, nous allons décrire quelques cas d'utilisation.

Résumé	Permet aux utilisateurs de s'authentifier sur la plateforme avec son compte Google ou Facebook.
Acteur(s)	Visiteur, Membre
Précondition(s)	Néant
Scénario nominal	<ol style="list-style-type: none"> 1. Un utilisateur demande à se connecter sur la plateforme 2. Le système affiche le formulaire de connexion avec des boutons de redirection vers le réseau social choisi (Google ou Facebook). 3. La plateforme du réseau social choisi ouvre une session afin que l'utilisateur s'authentifie dans sa plateforme et renvoie une réponse à notre plateforme pour confirmer l'existence du compte. 4. Le système confirme l'existence du compte dans notre plateforme puis ouvre la session. [A1] [A2] 5. La plateforme affiche la page d'accueil avec l'utilisateur connecté
Scénario Alternatif	<p>[A1] : Si l'utilisateur n'est pas déjà inscrit dans la plateforme alors le système l'inscrit puis ouvre la session.</p> <p>[A2] : Si le réseau social renvoie comme réponse l'inexistence du compte parmi ses membres alors le scénario reprend au point 2</p>
Post Condition	L'utilisateur accède à son compte

Tableau 2: Description textuelle du cas « Authentification par réseau social »

Résumé	L'utilisateur a la possibilité d'ajouter ou de supprimer un produit dans son panier .
Acteur(s)	Membre
Précondition(s)	<ul style="list-style-type: none"> ➤ L'utilisateur doit s'authentifier ➤ L'utilisateur doit accéder à son panier s'il s'agit d'une suppression
Scénario nominal	<p>A- Ajout de produit :</p> <ol style="list-style-type: none"> 1. L'utilisateur affiche son produit 2. L'utilisateur demande à ajouter un produit 3. Le système ajoute le produit au panier <p>B- Suppression de produit du panier</p> <ol style="list-style-type: none"> 1. L'utilisateur demande suppression d'un produit du panier, 2. Le système demande une confirmation, 3. L'utilisateur confirme alors la suppression [A1] 4. Le système supprime le produit du panier

Résumé	L'utilisateur a la possibilité d'ajouter ou de supprimer un produit dans son panier .
Scénario Alternatif	[A1] : Le système annule la suppression et garde l'utilisateur sur la page du panier

Tableau 3: Description textuelle du cas d'utilisation « Gestion du Panier »

Résumé	Le membre a la possibilité d'effectuer une commande avec paiement direct par carte bancaire (« VISA », « MasterCard »)
Acteur(s)	Membre
Précondition(s)	L'utilisateur doit être authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à son panier (Contenant les produits choisis) 2. L'utilisateur choisit le mode de paiement « Commande et paiement direct » 3. La plateforme affiche le formulaire de renseignement des coordonnées de la carte bancaire (Nom sur la carte, Numéro de carte, Code CVC (Card Verification Code), le mois d'expiration, l'année d'expiration). 4. L'utilisateur saisit les informations puis valide ces informations qui sont envoyées à stripe 5. Stripe renvoie un token représentant la carte de paiement au système. [A1] 6. Le système renvoie le token à stripe avec le montant à prélever 7. L'utilisateur confirme le paiement 8. Stripe effectue le prélèvement de sa carte 9. Le système renvoie une requête d'enregistrement vers la base de données [A2] 10. La requête est enregistrée et une alerte de succès est envoyée au système 11. Le système envoie une notification d'enregistrement à l'utilisateur par email
Scénario alternatif	<p>[A1] : en cas d'erreur stripe renvoie la réponse à la plateforme qui affiche un message d'erreur et le scénario reprend au point 4.</p> <p>[A2] : en cas d'échec de la transaction (solde insuffisant) stripe envoie une notification d'erreur à l'utilisateur</p>

Tableau 4:Description textuelle du cas d'utilisation « Effectuer Commande avec paiement direct »

c) Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation (Figure 4) fournit une représentation graphique des exigences du système, et aide à identifier la façon dont les utilisateurs interagissent avec ce dernier. Dans le contexte de notre application, il est représenté comme suit :

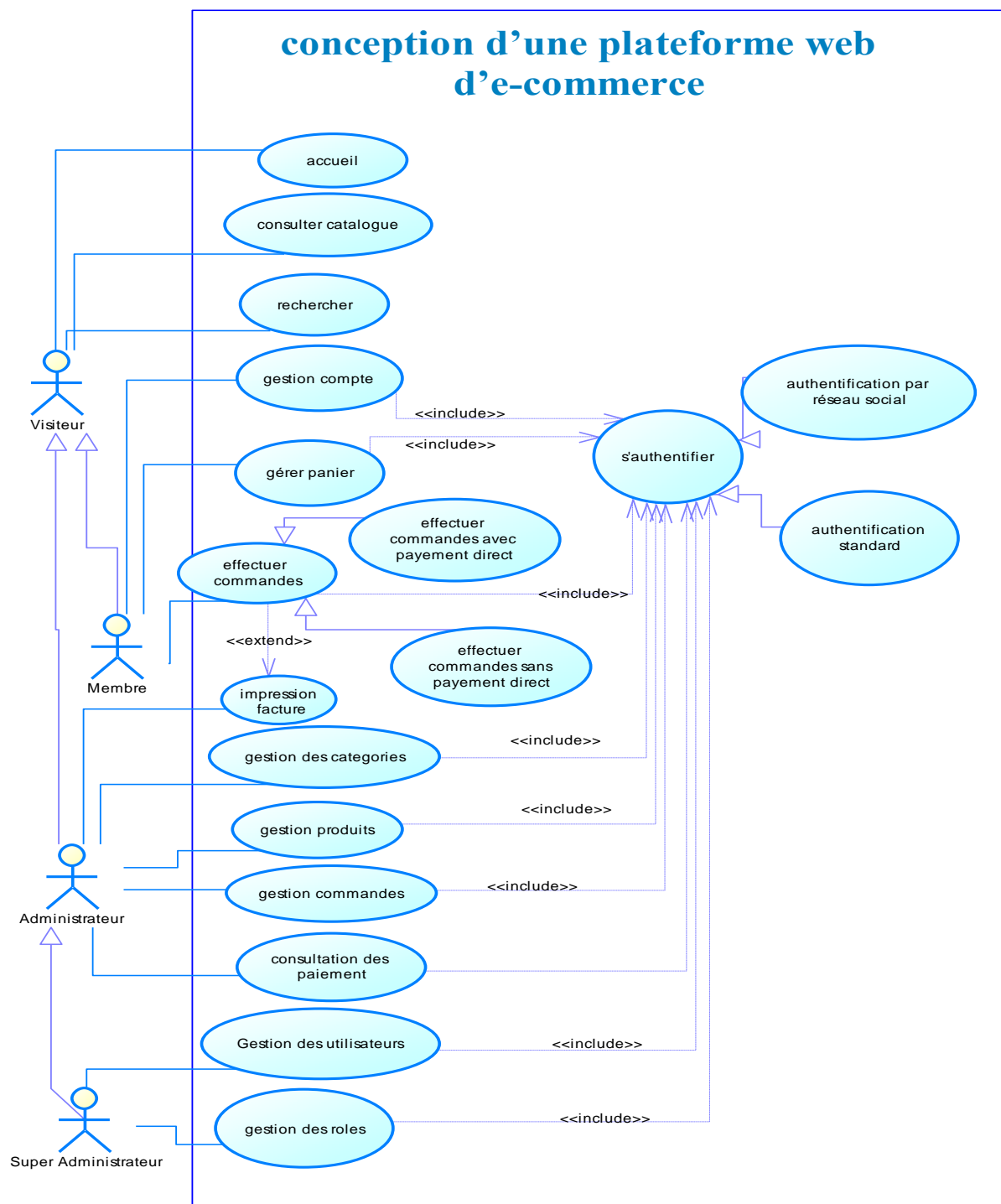


Figure 4: Diagramme de cas d'utilisation

Sur ce diagramme nous avons une illustration du fait que la plateforme permette d'effectuer des commandes avec paiement direct (composant graphique étiqueté « effectuer commande avec paiement direct »). Et cela, seulement les membres (objet étiqueté « membre » relié au composant graphique étiqueté « effectuer commande » qui est relié par une flèche non discontinu à l'objet graphique « effectuer commande avec paiement direct ») peuvent le faire mais en étant authentifié (l'objet graphique « s'authentifier » relié par une flèche discontinue).

d) Diagrammes de séquences vues de l'extérieur

Les diagrammes de séquences représentent les interactions entre le système et les utilisateurs en montrant sous forme de scénarios la chronologie des envois de messages issus d'un cas d'utilisation donné. Les diagrammes de séquences qui sont décrits dans cette partie sont essentiellement : « authentification par réseau social »(Figure 5), « effectuer commandes avec paiement direct »(Figure 6), et « gérer panier »(Figure 7).

- Diagramme de séquence « authentification par réseau social »

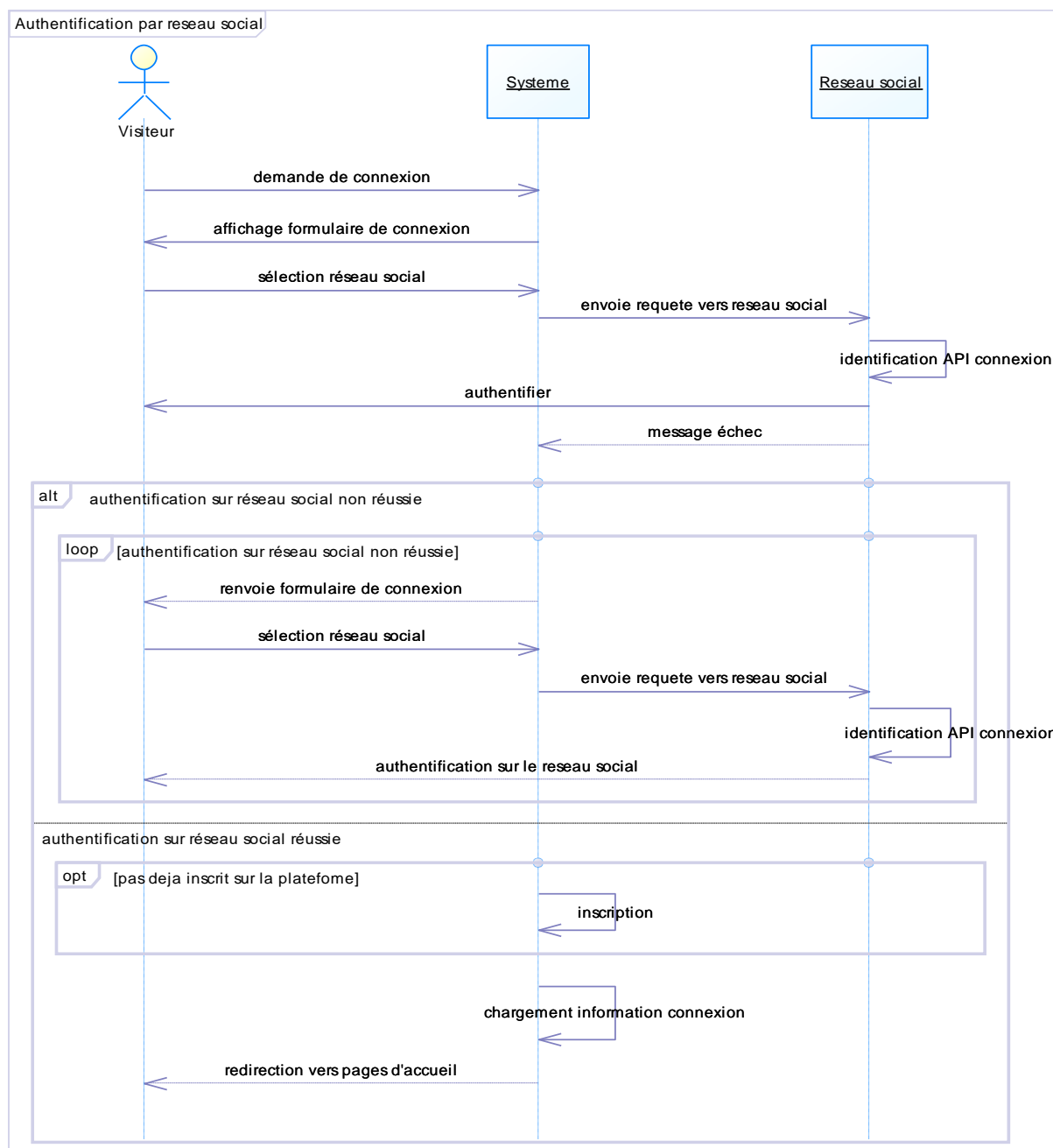


Figure 5: Diagramme de séquence « Authentification par réseau social »

Sur ce diagramme (Figure 5) Un acteur (objet étiqueté « visiteur » par exemple) demande à se connecter sur la plateforme (actions sont représentées par une flèche étiquetée « demande de connexion»), le système affiche le formulaire de connexion avec des boutons de redirection vers le réseau social choisi. Et ainsi les actions se succèdent et tout le scénario décrit dans Tableau 2 est réalisé.

