



Institut Africain d'Informatique
Etablissement Inter-Etats d'Enseignement Supérieur
BP : 2263 Libreville (Gabon) Tel. (00241) 07 70 55 00/07 70 56 00
Site web: www.iaisiege.com E-mail: contact@iaisiege.com

Immeuble Atlantis III BP : 8644 Libreville
(Gabon)
Tel: (241) 01 70 44 41 Fax (241) 01 70 45 04

MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme

D'Ingénieur en Informatique

Thème :

**CONCEPTION ET DÉVELOPPEMENT D'UNE PLATEFORME
DE COLLECTE ET DE DISTRIBUTION DE
DOCUMENTS NUMÉRIQUES AVEC GESTION
DE DROITS D'AUTEUR**

Présenté et soutenu par :
KENMEGNE FOPOUSSI Stéphanie

Superviseur

Pr Souleymane KOUSSOUBE
Maître de Conférences

Encadreur

Mr FEZE Séraphin
Ingénieur informaticien à Métrika

Année académique 2018-2019

DÉDICACE

*À mes parents,
FOPOUSSI & MEGUE KAMDEM.
Vous n'avez jamais cessé de veiller sur moi ni de vous investir
pour mon succès et mon bonheur.
Je vous dédie ce travail.*

ÉPIGRAPHE

« On dit que l'ignorant est celui qui est sans savoir, celui qui n'est pas instruit de certaines choses. Je crois que l'ignorant n'est pas celui qui n'a rien appris, mais celui qui a appris des choses erronées. »

Jean-Paul Pougala

REMERCIEMENTS

Je ne saurais débiter la rédaction de ce mémoire sans remercier mes nobles ancêtres de La'a Djo(Bandjoun) pour leur bienveillance à mon égard durant ces années d'études. En digne fille Tadjom que je suis, je les remercie davantage de m'avoir laissé cette terre sacrée et pleine de promesses et d'opportunités.

Mes remerciements les plus sincères vont à la République du Cameroun pour l'investissement multiforme consenti à faire de moi et de tous ses enfants en général des cadres de haut niveau. Je voudrais par la même occasion remercier l'État du Gabon pour l'accueil, l'assistance et les facilités qui nous ont été accordées pour permettre le bon déroulement de nos études en territoire gabonais.

Je poursuis en remerciant l'Institut Africain d'Informatique, son personnel dirigeant, administratif, enseignant pour l'environnement fourni pour mener à bien ces années d'études.

Je voudrais également remercier Métrika-IDB pour le stage offert, l'accueil et l'opportunité de mettre en valeur mes compétences d'ingénieur dans un projet ambitieux. Par la même occasion je voudrais remercier mon superviseur académique Pr Souleymane Koussoube pour les conseils et orientations prodigués.

Je voudrais aussi remercier tous mes collègues de promotion et toute la famille ASSECI pour la chaleur humaine, les échanges culturels, la collaboration qui ont été les nôtres au cours de cette formation à l'Institut Africain d'Informatique.

Un merci spécial à toute ma famille restée au Cameroun, en particulier à mon époux KENGNE MABOU pour ses conseils et son soutien sans faille durant ces années d'études.

À tous ceux qui, de près ou de loin, ont participé au succès de ce séjour d'étude à l'Institut Africain d'Informatique, je dis un grand **MERCI !!!**

AVANT-PROPOS

L'Institut Africain d'Informatique (I.A.I) est une École Inter États d'Enseignement Supérieur créé en 1971, regroupant actuellement onze (11) États Africains et dont le siège est au Gabon (Libreville). Les États membres actuellement impliqués sont le Bénin, le Burkina Faso, le Cameroun, la République Centrafricaine, le Congo, la Côte d'Ivoire, le Gabon, le Niger, le Sénégal, le Tchad et le Togo. Il intègre dans le cursus de formation des ingénieurs de conception en informatique, en fin de cycle, un stage de formation pratique d'une durée de cinq (5) mois en entreprise ou dans un laboratoire de recherche. Ce stage de fin d'étude vise à mettre les élèves-ingénieurs dans un environnement de conception d'applications informatiques ou dans un centre de Recherche et de Développement(laboratoire) afin de leur permettre une intégration en milieu professionnel et/ou scientifique.

Notre stage effectué au sein de METRIKA-IDB à Libreville au Gabon, s'inscrit donc dans ce cursus de formation des ingénieurs. En effet, ce stage pratique d'une durée de cinq(5) mois donne lieu à la rédaction d'un mémoire de fin d'études. Le présent document constitue donc l'aboutissement de trois (3) années de formation à l'Institut Africain d'Informatique de Libreville et cinq (5) mois de stage de Recherche et Développement effectué à METRIKA-IDB. Il tient lieu de mémoire de fin de formation d'ingénieur de conception en Informatique. Durant ce stage, notre étude a porté sur le thème :

« Conception et développement d'un système de collecte et de distribution de documents numériques avec gestion de droits d'auteur »

RÉSUMÉ

Les taux de pénétration d'internet, de la téléphonie mobile et du paiement électronique qui ont fortement accru ces dernières années au Gabon, ont favorisé l'avènement de la consommation d'informations sur internet, notamment sur smartphone et tablettes mobiles. Ainsi, le document papier est peu à peu substitué par le document numérique, le paiement en liquide par le paiement mobile. Cet état de choses occasionne la baisse de revenus des producteurs de contenu(éditeurs) au format physique.

Face à ces changements d'habitude de consommation, lesdits producteurs se doivent de s'arrimer à ce nouveau mode de diffusion d'informations. Lequel mode fait émerger la question des droits d'auteur. Les éditeurs étant divers, dispersés et exerçant dans des domaines variés, il convient de regrouper et stocker les contenus dans une base.

L'objectif de notre travail est de mettre sur pied une plateforme permettant aux éditeurs ou producteurs de contenus de distribuer leurs produits facilement et à moindre coût, et aux lecteurs d'accéder instantanément aux différents contenus dès leur publication. Tout ceci en limitant le piratage des œuvres des éditeurs. Notre solution a été implémenté en Java et comporte deux modules : Web et mobile(Android). Notre solution vise en priorité les documents numériques au format PDF.

L'application web permet aux éditeurs de publier du contenu et de visualiser leurs statistiques de vente. Elle permet également aux lecteurs de consulter le catalogue des documents regroupés par catégorie, d'acheter les documents par paiement électronique, de les lire et de consulter ses dernières transactions sur la plateforme. Pour cette application web, nous nous sommes servis de JSF comme framework MVC pour construire les applications web, PrimeFaces pour construire les interfaces utilisateurs, FlipBook 3D pour visualiser les PDFs, Apache Lucene comme bibliothèque de recherche et indexation, EclipseLink comme ORM.

L'application mobile quant à elle permet d'acheter des documents, de les télécharger et de les lire. Elle dispose d'une fonction anti-piratage (empêcher la copie privée des documents téléchargés) mis en œuvre aux moyens de technique de cryptographie symétrique fourni par les API Java notamment, l'algorithme AES. Nous avons en outre utilisé les bibliothèques Volley et PDFView respectivement pour les échanges avec le serveur d'application et la lecture des PDFs au sein de l'application mobile.

Pour la fonction de paiement électronique, disponible dans les applications web et mobile, nous avons intégré en priorité les API de paiement AIRTEL MONEY via la passerelle de paiement fourni par PVIT.

MOTS-CLÉS : Document numérique, paiement électronique, droits d'auteur, éditeur, moteur de recherche.

ABSTRACT

The penetration rates of Internet, mobile telephony and electronic payment, which have increased sharply in recent years in Gabon, have favored the advent of the consumption of information on Internet, especially on smartphones and mobile tablets. Thus, the paper document is gradually replaced by the digital document and cash payment by mobile payment. This state of affairs causes the drop in income of publishers who print their content on paper.

Faced with these changes in consumption habits, the said producers have to adapt to this new mode of information dissemination. Which mode brings up the question of copyright. The publishers being diverse, dispersed and excelling in various fields, it is advisable to group and store the contents in a base.

Our objective was to set up a platform allowing publishers or producers of contents to distribute their products easily and at low cost, and readers to instantly access the different content as soon as they are published. All this by protecting publishers's works from piracy. Our solution has been implemented in Java and has two modules: Web and mobile (Android). Our solution primarily targets digital documents in PDF format.

The web application allows publishers to publish content and view their sales statistics. It also allows readers to consult the catalog of documents grouped by category, to purchase documents by electronic payment, to read them and to consult their latest transactions on the platform. For this web application, we used JSF as MVC framework to build web applications, PrimeFaces to build user interfaces, FlipBook 3D to view PDFs, Apache Lucene as search and indexing library, EclipseLink as ORM.

The mobile application allows you to buy documents, download and read them. It has an anti-piracy function (prevent private copying of downloaded documents) implemented using symmetric cryptography techniques provided by Java APIs in particular, the AES algorithm. We also used the Volley and PDFView libraries respectively for the communications with the application server and the reading of PDFs within the mobile application.

For the electronic payment, available in web and mobile applications, we have integrated the AIRTEL MONEY payment APIs via the payment gateway provided by PVIT.

