



RAPPORT DE STAGE

8 septembre 2014 au 18 mars

2015

Concepteur Développeur
Informatique

Vanac KEU



Numérique Technologie . Protection

REMERCIEMENTS

Je tiens à remercier :

- Monsieur **Julien GUILLARDEAU** pour m'avoir permis d'effectuer mon stage dans son entreprise et surtout pour sa disponibilité pour répondre à mes questions.
- Monsieur **Pascal BUGUET** pour sa grande disponibilité et son accompagnement tout le long de cette formation.
- **M2I Formation** pour avoir tout mis en œuvre pour qu'on puisse suivre cette formation sans problème.
- A mes **collègues et amis** avec qui j'ai passé ces 6 mois de formation, pour leur aides, soutiens et amitiés
- A ma **famille** qui m'ont permis de suivre cette formation et bien sûr soutenu.

SOMMAIRE

Remerciements	2
Sommaire	3
Chapitre 1 : Présentation	5
1. L'entreprise.....	6
1.1 Identification	6
1.2 Activité	6
2. Le stagiaire	6
2.1 Le profil	6
2.2 Le tuteur de stage	6
3. Analyse.....	7
3.1 Le PROJET	7
Chapitre 2 : Définition du projet et étude préalable.....	8
1. Description de l'étude.....	9
1.1 Analyse de l'existant	9
1.2 Principe et objectifs	10
1.3 Diagramme de cas d'utilisation	10
1.5 L'enveloppe budgétaire	12
1.6 Le délais	12
1.7 Gestions des droits	13
1.8 DESCRIPTION fonctionnelle	15
Chapitre 3 : Conception	18
1. Maquette Interface Humain Machine.....	19
1.1 Front-office	19
1.2 Back-office	23
2. Diagrammes.....	32
2.1 Diagramme de navigation	32
2.2 Diagramme d'ACTIVITE	36
Chapitre 4 : Implémentation	43
1. Diagramme de classes du domaine.....	44
1.1 Analyse	45
1.3 Les dépendances fonctionnelles	48
1.4 Le graphe des dependances fonctionnelles	50
1.5 Le diagramme de classe	52
1.6 Règles de passage du modèle objet au modèle relationnel.	54
1.7 LE diagramme de classes totales	55

2. Base de données.....	56
2.1 Introduction	56
2.2 Création de la base de données	56
2.3 Création d'une table	57
2.4 Modele de la base de donnees	58
3. Architecture n-Tiers	59
4. L'arborescence des fichiers	62
5. Codes.....	63
5.1 Fichier config	63
5.2 Connexion à la base de données	64
5.3 Authentification et gestions d'erreurs	64
5.4 Ma route	71
5.5 Recherche du stock via Ajax et jquery	73
5.6 Diagramme de déploiement	79
6. Les logiciels utilisés	80
7. Conclusion.....	81
Bilan	81
LE stage.....	81
PROJET FUTUR.....	81
8. Abstract.....	82
Results.....	82
INTERNSHIP	82
FUTURE PROJECT.....	82
9 . Annexe	83
Autres Descriptions détaillées	83
Diagrammes de navigation	84
Diagramme d'activite	85
La table des illustrations	93

CHAPITRE 1 : PRÉSENTATION

1. L'ENTREPRISE

1.1 IDENTIFICATION

L'entreprise NTPROTEC crée 2011 se situe au 9, Rue des Maraîchers 95130 FRANCONVILLE LA GARENNE.

1.2 ACTIVITÉ

NTPROTECT est une entreprise spécialisée dans la sécurité électronique. Celle-ci a été fondé par 2 artisans diplômés en électronique audiovisuel, informatique et spécialisé dans la sécurité électronique.

"Vous faire bénéficier de son avance technologique tout en conservant un budget équivalent voir inférieur à celui d'un système traditionnel."

2. LE STAGIAIRE

2.1 LE PROFIL

Après avoir travailler dans la restauration pendant de nombreuses années, j'ai décidé de me réorienter vers l'informatique. En 2014, j'ai obtenu un BTS développeur multimédia à l'AFCI NEWSOFT à Montreuil.



Suite à ce diplôme, je souhaitais parfaire mes connaissances informatique en intégrant la formation **CONCEPTEUR DEVELOPPEUR INFORMATIQUE** à M2I Formation. Cette formation a duré 6 mois avec 2 mois de stage en entreprise inclus (8 septembre 2014 au 18 mars 2015).

2.2 LE TUTEUR DE STAGE

Monsieur GUILLARDEAU Julien a joué le rôle de directeur de stage pendant la période d'application en entreprise effectuée chez NTPROTEC dont il est l'un des associés.

Ses compétences et son expérience en gestion de projet informatique lui ont permis de superviser ce projet.

3. ANALYSE

3.1 LE PROJET

L'entreprise souhaite développer sa stratégie de communication.



La méthode choisie est : **la création d'un site internet.**

Les objectifs du site seront avant tout :

- De faire connaître l'entreprise
- Toucher de nouveaux clients.
- Faire vendre ses services.

Les buts du site seront :

- Augmenter son chiffre d'affaire
- Diminuer ses dépenses

Aujourd'hui, la quasi-totalité des internautes recherchent un prestataire sur internet.

CHAPITRE 2 : DÉFINITION DU PROJET ET ÉTUDE PRÉALABLE

1. DESCRIPTION DE L'ÉTUDE

1.1 ANALYSE DE L'EXISTANT

Je suis actuellement le seul stagiaire à NT Protec. Un ancien stagiaire avait réalisé le site internet statique mais celui-ci ne correspond plus aux attentes de l'entreprise.

Les principaux facteurs :

- Look dépassé
- Technologie obsolètes (non mobile)
- Pas d'accès aux réseaux sociaux



Figure n° 1 : Ancien site NT Protec

L'ancien site ne disposait qu'un formulaire de contact. Il n'avait pas de back-office non plus.

Le nouveau site sera donc composé d'un front-office et d'un back-office

Le front-office est l'interface visible par l'internaute.

Le back-office est l'interface non visible par le client, il permet la gestion d'un site web.

1.2 PRINCIPE ET OBJECTIFS

L'internaute : il a la possibilité de contacter l'entreprise par 2 moyens :

- Le formulaire de demande de devis
- Le formulaire de contact

Le gérant : Il aura la possibilité de gérer ces demandes faites par les internautes via son back-office. Il peut donc gérer les demandes de devis et gérer les contacts.



1.3 DIAGRAMME DE CAS D'UTILISATION



C'est le diagramme de cas d'utilisation qui permettent de donner une vision globale du comportement fonctionnel du système.

- L'internaute:

La création du site et surtout le front-office va permettre à l'internaute de naviguer et découvrir l'entreprise afin d'effectuer une demande de devis pour les comparer à d'autres prestataires. Il peut aussi s'inscrire à une newsletter pour avoir des informations régulières par l'entreprise.

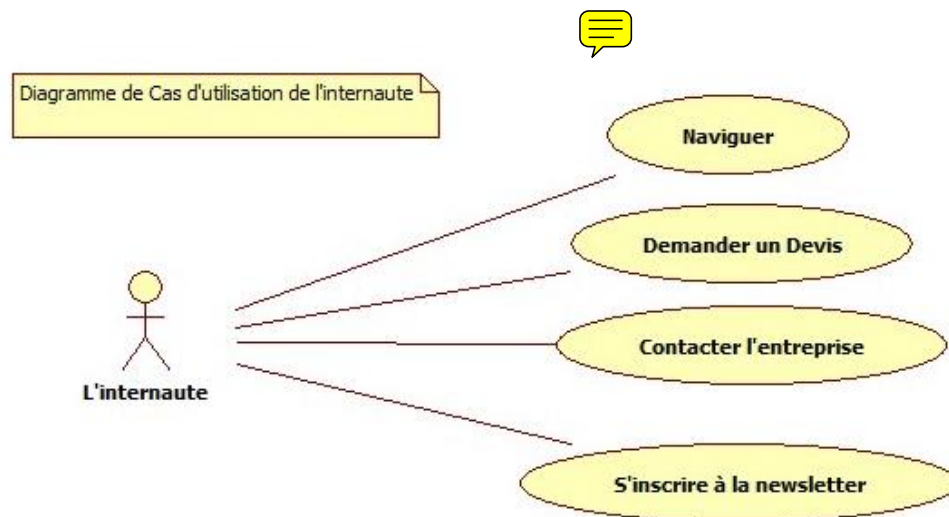


FIGURE N° 2 : DIAGRAMME DE CAS D'UTILISATION INTERNAUTE

- L'entreprise

La création d'une interface pour le gérant c'est à dire back-office va lui permettre de gérer les demandes des internautes et également gérer son stock. Il peut informer des nouveautés ou autres via la newsletter.

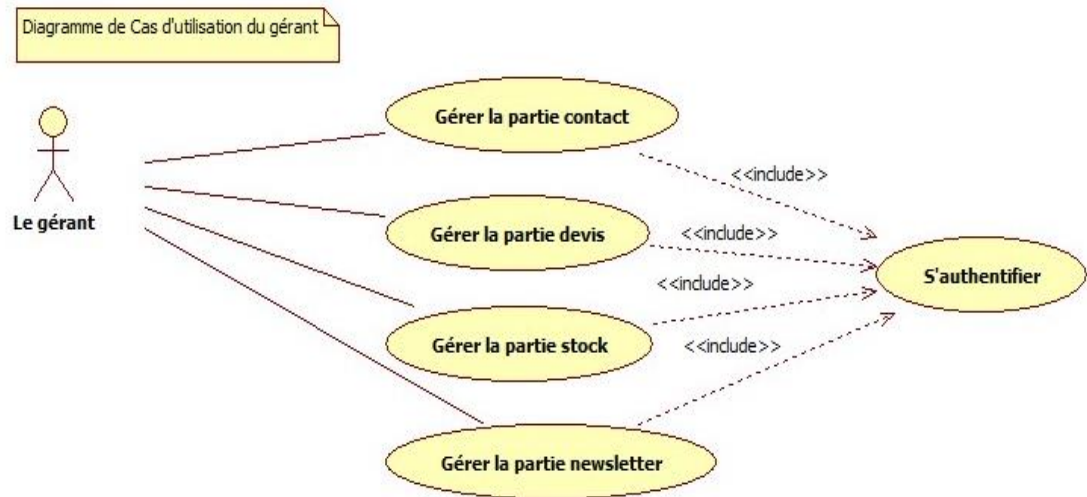


FIGURE N° 3 : DIAGRAMME DE CAS D'UTILISATION ENTREPRISE

1.4 PÉRIMÈTRE

Nous allons nous concentrer sur le site internet.

Le Front-office sera responsive mobile mais pas le back-office.

En imaginant par la suite une possible évolution vers des applications mobiles des 2 interfaces.

1.5 L'ENVELOPPE BUDGÉTAIRE

Le budget alloué est de 0 euro. Le site sera réalisé par des stagiaires.

1.6 LE DÉLAIS



L'entreprise souhaite un site fonctionnel dans les 2 mois de stage.

1.7 GESTIONS DES DROITS

Les tableaux ci-dessous représentent les droits des utilisateurs par rapport au système.

Le C.R.U.D : Create, Read, Update, Delete , désigne les 4 opérations de base pour la persistance des données et dans notre cas le stockage d'informations en base de données.



■ Front-office :

		Gérant	Internaute
Accueil	c		
	r	x	x
	u		
	d		
Devis	c	x	x
	r		
	u		
	d		
Contact	c	x	x
	r		
	d		
	u		

■ Back-office :

Ici, on va imaginer que l'entreprise s'agrandit et recrute des nouveaux collaborateurs

		Gérant	Futur collaborateur
Accueil	c		
	r	x	x
	u		
	d		
Authentification	r	x	x
Déconnexion	r	x	x
Gestion devis	c		
	r	x	x
	u		
	d	x	
Gestion newsletter	c	x	x
	r	x	x
	u		
	d	x	
Gestion des contacts (formulaire)	c		
	r	x	x
	u		
	d	x	
Gestion des stocks	c	x	x
	r	x	x
	u	x	x
	d	x	
Gestion fichier clients	c	x	x
	r	x	x
	u	x	x
	d	x	

1.8 DESCRIPTION FONCTIONNELLE

En tant que gérant, je veux pouvoir :

- Gérer les devis.
- Envoyer des newsletters.
- Gérer les messages contacts.
- Gérer le stocks.

En tant que gérant je veux pouvoir télécharger un devis en PDF, l'envoyer et/ou imprimer.

Entant que gérant je veux pouvoir faire des recherches de références qui sont dans le stock

En tant qu'internaute, je veux pouvoir :

- Me renseigner sur l'entreprise.
- Contacter l'entreprise.
- Faire une demande de devis.

1.9 Descriptions détaillées

Je vous présente ici quelques fonctionnalités :

Fonction : Gérer le stock	
Objectif	Création, modification, suppression des produits
Contraintes / Règles de gestion	Être authentifié comme gérant
Niveau de priorité	Haute

Fonction : Gérer les messages contact	
Objectif	Visualisation, suppression des messages
Contraintes / Règles de gestion	Être authentifié comme gérant
Niveau de priorité	Moyenne

D'autres descriptions sont disponibles dans l'annexe

1.9 LE PLANNING

GANTT project					
Nom	Date de début	Date de fin	Durée	Avancée	
☐ • Projet NTPROTEC	20/01/15	18/03/15	42		85
☐ • Projet	22/01/15	23/01/15	2		100
☐ • Présentation de l'entreprise	22/01/15	22/01/15	1		100
☐ • Définition du projet	22/01/15	23/01/15	2		100
☐ • Conception	28/01/15	13/02/15	13		100
☐ • Maquette IHM	28/01/15	03/02/15	5		100
☐ • Diagrammes	05/02/15	13/02/15	7		100
☐ • Implémentation	20/01/15	18/03/15	42		82
☐ • Analyse	16/02/15	20/02/15	5		100
☐ • Analyse	16/02/15	19/02/15	4		100
☐ • Conception de la base de données	17/02/15	20/02/15	4		100
☐ • Création du dictionnaire des données	17/02/15	18/02/15	2		100
☐ • Diagrammes de classes	19/02/15	20/02/15	2		100
☐ • Dépendences fonctionnelles	17/02/15	19/02/15	3		100
☐ • Diagrammes de classes générales	19/02/15	20/02/15	2		100
☐ • Base de données	23/02/15	24/02/15	2		100
☐ • Modèle de la base de données	23/02/15	24/02/15	2		100
☐ • Création de la base de données	23/02/15	24/02/15	2		100
☐ • Création des tables	23/02/15	24/02/15	2		100
☐ • Architecture n-tiers	24/02/15	24/02/15	1		100
☐ • Arborescence des fichiers	24/02/15	24/02/15	1		100
☐ • Choix de l'architecture	24/02/15	24/02/15	1		100
☐ • Codes	25/02/15	18/03/15	16		91
☐ • Classe de connexion	25/02/15	25/02/15	1		100
☐ • Authentification avec gestion d'erreurs	26/02/15	27/02/15	2		100
☐ • Fichier Config	27/02/15	27/02/15	1		100
☐ • Route et dispatcher	02/03/15	06/03/15	5		100
☐ • Création de classes DAO	09/03/15	11/03/15	3		100
☐ • HTML	11/03/15	13/03/15	3		90
☐ • JavaScript / jQuery / Ajax	09/03/15	18/03/15	8		80
☐ • PHP	25/02/15	18/03/15	16		90
☐ • Tests	23/01/15	18/03/15	39		80
☐ • Validation	23/01/15	18/03/15	39		80
☐ • Rapport de stage	20/01/15	20/01/15	1		0
☐ • Validation du projet	20/01/15	20/01/15	1		0

CHAPITRE 3 : CONCEPTION

1. MAQUETTE INTERFACE HUMAIN MACHINE

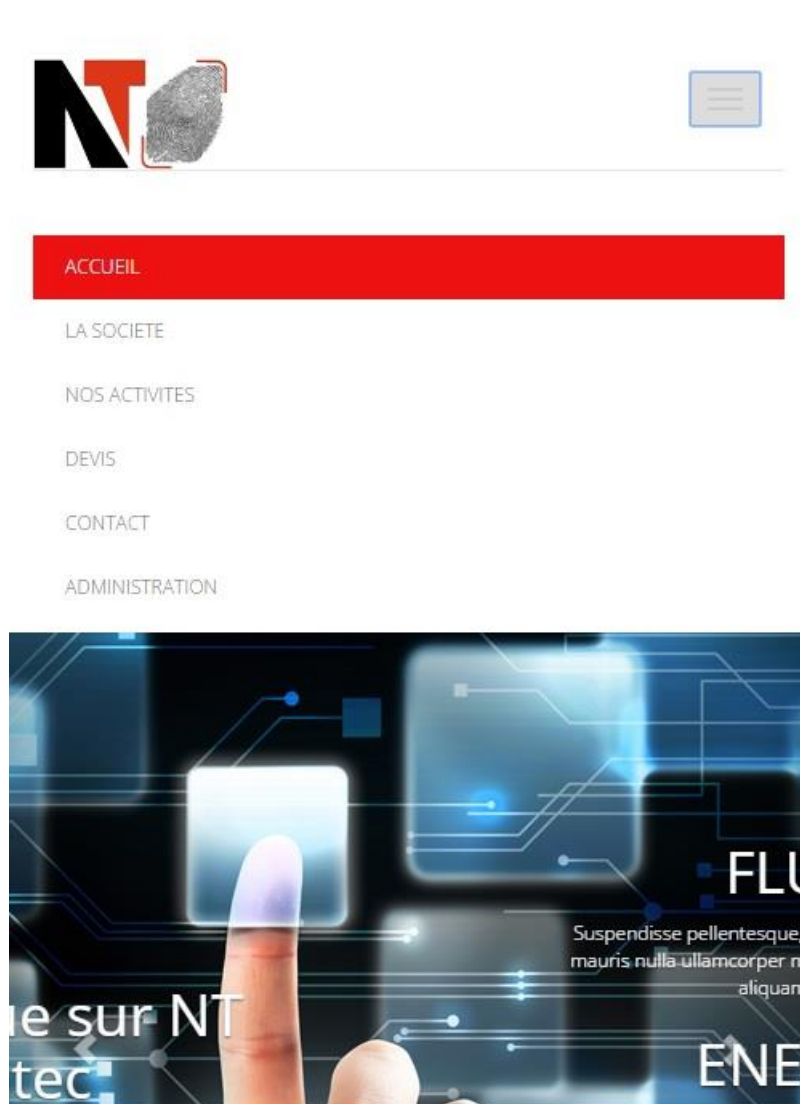
1.1 FRONT-OFFICE

- Page d'accueil :



FIGURE N° 4 : NOUVEAU SITE FRONT OFFICE

- Version mobile



■ Page du devis :

La page devis est séparée en deux. Une partie pour les particuliers et une autre pour les entreprises

Particulier

Entreprise

Votre nom...

Votre prenom...

Votre Email...

Votre adresse

Votre numéro de téléphone

Votre habitation

☐ Maison ☐ Appartement

Situation

☐ Ville ☐ Campagne

Comment accède-t-on à votre habitation ?

☐ Par un portail sur rue

☐ Via une résidence sécurisée

☐ Via une résidence non sécurisée

Votre habitation possède t-elle un accès secondaire

☐ Un jardin

☐ Un balcon ou une terrasse

☐ Aucun

Vos fenêtres ont-elles des barreaux ?

☐ Oui

☐ Non

Combien d'heures par jour votre maison est-elle inoccupée ?

☐ Elle n'est presque jamais inoccupée

☐ Moins de 3 heures par jour

☐ Plus de 5 heures par jour

Vous ou vos voisins avez-vous été victimes d'un cambriolage ?

☐ Oui

☐ Non

Voulez vous souscrire à notre newsletter?

☐ Oui

☐ Non

Votre message...


Envoyer

FIGURE N° 5 : PAGE DEVIS FRONT OFFICE

- Page de contact :

La page contact va permettre à l'internaute de contacter l'entreprise via ce formulaire sans passer par le devis.

CONTACTEZ NOUS



Des questions? Vous pouvez nous les poser via ce formulaire de contact.

9, Rue des Maraîchers 95130 FRANCONVILLE LA GARENNE

01.34.37.52.97

www.ntprotec.fr

Votre nom...

Votre email...

Voulez-vous vous inscrire à notre newsletter ?

☐ Oui ☐ Non

Votre message...

Envoyer

Vous pouvez nous trouver au 9, Rue des Maraîchers FRANCONVILLE LA GARENNE

FIGURE N° 6 : PAGE CONTACT FRONT OFFICE

1.2 BACK-OFFICE

- Page d'authentification :

L'interface de la page d'authentification de l'application.



BACK OFFICE
Bienvenue - Administrateur

ACCUEIL CLIENTS DEVIS STOCKS NEWSLETTERS DECONNEXION

Bienvenue : Veuillez-vous connecter

Authentification

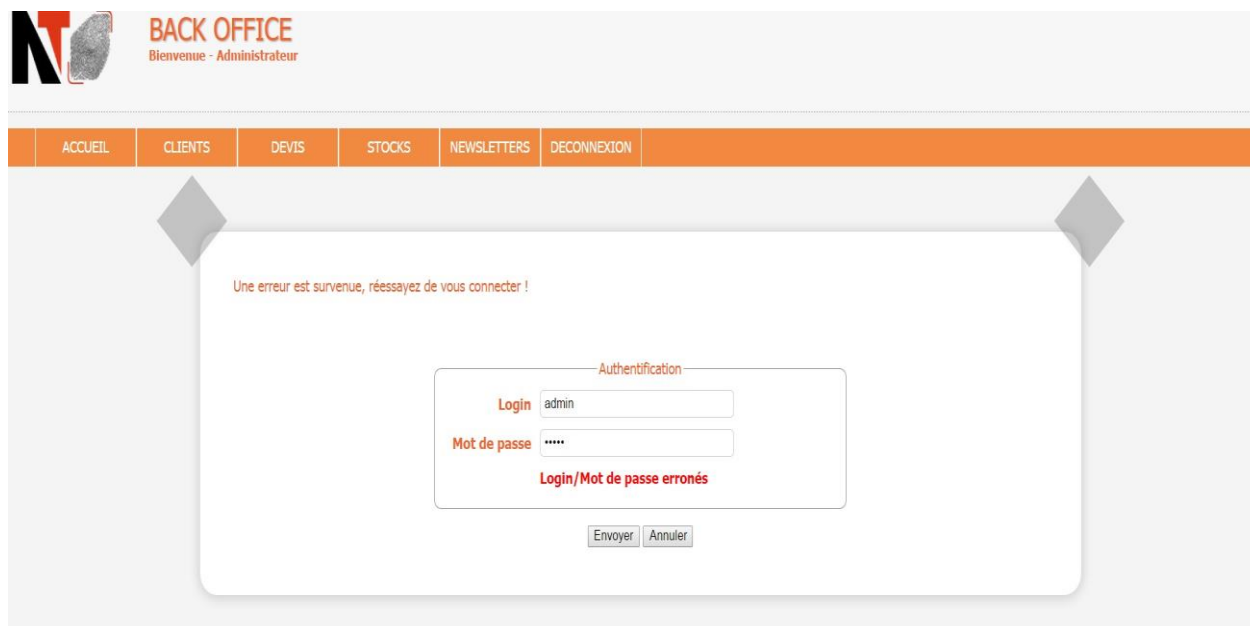
Login

Mot de passe

Envoyer Annuler

FIGURE N° 7 : PAGE D AUTHENTIFICATION BO

Ici, l'affichage d'erreur.