- Diagramme de séquence « effectuer commandes avec paiement direct »

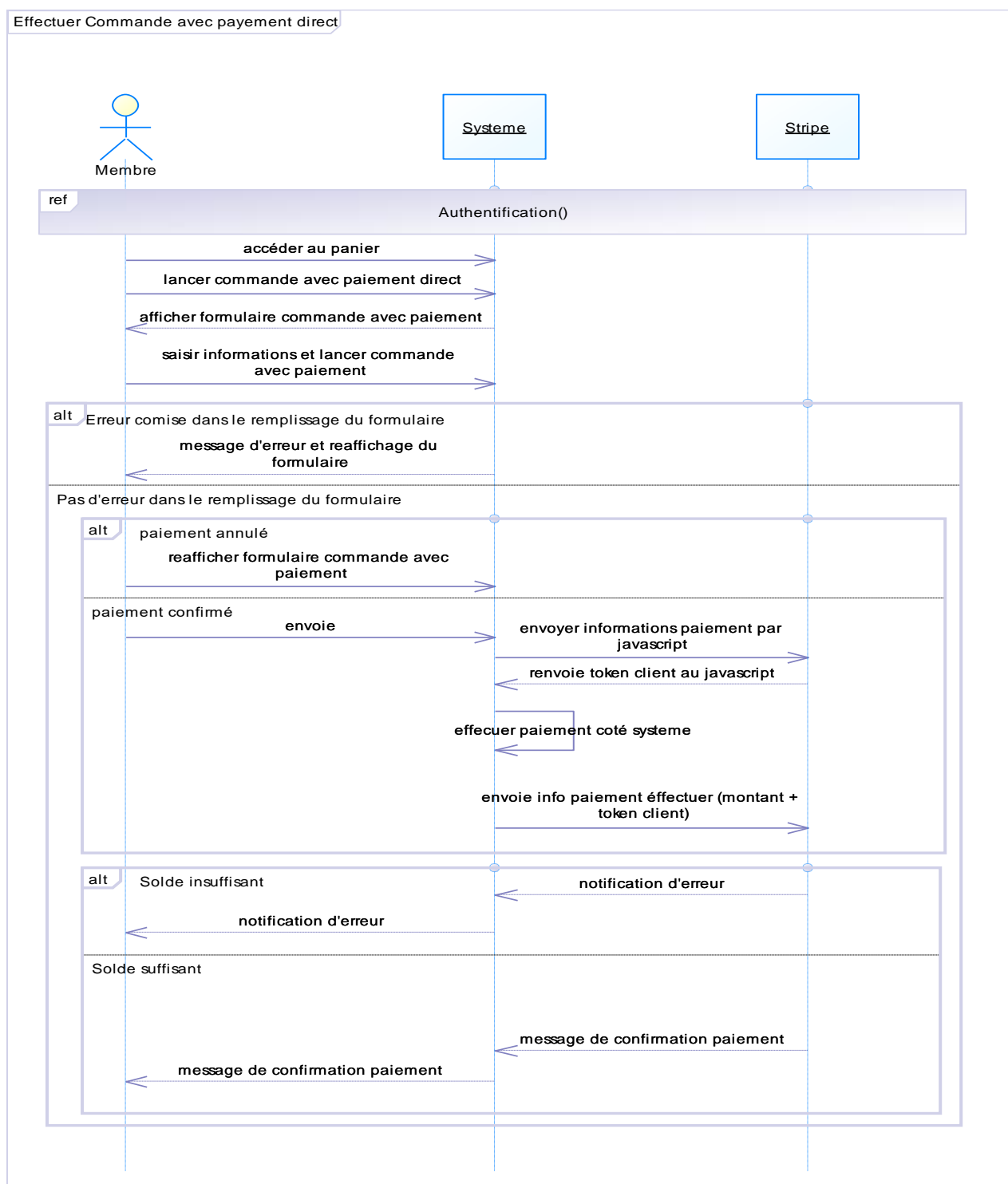


Figure 6: Diagramme de séquence « Effectuer commandes avec paiement direct »

Sur ce diagramme (Figure 6) un membre (objet étiqueté « membre ») accède à son panier (actions représentées par la flèche étiqueter « accéder au panier »). Et ainsi les actions se suivent et tout le scénario décrit dans le Tableau 4 est réalisé.

- Diagramme de séquence « gestion du panier »

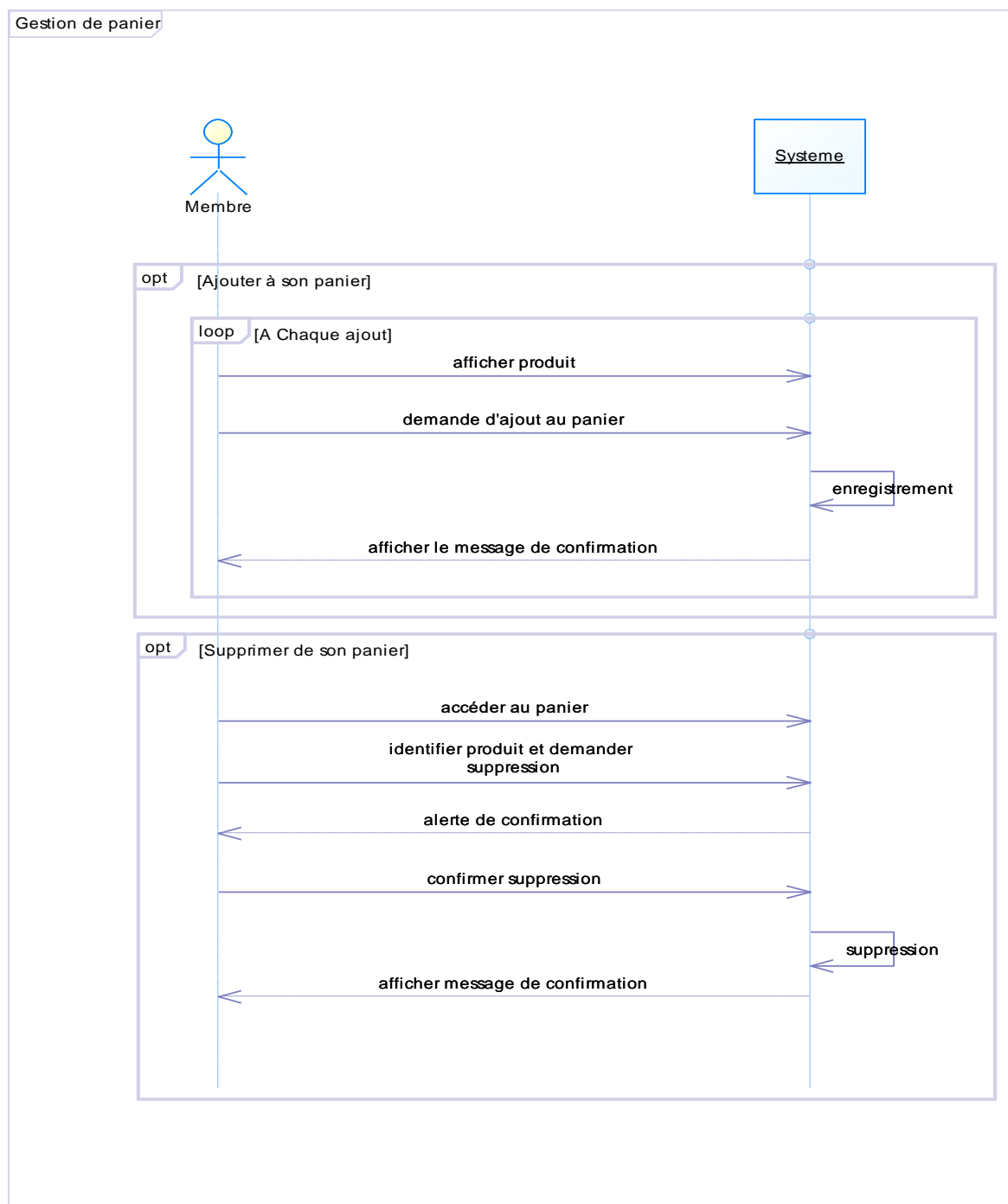


Figure 7: Diagramme de séquence « Gestion du Panier »

Sur ce diagramme (Figure 7) un membre (objet étiqueté « membre ») choisit de faire un ajout par exemple puis demande au système d'afficher le produit (actions représentées par la flèche étiquetée « afficher produit » dans le cadre étiqueté « Ajouter à son panier »). Et ainsi les actions se succèdent et tout le scénario décrit dans le Tableau 3 est réalisé.

e) Diagrammes d'activités

Le diagramme d'activités permet de faire une représentation graphique du déroulement d'un cas d'utilisation. Un diagramme d'activités est donc un bon complément à la fiche descriptive d'un cas d'utilisation complexe. Si un cas d'utilisation contient de nombreux scénarios, le diagramme d'activité permet de donner une vision globale de l'ensemble des scénarios possibles. Les cas d'utilisation qui sont mis en exergue ici sont « effectuer commandes avec paiement direct »(Figure 9) et « supprimer produit du panier »(Figure 8).

- Diagramme d'activités « supprimer produit du panier »

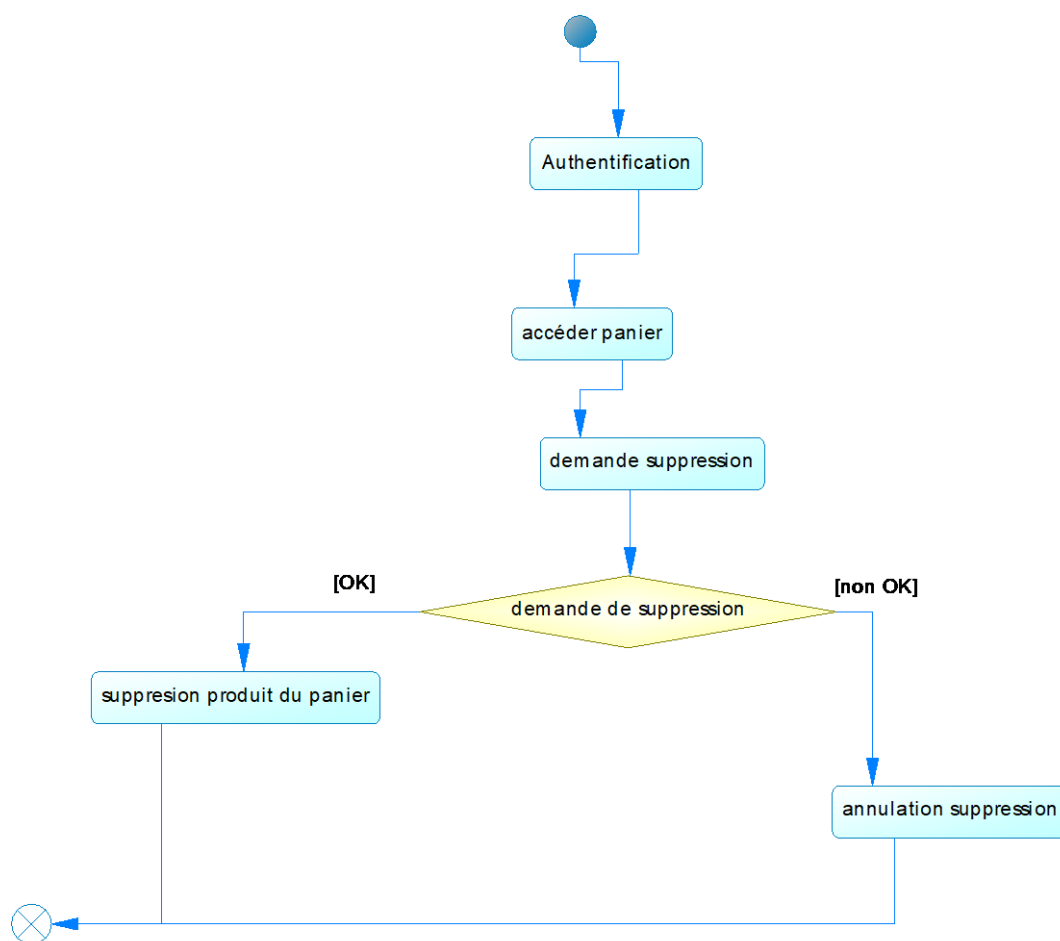


Figure 8:Diagramme d'activités « supprimer produit du panier »

Ce diagramme (Figure 8) montre que pour supprimer un produit du panier l'utilisateur s'authentifie, puis accède à son panier, ensuite demande suppression du produit, et un formulaire de confirmation est affiché et si l'utilisateur confirme le produit est supprimé sinon la suppression est annulée. Et le CU se termine.

- Diagramme d'activités « effectuer commandes avec paiement direct »

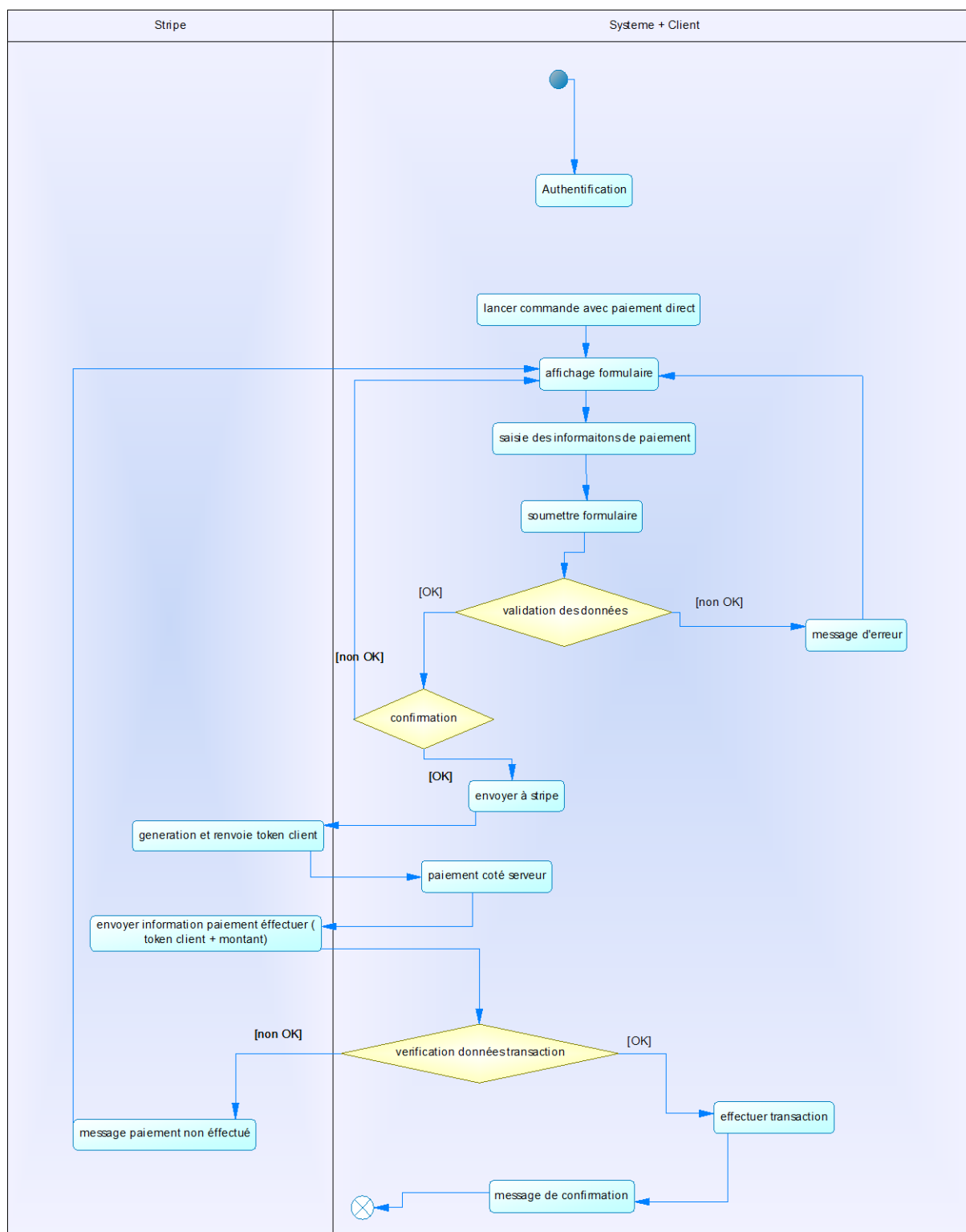


Figure 9: Diagramme d'activités « effectuer commandes avec paiement direct »

L'utilisateur, pour effectuer une commande avec paiement direct, doit se connecter. Ensuite accède à son panier puis lance une commande avec paiement direct. Et par continuité tout le scénario décrit au Tableau 4 est réalisé.