KEYWORDS: Digital document, electronic payment, copyright, publisher, search engine.

LISTE DES FIGURES

Figure 1 : Organigramme de METRIKA.....	4
Figure 2 : Échantillon de tablette sumérienne.....	7
Figure 3 : Aperçu d'un QR code.....	14
Figure 4 : Image paiement par NFC.....	16
Figure 5: Phases et activités du Processus Unifié.....	28
Figure 6 : Processus de développement en Y.....	29
Figure 7: Diagramme de contexte statique.....	33
Figure 8: Diagramme de contexte dynamique.....	34
Figure 9: Diagramme des cas d'utilisation.....	44
Figure 10: Architecture MVC.....	53
Figure 11: Architecture 2-tiers.....	54
Figure 12: Architecture 3-tiers.....	55
Figure 13: Architecture logique de notre solution.....	56
Figure 14: Diagramme de classes préliminaire.....	57
Figure 15: Diagramme de navigation globale.....	58
Figure 16: Diagramme de composants du système.....	58
Figure 17: Diagramme des cas d'utilisation du projet «Gestion des comptes client».....	60
Figure 18: Diagramme classe du projet «Gestion des comptes client».....	61
Figure 19: DSS du cas d'utilisation «Liste les comptes client».....	62
Figure 20: DSS du cas d'utilisation «Créer compte client».....	62
Figure 21: DSS du cas d'utilisation «Modifier un compte client».....	63
Figure 22: DSS du cas d'utilisation «Désactiver un compte client».....	63
Figure 23: DCP du cas d'utilisation «Créer compte client».....	64
Figure 24: DCP du cas d'utilisation «Modifier compte client».....	65
Figure 25: DCP du cas d'utilisation «Désactiver compte client».....	65
Figure 26: DCP du cas d'utilisation «Lister les comptes clients».....	65
Figure 27: DSOP du cas d'utilisation «Créer compte client».....	66
Figure 28: DSOP du cas d'utilisation «Lister les comptes clients».....	66
Figure 29: DCD du projet «gestion des comptes client».....	67
Figure 30: DSOD du cas d'utilisation «Créer compte client».....	67
Figure 31: DSOD du cas d'utilisation «Lister les comptes clients».....	68
Figure 32: Implémentation de la classe «CreationUserBean».....	68
Figure 33: Implémentation de la classe «UtilisateurDao».....	69
Figure 34: Diagramme des cas d'utilisation du projet «Acquisition des documents».....	71
Figure 35: Diagramme de classes du projet «Acquisition de documents».....	72
Figure 36: DSS du cas d'utilisation «Rechercher un document».....	72
Figure 37: DSS du cas d'utilisation «Acheter un document».....	73
Figure 38: DSS du cas d'utilisation «Ouvrir un document».....	73
Figure 39: DCP du cas d'utilisation «Ouvrir un document».....	74
Figure 40: DCP du cas d'utilisation «Acheter un document».....	74
Figure 41: DCP du cas d'utilisation «Rechercher un document».....	75

Figure 42: DSOP du cas d'utilisation «Ouvrir un document».....	75
Figure 43: DSOP du cas d'utilisation «Rechercher un document».....	76
Figure 44: DSOP du cas d'utilisation «Acheter un document».....	76
Figure 45: Diagramme de classes détaillées du projet «Acquisition des documents»	77
Figure 46: DSOD du cas d'utilisation «Ouvrir un document».....	77
Figure 47: DSOD du cas d'utilisation «Rechercher un document».....	78
Figure 48: Implémentation de la classe «PanierBean».....	78
Figure 49: Implémentation de la classe «Transaction».....	79
Figure 50: Diagramme de déploiement de la solution.....	81
Figure 51: Fenêtre d'accueil application web.....	82
Figure 52: Fenêtre de lecture de PDF au sein de l'application web.....	82
Figure 53: Fenêtre de statistiques de ventes - diagramme circulaire.....	83
Figure 54: Fenêtre de statistiques de ventes - diagramme à bandes.....	83
Figure 55: Diagramme de Gantt prévisionnel.....	85
Figure 56: Diagramme de Gantt réel.....	86

LISTE DES TABLEAUX

Tableau 1: Formats de documents numériques.....	9
Tableau 2: Récapitulatif des messages émis et reçus par le système.....	33
Tableau 3: Cas d'utilisation et acteurs associés.....	36
Tableau 4: Structuration des cas d'utilisation en package.....	45
Tableau 5: Découpage des cas d'utilisation en itérations.....	47
Tableau 6: Récapitulatif des outils de développement.....	52
Tableau 7: Messages émis et reçu par le système dans le projet «Gestion des comptes client».....	60
Tableau 8: Cas d'utilisation du projet « Gestion des comptes client ».....	60
Tableau 9: Messages émis et reçu par le système dans le projet «Acquisition des documents».....	70
Tableau 10: Cas d'utilisation du projet «Acquisition des documents».....	71
Tableau 11: Estimation des coûts et charges du projet en tant que stagiaire.....	86
Tableau 12: Coût pour un particulier voulant se procurer notre solution.....	87

GLOSSAIRE

Abréviations	Signification
2TUP	2-Track Unified Process
AES	Advanced Encryption Standard
ANINF	Agence Nationale des Infrastructures Numériques et des Fréquences
API	Application Programming Interface
DRM	Digital Rights Management
EDI	Environnement de développement intégré
HTTP	HyperText Transfer Protocol
IHM	Interface homme-machine
JEE	Java Enterprise Edition
JSF	Java Server Faces
JSON	JavaScript Object Notation
MVC	Modèle Vue Contrôleur
ORM	Object Relational Mapping
PDF	Portable Document Format
REST	Representational State Transfer
SGBD	Systèmes de gestion de bases de données
SMS	Short Message Service
SSL	Secure Sockets Layer
SQL	Structured Query Language
UML	Unified Modeling Language
UP	Unified Process
USSD	Unstructured Supplementary Service Data
VOD	Video on Demand
XML	Extensible Markup Language

TABLE DES MATIÈRES

DÉDICACE.....	i
ÉPIGRAPHE.....	ii
REMERCIEMENTS.....	iii
AVANT-PROPOS.....	iv
RÉSUMÉ.....	v
ABSTRACT.....	vi
LISTE DES FIGURES.....	vii
LISTE DES TABLEAUX.....	ix
TABLE DES MATIÈRES.....	xi
INTRODUCTION GÉNÉRALE.....	1
Partie I : LE SUJET ET SON CONTEXTE.....	2
Chapitre I: LA STRUCTURE D'ACCUEIL ET LE SUJET.....	3
Introduction.....	3
I.1 Structure d'accueil : Metrika-IDB.....	3
I.1.1 Présentation générale.....	3
I.1.2 Métiers de Metrika-IDB.....	3
I.1.3 Organigramme.....	4
I.2 Présentation du sujet.....	5
I.2.1 Contexte.....	5
I.2.2 Spécification du problème.....	5
I.2.3 Objectifs généraux.....	6
Chapitre II: CONCEPTS LIÉS A LA DISTRIBUTION DES DOCUMENTS NUMÉRIQUES.....	7
Introduction.....	7
II.1 Historique.....	7
II.2 Principales notions.....	8
II.2.1 Qu'est ce qu'un document numérique?.....	8
II.2.1.1 Définition.....	8
II.2.1.2 Supports de diffusion des documents numériques.....	8
II.2.1.3 Formats de documents numériques.....	8
II.2.2 Distribution numérique.....	9
II.2.3 L'indexation.....	10
II.2.3.1 Procédure d'indexation de documents.....	10
II.2.3.2 Types d'indexation.....	10
II.2.4 La recherche.....	11
II.2.4.1 Recherche non-informée.....	11
II.2.4.2 Recherche informée ou éclairée.....	12
II.2.4.3 Recherche contradictoire ou avec adversaire.....	12
II.2.4.4 Recherche par interpolation.....	12
II.2.5 Gestion des droits numériques.....	12

II.2.6 Paiement électronique.....	13
Conclusion.....	17
Chapitre III: AUDIT DE L'EXISTANT ET CAHIER DE CHARGES.....	18
III.1 Solutions existantes.....	18
III.1.1 Kiosque numérique de Sogapresse.....	18
III.1.2 Autres kiosques numériques.....	18
III.2 Cahier des charges.....	19
III.2.1 Collecte des documents.....	19
III.2.2 Acquisition de documents.....	19
III.2.3 Recherche de documents.....	19
III.2.4 Gestion des droits numériques.....	20
III.2.5 Élaboration des statistiques.....	20
Partie II : RÉALISATION DU SYSTÈME.....	21
Chapitre IV: MÉTHODOLOGIE DE DÉVELOPPEMENT.....	22
IV.1 Tour d'horizon des méthodes de développement.....	22
IV.1.1 Les méthodes cartésiennes ou fonctionnelles.....	22
IV.1.2 Les méthodes systémiques.....	23
IV.1.3 Les méthodes orientées-objet.....	24
IV.1.4 Les méthodes agiles.....	25
IV.1.5 Les méthodes formelles.....	25
IV.2 Choix d'une méthode.....	26
IV.2.1 Le langage UML.....	26
IV.2.2 2TUP.....	27
Chapitre V: MISE EN ŒUVRE DE LA SOLUTION.....	31
V.1 Projet 1 : Étude globale du système.....	31
V.1.1 Étude préalable.....	31
V.1.1.1 Projet à réaliser.....	31
V.1.1.2 Recueil des besoins fonctionnels.....	31
V.1.1.3 Recueil des besoins opérationnels.....	31
V.1.1.4 Identification des acteurs.....	32
V.1.1.5 Identification des messages.....	32
V.1.1.6 Modélisation du contexte.....	33
V.1.1.6.1 Diagramme de contexte statique.....	33
V.1.1.6.2 Diagramme de contexte dynamique.....	34
V.1.1.7 Définition des choix techniques.....	34
V.1.1.7.1 Architecture matérielle.....	34
V.1.1.7.2 Langage de programmation.....	35
V.1.1.7.3 Architecture logicielle.....	35
V.1.1.7.4 Outils de travail.....	35
V.1.2 Capture des besoins fonctionnels.....	35
V.1.2.1 Identification des cas d'utilisation.....	35
V.1.2.2 Description textuelle des cas d'utilisation.....	36
V.1.2.3 Diagramme des cas d'utilisation.....	43