The screenshot displays a web application interface for a 'BACK OFFICE' system. At the top left, there is a logo with the letters 'NT' and a small image of a person. To its right, the text 'BACK OFFICE' is displayed in a bold, orange font, with 'Bienvenue - Administrateur' in a smaller, lighter orange font below it. A horizontal navigation bar with an orange background contains several menu items in white capital letters: 'ACCUEIL', 'CLIENTS', 'DEVIS', 'STOCKS', 'NEWSLETTERS', and 'DECONNEXION'. The main content area has a light gray background. In the center, a white rectangular box with a subtle drop shadow contains an error message and a login form. The error message, 'Une erreur est survenue, réessayez de vous connecter !', is written in a small, orange font. Below it, the login form is titled 'Authentification' in a small, orange font. It includes two input fields: 'Login' with the text 'admin' and 'Mot de passe' with masked characters '*****'. A red error message, 'Login/Mot de passe erronés', is positioned below the input fields. At the bottom of the form are two buttons: 'Envoyer' and 'Annuler'.

NT

BACK OFFICE
Bienvenue - Administrateur

ACCUEIL CLIENTS DEVIS STOCKS NEWSLETTERS DECONNEXION

Une erreur est survenue, réessayez de vous connecter !

Authentification

Login admin

Mot de passe *****

Login/Mot de passe erronés

Envoyer Annuler

FIGURE N° 8 : PAGE D'AUTHENTIFICATION AVEC ERREUR

- Page d'accueil :

L'interface de la page d'accueil de l'application.

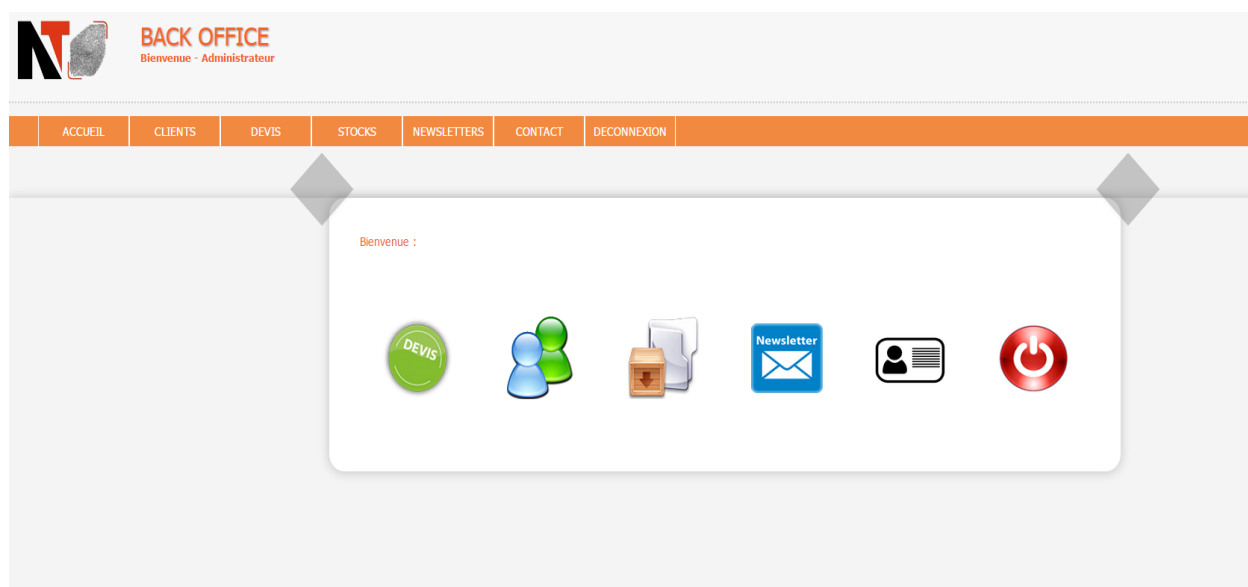


FIGURE N° 9 : PAGE D'ACCUEIL BACK OFFICE

- Page clients :

Le gérant aura le choix entre la page particulier ou entreprise.

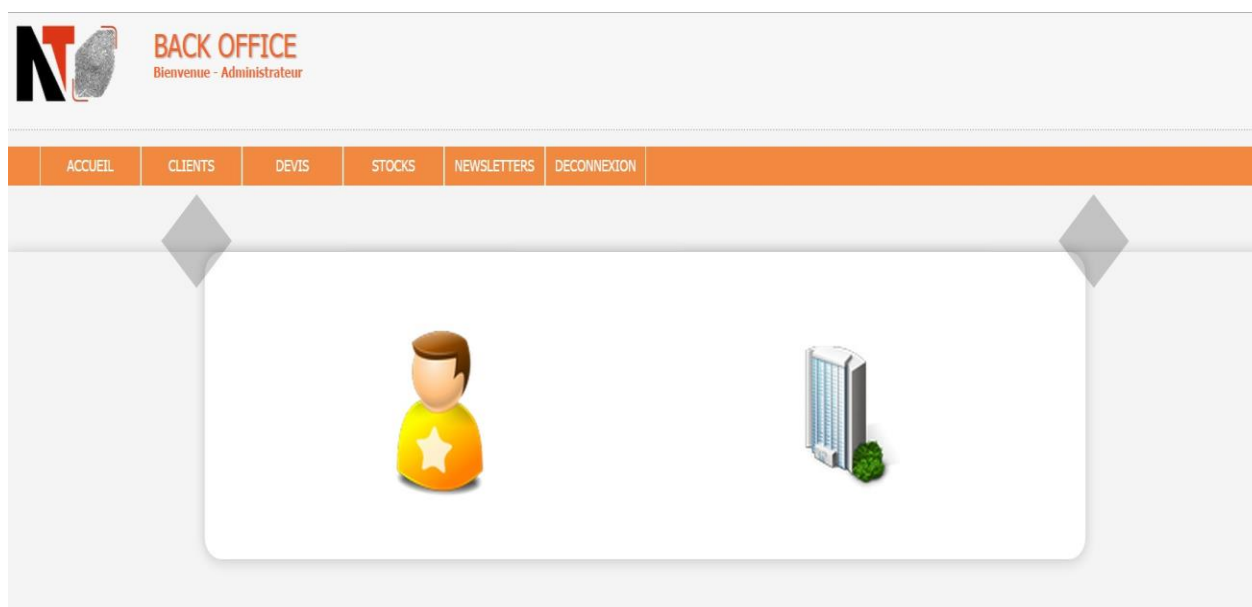
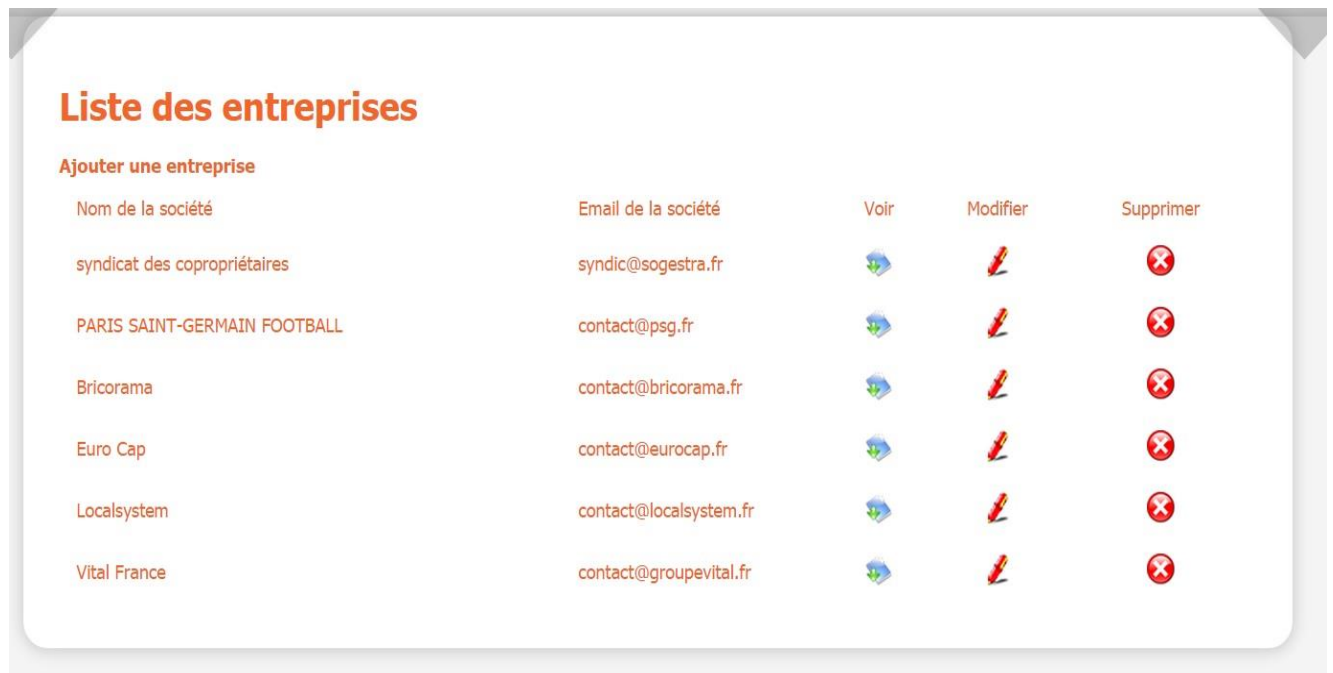


FIGURE N° 10 : PAGE CLIENT BACK OFFICE

- Page des entreprises :

En cliquant sur la page entreprise, on voit apparaître la liste des entreprises. On aura la possibilité ici de choisir si on veut voir, modifier, supprimer une entreprise ou en ajouter une nouvelle.





















Nom de la société	Email de la société	Voir	Modifier	Supprimer
syndicat des copropriétaires	syndic@sogestra.fr			
PARIS SAINT-GERMAIN FOOTBALL	contact@psg.fr			
Bricorama	contact@bricorama.fr			
Euro Cap	contact@eurocap.fr			
Localsystem	contact@localsystem.fr			
Vital France	contact@groupevital.fr			

FIGURE N° 11 : LISTE ENTREPRISES BACK OFFICE

▪ L'inscription d'une nouvelle entreprise :

Le formulaire d'ajout d'une entreprise est similaire à la page devis du front-office.

Inscription d'une entreprise

Inscription

Entreprise :

Numéro de siret:

Email :

Adresse :

Téléphone :

Où votre entreprise est-elle installée ?

- ☐ Dans une rue commerçante
- ☐ Dans un centre commercial
- ☐ Dans une zone d'activité
- ☐ Dans un immeuble de bureaux

Situation

- ☐ Ville ☐ Campagne

Combien de salariés avez-vous dans votre entreprise ?

- ☐ Un seul employé
- ☐ De 2 à 5 employés
- ☐ Plus de 5 employés

Quelles sont les horaires d'ouverture de votre entreprise ?

- ☐ Le matin seulement
- ☐ L'après midi seulement
- ☐ Le matin et l'après-midi
- ☐ La nuit
- ☐ 24H/24

Quelle est la valeur des biens présents dans votre entreprise ?

- ☐ Valeur faible
- ☐ Valeur moyenne
- ☐ Valeur forte

Votre entreprise ou des entreprises à proximité de la vôtre ont-elles été cambriolées ?


- ☐ Oui
- ☐ Non

Autres informations ...

FIGURE N° 12 : FORMULAIRE AJOUT ENTREPRISE BACK OFFICE

- Interface du stock :

L'interface du stock affiche la liste des produits qui sont dans la base de donnée.

Il y'a une champs de recherche qui va permettre au gérant de pouvoir rechercher rapidement le produit qu'il voudra. Il aura aussi la possibilité d'ajouter une produit ou d'en modifier, supprimer un. Il pourra également ajouter une catégorie. 

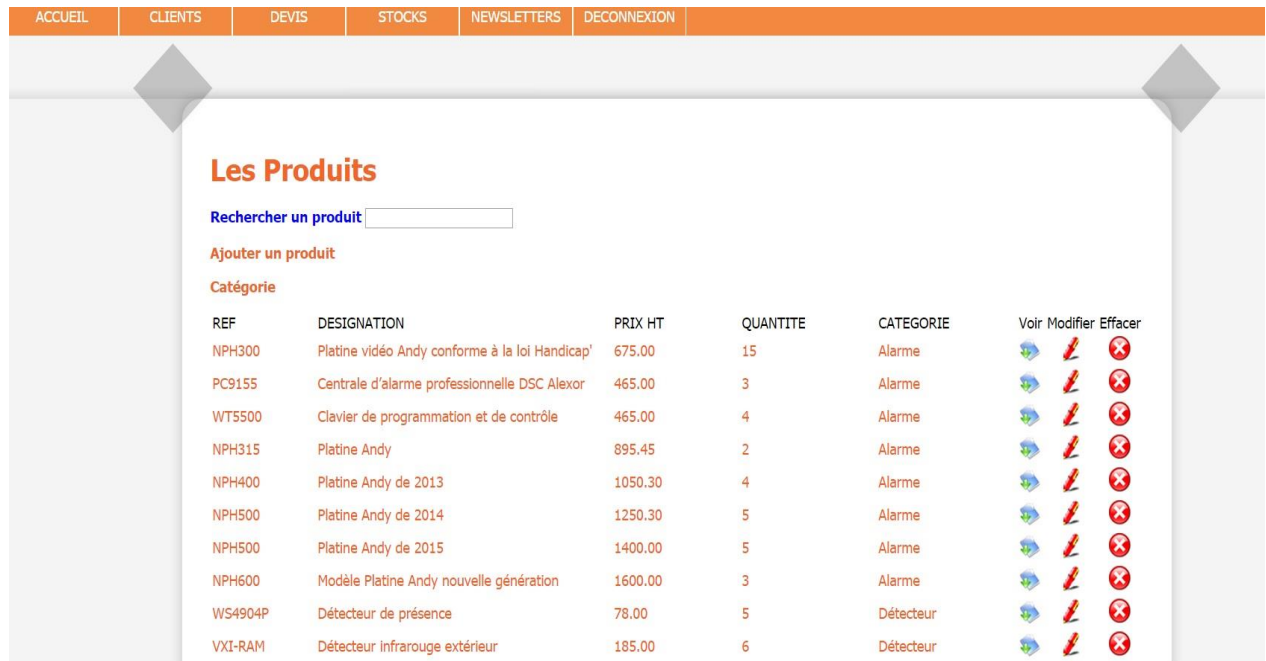
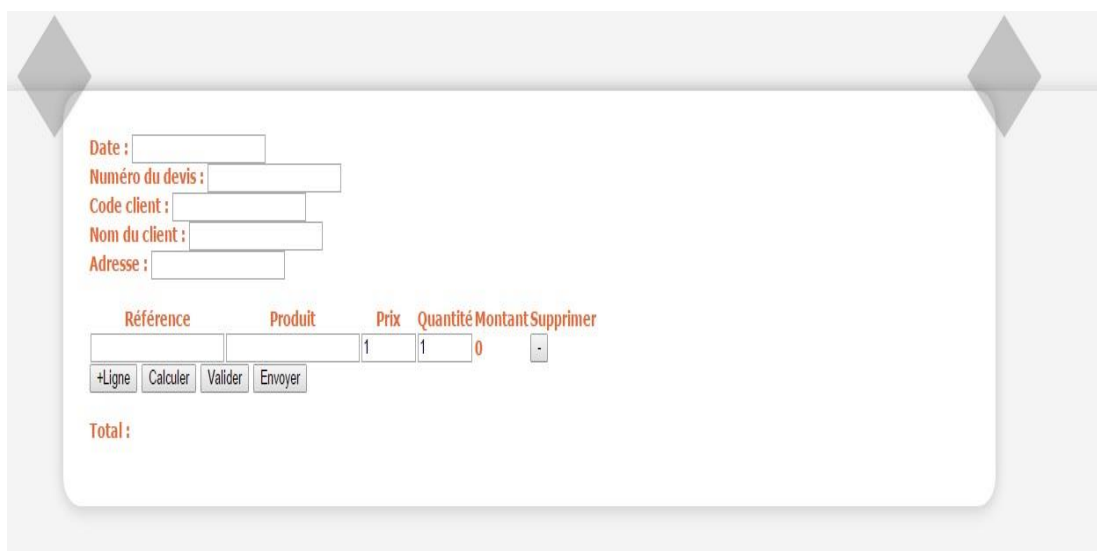


FIGURE N° 13 : INTERFACE STOCK BACK OFFICE

- Maquette formulaire de devis :



Maquette d'un formulaire de devis. Le formulaire est présenté sur un fond gris avec des coins arrondis et des ombres. Il contient les champs suivants :

- Date :
- Numéro du devis :
- Code client :
- Nom du client :
- Adresse :

En dessous, une table à 6 colonnes :

Référence	Produit	Prix	Quantité	Montant	Supprimer
<input type="text"/>	<input type="text"/>	1	1	0	<input type="button" value="-"/>

En dessous de la table, quatre boutons :

Enfin, un label "Total :" est présent à gauche.

Interface de la newsletter :

L'interface de la newsletter va permettre au gérant d'envoyer une newsletter à toutes les personnes qui auront souscrit à celle-ci.

BACK OFFICE
Bienvenue - Administrateur

ACCUEIL CLIENTS DEVIS STOCKS NEWSLETTERS DECONNEXION

Voir les anciennes newsletters

Newsletter

Objet :

Sujet : Nouveau produit

Edito :

Signature : Romain Delaville

Envoyer Annuler

FIGURE N° 14 : INTERFACE NEWSLETTER BACK OFFICE

2. DIAGRAMMES

2.1 DIAGRAMME DE NAVIGATION

- Front-office

Voici , le diagramme de navigation du front-office. Ce diagramme permet de visualiser les actions possibles sur l'application.