2) Spécification technique

Les spécifications techniques sont la description des choix techniques pris pour satisfaire les spécifications fonctionnelles et les besoins.

a) Mise à disposition des conditions de travail

Notre application étant une application web elle a besoin d'un serveur web en ligne. Ainsi AMITECH a mis à notre disposition un serveur en ligne pour l'hébergement de la solution. En effet cet hébergement nous permet de faire fonctionner notre application avec nos outils favoris et convenables à l'architecture logicielle choisie.

Le choix de ses outils de développement est présenté au point Le choix des outils de réalisation.

b) Architecture de développement

Une architecture en informatique désigne un mode de communication à travers un réseau entre plusieurs éléments physiques et/ou logiques. Il en existe plusieurs types dont les plus répandus et performants sont du type 2 tiers et 3 tiers. Cette étude consiste à présenter l'architecture autour de laquelle sera développé le futur système.

Néanmoins une étude comparative des différentes architectures est nécessaire. Ainsi cela est présenté dans le tableau suivant:

Architecture	Avantages	Inconvénients
Architecture un (1) tiers	<ul style="list-style-type: none"> ✓ rapidité d'exécution ✓ demande moins de ressources matérielles 	<ul style="list-style-type: none"> ✓ coût de déploiement élevé ✓ évolution difficile du système
Architecture deux (2) tiers	<ul style="list-style-type: none"> ✓ exploitation multi-utilisateur de la base de données de l'application ✓ répartition des charges entre les machines. ✓ offre une bonne sécurité des données. 	<ul style="list-style-type: none"> ✓ coût de déploiement élevé ✓ l'évolution difficile du système
Architecture trois (3) tiers	<ul style="list-style-type: none"> ✓ Une bonne disponibilité et une évolution facile. ✓ Un déploiement aisé ✓ Une sécurité accrue 	<ul style="list-style-type: none"> ✓ La mise en place et la maintenance demande un personnel informatique initié ✓ demande une bonne connexion réseau

Architecture	Avantages	Inconvénients
Architecture n-tiers	<ul style="list-style-type: none"> ✓ elle permet l'utilisation d'interfaces utilisateur riches ; ✓ elle sépare nettement tous les niveaux de l'application ; ✓ elle offre de grandes capacités d'extension ; ✓ elle facilite la gestion des sessions 	<ul style="list-style-type: none"> ✓ La mise en place et la maintenance demande un personnel informatique initié 1. demande une excellente connexion réseau

Tableau 5: Etude comparative des différentes architectures

Solution retenue

A la suite de l'étude comparative (Tableau 5) et vue les conditions de la mise en place de la plateforme, nous avons opté pour l'architecture trois (03) tiers car la plateforme devrait être développée et hébergée sur un serveur en ligne. Dans ce cas, nous avons une architecture composée d'un serveur de données qui héberge la base de données MySQL, d'un serveur d'applications qui est Apache et de postes clients. La Figure 10 illustre plus clairement l'architecture client-serveur web en trois (03) tiers adoptée :

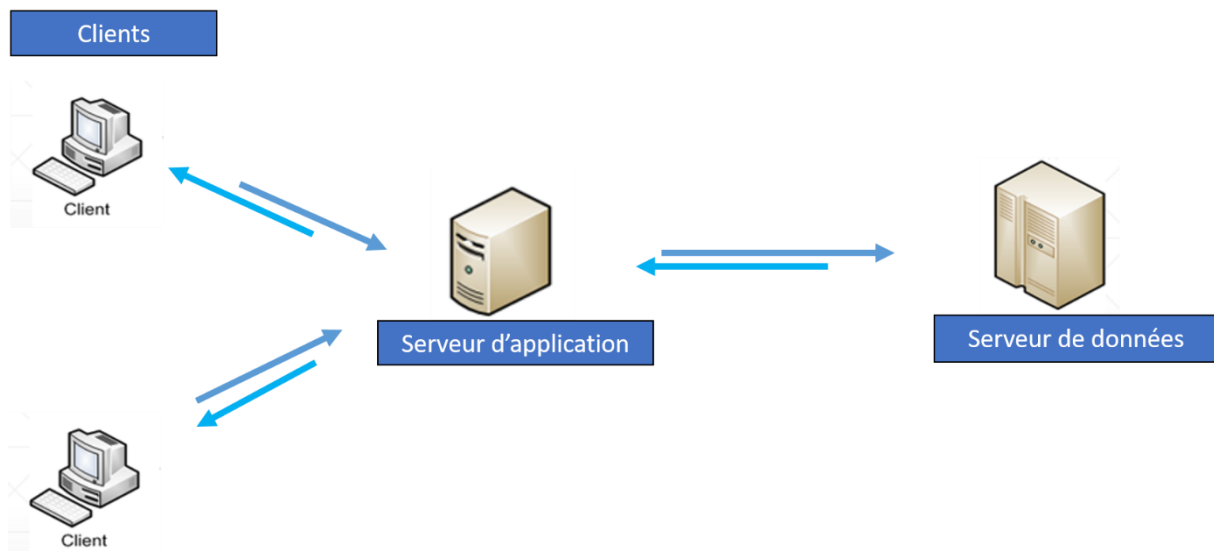


Figure 10:Schéma de l'architecture client-serveur web.

III. Conception globale

Pendant cette phase nous étudierons les spécifications de l'architecture générale du logiciel.

1) Diagramme de classes

Le diagramme de classes exprime la structure statique du système en termes de classes (avec les types de données manipulées et des méthodes) et la relation entre ces classes.

a) Dictionnaire de données

Le dictionnaire de données (Tableau 6) contient l'ensemble des descriptions des attributs de chaque classe qui est utilisée dans le système. Pour faciliter la lecture, nous allons faire la description classe par classe dans le tableau ci-après :

Attribut	Description	Type de données
Celle-ci ont la même signification dans toutes les Classes dans lesquelles elles sont		
id	Identifiant de la table dans la BD	Int
slug	le nom ou le titre de la table sans espace dans le nom	String
created_at	Date de création du champ de la table dans la BD	Date
updated_at	Date de mise à jour du champ de table dans la BD	Date
Classe « User » (La Classe des utilisateurs)		
name	Le nom complet de l'utilisateur	String
email	L'adresse email	String
image	Sa photo de profil	String
provider	Représente pour le réseau social d'authentification	String
provider_id	L'identifiant du réseau social d'identification	Int
email_verified_at	Date de vérification de l'email	String
password	Le mot de passe de connexion	String
remember_token	Permet de mémoriser une connexion	String
Classe « Roles » (La représente un privilège d'accès de l'utilisateur)		
role	Nom du privilège de l'utilisateur	String
Classe « Media » (Représente un media comme une vidéo, images, ...)		
title	Titre du media	String
src	Lien vers le media	String
type	Le type de media	String
Classe « Products » (Représente un produit de vente)		
name	Nom du produit	String
details	Certains détails sur le produits	String
price	Le prix du produit	Int
short_description	Une courte description sur le produit	String
complete_description	Une description complète sur le produit	String
quantity	La quantité disponible du produit	Int
images	Différentes images du produit	String
old_price	Ancien prix du produit	Int
Classe « Cart » (Représente le panier de l'utilisateur)		
name	Le nom du panier (Nom de l'utilisateur par défaut)	String
Classe « ProductsCart » (Les commandes de produits dans un panier)		
quantity	La quantité voulue du produit	Int
All_details	Tous les détails sur la commande	String

Attribut	Description	Type de données
Classe « Order » (Représente une commande)		
billing_name	Le nom de la commande	String
billing_email	L'adresse email sur la commande	String
billing_address	Autre adresse de l'utilisateur	String
billing_country	Pays de réception de la commande	String
billing_city	Ville de réception de la commande	String
billing_province	Province de réception de la commande	String
billing_postal_code	Adresse postal de réception de la commande	String
billing_phone	Téléphone de l'utilisateur	String
billing_name_on_card	Nom de la carte bancaire	String
billing_discount	Prix de rabais	String
billing_discount_code	Code de rabais	String
billing_sub_total	Prix total de la commande hors taxes	Int
billing_tax	Prix des taxes sur la commande	Int
delivery_price	Prix de livraison de la commande	Int
billing_total	Prix TTC total de la commande et livraison	Int
Classe « BankPayment » (Représente un paiement par carte bancaire)		
card_type	Le type carte utilisé pour le paiement	String
card_last_four_digits	Les quatre derniers chiffres sur la carte	Int
Classe « Category » (Représente une catégorie de produit)		
name	Le nom de la catégorie	String
Classe « OrderState » (Représente l'état d'évolution dans la livraison de la		
state	Nom montrant l'état d'évolution de la commande	String

Tableau 6:Dictionnaire de données

b) Diagramme de classes

Notre diagramme de classes est présenté comme suit :

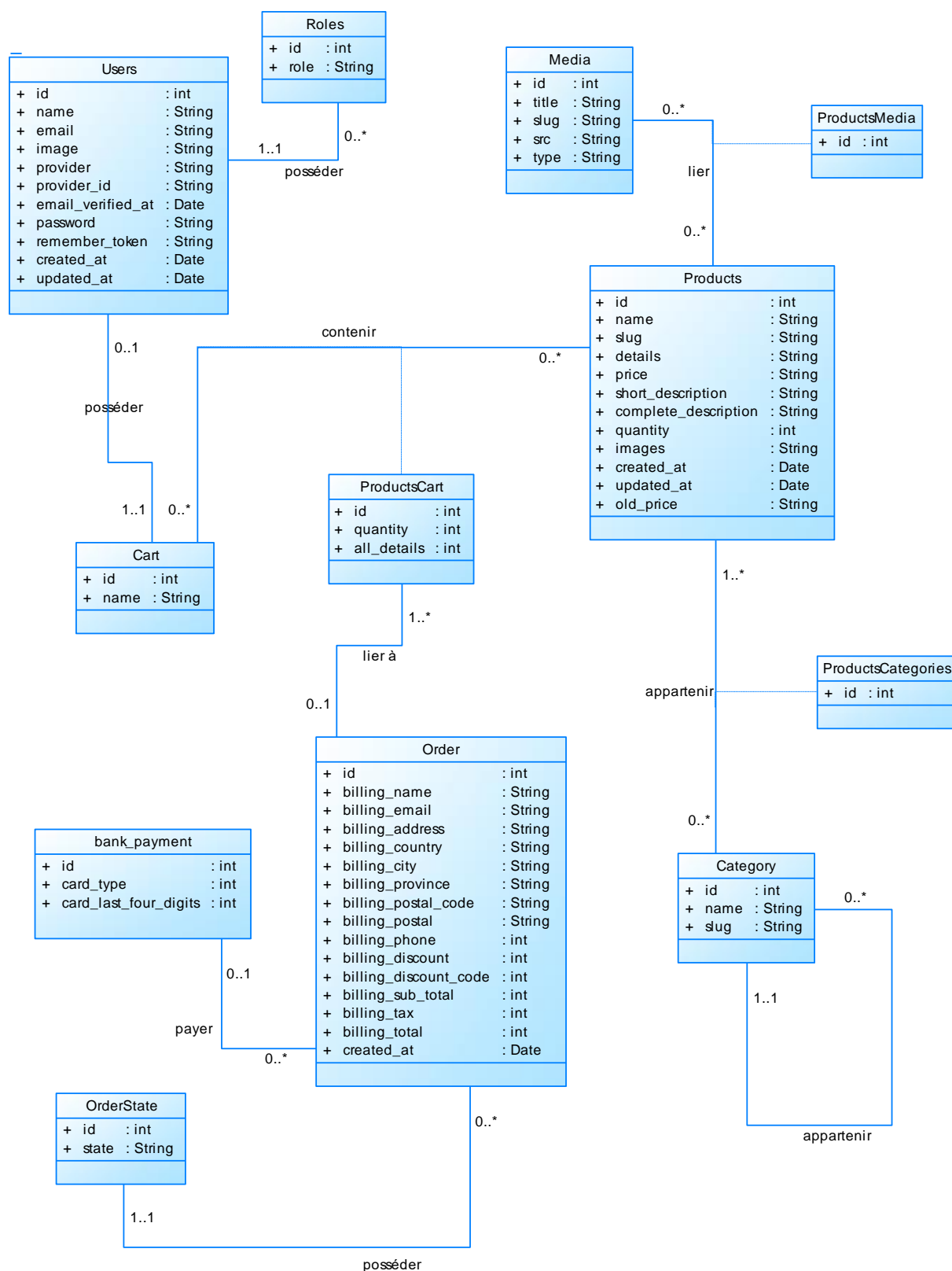


Figure 11: Le Diagramme de classes

Notre diagramme de classes (Figure 11) nous montre par exemple qu'un utilisateur (classe « Users ») possède 0 ou 1 Panier (classe « Cart ») et que un panier peut contenir 0 ou plusieurs produits (classe « Products »).