V.1.2.4 Structuration des cas d'utilisation en package.....	44
V.1.2.5 Planification du travail en incréments et en itérations.....	45
V.1.3 Capture des besoins techniques.....	47
V.1.4 Conception générique.....	53
V.1.4.1 Le MVC.....	53
V.1.4.2 Architecture client-serveur.....	54
V.1.4.2.1 Types d'architecture client-serveur.....	54
V.1.4.2.2 Types de client.....	55
V.1.4.3 Présentation de notre architecture.....	55
V.1.5 Analyse.....	57
V.1.5.1 Classes candidates.....	57
V.1.5.2 Diagramme d'activités de navigation.....	57
V.1.5.3 Diagramme de composants.....	58
V.1.6 Autres activités.....	59
V.2 Projet 2 : Gestion des comptes client.....	59
V.2.1 Étude préalable.....	59
V.2.1.1 Projet à réaliser.....	59
V.2.1.2 Recueil des besoins fonctionnels.....	59
V.2.1.3 Recueil des besoins opérationnels.....	59
V.2.1.4 Identification des acteurs.....	59
V.2.1.5 Identification des messages.....	60
V.2.2 Capture des besoins fonctionnels.....	60
V.2.2.1 Identification des cas d'utilisation.....	60
V.2.2.2 Diagramme des cas d'utilisation.....	60
V.2.3 Capture des besoins techniques.....	61
V.2.4 Analyse.....	61
V.2.4.1 Diagramme de classes d'analyse.....	61
V.2.4.2 Diagrammes de séquence système (DSS).....	61
V.2.5 Conception préliminaire.....	64
V.2.5.1 Diagramme de classes participantes.....	64
V.2.5.2 Diagrammes de séquence-objet préliminaire (DSOP).....	66
V.2.6 Conception détaillée.....	67
V.2.6.1 Diagramme de classes détaillées (DCD).....	67
V.2.6.2 Diagramme de séquences-objets détaillés (DSOD).....	67
V.2.7 Codage.....	68
V.3 Projet 3 : Acquisition des documents.....	69
V.3.1 Étude préalable.....	69
V.3.1.1 Projet à réaliser.....	69
V.3.1.2 Recueil des besoins fonctionnels.....	69
V.3.1.3 Recueil des besoins opérationnels.....	69
V.3.1.4 Identification des acteurs.....	70
V.3.1.5 Identification des messages.....	70
V.3.2 Capture des besoins fonctionnels.....	70

V.3.2.1 Identification des cas d'utilisation.....	70
V.3.2.2 Diagramme des cas d'utilisation.....	71
V.3.3 Capture des besoins techniques.....	71
V.3.4 Analyse.....	72
V.3.4.1 Diagramme de classes d'analyse.....	72
V.3.4.2 Diagramme de séquence système (DSS).....	72
V.3.5 Conception préliminaire.....	74
V.3.5.1 Diagramme de classes participantes (DCP).....	74
V.3.5.2 Diagrammes de séquence-objet préliminaire(DSOP).....	75
V.3.6 Conception détaillée.....	77
V.3.6.1 Diagramme de classes détaillées (DCD).....	77
V.3.6.2 Diagrammes de séquence-objet détaillés (DSOD).....	77
V.3.7 Codage.....	78
Partie III : FINALISATION DE LA SOLUTION.....	80
Chapitre VI: DÉPLOIEMENT ET RÉSULTATS.....	81
VI.1 Diagramme de déploiement.....	81
VI.2 Résultats obtenus.....	82
Chapitre VII: CONDUITE DE PROJET.....	85
VII.1 Diagrammes de planification du projet.....	85
VII.2 Évaluation du coût du projet.....	86
VII.2.1 Évaluation réelle.....	86
VII.2.2 Évaluation nominale.....	87
VII.3 Bilan et perspectives.....	87
VII.3.1 Bilan.....	87
VII.3.1.1 Apports.....	87
VII.3.1.2 Difficultés.....	88
VII.3.2 Perspectives.....	88



INTRODUCTION GÉNÉRALE

Alors qu'un nouveau média, Internet et plus généralement les supports numériques voyaient le jour, les éditeurs n'avaient aucune visibilité sur les bouleversements d'usages qu'allaient apporter une telle révolution. Internet a révolutionné la manière de «consommer» l'information. En entrant dans l'ère du numérique, le document a connu un grand nombre de mutations qui se sont traduites au cours de ces trente dernières années par l'apparition d'enjeux de société et l'explosion de nouveaux métiers sur le marché de l'emploi. La vente de fichiers numériques se démocratise dans tous les secteurs. Les indépendants et les entreprises sont de plus en plus nombreux à vendre des documents dématérialisés, des photos ou images, des vidéos ou de la musique. Le numérique s'impose de plus en plus, grâce à la rentabilité de ce format, par rapport au papier pour les éditeurs. Le document électronique permet ainsi aux auteurs de publier leurs ouvrages sans passer par l'intermédiaire d'une maison d'édition(cas des livres), ce qui peut être très pratique. Ainsi, les livres, journaux et magazines sont aujourd'hui remplacés par des documents informatiques disponible en différents formats.

Fort de tout cela, il est donc primordial de mettre en place un système permettant aux producteurs de contenus de publier leurs documents et de faciliter l'acquisition desdits documents par les lecteurs au moyen du paiement électronique. Le défi ici sera celui de gérer les droits d'auteur des éditeurs tout en rendant possible l'accès aux documents acquis sur web et sur mobile.

Le présent mémoire de fin d'études s'attelle donc à proposer une solution à cela et s'articule en trois(3) parties :

La première partie est dédiée à la présentation de notre structure d'accueil et des concepts clés nécessaires à la compréhension du sujet.

Dans la seconde partie, nous ferons d'abord une brève étude de l'existant, puis nous procéderons à la présentation des travaux de conception et de réalisation pour la mise sur pied de notre plateforme web et mobile de collecte et de distribution de documents numériques.

Enfin la dernière partie est consacrée à la conduite, le bilan et les perspectives de notre projet de fin d'études.



Partie I : LE SUJET ET SON CONTEXTE

APOSTILLE

Dans cette partie nous nous proposons de faire une présentation succincte de la structure qui nous accueille dans le cadre de ce stage académique. Nous présenterons d'abord les services et missions de l'entreprise, puis les concepts clés soulevés par notre sujet et nous terminerons par un audit de ce qui existe en la matière après quoi nous élaborerons notre cahier de charges.

CHAPITRES

CHAPITRE I : LA STRUCTURE D'ACCUEIL ET LE SUJET

CHAPITRE II : CONCEPTS LIÉS A LA DISTRIBUTION DES DOCUMENTS NUMÉRIQUES

CHAPITRE III: AUDIT DE L'EXISTANT ET CAHIER DE CHARGES



Chapitre I: LA STRUCTURE D'ACCUEIL ET LE SUJET

Introduction

Dans ce chapitre, nous présentons la structure d'accueil et le sujet sur lequel nous avons travaillé.

I.1 Structure d'accueil : Metrika-IDB

I.1.1 Présentation générale

La conception et le développement de nouvelles générations de systèmes d'information d'aide à la décision (systèmes d'information décisionnels) sont à l'origine de la création de Métrika-idb en 1995. La société a son siège social à l'immeuble ATLANTIS II situé à ACAE (Libreville GABON).

Métrika possède une filiale en France (Paris) et un important réseau de sociétés partenaires à l'échelle internationale. Pour des tâches de développement et de recherche, Métrika-idb dispose d'une équipe pluridisciplinaire, constituée, d'une part, de praticiens et d'universitaires hautement qualifiés (ingénieurs informaticiens, mathématiciens appliqués, statisticiens économistes). Et d'autre part, de nombreux thématiciens issus des secteurs d'activité tels que : l'éducation, la santé, la foresterie, la macroéconomie, la finance, la banque et la téléphonie (Filaire et GSM).

