



INSTITUT AFRICAIN D'INFORMATIQUE
Représentation du TOGO
(IAI-TOGO)
07 BP 12456 Lomé 07, TOGO
Tél : (+228) 22 20 47 00
E-mail : iaitogo@iai-togo.tg / iaitogo@yahoo.fr
Site Web: www.iai-togo.tg



Management & IT Consulting
Tél: (+228) 22 43 06 12 / (+ 228) 93 21 82 38
Fax: 00 228 22 43 07 15
TicBuilder - SARL - www.tic-builder.com

RAPPORT DE STAGE PRATIQUE EN ENTREPRISE

Type de stage : Génie Logiciel

Plateforme Web d'achat de produits en ligne

Du 11 Juin au 11 Août 2018

Rédigé et soutenu par :

NIKABOU Siba Fidèle

Etudiante en 2ème année Tronc Commun

Année universitaire 2017-2018

MAITRE DE STAGE :

M. Rodrigue VOKOU
Ingénieur – Analyste - Programmeur

SUPERVISEUR :

M. Kodjo AGBETI
Enseignant à l'IAI-TOGO

Sommaire

INTRODUCTION	2
Partie I : Rapport d'insertion	
I. PRESENTATIONS	4
II. THEME DE STAGE	8
Partie II : Rapport de pré-programmation	
I. ETUDE DE L'EXISTANT	11
II. CRITIQUES DE L'EXISTANT	11
III. PROPOSITIONS DE SOLUTIONS	12
IV. EVALUATION FINANCIERE DES SOLUTIONS	15
V. CHOIX DES SOLUTIONS	19
VI. PLANNING PREVISIONNEL DE REALISATION	21
VII. PRESENTATION DE LA METHODE D'ANALYSE	24
VIII. PRESENTATION DE L'OUTIL DE MODELISATION	27
IX. ETUDE DETAILLEE DE LA SOLUTION	29
Partie III : Réalisation et mise en œuvre	
I. MISE EN ŒUVRE	63
II. PRESENTATION DE L'APPLICATION	69
CONCLUSION	86
➤ Bibliographie	87
➤ Table des matières	88

INTRODUCTION

Depuis la découverte de l'informatique, de nombreuses activités de la vie courante ont été simplifiées. Actuellement, les individus peuvent facilement traiter des informations en se servant des logiciels et des réseaux informatiques. Compte tenu de son évolution, ce système caractérise la majorité des grandes entreprises quel que soit le secteur d'activité.

Nous sommes dans un monde de plus en plus connecté, de plus en plus reliés aux outils informatiques. La technologie est aujourd'hui un élément fondamental pour toute entreprise. L'évolution dans le domaine de l'informatique offre des opportunités pour améliorer les processus de flux de travaux, favoriser l'innovation et acquérir un avantage concurrentiel.

Par conséquent, dans le cadre de la formation des Ingénieurs des Travaux Informatiques, l'Institut Africain d'Informatique représentation du TOGO (IAI-TOGO) a mis en place, pour les étudiants de deuxième (2^{ème}) année, un stage pratique de deux (2) mois en entreprise. Son objectif est la mise en pratique des connaissances en analyse et programmation acquises pendant les deux premières années de formation, et la réalisation d'un projet concret, qui devra fournir une application viable et prête à l'emploi.

A cet effet, nous avons effectué un stage du 11 Juin au 11 Août 2018 à TicBuilder SARL. Notre travail consistait à développer une plateforme web d'achat de produits en ligne, d'où le thème de notre stage : Plateforme web d'achat de produits en ligne.

Notre document sera réparti en trois (03) grandes parties : le rapport d'insertion, le rapport de pré-programmation et la réalisation et mise en œuvre.



PARTIE I : RAPPORT D'INSERTION

I. PRESENTATIONS

1. Présentation de l'IAI-TOGO

L'institut Africain d'Informatique (IAI) est une école supérieure en informatique. La convention portant création de l'institut et les statuts y afférent ont été signés le 29 janvier 1972 à Fort Lamy (actuel N'Djamena) en république du TCHAD.

L'accord de siège entre l'IAI et le GABON (siège de l'Institut) a été signé en janvier 1975. Il est par conséquent un Etablissement Inter-Etats d'Enseignement Supérieur.

En application de la décision du Conseil d'Administration de délocaliser l'IAI, la Représentation du TOGO (IAI-TOGO) a ouvert ses portes le 24 Octobre 2002. IAI-TOGO est situé sur la rue de la Kozah à Nyékonakpoè, logé dans le bâtiment du Centre National d'Etudes et de Traitements Informatique (CENETI) et placé sous la tutelle du Ministère de la Planification du Développement. L'accord d'établissement entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006 à Lomé.

L'Institut Africain d'informatique, représentation du TOGO (IAI-TOGO) propose actuellement les cycles de formation d'Ingénieurs de Travaux Informatiques (Licence professionnelle en informatique), en Génie Logiciel (GL), Système & Réseaux (SR) et Multimédia Technologies Web et Infographie (M-TWI). La durée de ce cycle est de trois (03) ans. Au terme des trois années de formation, les diplômés peuvent poursuivre leurs études supérieures au siège, au GABON ou dans les universités occidentales ou asiatiques (UTBM en FRANCE, Université-Laval du Québec au CANADA, etc.), bref dans le monde entier.

L'IAI-TOGO est joignable à l'adresse suivante :

BP : 07 BP 12456 Lomé 07, TOGO

Tél : (+228) 22 21 27 06 / 22 20 47 00

E-mail : iaitogo@iai-togo.com

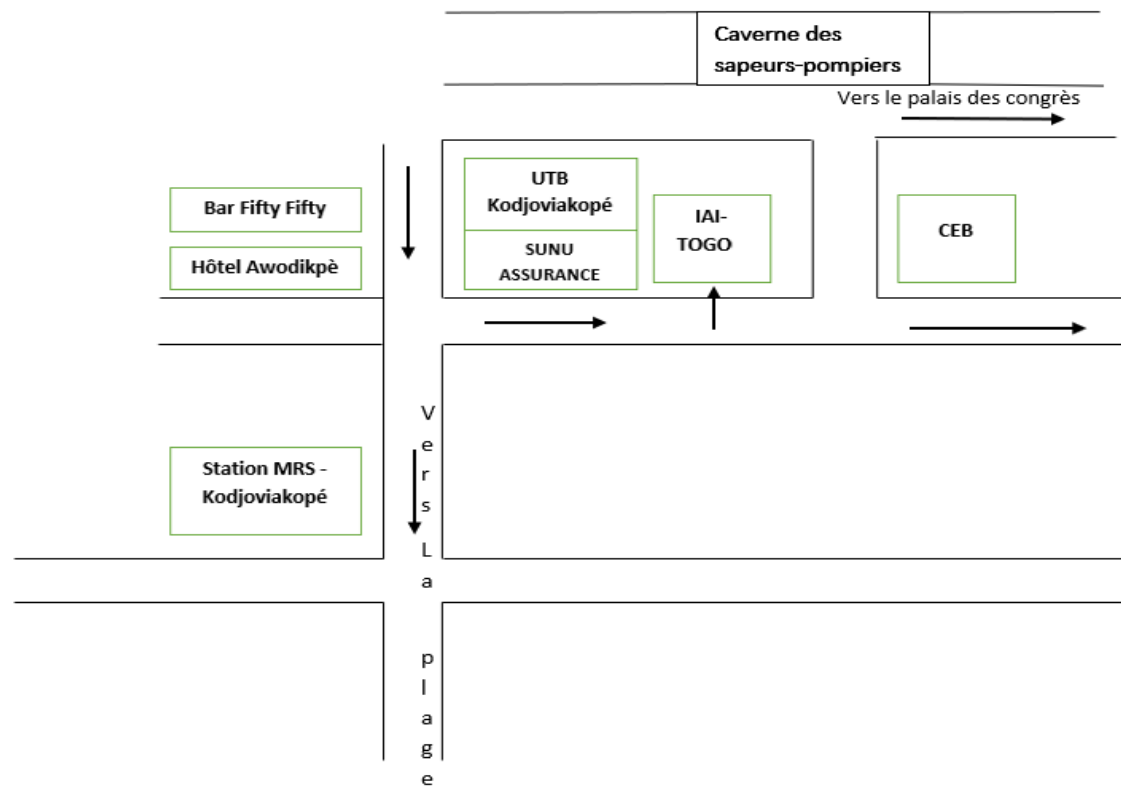


Figure 1: Localisation géographique de l'IAI-TOGO

2. Présentation du cabinet « TicBuilder »

La présentation de notre cadre de stage se subdivisera en sept (05) parties. Elle se fera à travers sa mission, ses activités, ses réalisations, son organigramme et son plan de localisation.

➤ Mission

Elle a pour mission :

- ✓ D'accompagner les entreprises dans la conception de leur système d'information ;
- ✓ D'assurer la pérennité des systèmes d'informations de ces partenaires ;
- ✓ De mettre en place des solutions informatiques adaptées à ces entreprises pour une bonne production ;
- ✓ De promouvoir les Nouvelles Technologies de l'Information et de la Communication ;
- ✓ De proposer et développer des solutions adéquates en vue de répondre aux besoins ressentis sur le plan communautaire ;

➤ Activités

Les principales activités de TicBuilder SARL se basent sur les solutions informatiques telles que :

- Le développement d'applications web et de web services (micro service) ;

- Intégrateur de progiciels (ERP) ;
- Administration de bases de données et la conception d'architecture réseau ;
- Sécurité informatique ;
- Gestion de projet Informatique.

➤ Quelques réalisations

Depuis son émergence, TicBuilder SARL a œuvré pour certaines réalisations informatiques par lui-même et par collaboration. Ces principales sont :

- Audit et conseil ;
- Réalisation d'un outil de gestion commerciale (CRM) ;
- Réalisation d'un outil de gestion des ressources humaines (HRM) ;
- Réalisation d'un outil de Marketing Automatique (SFA) ;
- Prestation web sur Prestashop ;
- Gestion de Projet.
- Des applications mobiles et web pour entreprises

➤ Organigramme

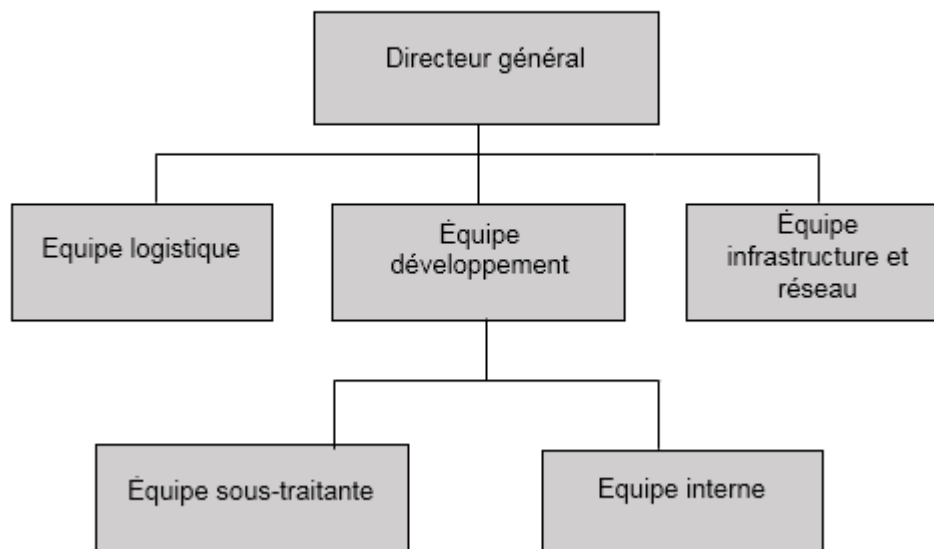


Figure 2: Organigramme de TicBuilder

➤ Plan de localisation

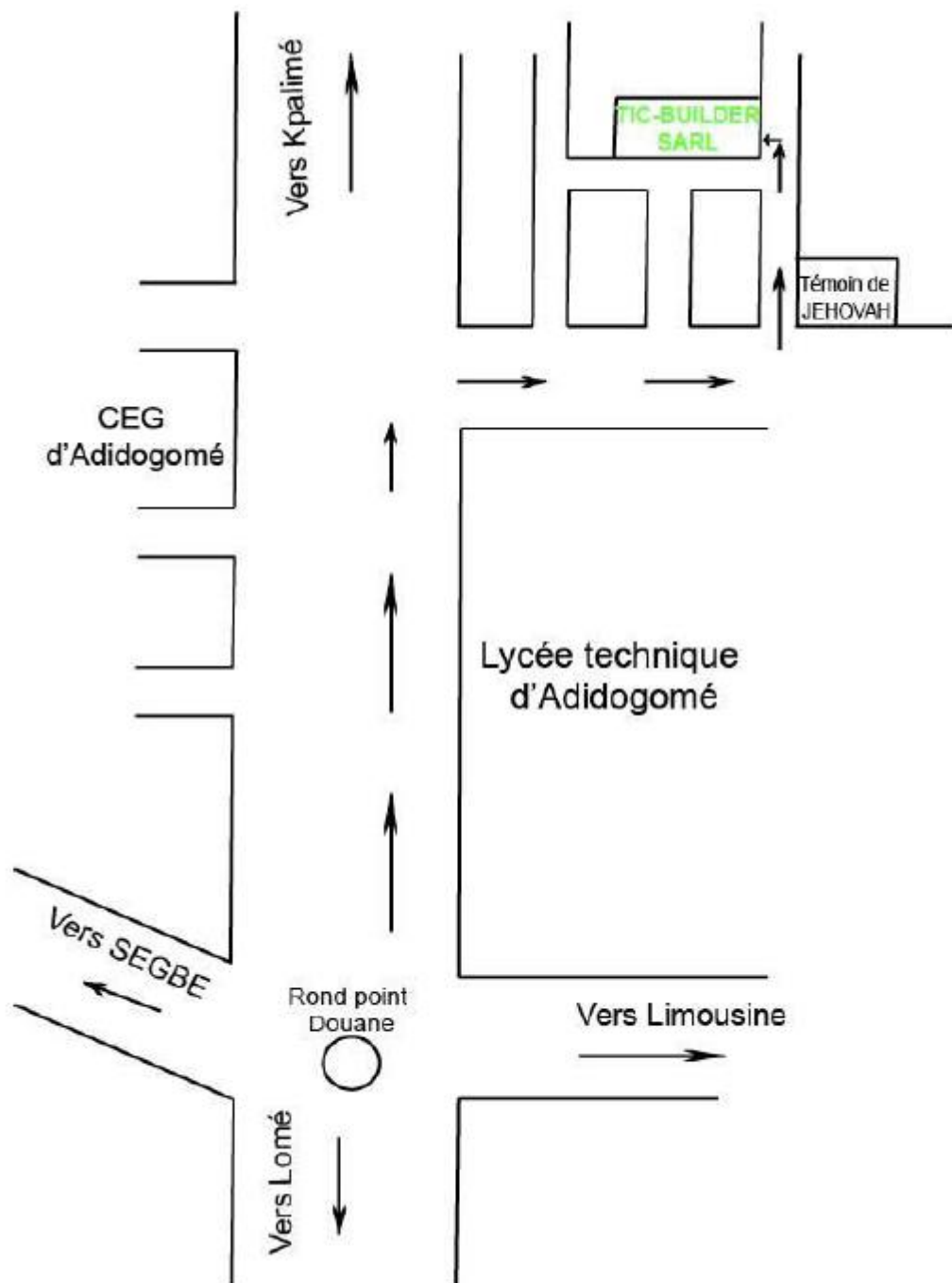


Figure 3: Plan de localisation de TicBuilder

II. THEME DE STAGE

1. Présentation du sujet

L'essor de l'internet a révolutionné les habitudes et les façons de faire dans presque tous les secteurs d'activités. Depuis quelques années, le commerce en ligne connaît un succès impressionnant grâce aux réseaux informatiques. Ceci s'explique sans nul doute par les nombreux possibilités et avantages que propose le commerce électronique ou l'e-commerce.

L'importance du commerce en ligne se lie aisément à travers ses nombreux avantages entre autres la rapidité, le commerce mondial, la vente ciblée.

A cet effet, il s'avère important de disposer d'un site web de vente de produits en ligne, ce que fait notre application développée en Java Play et commercialisé par TicBuilder.

2. La problématique du sujet

Dans le domaine commercial, il est primordial de dépenser moins et de gagner gros. En effet, jadis le traitement des données se faisait manuellement, ce qui était lent, ennuyeux, agaçant et sujet a des nombreuses erreurs. Ses multiples problèmes du système manuel et surtout l'évolution scientifique ont conduit à l'émergence d'un système informatique. De là ce dernier s'est développé et a connu une grande expansion a tel enseigne qu'aucun domaine de la vie socio-économique ne peut s'en passer.

De ce fait, notre préoccupation majeure est d'analyser la gestion des ventes des produits d'un magasin. Après nos investigations, les problèmes ci-après ont été relevés :

- Comment pourrions-nous éviter l'attroupement des clients pour les commandes et les livraisons des produits ?
- Comment assurer l'achat effectif des produits ?
- Peut-on avoir la liste des comptes clients et administrateurs ?
- Peut-on être à mesure d'établir une liste des ventes journalières ?
- Le système pourrait-il prendre en charge la gestion des ventes journalières, semestriel, mensuel, trimestriel et annuel ?

Ces sont des interrogations qui suscitent un intérêt certain auxquelles il faudrait impérieusement trouver une solution en vue d'une dynamique au sein de notre application.

3. Intérêt du sujet

a. Objectifs

- Objectif global :

L'objectif global de cette application est de permettre à un client de consulter des catalogues de produits, d'effectuer une commande et de la recevoir.

- Objectifs spécifiques :

Notre application permettra de :

- Créer un compte et y accéder au moment voulu
- Gérer les produits (ajouter, modifier, supprimer)
- Gérer la clientèle
- Gérer les commandes (enregistrer, valider)
- Mettre en place les promotions
- Assurer les livraisons
- L'accessibilité des données doit être indépendante de la position du client

En effet, cette application donne aux internautes la possibilité de s'inscrire, effectuer leurs commandes en ligne, et de recevoir une confirmation immédiate. En plus, les internautes peuvent consulter en ligne le catalogue et toutes ses nouveautés.

b. Résultats attendus

- Les comptes sont créés et sont accessibles
- Le catalogues des articles est consulté et accessible
- Le panier d'articles est fait par le client
- Le récapitulatif du panier est affiché
- Le choix du mode de livraison est fait par le client
- Le choix du mode de paiement est fait par le client
- Les articles sont commandés et reçus
- Les produits et les catégories sont enregistrés
- Les commandes sont consultées et préparées
- Les factures sont imprimées puis attachées lors de la livraison
- Les commandes des clients sont consultées



PARTIE II : RAPPORT DE PRE-PROGRAMMATION

I. ETUDE DE L'EXISTANT

Pour acheter un produit, le client doit se déplacer directement au local de la société afin de chercher une offre de vente qui satisfait ses besoins. Ses déplacements peuvent être inutiles et même peuvent provoquer un gaspillage de temps.

D'ailleurs, même le vendeur n'a aucun moyen pour mettre à disposition ses annonces de vente et services, à l'exception des supports tels que les journaux ou les petites affiches. Ainsi, un moyen fiable et automatisé permettant d'informer un grand nombre de clients des offres de vente et des services nécessaires. Divers autres traitements sont, d'ailleurs, sources de problèmes, adoptant les méthodes traditionnelles de travail :

- les clients ne sont pas enregistrés ou s'ils le sont, l'enregistrement se fait manuellement sur papier.
- les produits sont classés par catégorie et par sous-catégorie non liées et non hiérarchisées, ce qui rend la recherche plus pénible.
- les documents sont nombreux et mal organisés.

Vu l'accroissement de la technologie Internet, l'achat en ligne est devenu une nécessité incontournable pour les commerçants.

II. CRITIQUES DE L'EXISTANT

Le processus de vente décrit plus haut rencontre plusieurs lacunes :

- D'énormes dépenses : pour assurer la bonne vente de vos produits partout ailleurs, et augmenter votre chiffre d'affaire, vous êtes obligé d'implanter des locaux un peu partout, ce qui vous revient très cher finalement.
- La perte de la clientèle est possible, surtout quand le traitement de leurs demandes traîne ou prend beaucoup de temps pour être livré.
- L'impossibilité d'élaborer des statistiques : Les informations sur les ventes, les dépenses, les incidents... sont traitées manuellement ce qui rend impossible l'élaboration des statistiques sur une longue période.
- L'impossibilité de faire un inventaire correct : Les calculs qui interviennent dans l'inventaire sont faits manuellement et sont donc exposés aux erreurs.
- L'abondance des documents dans l'entreprise qui peut ralentir les services.
- Risque de mélanger les documents ; ce qui peut être fatal.

III. PROPOSITIONS DE SOLUTIONS

Face aux critiques énoncées ci-dessus, trois solutions s'offrent à nous et qui peuvent être envisagées par TicBuilder :

- ✓ Mettre ses produits en vente chez les grands du e-commerce ;
- ✓ Créer un site web via un CMS (Content Management System : désigne une famille d'applications qui ont pour but de créer et de mettre à jour facilement un site web dynamique) et y mettre les produits en vente ;
- ✓ Développer une plateforme web en tenant compte de nouvelles fonctionnalités et des besoins des administrateurs et des partenaires.

A. Première solution

Cette première solution consiste à mettre les produits en vente sur les plateformes de ventes de produits en ligne déjà existants. Nous avons recensé des plateformes tels que :

- Vendre en ligne sur **PriceMinister** sur <https://www.priceminister.com/v/vendre>
- Vendre sur **Amazon** sur <https://services.amazon.fr/services/vendre-sur-internet/avantages.html>

Pour voir les frais et commissions perçues sur les ventes, visitez :
<https://lemag.igraal.com/sites-revente-entre-particuliers/>

Quels sont les avantages et inconvénients de cette solution ?

✓ Avantages

- Gain de temps et d'effort
- La mise en œuvre de cette solution nous permettra de gagner considérablement

en temps de réalisation du projet.

✓ Inconvénients

- Les fonctionnalités spécifiques ne sont pas prises en compte
- Frais de publicité élevés
- Problème de sécurité
- Problème de connectivité à internet

B. Deuxième solution

Cette seconde solution consiste à créer un site web via un CMS (Content Management System : désigne une famille d'applications qui ont pour but de créer et de mettre à jour facilement un site web dynamique) et y mettre les produits en vente. Voici quelques plateformes offrant ce service de création de sites web de ventes de produits en ligne :

- Créer un site avec **PrestaShop** sur <https://www.prestashop.com/en> :

Prestashop permet de se créer un site de commerce en ligne gratuitement

- Créer un site grâce à **Wix** sur <https://fr.wix.com/> :

Wix permet à monsieur et madame tout le monde de se créer un site web ou un site de commerce en ligne gratuitement. Il arrive derrière Prestashop pour la simple raison qu'il ne s'agit pas d'une plateforme spécialisée.