2) **Diagrammes de séquences vues de l'intérieur**

Dans cette partie, nous illustrons des interactions entre le système et les acteurs. Le cas d'utilisation mis en exergue est « gestion panier » (Figure 12) avec une vue sur les interactions qui se passent au sein du système. Ce diagramme est illustré par la figure suivante :

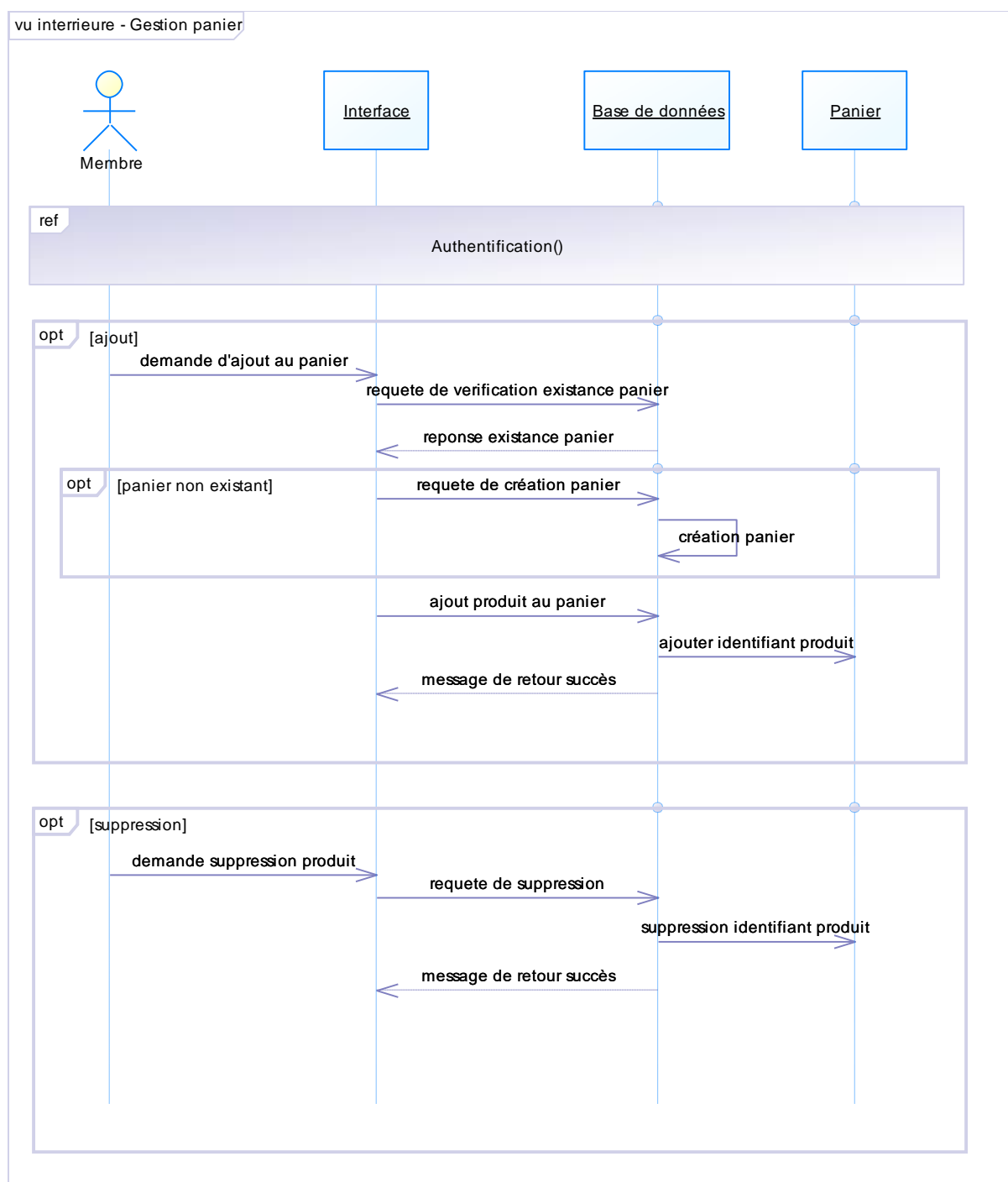


Figure 12: Diagramme de séquence vu de l'intérieur du CU « gestion panier »

Sur ce diagramme (Figure 12) un membre (objet étiqueté « membre ») s'authentifie puis choisit sur l'interface de faire un ajout par exemple, puis demande sur l'interface d'ajouter le produit au panier. Ensuite une requête de vérification de l'existence du panier est envoyée vers la base de données. Et ainsi les actions se suivent et tout le scénario décrit dans le Tableau 3 est réalisé.

3) Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels.

Notre diagramme de déploiement (Figure 13) comprend trois (03) nœuds qui représentent le serveur d'application, le serveur de données ainsi qu'une occurrence de poste client. Ce diagramme de déploiement laisse entrevoir l'architecture logicielle trois (03) tiers qui est utilisée pour ce système.

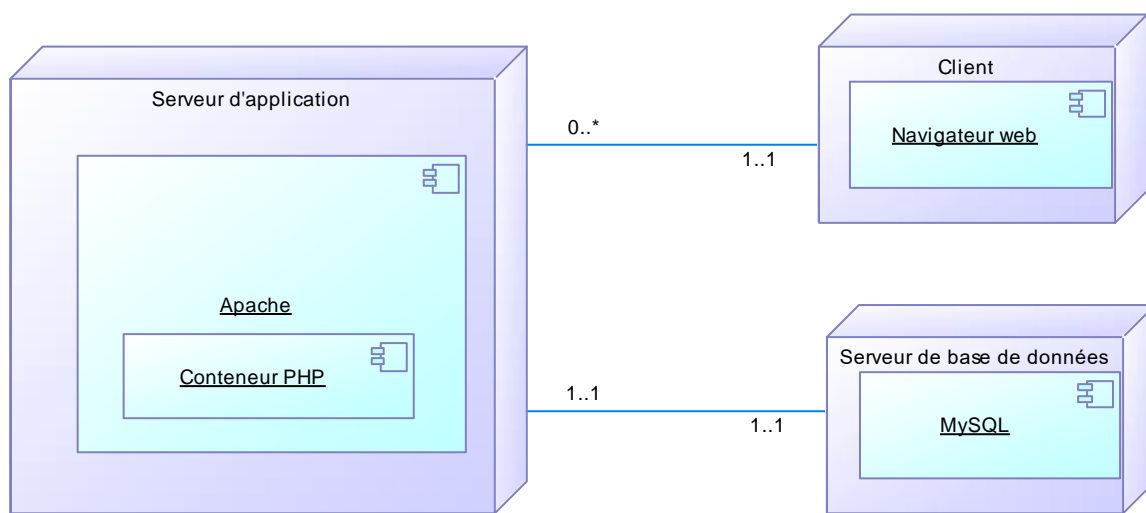


Figure 13: Diagramme de déploiement

Ce diagramme (Figure 13) montre les trois (3) nœuds qui représentent notre architecture. Un ou plusieurs clients web peuvent envoyer des requêtes vers le serveur d'applications qui puise ses données dans un autre nœud, le serveur de base de données. Le serveur d'application (apache) contient le moteur d'interprétation des scripts PHP qui traite et génère le code final de la page qu'il renvoie au navigateur (client web). Le serveur d'applications récupère les données enregistrées demandées dans le serveur de base de donnée qui est MySQL.

IV. Réalisation

Plusieurs éléments importants entrent en compte dans la réalisation d'un logiciel. Ces éléments peuvent être le choix des outils de réalisation, l'architecture de l'application, les maquettes, les politiques de sécurité et le coût de réalisation de la solution, etc. Nous les mettons en lumière dans cette partie de l'étude.

1) Choix des outils de réalisation

Le choix des outils de réalisation est très important car ils auront un impact dans la rapidité de développement et dans la performance du système.

a) **Système de Gestion de Base de Données**

Un système de gestion de base de données (SGBD) relationnelle est un logiciel qui permet de manipuler le contenu des bases de données relationnelles. Pratiquement tous les systèmes relationnels utilisent le langage SQL pour interroger les bases de données.

Pour le choix de la base de données à utiliser dans la réalisation de notre plateforme, nous allons, dans un tableau (Tableau 7), procéder à l'étude comparative de quelques SGBD bien connus à savoir Oracle, MySQL, Microsoft SQL Server et PostgreSQL.

	Avantages	Inconvénients
Oracle	<ul style="list-style-type: none"> - Grande base de données. Pour plus de 200Go et plus 300 utilisateurs, les écarts de performances sont visibles par rapport à d'autre SGBD - Richesse fonctionnelle - Tout est paramétrable. - Service web très puissant 	<ul style="list-style-type: none"> - Prix élevé: environ 45000€ pour un serveur - Administration complexe - Gourmand en ressources mémoire - certains bogues
MySQL	<ul style="list-style-type: none"> - Gratuit et Open Source - Simple d'installation et d'utilisation - Fonctionnel rapidement - Bonne intégration dans l'environnement Apache/PHP Service Web 	<ul style="list-style-type: none"> - Ne convient pas pour de grosses bases de données - Sécurité moyenne - Pas de sauvegarde à chaud - Peu de richesse fonctionnelle
Microsoft SQL Server	<ul style="list-style-type: none"> - Administration aisée - Une des bases les plus performantes sous Windows - Services Web - Support XML - Supporte les 4 niveaux d'isolation transactionnelle de la norme SQL - Compression des données 	<ul style="list-style-type: none"> - Disponible seulement sur Windows - Mono-plateforme (MS Windows) - Il peut être plus « coûteux » de parcourir les pages d'index que d'analyser la table dans son intégralité. - Pas de java, orientation C#

	Avantages	Inconvénients
PostgreSQL	<ul style="list-style-type: none"> - Open Source et gratuit - Fiable et relativement performant, - Supporte la majorité du standard SQL-92 - Possède de nombreuses d'extensions (Java, Ruby, PL-SQL) - Très riche fonctionnellement, - Simple d'utilisation et d'administration - Héritage de tables 	<ul style="list-style-type: none"> - Pas possible de requêter sur plusieurs bases à la fois : chaque base nécessite sa connexion - Sauvegardes peu évoluées - Supporte les bases de moyenne importance - Pas de services Web - Pas d'ordonnanceur intégré - Pas de fonctions d'agrégat OLAP - Pas de requêtes récursives

Tableau 7: Tableau comparatif des SGBD

Au regard des forces et des faiblesses que présente chaque SGBD, nous avons choisi MySQL. Ce choix est justifié par le fait que MySQL est un SGBD très simple à comprendre, à utiliser, est bien documenté, open-source et surtout gratuit.

b) Serveur d'application

Un serveur d'applications expose la logique métier aux applications clientes via divers protocoles, y compris éventuellement HTTP. C'est-à-dire qu'il permet aux utilisateurs d'accéder aux différents services qu'il héberge. Ces applications sont accessibles via un navigateur web. Il existe de nombreux serveurs d'application dont nous pouvons citer : Apache, NGINX et Tomcat, etc.

Le tableau (Tableau 8) suivant affiche une étude comparative de ces trois serveurs.

SGBD	Fonctionnalité(s) possédée(s)	Fonctionnalité(s) absente(s)	Plateforme(s)
Apache	<ul style="list-style-type: none"> ➤ Authentification ➤ Support HTTPS ➤ Domaine virtuel ➤ Console d'administration ➤ Support IPv6 ➤ ASP.NET (via module) 	Java Servlets	<ul style="list-style-type: none"> ➤ Windows ➤ Linux ➤ Mac OS
NGINX	<ul style="list-style-type: none"> ➤ Authentification (basique seulement) 	<ul style="list-style-type: none"> ➤ ASP.NET ➤ Java Servlets 	<ul style="list-style-type: none"> ➤ Windows ➤ Linux ➤ Mac OS

SGBD	Fonctionnalité(s) possédée(s)	Fonctionnalité(s) absente(s)	Plateforme(s)
	<ul style="list-style-type: none"> ➤ Support HTTPS ➤ Domaine virtuel ➤ Support IPv6 	<ul style="list-style-type: none"> ➤ Console d'administration 	
Tomcat	<ul style="list-style-type: none"> ➤ Authentification ➤ Support HTTPS ➤ Domaine virtuel ➤ Java Servlets ➤ Console d'administration ➤ Support IPv6 ➤ ASP.NET (via module) 	ASP.NET	<ul style="list-style-type: none"> ➤ Windows ➤ Linux ➤ Mac OS

Tableau 8: Etude comparative de quelques serveurs d'application

Dans le développement de notre plateforme nous avons choisi Apache comme serveur d'application. Notre choix se justifie par le fait qu'Apache fonctionne parfaitement avec de nombreux Framework web (Django, Laravel, etc.) et de langages de programmation. Aussi, il a une grande communauté d'utilisateurs. Cela en fait un choix solide pour tous les types de plateformes d'hébergement web, comme un VPS (Virtual Private Server ou Serveur Dédié Virtuel en français) ou l'hébergement web mutualisé.

c) Langages de programmation

Les langages de programmation que nous avons utilisés sont :

- **PHP 7.1.3**: PHP est un langage de programmation procédurale et orientée objet. Nous l'avons choisi car c'est l'un des langages de programmation web les plus utilisés.
- **JavaScript** : un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation de Node.js.
- **HTML 5** : est un langage de marquage et de balisage servant à écrire des pages pour le World Wide Web.
- **CSS 3** : Le CSS (Cascading Style Sheet) permet de faire la mise en forme des pages web.

d) Outils de conception

Les outils de conception représentent l'ensemble des outils utilisés depuis la conception jusqu'à la réalisation de l'application. Ces outils sont :

- **Sublime Text 3** : Un éditeur de texte sophistiqué pour le code. Il nous a permis d'écrire les codes de notre application.