I.1.2 Métiers de Metrika-IDB

Métrika-IDB est une société spécialisée dans la conception, le développement et le déploiement en architecture client/serveur multi tiers, d'outils logiciels métiers et d'aide à la décision. Ses activités sont les suivantes :

- **Editeur d'outils logiciels métiers d'aide à la décision** : Plus précisément, Métrika est spécialisé dans la conception, le développement et le déploiement (physique et logique) en architecture client-serveur, de nouvelles générations de systèmes d'information décisionnels, notamment, ceux reposant sur la collecte (en temps réel ou en temps différé), l'organisation (en un guichet unique), l'historisation, l'agrégation, la mise en état non volatile, et le contrôle de cohérence et d'intégrité des données multi-sources et multi-temporelles.
- **Encadrement de la recherche et des tâches de développement**, d'outils logiciels métiers, ainsi que celles de conception, d'implémentation et de

déploiement (physique et logique) de systèmes d'information décisionnels, est assuré par des professeurs d'Université, par des praticiens, et par des professionnels expérimentés, ayant participé à la mise au point de méthodes (formelles et heuristiques d'aide à la décision) innovantes, dans des secteurs tels que l'éducation, la santé, la foresterie, la macroéconomie, la finance, la banque, la téléphonie (Filaire et GSM). Métrika assure depuis sa création, des réponses objectives et innovantes, basées sur des analyses précises des problèmes décisionnels dans de nombreux domaines et secteurs d'activité.

- **Intégration de solutions informatiques et édition de logiciels :** Cette activité a pour objectif de développer des applications, de valider la compatibilité entre les équipements et logiciels choisis, de vérifier des capacités des infrastructures déjà en place et de l'impact futur sur l'utilisation des ressources et de communiquer les changements aux utilisateurs afin de leur permettre de s'approprier les nouvelles capacités et fonctionnalités mises en place.
- **Participation aux tests de nouveauté :** Cette activité concrétise les relations privilégiées de Métrika avec les plus grands constructeurs et importateurs (notamment AST, MAP/INFO, CANON, MITSUBISHI, OCE GRAPHIC, CISA, STAT-GRAPH). Métrika est un centre de compétence ARC/INFO, et jouit par ailleurs du label OVS (Oracle Value Service), lequel leur donne l'opportunité et le privilège de participer aux tests nouveautés des produits ORACLE.

I.1.3 Organigramme

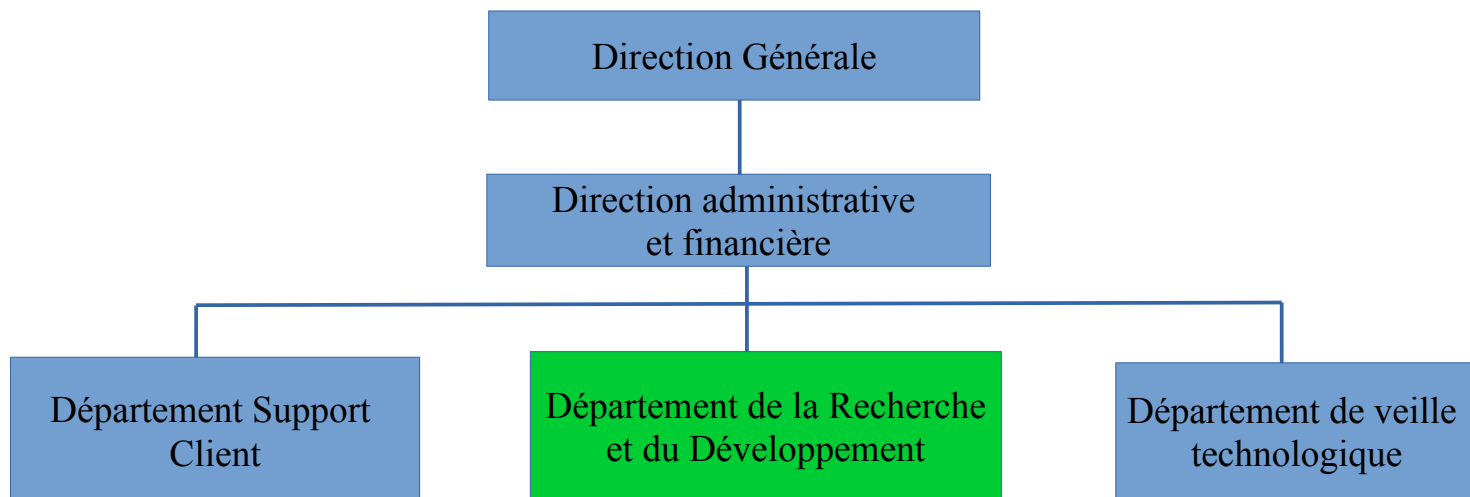


Figure 1 : Organigramme de METRIKA



C'est au sein du département de la recherche et du développement que nous avons effectué notre stage.

I.2 Présentation du sujet

La distribution, plus précisément la vente des documents immatériels est une filière en pleine expansion. La collecte ou l'acquisition des documents auprès des éditeurs va permettre de regrouper, d'organiser et mettre en ligne les documents afin d'en faciliter l'accès. Les lecteurs n'ont plus à se déplacer vers les kiosques, librairies ou bibliothèques pour se procurer un journal, un magazine ou un livre.

I.2.1 Contexte

Selon les statistiques du marché de l'internet livrés par l'ANINF(Agence Nationale de Infrastructures numériques et des Fréquences), au 31 janvier 2019, le Gabon compte officiellement un million d'internautes pour un taux de pénétration estimé à 48%. L'internet mobile est le plus prisé par ce million d'internautes. Il a représenté près de 99,11% du marché. Au niveau de la téléphonie mobile, le Gabon comptabilise 3 millions d'abonnés pour un taux de pénétration de 144%. 50% des utilisateurs du mobile l'utilise également en matière de transaction financière notamment à travers la technologie Mobile Money.

Ces taux de pénétration d'internet, de la téléphonie mobile et du paiement électronique qui ont fortement accru ces dernières années au Gabon, ont favorisé l'avènement de la consommation d'informations sur internet, notamment sur smartphone et tablettes mobiles. Face à ces booms enregistrés dans le secteur de la technologie, de l'information et de la communication, notre entreprise qui est un intégrateur de solutions informatiques et un éditeur de logiciels, a identifié une opportunité commerciale : celle de développer une solution permettant aux éditeurs de contenu de distribuer leurs produits rapidement et à moindre coût.

I.2.2 Spécification du problème

Les mutations profondes qu'a connu le monde de l'éditeur de contenu ces vingt dernières années avec le boom des moyens de traitement électronique de l'information ainsi que des systèmes de diffusion de ces informations notamment Internet, oblige les acteurs de ce métier à se réinventer, à s'adapter. En effet, ces dernières années ont vu les éditeurs être confrontés à une baisse de revenus dû à un changement des habitudes de consommation des usagers, avec une baisse de fréquentation des kiosques physiques. De plus, sont nés de nombreux média et sites en ligne qui très souvent propose un accès gratuit à du contenu pas toujours de bonne qualité, financé par la publicité. Aussi, ayant désormais accès à un vaste choix de contenu gratuit, de nombreux usagers se plaignent du coût d'accès élevé au contenu par les moyens physiques classiques. Aussi la stratégie de distribution de contenu par



multiplication de boutiques physiques n'est pas toujours rentable comme l'atteste les difficultés(banqueroute) rencontrées très récemment par l'américain Blockbuster (fournisseur de services de location de films et de jeux vidéo à domicile via un magasin de location de vidéos ; celui-ci était restée dans la logique de kiosque physique alors que des innovations tels que le streaming et la vidéo à la demande voyaient le jour).

Face à toutes ces difficultés apparentes, un ensemble de questions émergent :

- Comment fournir une solution qui permettent aux éditeurs de distribuer à moindre coût les contenus qu'ils éditent et toucher le plus grand nombre de clients et notamment cette grande masse de clients qui consomment désormais par voie numérique?
- Comment fournir une solution permettant aux usagers de rechercher, trouver et consulter rapidement les contenus qui les intéressent et de les acquérir aisément ?
- Comment fournir une solution qui permette aux éditeurs de diffuser largement leur contenu, gagner de l'argent tout en garantissant la protection des contenus?

I.2.3 Objectifs généraux

L'objectif de ce présent stage est de mettre en œuvre une plateforme informatique de collecte et distribution de documents remplissant les attentes suivantes :

- Permettre aux éditeurs de publier et distribuer leurs contenus en ligne
- Permettre aux utilisateurs de rechercher, consulter et acheter aisément(par paiement électronique) les contenus qui les intéressent
- Protéger les droits d'auteur et propriétés intellectuelles des éditeurs en restreignant ou en empêchant la copie privée des documents(transfert vers un appareil externe).
- Fournir une offre qui exploite l'immense opportunité que représente le mobile(mobile-first), dans un contexte où le mobile tend à devenir la première interface d'accès à Internet selon les dernières études.

Conclusion

Ce chapitre nous a permis de présenter la structure où nous avons effectué notre stage et le travail à réaliser. Dans celui qui suit, nous aborderons les concepts indispensables à la compréhension de notre sujet d'étude.