- Créer un site avec **Woo Commerce** sur <https://woocommerce.com/>

Le CMS WordPress est l'un des plus répandus dans le monde. L'une des raisons est qu'il permet à des gens ordinaires de faire des sites aux allures très professionnelles. L'extension Woo Commerce va dans le même sens en permettant la création d'un commerce en ligne sans trop donner mal à la tête.

- Créer le site avec **Shopify** sur <https://www.shopify.ca/>

L'interface est simple et permet d'atteindre d'excellents résultats avec un minimum de complexité.

Quels sont les avantages et les inconvénients de cette solution ?

✓ Avantages

- Gain de temps
- Il n'y a pas de commission à verser sur vos ventes avec PrestaShop.
- Possibilité de changer le thème du site au moment voulu

✓ Inconvénients

- La mise en ligne des produits n'est pas adaptée à nos réalités
- Les fonctionnalités spécifiques ne sont pas prises en compte
- Problème d'insécurité
- Problème de connectivité à internet

C. Troisième solution

Cette solution consiste à créer une plateforme web dynamique et efficace. Quels sont les avantages et inconvénients de cette solution ?

✓ Avantages

- L'application sera une application web donc pourra être hébergée en interne ou en externe
- Une fois déployée, l'application est accessible par un navigateur
- Sécurité adaptée au problème
- Ouverture facile de l'application
- L'application partira d'une analyse donc comportera les fonctionnalités nécessaires et suffisantes
- La maintenance de l'application n'entraîne juste que le redéploiement de la nouvelle version sur le serveur d'application sans perturber les utilisateurs.

✓ Inconvénients

- La plate-forme n'a pas encore été conçue
- Problème de connectivité à internet

IV. EVALUATION FINANCIERE DES SOLUTIONS

On se doit d'estimer les coûts des trois solutions qu'ils soient d'hébergement, humains, matériels ou logiciels et d'en ressortir les coûts totaux.

A- Première solution

La solution qui s'offre à nous est de mettre les produits en vente sur les plateformes de ventes de produits en ligne. Nous évaluons les coûts d'hébergements et matériels.

1. Coût d'hébergement

Cas de chez Amazone

Frais par mois (En Euro)	Coût par mois (FCFA)
39 (*)	25.570
Total	25.570

Tableau 1: Coût d'hébergement de la première solution

(*) Source : <https://services.amazon.fr/services/vendre-sur-internet/comment-ca-marche-pro.html>

2. Coût matériel

Désignation	Exigence	Prix Unitaire	Quantité	Coût (FCFA)
Ordinateur	DELL-PC Core i3 4 Go RAM 500Go	433,06 € (*)	2	283.654,3
Total				283.650

Tableau 2: Coût matériel de la première solution

*Source : <https://www.cdiscount.com>

3. Coût total

Coûts	Totaux
Coût d'hébergement	25.570
Coût matériel	283.650
Total	309.220

Tableau 3: Coût total de la première solution

B. Deuxième solution

La solution qui s'offre à nous est de créer un site web et y mettre nos produits en vente. Nous évaluons les coûts d'hébergements, humains et matériels.

1. Coût d'hébergement

Cas de **OVH** : OVH est une entreprise française spécialisée dans les services de cloud computing. Elle propose des solutions de cloud public et privé, des serveurs dédiés, de l'hébergement mutualisé, du housing (ou colocation), de l'enregistrement de noms de domaines, de la fourniture d'accès Internet.

Frais par mois (en Euro)	Coût (en FCFA)
7,19 (*)	4.714
Total	4.714

Tableau 4: Coût d'hébergement de la deuxième solution

*Source : <https://www.ovh.com/fr/hebergement-web/>

2. Coût humain

Qualité	Nombre	Tâches	Nombre d'heures	Coût horaire	Coût (FCFA)
Analyste-programmeur	2	Maintenance de l'application	424 (8heures x 53 jours)	5000	2.120.000
Total					2.120.000

Tableau 5: Coût humain de la deuxième solution

3. Coût matériel

Désignation	Exigence	Prix Unitaire (en Euro)	Quantité	Coût (FCFA)
Ordinateur	DELL-PC Core i3 4 Go RAM 500Go	433,06 (*)	2	283.654,3
Total				283.650

Tableau 6: Coût matériel de la deuxième solution

*Source : <https://www.cdiscount.com>

4. Coût total

Coûts	Totaux
Coût d'hébergement	4.714
Coût humain	2.120.000
Coût matériel	283.650
Total	2.408.364

Tableau 7: Coût total de la deuxième solution

C. Troisième solution

La dernière solution qui s'offre à nous, est de développer une plateforme web d'achat de produits en ligne pour un magasin donné. Nous évaluons également les coûts humains et matériels de cette solution.

1. Coût d'hébergement

Cas de **OVH**

Frais par mois (en Euro)	Coût (en FCFA)
7,19 (*)	4.714
Total	4.714

Tableau 8: Coût d'hébergement de la troisième solution

*Source : <https://www.ovh.com/fr/hebergement-web/>

2. Coût humain

Qualité	Nombre	Tâches	Nombre d'heures	Coût horaire	Coût (FCFA)
Analyste-programmeur ; programmeur	3	Conception et déploiement de l'application	672 (8 heures x 84 jours)	5.000	3.360.000
Formateur	2	Formation des utilisateurs	28 (2heures x 14jours)	3.000	84.000
Maintenancier	2	Maintenance de l'application	270 (5heures x 54jours)	3000	810.000
Total					4.254.000

Tableau 9: : Coût humain de la troisième solution

3. Coût matériel

Désignation	Exigence	Prix Unitaire	Quantité	Coût (FCFA)
Routeur	TP-LINK	35.000	2	70.000
Souscription à une connexion internet	Au moins 2Mo	25.000	-	25.000
Câbles Ethernet	Blindés ou non	2.000	5	10.000
Total				105.000

Tableau 10: Coût matériel de la troisième solution

4. Coût total

Coûts	Totaux
Coût d'hébergement	4.714
Coût humain	4.254.000
Coût matériel	105.000
Total	4.363.714

Tableau 11: Coût total de la troisième solution

V. CHOIX DES SOLUTIONS

À la suite de l'étude et l'analyse des contraintes et avec l'accord de notre maître de stage, il a été retenu comme solution le développement et le déploiement de la plateforme web.

A. Spécifications fonctionnelles du projet

PROFILS	DROITS
Administrateur	<ul style="list-style-type: none"> ➤ S'authentifier ➤ Sauvegarder et restaurer la base de données ➤ Enregistrer des articles ou produits avec leurs prix ➤ Gérer les comptes ➤ Consulter les commandes ➤ Consulter l'historique des commandes livrées ➤ Consulter le journal des activités
Agent	<ul style="list-style-type: none"> ➤ S'authentifier ➤ Consulter les commandes ➤ Préparer et valider les commandes ➤ Consulter l'historique des commandes déjà livrées ou non ➤ Vérifier la disponibilité des produits ➤ Emettre et imprimer des factures ➤ Valider les livraisons
Livreur	<ul style="list-style-type: none"> ➤ S'authentifier ➤ Valider les livraisons ➤ Consulter l'historique des commandes déjà livrées ou non
Client	<ul style="list-style-type: none"> ➤ Consulter les catalogues de produits ➤ S'authentifier ➤ Faire un panier de produits à acheter ➤ Choisir le mode de réception de ses commandes ➤ Choisir son mode de paiement ➤ Commander les produits et recevoir un code de livraison ➤ Faire un tableau récapitulatif distinct des paiements électroniques et espèces ➤ Consulter l'historique des commandes retirées ainsi que leurs détails

Tableau 12: Spécifications fonctionnelles du projet

B. Spécification de sécurité

La sécurité est prise en compte dans notre application, surtout que cette dernière sera hébergée. Nous prendrons en compte deux niveaux de sécurité : la sécurité du côté de l'hébergeur de l'application et la sécurité de l'application elle-même.

Il faudra donc pour un utilisateur un compte qui lui permettra de s'authentifier avant de faire usage de l'application. Il disposera donc de droits qui lui permettront de réaliser des actions précises.

C. Spécifications techniques

Méthode d'analyse	Approche Objet (UML + 2 TUP)
Outils de modélisation	Sybase PowerAMC 15.1
Système de Gestion de Base de Données	MySQL
Langage de programmation	Java
Environnement de Développement Intégré (EDI)	NetBeans 8.1

Tableau 13: Spécifications techniques du projet

VI. PLANNING PREVISIONNEL DE REALISATION

N°	Activités	Tâches	Prévisions		Réalisation		GAP	Observations
1	Insertion et prise de contact avec le centre d'accueil	Prise de contact avec le centre d'accueil	11-06-2018	12-06-2018	11-06-2018	12-06-2018	-	Tâche effectuée
		Projet test et familiarisation avec le framework Play	13-06-2018	21-06-2018	13-06-2018	21-06-2018	-	Tâche terminée
		Prise en connaissance du thème de stage	25-06-2018		25-06-2018		-	Tâche effectuée
2	Analyse, proposition et discussions sur le thème de stage	Discussions par rapport au thème de stage	25-06-2018	29-06-2018	25-06-2018	28-06-2018	+ 1 jour	Tâche effectuée
		Création de la base de données	29-06-2018	01-07-2018	29-06-2018	30-06-2018	+ 1 jour	Tâche terminée
3	Conception de la maquette	Concevoir l'interface d'accueil et du processus d'achat	02-07-2018	06-07-2018	01-07-2018	11-07-2018	- 3jours	Tâche terminée (à récupérer sur la tâche suivante)
		Concevoir l'interface partenaires	07-07-2018	12-07-2018	12-07-2018	15-07-2018	- 3jours	Tâche terminée (à récupérer sur la tâche suivante)

4	Réaliser le processus d'achat	Permettre au client de consulter les catalogues de produits	17-07-2018	18-07-2018	17-07-2018	17-07-2018	+1 jour	Tâche terminée
		Permettre au client de faire son panier	19-07-2018	20-07-2018	18-07-2018	19-07-2018	+1 jour	Tâche terminée
		Permettre au client de valider sa commande et de suivre le processus d'achat	20-07-2018	22-07-2018	20-07-2018	29-07-2018	-7jours	Tâche terminée (à récupérer sur la tâche suivante)
		Permettre au client de gérer son compte	23-07-2018	24-07-2018	29-07-2018	29-07-2018	-5jours	Tâche terminée (à récupérer sur la tâche suivante)
5	Réaliser la page de gestion pour les agents et les livreurs	Permettre d'effectuer une commande	25-07-2018	27-07-2018	30-07-2018	03-08-2018	-3jours	Tâche terminée
		Permettre une livraison et une facturation	30-07-2018	31-07-2018	30-07-2018	06-08-2018	-5jours	Tâche terminée
6	Déploiement de l'application	Tester l'application	01-08-2018	03-08-2018	06-08-2018	07-08-2018	-5jours	Tâche terminée
		Gérer les insertions et les messages, corriger les erreurs, effectuer toutes les vérifications	03-08-2018	06-08-2018	08-08-2018	13-08-2018	-7jours	Tâche terminée
7	Rédaction du document	Rédiger le document	18-06-2018	03-08-2018	17-06-2018	07-08-2018	-5jours	Tâche effectuée

8	Correction du document	Corriger le document	06-08-2018	10-08-2018	08-08-2018	21-08-2018	-11jours	Tâche effectuée
9	Dépôt du document		20-08-2018		22-08-2018		-2jours	Tâche effectuée

Tableau 14: Planning prévisionnel de réalisation

VII. PRESENTATION DE LA METHODE D'ANALYSE

A. Le langage de modélisation

I- Présentation



Figure 4: Logo d'UML

Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation graphique à base de pictogrammes (dessins figuratifs stylisés ayant fonction de signe) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

II- UML est un formalisme

UML est un langage de conception objet qui permet de modéliser les solutions informatiques par des diagrammes (cités ci-dessus). Pourtant, le langage ne propose pas d'ordre qui doit régir ces diagrammes. C'est ainsi que les méthodes viennent appuyer le langage pour rendre totale et parfaite l'approche objet. Les méthodes imposent au langage l'ordre et la démarche. Il existe une multitude de méthodes de conception objet telles que : 2TUP (Two Track Unified Process), XP (Extreme Programming), Scrum, RUP (Rational Unified Process). La méthode retenue pour notre projet est le 2TUP.

B. Le processus de modélisation

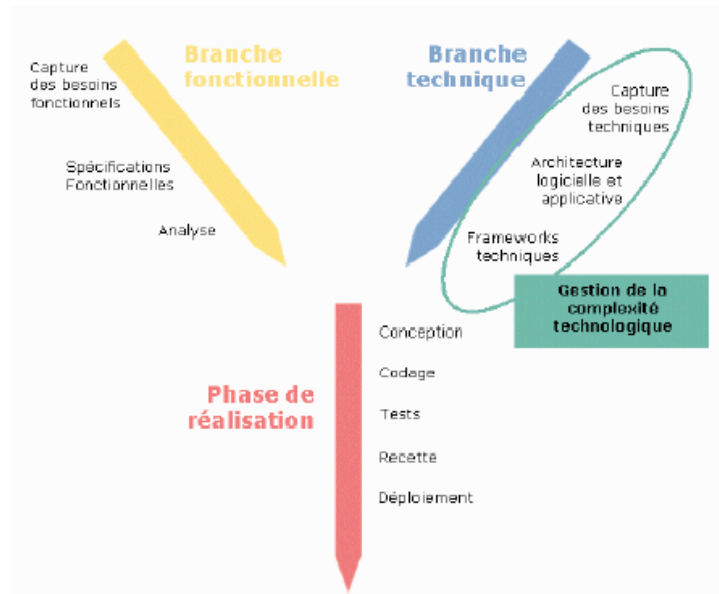


Figure 5: Illustration du 2TUP

2 Track Unified Process (2TUP), à prononcer "toutiyoupi" est un processus de développement Logiciel qui implémente le Processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système. Ensuite à produire le cahier des charges et à modéliser le contexte (le système est une boîte) noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de 3 phases essentielles :

- ✓ Une branche technique ;
- ✓ Une branche fonctionnelle ;
- ✓ Une phase de réalisation.
- La branche technique (droite)
 - La capture des besoins techniques qui recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (prérequis d'architecture technique) ;
 - La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

- La branche fonctionnelle (gauche)
 - Elle capture des besoins fonctionnels qui produisent le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs ;
 - L'analyse qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.
- La phase de réalisation (milieu)

Elle correspond à :

- Une conception préliminaire qui représente une étape délicate car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;
 - La conception détaillée qui étudie comment réaliser chaque composant ;
 - L'étape de codage qui produit ses composants et teste au fur et à mesure les unités de code réalisées ;
- L'étape de recette, qui consiste à valider les fonctionnalités du système développé.

VIII. PRESENTATION DE L'OUTIL DE MODELISATION

Dans la modélisation, on pouvait se contenter d'utiliser uniquement UML, mais sur quels supports ? De nos jours, on ne peut pas se contenter des dessins faits sur papier ni au tableau ; c'est là qu'intervient les logiciels de modélisation qui permettent la réutilisation des modèles, leur portabilité (réalisées une fois et utilisables partout), etc.

L'outil retenu pour la modélisation est : Sybase PowerAMC version 15.1.

A. Qu'est-ce que PowerAMC ?



Figure 6: Logo de PowerAMC

PowerAMC est un environnement graphique de modélisation d'entreprise très simple d'emploi. Il a été créé par la société SDP sous le nom de AMC*Designor, racheté par PowerSoft qui lui-même a été racheté par Sybase en 1995. PowerAMC prend en compte plusieurs outils de modélisation tels UML et MERISE.

PowerAMC permet d'effectuer les tâches suivantes :

- ✓ Modélisation intégrée via l'utilisation de méthodologies et de notation standards :
 - Données (E/R, Merise) ;
 - Métiers (BPMN, BPEL, ebXML) ;
 - Application (UML).
- ✓ Génération automatique de code via des Template personnalisables :
 - SQL (avec plus de 50 SGBD) ;
 - Java ;
 - .NET
- ✓ Fonctionnalités de reverse engineering pour documenter et mettre à jour des systèmes existants ;

- ✓ Une solution de référentiel d'entreprise avec des fonctionnalités de sécurité et de gestion des versions très complètes pour permettre un développement multiutilisateur
- ✓ Fonctionnalités de génération et de gestion de rapports automatisés et personnalisables
- ✓ Un environnement extensible, qui vous permet d'ajouter des règles, des commandes, des concepts et des attributs à vos méthodologies de modélisation et de codage.

B. Modélisation avec PowerAMC

PowerAMC fournit un jeu unique d'outils de modélisation professionnels qui associent les techniques et notations standard de la modélisation de processus métiers, de la modélisation des données et de la modélisation des diagrammes UML et d'autres fonctionnalités puissantes afin d'aider à analyser, concevoir, construire et maintenir des applications, en utilisant les techniques les plus élaborées d'ingénierie logicielle.

La solution de modélisation PowerAMC permet d'intégrer étroitement la conception et la maintenance des couches de données centrales de l'application et exigences de projet, processus métiers, code orienté objet, vocabulaires XML et informations de réplication de base de données.

IX. ETUDE DETAILLEE DE LA SOLUTION

Dans cette partie, nous explorerons l'application d'un point de vue interne et externe. Les éléments des spécifications fonctionnelles y seront détaillés afin d'avoir une vision claire et sans ambiguë de l'application.

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système.

A. Vue de modélisation

Diagramme de contexte statique

Le diagramme de contexte statique délimite le domaine d'étude en précisant ce qui est à la charge du système et en identifiant l'environnement extérieur au système étudié avec lequel ce dernier communique.

Ces composants sont :

- ✓ Les acteurs externes : entités externes au système étudié interagissant avec lui
- ✓ Un processus unique symbolisant le Système Information étudié
- ✓ Echange entre le système étudié et son environnement

Ce diagramme n'est pas officiellement désigné comme diagramme UML. Il ne fait donc pas partie des diagrammes « officiels », mais il est utile pour la description du domaine d'étude avant d'aborder les aspects plus fins.

Le diagramme de contexte statique de notre système est représenté comme suit :

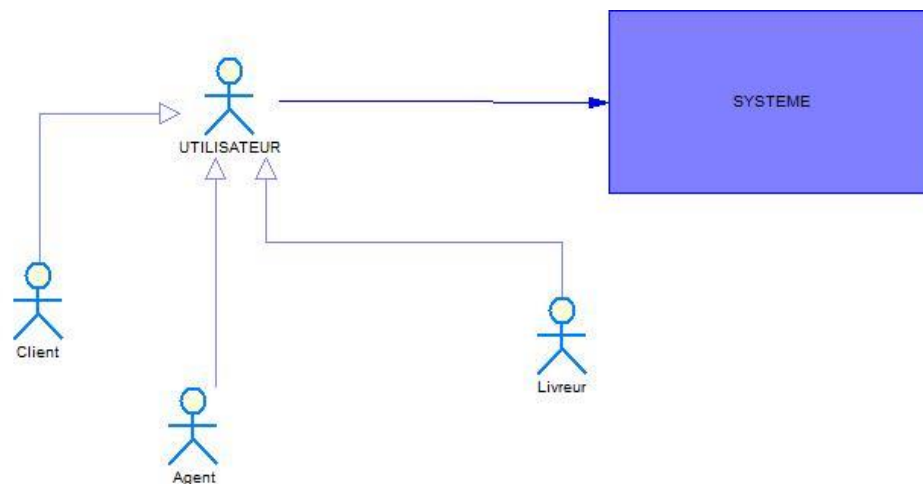


Figure 7: Diagramme de contexte statique de notre système

B. Vue de cas d'utilisation

1. Le diagramme de cas d'utilisation

a. Définition du cas d'utilisation

Un diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation. Il traduit les besoins des utilisateurs vis-à-vis du système développé.

b. Eléments de modélisation des cas d'utilisation

- Les acteurs

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie.

Le diagramme des cas d'utilisation d'UML distingue deux acteurs :

- ✓ Les acteurs principaux (qui modifient l'état du système ou qui consultent cet état)
- ✓ Les acteurs secondaires (acteurs auxquels le système fait appel pour répondre aux sollicitations d'un acteur principal).

Dans notre projet, nous avons décelé uniquement des acteurs principaux qui sont :

- L'administrateur
- L'agent
- Le livreur
- Le client

- Le cas d'utilisation

Un cas d'utilisation exprime le comportement du système en termes d'actions et réactions face à un besoin d'un utilisateur, c'est une fonctionnalité offerte par le système.

Le cas d'utilisation (ou use case) correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des cas d'utilisation décrit les objectifs (le but) du système.

- Les relations entre acteur et cas d'utilisation

Elles expriment l'interaction existant entre un acteur et un cas d'utilisation. Elles lient l'acteur au cas d'utilisation et illustre l'utilisation de la fonctionnalité par l'utilisateur. Elle est représentée par un trait simple.

Le résumé des cas d'utilisations de notre projet est consigné dans le tableau suivant :

Cas d'utilisation		Acteurs
Gérer les catégories de produits	Enregistrer, modifier, supprimer une catégorie ; Ajouter, supprimer un produit dans une catégorie ;	Administrateur
Gérer les produits	Ajouter, supprimer ou modifier des produits dans le magasin	Administrateur
Gérer les clients	Consulter les comptes clients	Administrateur
Valider la livraison, consulter l'historique des commandes		Livreur
Faire un panier de produits à acheter	Choisir les produits dans les catalogues et constituer un panier	Client
Consulter le journal des activités		Administrateur
Gérer les profils	Enregistrer, modifier, supprimer les profils	Administrateur
Gérer les règlements	Enregistrer, consulter les règlements	Administrateur, agent
Gérer les livraisons	Enregistrer, consulter, supprimer une livraison	Agent, administrateur
Gérer les commandes	Enregistrer, supprimer, modifier une commande	Administrateur, agent
Consulter le catalogue d'articles		Client
Passer une commande		Client
S'authentifier		Administrateur, agent, livreur, client, profil

Tableau 15: Les différents cas d'utilisation du système

2. Diagramme de cas d'utilisation par acteur

Il est représenté ici les diagrammes de cas d'utilisation par acteurs identifiés plus haut.