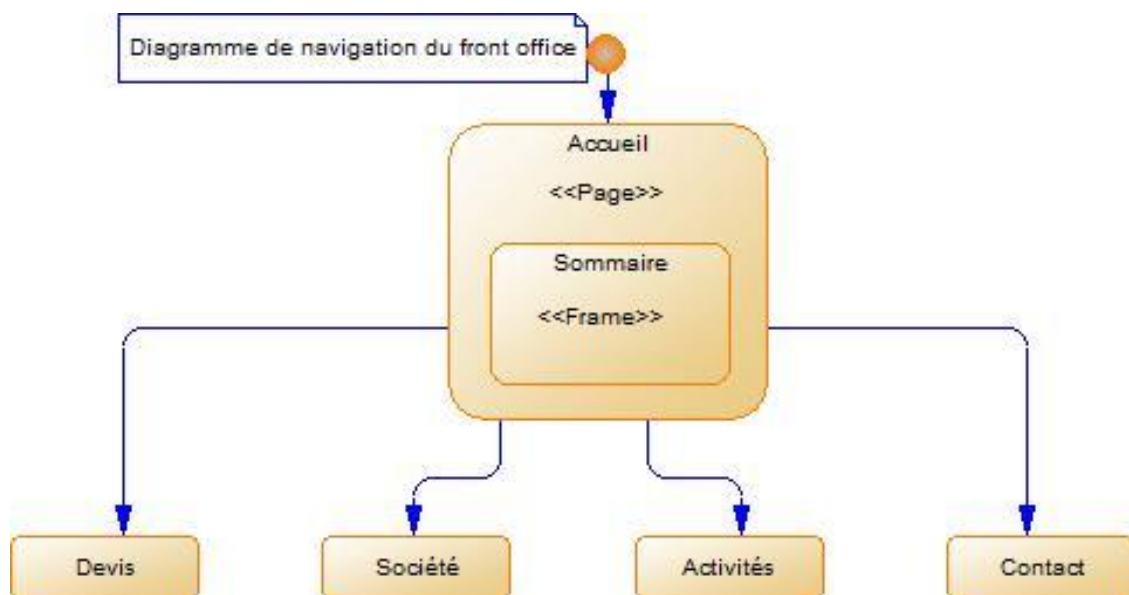


FIGURE N° 15 : DIAGRAMME DE NAVIGATION FO

- Back-office général simplifié

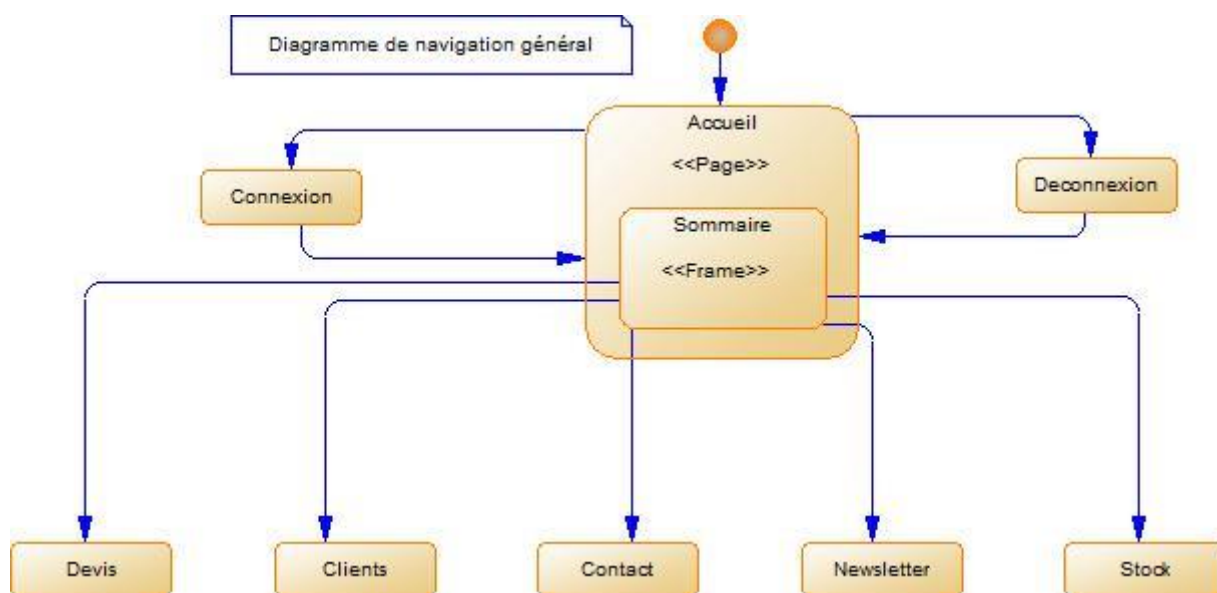


FIGURE N° 16 : DIAGRAMME DE NAVIGATION SIMPLIFIE BACK OFFICE

- Back-office stock détaillé

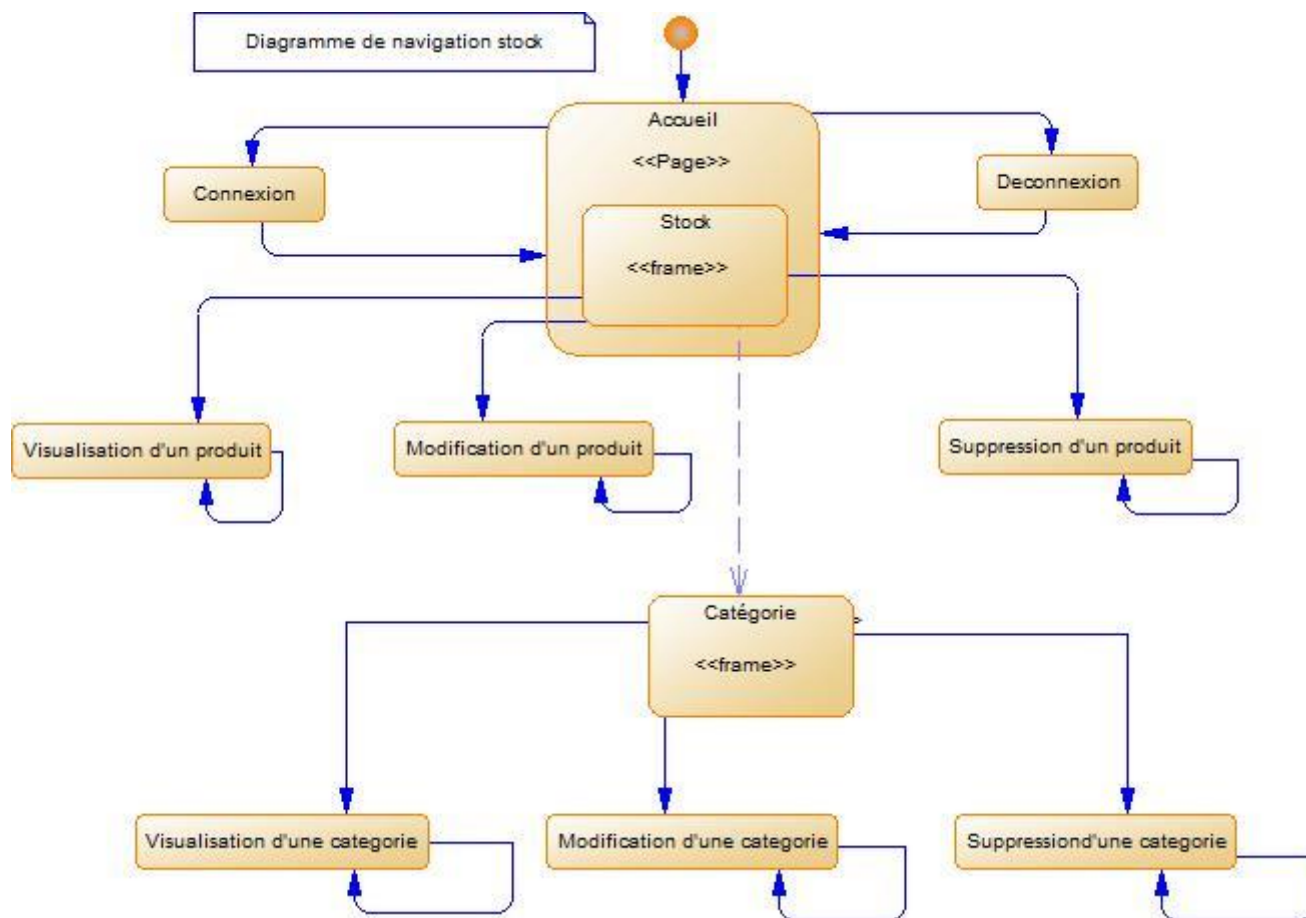


FIGURE N° 17 : DIAGRAMME DE NAVIGATION NEWSLETTER ET STOCK

D'autres diagrammes de navigation sont disponibles dans les annexes.

- Back-office général détaillé

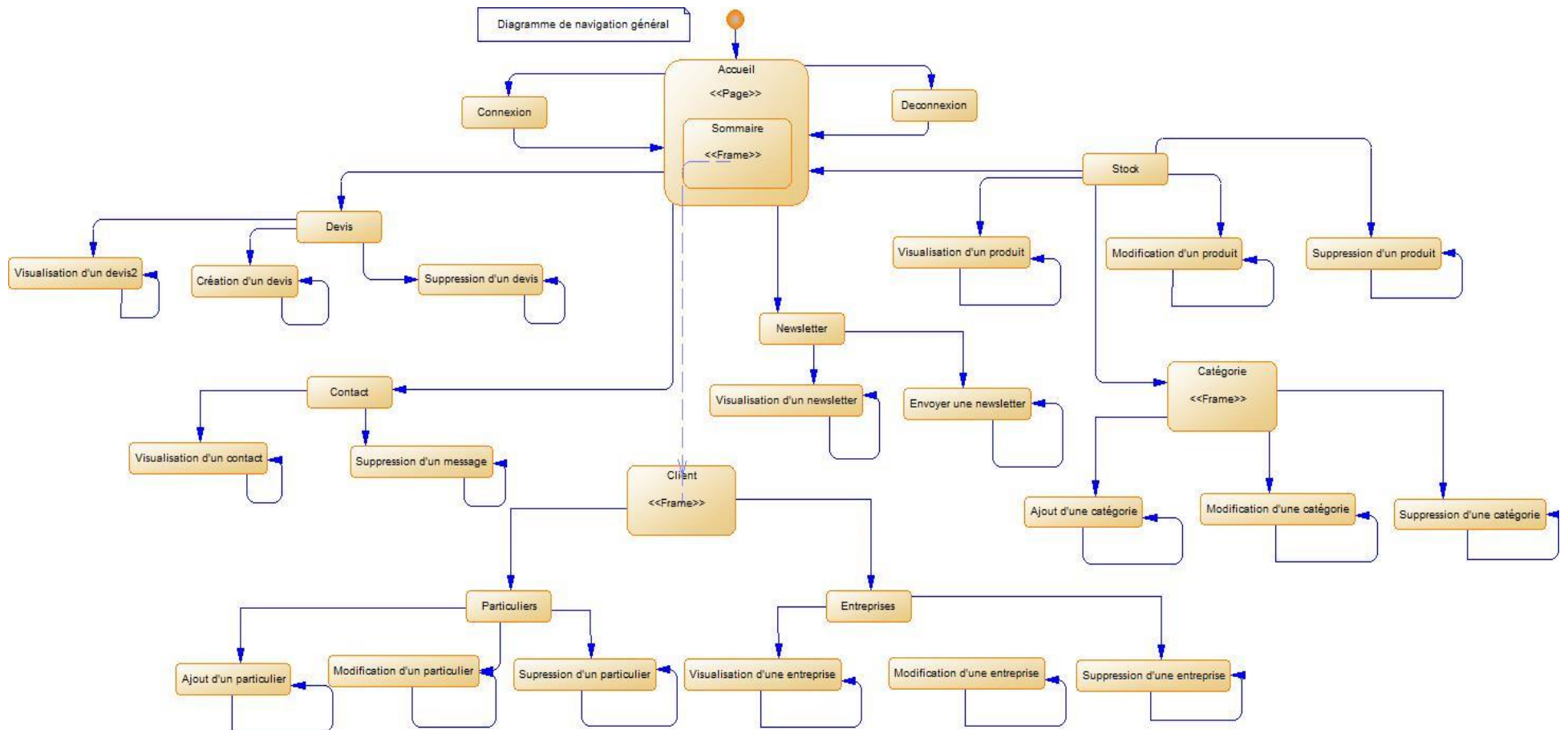


FIGURE N° 18 : DIAGRAMME DE NAVIGATION GENERAL

2.2 DIAGRAMME D'ACTIVITE

Le diagramme d'activité permet de représenter le comportement d'une méthode ou le déroulement d'un cas d'utilisation.



- Front-office

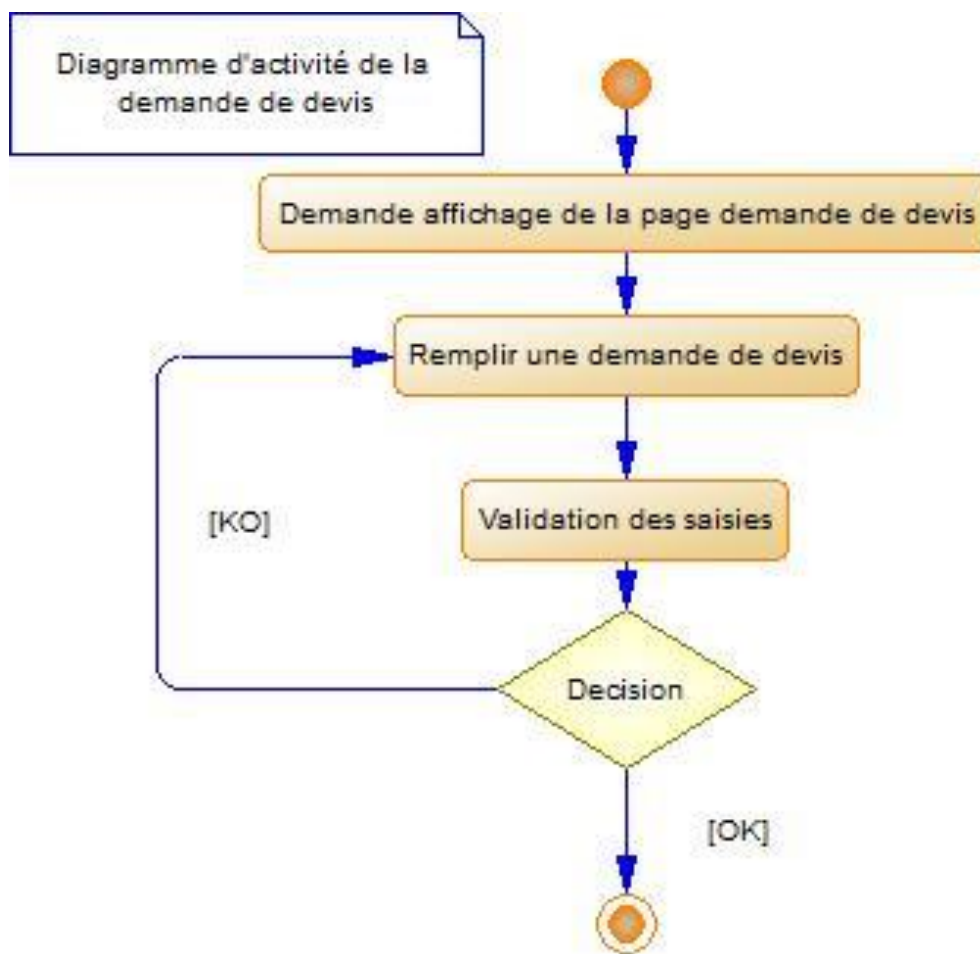


FIGURE N° 19 : DIAGRAMME D'ACTIVITE PAGE DEVIS

■ Back-office

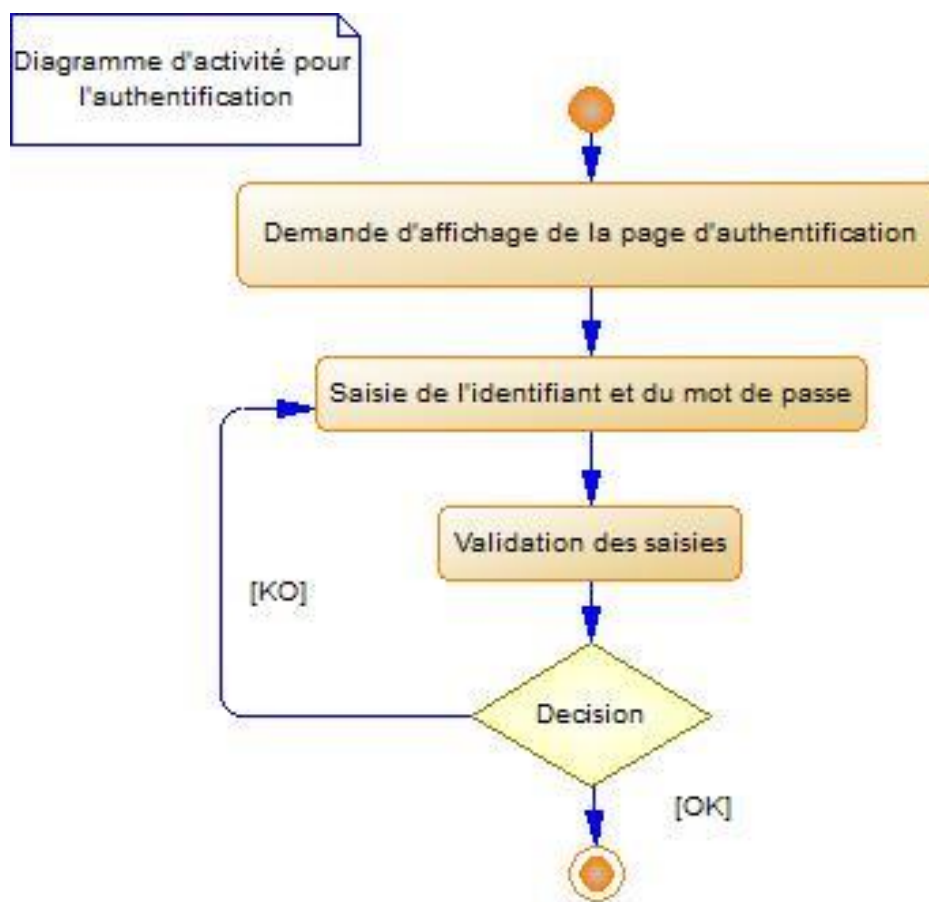


FIGURE N° 20 : DIAGRAMME D'ACTIVITE PAGE AUTHENTIFICATION

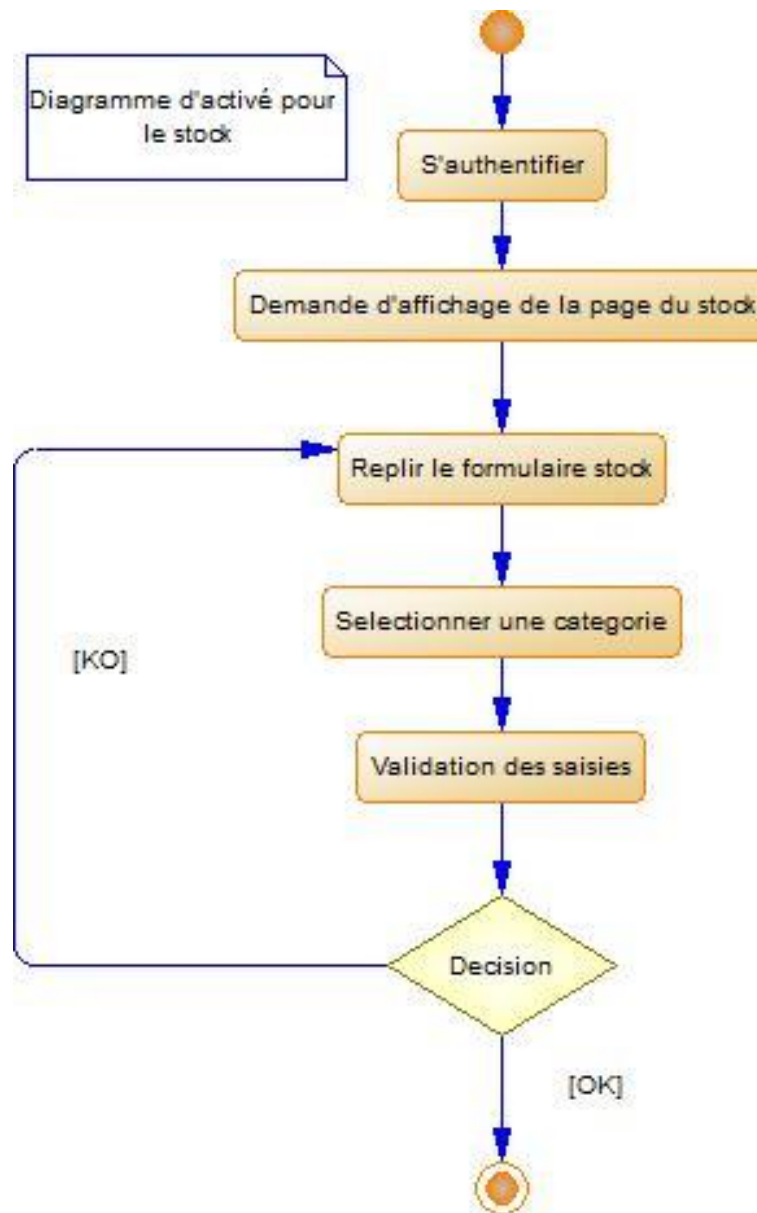
Diagramme d'activité de la page stock

FIGURE N° 21 : DIAGRAMME D'ACTIVITE AJOUTE AU STOCK

2.3 Diagramme de séquence



Je ne citerai que quelques scénarios car tous les présentés seraient beaucoup trop long.

Le diagramme de séquence permet de décrire et de formaliser les interactions entre les acteurs et le système ou les classes, par l'intermédiaire de messages, d'un point de vue chronologique et spatial. Le temps est représenté à la verticale et l'espace à l'horizontal.