- **GitLab** : c'est un logiciel web libre offrant un système de suivi des bugs et permettant l'intégration continue et la livraison continue. Présenté comme la plateforme des développeurs modernes, il offre la possibilité de gérer ses dépôts Git et ainsi de mieux appréhender la gestion des versions de vos codes sources. Il nous a permis d'héberger, de gérer, et de sécuriser le code de notre projet.
- **Git 2.22.0**: Git est un logiciel de gestion de versions décentralisé. Il nous a permis de gérer la mise sur gitlab de notre plateforme.
- **Laragon Full 3.1.9** : Laragon est un environnement de développement web rapide, flexible, intuitif, productif et puissant, pouvant faire fonctionner Apache et MySQL que nous avons utilisés dans notre développement.
- **phpMyAdmin 4.51**: phpMyAdmin nous a permis d'administrer notre base de données.
- **Power AMC version 15.1** : power AMC est un outil de conception. Il nous a aidé à la conception de nos différents diagrammes UML.
- **Microsoft Office Word et Power Point 2016**: ce sont des outils efficaces pour le traitement des textes. Ils nous ont servi à rédiger la documentation nécessaire à la connaissance et à l'utilisation de l'application.
- **Adobe Illustrator CC 2018** : c'est un logiciel de création graphique vectorielle qui nous a aidé dans certaines conceptions graphiques.

e) Plateformes de développement

Les plateformes de développement (ou Framework) que nous avons utilisées sont :

- **Laravel 5.8**: Laravel est un Framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub. Et nous avons choisi ce Framework car il possède une grande communauté et une documentation très bien faite avec beaucoup de bibliothèques externes, aussi il est relativement facile à prendre en main. Laravel était donc adapté à ce type de projet.
- **jQuery 3.4.1** : jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Il nous a permis aussi d'écrire des requêtes Ajax (pour la récupération de données dans le serveur sans rechargement de la page).
- **Bootstrap 4.3.1** : Bootstrap est une boîte à outils open source pour le développement avec HTML, CSS et JS.

2) Présentation de l'architecture MVC de notre application

Laravel, comme une grande partie des autres Framework PHP, a une architecture dite MVC (Model – View – Controller).

Chaque action de l'utilisateur passe par le contrôleur (controller) qui envoie des demandes de manipulation d'un objet au modèle. Le modèle (model) effectue les changements de l'objet et le renvoie au contrôleur qui à son tour le passe à la vue (view).

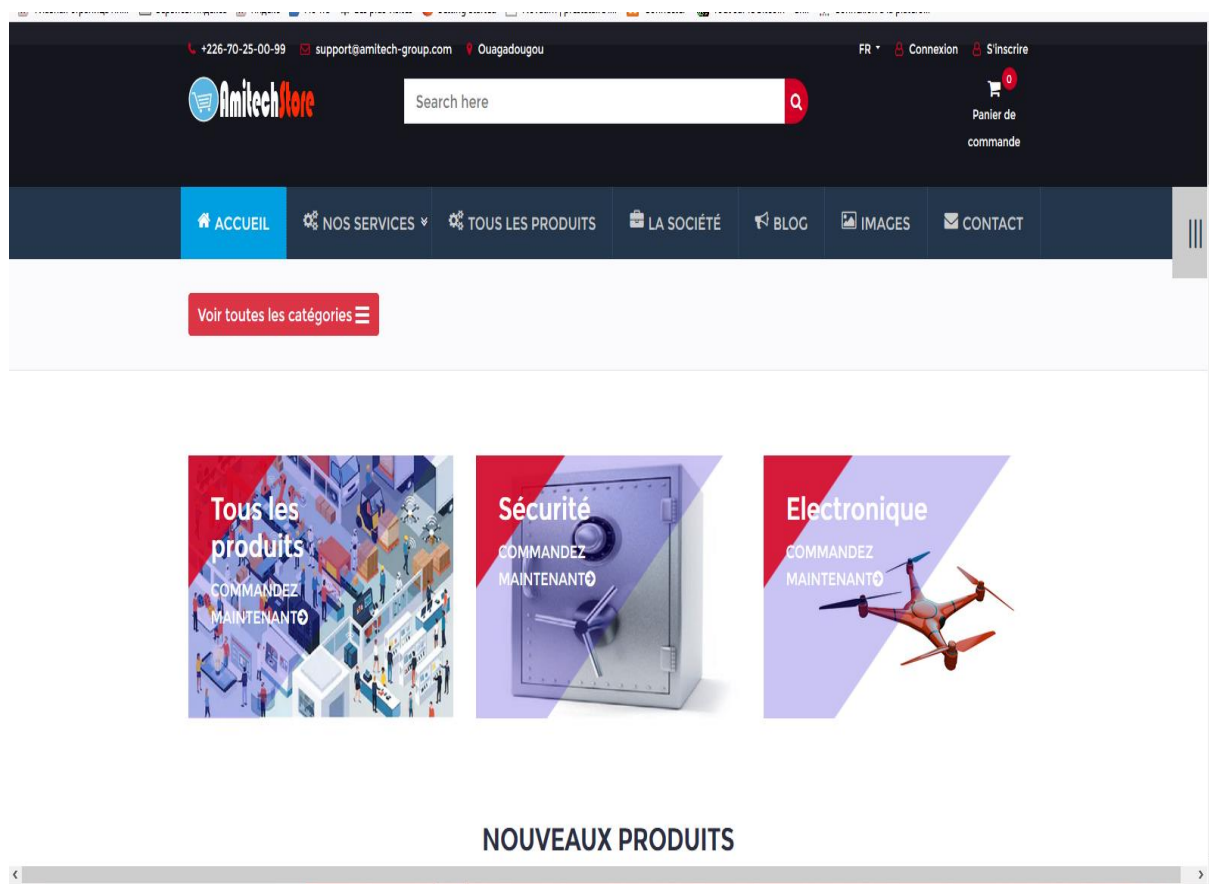
Ainsi dans le cas de notre application nous avons:

- Le **model** qui contient les données et leur logique. Dans notre application les modèles sont stockés dans le dossier « *app* » situé à la racine des dossiers du site. Aussi notre application Laravel comporte en son sein un ORM performant qui s'appelle Eloquent. Un fichier de configuration permet de paramétrer la connexion à la base de données, c'est tout ce qu'il y a à faire. Laravel s'occupera alors de faire ce qu'on lui demandera.
- La **view** contient la présentation graphique à renvoyer à l'utilisateur. Aussi dans notre application les vues sont mises dans le dossier « *resources/views* » à partir de la racine des dossiers du site.
- Le **controller** traite les actions utilisateurs (via des requêtes), demande au modèle d'effectuer les changements, puis passe les données à la vue. Dans notre application les contrôleurs sont dans le dossier « *app/Http/Controllers* » à partir de la racine des dossiers du site.

Le **controller** a donc une place centrale dans cette architecture. Il est le pont entre les interactions utilisateurs et les traitements de données.

3) Présentation de quelques maquettes IHM de notre application

Nous présentons ici quelques maquettes IHM de notre application :

a) Page d'accueil (Figure 14)*Figure 14: La page d'accueil***b) Page d'authentification (Figure 15)**The screenshot shows the login page titled 'Connexion'. It features two social login buttons: 'G Sign in with Google' and 'f Sign in with Facebook'. Below these are input fields for 'Adresse email' and 'Mot de passe'. A checkbox labeled 'Se souvenir de moi' is present. A blue 'Connexion' button is located below the password field. A link 'Vous avez oublié votre mot de passe ?' is positioned below the 'Connexion' button. At the bottom, there is a red 'S'inscrire' button.*Figure 15: La page d'authentification*

c) Le catalogue des produits (Figure 16)

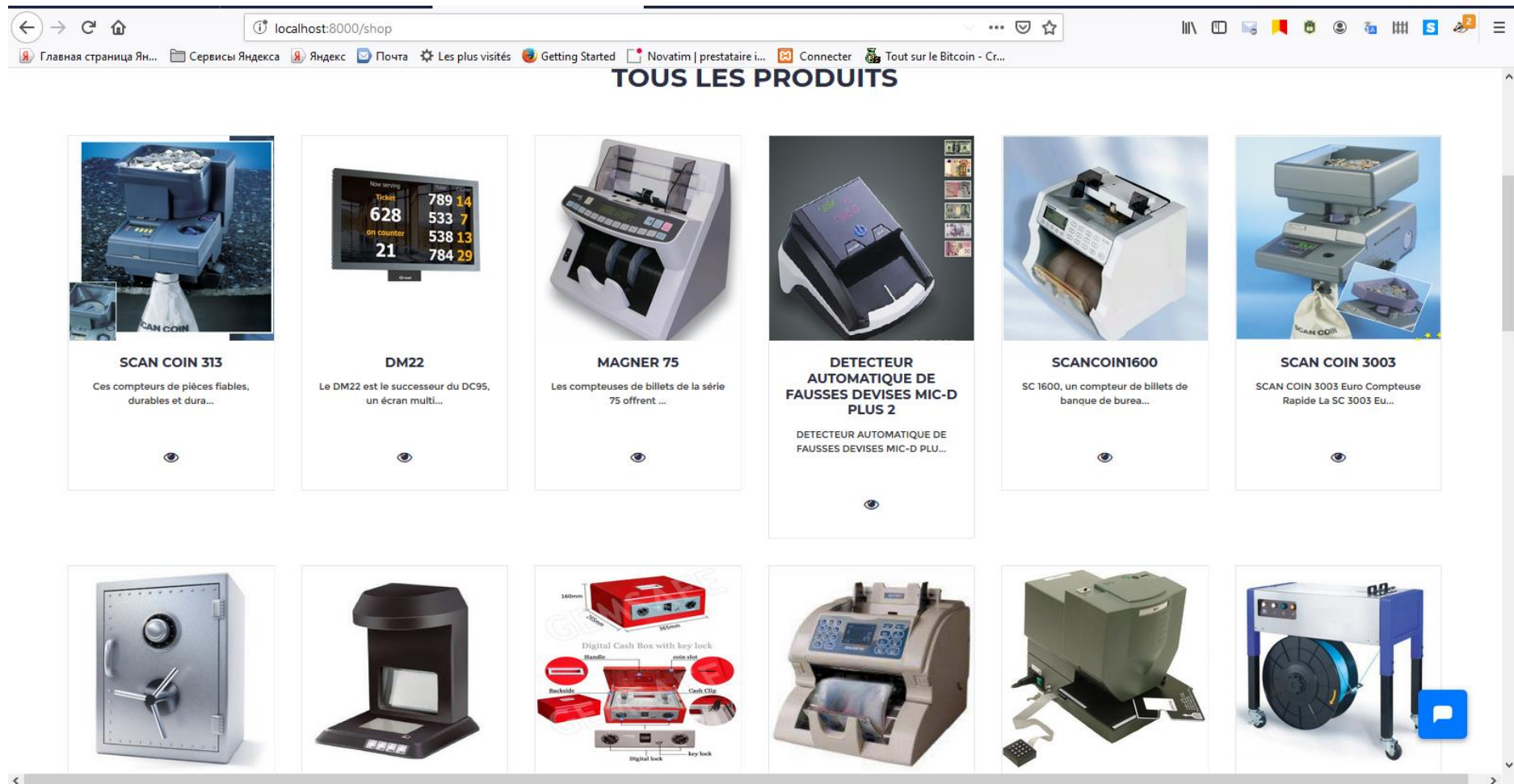




Figure 16: Le catalogue des produits

d) Page de détails sur un produit (Figure 17)

SCAN COIN 22 TRIEUSE DE PIÈCES

IN STOCK

La trieuse de pièces SC 22 est compacte et légère. Grâce à ses fameux capteurs SCAN COIN, elle est d'une précision constante et est dotée de fonctionnalités extrêmement fiables.

QUANTITY

ADD TO CART

CATEGORY: TRIEUSE DE PIÈCES

SHARE: [f](#) [t](#) [G+](#) [e](#)

Most details

Performance

La trieuse de pièces SC 22 est reconnue dans le monde entier pour l'alliance de sa technologie de pointe, sa petite taille et son fonctionnement simple. Cette machine est dotée du rejet automatique des pièces étrangères ou endommagées et d'arrêts automatiques programmables. Elle trie jusqu'à 8 dénominations de pièces. L'affichage alpha-numérique facilite l'exploitation du logiciel avec des sous-totaux, des grands totaux, la date et l'heure. L'interface série standard permet la connexion à une imprimante, à un PC ou à notre système de gestion des espèces (CSS).

Compacte et simple d'utilisation

La SC 22 est une trieuse de pièces très compacte et légère. La fiabilité de sa technologie de comptage répond aux besoins des établissements bancaires, postaux, de la GMS et de la restauration rapide. La simplicité de son fonctionnement et de son entretien sont des atouts de la SC 22. De conception solide et dotée de composants de très haute qualité, elle nécessite un minimum de maintenance. Grâce à sa taille la SC 22 requiert peu d'espace et est facile à déplacer et à manipuler.




Flexibilité des équipements

La SC 22 peut être facilement équipée pour la mise en sacs ou en rouleaux de quantités prédéterminées. Avec son stand de mise en sacs, la SC 22 peut aussi être transformée en unité indépendante. D'autres accessoires sont également disponibles notamment une imprimante additionnelle et une grande trémie de pièces.