✓ Diagramme de cas d'utilisation de l'acteur agent

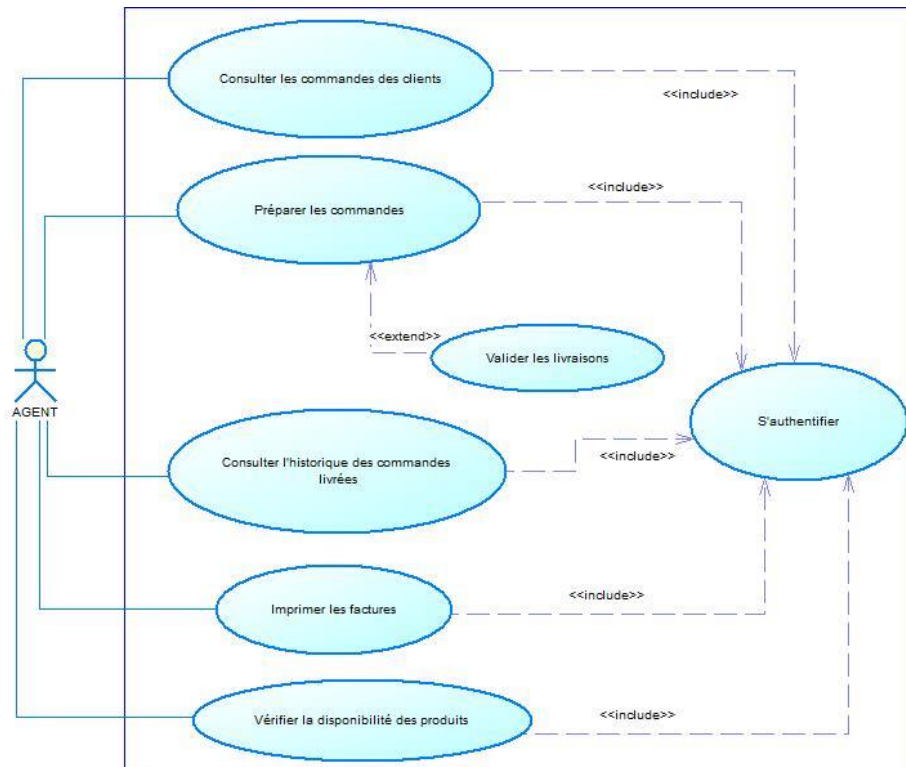


Figure 8: Diagramme de cas d'utilisation de l'acteur agent

✓ Diagramme de cas d'utilisation de l'acteur livreur

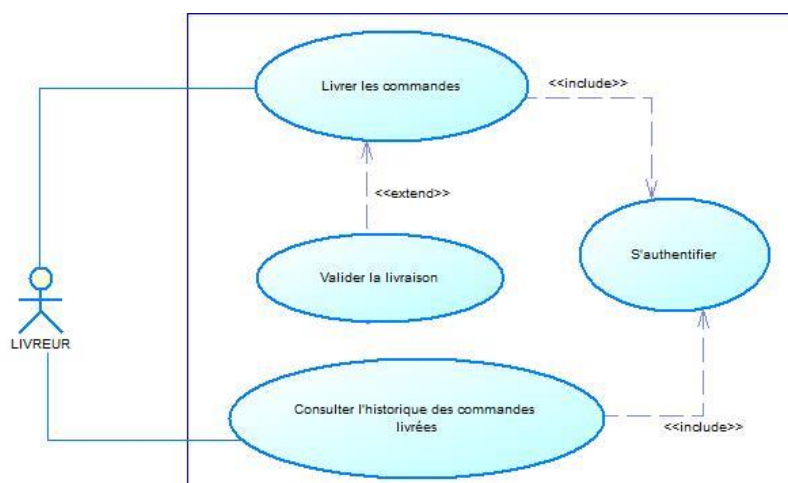


Figure 9: Diagramme de cas d'utilisation de l'acteur livreur

✓ Diagramme de cas d'utilisation de l'acteur client

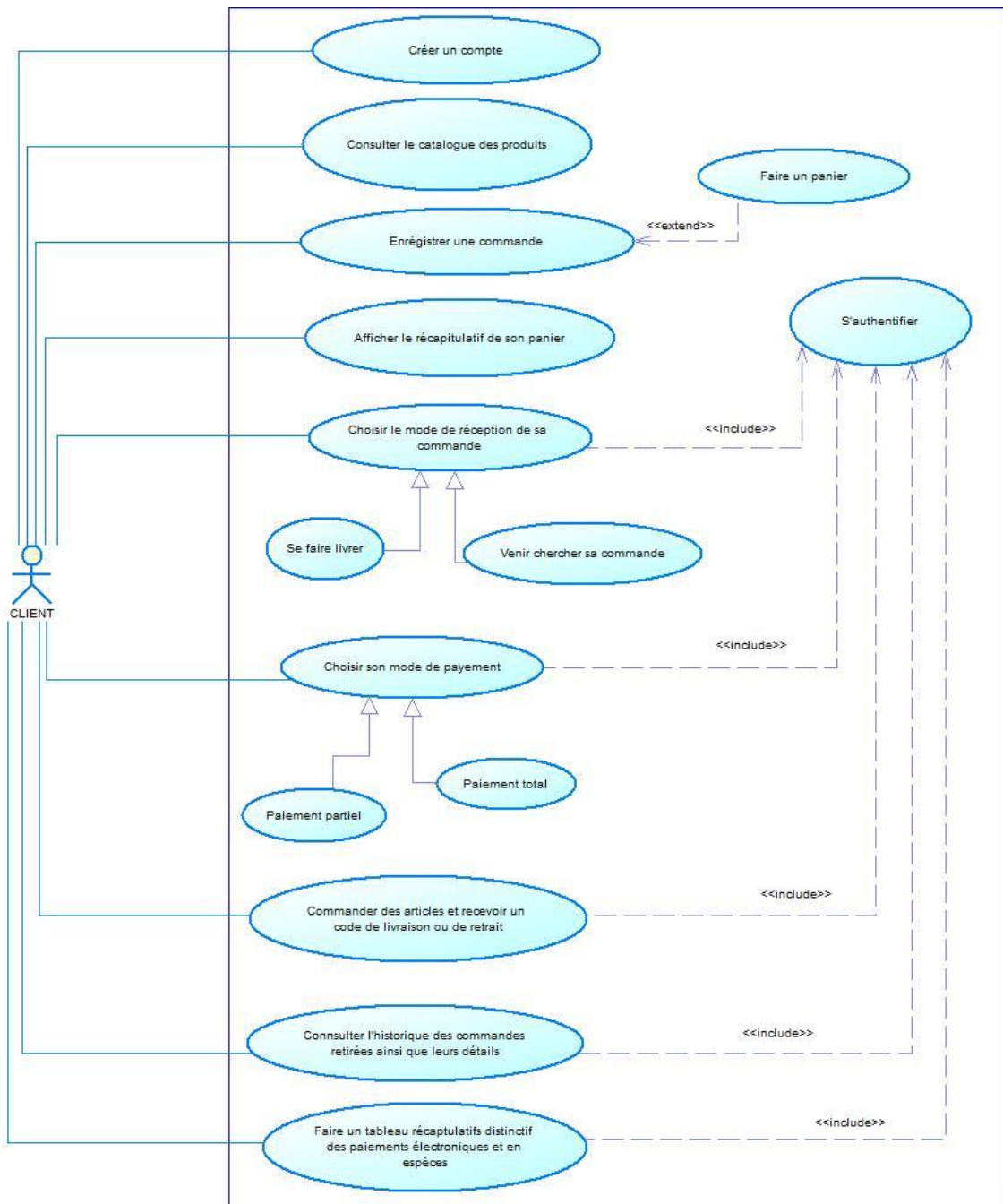


Figure 10: Diagramme de cas d'utilisation de l'acteur client

✓ Diagramme de cas d'utilisation de l'acteur administrateur

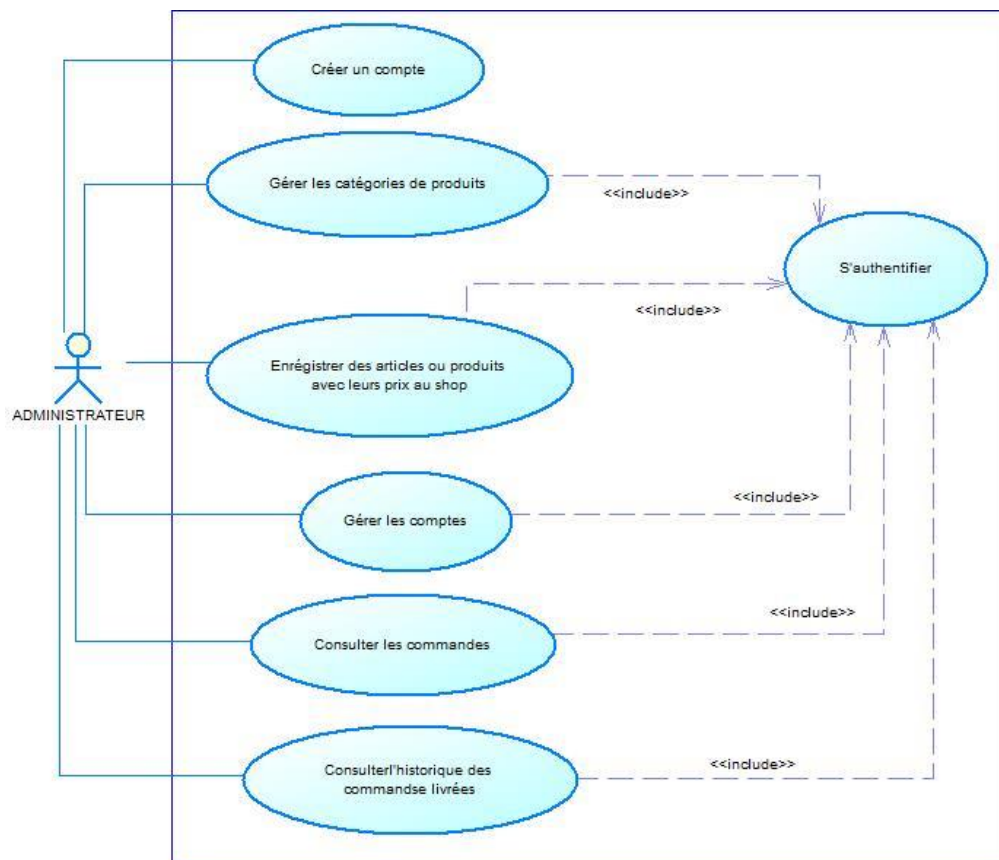


Figure 11: Diagramme de cas d'utilisation de l'acteur administrateur

✓ Diagramme de cas d'utilisation du système

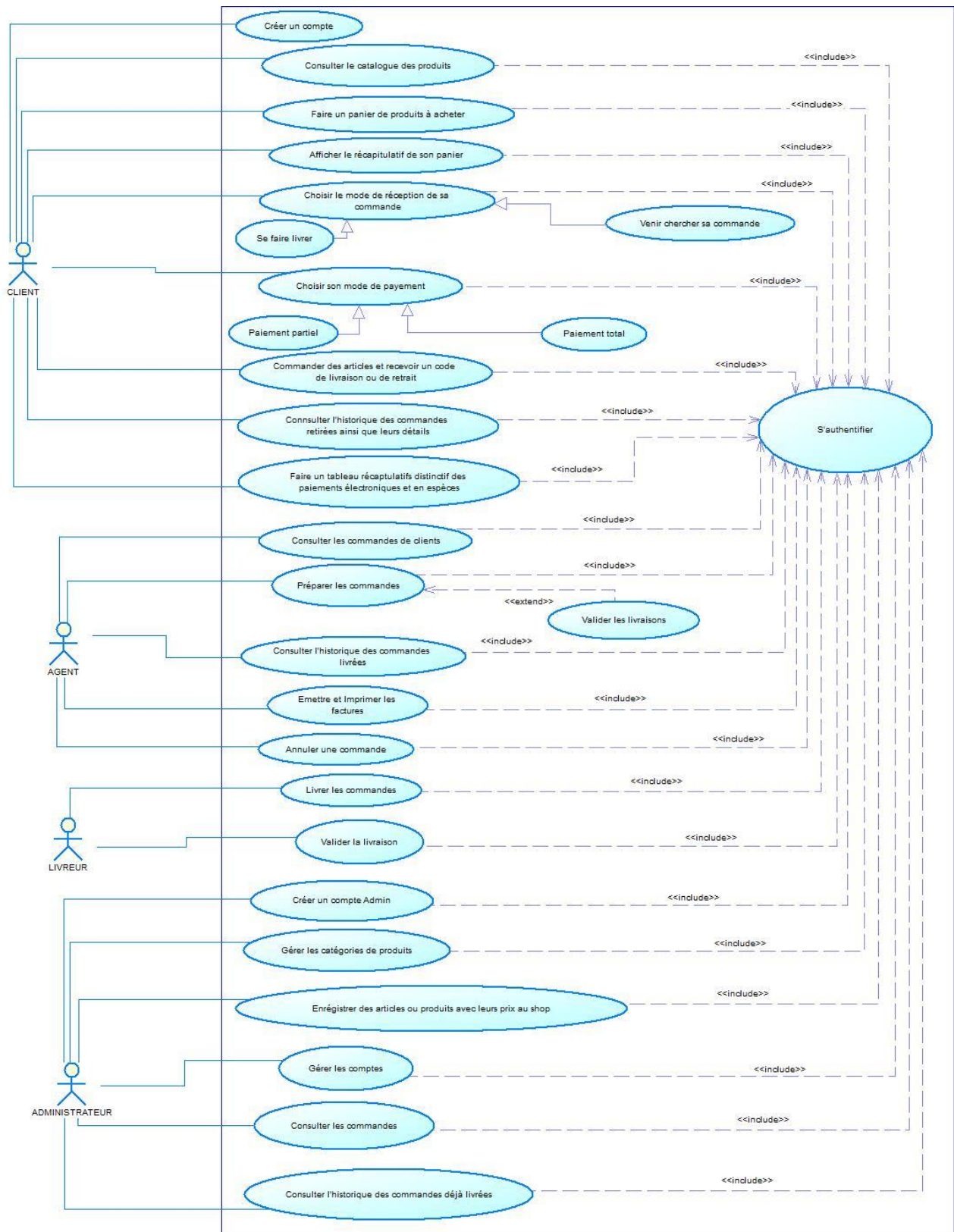


Figure 12: Diagramme de cas d'utilisation du système

C. Vue des processus

1. Description textuelle de quelques cas d'utilisation

La description d'un cas d'utilisation permet de :

- Clarifier le déroulement de la fonctionnalité ;
- Décrire la chronologie des actions qui devront être réalisées ;

a. S'authentifier

Titre : S'authentifier

Acteurs : Administrateur, agent, livreur et client

Résumé : Permet à un utilisateur du système, d'être reconnu par ce dernier en saisissant les paramètres de connexion afin d'y effectuer les actions qui lui sont attribuées.

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions** :
 - Le système est fonctionnel
 - La page d'authentification est accessible
 - Disposer d'un compte
- **Scénario nominal** :
 1. L'utilisateur saisit les paramètres de connexion (E1)
 2. Le système vérifie la validité des paramètres saisis par l'utilisateur (A1)
 3. Le système affiche la page d'accueil avec les menus correspondant aux droits de l'utilisateur.

- **Scénario alternatif** :

A1 : Les paramètres de connexion sont incorrects : Le scénario alternatif est déclenché au point 2 du scénario nominal

- ✓ Le système signale que le mot de passe est erroné : Le scénario reprend au point 1 du scénario nominal

- **Scénario d'exception** :

E1 : L'utilisateur annule la connexion

- ✓ Le scénario nominal est interrompu

- **Post conditions :**
 - ✓ L'utilisateur est authentifié
 - ✓ Le journal est mis à jour
 - ✓ Le système est toujours fonctionnel

b. *Enregistrer un compte utilisateur*

Titre : Enregistrer un compte utilisateur

Acteur : Administrateur

Résumé : Permet à un administrateur d'ajouter un nouveau compte utilisateur au système

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions :**
 - Le système est fonctionnel
 - L'utilisateur s'est authentifié
- **Scénario nominal :**
 1. L'utilisateur saisit les informations sur le compte utilisateur
 2. L'utilisateur saisit les paramètres d'authentification du compte
 3. L'utilisateur choisit le profil à attribuer au compte (E1)
 4. L'utilisateur valide les informations
 5. Le système contrôle les informations saisies par l'utilisateur (A1)
 6. Le système enregistre les informations saisies par l'utilisateur
 7. L'utilisateur reçoit un message de réussite de l'enregistrement

- **Scénario alternatif :**

A1 : Les informations saisies sont incorrectes

- L'utilisateur est invité à saisir des informations correctes

- **Scénario d'exception :**

E1 : L'utilisateur annule l'opération

- Le scénario nominal est interrompu

- **Post conditions**

- Nouveau compte utilisateur créé
- Le journal est mis à jour
- Le système est toujours fonctionnel

c. Enregistrer une catégorie de produits

Titre : Enregistrer une catégorie de produits

Acteur : Administrateur

Résumé : Permet à un administrateur d'ajouter une nouvelle catégorie au système

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions** :

- Le système est fonctionnel
- L'administrateur s'est authentifié

- **Scénario nominal** :

1. L'administrateur saisit les informations sur la catégorie (E1)
2. L'administrateur valide les informations
3. Le système contrôle les informations saisies par l'utilisateur (A1)
4. Le système enregistre les informations saisies par l'utilisateur
5. L'utilisateur reçoit un message de réussite de l'enregistrement

- **Scénario alternatif** :

A1 : Les informations saisies sont incorrectes

- L'administrateur est invité à saisir des informations correctes

- **Scénario d'exceptions** :

E1 : L'utilisateur annule l'opération

- Le scénario nominal est interrompu

- **Post conditions**

- Nouvelle catégorie ajoutée
- Le journal est mis à jour
- Le système est toujours fonctionnel

d. Supprimer une catégorie de produits

Titre : Supprimer une catégorie de produits

Acteur : Administrateur

Résumé : Permet à un administrateur de supprimer une occurrence de catégorie du système

Responsable : NIKABOU

Version : 1.0

Date de création : 28/07/2018

Date de modification : 18/08/2018

Description des scénarii :

- **Pré conditions :**

- ✓ Le système est fonctionnel
- ✓ L'utilisateur s'est authentifié

- **Scénario nominal :**

1. L'administrateur souhaite supprimer une catégorie de produit
2. Le système demande une confirmation
3. L'administrateur valide (E1)
4. Le système supprime la catégorie de produits et les produits appartenant à cette catégorie
5. Le système envoie un message de confirmation

- **Scénario d'exception :**

E1 : *L'administrateur annule la suppression*

- ✓ Le scénario nominal est interrompu

- **Post conditions :**

- Le journal est mis à jour
- Le système est toujours fonctionnel

e. Enregistrer un produit

Titre : Enregistrer un produit

Acteur : Administrateur

Résumé : Permet à un administrateur d'ajouter un nouveau produit au système

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions :**

- Le système est fonctionnel
- L'administrateur s'est authentifié

- **Scénario nominal :**

1. L'administrateur saisit les informations sur le produit (E1)
2. L'administrateur choisit la catégorie du produit
3. L'administrateur choisit l'image correspondante au produit
4. L'administrateur valide les informations
5. Le système contrôle les informations saisies par l'utilisateur (A1)
6. Le système enregistre les informations saisies par l'utilisateur
7. L'utilisateur reçoit un message de réussite de l'enregistrement

- **Scénario alternatif :**

A1 : Les informations saisies sont incorrectes

L'administrateur est invité à saisir des informations correctes

- **Scénario d'exceptions :**

E1 : L'utilisateur annule l'opération

- Le scénario nominal est interrompu

- **Post conditions**

- Nouveau produit ajouté
- Le journal est mis à jour
- Le système est toujours fonctionnel

f. Consulter les catalogues des produits

Titre : Consulter les catalogues des produits

Acteurs : Client

Résumé : Permet au client, de consulter les catalogues de produits.

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions** :
 - Le système est fonctionnel
 - La connexion internet est fluide
- **Le scénario nominal** :
 1. Le système affiche une page contenant la liste les catégories de produits.
 2. Le client sélectionne une des catégories. (E1) (E2)
 3. Le système recherche les produits qui appartiennent à cette catégorie.
 4. Le système affiche une description et une photo pour chaque produit trouvé.
 5. Le client peut sélectionner un produit parmi ceux affichés. (E1) (E2)
 6. Le système affiche les informations détaillées du produit choisi.
 7. Le client peut ensuite quitter cette description détaillée.
 8. Le système retourne à l’affichage des produits de la catégorie (retour à l’étape 4)
- **Le scénario alternatif** : Aucun
- **Les scénarii d’exceptions**

E1 : *Le client décide de quitter la consultation de la catégorie de produits choisie.*
✓ Le système retour à la page d’accueil

E2 : *Le client décide de quitter la consultation du catalogue.*
✓ Le système retourne à la page d’accueil
- **Post condition** : Aucun

g. Valider une commande

Titre : Valider une commande

Acteurs : Agent

Résumé : Permet à un agent de valider une commande

Responsable : NIKABOU

Version : 1.0

Date de création : 28/07/2018

Date de modification : 18/08/2018

Description des scénarii :

- **Pré condition :**

- ✓ Le système est fonctionnel
- ✓ L'agent s'est authentifié

- **Scénario nominal :**

1. L'agent consulte l'historique des commandes en attente
2. L'agent clique sur le bouton de consultation de détails d'une commande à valider
3. Le système affiche le détail
4. L'agent vérifie la disponibilité des produits (A1)
5. L'agent valide la commande

- **Scénario alternatif :**

A1 : *Certains produits ne sont pas disponibles*

- ✓ L'agent annule la commande
- ✓ Le système informe le client de la non disponibilité de ces produits : Le scénario reprend au niveau 1 du scénario nominal

- **Post conditions :**

- ✓ Le système est toujours fonctionnel

h. Commander des articles

Titre : Commander des articles

Acteurs : Client

Résumé : Permet à un client d'effectuer une commande.

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions** :
 - Le système est fonctionnel
 - La connexion est fluide
- **Scénario nominal** :
 1. Le client consulte les catalogues
 2. Le client choisit de faire des achats et de faire son panier
 3. Le système affiche les produits dans leurs catégories
 4. Le client choisit les produits qu'il souhaite acheter
 5. Le client termine son achat
 6. Le système affiche le récapitulatif du panier du client
 7. Le client souhaite valider sa commande (E1)
 8. Le système récupère les paramètres nécessaires
 9. Le système envoie un bon de livraison au client
- **Scénario alternatif** : Aucun
- **Scénario d'exceptions** :

E1 : *Le client annule l'opération*

- ✓ Le système revient à la page de liste des catégories

- **Post conditions** : Aucun

i. *Valider une livraison*

Titre : Valider une livraison

Acteurs : Agent, livreur

Résumé : Permet à un agent ou à un livreur de valider les livraisons.