Chapitre II: CONCEPTS LIÉS A LA DISTRIBUTION DES DOCUMENTS NUMÉRIQUES

Introduction

Notre travail requiert des connaissances sur la gestion des documents numériques, la distribution numérique, le paiement mobile et la gestion des droits numériques. Dans ce chapitre nous passerons en revue ces principaux concepts.

II.1 Historique

Deux sources au document numérique peuvent être trouvées dans l'histoire. D'une part, la présentation du document numérique à l'écran évoque souvent chez certains chercheurs une association aux tablettes sumériennes en argile (3400-3200 av. J.C.). La manière de faire défiler le texte est semblable à la lecture d'un manuscrit enroulé. D'autre part, la partie invisible du document, c'est-à-dire le codage de l'information, renvoie à l'étymologie du mot « numérique ». Apparu en 1616, il provient du latin *numerus* « qui a rapport aux nombres, qui appartient aux nombres ». Le document numérique commence à émerger avec le traitement de texte et, plus précisément, dans le domaine de la bureautique. Dans les années 1970-1980, l'arrivée de la micro-informatique facilite dans les bureaux le traitement de texte. Le principe d'affichage du document à l'écran ainsi que son obtention telle que vue directement sur l'ordinateur lors de l'impression est nommé *wysiwyg* (What you see is what you get). Dès les années 1980, le document structuré voit le jour. Il se dote de nouvelles caractéristiques : l'interactivité, l'insertion de nouveaux modes de communication dans les documents (images, sons, etc.). Actuellement, les documents en réseau se produisent avec un langage dynamique et sont reliés à une base de données dont le contenu peut varier.



Figure 2 : Échantillon de tablette sumérienne



II.2 Principales notions

II.2.1 Qu'est ce qu'un document numérique?

II.2.1.1 Définition

Un document numérique est une forme de représentation de l'information consultable à l'écran d'un appareil électronique.

II.2.1.2 Supports de diffusion des documents numériques

L'enregistrement digital de l'information facilite le développement de la numérisation et la multiplication des documents numériques. Les supports de haute densité capables d'enregistrer des données binaires exigent de plus en plus de perfectionnement.

➤ Les disques optiques

Un disque optique est un disque plat servant de média amovible. Ils sont utilisés comme mémoires de masse. Les plus connus sont : les CD, les DVD et les Blu-ray.

➤ Le livre électronique

Aussi connu sous les noms de livre numérique, de livrel et E-book, est un livre édité et diffusé en version numérique, disponible sous la forme de fichiers, qui peuvent être téléchargés et stockés pour être lus sur un écran(ordinateur personnel, téléphone portable, liseuse, tablette tactile), sur une plage braille, un dispositif de lecture de livres audio, ou un navigateur.

➤ Wikis et blog

Les outils d'écriture collective appelés wikis, dont les plus connus sont l'encyclopédie ouverte Wikipédia et les carnets de notes collectifs ou personnels nommés blogs, illustrent le principe de la publication libre et accessible des documents numériques au plus grand nombre.

II.2.1.3 Formats de documents numériques

Tous les fichiers créés ou traités sur un ordinateur sont identifiés avec un nom, un format ou type de fichier(extension), une taille et des dates de création et modification. Le choix du format de fichier est nécessaire en fonction de son utilisation, sa qualité et sa taille souhaitée. Ci-dessous quelques formats existants.



Catégorie	Format - Extension
<i>Images</i>	<i>PNG, TIFF, JPEG, GIF, BMP</i>
<i>Diaporama</i>	<i>PPS, ODP</i>
<i>Son</i>	<i>OGG, FLAC, MP3, WAV, WMA, AAC</i>
<i>Vidéo</i>	<i>MPEG, OGM(DVD, DivX, XviD), AVI, FLV</i>
<i>Page</i>	<i>PDF, PostScript, HTML, XHTML</i>
<i>Traitement de texte</i>	<i>ODT, TXT, DOC, DOCX, RTF</i>
<i>Tableur</i>	<i>ODS, XLS, XLSX, DBF</i>
<i>Exécutable</i>	<i>BIN, ELF, EXE</i>
<i>Archive(fichier compressé)</i>	<i>TAR, GZIP, ZIP, LZW, JAR RAR, 7Z</i>

Tableau 1: Formats de documents numériques

II.2.2 Distribution numérique

La distribution numérique ou distribution en ligne, décrit la fourniture de contenus multimédia, tels que l'image, le son, la vidéo, les logiciels, le texte ou encore l'hypertexte, sans l'utilisation d'un support physique conventionnel. La distribution numérique contourne les méthodes classiques de distribution physique, que sont le papier ou le disque optique. Avec l'avancement des capacités de bande passante, la distribution numérique a pris de l'importance dans les années 2000. Le contenu distribué en ligne peut être lu en ligne (en streaming pour les vidéos) ou téléchargé. La distribution numérique apporte des avantages au rang desquels :

➤ La vente directe

Les produits distribués numériquement requièrent moins d'intermédiaires, et par conséquent les éditeurs obtiennent plus de bénéfices. Cela limite aussi l'impact écologique du produit, puisque la production de médias physiques est éliminée dans sa quasi-totalité.

➤ Une disponibilité globale

La distribution numérique par inhérence possède l'un des plus grands forts points d'Internet : sa disponibilité. Celle-ci est telle que n'importe quelle personne dans le monde avec une connexion à Internet peut acheter et obtenir facilement une



copie numérique. De cette façon, le contenu est théoriquement illimité, les prix d'envoi sont éliminés ainsi que les temps d'attente.

➤ Une production facile

À la différence de la distribution physique, dans lesquelles une condition requise minimale de produits physiques existe, le produit numérique n'a pas besoin d'être produit en masse pour approvisionner les vendeurs. Avec une seule copie originale, on peut doubler rapidement des copies pour satisfaire la demande. Cela peut réduire en grande partie les prix de maintenance, puisqu'il n'est pas nécessaire d'avoir un dépôt ou un magasin pour préserver le produit (humidité, sécurité, etc.).

II.2.3 L'indexation

Le concept d'indexation est au cœur de tous les moteurs de recherche. L'indexation peut être définie comme la transformation des données d'origine en une référence croisée très efficace afin de faciliter la recherche rapide.

Elle correspond à la représentation d'un texte ou d'un document par un indice ou un mot clé, en vue d'en faciliter le repérage et la consultation. L'indexation est le processus selon lequel le contenu d'un document est analysé pour être ensuite reformulé dans une forme permettant d'accéder au contenu et de le manipuler. Le terme d'indexation qualifie à la fois le processus et son résultat. Une indexation est par conséquent la description d'un document effectuée dans la perspective d'une utilisation et exploitation données.

II.2.3.1 Procédure d'indexation de documents

Le résultat concret de l'étape d'indexation est premièrement la production d'index qui vont associer à chaque document (ou partie d'un document), d'un corpus, des termes censés représenter les sujets dont ils traitent. En second lieu, l'indexation fixe des stratégies de recherche, qui vont utiliser les termes indexés pour repérer le document "pertinent" parmi la masse documentaire. Indexer un document c'est donc élaborer un jeu d'éléments censé le représenter au mieux, puis définir les moyens d'accès à ce document à l'aide de ces éléments représentatifs.

II.2.3.2 Types d'indexation

On distingue deux types d'indexation : Indexation manuelle et indexation automatique.

■ L'indexation manuelle

Elle se scinde en 3 sous-types:

▼ L'indexation alphabétique



L'indexation alphabétique d'un texte consiste à repérer dans celui-ci certains mots ou expressions particulièrement significatifs (appelés termes) dans un contexte donné, afin de créer un index terminologique. Il s'agit par exemple d'index de sujets, de lieux, de personnes en fin d'ouvrage.

✓ L'indexation systématique

Elle correspond à l'attribution d'un indice à un document de manière à le situer dans un ensemble de la connaissance. Il dépend d'une classification préalable, encyclopédique (classification décimale universelle, classification Dewey) ou non.

✓ L'indexation matière

Encore appelée indexation analytique, c'est le processus qui consiste à indiquer dans un catalogue, un instrument de recherche ou une base de données bibliographiques le sujet d'un document. Il s'appuie le plus généralement sur un répertoire de mots normalisés. L'indexation Rameau est un exemple d'indexation analytique.

■ L'indexation automatique

L'indexation automatique de documents utilise des méthodes algorithmiques. Il peut s'agir de techniques d'analyse statistique à base de comptages d'occurrences, ou de techniques linguistiques d'analyse morpho-syntaxique du texte : extraction de mots simples (unitermes) ou composés (synapsies). La multiplicité des types de documents (textuels, audiovisuels, Web) donne lieu à des approches très différentes, notamment en termes de représentation des données. Elles reposent néanmoins sur un socle de théories communes, telles que l'extraction de caractéristiques, le partitionnement de données, la quantification, et plus généralement la recherche d'information. Elle est notamment utilisée par les moteurs de recherche.