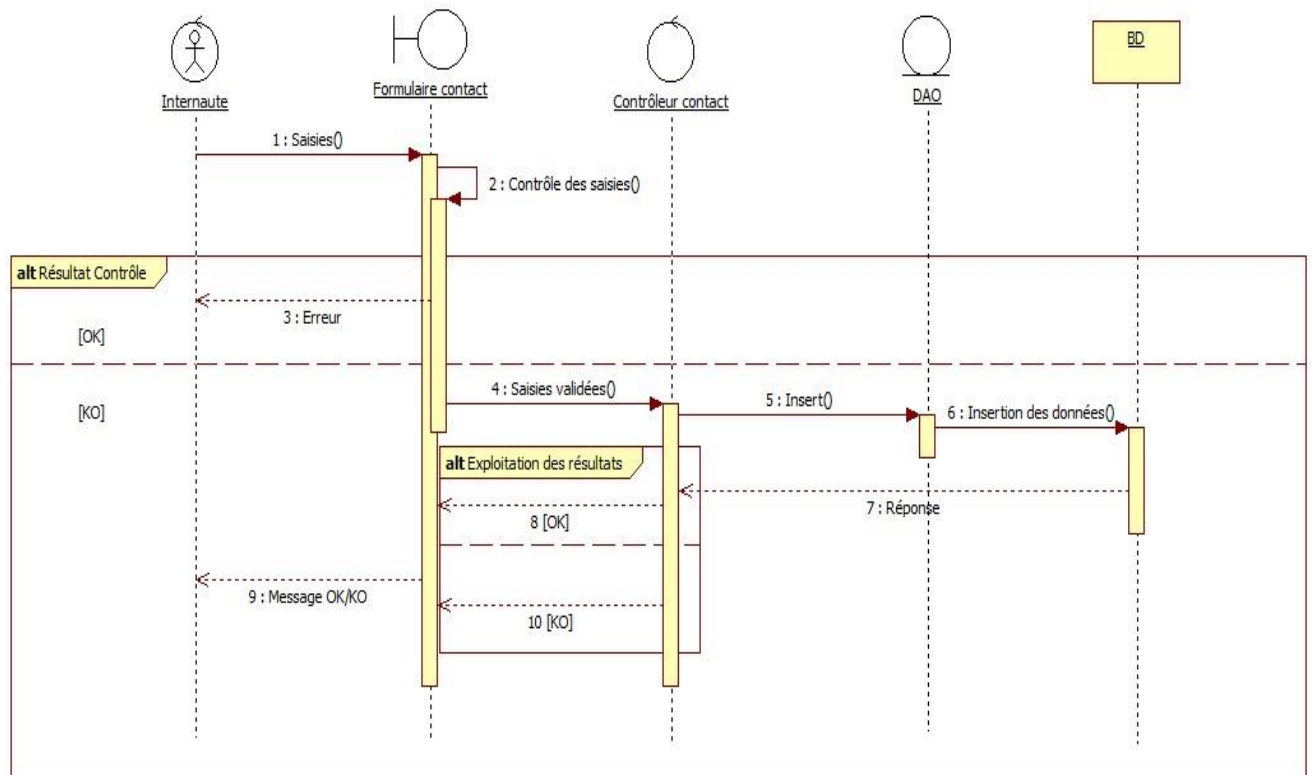


FIGURE N° 22 : DIAGRAMME DE SEQUENCE PAGE CONTACT

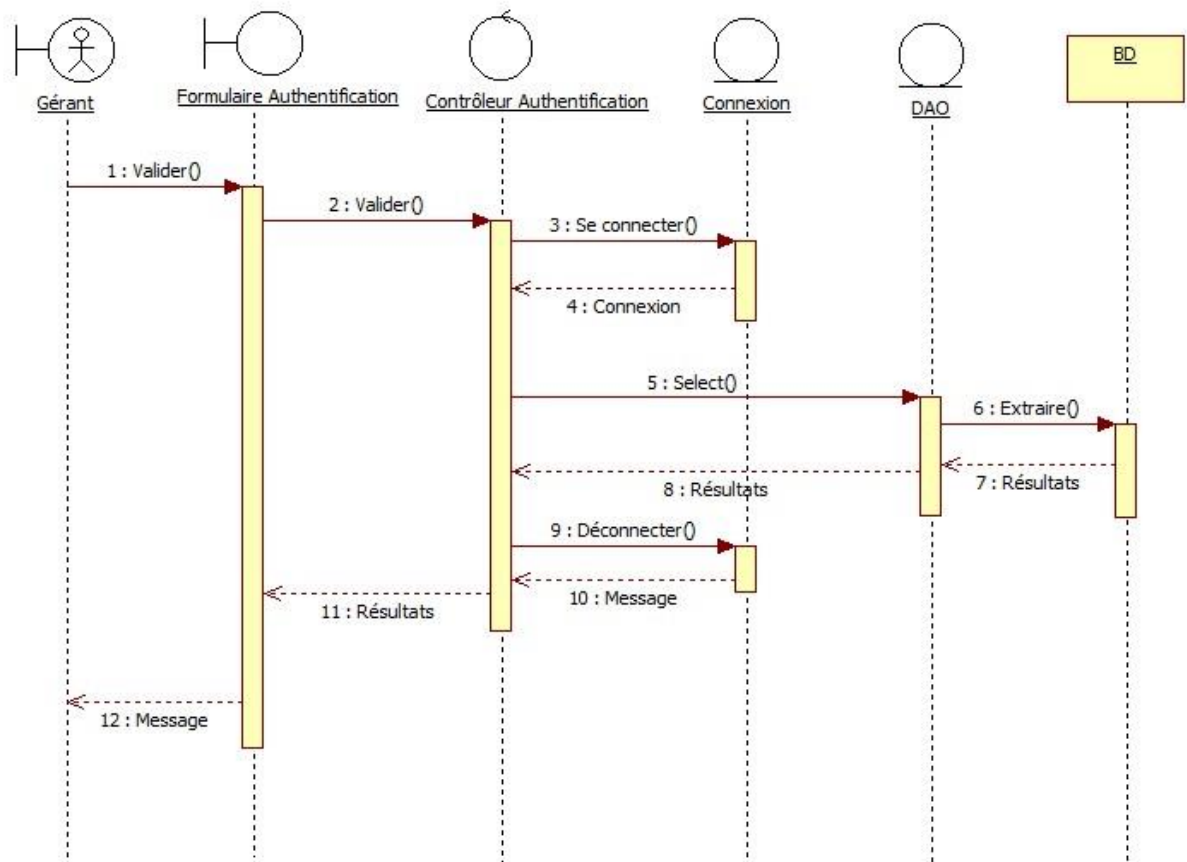


FIGURE N° 23 : DIAGRAMME DE SEQUENCE AUTHENTICATION

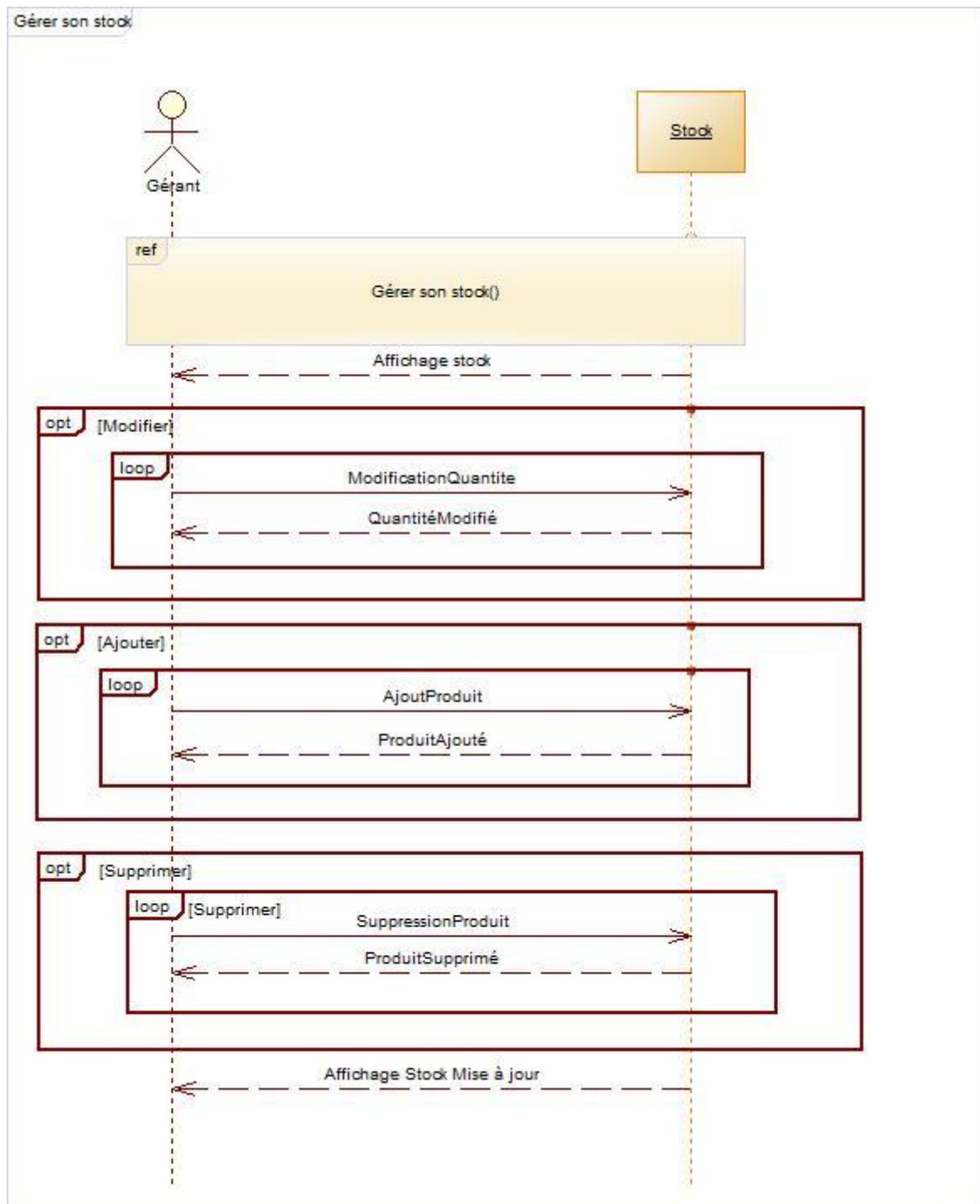
Diagramme de séquence du stock

FIGURE N° 24 : DIAGRAMME DE SEQUENCE STOCK

2.4 Diagramme d'états-transition

Les Diagramme d'états transitions décrivent le comportement interne d'un objet à l'aide d'un automate à états finis.

Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements.

Diagramme d'états-transition devis

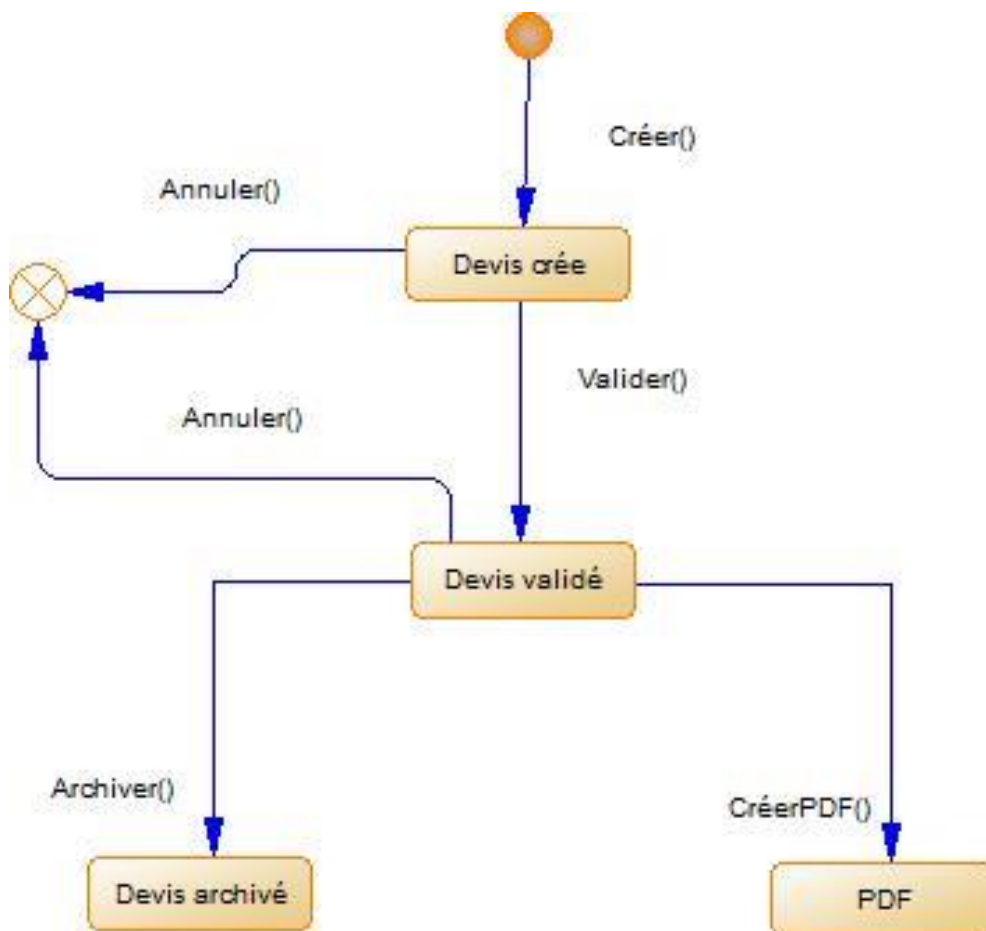
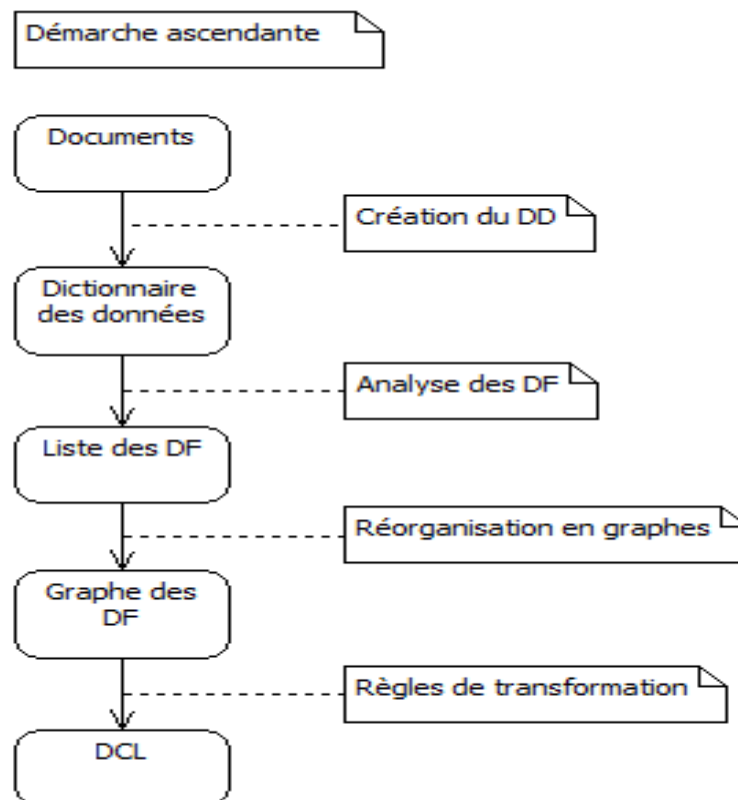


FIGURE N° 25 : DIAGRAMME D'ETATS TRANSITION DEVIS

CHAPITRE 4 : IMPLÉMENTATION

1.DIAGRAMME DE CLASSES DU DOMAINE

Nous allons utiliser la démarche suivant que l'on a vu en cours cette année pour créer le diagramme de classes.



Démarche ascendante : du concret vers l'abstrait

Tableau des sigles :

DD	Dictionnaire des données
DF	Dépendance fonctionnelle
GDF	Graphe des dépendances fonctionnelles
DCL	Diagramme de classes

1.1 ANALYSE

Construction du diagramme de classes à partir de la page du devis.

Rappel du modèle :

The image shows a web form for creating a quote (devis). It includes several input fields for client information, a table for line items, and a total field.

Fields:

- Date :
- Numéro du devis :
- Code client :
- Nom du client :
- Adresse :

Référence	Produit	Prix	Quantité	Montant	Supprimer
<input type="text"/>	<input type="text"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="button" value="-"/>

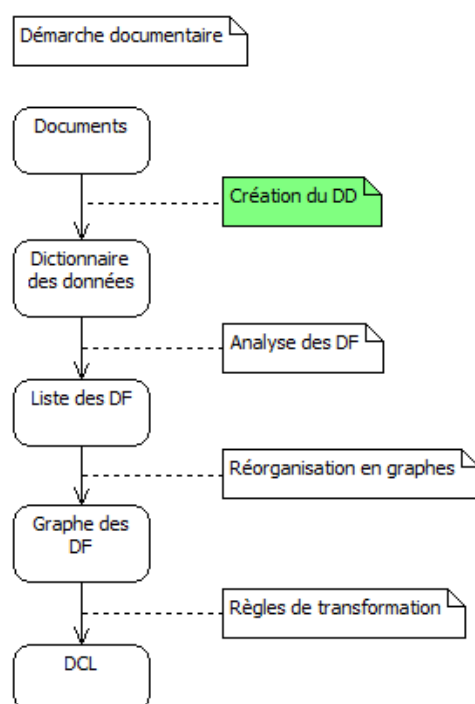
Buttons:

Total :

FIGURE N° 26 : MODELE DEVIS

Il y'a 11 informations dans ce document.

1.2 Dictionnaire des données

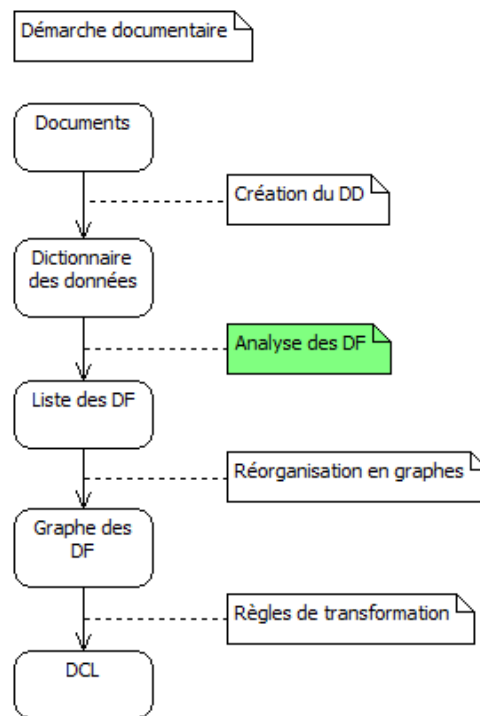


A partir des documents disponibles, on établit le dictionnaires des données puis on épure le dictionnaire.

Ce qui nous donne ce tableau suivant:

Libellé	Code	Type	Longueur	Nature	Calcul	Autre règle
N° Devis	idDevis	Numérique	5	Calculée	+1	
Date	dateDevis	Date	10	Elémentaire		JJ/MM/AAAA
Code client	idClient	Numérique	5	Calculée	+1	
Nom	nomClient	Texte	50	Elémentaire		
Adresse	adresseClient	Texte	100	Elémentaire		
Réf	referenceProduit	Numérique	5	Calculée	+1	
Désignation	designationProduit	Texte	50	Elémentaire		
Prix	prixProduit	Numérique	8	Elémentaire		
Quantité	quantiteProduit	Numérique	5	Elémentaire		>0
Montant	montantLigneDevis	Numérique	8	Calculée	prix * quantité	
Total	totalDevis	Numérique	8	Calculée	somme(prix * quantité)	

1.3 LES DÉPENDANCES FONCTIONNELLES



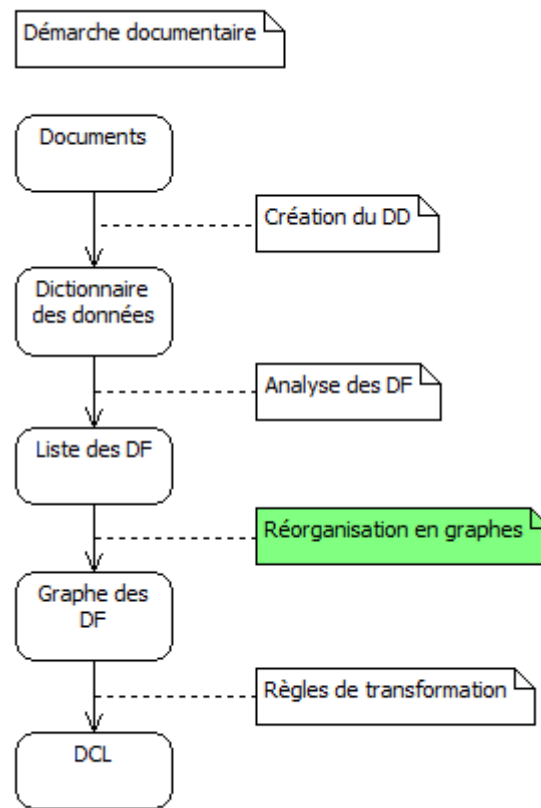
Il y a dépendance fonctionnelle entre deux attributs lorsque la **connaissance d'une valeur** d'un attribut permet de déterminer **une et une seule** valeur d'un autre attribut.

Ce qui nous donne la matrice des dépendances fonctionnelles :

Matrice des DF

Attributs		1	2	3	4	5	6	7	8	9	10	11
idDevis	1	x	x	x	x	x	x					x
dateDevis	2		x									
idClient	3			x	x	x						
nomClient	4				x							
adresseClient	5					x						
idProduit	6						x	x	x			
designationProduit	7						x	x	x			
prixProduit	8								x			
quantiteDevis	9									x		
montantLigneDevis	10										x	
totalDevis	11											x

1.4 LE GRAPHE DES DEPENDANCES FONCTIONNELLES



Grâce à cette matrice on en déduit le graphe des dépendances fonctionnelles :

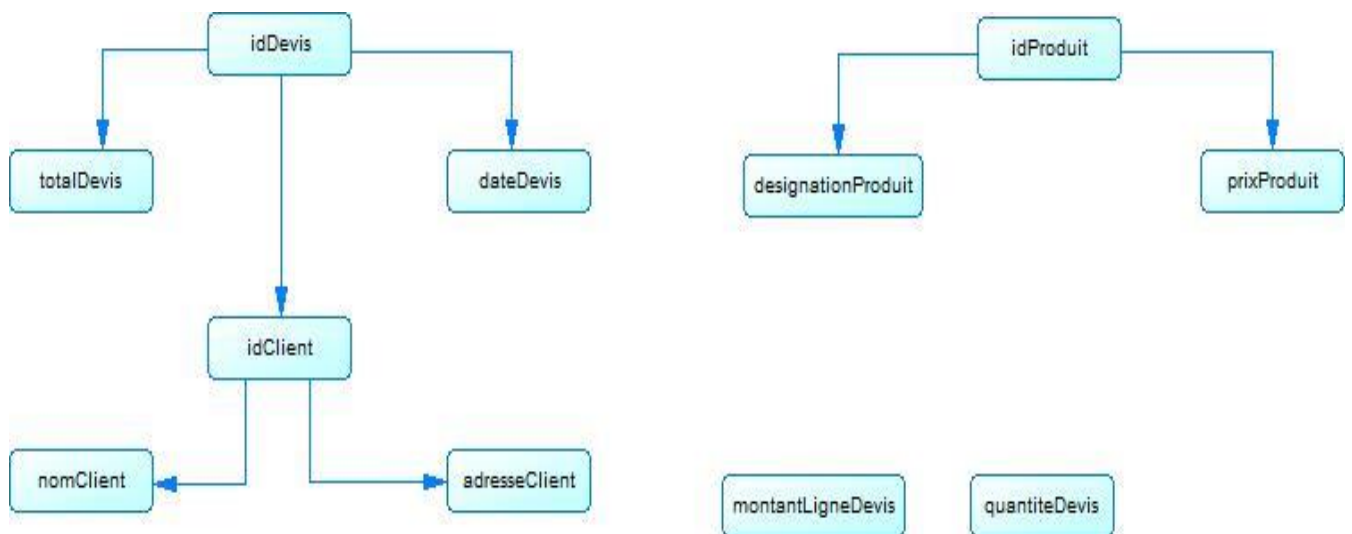


FIGURE N° 27 : GRAPHE DES DEPENDANCES FONCTIONNELLES

Certains attributs sont orphelins, il faut donc rechercher des dépendances fonctionnelles multi-sources.

Il existe deux dépendances fonctionnelles multi-sources :

$\text{idDevis} + \text{idProduit} \rightarrow \text{montantLigneDevis}$

$\text{idDevis} + \text{idProduit} \rightarrow \text{quantiteDevis}$

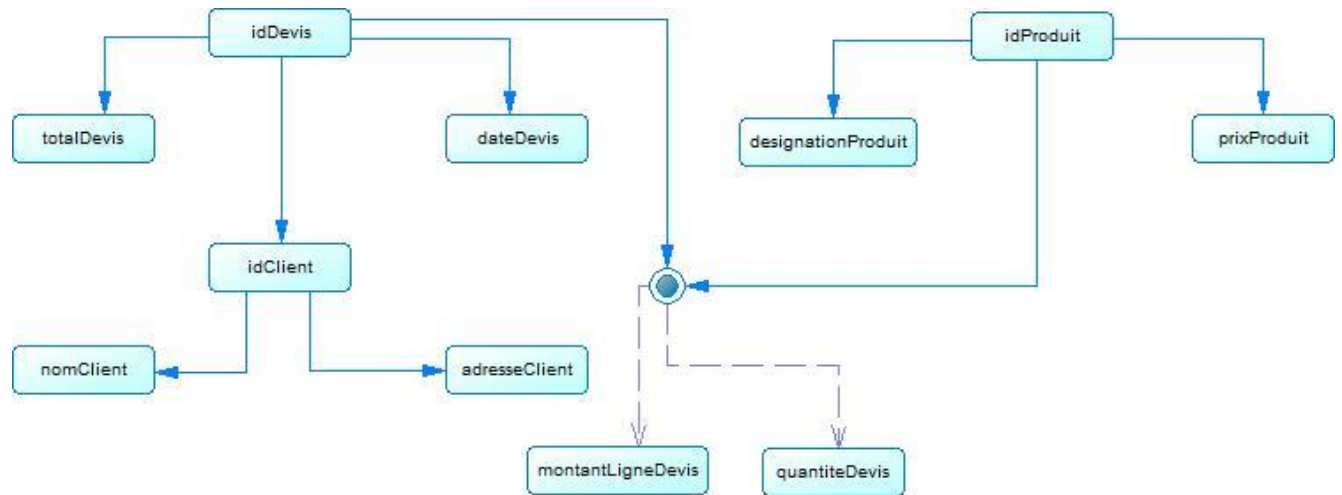
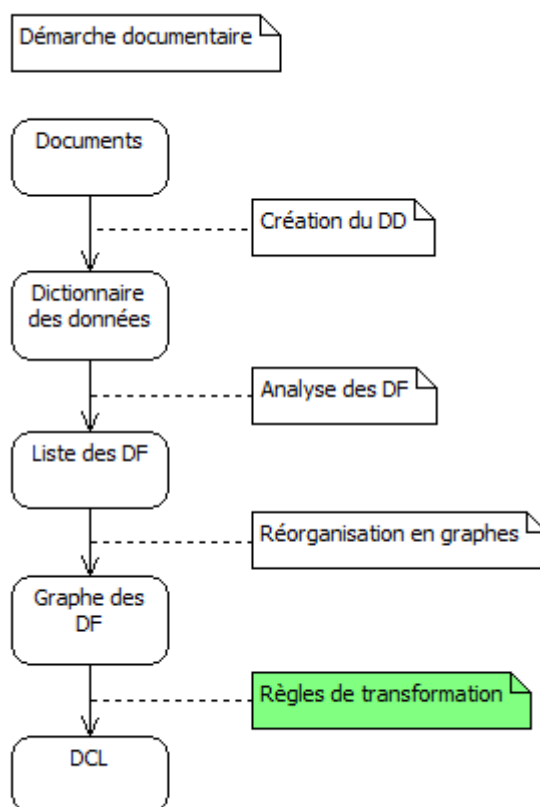


FIGURE N° 28 : GRAPHE DES DEPENDANCES FONCTIONNELLES MULTI-SOURCES

1.5 LE DIAGRAMME DE CLASSE

A partir du graphe de dépendances, on établit le diagramme de classe.