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions** :
 - Le système est fonctionnel
 - La connexion est fluide
 - L'utilisateur s'est authentifié
 - La commande est validée
- **Scénario nominal** :
 - ✓ L'utilisateur consulte les livraisons en attente
 - ✓ Le système affiche la liste des livraisons en attente
 - ✓ L'utilisateur affiche le détail de la commande
 - ✓ L'utilisateur désire valider une livraison
 - ✓ Le système demande le bon de livraison du client
 - ✓ L'utilisateur saisit le bon du client
 - ✓ Le système vérifie l'information saisit(A1) (E1)
 - ✓ L'utilisateur enregistre le règlement
 - ✓ L'utilisateur valide la livraison
- **Scénario alternatif** :

A1 : *Les paramètres saisis sont incorrects*

- ✓ Le système informe l'utilisateur et demande de nouveau la saisie

- **Scénario d'exceptions** :

E1 : *L'utilisateur annule l'opération*

- ✓ Le système revient à la page d'accueil

- **Post conditions** : Aucun

j. *Consulter l'historique des commandes et des livraisons*

Titre : Consulter l'historique des commandes et des livraisons

Acteurs : Administrateur, agent, livreur

Résumé : Permet à un administrateur, à un agent et à un livreur de connaître les statistiques des commandes et des livraisons.

Responsable : NIKABOU

Version : 1.0

Date de création : 28/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions :**

- Le système est fonctionnel
- L'administrateur, l'agent ou le livreur est authentifié

- **Scénario nominal :**

1. Le système établit les statistiques de commandes, des livraisons et des règlements.
2. Le système affiche le résultat

- **Post condition :**

- ✓ Le système est toujours fonctionnel

k. Imprimer une facture

Titre : Imprimer une facture

Acteurs : Agent

Résumé : Permet à un agent d'imprimer une facture

Responsable : NIKABOU

Version : 1.0

Date de création : 05/07/2018

Date de modification : 10/08/2018

Description des scénarii :

- **Pré conditions :**

- Le système est fonctionnel
- La connexion est fluide
- L'utilisateur s'est authentifié
- La livraison est validée

- **Scénario nominal :**

1. L'utilisateur consulte l'historique des commandes validées

2. L'utilisateur désire imprimer la facture d'une commande validée et livrée
3. Le système affiche l'aperçu avant impression
4. L'utilisateur valide (A1)
5. Le système imprime la facture

- **Scénario alternatif :**

A1 : *L'utilisateur annule l'opération*

- ✓ Le scénario nominal est interrompu

- **Post conditions :**

- ✓ Disposer d'un compte actif

1. *Consulter les statistiques*

Titre : Consulter les statistiques

Acteur : Administrateur

Résumé : Permet à un administrateur de connaître les statistiques des opérations menées.

Responsable : NIKABOU

Version : 1.0

Date de création : 28/07/2018

Date de modification : 18/08/2018

Description des scénarii :

- **Pré conditions :**

- Le système est fonctionnel
- L'utilisateur s'est authentifié

- **Scénario nominal :**

1. Le système établit les statistiques des opérations effectuées
2. Le système affiche le résultat

- **Post condition :**

- Le système est toujours fonctionnel

2. Diagramme de séquence

a. Généralités

Les **diagrammes de séquences** sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

Le diagramme de séquences système lui, est présenté en montrant l'acteur principal à gauche, puis un objet unique représentant le système en boîte noire, et enfin les éventuels acteurs secondaires sollicités durant le scénario à droite du système.

Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

Les diagrammes de séquence représentent la séquence de messages transmis entre des objets. Ils peuvent également représenter les structures de contrôle entre des objets.

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie :

- Ils sont représentés par des flèches
- Ils sont présentés du haut vers le bas le long des lignes de vie, dans un ordre chronologique

Un message définit une communication particulière entre des lignes de vie (objets ou acteurs).

Plusieurs types de messages existent, dont les plus courants :

- l'envoi d'un signal ;
- l'invocation d'une opération (appel de méthode) ;
- la création ou la destruction d'un objet.

Un message synchrone bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.

- Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé.
- On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone.

Un message asynchrone n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

b. Quelques diagrammes de séquence

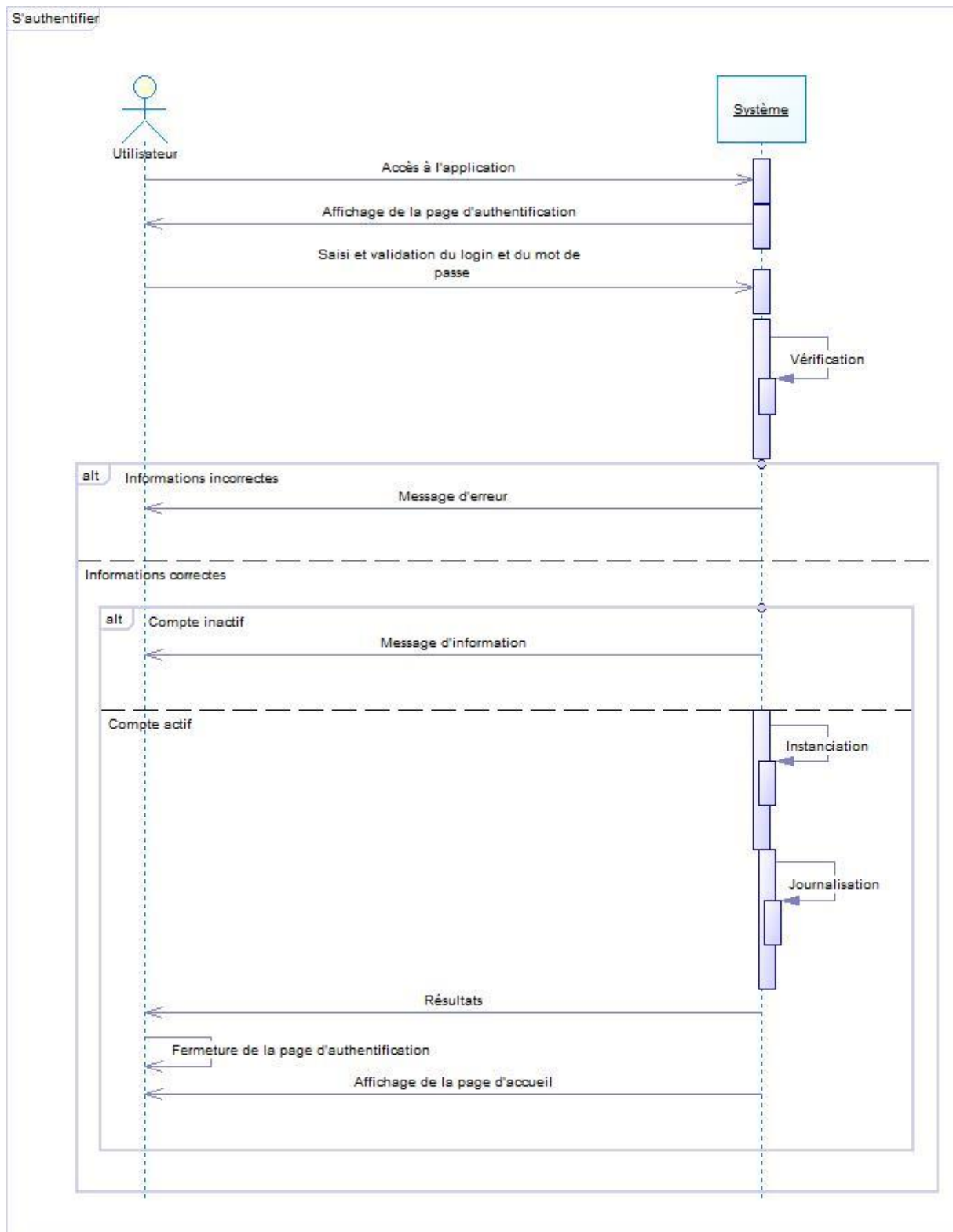


Figure 13: Diagramme de séquence du cas d'utilisation « s'authentifier »

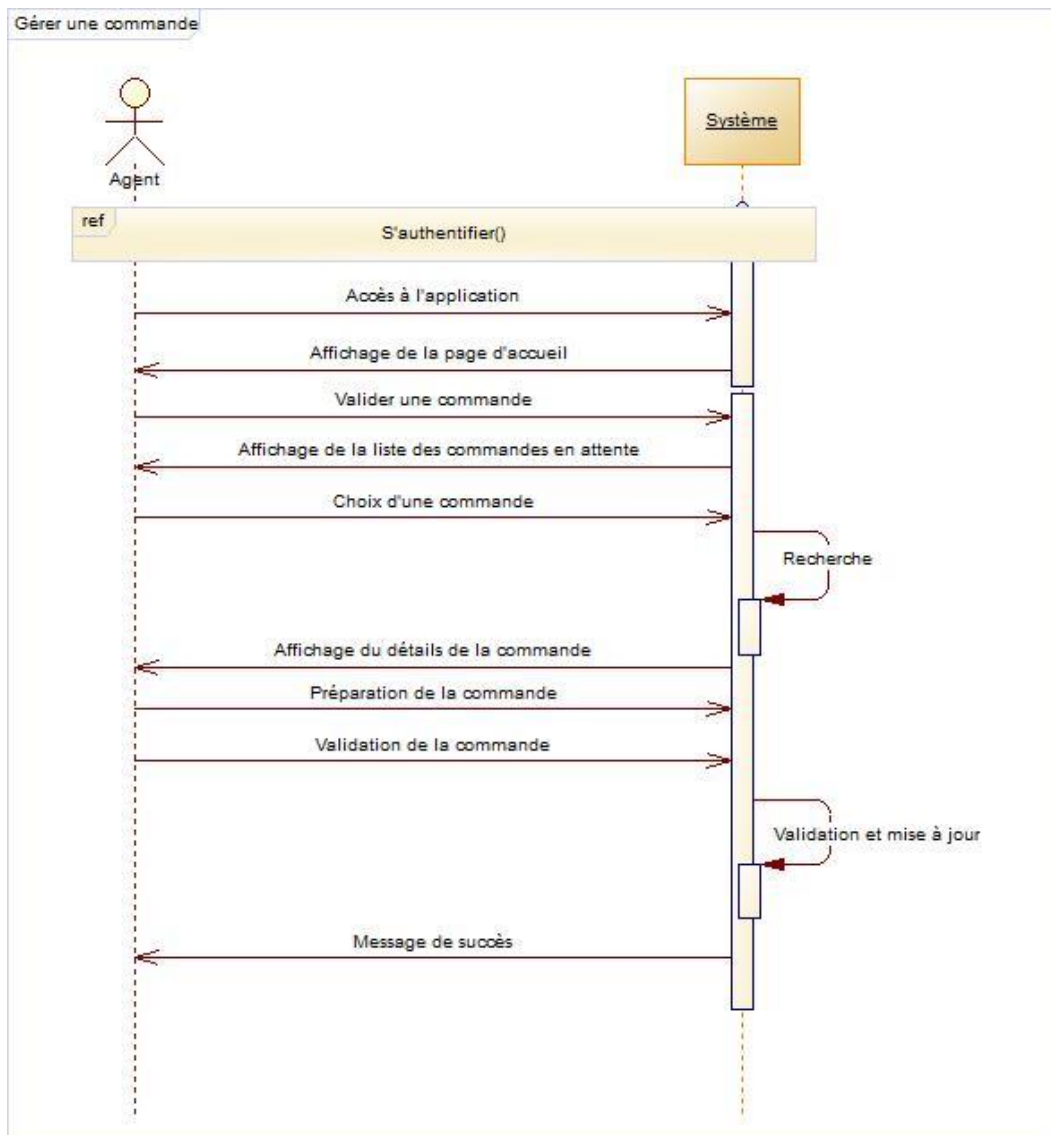


Figure 14: Diagramme de séquence du cas d'utilisation « gérer les commandes »

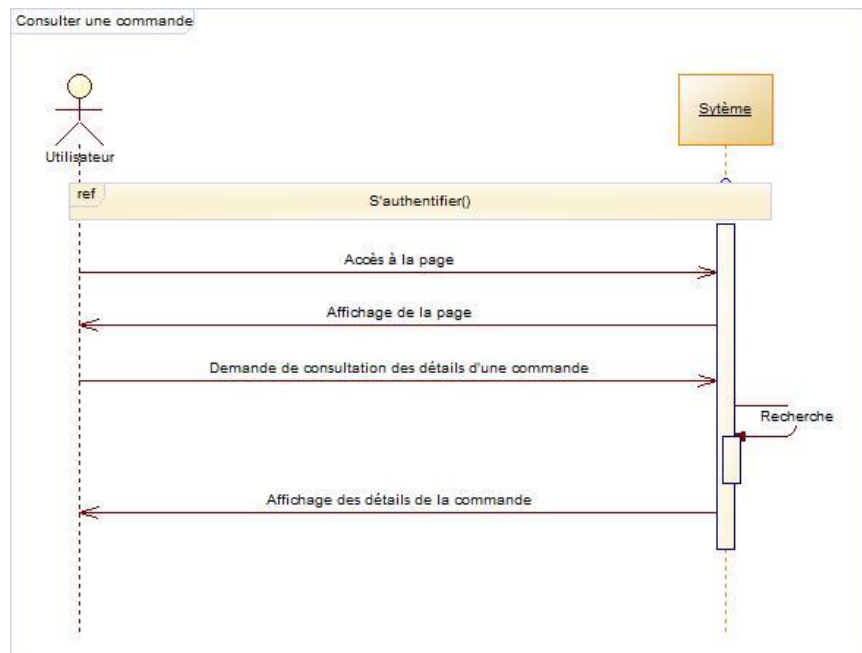


Figure 15: Diagramme de séquence du cas d'utilisation « consulter une commande »

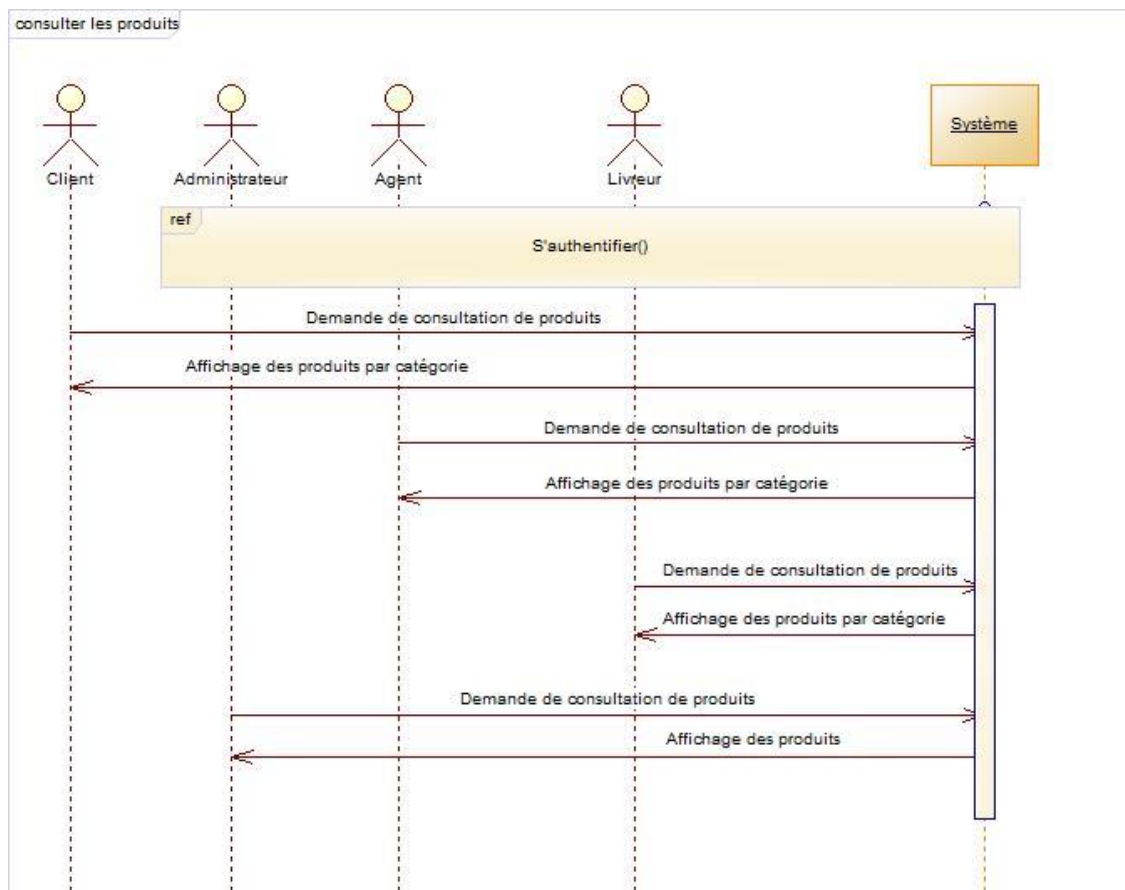


Figure 16: Diagramme de séquence du cas d'utilisation « consulter les produits »

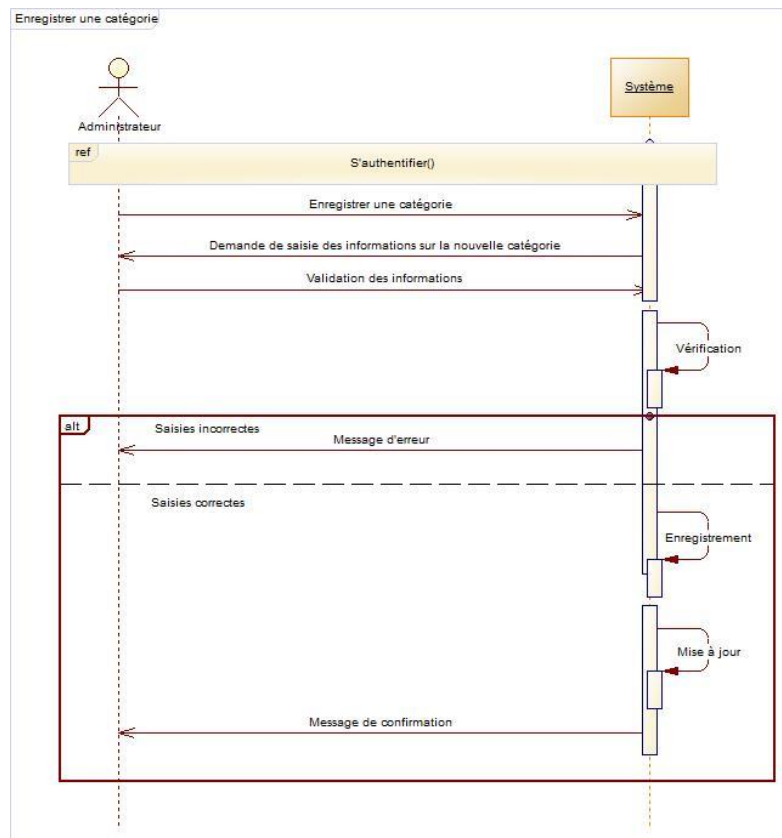


Figure 17: Diagramme de séquence du cas d'utilisation « enregistrer une catégorie »

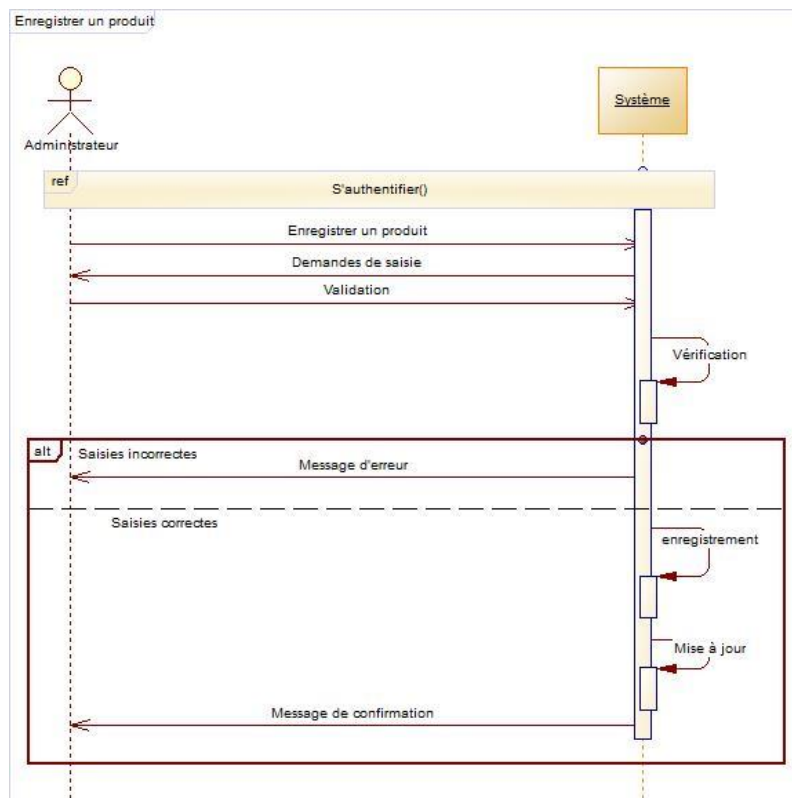


Figure 18: Diagramme de séquence du cas d'utilisation « enregistrer un produit »