II.2.4 La recherche

En informatique, la recherche peut être définie comme un algorithme prenant un problème en entrée et renvoie une solution au problème, généralement après avoir évalué un certain nombre de solutions possibles. L'ensemble de toutes les solutions possibles à un problème s'appelle l'espace de recherche. La recherche peut être classifiée comme suit :

II.2.4.1 Recherche non-informée

Un algorithme de recherche non informée ne prend pas en compte la nature spécifique du problème. Les algorithmes de recherche non informés sont généralement implémentés, puis la même implémentation est utilisée pour un large éventail de problèmes à l'aide de l'abstraction. Le désavantage de ce type de



recherche est celui des espaces de recherche extrêmement grands, et ladite recherche prendra un quantité raisonnable de temps pour de petits exemples. Par conséquent, pour accélérer le processus, il faut utiliser une recherche éclairée. Des exemples de recherche non informée incluent la recherche dans une liste, dans les arbres et dans un graphe.

II.2.4.2 Recherche informée ou éclairée

La technique de recherche informée utilise les connaissances spécifiques au problème afin de donner une idée de la solution du problème. Et donc, elle fournit la direction concernant la solution, ce qui lui permet de trouver la solution plus rapidement. Les algorithmes A* et de recherche du meilleur d'abord (Best-First Search) sont des exemples d'algorithmes de recherche informée.

II.2.4.3 Recherche contradictoire ou avec adversaire

Plus connu sous son appellation anglaise « adversarial search », la recherche contradictoire est une recherche, où on examine le problème qui se pose lorsqu'on essaye de planifier à l'avance, alors que d'autres agents planifient contre nous. Sa caractéristique principale est de tenir compte de tout mouvement possible que l'adversaire peut prendre. Elle est essentiellement utilisée dans les jeux et l'intelligence artificielle.

II.2.4.4 Recherche par interpolation

Une recherche par interpolation tente de trouver l'élément en estimant la distance à laquelle il est susceptible de se trouver par rapport à la position actuelle. La recherche par interpolation est analogue à la recherche dans un dictionnaire.

II.2.5 Gestion des droits numériques

L'avènement d'internet dans les années 1990, a entraîné la dématérialisation des œuvres et a compliqué la juste rémunération des créateurs. Les éditeurs et distributeurs ont donc mis en place un écosystème favorisant le contrôle des créations. Le contrôle numérique des droits était né. Plus connu sous son acronyme anglais DRM (Digital Rights Management), la gestion des droits numériques a pour objectif de contrôler l'utilisation qui est faite des œuvres numériques grâce à un système d'accès conditionnel. Ces dispositifs techniques ou logiciels peuvent s'appliquer à tout types de supports numériques et peuvent viser à :

- Restreindre la lecture du support à une zone géographique prévue ;
- Restreindre la lecture du support à du matériel spécifique ;
- Restreindre la lecture du support à un constructeur ou vendeur afin de bloquer la concurrence ;



- Restreindre ou empêcher la copie privée du support (transfert vers un appareil externe) ;
- Restreindre ou verrouiller certaines fonctions de lecture du support ;
- Identifier et tatouer numériquement toute œuvre et tout équipement de lecture ou enregistrement ;

Le principe du DRM est d'éviter que la diffusion d'une œuvre numérique échappe au contrôle de son auteur ou de ses ayants droit (producteur, éditeur, distributeur, etc.). La DRM fonctionne en deux temps. Tout d'abord, des droits (ce que l'utilisateur peut et ne peut pas faire) sont fixés pour chaque fichier numérique. Puis, le fichier et les droits qui y sont rattachés sont cryptés à l'aide de puissants algorithmes, afin d'éviter toute tentative de piratage. Le fichier ainsi protégé peut alors être diffusé en toute sécurité. Il n'est lisible que par un programme compatible avec la technologie DRM, qui vérifiera que l'utilisateur possède bien toutes les autorisations requises avant de le décrypter. Mieux : la DRM permet aussi de fixer les droits d'accès au fichier en fonction du matériel utilisé. Ainsi, un document numérique ne pourra être lu que sur le matériel à partir duquel il a été acquis et uniquement sur celui-là.

II.2.6 Paiement électronique

Le paiement électronique est une méthode de paiement qui ne nécessite pas l'usage d'argent liquide. Cette solution est pratique pour accéder à des services sans avoir à se déplacer. Son principal avantage demeure dans la rapidité. Les différents modes de paiement électronique seront développés dans les paragraphes suivants.

■ Paiement par carte bancaire

Ce type de paiement est celui qui permet au consommateur de se connecter à Internet depuis son appareil (ordinateur, smartphone) et de s'offrir des biens et services directement sur Internet à travers une application dédiée ou un navigateur. Il y a plusieurs façons d'effectuer de tels règlements. On peut entrer manuellement ses informations de carte bancaire sur un site pour régler une commande ou utiliser un service de paiement en ligne comme *Paypal*, *Amazon Pay*.

■ Paiement via les cryptomonnaies

Basé sur un puissant algorithme de cryptage appelé blockchain, les cryptomonnaies permettent de s'affranchir des tiers de confiance (établissements bancaires) réduisant ainsi les coûts et les délais de transfert d'argent. Ils offrent la possibilité à toutes personnes disposant d'une connexion internet de désormais



envoyer et recevoir de l'argent à travers le monde de manière sécurisée et sans intermédiaires.

■ Paiement par SMS

Également appelés SMS+ ou premium SMS, les paiements par SMS sont disponibles sur tous les téléphones et consistent à payer un bien ou un service par message textuel. En échange d'un SMS, l'utilisateur reçoit sur son téléphone le bien ou le service qu'il a acheté. Il paye donc via sa facture mobile. L'opérateur reverse ensuite un certain quota au fournisseur d'origine. Comme exemple, on peut citer les publicités proposant des sonneries de téléphone ou des jeux.

■ Paiement par QR code

Créé en 1994 par Dense-Wave, une entreprise japonaise, le QR code est un code-barre à deux dimensions, constitué de modules noirs disposés dans un carré à fond blanc. L'agencement de ces points définit l'information que contient le code. QR(Quick Response) vient du fait qu'il peut-être décodé très rapidement. L'utilisation de deux dimensions permet à ce flashcode de stocker plus d'informations(7089 caractères numériques) qu'un code-barre, et surtout des données directement reconnues par des applications, permettant ainsi de déclencher facilement des actions dont le paiement mobile.

Pour pouvoir effectuer un paiement par QR code, il suffit de lier sa carte ou son compte bancaire à une application de paiement par QR code, de la télécharger et de posséder un smartphone équipé d'un appareil photo. Le principe de paiement consiste à se servir de l'application téléchargée pour scanner le code présenté par le commerçant, et valider pour que la transaction soit réalisée.

Quelques applications permettant de régler via l'utilisation d'un QR code : *Flashiz, Lydia, Zapper.*



Figure 3 : Aperçu d'un QR code



■ Paiement par mobile via USSD

Aussi connu sous les noms Mobile Money et de paiement de mobile à mobile, le paiement par mobile via USSD est une technologie qui permet aux gens de recevoir, garder et dépenser de l'argent en utilisant un téléphone portable. Il est habituel que les services de portefeuille mobile soient fournis par les opérateurs de téléphonie mobile et ils sont disponibles aux abonnés prépayés. Chaque utilisateur du Mobile Money a un numéro de compte unique, généralement identique à leur numéro de téléphone portable. Au lieu de payer par carte de crédit ou bien en argent comptant, un consommateur peut utiliser un téléphone mobile pour se procurer des biens et des services. Pour ce faire, il faut entrer le numéro de téléphone sur une page de paiement ou sur une application, puis passer une ou plusieurs étapes d'identification qui confirment l'identité du propriétaire de ce numéro, par exemple en répondant à un SMS.

Les portefeuilles mobiles sont une alternative populaire aux espèces et aux banques parce qu'ils sont faciles à utiliser, sécurisés et on peut les utiliser partout où il y a un signal de téléphone portable. Aussi, les utilisateurs peuvent retirer de l'argent de leur portefeuille mobile et l'alimenter avec un dépôt de cash auprès des agences agréées.

■ Paiement par NFC

Inventée par Charles Walton en 1983, la NFC (Near Field Communication) utilisait alors l'identification par radiofréquence RFID (Radio Frequency Identification) pour envoyer des fréquences radio à des récepteurs spécifiques RFID. C'est en 2010 que les consommateurs voient la première introduction du NFC sur le marché des smartphone. La technologie NFC est un protocole de communication sans fil, à courte portée (10 centimètres maximum), permettant l'échange d'informations entre deux périphériques, tels qu'un téléphone et un terminal de paiement électronique.

Pour réaliser son paiement, l'utilisateur doit passer son téléphone mobile sur une borne de paiement puis valider son règlement par un code. Pour que le paiement sans contact fonctionne, tous les dispositifs impliqués doivent avoir une puce NFC. Cette puce entre en communication avec une autre puce NFC appartenant à un autre appareil et, ce faisant, les deux appareils peuvent commencer à partager des données.