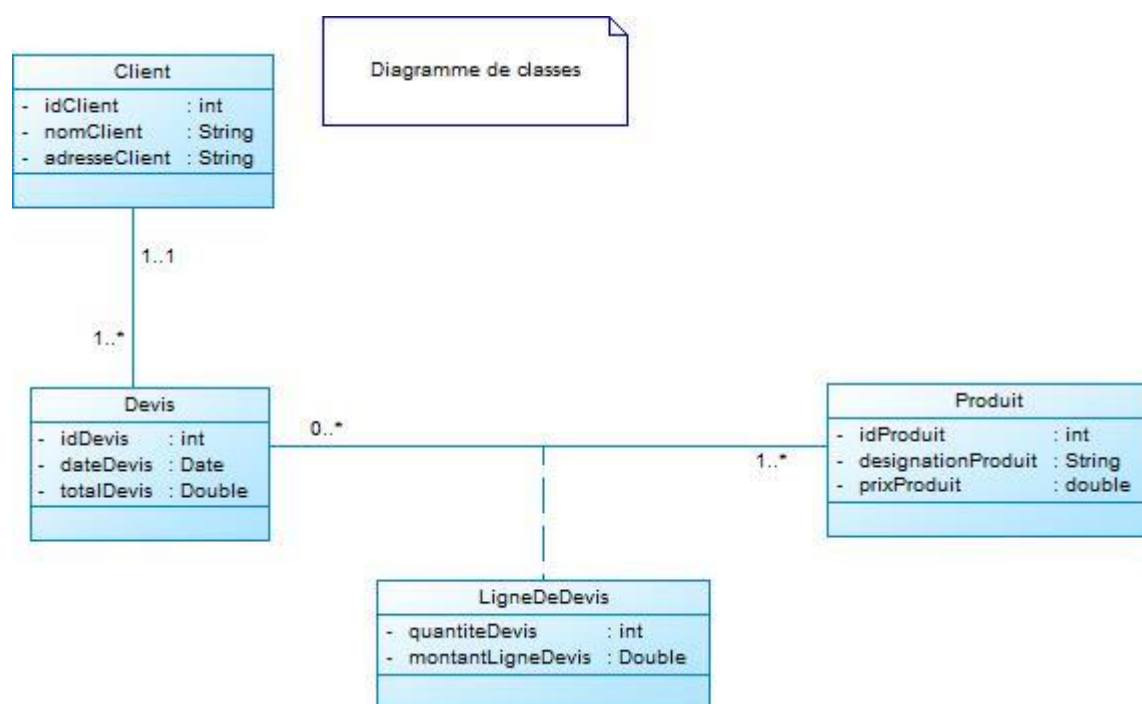


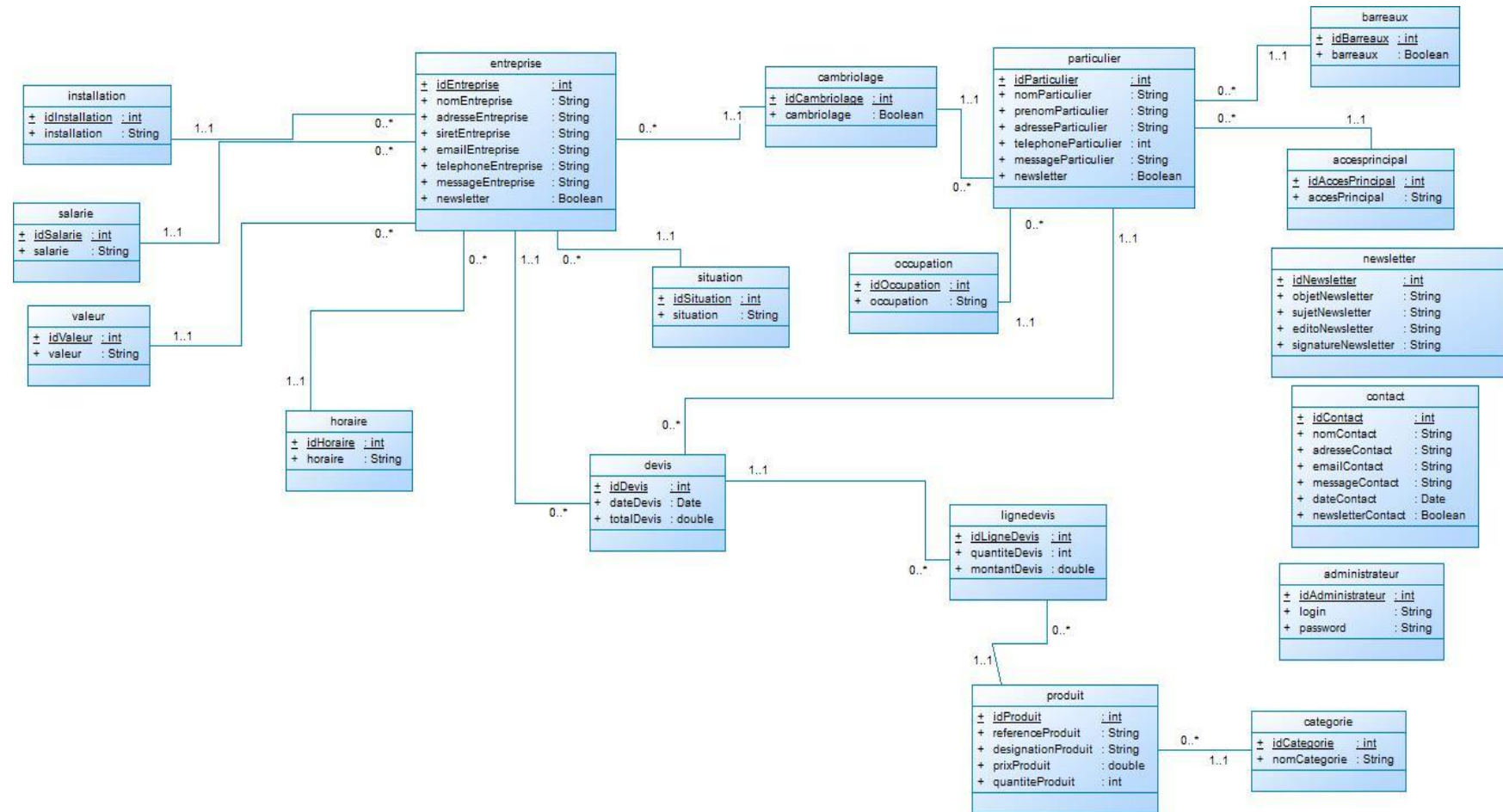
FIGURE N° 29 : DIAGRAMME DE CLASSES

Les autres diagrammes sont disponibles dans les annexes

1.6 RÈGLES DE PASSAGE DU MODÈLE OBJET AU MODÈLE RELATIONNEL.

Elément UML	Elément relationnel	Commentaire
Classe	Table	
Attribut <i>identifiant</i>	Clé primaire	
Attribut	Colonne	
Association de multiplicité 1-1..*	Clé étrangère	Deux classes en association 1-* = deux tables dont une avec une FK.
Association de multiplicité 1-1	Clés étrangères « réciproques »	Deux classes en association 1-1 = deux tables + 2 FK d'auto-référencement.
Association de multiplicité *-*	Table de jointure avec une clé primaire composée de 2 clés étrangères	Deux classes en association *-* = trois tables dont une table de jointure avec une PK composée de 2 FK.
Classe association	Table de jointure avec une clé primaire composée de 2 clés étrangères et autant de colonnes que d'attributs de la classe-association	Deux classes en association *-* et une classe-association = trois tables dont une table de jointure avec une PK composée de 2 FK et des colonnes correspondantes aux attributs de la classe-association.

1.7 LE DIAGRAMME DE CLASSES TOTALES



2. BASE DE DONNÉES

2.1 INTRODUCTION

Une base de données sert à stocker des informations pour ensuite être capable de les traiter, trier .. afin d'en extraire soit des statistiques, récupérer des données ...

Pour accéder à la base de données, nous allons soumettre des requêtes en utilisant le langage SQL au SGBDR.

SGBD = **S**ystèmes de **G**estion de **B**ases de **D**onnées **R**elationnelle.

Les SGBDR sont les programmes qui se chargent du stockage de données.

La base de données travaille avec des tables. Chaque table est composé de champs que l'on appelle des colonnes et chaque table contient de 0 à plusieurs enregistrements qu'on appelle des lignes.

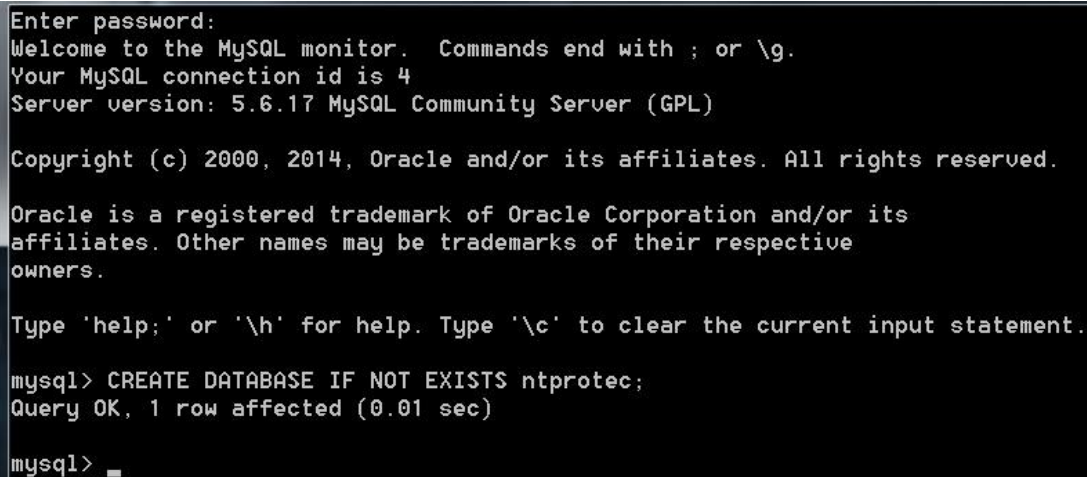
Le gérant dispose déjà de son hébergeur pour l'application.
C'est un moteur InnoDB celui-ci prend en charge les clefs étrangères.

Les clés primaires servent à identifier une ligne de manière unique.

Les clés étrangères, qui permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données.

2.2 CRÉATION DE LA BASE DE DONNÉES

Pour la base de données, je travaillerai avec un serveur local qui sera XAMPP pour développer plus rapidement.



```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE IF NOT EXISTS ntprotec;
Query OK, 1 row affected (0.01 sec)

mysql> _
```

FIGURE N° 30 : CREATION BASE DE DONNEES

2.3 CRÉATION D'UNE TABLE

Je suis passé par MySQL Workbench pour créer les tables.

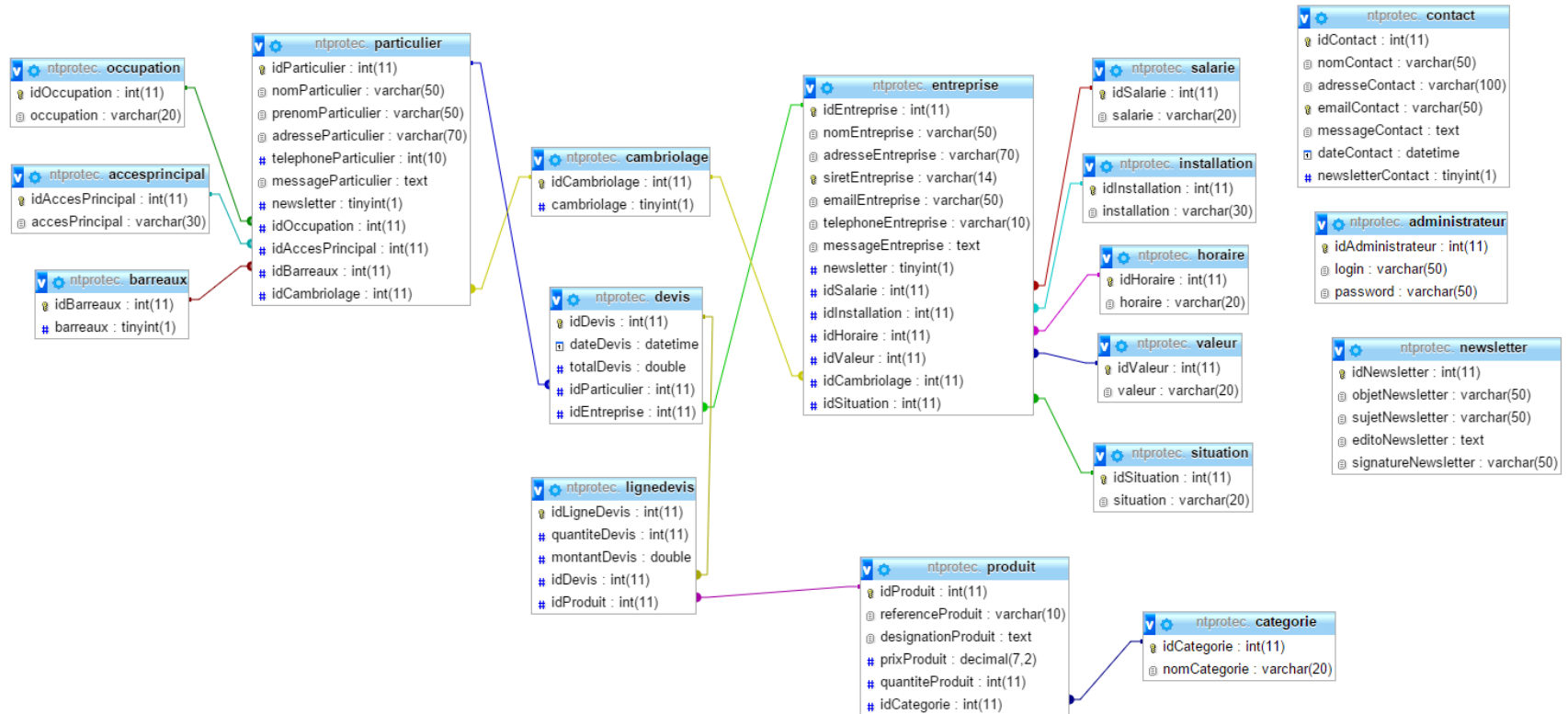
```
CREATE TABLE IF NOT EXISTS `contact` (  
  `idContact` int(11) AUTO_INCREMENT PRIMARY KEY,  
  `nomContact` varchar(50) NOT NULL,  
  `adresseContact` varchar(100) NOT NULL,  
  `emailContact` varchar(50) NOT NULL,  
  `messageContact` text NOT NULL,  
  `dateContact` datetime NOT NULL,  
  `newsletterContact` tinyint(1) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

FIGURE N° 31 : CREATION D'UNE TABLE

Quelques syntaxes pour ajouter les clés étrangères.

```
ALTER TABLE ligneDevis  
ADD CONSTRAINT fk_ligneDevis_devis foreign key (idDevis) REFERENCES devis(idDevis);  
  
ALTER TABLE entreprise  
ADD CONSTRAINT fk_entreprise_horaire foreign key (idHoraire) REFERENCES horaire(idHoraire);  
  
ALTER TABLE entreprise  
ADD CONSTRAINT fk_entreprise_situation foreign key (idSituation) REFERENCES situation(idSituation);
```

2.4 MODELE DE LA BASE DE DONNEES



3. ARCHITECTURE N-TIERS

Le choix de l'architecture : MVC

Pourquoi?

Une application web possède souvent une architecture 3-tiers éventuellement n-tiers. Son principal intérêt est la séparation des données (modèle), de l'affichage (vue) et les actions la logique métier (contrôleur).

Les avantages et coûts qu'apportent le MVC :

- Une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
- Un gain de temps de maintenance et d'évolution du site.
- Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l'affichage(webdesign)et des actions).

Le coût majeur du modèle MVC n'est visible qu'à la réalisation. Le fait de la séparation en différentes couches nécessite la création de plus de fichiers :

- Un fichier pour le modèle.
- Un fichier pour le contrôleur.
- Un fichier pour la vue.

MVC, comment ça marche?

L'architecture MVC est donc décomposée en trois étapes :

1. Le Modèle

Le modèle correspond aux données, la plupart du temps stockées dans une base de données mais elles peuvent également être contenues dans des fichiers XML ou des fichiers textes.

Les données peuvent être exploitées sous forme de classes, dans un langage de programmation orientée objet.

2. Le Contrôleur

Le contrôleur est l'élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue.

Le contrôleur peut donc instancier différents objets (classe User...) qui enverront des requêtes vers la base de données ou récupéreront des données XML.

La transmission à la vue des données récupérées se fait généralement à l'aide d'un Template : insertion des données dans un fichier HTML qui va être envoyé au navigateur.

3. La Vue

La vue correspond à un fichier HTML ou un fichier TPL avec un moteur de templates Smarty mais cela peut aussi être un fichier XML avec une feuille de style XSL. Elle se contente d'afficher le contenu qu'elle reçoit sans avoir connaissance des données.

Elle ne contient que des informations liées à l'affichage.

Dans mon cas :

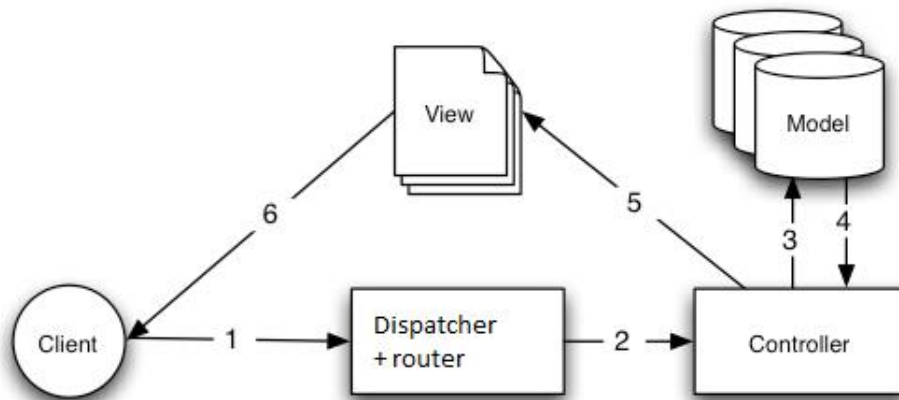


FIGURE N° 32 : EXEMPLE MVC

ICI ROUTER + DISPATCHER JE DOIS MODIFIER LE DESSIN

Le traitement d'une demande d'un utilisateur se déroule selon les étapes suivantes :

- 1/ Le client fait une demande au Router qui traite la demande .
- 2/ Le router l'envoie sur le dispatcher qui lui choisit le bon contrôleur.

Le contrôleur C traite cette demande. Pour ce faire, il peut avoir besoin de l'aide de la couche métier. Une fois la demande du client traitée, celle-ci peut appeler diverses réponses. Un exemple classique est :

- une page d'erreurs si la demande n'a pu être traitée correctement.
- une page de confirmation.

3/ Le contrôleur choisit la réponse (une vue) à envoyer au client. Choisir la réponse à envoyer au client nécessite plusieurs étapes:

- choisir l'objet qui va générer la réponse. C'est ce qu'on appelle la vue V, le V de MVC. Ce choix dépend en général du résultat de l'exécution de l'action demandée par l'utilisateur.
- lui fournir les données dont il a besoin pour générer cette réponse. En effet, celle-ci contient le plus souvent des informations calculées par le contrôleur. Ces informations forment ce qu'on appelle le modèle M de la vue, le M de MVC. L'étape 3 consiste donc en le choix d'une vue V et en la construction du modèle M nécessaire à celle-ci.

4/ Le contrôleur C demande à la vue choisie de s'afficher. Il s'agit le plus souvent de faire exécuter une méthode particulière de la vue V chargée de générer la réponse au client.

5/ La vue V utilise le modèle M préparé par le contrôleur C pour initialiser les parties dynamiques de la réponse qu'il doit envoyer au client.

6/ La réponse est envoyée au client.

4. L'ARBORESCENCE DES FICHIERS

L'interface de développement est NetBeans 8.0.2. C'est un logiciel placé en Open source.

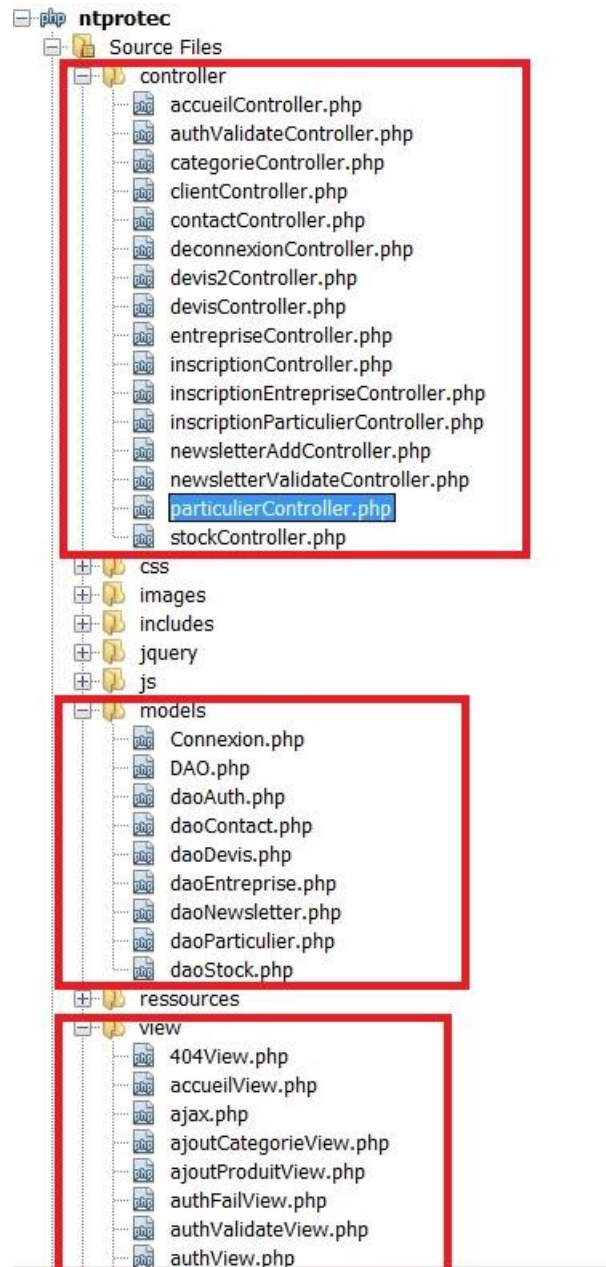


FIGURE N° 33 : ARBORESCENCES FICHIERS

5. CODES

Je vais vous présenter ici comment on se connecte à la base de données.

5.1 FICHIER CONFIG

J'utilise un fichier config.php pour stocker les paramètres de connexion à la base de données.