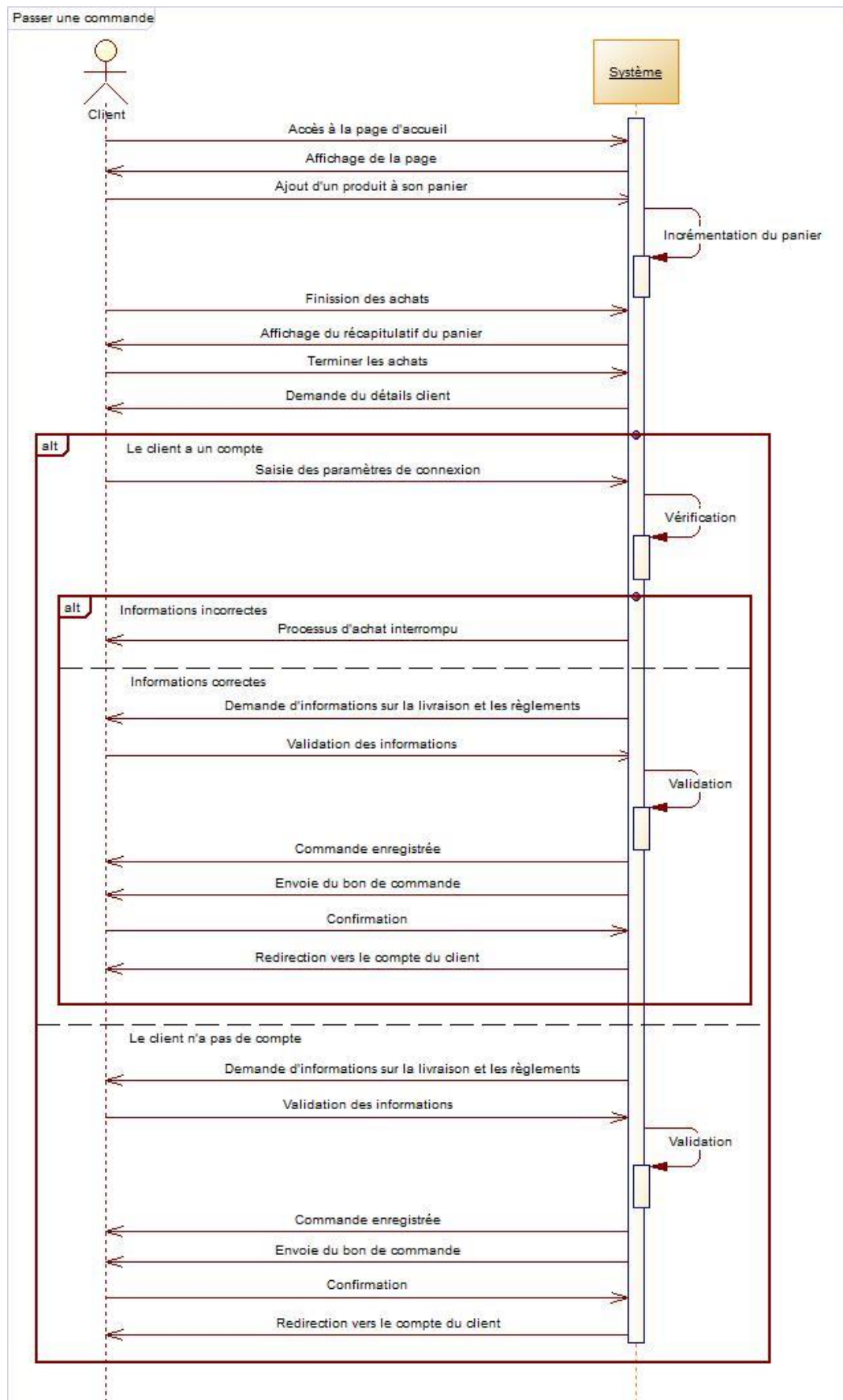


Figure 19: Diagramme de séquence du cas d'utilisation passer une commande

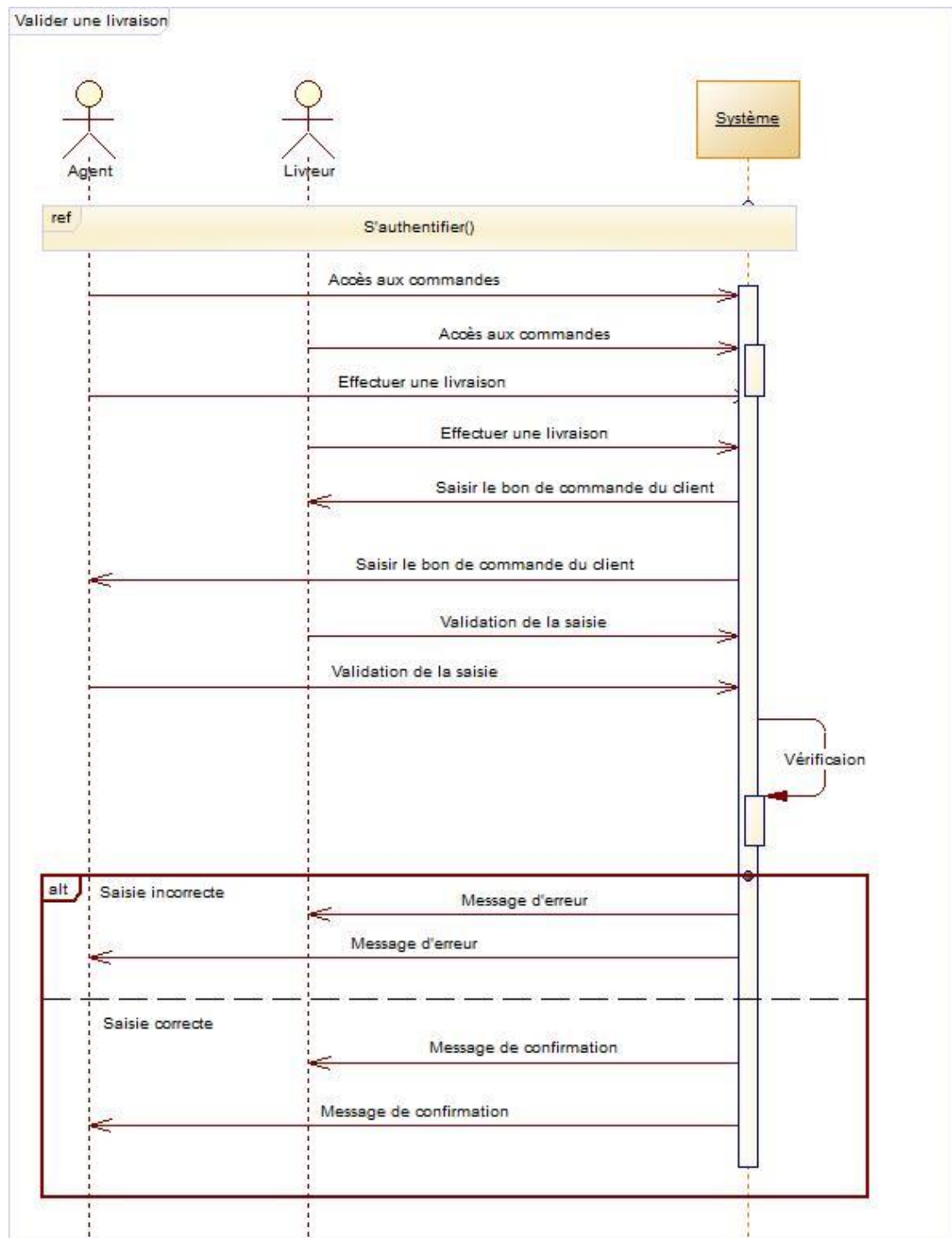


Figure 20: Diagramme de séquence du cas d'utilisation « valider une livraison »

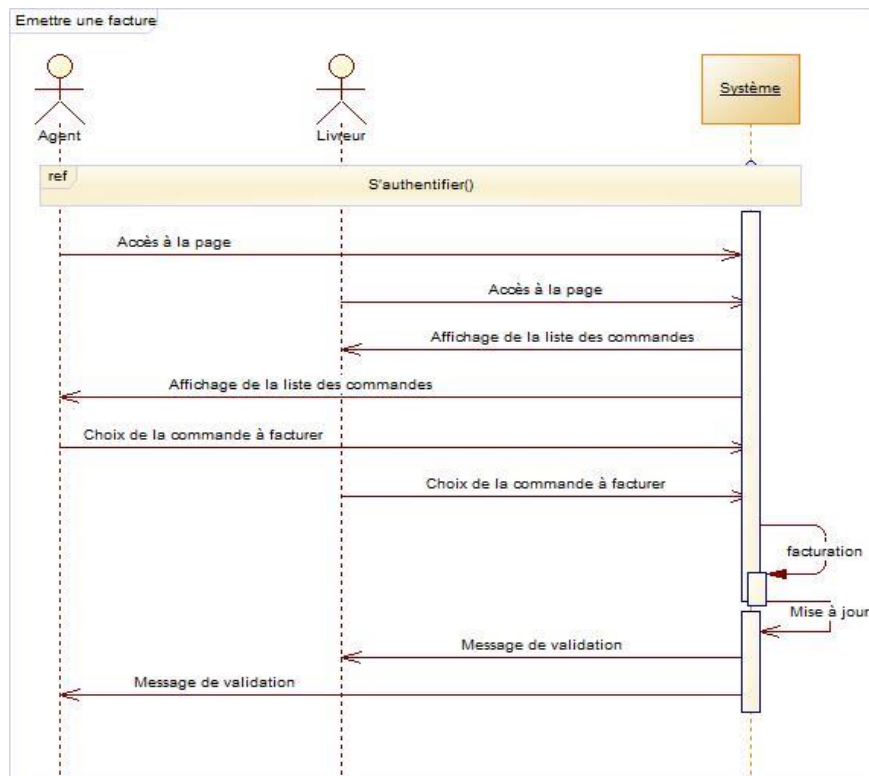


Figure 21: Diagramme de séquence du cas d'utilisation « émettre une facture »

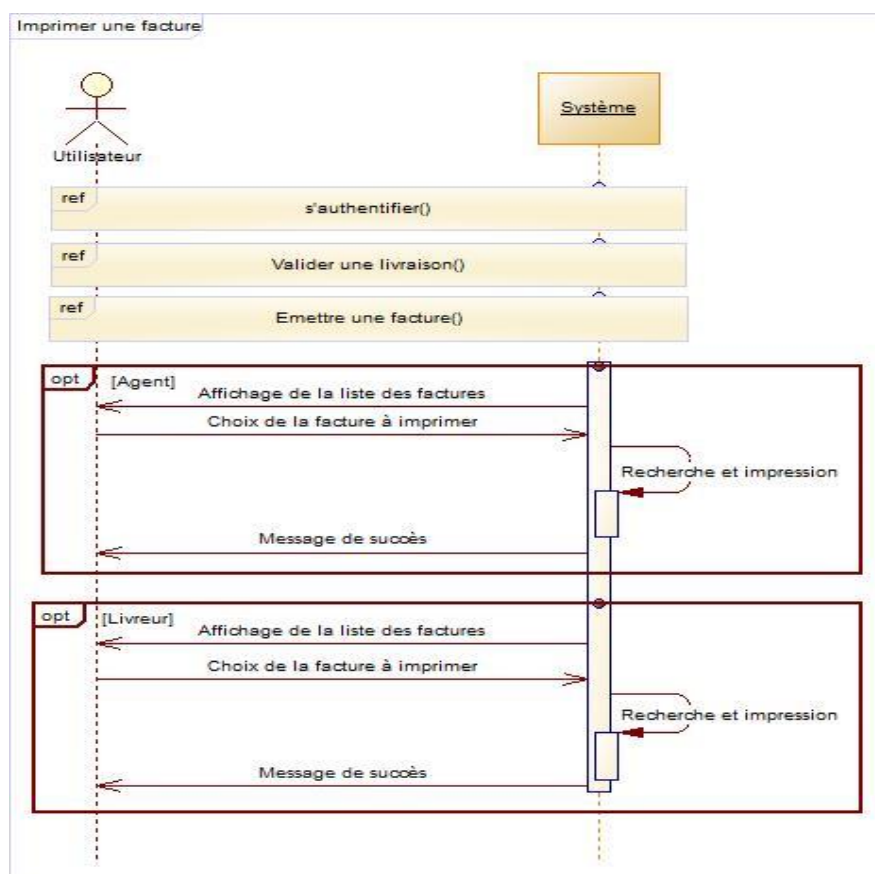


Figure 22: Diagramme de séquence du cas d'utilisation « imprimer une facture »

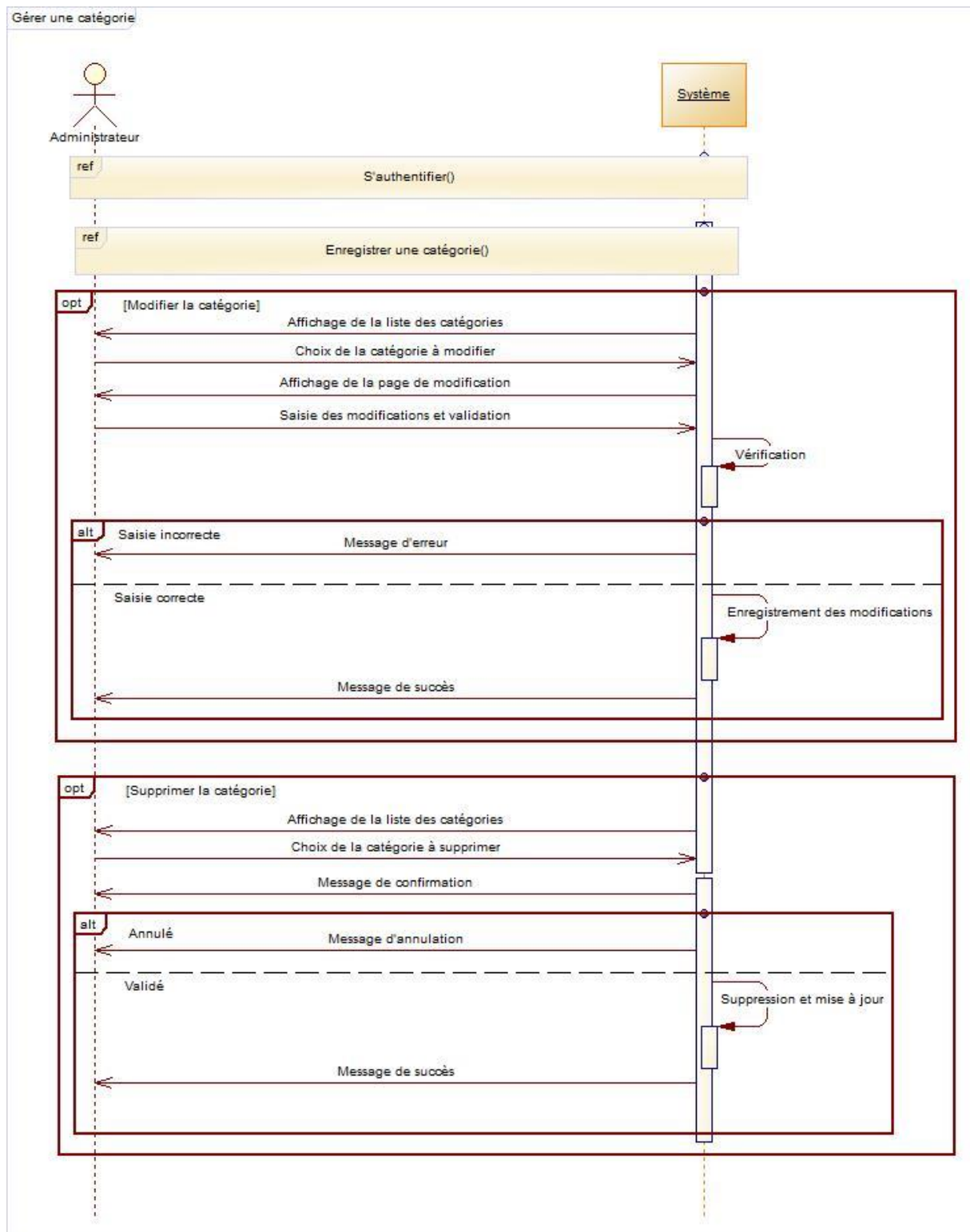


Figure 23: Diagramme de séquence du cas d'utilisation « gérer une catégorie »

3. Diagramme d'activité

a. Généralités

Le diagramme d'activités se présente comme un organisme qui décrit les différents enchainements ou étapes du déroulement d'un cas d'utilisation ou groupe de cas d'utilisation.

Le diagramme d'activité est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Certains des composants les plus courants d'un diagramme d'activités sont :

- **Actions** : étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée.
- **Nœud de décision** : embranchement conditionnel dans le flux qui est représenté par un losange. Il comporte une seule entrée et deux sorties ou plus.
- **Flux de contrôle** : autre nom des connecteurs qui illustrent le flux entre les étapes du diagramme.
- **Nœud de départ** : symbolise le début de l'activité. Cet élément est représenté par un cercle noir.
- **Nœud de fin** : représente l'étape finale de l'activité. Il est représenté par un cercle noir avec un contour.

b. Quelques diagrammes d'activités de notre système

Le diagramme d'activités présente uniquement les actions du système. Voici quelques diagrammes d'activité de notre système :

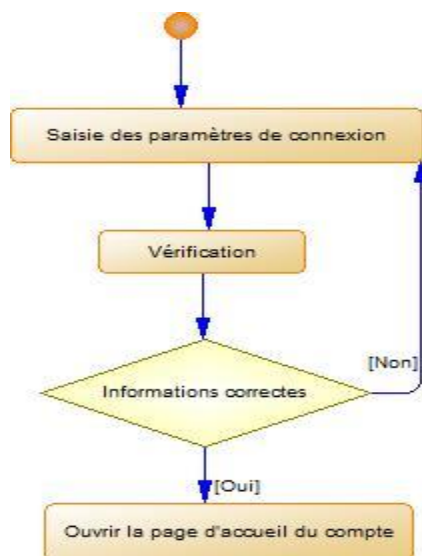


Figure 24: Diagramme d'activité du cas d'utilisation « s'authentifier »

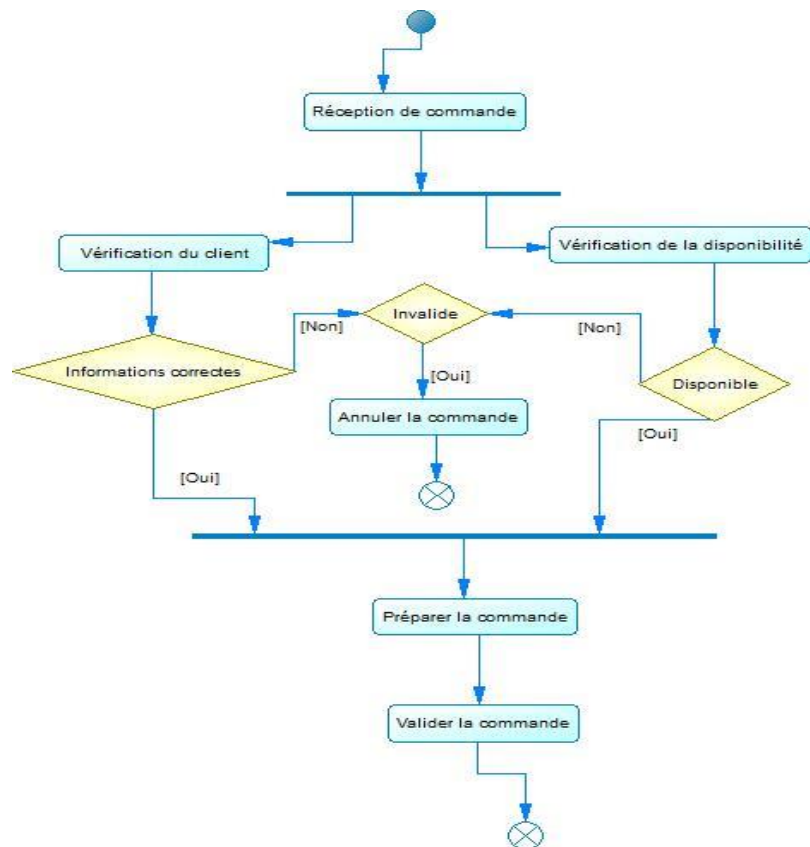


Figure 25: Diagramme d'activité du cas d'utilisation « valider une commande »

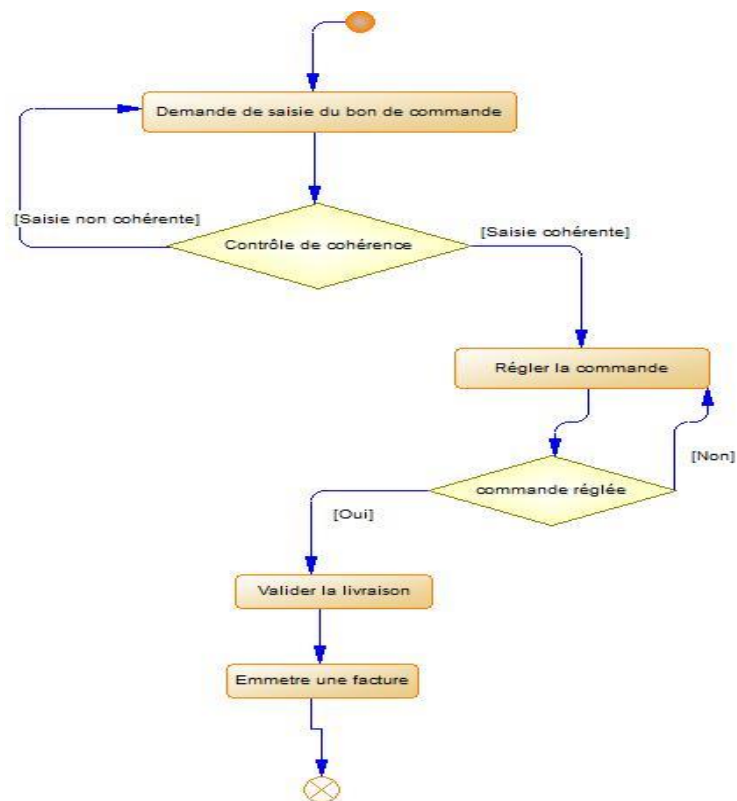


Figure 26: Diagramme d'activité du cas d'utilisation « effectuer une livraison »