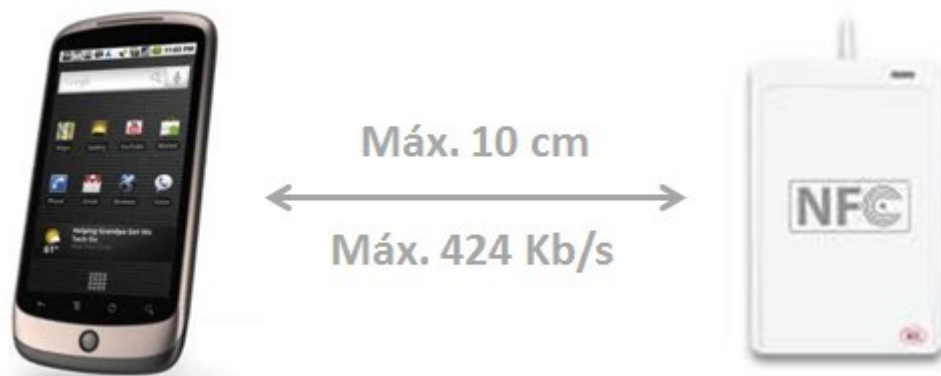


Figure 4 : Image paiement par NFC

■ Paiement par ondes sonores

Le paiement électronique par ondes sonores ou par signal sonore est une nouvelle solution de paiement pour mobiles à la pointe de la technologie. Pas besoin d'internet : les transactions sont rendues possibles par l'utilisation d'ondes sonores uniques contenant les données cryptées du paiement. Les ondes sonores sont envoyées depuis le terminal vers le téléphone mobile pour transmettre les détails du paiement, puis le téléphone du client convertit ces données en signaux analogiques qui finalisent la transaction. Ce système implique l'installation d'un logiciel pour fonctionner avec un portefeuille numérique, une application bancaire ou un lecteur de carte. C'est une solution abordable, en particulier dans les régions du monde où la population ne peut s'offrir les smartphones dernier cri, et doit s'appuyer sur des technologies plus simples pour les paiements mobiles.

■ Paiement par transmission magnétique sécurisée(MST)

La transmission magnétique sécurisée est une technologie qui transmet un signal magnétique d'un appareil à un terminal de paiement. Dans le système MST(Magnetic Secure Transmission), le téléphone émet un signal magnétique qui imite la bande magnétique située sur la carte bancaire du payeur, signal que le terminal de paiement saisit et traite comme si une carte physique était passée dans la machine. Certaines machines à carte nécessitent une mise à jour de leur logiciel pour accepter le MST, mais la plupart des terminaux l'acceptent déjà. MST est conçu pour transmettre à partir de 76 mm du lecteur de carte magnétique.



Conclusion

Après nous être familiarisé avec les concepts clés de notre sujet, nous allons faire un tour d'horizon des solutions existantes et de leurs éventuels manquements afin d'en dégager notre cahier de charges.



Chapitre III: AUDIT DE L'EXISTANT ET CAHIER DE CHARGES

Introduction

Dans ce chapitre, nous allons procéder séquentiellement à l'étude de l'existant et à l'élaboration d'un cahier de charges.

III.1 Solutions existantes

Avant de développer notre solution, il est nécessaire de faire un tour d'horizon de ce qui existe en la matière pour en dégager les qualités ainsi que les limites. Cela nous permettra de bâtir un cahier de charge cohérent et conséquent.

III.1.1 Kiosque numérique de Sogapresse

SOGAPRESSE(Société Gabonaise de Presse) existe depuis 1983, avec pour cœur de métier la diffusion impartiale de l'ensemble des journaux et périodiques locaux et internationaux en République Gabonaise. Elle dispose d'un kiosque numérique dont le site web est www.e-kiosque-sogapresse.com, permettant de consulter et d'acquérir des documents au moyen de Paypal(système de paiement en ligne), et du Mobile Money(MobiCash, Airtel Money). Où Mobicash et Airtel Money sont les services de paiement offert respectivement par Libertis et Airtel qui sont des entreprises de téléphonie mobile implantés au Gabon. Recherche pas assez performante, lecture des documents fastidieuse(tous les écrits ne sont toujours visibles) sont les principales lacunes du e-kiosque de sogapresse. De plus il n'est pas assez intuitif et n'offre pas la possibilité aux lecteurs de consulter les documents acquis hors ligne.

III.1.2 Autres kiosques numériques

Afin de mieux appréhender l'existant, nous nous sommes intéressés à la distribution des documents numériques dans un pays dont le taux de pénétration du mobile est similaire à celui du Gabon. Celui qui a attiré notre attention est le Cameroun ; la distribution de la presse dans ce pays a connu une grande avancée cette dernière décennie, notamment avec l'avènement des kiosques numériques. C'est ainsi qu'a le jour le site de e-kiosque www.ekiosque.cm qui regroupe une trentaine d'éditeurs. Il permet aux lecteurs d'acheter les documents(quotidiens et magazines camerounais) principalement par Mobile Money(Orange Money et Mtn Money) et par Paypal. Orange Money et Mtn Money étant les services de paiement offert respectivement par Orange et Mtn qui sont des entreprises de téléphonie mobile



implantées au Cameroun. La recherche sur ce site n'est pas assez performante ; ce qui constitue un frein pour un lecteur qui solliciterait un document traitant d'un sujet particulier.

A côté des plateformes sus-mentionnés, il existe une multitude de canaux de distribution de documents numériques, à savoir :

- Plateformes généralistes de e-commerce comme RebBox, Amazon, Spotify, Netflix (Streaming et VOD). Ces derniers ont l'inconvénient de ne pas intégrer les paiements locaux et fonctionne dans une logique de One-Stop-Shop(entreprise ou bureau qui offre différents types de services ou de produits aux clients).
- Sites web d'informations des grands médias traditionnels comme Jeune Afrique, Le monde, qui ont comme point faible de ne proposer que leurs offres de contenu.
- Sites web de téléchargement comme Windows Store, Apple Store, Google Play, Softonic qui ne traitent que des logiciels.
- Sites de vente des magazines et de journaux tels que Calameo(calameo.com), Issuu(issuu.com), Publitas(publitas.com) et Joomag (joomag.com) qui n'offrent que des moyens de paiement par carte bancaire.

III.2 Cahier des charges

Après une étude de la problématique, des objectifs globaux, des concepts clés du domaine, des limites des solutions existantes et des ambitions de METRIKA, nous avons établi notre cahier de charge. Le travail consistera à développer une solution de distribution sous forme d'applications web et mobile et ce, dans le paradigme orienté objet. Nous présentons ci-dessous et façon sommaire quelques points de ce cahier de charge.

III.2.1 Collecte des documents

Notre solution va permettre aux différents éditeurs de publier et d'organiser leurs documents au format PDF sur la plateforme.

III.2.2 Acquisition de documents

Les documents publiés pourront être consultés puis acquis par les lecteurs au moyen du paiement électronique, notamment le Mobile Money.

III.2.3 Recherche de documents

La solution offrira aux lecteurs la possibilité d'effectuer des recherches plein texte sur des sujets traités par les documents existants dans le système.



III.2.4 Gestion des droits numériques

La solution future intégrera la gestion des droits d'auteur. Plus précisément, la solution devra empêcher le transfert des documents numériques vers un appareil externe et vers toute autre destination. Ceci en restreignant leur lecture ou accès à deux modes : au matériel à partir duquel ils ont été acquis pour le cas de l'application mobile et au site de l'application web.

III.2.5 Élaboration des statistiques

La solution à mettre en œuvre devra offrir un tableau de bord permettant aux éditeurs de suivre l'évolution des ventes de leurs documents et visualiser les bénéfices engrangés.

Conclusion

Par ce chapitre dédié à l'étude de l'existant et à l'élaboration du cahier de charges, nous mettons un terme à la première partie de notre travail. La partie suivante portera d'abord la méthodologie de développement. Puis nous terminerons par une analyse et une modélisation de notre problème.



Partie II : RÉALISATION DU SYSTÈME

APOSTILLE

La qualité d'une solution dépend du soin accordé à la démarche et aux choix des outils pour la mise en œuvre de ladite solution. Cette partie présente la méthode de développement que nous avons utilisé pour parvenir à une solution ainsi que les différentes activités menées tout au long du processus d'élaboration (analyse, conception) de notre système.

CHAPITRES

CHAPITRE IV: MÉTHODOLOGIE DE DÉVELOPPEMENT

CHAPITRE V: ANALYSE ET MODÉLISATION



Chapitre IV: MÉTHODOLOGIE DE DÉVELOPPEMENT

Introduction

Dans tous les domaines d'activités, lorsque l'on désire créer quelque chose, il est recommandé de suivre une procédure. Une méthodologie de développement est donc un cadre utilisé pour structurer, planifier et contrôler le développement d'une application ; plus généralement, elle est une composition d'une ou plusieurs méthodes en un tout cohérent dans le but de livrer un produit. La méthodologie couvre tout le cycle de vie du logiciel. Dans ce chapitre nous allons procéder à la revue des méthodes de développement existantes en vue d'en choisir une qui nous servira de garde-fou tout au long de notre travail.

IV.1 Tour d'horizon des méthodes de développement

Une méthode est une démarche rationnelle et communicable permettant d'obtenir un certain résultat, souvent à l'aide d'outils. Une méthode de développement des systèmes informatiques est composée :

- **des langages** : systèmes de communication permettant de spécifier, créer et documenter les éléments d'un système logiciel.
- **d'un processus** : ensemble structuré d'activités nécessaires pour développer un logiciel.
- **des modèles** : ensembles de concepts et de règles destinés à expliquer et construire la représentation de phénomènes organisationnels
- **des outils** qui aident à la mise en œuvre des trois composantes ci-dessus.