```
<?php

// Le système de gestion de la base de données SGBD
$protocole = "mysql";
// L'adresse de la base de données
$serveur="127.0.0.1";
// Le port logiciel
$port="3306";
// Le nom de la base de données
$bd="ntprotec";
// Le nom de l'utilisateur
$user="root";
// Le mot de passe pour accéder à la base de données
$pwd="ntprotec";

?>
```

FIGURE N° 34 : CODE FICHIER CONFIG.PHP

5.2 CONNEXION À LA BASE DE DONNÉES

```
class Connexion {

    /**
     *
     * @return \PDO
     *
     */
    private static function getPDO() {

        include_once 'includes/config.php';
        //Déclaration d'une variable pour l'objet de connexion
        $lcn = NULL;

        try {
            // Initialisation de la connexion
            $lcn = new PDO("$protocole:host=$serveur;port=$port;dbname=$bd;", $user, $pwd);
            // Gestion des erreurs en mode Exception
            $lcn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            // Méthode pour forcer l'encodage de PDO
            $lcn->exec("set names 'UTF8'");
        } catch (Exception $ex) {
            $lcn = NULL;
            echo 'Impossible de se connecter à la base';
            echo '<p>' . $ex . '</p>';
        }

        // On renvoie un objet de connexion à la base de données
        return $lcn;
    }
}
```

FIGURE N° 35 : CODE CONNEXION.PHP

5.3 AUTHENTIFICATION ET GESTIONS D'ERREURS

Bienvenue : Veuillez-vous connecter

Authentification

Login

Mot de passe

Envoyer Annuler

FIGURE N° 36: PAGE VIEW AUTHENTIFICATION

1ère vérification en JavaScript avec affichage d'erreurs sous forme de pop-up.

Pourquoi fait-on une première vérification en JavaScript avant celle de PHP?

Comme JavaScript est coté client, au moment de la soumission du formulaire, on peut vérifier les données transmises avant l'envoi et traitement du formulaire.

Fonction de validation du formulaire

```
function validationFormulaire()
{
    // création d'un tableau contenant l'ensemble des erreurs du formulaire
    var _listErreurs = new Array();
    var _elementHTML;
    var _txtErreurs = "";
    var _listElementsRequired = document.getElementsByClassName("required");

    document.getElementById("messageErreur").style.display = "none";
    for (var i = 0; i < _listElementsRequired.length; i++)
    {
        _elementHTML = _listElementsRequired[i];
        //JE ré-initialise la couleur de bordure du champ texte
        _elementHTML.style.borderColor = "#CCC";
        //tester le contenu du champ texte
        if (estRempli(_elementHTML.value) == false) {
            // Le champ est vide -> Ajouter un message d'erreur dans mon tableau
            _listErreurs.push(_elementHTML.title + "est Obligatoire !");
            _elementHTML.style.borderColor = "#900";
        }
    }
}
```

```
/*
Je vérifie le nombre d'élément dans mon tableau d'erreur si j'ai plus de 0 éléments ( donc au moins 1 erreur )
le formulaire n'est pas validé
_listErreurs.length : compter le nombre d'élément d'un tableau
*/
if (_listErreurs.length > 0)
{
    //parcourir l'ensemble de mon tableau pour récupérer chaque message d'erreur
    for (var i = 0; i < _listErreurs.length; i++)
    {
        _txtErreurs += "<li>" + _listErreurs[i] + "</li>";
    }
    _txtErreurs = "<ul>" + _txtErreurs + "</ul>";
    // je cible une balise HTML Particulière grâce à son id,
    //et j'ordonne l'écriture du contenu de ma variable à l'intérieur de celle-ci
    document.getElementById("listErreurs").innerHTML = _txtErreurs;
    document.getElementById("messageErreur").style.display = "block";
} else {
    //J'envoie les données du formulaire
    document.form.submit();
}
return false;
}
```

FIGURE N° 37 : CODE VERIFICATION JS

Fonction fermerMessageErreur

```

function fermerMessageErreur()
{
    // Rechercher le POP-UP "messageErreur" et lui appliquer le style display:none
    document.getElementById("messageErreur").style.display = "none";
    return false;
}

//validation des modifications apporter par l'utilisateur
// _elHTML: élément HTML qui a émis l'évènement onChange
function validationElement(element_HTML)
{
    //ré-initialisation de la bordure de mon champ texte
    element_HTML.style.borderColor = "#CCC";

    //vérification si le champ contient la propriété required
    if (element_HTML.className == "required") {
        //Tester le contenu saisi
        if (estRempli(element_HTML.value) == false)
        {
            element_HTML.style.borderColor = "#900";
        }
    }
}

```

Fonction estRempli

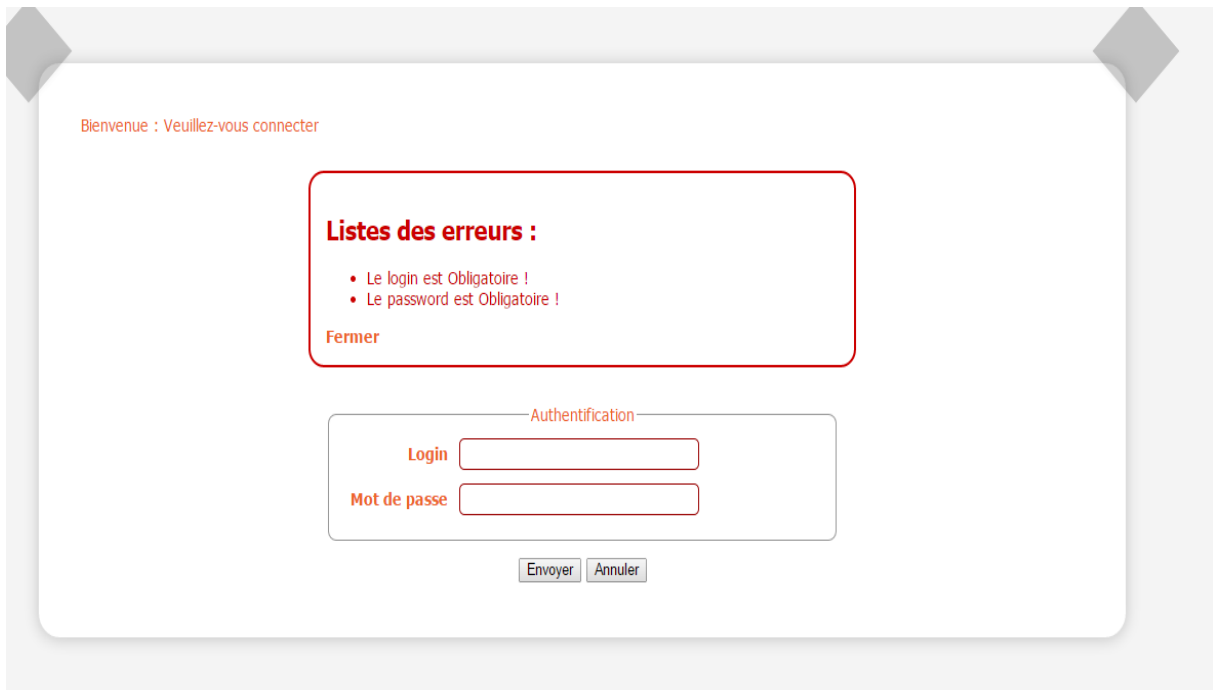
```

/**
 * FUNCTION DE VALIDATION
 */
//définition de la fonction permettant de vérifier qu'une valeur a été saisie
function estRempli(valeurAtester)
{
    // on instancie _resultat à false
    var _resultat = false;
    // Si la valeur tester est différent d'une chaîne de caractère vide et qu'il est différent de null
    if (valeurAtester != "" && valeurAtester != null)
    // le _resultat passe à true
        _resultat = true;
    // on retourne _resultat
    return _resultat;
}

```

FIGURE N° 38 : VERIFICATION CODE JS SUITE

Voici le résultat si les champs sont vides :



The image shows a web interface for user authentication. At the top, it says "Bienvenue : Veuillez-vous connecter". Below this is a red-bordered box titled "Listes des erreurs :" containing two bullet points: "Le login est Obligatoire !" and "Le password est Obligatoire !". A "Fermer" button is located at the bottom left of this box. Below the error box is the "Authentification" section, which includes labels "Login" and "Mot de passe" next to empty input fields. At the bottom of this section are two buttons: "Envoyer" and "Annuler".

FIGURE N° 39 : POPUP ERREUR VIA JS

Si tout se passe bien on envoie le formulaire. C'est ici qu'il y aura une seconde vérification.

Il faut une seconde vérification coté serveur cette fois ci en PHP.

Il est possible que le JavaScript soit désactivé donc la 1ere étapes de vérification saute. Il faut donc vérifier les informations reçus via le formulaire qui a été envoyer.

Cette vérification va permettre de vérifier si les informations sont correctes.

Ici, dans ma classe authDAO, j'ai une méthode qui permet exécuter comparer les données entrées avec celle contenue dans la base de données.

```
// Initialisation depuis la BDD
public function getadministrateurDB() {
    // Etablissement de la connexion
    $lcn = Connexion::getConnexion();
    // Préparation de la requête
    $sql = $lcn->prepare('select * from administrateur where login=:login and password=:password');
    try {
        // On envoi la requête
        $sql->execute(array('login' => $this->login, 'password' => $this->password));
        // Traitement des résultats et on le retourne dans un tableau associatif
        $result = $sql->fetchAll(PDO::FETCH_ASSOC);
        //On retourne le résultat
        return $result;
    } catch (Exception $e) {
        return 'Erreur de requête : ' . $e->getMessage();
    }
}
```

FIGURE N° 40 : CODE AUTHDAO

Dans mon contrôleur authValidate

```

<?php

require_once 'models/daoAuth.php';
// On récupère le post[login] pour la stocker dans une variable $login
//FILTER_SANITIZE_SPECIAL_CHARS : Transforme en entité HTML les caractères "<>&" et les caractères ASCII
// de valeur inférieur à 32, et supprime ou encode les autres caractères spéciaux.
$login = filter_input(INPUT_POST, 'login', FILTER_SANITIZE_SPECIAL_CHARS);
// On récupère le post[password] pour la stocker dans une variable $password
$password = filter_input(INPUT_POST, 'password');

//Retourne la valeur de la variable demandée en cas de succès,
//FALSE si le filtre échoue, ou NULL si la variable variable_name n'est pas définie.

if ($login && $password) {

    try {

        //On instancie un nouveau DAO
        $adminDAO = new daoAuth();
        //On set le login et le password avec les informations recus dans le get
        $adminDAO->set_login($login);
        $adminDAO->set_password($password);
        //on récupère les valeurs comparéssous forme de tableau
        $tVerification = $adminDAO->getadministrateurDB();
        //Si le tableau est vide alors:
        if (empty($tVerification)) {
            // on redirige vers la vue d'erreur
            header("Location:http://localhost/ntprotec/authFail");
        } else {
            //Sinon on redirige vers l'accueil
            header("Location:http://localhost/ntprotec/accueil");
            $_SESSION["connecte"] = '1';
        }
    } catch (Exception $exc) {
    }
} else {
    //Redirige vers la 2eme vue d'erreur si les champs sont vides
    header("Location:http://localhost/ntprotec/authFail2");
}
?>

```

FIGURE N° 41 : CODE AUTHVALIDATE

En cas d'erreur, on a bien la vue authFailView.php

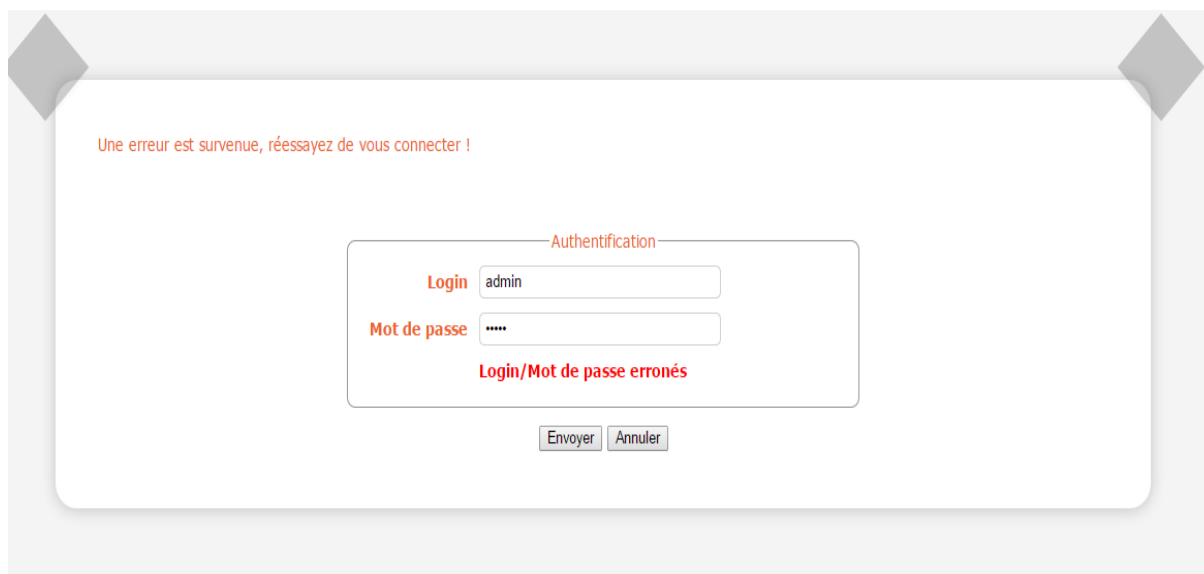


FIGURE N° 42 : ERREUR VIA PHP

Si tout se passe bien, on est bien redirigé vers la page d'accueil :

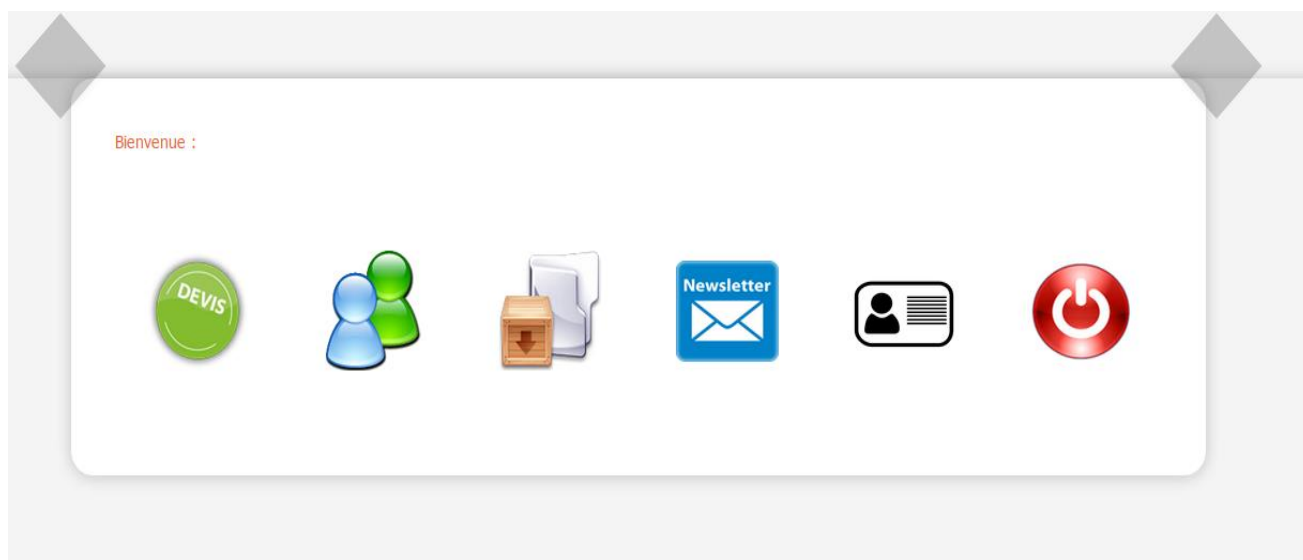


FIGURE N° 43 : PAGE ACCUEIL

5.4 MA ROUTE

Dans notre cas le fichier .htaccess est utilisé pour réécrire l'url.

Fichier .htaccess à la racine du site

```
#On autorise la réécriture
RewriteEngine On
#On accepte tous les fichiers terminant par .css|png|jpg|js
RewriteCond %{REQUEST_FILENAME} !\.(css|png|jpg|js)$
#On redirige tout sur l'index.php
RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]
```

FIGURE N° 44 : FICHIER HTACCESS

Mon index.php me servira de route.

```
<?php
// On lance une session, il sert pour l'authentification
session_start();
// Pour la partie test en production
//define('BASE_URI', '../');
// Pour la partie developpement local
define('BASE_URI', '../ntprotec/');

$missMatch = false;
//On contrôle de l'url
if (!empty($_GET['url'])) {
    //on decoupe l'url avec les /
    $url = explode('/', $_GET['url']);
    // on vérifie le nombre d'élément passé dans l'url
    $element_url = sizeof($url);
    $index = 0;

    while ($index < $element_url) {
        if (!preg_match("#^[a-zA-Z0-9]*$#", $url[$index])) {
            $missMatch = true;
            break;
        }
        $index++;
    }
} else {
    // URL est vide
    $url[0] = "";
}
```

FIGURE N° 45 : CODE ROUTE

```

//Condition qui permet la gestion de l'authentification
if (!isset($_SESSION["connecte"]) && ($url[0] != "authValidate") && ($url[0] != "authFail") && ($url[0] != "authFail2")) {
    $url[0] = "auth";
}
// Met le contenu en tampon
ob_start();
//Si url détecte l'appelle de la page ajax.php on le laisse afficher cette page
if ($url[0] == "ajax.php") {
    include_once BASE_URI . "view/ajax.php";
} else {
    // On redirige ici vers l'accueil
    if ($missMatch && empty($url[0])) {
        // Traitement : page 404 ou renvoi sur l'accueil
        //include_once "view/404View.php";
        include_once "view/accueilView.php";
    } else {
        //Sinon on utilise le switch case qui permet à l'url d'afficher soit une vue | vue + controlleur | un controlleur
        switch ($url[0]) {

            case "auth": case "authFail": case "authFail2": case "newsletter": case "client": case "accueil": case "devis2": case "inscriptionParticulier":
            case "ajoutProduit": case "ajoutCategorie":
                include_once BASE_URI . "view/" . $url[0] . "View.php";
                break;

            case "authValidate": case "deconnexion": case "categorie":
                include_once BASE_URI . "controller/" . $url[0] . "Controller.php";
                break;

            case "newsletterValidate": case "entreprise": case "stock": case "contact":
                include_once BASE_URI . "controller/" . $url[0] . "Controller.php";
                include_once BASE_URI . "view/" . $url[0] . "View.php";
                break;

            default:
                include_once BASE_URI . "view/accueilView.php";
                break;
        }
    }
}
//Lit le contenu courant du tampon de sortie puis l'efface.
$corps = ob_get_clean();
include_once 'includes/header.php';
include_once 'includes/page.php';
}
?>

```

FIGURE N° 46 : CODE ROUTE SUITE

5.5 RECHERCHE DU STOCK VIA AJAX ET JQUERY

AJAX = Asynchronous JavaScript and XML

Le langage JavaScript est utilisé pour demander des données au serveur. Ces données lui sont retournées de façon asynchrone sous une forme XML.

Grâce à Ajax, la recherche d'un produit qui est dans la liste de stock sans rechargement de la page.

Dans ma page stockView, j'ai ajouté un formulaire de recherche.