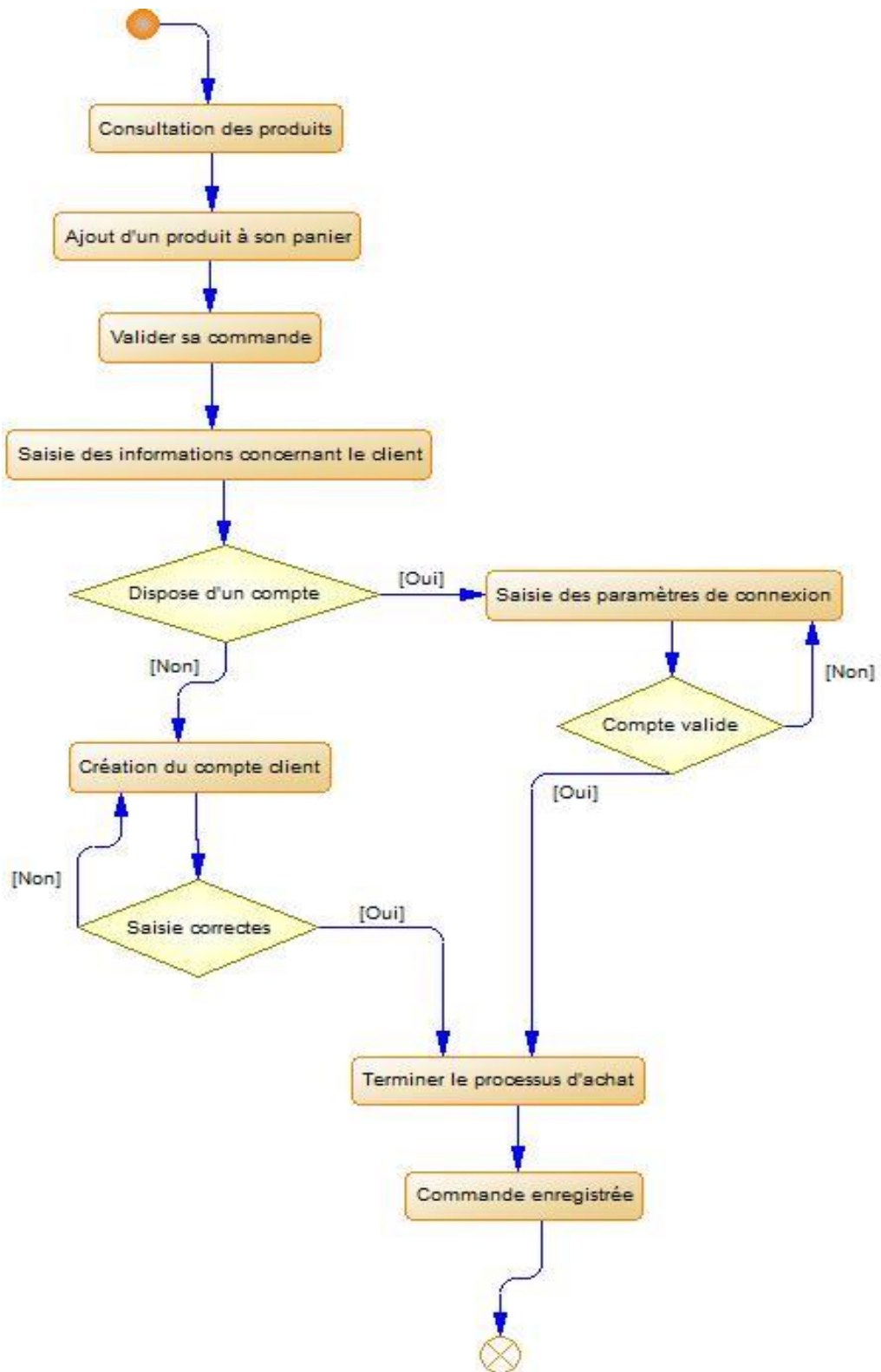


Figure 27: Diagramme d'activité du cas d'utilisation « effectuer une commande »

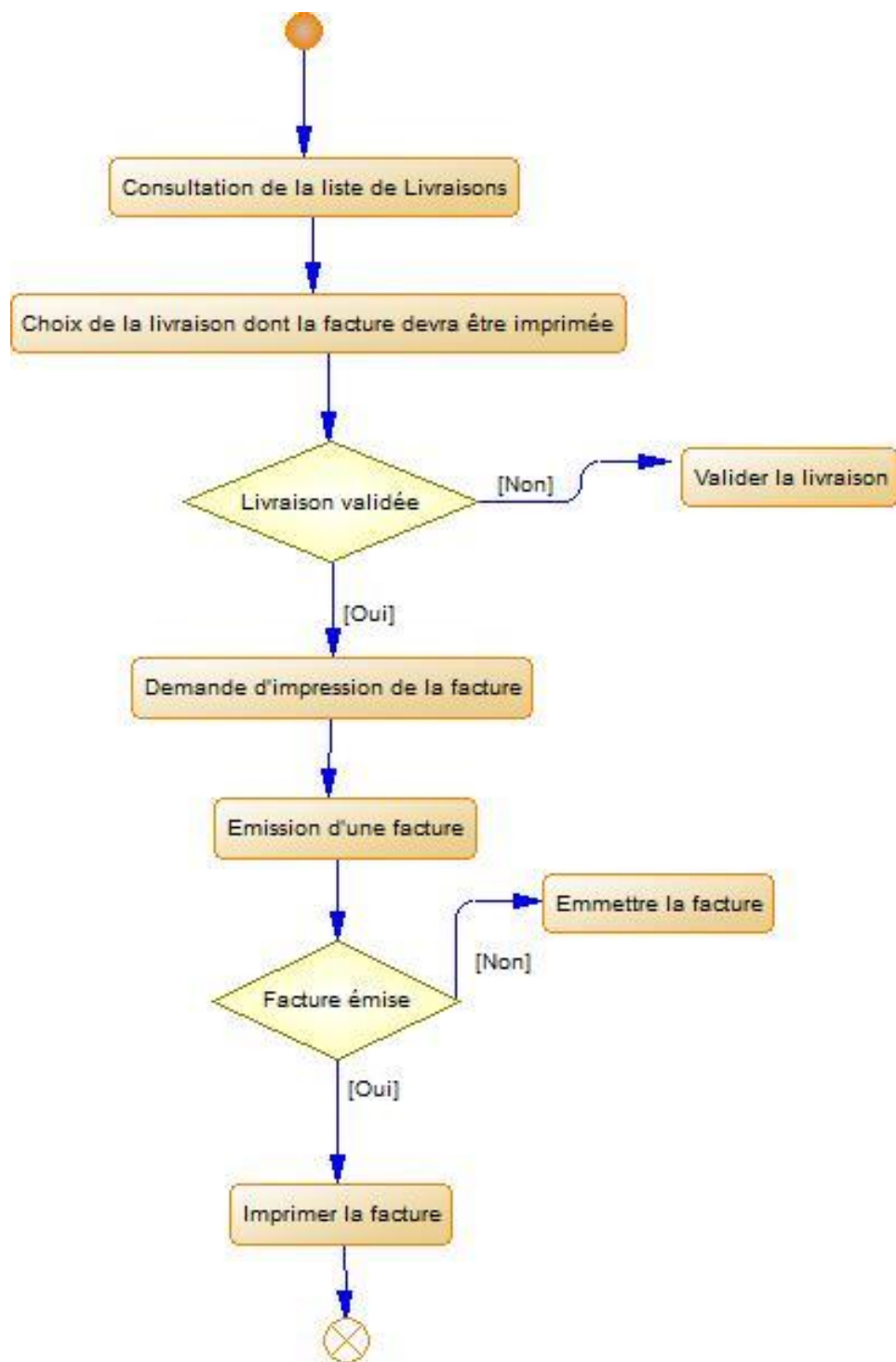


Figure 28: Diagramme d'activité du cas d'utilisation « imprimer une facture »

D. VUE LOGIQUE

Diagramme de classe-entité

a. Généralités

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Le diagramme des classes comporte les concepts suivants :

- ✓ Classe
- ✓ Attribut
- ✓ Identifiant
- ✓ Opération
- ✓ Relation
- ✓ Généralisation

b. Diagramme des classes de notre système

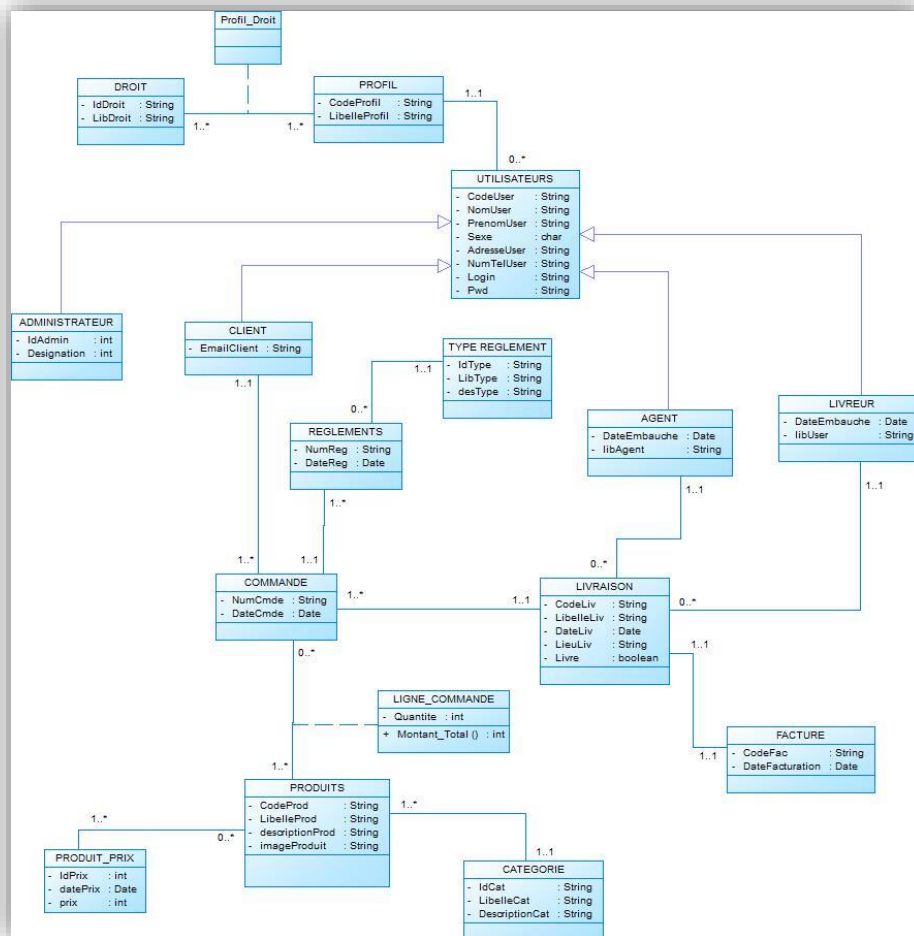


Figure 29: Diagramme des classes de notre système

E. Vue de déploiement

Cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le où). Le pourquoi, n'est pas défini dans UML.

Diagramme de déploiement

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels.

- Les ressources matérielles sont représentées sous forme de nœuds
- Les nœuds sont connectés entre eux, à l'aide d'un support de communication
- La nature des lignes de communication et leurs caractéristiques peuvent être

Précisées

- Les diagrammes de déploiement peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.
- Les diagrammes de déploiement correspondent à la vue de déploiement d'une architecture logicielle.

Le diagramme de déploiement de notre système se présente comme suit :

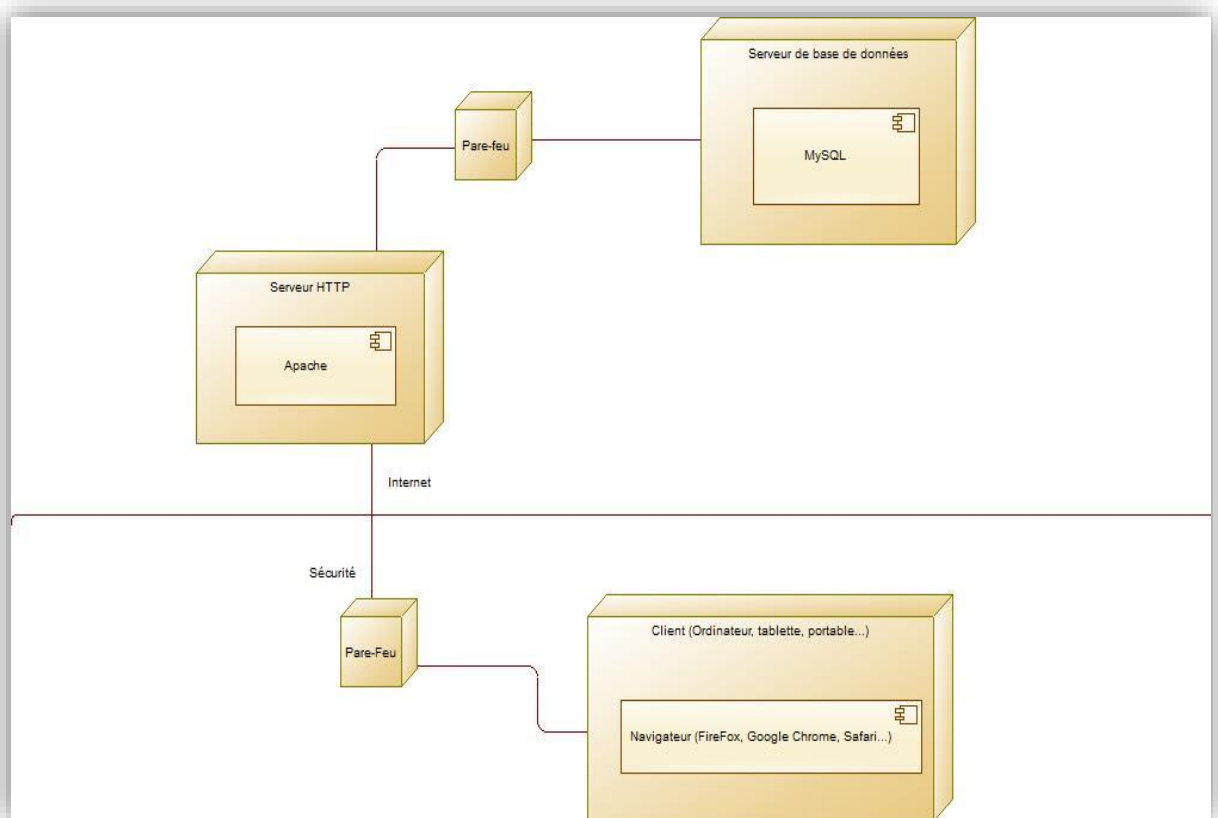


Figure 30: Diagramme de déploiement du système



PARTIE III- REALISATION ET MISE EN OEUVRE

I. MISE EN ŒUVRE

La mise en œuvre constitue la réalisation effective de l'application. Faisant intervenir divers outils, elle comprend donc le choix du matériel, des logiciels utilisés et la sécurité de l'application.

1. Choix matériels

Notre matériel de développement durant notre stage est un ordinateur portable qui présente les caractéristiques suivantes :

- ✓ Marques : Hewlett-Packard (HP)
- ✓ Modèle : HP-Notebook 2200
- ✓ Processeur : Intel(R) Pentium(R) CPU 3825U @ 1,90GHz
- ✓ Mémoire RAM : 4,00 Go
- ✓ Disque dur : 500 Go
- ✓ Lecteur : DVD-RW et CD-ROM

2. Choix logiciels

Pour la réalisation de ce projet de stage, nos choix logiciels ont été très étendus. Nous présenterons d'une part les outils d'implémentation de la base de donnée et d'autre part les outils de programmation et de développement.

a) Outils d'implémentation de la base de données

Une base de données est implémentée sur un outil logiciel appelé SGBD (Système de Gestion de Bases de Données) qui se charge de gérer son interaction avec les utilisateurs ou leurs programmes. Le SGBD retenu pour l'implémentation de la Base de Données est MySQL.

- ✓ *MySQL*

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

C'est un logiciel libre développé sous double licence en fonction de l'utilisation qui en est faite : dans un produit libre ou dans un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Nous utiliserons la version 5.6.12 de MYSQL.

✓ WampServer



Figure 31: Logo de WampServer

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

✓ PowerAMC version 15.1

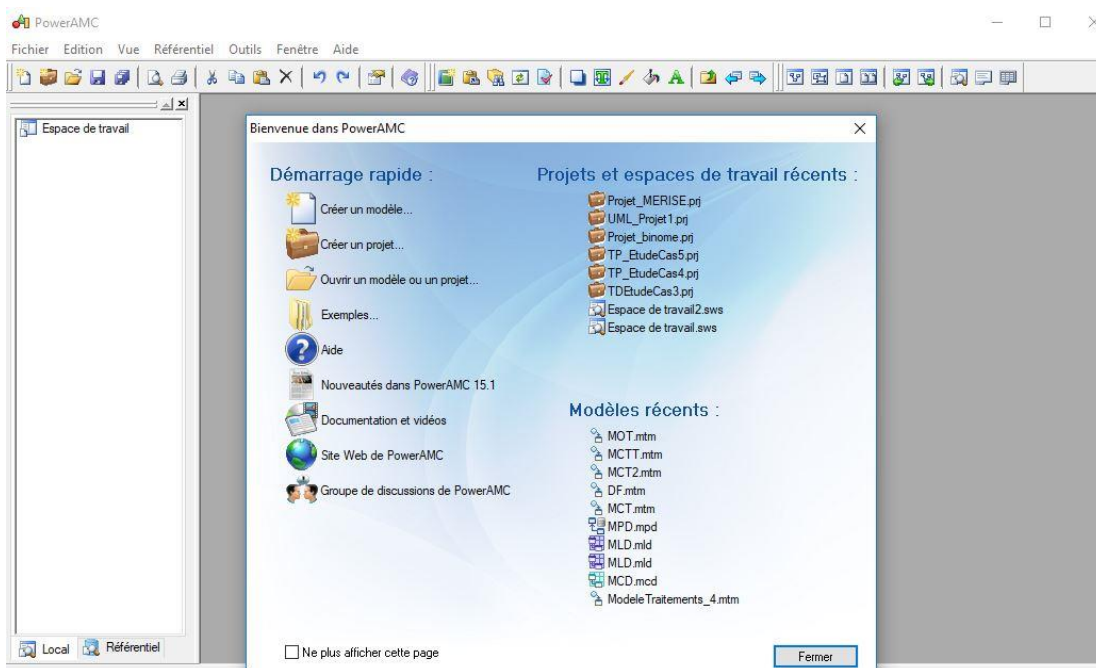


Figure 32: Interface d'accueil de PowerAMC

b) Outils de programmation et de développement

Afin d'aboutir à une plateforme fonctionnelle, un certain nombre d'outils de programmation ont été nécessaires. On a le langage de programmation qui sert à donner des instructions machines puis l'environnement de développement intégré (EDI) qui sert à piloter le codage. Le langage de programmation utilisé est le JAVA ; L'EDI utilisé est NetBeans.

✓ Java

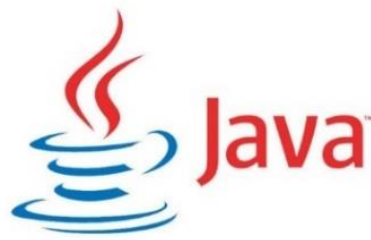


Figure 33: : Logo de Java

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications, mais qui ont l'inconvénient d'être plus lourd à l'exécution (en mémoire et en temps processeur) à cause de sa machine virtuelle. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

✓ HTML 5 et CSS 3



Figure 34: logo de HTML 5 et CSS 3

HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014. HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS).

Les CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-

dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.

✓ *JavaScript*



Figure 35: Logo de JavaScript

Le Javascript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

Ainsi le langage Javascript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu.

✓ *Bootstrap*



Figure 36: Logo de Bootstrap

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub (c'est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions décentralisé (Git)).

✓ *Play Framework**Figure 37: Logo de Play*

Play est un framework web open source qui permet d'écrire rapidement des applications web en Java ou en Scala. Créé par Guillaume Bort en 2007, co-fondateur et associé de Zengularity SA, Play Framework s'inspire d'autres frameworks comme Ruby on Rails ou Django, en visant à apporter un outil simple et productif sur la machine virtuelle Java. Play Framework a pour particularité de ne pas être basé sur le moteur Java de Servlet. C'est un choix délibéré de l'auteur visant à offrir un système plus simple et plus puissant pour développer une application Web en Java.

✓ *L'Environnement de Développement Intégré NetBeans**Figure 38: Logo de NetBeans*

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en Juin 2000 sous licence CDDL (Common Development and Distribution Licence) et GPLv2. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web).

Conçu en Java (qui est multiplateforme), NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), MAC OS X ou sous une version indépendante des systèmes d'exploitation (requérant la machine virtuelle de Java, la JVM).

Dans le développement de notre application, il a été utilisé la version 8.2 de NetBeans interfacée par l'image suivante :

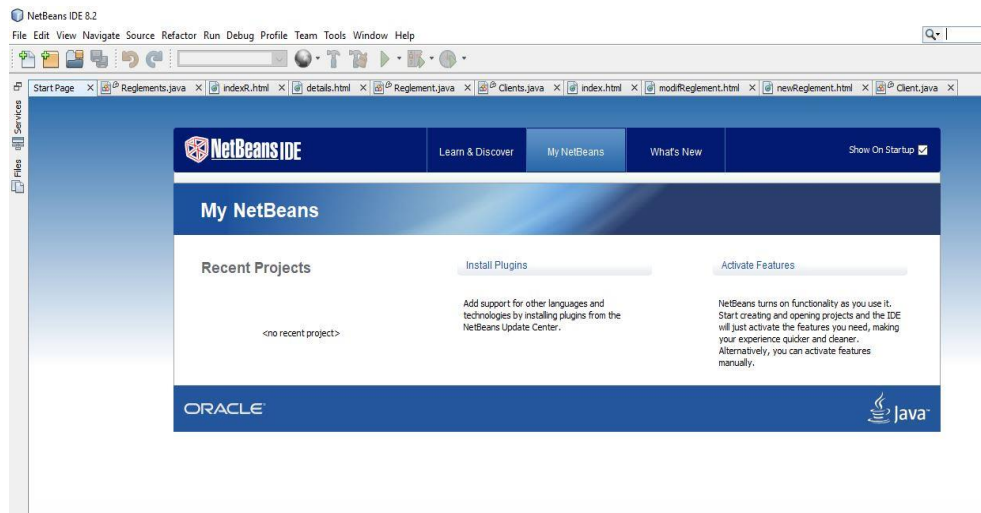


Figure 39: Interface de NetBeans

3. Sécurité de l'application

a. Sécurité au niveau de l'hébergeur de l'application web

Depuis plus de quinze ans, OVH héberge les données et porte une attention particulière à leur protection. Pour garantir les plus hauts niveaux de sécurité, OVH a pris soin de mettre en place des dispositifs efficaces ainsi qu'un arsenal de bonnes pratiques à tous les niveaux de leur organisation et de leurs infrastructures :

- ✓ Des datacenters placés sous haute protection
- ✓ Gestion des risques d'incendie
- ✓ Sécurité côté serveur
- ✓ Des centres de données géographiquement distants
- ✓ Protection anti-DDoS
- ✓ Double authentification
- ✓ Confidentialité, disponibilité, intégrité des informations

b. Sécurité au niveau de l'application

Vu les expositions des applications Web aux risques, un certain nombre de règles ont été instaurées dans le processus de développement pour sécuriser la plateforme :

✓ **Une page d'authentification** : première page de l'application permettant aux utilisateurs de se connecter à partir des paramètres de connexions (login et mot de passe) ;

✓ **Compte utilisateur** : chaque client possède un seul compte utilisateur. Un compte peut être à l'état « actif » ou « inactif » (précaution prise pour bloquer les comptes en cas de piratage par exemple).

✓ **Journalisation des opérations** : toute action ou opération effectuée par un utilisateur est journalisée (enregistrées dans un journal portant les informations suivantes : date de l'événement, type d'événement, libellé de l'action, etc.), histoire de les retracer.

II. PRESENTATION DE L'APPLICATION

Cette partie présente l'application d'un point de vue graphique (les Interfaces Hommes Machines (IHM)) et logique (le script de création de la base de données).