Figure 17: La page des détails sur un produit

e) Le panier d'achat (Figure 18)

LES PRODUITS DANS VOTRE PANIER

#	Image	Name	Quantity	Action
1		Kisan	1	✕ Remove form cart
2		Indupack	1	✕ Remove form cart
3		Coffre fort	1	✕ Remove form cart

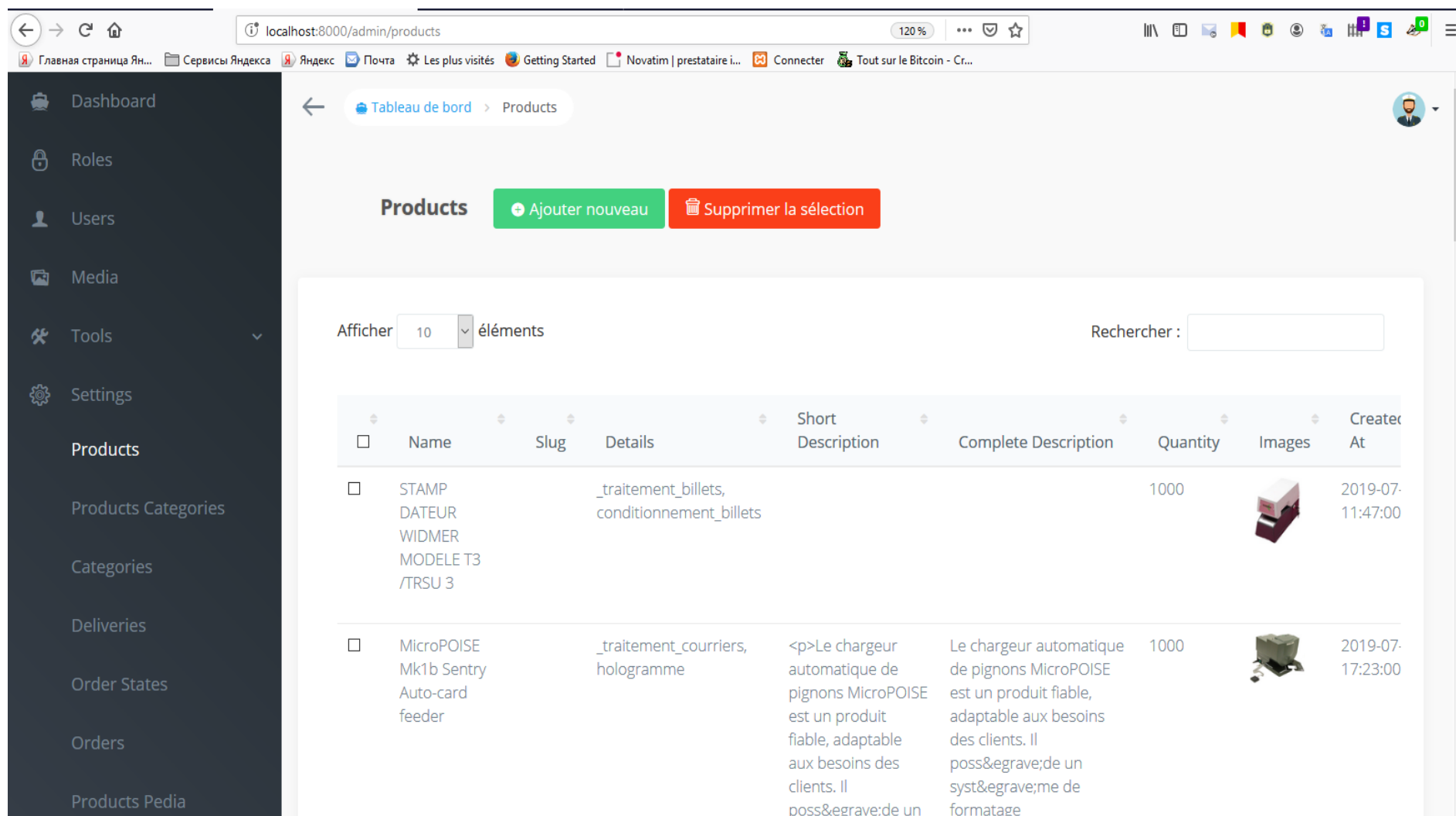
ADRESSE DE COMMANDE

VOTRE COMMANDE

[PASSEZ LA COMMANDE PAR MAIL](#)

Figure 18:Le panier d'achat

f) Page d'administration des produits



The screenshot displays the 'Products' administration page. The sidebar on the left contains links for Dashboard, Roles, Users, Media, Tools, Settings, Products, Products Categories, Categories, Deliveries, Order States, Orders, and Products Pedia. The top navigation bar shows the current location as 'Tableau de bord > Products' and includes a user profile icon. The main content area features a 'Products' header with two buttons: 'Ajouter nouveau' (Add new) and 'Supprimer la sélection' (Delete selection). Below this, there is a search bar and a table of products.

Products Ajouter nouveau Supprimer la sélection

Afficher 10 éléments Rechercher :



<input type="checkbox"/>	Name	Slug	Details	Short Description	Complete Description	Quantity	Images	Created At
<input type="checkbox"/>	STAMP DATEUR WIDMER MODELE T3 /TRSU 3		_traitement_billets, conditionnement_billets			1000		2019-07-11:47:00
<input type="checkbox"/>	MicroPOISE Mk1b Sentry Auto-card feeder		_traitement_courriers, hologramme	<p>Le chargeur automatique de pignons MicroPOISE est un produit fiable, adaptable aux besoins des clients. Il possède un	Le chargeur automatique de pignons MicroPOISE est un produit fiable, adaptable aux besoins des clients. Il possède un système de formatage	1000		2019-07-17:23:00

Figure 19: Page d'administration des produits

4) **Politique de sécurité**

La sécurité est un critère très important dans la grande majorité des applications web et notre application en fait partie. De ce fait nous avons pensé à cette sécurité à tous les niveaux de son cycle de vie. Ainsi nous avons pris en compte différents paramètres qui entrent en jeu dans la sécurisation des applications web et nous nous sommes inspirés en partie des recommandations de OWAPS (Open Web Application Security Project) qui s'illustre comme une référence dans la sécurisation des applications web. Nous avons donc pris plusieurs mesures de sécurité en développement et nous préconisons d'autres qui doivent être prises en compte en production.

➤ **Politique de sécurité dans l'application**

✓ **La sécurité dans le choix du Framework Laravel :**

Laravel est une plateforme de développement populaire, réputée pour ses performances et sa communauté d'utilisateurs actifs. Laravel est assez sécurisé. A chaque fois qu'une faille est découverte, l'équipe de maintenance en prend soin dès que possible.

✓ **Violation de gestion d'authentification et de session :**

Cette faille de sécurité regroupe toutes les vulnérabilités pouvant mener à une usurpation d'identité. Ces points de faiblesse dans les applications Web peuvent ouvrir à des attaquants des accès à des fonctionnalités des applications Web auxquelles ils n'ont pas le droit normalement. Cela peut donc leur permettre de voler des informations ou d'endommager le bon fonctionnement de l'application.

Nous avons utilisé un système d'authentification et défini une durée de vie d'une session. Une période d'inactivité maximale de 120 secondes a été définie. Si l'utilisateur n'utilise pas l'application pendant ce laps de temps, la session devient inutilisable et l'utilisateur doit se reconnecter.

En plus, un système de filtrage est utilisé pour filtrer les accès en fonction du profil de l'utilisateur. En effet la mise en place de middleware (contrôleur) dans Laravel est très facile et efficace pour les contrôles d'accès et beaucoup d'autres types de contrôles.

Pour contrer les attaques de force brute, les comptes sont verrouillés après trois (3) tentatives consécutives infructueuses de connexion. La procédure de déverrouillage est automatique après un laps de temps prédéfini.

✓ **Réduction des vulnérabilités provenant de CSRF (falsification de requêtes inter-sites en anglais *cross-site request forgery*)**

Laravel utilise généralement des jetons CSRF pour s'assurer que des tiers externes ne peuvent pas générer de fausses demandes et ne doivent pas violer les vulnérabilités de sécurité de Laravel. Pour cela, Laravel crée et intègre un jeton valide dans chaque requête provenant d'une forme ou d'un appel AJAX. Lorsque la demande est appelée, Laravel compare le jeton de demande à celui enregistré dans la session de l'utilisateur. Si le jeton ne correspond pas, la demande est classée comme non valide et aucune autre action n'est exécutée.

✓ **Protection contre XSS (Cross-Site Scripting)**

Lors d'attaques XSS, l'attaquant entre du JavaScript (généralement dans les zones de texte d'un formulaire) dans votre site Web. Désormais, chaque fois que de nouveaux visiteurs accèderont à la page de formulaire concernée, le script sera exécuté avec un impact malveillant. Laravel offre un support natif qui protège le code des attaques XSS. La fonctionnalité se déclenche automatiquement et protège la base de données dans le processus.

✓ **Injection SQL**

Eloquent ORM de Laravel utilise une liaison PDO (PHP Data Objects) qui protège des injections SQL. Cette fonctionnalité garantit qu'aucun client ne peut modifier l'intention des requêtes SQL.

✓ **La sécurité SSL (Secure Sockets Layer)**

Pour notre système, il était nécessaire d'ajouter un module supportant les protocoles sécurisés comme SSL afin d'assurer des fonctions de sécurité très élevées. En effet, la technologie SSL est une norme dans le domaine de la sécurisation des transactions Internet. Le certificat SSL est donc une sorte de signature numérique du site Internet. Le certificat va être utilisé pour crypter l'ensemble des transactions entre un ordinateur et le site Internet.

Le certificat SSL assure la confidentialité et empêche l'espionnage ou l'interception de données. Il rend très difficile voire impossible le piratage des informations échangées entre le serveur et le navigateur. Il permet d'authentifier un site et assure au visiteur qu'il s'agit bien du site officiel recherché.

➤ **Sécurité dans le système de paiement à travers Stripe**

Toute personne impliquée dans le traitement, la transmission ou le stockage des données des cartes doit se conformer aux normes de sécurité des données de l'industrie des cartes de paiement (**Payment Card Industry Data Security Standard** ou **PCI DSS**). Stripe a été audité par un PCI QSA(PCI Qualified Security Assessor) indépendant et est certifié PCI Level 1

Service Provider. Il s'agit du niveau de certification le plus rigoureux disponible dans l'industrie des paiements.

La conformité PCI est une responsabilité partagée et s'applique à Stripe et à l'entreprise utilisatrice. Lorsque sont acceptés des paiements, cela doit se faire d'une manière conforme à la norme PCI. La façon la plus simple d'être conforme à la norme PCI est de ne jamais voir (ou d'avoir accès) les données de la carte. Avec Stripe, c'est facile à mettre en œuvre car il fait le plus gros du travail pour protéger les renseignements sur les cartes des clients. Nous pouvons simplifier notre conformité PCI tant que nous le souhaitons et cela nous devons:

- ✓ utiliser l'une des recommandations d'intégrations de paiement pour recueillir des informations de paiement, qui sont transmises directement à Stripe sans passer par nos serveurs.
- ✓ servir nos pages de paiement en toute sécurité avec la sécurité de la couche transport pour qu'elles utilisent HTTPS.
- ✓ vérifier et valider la conformité PCI de notre compte chaque année.

Plus de détails sur stripe en ANNEXE 3 : L'utilisation et le fonctionnement de Stripe

➤ **Mesure de sécurité prise pendant le développement**

Pendant le développement de notre application nous avons hébergé le code source sur un dépôt Git privé afin de prévenir tout incident qui pourrait affecter la machine de développement et causer des pertes de données par exemple.

➤ **Politique de sauvegarde**

La politique de sauvegarde consiste à prendre des mesures nécessaires pour préserver l'intégrité des données en cas de dysfonctionnement du système. Alors, nous préconisons :

- ✓ des sauvegardes journalières qui ont une durée d'une semaine ;
- ✓ des sauvegardes hebdomadaires qui ont une durée d'un mois ;
- ✓ des sauvegardes mensuelles qui ont une durée de six (06) mois ;
- ✓ des sauvegardes semestrielles qui ont une durée d'un an ;
- ✓ des sauvegardes annuelles qui seront conservées définitivement.

5) **Coût de réalisation**

La détermination du cout de développement est une étape clé dans le processus de réalisation d'un logiciel. En ce sens qu'elle permet d'estimer l'effort à fournir et le temps que prendra le

développement logiciel en fonction des ressources qui seront allouées. Il existe néanmoins plusieurs méthodes d'estimation de cout de développement. Mais, dans le contexte de notre projet, notre choix s'est porté sur la méthode COCOMO simplifiée (CONstructive COSt MOdel) car cette méthode permet d'estimer le coût d'un projet logiciel dans le but d'éviter les erreurs de budget et les retards de livraison. Toutes les explications sur l'utilisation de cette méthode vous sont présentées en ANNEXE 2 : La méthode COCOMO.

Notre application correspond aux critères d'une application de types S suivant la méthode COCOMO. De plus, nous estimons le nombre de lignes de code de notre application à cinq mille (5000).

Nous pouvons donc estimer le cout de développement comme suit :

$$\text{Effort} = 2,4 * (5000/1000)^{1,05} = 13 \text{ Hommes/Mois}$$

$$\text{TDEV} = 2,5 * \text{Effort}^{0,38} = 2,5 * 13^{0,38} = 6,62 \text{ mois}$$

$$\text{CDEV} = 13 * X \text{ FCFA} = 13 * 200\,000 \text{ FCFA} = 2\,600\,000 \text{ FCFA}$$

NB : Nous pouvons estimer le coût total du projet en considérant le coût du développement, le coût de la formation des administrateurs du site, ainsi que celui du matériel informatique et des logiciels que nécessite le projet. En supposant que le salaire d'un ingénieur informatique soit égal à 200000 F CFA et en supposant une somme forfaitaire de 125 000 FCFA pour cinq (05) jours de formation des utilisateurs en raison de 25000F FCFA par séance de formation, on obtient le coût total du projet tel que présenté dans le Tableau 9. **Erreur ! Source du renvoi introuvable..**

Désignation	Quantité	Cout Unitaire (en FCFA)	Disponibilité ou cout total (en FCFA)
Ordinateur Portable	1	250 000	250 000
Système Windows	-	-	Version d'essais
Anti-virus(Kaspersky Total Security)	1	16 000	16 000
Gantt project	1	-	Licence gratuite
Sybase PowerAMC Concepteur Studio	1	3 714 505	Version d'évaluation
Sublime Text 3	1	-	Version gratuite

Désignation	Quantité	Cout Unitaire (en FCFA)	Disponibilité ou cout total (en FCFA)
Laragon	1	-	Licence gratuite
Nom de domaine (.com)	1	4 600	4 600
Hébergement (chez l'hébergeur OVH)	1	60 000	60 000
Utilisation Stripe	-	-	Inscription gratuite et les commissions sont perçues à chaque paiement effectués au taux de 2,9%+180FCFA
Connexion internet	-	-	Disponible
Formation	5	25 000	125 000
Développement	-	-	2 600 000
COUT TOTAL			3 055 600 FCFA

Tableau 9: Cout de réalisation de la plateforme.