```
<form class="ajax" action="" method="post">
  <p>
    <label for="q" style="color: blue">Rechercher un produit</label>
    <input type="text" name="q" id="q" />
  </p>
</form>
<div>
```

FIGURE N° 47 : FORMULAIRE RECHERCHE

Voici le script Ajax-jQuery

```
<script>
$(document).ready(function () {
    // détection de la saisie dans le champ de recherche
    $('#q').keyup(function () {
        $field = $(this);
        // alert($field.val());
        $('#results').html(''); // on vide les resultats
        $('#ajax-loader').remove(); // on retire le loader
        // on envoie la valeur recherché en GET au fichier de traitement
        $.ajax({
            type: 'GET', // envoi des données en GET ou POST
            url: 'ajax.php', // url du fichier de traitement
            data: 'q=' + $field.val(), // données à envoyer en GET ou POST
            cache: false,
            beforeSend: function () { // traitements JS à faire AVANT l'envoi
                $field.after(''); // ajout d'un loader pour signifier l'action
            },
            success: function (data) { // traitements JS à faire APRES le retour du controller
                $('#ajax-loader').remove(); // on enleve le loader
                $('#results').html(data); // affichage des résultats dans le bloc
            }
        });
    });
});
</script>
```

FIGURE N° 48 : CODE AJAX JQUERY

Code ajax.php

```

<?php
//
require_once 'models/Connexion.php';
//connexion à la base de données
$lc = Connexion::getConnexion();
// récupération de la valeur dans le get
$q = filter_input(INPUT_GET, 'q');
// Préparation de la requête
$sql = "SELECT produit.idCategorie,
        categorie.nomCategorie,
        produit.idProduit,
        produit.referenceProduit,
        produit.designationProduit,
        produit.prixProduit,
        produit.quantiteProduit
FROM ntprotec.produit produit
      INNER JOIN ntprotec.categorie categorie
      ON (produit.idCategorie = categorie.idCategorie)";

//Si on détecte une valeur dans le champs de recherche, on concatène avec un where
if ($q) {
    // ajout à la requête
    $sql .= " where referenceProduit like '%" . $q . "%'";
}

//
$req = $lc->prepare($sql);
//on execute la requête
$req->execute();
//on mets dans une variable le resultat du count
$count = $req->rowCount();
//vérifie si le tableau est vide
$nbReq = empty($req);

if ($count == 0) {
    echo 'Le produit'. ' '. $q. ' '. 'est inexistant';
} else {
    //retourne un tableau
    $tstock = $req->fetchAll();
}

?>

```

FIGURE N° 49 : CODE AJAX JQUERY SUITE

Affichage du block HTML

```

<table width="1000px" cellspacing="1" cellpadding="2" class="listeProduit">

    <tr>

        <td width="130" >REF</td>
        <td width="500" >DESIGNATION</td>
        <td width="200" >PRIX HT</td>
        <td width="204" >QUANTITE</td>
        <td width="204" >CATEGORIE</td>

        <td align="center" >Voir</td>
        <td align="center">Modifier</td>
        <td align="center">Effacer</td>
    </tr>
    <?php
    foreach ($tstock as $stock) {
    ?>

    <tr id="<?php echo $stock["referenceProduit"]; ?>">

        <td width="130"><?php echo $stock["referenceProduit"]; ?></td>
        <td width="500" ><?php echo $stock["designationProduit"]; ?></td>
        <td width="130"><?php echo $stock["prixProduit"]; ?></td>
        <td width="204" ><?php echo $stock["quantiteProduit"]; ?></td>
        <td width="204" ><?php echo $stock["nomCategorie"]; ?></td>

        <td align="center" ></td>
        <td align="center"></td>
        <td align="center"></td>
    </tr>