1. Présentation

L'application que nous avons développée est nommée « **ShopOnline** ». Cette application est un module web écrit dans le langage Java Play, permettant d'effectuer des achats de produits en ligne.

Elle offre des interfaces simples et facilement manipulables pour une bonne gestion des clients et des achats.

Elle utilise une architecture Client-Serveur deux-tiers et sera donc installée sur un serveur distant et les postes client y accéderont avec une URL dans un navigateur.

2. Architecture

Les menus de l'application sont accessibles en fonction du profil utilisateur. Nous présentons la structure de quelques onglets en fonction des profils.

✓ Onglets du profil « livreur »

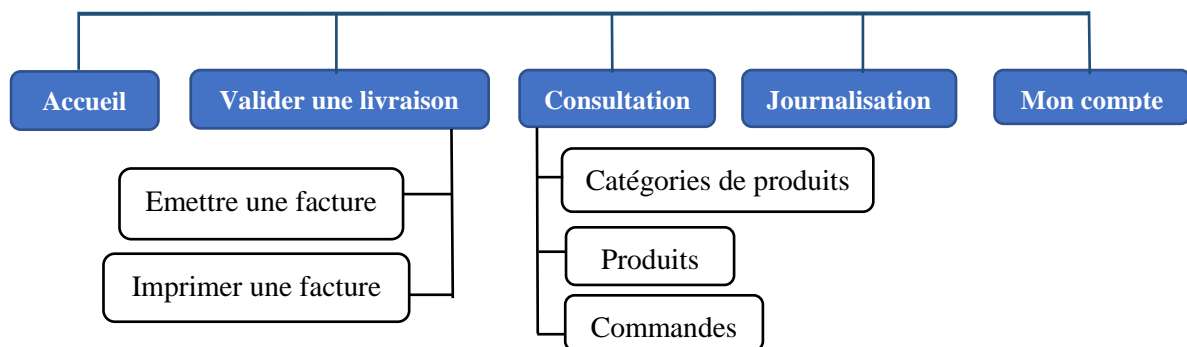


Figure 40: Les onglets sur l'interface du profil « livreur »

✓ Onglets du profil « administrateur »

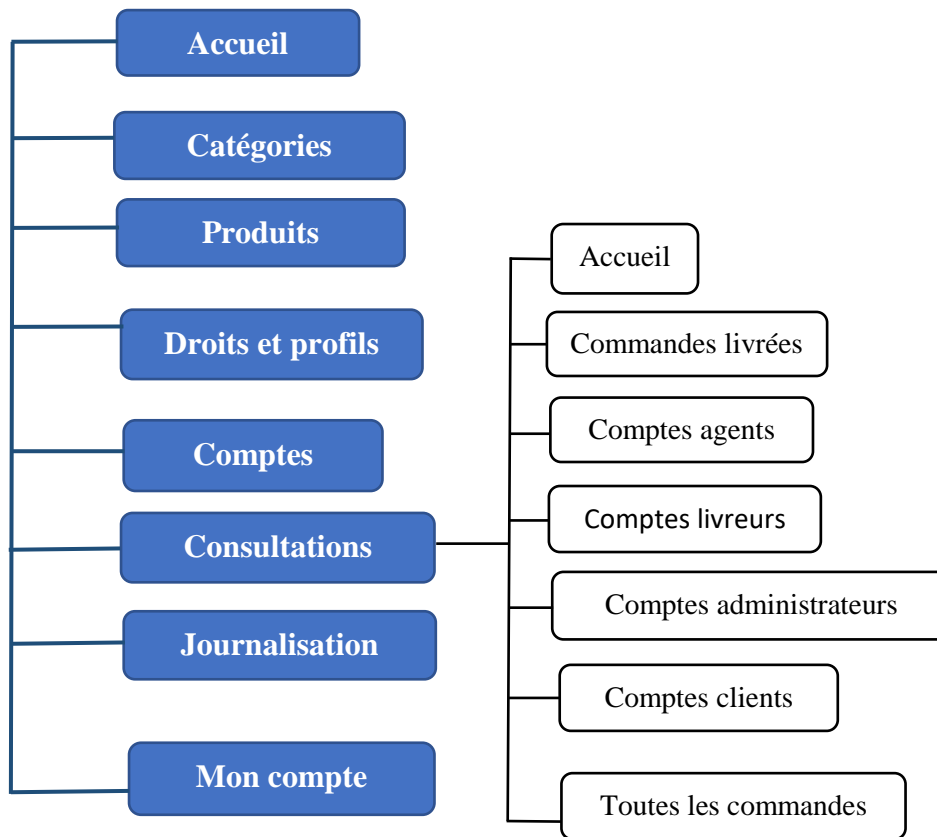


Figure 41: Onglets de l'interface du profil « administrateur »

✓ Onglets de la page d'achat de l'application



Figure 42: Onglets de la page d'achat de l'application

✓ Onglets du profil « agent »

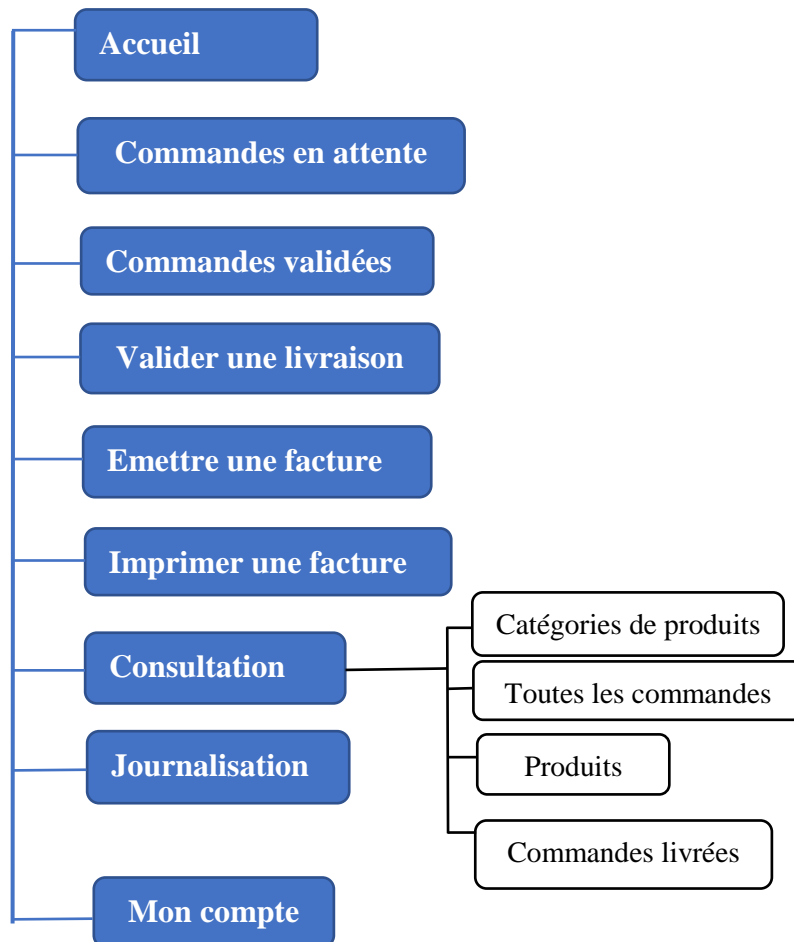


Figure 43: Onglets de l'interface du profil « agent »

✓ Onglets de la page compte du Client



Figure 44: Onglets de la page compte du client

3. Script de création de la base de données

Ce script a été généré à partir du Modèle Physique de Données de notre analyse, grâce à l'outil de modélisation PowerAMC 15.1. Voici présenté le script de création de notre base de données :

```
CREATE DATABASE SHOPONLINE ;

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time_zone = "+00:00";

-- Base de données : `shoponline`

-- -----

-- Structure de la table `administrateur`

DROP TABLE IF EXISTS `administrateur`;

CREATE TABLE IF NOT EXISTS `administrateur` (

  `id` bigint(20) NOT NULL AUTO_INCREMENT,

  `NumTel` varchar(15) NOT NULL,

  `adresse` varchar(30) NOT NULL,

  `compteid` bigint(20) NOT NULL,

  `designation` varchar(20) NOT NULL,

  `nom` varchar(20) NOT NULL,

  `prenom` varchar(40) NOT NULL,

  `sexe` varchar(8) NOT NULL,

  PRIMARY KEY (`id`),

  UNIQUE KEY `UK_4tqkc bu2ojbodhcmxh288wfj` (`NumTel`),

  UNIQUE KEY `UK_b98igeeemoic5cru2hdn4qir` (`compteid`),

  UNIQUE KEY `UK_9uj3iy3mbdpl1l0pjmux0halv` (`nom`),

  UNIQUE KEY `UK_3vsr0tii6g0q9p7kn37hnhur1` (`prenom`)

) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- Structure de la table `agent`

DROP TABLE IF EXISTS `agent`;

CREATE TABLE IF NOT EXISTS `agent` (

  `id` bigint(20) NOT NULL AUTO_INCREMENT,
```

```
`NumTel` varchar(15) NOT NULL,
`adresse` varchar(30) NOT NULL,
`compteid` bigint(20) NOT NULL,
`dateEmbauche` datetime NOT NULL,
`designation` varchar(255) DEFAULT NULL,
`nom` varchar(20) NOT NULL,
`prenom` varchar(40) NOT NULL,
`sexe` varchar(8) NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `UK_ec7yuhntcp3aqh9i63dghkmsx` (`NumTel`),
UNIQUE KEY `UK_r72co3e7lks9yhnvgerp5vejh` (`compteid`),
UNIQUE KEY `UK_4bvf6voa1j1aw3cd9502hsk5` (`nom`),
UNIQUE KEY `UK_aavjt7p6udbmhvjecl771s7xm` (`prenom`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- Structure de la table `categorie`
DROP TABLE IF EXISTS `categorie`;
CREATE TABLE IF NOT EXISTS `categorie` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `description` varchar(255) DEFAULT NULL,
  `libCategorie` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_fsi8key9eqpp9cj41e73uut1t` (`libCategorie`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

-- Structure de la table `client`
DROP TABLE IF EXISTS `client`;
CREATE TABLE IF NOT EXISTS `client` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `NumTel` varchar(15) NOT NULL,
  `adresse` varchar(150) NOT NULL,
  `compteid` bigint(20) NOT NULL,
  `email` varchar(20) NOT NULL,
  `nom` varchar(20) NOT NULL,
```

```
`prenom` varchar(40) NOT NULL,
`sexe` varchar(8) NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `UK_c44mgi4u9r67qco1igsilspbo` (`NumTel`),
UNIQUE KEY `UK_gwo27kejv1kc32ber9bvpmd3b` (`compteid`),
UNIQUE KEY `UK_bfgjs3fem0hmjvhv80158x29` (`email`),
UNIQUE KEY `UK_edver7gtbfldp50jtkypbtr89` (`nom`),
UNIQUE KEY `UK_d03eh5lo4jua7h141etdwfv2p` (`prenom`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

-- Structure de la table `commande`
DROP TABLE IF EXISTS `commande`;
CREATE TABLE IF NOT EXISTS `commande` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `adresseFacturation` varchar(255) NOT NULL,
  `codeCommande` varchar(255) DEFAULT NULL,
  `dateCom` datetime NOT NULL,
  `livre` tinyint(1) NOT NULL,
  `valide` tinyint(1) NOT NULL,
  `client_id` bigint(20) DEFAULT NULL,
  `livraison_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_nxmcovc3m6dxgwmij082yy6nf` (`client_id`),
  KEY `FK_h0tawbtnw0j6nntmciqk9g1qf` (`livraison_id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Structure de la table `compte`
DROP TABLE IF EXISTS `compte`;
CREATE TABLE IF NOT EXISTS `compte` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `password` varchar(255) DEFAULT NULL,
  `username` varchar(255) DEFAULT NULL,
  `profil_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
```

```
KEY `FK_501scyd2yqkhvxjnv2vlvcwg1` (`profil_id`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
-- Structure de la table `droit`
DROP TABLE IF EXISTS `droit`;
CREATE TABLE IF NOT EXISTS `droit` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `libelleDroit` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_njnnbrbhhweexp2hpu90y1qjd` (`libelleDroit`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
-- Structure de la table `lignecommande`
DROP TABLE IF EXISTS `lignecommande`;
CREATE TABLE IF NOT EXISTS `lignecommande` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `quantite` int(11) NOT NULL,
  `commande_id` bigint(20) DEFAULT NULL,
  `produit_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_33ye4opkutvr4a0xldbvpysd` (`commande_id`),
  KEY `FK_m8tyb43j0iib0pgqufams726d` (`produit_id`)
) ENGINE=InnoDB AUTO_INCREMENT=40 DEFAULT CHARSET=utf8;
-- Structure de la table `livraison`
DROP TABLE IF EXISTS `livraison`;
CREATE TABLE IF NOT EXISTS `livraison` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `adrLiv` varchar(255) NOT NULL,
  `dateLiv` datetime DEFAULT NULL,
  `livre` tinyint(1) NOT NULL,
  `agent_id` bigint(20) DEFAULT NULL,
  `livreur_id` bigint(20) DEFAULT NULL,
  `type_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
```

```
KEY `FK_pkm4gn8rx2xoew5udnpfanxxb` (`agent_id`),
KEY `FK_oyydpv1avhtaurkyvia0vtv3` (`livreur_id`),
KEY `FK_55pmgkhd1kqqr6hq8vejvfj7h` (`type_id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Structure de la table `livreur`
DROP TABLE IF EXISTS `livreur`;
CREATE TABLE IF NOT EXISTS `livreur` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `NumTel` varchar(15) NOT NULL,
  `adresse` varchar(30) NOT NULL,
  `compteid` bigint(20) NOT NULL,
  `dateEmbauche` datetime NOT NULL,
  `designation` varchar(255) DEFAULT NULL,
  `nom` varchar(20) NOT NULL,
  `prenom` varchar(40) NOT NULL,
  `sexe` varchar(8) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_b18gt2e9t097ucx7wv703nxjd` (`NumTel`),
  UNIQUE KEY `UK_7mhjn5jaqsv8ej329ltwo11aw` (`compteid`),
  UNIQUE KEY `UK_qjpp5ptvvdkg0hf0irgusrrw` (`nom`),
  UNIQUE KEY `UK_tfutrrmt73k7p143oepb5u07u` (`prenom`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

-- Structure de la table `produit`
DROP TABLE IF EXISTS `produit`;
CREATE TABLE IF NOT EXISTS `produit` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `description` varchar(255) NOT NULL,
  `imageProd` varchar(255) NOT NULL,
  `libProduit` varchar(255) NOT NULL,
  `prix` int(11) NOT NULL,
  `categorie_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
```

```
UNIQUE KEY `UK_r5u3bwncs4n50wav4mbv8h0kh` (`description`),
UNIQUE KEY `UK_h5asmgtiaa0qm56nrx6k4kj56` (`libProduit`),
KEY `FK_9to2ll6a02guk5wqqmtugqlq9` (`categorie_id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
-- Structure de la table `produitprix`
DROP TABLE IF EXISTS `produitprix`;
CREATE TABLE IF NOT EXISTS `produitprix` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `date_Prix` datetime NOT NULL,
  `prix` int(11) NOT NULL,
  `produit_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_piq3hg2jee8fh7lusc91rlh5a` (`produit_id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=utf8;
-- Structure de la table `profil`
DROP TABLE IF EXISTS `profil`;
CREATE TABLE IF NOT EXISTS `profil` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `libelleProfil` varchar(20) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_feql11qmqvbtjfe01a7hifmpx` (`libelleProfil`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
-- Structure de la table `profildroit`
DROP TABLE IF EXISTS `profildroit`;
CREATE TABLE IF NOT EXISTS `profildroit` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `droit_id` bigint(20) DEFAULT NULL,
  `profil_id` bigint(20) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `FK_d6bggtick5kkse07kmib82b9y` (`droit_id`),
  KEY `FK_8w0xb6qnvc5rr5gajk7cvlhe3` (`profil_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----  
-- Structure de la table `reglement`  
DROP TABLE IF EXISTS `reglement` ;  
CREATE TABLE IF NOT EXISTS `reglement` (  
  `id` bigint (20) NOT NULL AUTO_INCREMENT,  
  `dateReg` datetime DEFAULT NULL,  
  `commande_id` bigint (20) DEFAULT NULL,  
  `typereglement_id` bigint (20) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FK_8w8ysyj5imef5iquxs6e9nolk` (`commande_id`),  
  KEY `FK_6ilyw19ay9477lhto008i6ujw` (`typereglement_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8 ;  
-- Structure de la table `typelivraison`  
DROP TABLE IF EXISTS `typelivraison`;  
CREATE TABLE IF NOT EXISTS `typelivraison` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `descriptif` varchar(255) DEFAULT NULL,  
  `libelle` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `UK_ojm0qk4er11h2rh8gyafdcwk7` (`libelle`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;  
-- Structure de la table `typereglement`  
DROP TABLE IF EXISTS `typereglement` ;  
CREATE TABLE IF NOT EXISTS `typereglement` (  
  `id` bigint (20) NOT NULL AUTO_INCREMENT,  
  `designation` varchar (255) DEFAULT NULL,  
  `libelleType` varchar (255) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `UK_g0fyvmref7nn9ro5k69p8j7lg` (`libelleType`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 ;
```

4. Quelques marques de saisies et de codes sources de l'application

a. Présentation de quelques codes sources de l'application

```

        &{flash.success}
    </p>
</div>
#{/if}

<div class="col-sm-6 col-sm-offset-3">
    <div class="card card-responsive" style="padding:60px;background-color: white;margin-top:15%;">
        <div class="card-header text-center" style="margin-top:0px;padding-top:0px;font-size:17px">
            <h3 style="color:royalblue;font-size:40px;font-family:-webkit-body;margin-top:0px">Authentification</h3><br/>
        </div>
        <div class="card-body">
            <div class="col-sm-12">
                <label for="Username"><b>Login</b></label>
                <input type="text" placeholder="Entrez votre nom d'utilisateur" name="username"
                    id="username" value="{flash.username}" autocomplete="off">
            </div>
            <div class="col-sm-12">
                <label for="pwd"><b>Mot de passe</b></label>
                <input type="password" placeholder="Entrez votre mot de passe" name="password"
                    id="password" value="" autocomplete="off">
            </div><br/>
            <div class="row form-group text-center">
                <div class="col-sm-12">
                    <div class="col-sm-6 pull-right"><button type="reset" class="btn btn-danger"
                        style="font-size:14px">Annuler</button> </div>
                    <div class="col-sm-6 pull-right"><button type="submit" id="signin" class="btn btn-success"
                        style="font-size:14px">Se connecter</button> </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

Figure 45: Une partie du code source de la page d'authentification

```

//Enregistrement d'un client
public static void saveClient(String nom, String prenom, String adresse, String sexe,
    String NumTel, String email, String login, String pwd, String lieuLiv,
    Long idLiv, List<Produit> prod, List<Integer> qu, Long idPaielement, String retrait, String domicile) {

    models.Compte c = models.Compte.find("username", login).first();
    if (c != null) {
        models.Client cl = models.Client.find("compteid", c.id).first();
        if (cl != null) {
            models.Livraison li = models.Livraison.findById(idLiv);
            models.Reglement re = models.Reglement.findById(idPaielement);

            try {
                /*J'enregistre la commande*/
                Commande commande = new Commande();
                try {
                    commande.dateCom = new Date();
                    commande.client = null;
                    commande.save();

                } catch (Exception e) {
                }
                /*J'enregistre la ligne de commande*/
                try {
                    for (Produit id : prod) {
                        LigneCommande ligne = new LigneCommande();
                        ligne.commande = models.Commande.findById(commande.id);
                        ligne.produit = models.Produit.findById(id);
                        ligne.save();
                        for (Integer i : qu) {
                            //ligne.quantite=qu;
                        }
                    }
                }
            }
        }
    }
}

```

Figure 46: Une partie du code source du Contrôleur permettant l'enregistrement d'un client


```

<div class="col-sm-12">
  <div class="bs-example form-group">
    <div class="row" style="display: flex; justify-content: space-between; flex-direction: row; font-size: 18px">
      <div class="test">Produit</div>
      <div class="test">Description</div>
      <div class="test">Quantité</div>
      <div class="test">Prix</div>
    </div>
    <#(list items:produits, as : 'produit')>
    <div class="row">
      <div class="card thumbnail" style="width:100%">
        <div class="card-body">
          <div class="col-sm-3">
            
          </div>
          <div class="col-sm-3">
            <h5 style="margin-left:40%;font-size: 16px">{produit.description}</h5>
          </div>
          <div class="col-sm-3">
            <h5 style="margin-left: 50%;font-size: 16px">{quantites[i]}</h5>
          </div>
          <div class="col-sm-3">
            <h5 style="margin-left: 26%;font-size: 16px">{produit.prix}</h5>
          </div>
          <div>
            <span class="glyphicon glyphicon-remove" style="margin-left:70%;margin-top:-2%></span>
          </div>
        </div>
      </div>
    </div>
    <span style="display:none;">{i++}</span>
  </div>
</div>
</div>

```

Figure 47: Une partie de la Vue permettant d'afficher le récapitulatif du panier du client