Ce chapitre nous a permis de mettre en lumière les différents éléments d'analyse et de conception de notre système. Ainsi nous avons détaillé les éléments concernant l'étude préalable, l'expression des besoins, la conception globale, et la réalisation. Tous ces éléments ont constitué la grande partie des travaux de notre stage.

CHAPITRE 3 : BILAN DE STAGE

Ce dernier chapitre fera l'objet de notre bilan de stage. En effet, nous allons tout d'abord présenter son déroulement et les activités menées au cours de ce stage et nous terminerons par la présentation de nos observations, suggestions et perspectives.

I. Présentation du déroulement du stage et des activités réalisées

Notre stage s'est passé durant la période du 01 juillet au 30 septembre 2019. Période au cours de laquelle nous avons eu à mener des activités fructueuses.

1) Activités menées

Durant toute la période de notre stage, notre travail fut essentiellement la conception et la mise en place de la plateforme web d'e-commerce. Mais parallèlement aux activités liées à notre projet de stage nous avons eu à travailler sur la réalisation d'un autre projet au compte de AMITECH à savoir le site web de l'entreprise.

2) Connaissances acquises

Ce stage nous a été très bénéfique. Ainsi, il nous a permis entre autres de :

- mettre en pratique les connaissances académiques engrangées durant les trois (03) ans de formation à l'IBAM notamment en développement web et génie logiciel ;
- renforcer nos compétences en développement d'applications web ;
- s'imprégner des réalités de la vie professionnelle ;
- connaître l'écosystème autour du commerce en ligne et leur implémentation ;
- démystifier l'utilisation et l'intégration des solutions de paiement.

II. Observations, suggestions et perspectives

Dans cette partie, nous présentons nos observations et nos suggestions que nous avons notées par rapport à ce stage de trois mois passé au sein de AMITECH et durant les 3 dernières années passées à l'IBAM.

1) Structure de formation

Nous présentons ici nos observations et nos suggestions que nous avons notées durant les 3 dernières années passées à l'IBAM.

a) Observations

A titre d'observation, au compte de l'IBAM, nous avons relevé qu'il y a :

- ✓ bases solides reçues en algorithmique et en programmation qui ont été utiles dans le cadre de ce stage
- ✓ utilisation, dans le cadre du stage, de certaines technologies pas enseignées dans les formations
- ✓ absences de cas pratiques de réalisation de site web de A à Z (c'est-à-dire de l'écriture de la première ligne de code jusqu'au déploiement sur serveur d'hébergement en ligne) ce qui peut causer un ralentissement à un certain niveau du planning de réalisation des solutions web.

b) Suggestions

Au regard de ces observations, nous avons quelques suggestions qui pourraient être utiles à notre institut et qui sont :

- dispenser des cours moins théoriques avec la réalisation de projets pratiques de A à Z et avec des technologies d'actualité;
- mettre à jour les modules techniques (programmation, plateforme de programmation, etc) de la formation ;

2) Structure d'accueil

Nous présentons ici nos observations et nos suggestions que nous avons notées par rapport à ce stage de trois mois passé au sein de AMITECH.

a) Observations

A titre d'observation nous avons relevé au compte de la structure d'accueil:

- un accueil chaleureux de la part des agents de l'entreprise ;
- un cadre agréable et convivial pour le travail ;
- des services essentiellement axés sur l'électronique ;
- un personnel en matière de développement logiciel insuffisant.

b) Suggestions

Vu les observations précitées, nous avons quelques suggestions qui pourraient être utiles à notre structure d'accueil et qui sont :

- développer un service de génie logiciel afin de faciliter la création de solutions performantes issues de la combinaison informatique et électronique ;
- recruter plus de développeurs pour travailler sur la plateforme de vente afin de la rendre rapidement très sophistiquée avec beaucoup de services connexes.
- développer un service de marketing entier en vue de faire la promotion de leurs produits dans ce secteur très concurrentiel.

c) Perspectives

La nouvelle plateforme de AMITECH représente un potentiel énorme. Ainsi comme perspective à l'endroit de AMITECH nous pouvons suggérer de :

- créer des espaces vendeurs personnalisés sur la plateforme et ouvrir l'application aux autres vendeurs du même secteur afin d'avoir une application très utilisée ;
- créer une version mobile de la plateforme afin de profiter du nombre important d'utilisateurs de smartphones ;

Nous venons dans ce chapitre de faire le bilan de notre stage. Nous avons présenté son déroulement, les activités menées ainsi que nos acquis. Nous avons par ailleurs fait ressortir nos observations, suggestions et perspectives et nous espérons qu'elles seront prises en compte par notre structure d'accueil.

CONCLUSION

En somme, nous avons effectué trois (3) mois de stage au cours desquels nous avons travaillé sur la conception et de la réalisation d'applications web et principalement sur la conception d'une plateforme de vente en ligne.

Dans les étapes du processus de développement de la plateforme nous avons eu à utiliser divers outils et méthodes d'analyse et de conception. Ainsi dans une première partie nous avons fait une étude préalable ou nous avons utilisé XUP comme processus de développement et UML comme langage de modélisation du futur système. Dans une seconde partie nous avons fait une étude des spécifications fonctionnelles et techniques. Et nous avons terminé par la réalisation du système après avoir présenté en troisième partie la conception globale du système.

Ce stage nous a vraiment été bénéfique sur plusieurs plans. D'un point de vue technique il nous a permis de nous perfectionner en matière de développement web avec l'utilisation d'outils divers. Ainsi nous avons été démystifiés quant à l'utilisation de certaines pratiques nouvelles dans le web à savoir l'intégration de solutions de paiement en ligne et l'utilisation d'API (interface de programmation d'applications) provenant de grandes plateformes comme Google et Facebook. Aussi d'autre part, il nous a permis de prendre connaissance du monde professionnel et de mettre en pratique nos connaissances théoriques solides acquises au cours de ces trois ans passés au sein de l'IBAM. Ce fut l'occasion également d'apprécier la formation reçue de l'IBAM qui semble être de bonne qualité de manière générale et selon notre appréciation personnelle.

A la fin de notre stage nous avons pu produire notre rapport de stage, renforcer nos compétences techniques. La plateforme qui a été réalisé, est hébergée et accessible en ligne mais en version bêta. Nous recommandons à AMITECH d'en faire la publicité dès que possible afin de s'intégrer rapidement parmi les leaders du e-commerce dans le pays et hors de ses frontières.

BIBLIOGRAPHIE ET WEBOGRAPHIE

I. Bibliographie

1. ZORE, M. (2018). Mise en place d'un système informatisé de gestion des immobilisations du trésor public. Rapport de stage : Méthodes Informatiques Appliquées à la GEstion. Ouagadougou : Institut Burkinabé des Arts et Métiers, 71 p.
2. KABRE, JW. (2018). Mise en place d'une plate-forme d'informations climatiques pour l'ANAM. Rapport de stage : Méthodes Informatiques Appliquées à la GEstion. Ouagadougou : Institut Burkinabé des Arts et Métiers, 67 p.
3. SAWADOGO, EW. (2017). Conception et réalisation de la plateforme GAF (Gouvernance Accès Facile). Rapport de stage : Méthodes Informatiques Appliquées à la GEstion. Ouagadougou : Institut Burkinabé des Arts et Métiers, 71 p.
4. AMITECH. (2019). Présentation de la société AMITECH. Ouagadougou. 2 p.

II. Webographie

1. Buleté. ARCHITECTURE MVC [en ligne]. (Page consultée le 20/09/2019). <https://walkerspider.com/cours/laravel/architecture-mvc>
2. Emourgeon. Développement web dlm3. [en ligne]. (Page consultée le 23/09/2019). <https://he-arc.github.io/slides-devweb/03-Laravel.html#6.0>
3. grafikart. Module de paiement Stripe [en ligne]. (Page consultée le 21/09/2019). <https://www.grafikart.fr/tutoriels/stripe-802>
4. guru99. DBMS Architecture: 1-Tier, 2-Tier & 3-Tier [en ligne]. (Page consultée le 20/09/2019). <https://www.guru99.com/dbms-architecture.html>
5. Salim CHEHIDA. Proposition d'un processus de développement pour la modélisation sécurisée des systèmes d'information [en ligne]. (Page consultée le 15/09/2019). <https://pdfs.semanticscholar.org/3828/cd13e0866dfd0b2193035cc30704c164a25a.pdf>
6. socialcompare. Comparison of Web servers [en ligne]. (Page consultée le 01/10/2019). <http://socialcompare.com/fr/comparison/comparison-of-web-servers>
7. stripe. stripe [en ligne]. (Page consulté le 29/09/2019). <https://stripe.com/fr-us>
8. stripe. stripe docs [en ligne]. (Page consultée le 29/09/2019). <https://stripe.com/docs/security/stripe>

TABLE DES MATIERES

SOMMAIRE	i
DEDICACE.....	iii
REMERCIEMENTS	iv
LISTE DES SIGLES ET ABREVIATIONS	v
LISTE DES FIGURES GRAPHIQUES	vi
LISTE DES TABLEAUX	vii
INTRODUCTION GÉNÉRALE.....	1
CHAPITRE 1 : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL	2
I. Présentation de la structure de formation (IBAM)	2
1) Historique	2
2) Objectif.....	2
3) Organisation	2
4) Filières de formation	3
II. Présentation de la structure d'accueil (AMITECH)	4
1) Historique	4
2) Objectifs (Attributions)	5
3) Organisation	5
CHAPITRE 2 : ANALYSE ET CONCEPTION	7
I. Etude Préalable.....	7
1) Présentation du Thème	7
a) Contexte.....	7
b) Problématique	7
c) Objectifs et attentes	8
2) Méthode d'analyse et de conception	8

a)	Processus de développement (XUP).....	9
b)	Langage de modélisation (UML)	10
3)	Groupe de travail	11
a)	Groupe de pilotage.....	11
b)	Groupe de projet	11
c)	Groupe des utilisateurs	11
4)	Planning de réalisation	12
II.	Expression des besoins.....	13
1)	Spécification fonctionnelle.....	13
a)	Identification des acteurs et cas d'utilisation.....	13
b)	Description de quelques cas d'utilisation	15
c)	Diagramme de cas d'utilisation	18
d)	Diagrammes de séquences vues de l'extérieur	19
e)	Diagrammes d'activités	23
2)	Spécification technique	25
a)	Mise à disposition des conditions de travail	25
b)	Architecture de développement	25
III.	Conception globale	26
1)	Diagramme de classes	27
a)	Dictionnaire de données	27
b)	Diagramme de classes	28
2)	Diagrammes de séquences vues de l'intérieur	30
3)	Diagramme de déploiement	32
IV.	Réalisation.....	32
1)	Choix des outils de réalisation	33
a)	Système de Gestion de Base de Données	33

b)	Serveur d'application.....	34
c)	Langages de programmation	35
d)	Outils de conception	35
e)	Plateformes de développement.....	36
2)	Présentation de l'architecture MVC de notre application	37
3)	Présentation de quelques maquettes IHM de notre application	37
a)	Page d'accueil (Figure 14).....	38
b)	Page d'authentification (Figure 15).....	38
c)	Le catalogue des produits (Figure 16).....	39
d)	Page de détails sur un produit (Figure 17).....	40
e)	Le panier d'achat (Figure 18).....	41
f)	Page d'administration des produits.....	42
4)	Politique de sécurité	43
5)	Coût de réalisation.....	45
CHAPITRE 3 : BILAN DE STAGE.....		48
I.	Présentation du déroulement du stage et des activités réalisées.....	48
1)	Activités menées	48
2)	Connaissances acquises.....	48
II.	Observations, suggestions et perspectives	48
1)	Structure de formation.....	48
a)	Observations	49
b)	Suggestions.....	49
2)	Structure d'accueil.....	49
a)	Observations	49
b)	Suggestions.....	49
c)	Perspectives	50

CONCLUSION	51
BIBLIOGRAPHIE ET WEBOGRAPHIE	52
I. Bibliographie	52
II. Webographie	52
TABLE DES MATIERES	53
ANNEXES	I
I. ANNEXE 1 : Le langage UML.....	I
II. ANNEXE 2 : La méthode COCOMO.....	II
III. ANNEXE 3 : L'utilisation et le fonctionnement de Stripe	III
1) Fonctionnement.....	III
2) Tarification	III
3) L'API	III
4) Les webhooks	V
IV. ANNEXE 4 : Le processus de développement en X (Tactiques de progression)..	VI

ANNEXES

I. ANNEXE 1 : Le langage UML

UML est l'acronyme anglais pour « Unified Modeling Language ». Il se traduit par « Langage de modélisation unifié ». C'est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée permettant de visualiser la conception d'un système. Ce langage est utilisé pour la spécification, la visualisation, la modification et la construction des documents nécessaires au bon développement d'un logiciel orienté objet. Il offre un standard de modélisation, pour représenter l'architecture logicielle. Grâce à UML, il est possible de générer tout ou une partie du code d'un logiciel à partir des divers documents réalisés.

Ce langage de modélisation nous offre principalement 13 diagrammes (depuis sa deuxième version) pour modéliser un système. Ces diagrammes peuvent être utilisés selon la phase du développement d'un logiciel.

En analyse, nous pouvons utiliser des :

- Diagrammes de cas d'utilisation : modélisent les besoins des utilisateurs ;
- Diagrammes de séquences vue de l'extérieur : présentent les scénarios entre les utilisateurs ;
- Diagrammes d'activités : c'est un enchainement d'actions représentant un comportement du logiciel.