    <?php
    }
    ?>

    <tr>
        <td width="150" rowspan="3" align="center" valign="middle">

        </td>
        <td height="10" colspan="2" valign="top">&nbsp;</td>
    </tr>
</table>

```

FIGURE N° 50 : AFFICHAGE BLOC HTML

Les Produits

Rechercher un produit

Ajouter un produit

Catégorie

REF	DESIGNATION	PRIX HT	QUANTITE	CATEGORIE	Voir	Modifier	Effacer
NPH300	Platine vidéo Andy conforme à la loi Handicap'	675.00	15	Alarme			
PC9155	Centrale d'alarme professionnelle DSC Alexor	465.00	3	Alarme			
WT5500	Clavier de programmation et de contrôle	465.00	4	Alarme			
NPH315	Platine Andy	895.45	2	Alarme			
NPH400	Platine Andy de 2013	1050.30	4	Alarme			
NPH500	Platine Andy de 2014	1250.30	5	Alarme			
NPH500	Platine Andy de 2015	1400.00	5	Alarme			
NPH600	Modèle Platine Andy nouvelle génération	1600.00	3	Alarme			
WS4904P	Détecteur de présence	78.00	5	Détecteur			
VXI-RAM	Détecteur infrarouge extérieur	185.00	6	Détecteur			

FIGURE N° 51 : PAGE STOCK INITIALE

Les Produits

Rechercher un produit

Ajouter un produit

Catégorie

REF	DESIGNATION	PRIX HT	QUANTITE	CATEGORIE	Voir	Modifier	Effacer
NPH300	Platine vidéo Andy conforme à la loi Handicap'	675.00	15	Alarme			
NPH315	Platine Andy	895.45	2	Alarme			
NPH400	Platine Andy de 2013	1050.30	4	Alarme			
NPH500	Platine Andy de 2014	1250.30	5	Alarme			
NPH500	Platine Andy de 2015	1400.00	5	Alarme			
NPH600	Modèle Platine Andy nouvelle génération	1600.00	3	Alarme			

FIGURE N° 52 : PAGE STOCK AVEC UNE RECHERCHE

Si le produit n'existe pas

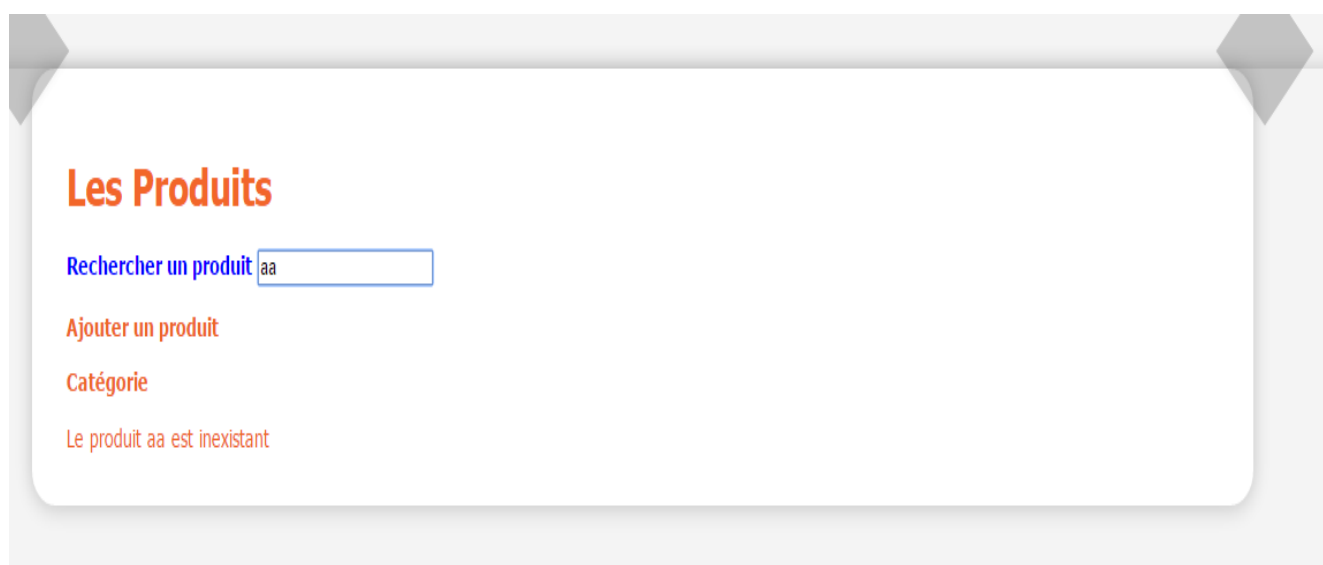


FIGURE N° 53 : PAGE RECHERCHE NON EXISTANT

5.6 DIAGRAMME DE DÉPLOIEMENT

Le diagramme de déploiement montre 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'une voie de communication.

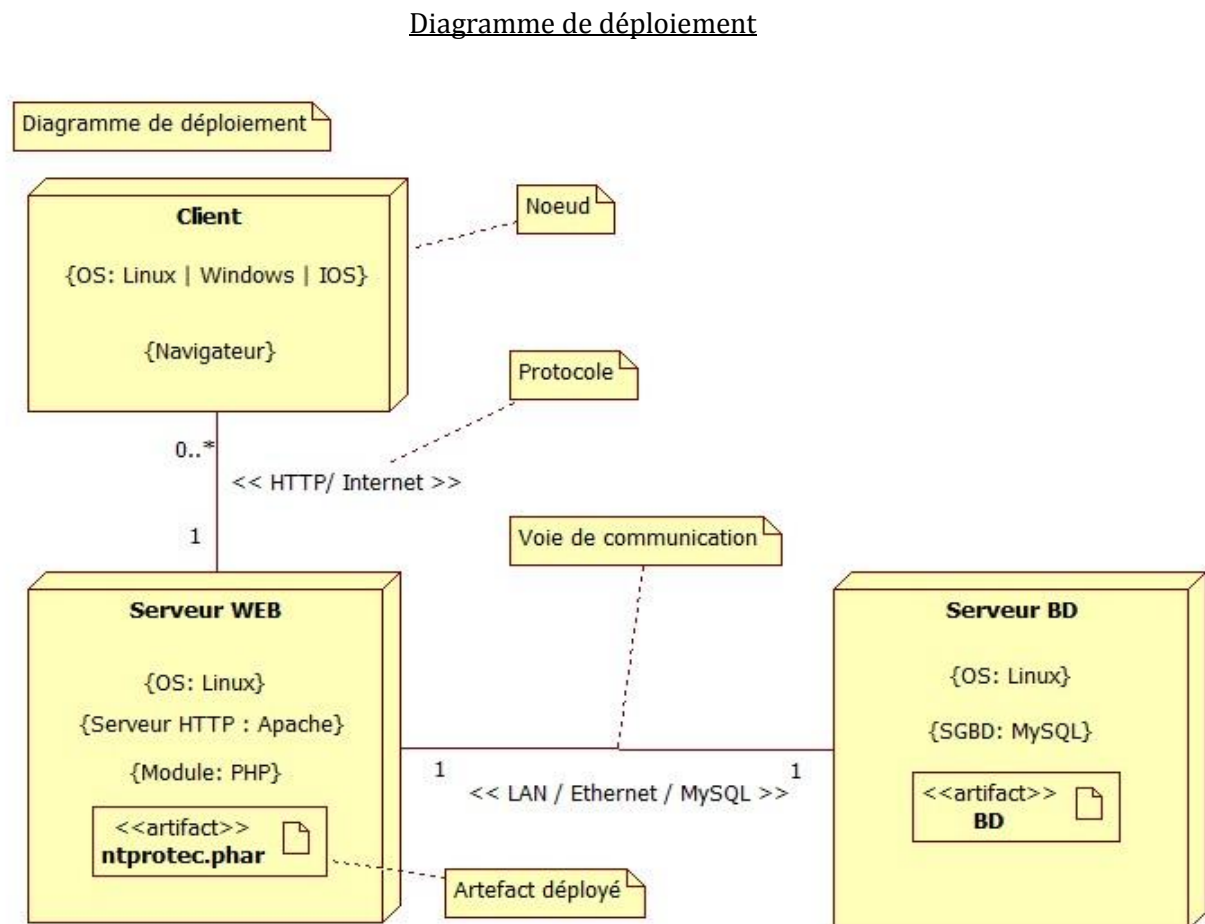


FIGURE N° 54 : DIAGRAMME DE DEPLOIEMENT

Ici, nous avons une architecture d'une application web à 3 niveaux.

Ce diagramme représente le client et son navigateur, le serveur WEB et le serveur BD avec un SGBDR.

Le client accède aux données de la BD MySQL située sur un serveur linux via HTTP Apache.

6. LES LOGICIELS UTILISÉS

Fonctionnalité	Langages et/ou Outils
Méthodologies et Gestion de projet	UML, Gantt Projet
AGLs	StarUML, PowerAMC
IDE	NetBeans
Maquettes	Photoshop
Réalisation des interfaces statiques	HTML,CSS, BootStrap
Réalisation de la dynamique des interfaces	CSS, JavaScript,jQuery,Ajax
SGBDR	MySQL
Clients du SGBDR pour développeur et administrateur de la BD	phpMyAdmin, Xampp, MySQL WorkBench, Toad For MySQL
Langage de développement coté serveur	PHP avec PDO, POO

7. CONCLUSION

BILAN

Le site internet front-office est terminé, il ne reste plus qu'à le mettre en ligne. L'étape suivante sera le référencement.

Le back-office n'est pas totalement terminé. Il reste à gérer :

- les problèmes liés à l'envoi massif de mail (éviter de se faire black-lister).
 - ✓ La solution serait de passer par un relais SMTP comme par exemple JANGO SMTP. Il permet même un suivi des emails envoyés comme par exemple le nombre d'email ouvert ou la gestion des échecs ...
- la gestion de devis : Il reste plus qu'à faire la conversion et enregistrement en PDF.

LE STAGE

Ce stage m'a permis de revoir certaines connaissances apprises lors de mes 4 mois de formation et ainsi consolider mes acquis en développement web.

J'ai également pu constater que la communication est très importante. On croit comprendre quelques choses mais en fait ce n'était pas ça.

La mise en place de la partie conception est certes un peu longue mais très importante. Elle m'a permis de ne pas perdre trop de temps pendant le développement.

La mise en place d'un petit MVC m'a vraiment mis l'eau à la bouche.

PROJET FUTUR

Je suis conscient que cette formation n'est qu'une étape pour devenir un bon Concepteur-Développeur. Il me reste encore beaucoup de choses à apprendre.

Je souhaiterais me spécialiser dans un Framework PHP comme Symfony2.

8. ABSTRACT

RESULTS

The front office website is completed, we just have to put it online. The next step will be referencing.

The back office is not completely finished. It remains to manage:

- the problems with sending mass mail (avoid getting backlisted).
 - ✓ The solution would be to go through an SMTP relay such as JANGO SMTP. There also have an emailtracking system .
- Quotation management: It just has to code the conversion to PDF.

INTERNSHIP

This internship allowed me to review some knowledge learned during my 4 months at M2i Formation and also consolidate my knowledge in web development.

I also realise that communication is very important. You think to understand some things but actually it was not that.

The implementation part is a bit long but very important. That help me to not lose too much time during coding.

FUTURE PROJECT

I am aware that this traineeship is only one step to become a good -developer. I still have a lot to learn.

I will wish to specialize in a PHP framework like Symfony2.

9 . ANNEXE

AUTRES DESCRIPTIONS DÉTAILLÉES

Fonction : Gérer les devis	
Objectif	Visualisation la demande de devis, avec possibilité de suppression
Description	Affichage du devis ou suppression du devis
Contraintes / Règles de gestion	Être authentifié comme gérant
Niveau de priorité	Moyenne

Fonction : Imprimer un devis	
Objectif	Imprimer un devis
Description	Lien pour télécharger le devis au format PDF
Contraintes / Règles de gestion	
Niveau de priorité	Moyenne

DIAGRAMMES DE NAVIGATION

Diagramme de navigation devis et contact

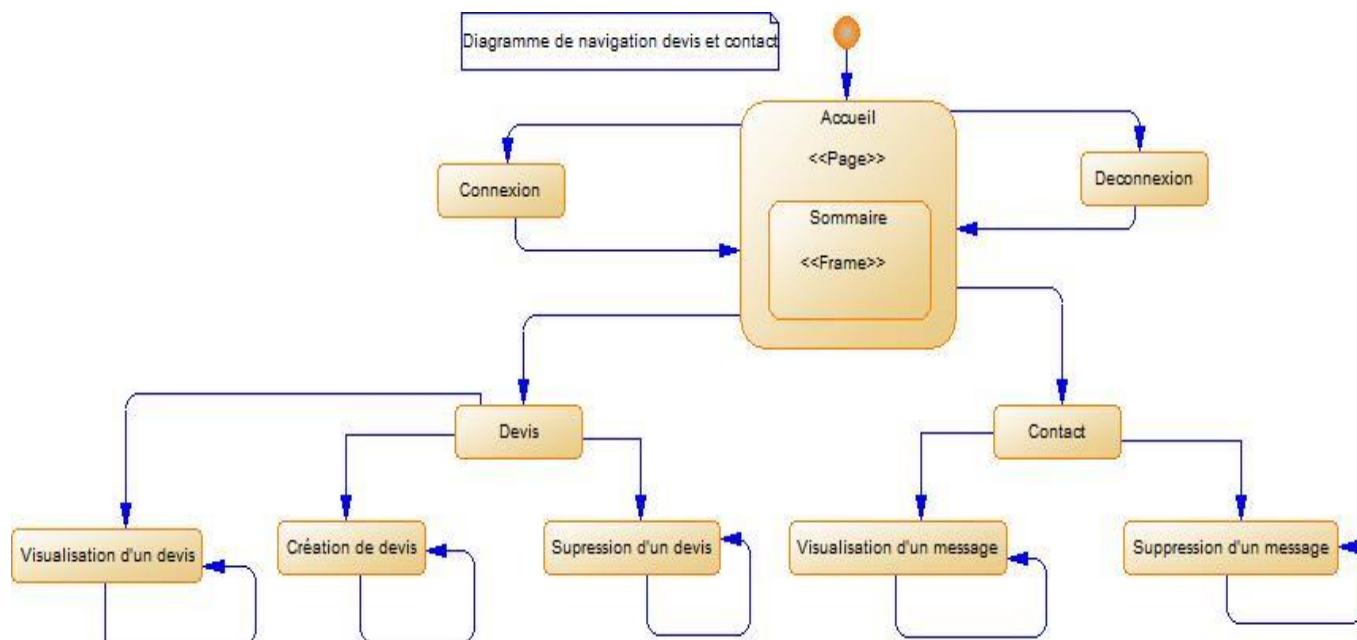


DIAGRAMME DE NAVIGATION CLIENTS

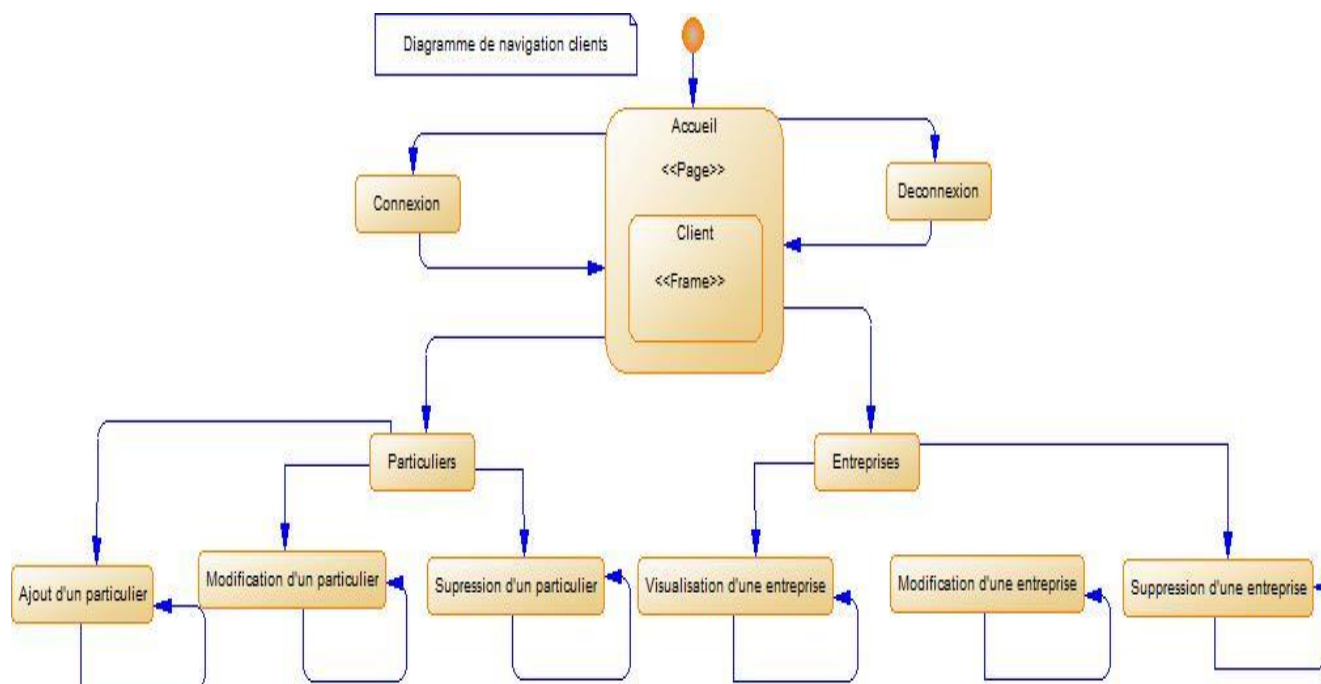


DIAGRAMME D'ACTIVITE

Diagramme d'activité newsletter

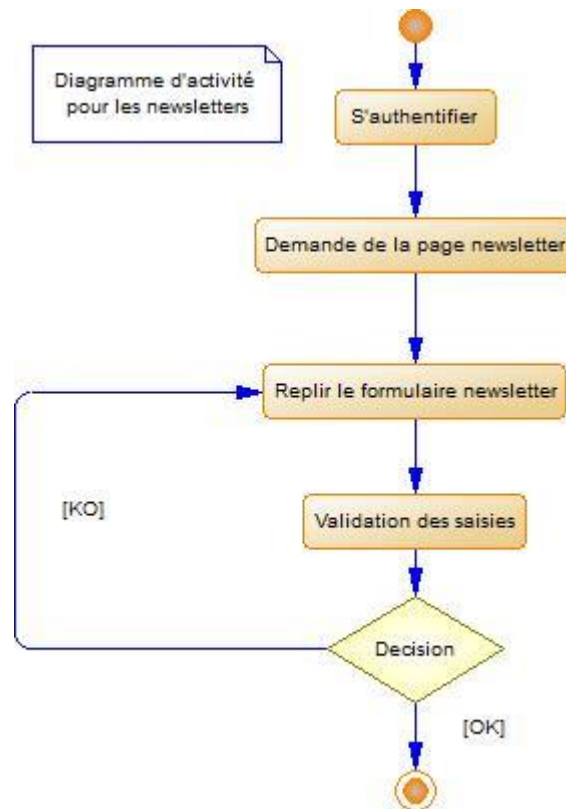


Diagramme d'activité Devis

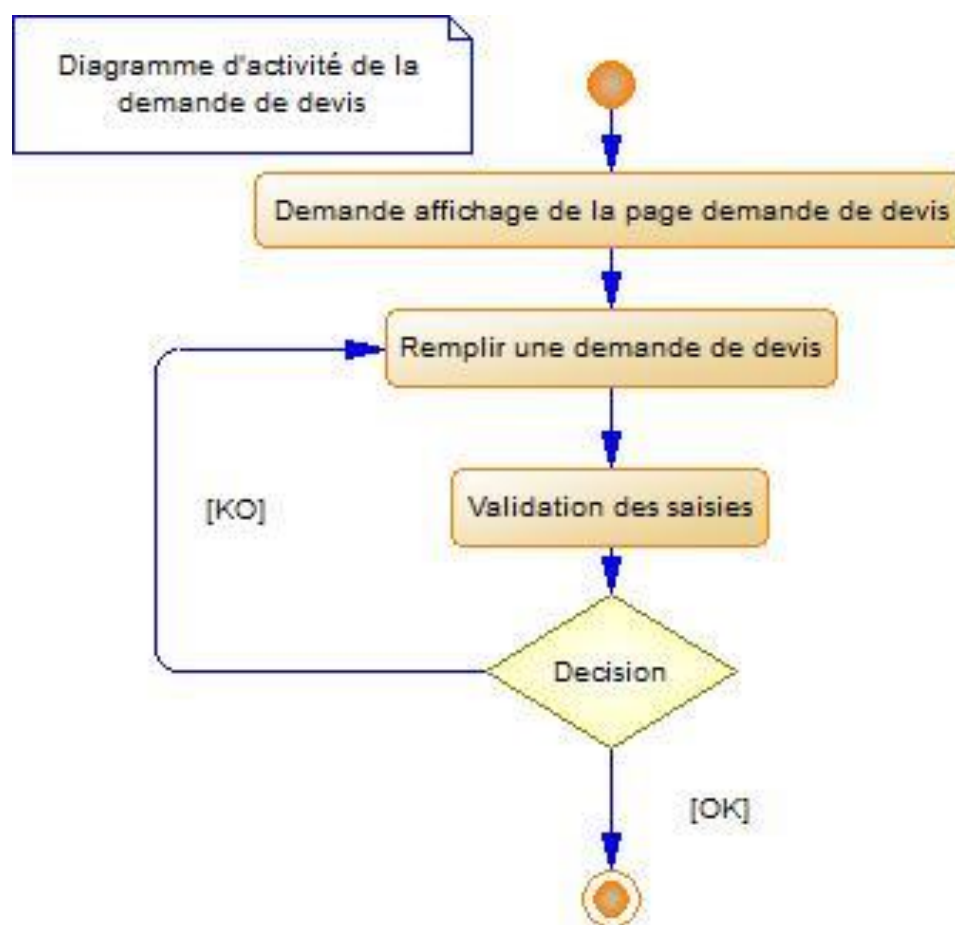


Diagramme d'activité Clients

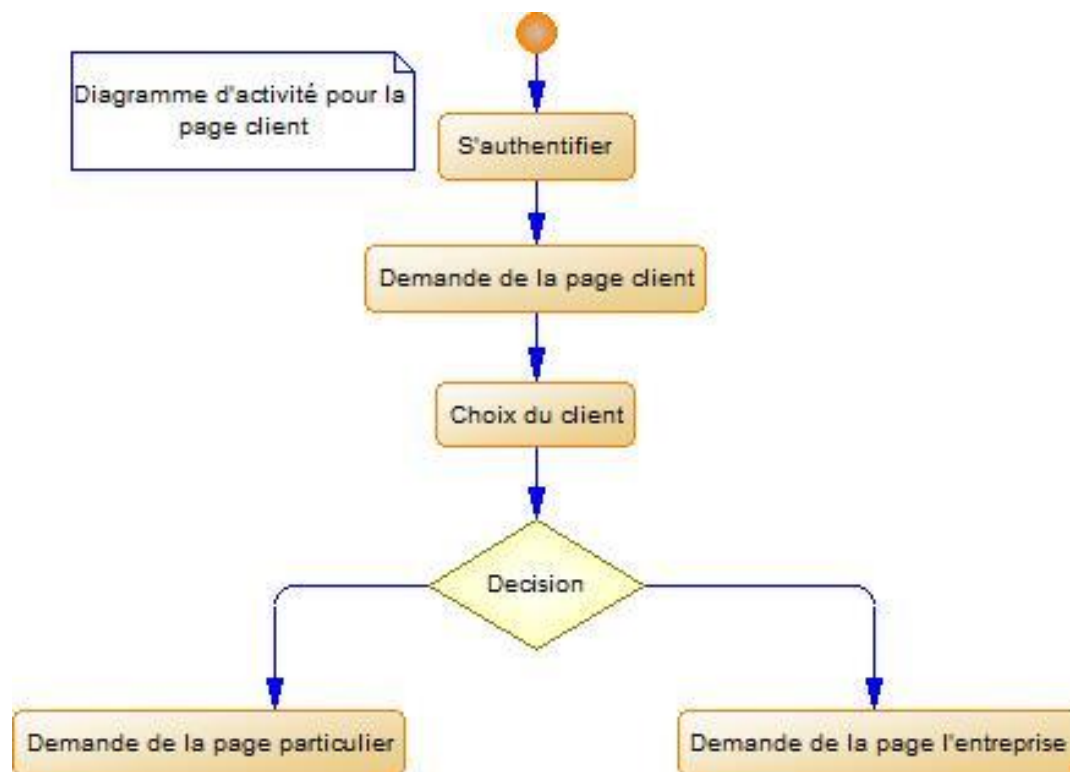


Diagramme d'activité Clients suite

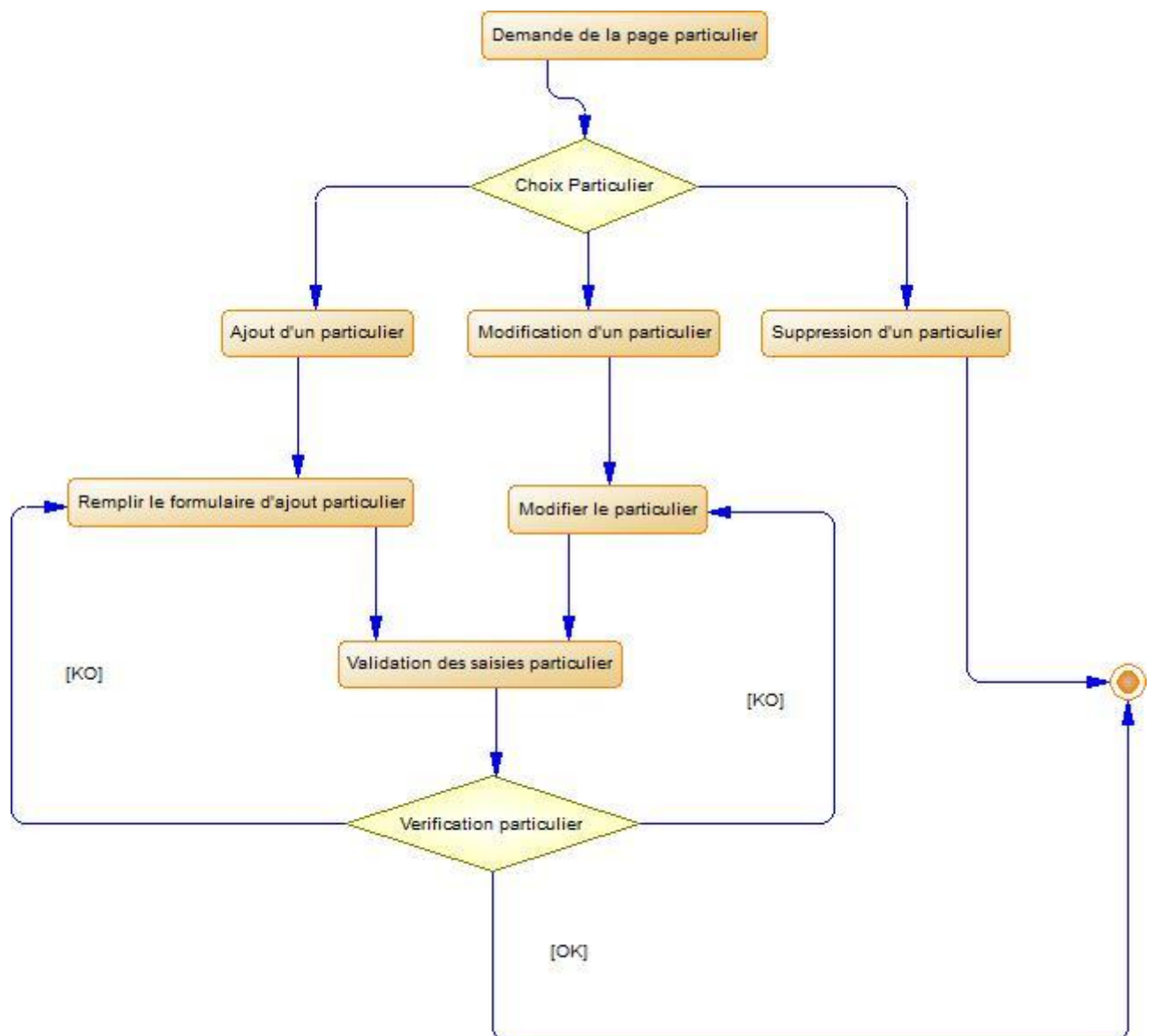
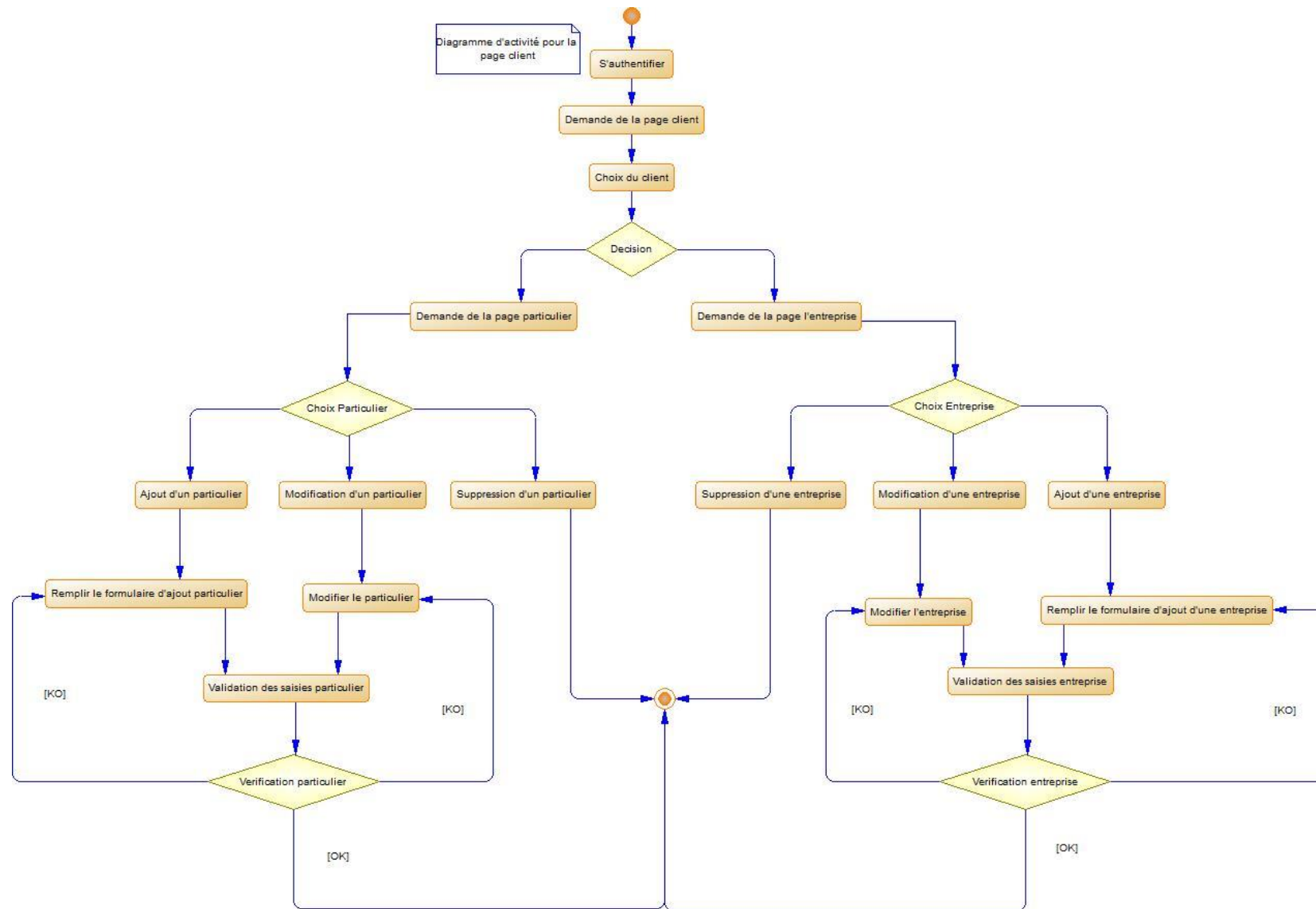
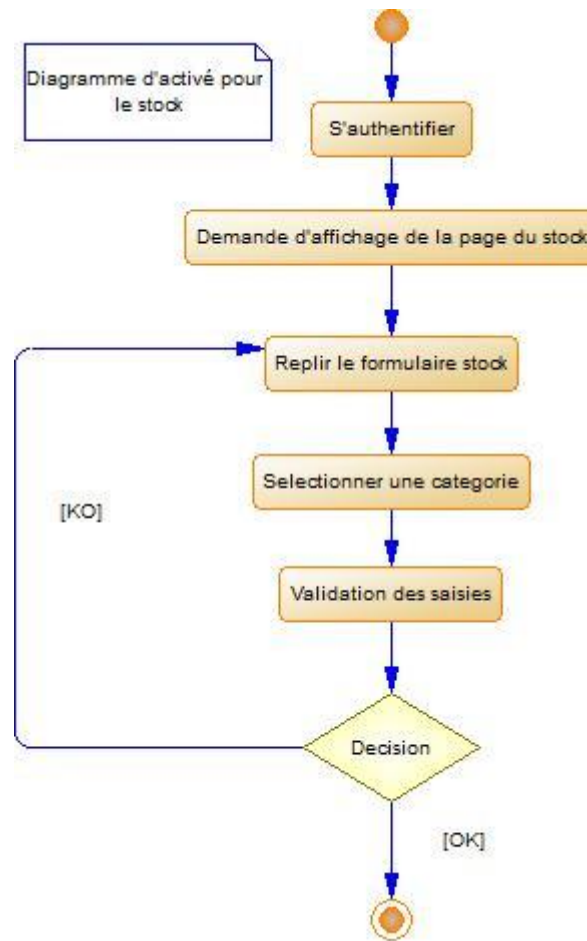
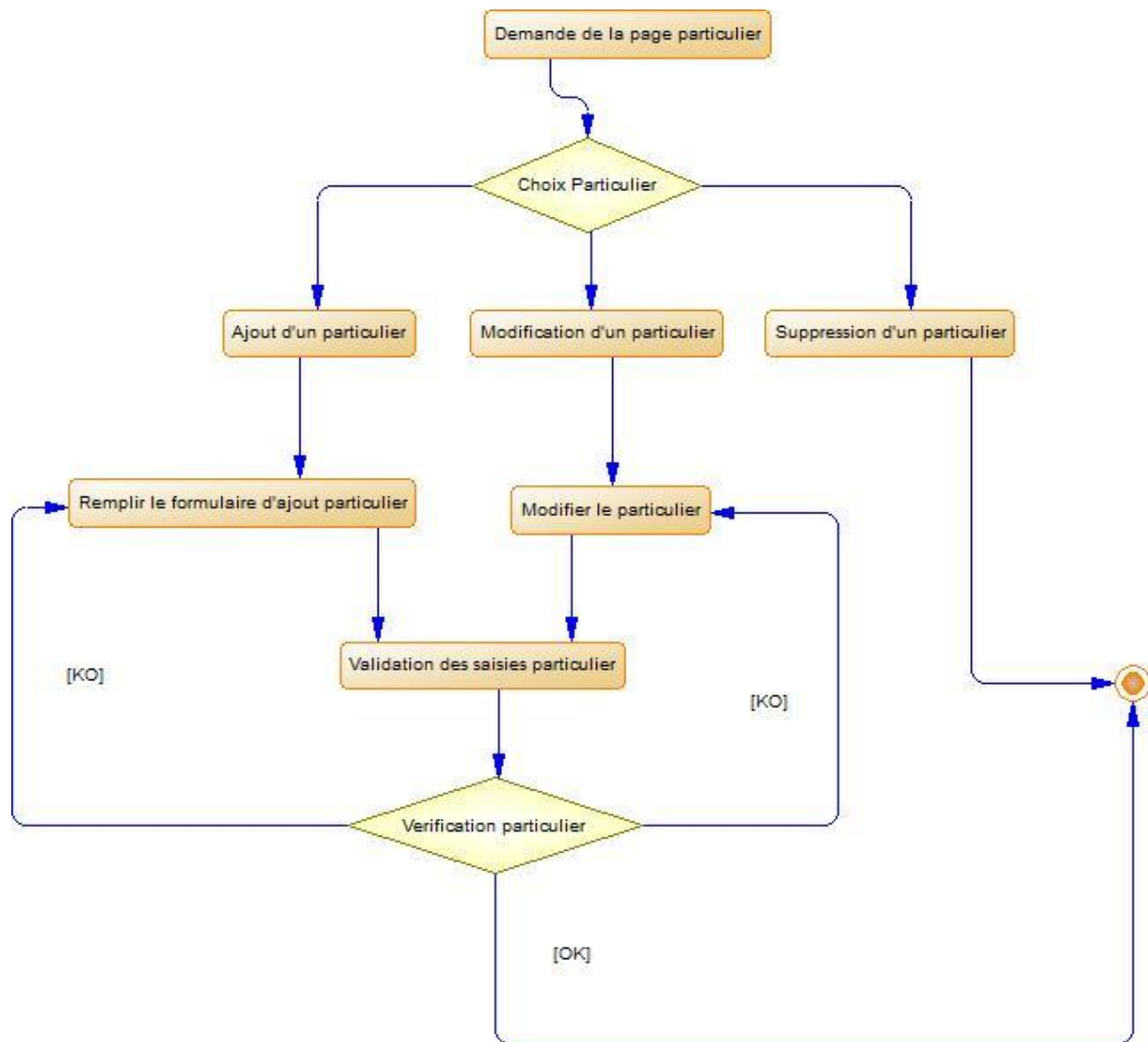


Diagramme d'activité général clients







LA TABLE DES ILLUSTRATIONS

<i>Figure n° 1 : Ancien site nt Protec.....</i>	9
<i>Figure n° 2 : Diagramme de cas d'utilisation internaute.....</i>	10
<i>Figure n° 3 : diagramme de cas d'utilisation entreprise.....</i>	11
<i>Figure n° 4 : nouveau site front office.....</i>	19
<i>Figure n° 5 : page devis front office.....</i>	21
<i>Figure n° 6 : page contact front office.....</i>	22
<i>Figure n° 7 : page d'authentification bo.....</i>	23
<i>Figure n° 8 : page d'authentification avec erreur.....</i>	24
<i>Figure n° 9 : page d'accueil back office.....</i>	25
<i>Figure n° 10 : page client back office.....</i>	26
<i>Figure n° 11 : liste entreprises back office.....</i>	27
<i>Figure n° 12 : formulaire ajout entreprise back office.....</i>	28
<i>Figure n° 13 : interface stock back office.....</i>	29
<i>Figure n° 14 : interface newsletter back office.....</i>	31
<i>Figure n° 15 : diagramme de navigation fo.....</i>	32
<i>Figure n° 16 : diagramme de navigation CONTACT stock et devis.....</i>	33
<i>Figure n° 17 : Diagramme de navigation newsletter et stock.....</i>	34
<i>Figure n° 18 : Diagramme de navigation clients.....</i>	Erreur ! Signet non défini.
<i>Figure n° 19 : Diagramme de navigation general.....</i>	35
<i>Figure n° 20 : Diagramme d'activite page devis.....</i>	36
<i>Figure n° 21 : Diagramme d'activite page contact.....</i>	Erreur ! Signet non défini.
<i>Figure n° 22 : Diagramme d'activite page authentification.....</i>	37
<i>Figure n° 23 : Diagramme d'activite ajoute au stock.....</i>	38
<i>Figure n° 24 : Diagramme d'activite page clients.....</i>	Erreur ! Signet non défini.
<i>Figure n° 25 : Diagramme d'activite page client suite.....</i>	Erreur ! Signet non défini.
<i>Figure n° 26 : Diagramme de sequence page contact.....</i>	39

<i>Figure n° 27 : diagramme de sequence authentication</i>	<i>40</i>
<i>Figure n° 28 : Diagramme de sequence stock</i>	<i>41</i>
<i>Figure n° 29 : Diagramme de deploiement</i>	<i>79</i>
<i>Figure n° 30 : diagramme d'etats transition devis</i>	<i>42</i>
<i>Figure n° 31 : modele devis</i>	<i>45</i>
<i>Figure n° 32 : graphe des dependances fonctionnelles.....</i>	<i>50</i>
<i>Figure n° 33 : graphe des dependances fonctionnelles multi-sources</i>	<i>51</i>
<i>Figure n° 34 : diagramme de classes</i>	<i>53</i>
<i>Figure n° 35 : creation base de donnees.....</i>	<i>56</i>
<i>Figure n° 36 : creation d'une table</i>	<i>57</i>
<i>Figure n° 37 : exemple mvc</i>	<i>61</i>
<i>Figure n° 38 : arborescences fichiers.....</i>	<i>62</i>
<i>Figure n° 39 : code fichier config.php</i>	<i>63</i>
<i>Figure n° 40 : code connexion.php.....</i>	<i>64</i>
<i>Figure n° 41: page view authentication</i>	<i>64</i>
<i>Figure n° 42 : code verification js</i>	<i>65</i>
<i>Figure n° 43 : verification code js suite.....</i>	<i>66</i>
<i>Figure n° 44 : popup erreur via js</i>	<i>67</i>
<i>Figure n° 45 : code authdao.....</i>	<i>68</i>
<i>Figure n° 46 : code authvalidate</i>	<i>69</i>
<i>Figure n° 47 : erreur via php</i>	<i>70</i>
<i>Figure n° 48 : page accueil</i>	<i>70</i>
<i>Figure n° 49 : fichier htaccess.....</i>	<i>71</i>
<i>Figure n° 50 : code route.....</i>	<i>71</i>
<i>Figure n° 51 : code route suite</i>	<i>72</i>
<i>Figure n° 52 : formulaire recherche.....</i>	<i>73</i>
<i>Figure n° 53 : code ajax jquery.....</i>	<i>74</i>
<i>Figure n° 54 : code ajax jquery suite</i>	<i>75</i>
<i>Figure n° 55 : affichage bloc html.....</i>	<i>76</i>

<i>Figure n° 56 : page stock initiale</i>	<i>77</i>
<i>Figure n° 57 : page stock avec une recherche</i>	<i>77</i>
<i>Figure n° 58 : page recherche non existant</i>	<i>78</i>