```

<div class="table-responsive">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th nowrap>Numero de la commande</th>
        <th nowrap>Date de la commande</th>
        <th nowrap>Adresse de facturation</th>
        <th nowrap>Client</th>
        <th nowrap>Etat</th>
        <th colspan="2">Actions</th>
      </tr>
    </thead>
    <tbody>
      <form>
        <#(list items:commandes , as: 'commande')>
        <tr>
          <td nowrap>{commande.id}</td>
          <td nowrap>{commande.dateCom}</td>
          <td nowrap>{commande.adresseFacturation}</td>
          <td nowrap>{commande.client}</td>
          <td nowrap>Validée</td>
          <td>
            <button class="btn btn-info detailCommande" data-ligne="" data-id="{commande.id}" >
              <span class="glyphicon glyphicon-zoom-in"></span> Détails</button>
            <input value="" id="code" name="code" hidden="true"/>
            <button type="submit" class="btn btn-primary"><span class="glyphicon glyphicon-share-alt"></span> Facturer</button>
          </td>
        </tr>
      </form>
    </tbody>
  </table>
</div>

```

Figure 48: Une partie de la vue affichant à un agent les commandes livrées

b. Présentation de quelques marques de saisies de l'application

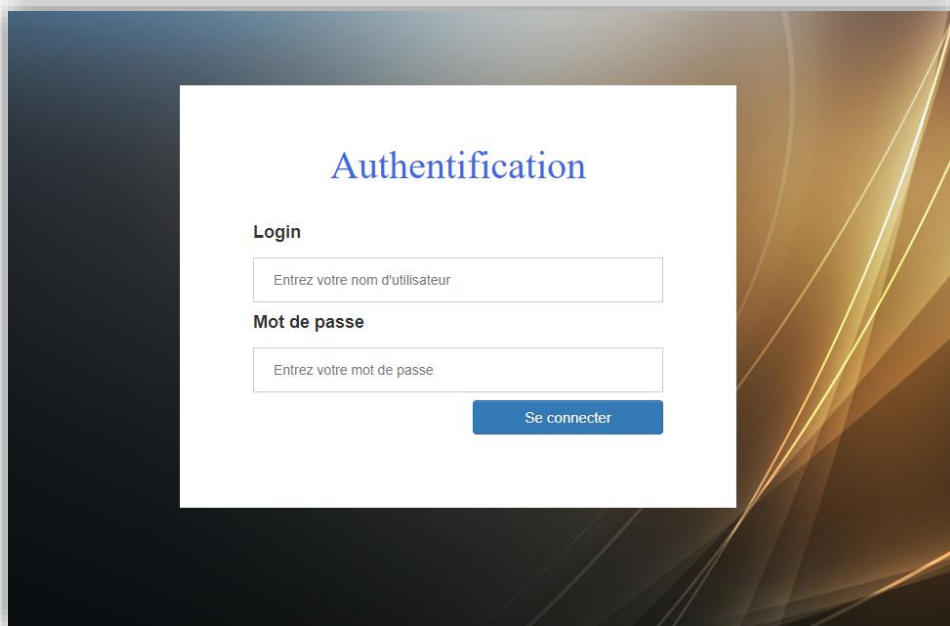


Figure 49: Page d'authentification de l'application

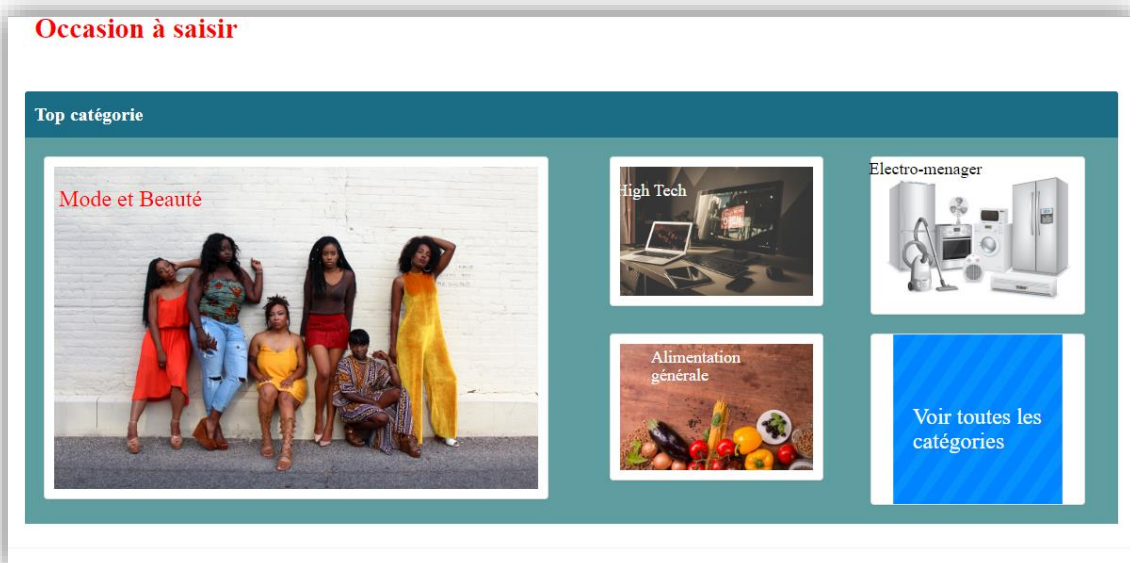


Figure 50: Page d'accueil de l'application

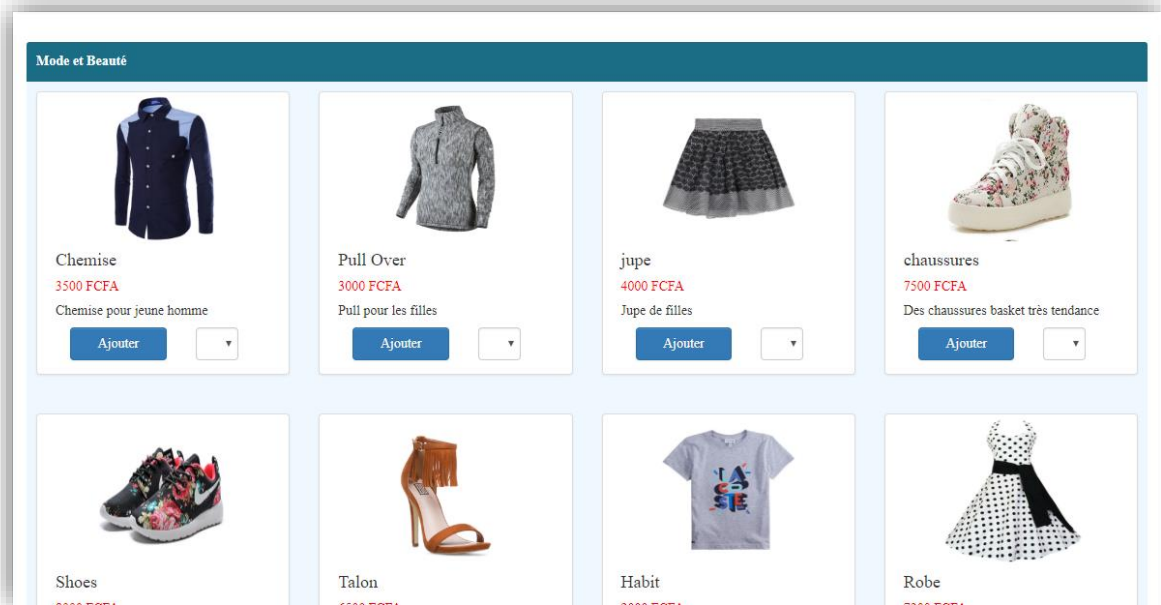


Figure 51: Une capture de la page qui présente les produits par catégorie

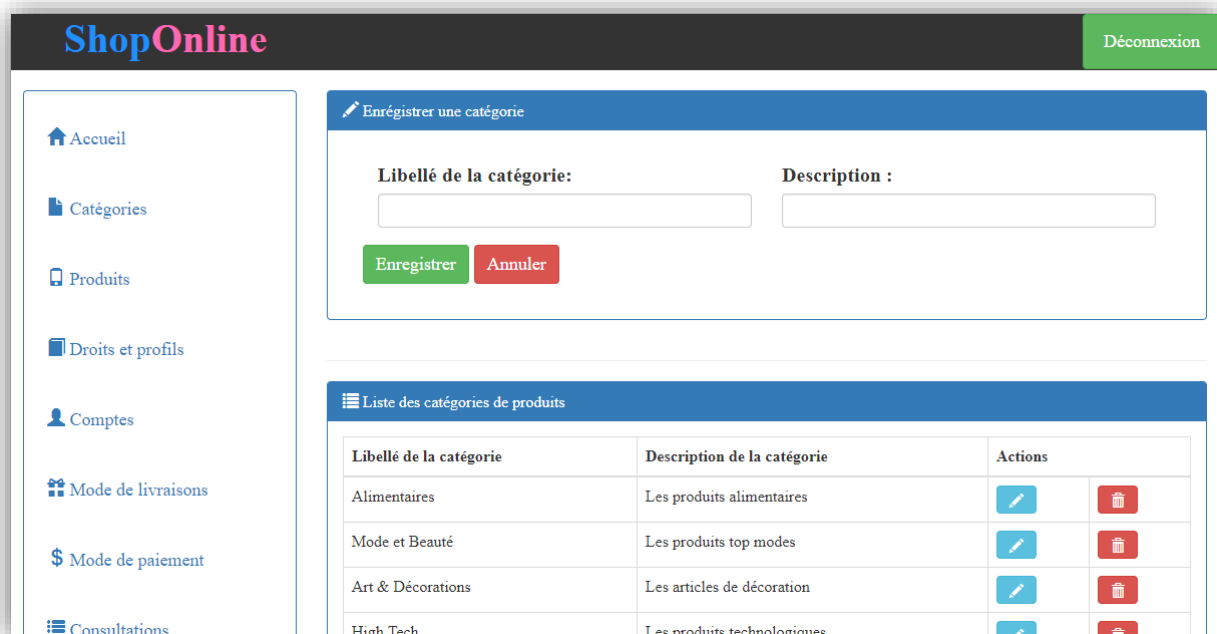


Figure 52: Capture de l'interface administrateur qui permet l'enregistrement des catégories de produits

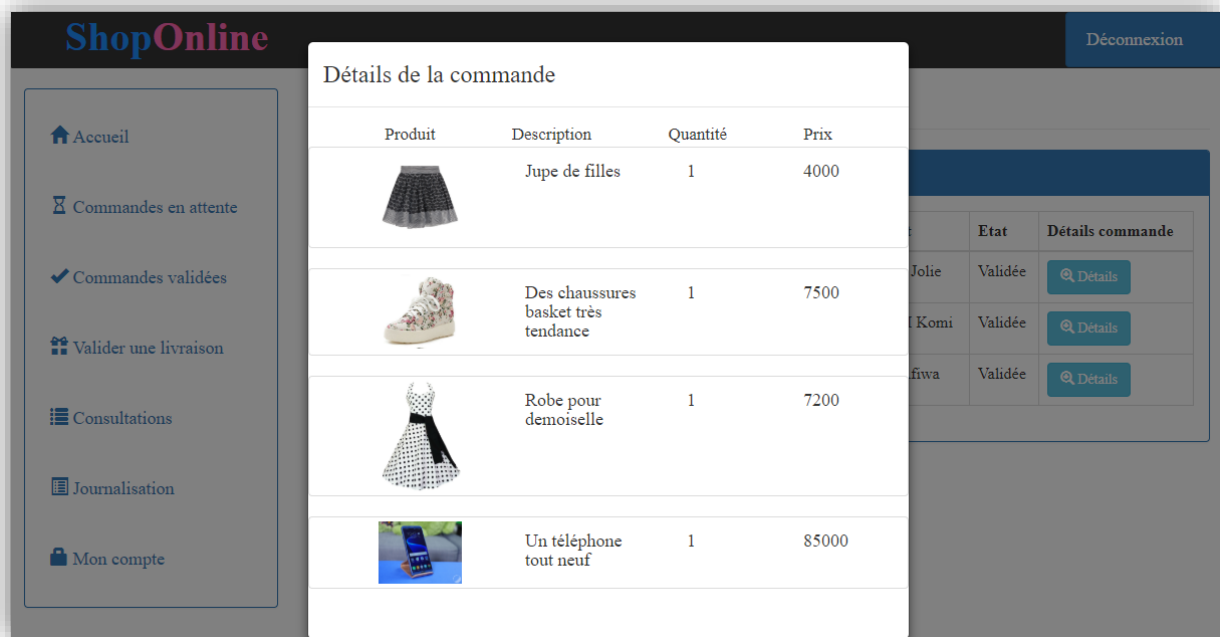


Figure 53: Capture de l'interface agent qui affiche les détails d'une commande

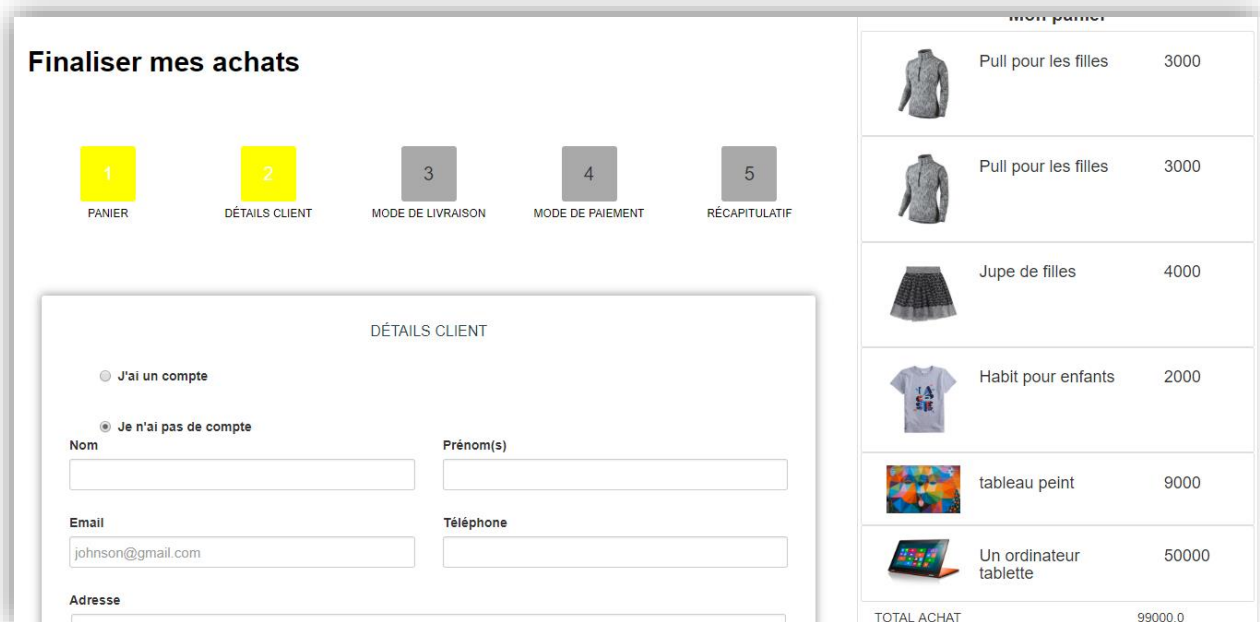


Figure 54: Capture du processus d'achat qu'effectue un client

DÉTAILS CLIENT

☐ J'ai un compte
☒ Je n'ai pas de compte

Nom

Prénom(s)

Email

Téléphone

Adresse

Sexe

Login

Mot de passe

Confirmation du mot de passe





	Jupe de filles	4000
	Habit pour enfants	2000
	tableau peint	9000
	Un ordinateur tablette	50000
TOTAL ACHAT		99000.0
TOTAL A PAYER		99000.0

Figure 55: Capture du formulaire d'enregistrement d'un nouveau client

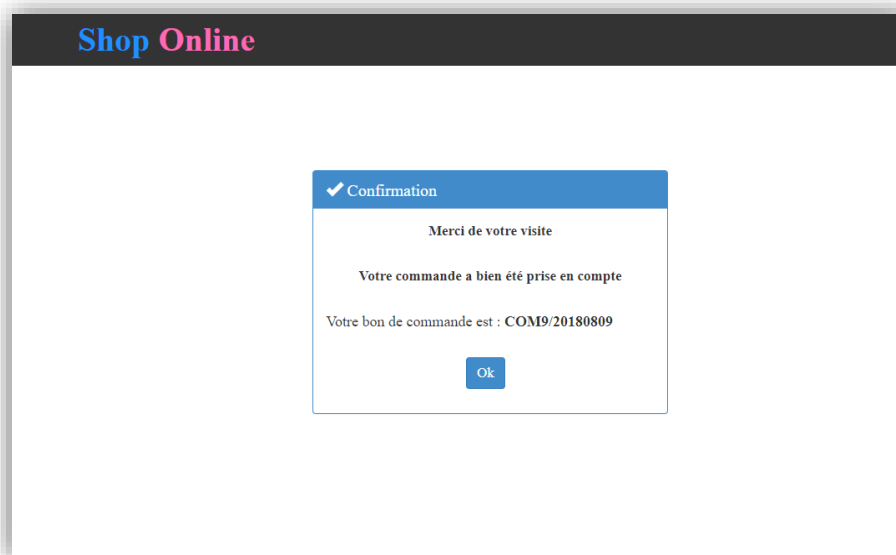


Figure 56: capture de la page de confirmation de la commande du client

5. Quelques états et statiques

13/08/2018

Facture

ShopOnline

Date de la commande : 2018-08-06 17:30:47.0

Adresse de facturation : Lomé-Togo

Client : CLAUDINE Claudia

Recepteur de la commande:

Produit	Quantité	Prix	Total
Pull pour les filles	1	3000	3000
Chassures à talons hauts	1	6500	6500
Un ordinateur tablette	1	50000	50000
Un téléphone tout neuf	1	85000	85000
TOTAL :			144500

Figure 57: Capture de la facture d'un client

CONCLUSION

Somme toute, ce stage de deux (02) mois, effectué au sein de TicBuilder SARL, nous a vraiment permis de mettre en pratique toutes nos connaissances acquises durant nos deux (02) premières années de formation à l'IAI-TOGO, tout en étant confronté aux difficultés réelles du monde du travail.

L'objectif du projet que nous avons réalisé durant le stage, était de mettre en place une plateforme de vente de produits en ligne. Nous avons donc étudié le problème posé, trouvé des solutions, tiré la meilleure et l'avons implémenté. Ce qui nous a fait accroître nos connaissances en analyse et programmation.

Dans la première partie titrée « Rapport d'insertion », nous avons décrit notre insertion et avons présenté notre thème de stage.

Dans la seconde partie titrée « Rapport de pré-programmation », nous avons privilégié la modélisation. Avec la modélisation, il a été plus facile de comprendre et de dessiner ce que serait notre système, afin de l'implémenter techniquement.

Dans la troisième partie titrée « Réalisation et mise en œuvre », nous avons entamé la phase de développement de notre système.

Le plus fascinant est l'intégration professionnelle qui en a découlé, ce qui nous fait adresser des encouragements aux responsables de l'IAI-TOGO.

Par la suite et vue l'évolution de la technologie mobile, il serait intéressant d'envisager la version mobile de cette application, afin d'améliorer l'accessibilité à nos produits, de fidéliser les clients, de permettre une ouverture facile, de permettre d'être déconnecté d'internet.

A cet effet, nous n'hésitons pas à offrir nos services pour le développement de la version mobile de cette application, durant notre stage de fin de formation à la fin de l'année académique 2018-2019.

➤ Bibliographie

✓ Ouvrages

- Play-guide par Romain Linsolas
- Play-Framework-Cookbook par Alexander Reelsen

✓ Notes de cours

- Cours d'UML de M. KPEGLO (Deuxième année, année académique :2017-2018) ;
- Cours d'implémentation des bases de données de M. KEOULA (Deuxième année, année académique 2017-2018).
- Java de Monsieur NOUKAFOU et Monsieur BAKPESSI (Deuxième année, année académique : 2017-2018)

✓ Sites web consultés

- <https://www.memoireonline.com> (13-06-2018 à 11h20)
- <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc31018.1610/doc/html/rad1232632576301.html> (22/06/2018 à 11h18)
- <https://fr.scribd.com/doc/57496887/U-M-L-Methode-2TUP> (27-06-2018 à 15h00)
- <https://laurent-audibert.developpez.com./cours-UML> (18-07-2018 à 14h30)
- www.UML-sysml.org (18-07-2018 à 14h 47)

✓ Anciens mémoires consultés

- Mémoire de Mlle Florence N'KPEDJI (2016-2017)
- Mémoire de Mr. AGOTSI Komi Gédéon Cyrille (2015-2016)
- Mémoire de Mr. KADABA Soklibou Joël (2016-2017)
- Mémoire de Mr. KOUASSI A. Gylas (2014-2015)

➤ Documents annexes

✓ Cours UML5

➤ **Table des matières**

INTRODUCTION	2
I. PRESENTATIONS	4
1. Présentation de l'IAI-TOGO	4
2. Présentation du cabinet « TicBuilder »	5
II. THEME DE STAGE	8
1. Présentation du sujet.....	8
2. La problématique du sujet	8
3. Intérêt du sujet	9
a. Objectifs	9
b. Résultats attendus	9
I. ETUDE DE L'EXISTANT	11
II. CRITIQUES DE L'EXISTANT.....	11
III. PROPOSITIONS DE SOLUTIONS	12
A. Première solution.....	12
✓ Avantages	12
✓ Inconvénients.....	12
B. Deuxième solution.....	13
✓ Avantages	13
✓ Inconvénients.....	13
C. Troisième solution	14
✓ Avantages	14
✓ Inconvénients.....	14
IV. EVALUATION FINANCIERE DES SOLUTIONS.....	15
A- Première solution.....	15
1. Coût d'hébergement	15
2. Coût matériel	15
3. Coût total	15
B. Deuxième solution.....	16
1. Coût d'hébergement	16
2. Coût humain	16
3. Coût matériel	16
4. Coût total	17
C. Troisième solution.....	17

1. Coût d'hébergement	17
2. Coût humain	17
3. Coût matériel	18
4. Coût total	18
V. CHOIX DES SOLUTIONS	19
A. Spécifications fonctionnelles du projet	19
B. Spécification de sécurité.....	20
C. Spécifications techniques	20
VI. PLANNING PREVISIONNEL DE REALISATION	21
VII. PRESENTATION DE LA METHODE D'ANALYSE	24
A. Le langage de modélisation.....	24
I- Présentation	24
II- UML est un formalisme.....	24
B. Le processus de modélisation	25
VIII. PRESENTATION DE L'OUTIL DE MODELISATION.....	27
A. Qu'est-ce que PowerAMC ?.....	27
B. Modélisation avec PowerAMC.....	28
IX. ETUDE DETAILLEE DE LA SOLUTION.....	29
A. Vue de modélisation	29
Diagramme de contexte statique	29
B. Vue de cas d'utilisation	30
1. Le diagramme de cas d'utilisation.....	30
2. Diagramme de cas d'utilisation par acteur	32
C. Vue des processus	36
1. Description textuelle de quelques cas d'utilisation	36
2. Diagramme de séquence.....	47
3. Diagramme d'activité.....	56
D. VUE LOGIQUE	60
Diagramme de classe-entité.....	60
E. Vue de déploiement	61
Diagramme de déploiement.....	61
.....	62
I. MISE EN ŒUVRE	63
1. Choix matériels	63
2. Choix logiciels	63
a) Outils d'implémentation de la base de données	63

b) Outils de programmation et de développement	64
3. Sécurité de l'application.....	68
a. Sécurité au niveau de l'hébergeur de l'application web	68
b. Sécurité au niveau de l'application.....	68
II. PRESENTATION DE L'APPLICATION	69
1. Présentation	69
2. Architecture	69
3. Script de création de la base de données	72
4. Quelques marques de saisies et de codes sources de l'application	79
a. Présentation de quelques codes sources de l'application	79
b. Présentation de quelques marques de saisies de l'application.....	81
5. Quelques états et statiques.....	85
CONCLUSION.....	86
➤ Bibliographie	87
✓ Ouvrages.....	87
✓ Notes de cours	87
✓ Sites web consultés.....	87
✓ Anciens mémoires consultés	87
➤ Documents annexes	87
➤ Table des matières	88