En phase de conception, le développeur peut utiliser des :

- Diagrammes de classes : pour représenter la structure interne du logiciel ;
- Diagrammes d'objets : pour présenter l'état interne du logiciel à un instant donné ;
- Diagrammes d'états-transitions : pour présenter l'évolution de l'état d'un objet ;
- Diagrammes de séquence vue de l'intérieur : pour montrer les scénarios d'interactions avec les utilisateurs au sein du logiciel ;
- Diagrammes de composants : pour présenter les composants physiques du logiciel ;
- Diagrammes de déploiement : pour l'organisation matérielle du logiciel.

Ces diagrammes sont rarement tous implémentés dans le cadre du même projet. Le choix des diagrammes à mettre en œuvre dans la modélisation est généralement fonction de la nature du projet et de sa taille.

II. ANNEXE 2 : La méthode COCOMO

Un grand nombre de méthodes est mis à la disposition des développeurs pour estimer le coût de leurs plateformes. Nous utiliserons la méthode Constructive Cost Model (COCOMO) pour l'estimation du cout total du développement (CTDEV) de notre application du fait de sa fiabilité. De plus, cette méthode permet également d'estimer le temps de développement (TDEV) du système correspondant au temps requis pour terminer le projet avec toutes les ressources disponibles. La méthode COCOMO se base principalement sur la complexité de l'application à développer qui correspond à l'un des trois (03) types suivants :

- S : ce sont des applications simples, n'ayant que peu de cas particuliers et de contraintes. Elles sont parfaitement déterministes.
- P : ce sont des applications intermédiaires, plus complexes que les applications de type S. Elles restent tout de même déterministes bien que le nombre de leurs cas particuliers et de tests soit plus important que pour les applications de type S.
- E : ce sont des applications très complexes, que ce soit au niveau de leurs contraintes, comme un système temps réel, où au niveau des données saisies, comme certaines interfaces graphiques ou on ne peut envisager toutes les possibilités de saisies qu'un utilisateur pourrait effectuer. Elles ne sont pas déterministes.

La formule de calcul COCOMO est présentée dans le Tableau 10.

Complexité	Effort(en Homme mois)	Temps de développement(TDEV en mois)
S	$\text{Effort} = 2,4 * \text{KLS}^{1,05}$	$\text{TDev} = 2,5 * \text{Effort}^{0,38}$
P	$\text{Effort} = 3 * \text{KLS}^{1,12}$	$\text{TDev} = 2,5 * \text{Effort}^{0,35}$
E	$\text{Effort} = 3,6 * \text{KLS}^{1,2}$	$\text{TDev} = 2,5 * \text{Effort}^{0,32}$

Tableau 10:Formule de calcul COCOMO

NB : HM est le nombre d'« homme mois » nécessaire à la réalisation du projet, et KLS est le nombre de Kilo Lignes Sources. Un homme mois correspond à 152 heures de travail effectif. Le nombre de personnes requis pour réaliser le projet dans cet intervalle de temps est donc : $N = \text{HM}/\text{TDEV}$. Etant donné que le salaire moyen d'un informaticien est de X FCFA, le coût total de développement pour ce projet est : $\text{CTDEV} = \text{HM} * X$.

III. ANNEXE 3 : L'utilisation et le fonctionnement de Stripe

Stripe est une plateforme qui vous permet de mettre en place un système de paiement sur votre site simplement grâce à une API facile à prendre en main et avec une administration facile.

1) Fonctionnement

Stripe va permettre de mettre en place le formulaire de paiement directement sur notre site. L'utilisateur ne sera pas redirigé vers une page intermédiaire pour rentrer ses informations bancaires. Le traitement des informations bancaires va être géré par stripe qui vous renverra un token de paiement à la place. Vous n'aurez pas à stocker d'informations sensibles.

2) Tarification

Niveau tarification le principe est à peu près le même que PayPal, un pourcentage par transaction en plus d'une somme de base par transaction :

- 1,4 % + 0,25 € pour les cartes européennes.
- 2,9 % + 0,25 € pour les cartes non européennes.
- 15 € en cas de contestation de paiement (frais qui seront remboursés si la banque tranche en votre faveur).

3) L'API

Pour commencer il faudra donc récupérer les informations bancaires entrées par l'utilisateur et obtenir un Token en contactant l'API de stripe. Illustration à la Figure 20.

```
<script type="text/javascript" src="https://js.stripe.com/v2/"></script>
<script>
Stripe.setPublishableKey('pk_test_Et9GdgxRPH9oBRXGCfMKX6RK');

var $form = $('#payment-form'); // On récupère le formulaire
$form.submit(function (e) {
  e.preventDefault();
  $form.find('button').prop('disabled', true); // On désactive le bouton submit
  Stripe.card.createToken({
    number: $('.card-number').val(),
    cvc: $('.card-cvc').val(),
    exp_month: $('.card-expiry-month').val(),
    exp_year: $('.card-expiry-year').val()
  }, function (status, response) {
    if (response.error) { // Ah une erreur !
      // On affiche les erreurs
      $form.find('.payment-errors').text(response.error.message);
      $form.find('button').prop('disabled', false); // On réactive le bouton
    } else { // Le token a bien été créé
      var token = response.id; // On récupère le token
      // On crée un champs cachée qui contiendra notre token
      $form.append($('
```

Figure 20: Récupérer des informations bancaires en JavaScript (grafikart. Module de paiement Stripe [en ligne]. (Page consultée le 21/09/2019).
<https://www.grafikart.fr/tutoriels/stripe-802>)

Ce token sera donc transmis au serveur et pourra être utilisé pour créer un customer (on peut aussi l'utiliser pour déclencher un paiement directement, mais conserver l'utilisateur sur Stripe permet une gestion plus facile par la suite). Illustration à la Figure 21.

```
\Stripe\Stripe::setApiKey("sk_test_MjzqdAm0H5j9rBy6Ip27pCu6");  
$customer = \Stripe\Customer::create(array(  
    "description" => "Customer for daniel.white@example.com",  
    "email" => "daniel.white@example.com",  
    "source" => "tok_18imUxDZd8whA19qRiYt8KNb" // obtenu via l'étape précédente  
));
```

Figure 21: Enregistrement des informations du token (grafikart. Module de paiement Stripe [en ligne]. (Page consultée le 21/09/2019). <https://www.grafikart.fr/tutoriels/stripe-802>)

Lors de cette étape on conservera surtout ID du customer, cet ID peut ensuite être utilisé pour prélever l'utilisateur. Code illustré à la Figure 22.

```
$charge = \Stripe\Charge::create(array(  
    "customer" => 2000, // En centimes ! 20 € ici  
    "currency" => "eur",  
    "customer" => "cus_90q2whhB99KsRq"  
));
```

Figure 22: Prélèvement de la somme à travers le token (grafikart. Module de paiement Stripe [en ligne]. (Page consultée le 21/09/2019). <https://www.grafikart.fr/tutoriels/stripe-802>)

4) Les webhooks

Dans le cas des abonnements, il sera important de capturer les nouveaux paiements ainsi que les paiements échoués. Pour cela il sera possible de créer des webhooks qui seront appelés par Stripe lors d'évènements particuliers. Code illustré à la Figure 23.

```
<?php  
http_response_code(200); // Il faut renvoyer un code 200 pour indiquer à Stripe qu  
// ...  
$input = file_get_contents('php://input');  
$received_event = json_decode($input); // On récupère l'évènement envoyé  
// On peut ici mettre un filtre pour ne faire un traitement que pour certains évèn  
$event = \Stripe\Event::retrieve($received_event->id); // On demande l'évènement à
```

Figure 23: Illustration de l'implémentation des webhooks (grafikart. Module de paiement Stripe [en ligne]. (Page consultée le 21/09/2019). <https://www.grafikart.fr/tutoriels/stripe-802>)

IV. ANNEXE 4 : Le processus de développement en X (Tactiques de progression)

XUP passe par un ensemble d'étapes successives de plus en plus détaillées, une telle progression organise le volume des informations collectées et définit les objectifs à atteindre pour chaque étape suivant un découpage en niveaux de détail croissants.

La Figure 24 suivante montre les différents niveaux d'avancement du processus en X.

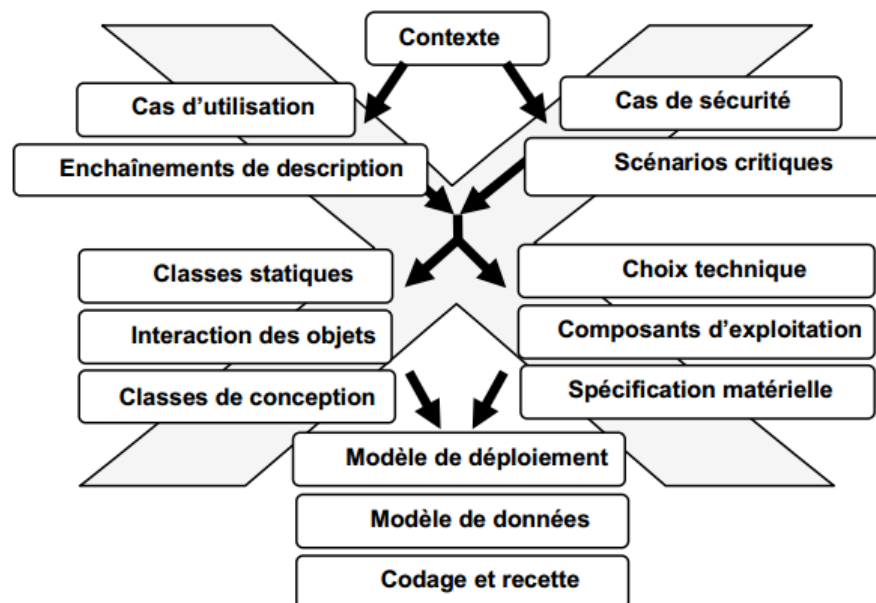


Figure 24: Niveaux de progression du processus en X (Salim CHEHIDA. Proposition d'un processus de développement pour la modélisation sécurisée des systèmes d'information [en ligne]. (Page consultée le 15/09/2019).

<https://pdfs.semanticscholar.org/3828/cd13e0866dfd0b2193035cc30704c164a25a.pdf>

- Pour la phase de recueil initial de besoins :

Le niveau « contexte » consiste à définir la frontière fonctionnelle et les différents services de sécurité attendus du système considéré comme une boîte noire.

- Pour la phase de spécification des besoins fonctionnels :

Le niveau « cas d'utilisation » a pour objet de montrer les différentes possibilités d'utilisation du système à partir du modèle de contexte afin d'identifier le comportement de ce système sans spécifier sa structure interne. Les cas d'utilisation permettent aussi de forcer l'utilisateur à définir ce qu'il attend du système. Les « enchaînements de description » consistent à établir une description des cas d'utilisation identifiés au niveau précédent, en présentant l'ensemble des interactions entre les acteurs et le système considéré comme une boîte noire. La description des

cas d'utilisation est indispensable, car elle permet de communiquer facilement et de manière précise avec les utilisateurs.

➤ Pour la phase de spécification des contraintes de sécurité :

Le niveau « cas de sécurité » permet de définir les services de sécurité fournis par le système (toujours envisagé comme une boîte noire) afin de répondre aux différentes exigences de sécurité identifiées au niveau contexte. Les cas de sécurité sont absolument distincts des cas d'utilisation ; ils ne produisent pas une valeur ajoutée fonctionnelle mais ils recouvrent en effet tout service de sécurité dont un utilisateur bénéficie. Par exemple : Assurer l'authentification d'un utilisateur, Assurer l'intégrité et la confidentialité des informations échangées, Assurer la non-répudiation d'une transaction, ...

Les « scénarios critiques » consistent à décrire les interactions ou les actions qui incluent un risque en mettant en jeu les différents services ou propriétés de sécurité spécifiées par les cas de sécurité. Par exemple : les scénarios qui permettent d'assurer la non-répudiation dans les transactions; il garantit que si une action est exécutée, elle ne peut pas être niée (elle est prouvée), les scénarios qui permettent d'assurer un échange équitable lors d'une transaction, les scénarios qui spécifient les interactions permettant l'échange des informations critiques (nécessite la confidentialité et l'intégrité)

➤ Pour la phase d'analyse et de conception :

Les « classes statiques » sont identifiées à partir des concepts métier extraits des scénarios de description des cas d'utilisation et des scénarios critiques. Ces concepts seront ensuite formalisés sous forme de classes et d'associations rassemblées dans un diagramme statique. Après le développement du modèle statique, nous remplaçons le système, considéré comme une boîte noire lors de la spécification, par un ensemble d'objets. Le niveau « interaction » consiste donc à représenter la collaboration entre ces objets à partir des scénarios décrivant l'exécution des cas d'utilisation ou de sécurité. Le niveau « classe de conception » consiste à optimiser les modèles statiques en affinant les classes, les associations et les attributs, et en ajoutant les opérations identifiées grâce à l'analyse dynamique des scénarios d'interaction. Dans le modèle de conception, on doit spécifier les propriétés de sécurité sur les données en profitant de la spécification des contraintes de sécurité.

➤ Pour la phase d'architecture technique :

Le niveau « choix technique » permet de spécifier les prérequis et les stratégies techniques, ainsi que le style d'architecture en tenant compte des contextes fonctionnels et de sécurité.

Après la spécification des choix techniques, le niveau « composants d'exploitation » définit la façon dont sera organisé les différents composants du système (Base de données, Serveurs, Applications, ...).

Le niveau « spécification matérielle » permet de définir l'organisation physique en termes de machines connectées par des moyens divers. En effet, les machines et les connexions sont toutes en relation directe avec les composants d'exploitation. Le modèle matériel doit intégrer les dispositions de prévention pour répondre aux contraintes de sécurité.

➤ Pour la phase Intégration, codage et tests :

Le « modèle de déploiement » permet de représenter les postes de travail, la répartition des fonctions du système, ainsi que la localisation des composants d'exploitation sur le réseau physique. Le « modèle de données » permet la conception du stockage des données en étudiant sous quelle forme les instances sont sauvegardées sur un support physique. Le niveau « Codage et recette » permet de produire et tester les unités de code puis de valider les fonctions de système développé.