Il existe une panoplie de méthodes de développement des systèmes d'information, nous avons identifiés les suivantes :

IV.1.1 Les méthodes cartésiennes ou fonctionnelles

Elles analysent les traitements d'un système en termes d'entrées/sorties. Elles abordent le système étudié par les fonctions qu'il doit assurer plutôt que par les données qu'il doit gérer. La démarche conceptuelle préconisée par les approches fonctionnelles consiste à décomposer hiérarchiquement une application en un ensemble de sous applications. Les fonctions de chacune de ces dernières sont affinées successivement en sous fonctions simples à coder dans un langage de



programmation donné. SADT(Structured-Analysis-Design-Technique) est un exemple de méthode cartésienne.

Les points forts des méthodes fonctionnelles sont les suivants :

- Simplicité du processus de conception préconisé,
- Capacité à répondre rapidement aux besoins ponctuels de leurs utilisateurs.

Les points faibles des méthodes fonctionnelles sont les suivants :

- Difficulté de fixer des limites pour les décompositions hiérarchiques,
- Éventuelle redondance des données.

IV.1.2 Les méthodes systémiques

Elles traitent les systèmes à modéliser comme étant un ensemble d'entités(processus) communiquant entre elles et avec l'extérieur par des événements entrants et sortants. Les méthodes systémiques proposent une double démarche de modélisation : la modélisation des données et celle des traitements. Cette double démarche est le résultat direct de l'influence des systèmes de gestion de bases de données (hiérarchique, réseau et relationnel).

Exemple : MERISE(Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise), Axial(Analyse et conception de système d'information assistées par logiciels), Racine(Rationalisation des choix informatiques).

Les points forts des méthodes systémiques sont les suivants :

- Approche globale qui prend en compte la modélisation des données et des traitements,
- Introduction des niveaux d'abstraction dans le processus de conception (niveau conceptuel, niveau logique et niveau physique),
- Bonne adaptation à la modélisation des données et à la conception des bases de données.

Les points faibles des méthodes systémiques sont les suivants :

- Double démarche de conception : les données et les traitements,
- Pas de fusion possible des deux aspects (données et traitements).

Ces méthodes (fonctionnelles et systémiques) sont potentiellement de type descendant : le problème à résoudre est décomposé itérativement en sous problèmes jusqu'à l'identification de procédures élémentaires. Elles ne favorisent pas les critères de :

- **Ré-utilisabilité** : les modules ne sont pas généraux, mais adaptés aux sous problèmes pour lesquels ils ont été conçus,



- **Extensibilité** : l'architecture du logiciel est fondée sur les traitements. Or ces derniers sont moins stables que les données d'où cette approche est inadaptée à la conception de gros logiciels.

IV.1.3 Les méthodes orientées-objet

L'approche orientée objet considère le logiciel comme une collection d'objets dissociés définis par des propriétés. Une propriété est soit un attribut : une entité élémentaire (donnée) de la description de l'état de l'objet, ou une opération : entité élémentaire de la description du comportement de l'objet. Un objet comprend donc à la fois une structure de données (son état sous forme de collection d'attributs) et une collection d'opérations (son comportement).

L'approche orientée objet est potentiellement de type ascendant : un effort de regroupement basé sur l'abstraction des données est entretenu tout au long du processus de conception. En effet, une conception par objets commence par l'identification des objets (entités de base). Ces objets sont regroupés dans des classes selon leurs propriétés. Ensuite ces classes sont à nouveau regroupées en classes plus abstraites appelées modules ou sous-systèmes jusqu'à la modélisation du problème posé.

Les points forts des méthodes orientées objets sont les suivants :

- Intègre dans l'objet des données et des traitements,
- Profite des avantages des concepts objets : phase d'analyse et de conception
- Prend en compte une plus large gamme d'applications,
- Favorise la conception et la réutilisation des composants : concevoir dans un but de réutilisation et non pas pour répondre à un besoin ponctuel,
- Améliore la productivité et la rentabilité en utilisant des bibliothèques de composants réutilisables,
- Simplifie le passage conceptuel/physique,
- Facilite le prototypage.

Les points faibles des méthodes orientées-objet sont les suivants :

- Tous les projets ne peuvent être modélisés avec précision par le modèle d'objets ;
- Modélisation souvent subjective : Ce qui constitue un objet pour un concepteur peut ne pas l'être pour un autre.

Trois méthodes ont véritablement émergé :

- OMT (Object Modeling Technic) de Rumbaugh,
- BOOCH'93 de Booch,
- OOSE(Object Oriented Software Engineering) de Jacobson.



Ces méthodes ont été mises au point autour des années 90. Une de leurs limites était due au fait qu'elles ne disposaient d'aucune dimension méthodologique dans la conception.

À partir de 1994, une méthode utilisant UML(Unified Modeling Language) est née de la fusion des méthodes OMT et BOOCH. Elles sont rejointes en 1995 par Jacobson pour mettre au point une méthode dite unifiée, incorporant les avantages de chacune des méthodes précédentes. UML devient alors une notation universelle pour la modélisation objet.

IV.1.4 Les méthodes agiles

La méthodologie agile est basée sur un développement itératif et incrémenté où les caractéristiques et solutions viennent de la collaboration entre équipes organisées individuellement, mais avec le même but commun. Les méthodes agiles sont itératives, parce que les développements sont organisés selon des cycles courts, plutôt 15 jours ou un mois. Elles sont incrémentales, parce que les nouveaux développements s'ajoutent aux précédents par strates successives. Cette approche itérative et incrémentale est particulièrement adaptée aux logiciels qui mettent en œuvre une interaction homme-machine. Parmi les méthodes agiles importantes, nous pouvons inclure SCRUM et XP(eXtreme Programming).

Les points forts des méthodes agiles sont les suivants :

- Un logiciel utilisable est délivré fréquemment (quelques semaines) ;
- Coopération rapprochée entre développeurs et clients ;
- Le client est au centre des démarches ;
- Projet morcelé en plusieurs étapes réalisables en des délais plus courts ;
- Le travail d'équipe est encouragé.

Les points faibles des méthodes agiles sont les suivants :

- Manque de documentation sur le projet ;
- Méthode adaptée à des projets limités dans le temps ;
- Méthode qui repose beaucoup sur les individus, en particulier le chef de l'équipe de développement (qui doit avoir à la fois des capacités de coach, de leader, et savoir développer) et le client (il doit être disponible, bien connaître l'utilisation future du produit).

IV.1.5 Les méthodes formelles

Elles sont des techniques basées sur les mathématiques pour décrire et modéliser les propriétés et comportements d'un système ; elles fournissent aussi un cadre systématique pour développer le système, le valider et le vérifier. Comme

exemple de méthodes formelles nous avons RAISE(Rigorous Approach for Industrial Software Engineering).

Les points forts des méthodes formelles sont les suivants :

- L'utilisation des concepts(notion d'ensemble, de relation, de quantificateurs existentiel et universel, etc) de la logique et de la technique mathématique pour décrire un problème permet d'établir une spécification de façon simple et claire, et de démontrer les propriétés cette spécification ;
- Absence d'ambiguïté et d'imprécisions : les termes de spécifications formelles n'ont qu'une seule interprétation.

Les points faibles des méthodes formelles sont les suivants :

- L'utilisation d'une méthode formelle allonge la phase de spécification et réduit la phase de test ;
- Elles exigent une formation particulière du personnel ;
- Elles sont incompatibles avec les méthodes existantes ;
- Absence de méthode de spécification formelle universelle.

IV.2 Choix d'une méthode

Pour notre projet, nous préconisons un processus de développement bien défini qui va de la détermination des besoins jusqu'à la conception, le codage final et la validation. C'est juste à titre que nous avons opté pour une démarche basée sur le processus 2-TUP. Elle exploite le langage UML, standard incontournable de la modélisation objet.

IV.2.1 Le langage UML

UML (Unified Modeling Language) se définit comme un langage de modélisation destiné à décrire des besoins, spécifier et documenter des systèmes. Ce langage ayant pour visée le développement logiciel unifie les principales notations et concepts orientés objet. Il fournit treize diagrammes pour permettre la modélisation d'un système depuis la spécification jusqu'à l'implémentation. UML représente un système logiciel selon deux aspects complémentaires : la modélisation statique ou structurelle et la modélisation dynamique ou comportementale.

Notre choix s'est porté sur UML car il répond aux exigences suivantes :

- La syntaxe de UML définit l'ensemble des concepts de modélisation UML, leurs attributs et leurs relations, ainsi que les règles permettant d'associer ces concepts afin de créer des modèles UML partiels ou complets.
- Il fournit une explication détaillée de la sémantique de chaque concept de modélisation UML. La sémantique définit, d'une façon indépendante de la



technologie, comment les concepts UML doivent être mis en œuvre par les ordinateurs.

- Il spécifie des éléments de notation lisibles par l'homme pour représenter chaque concept de modélisation UML, ainsi que les règles pour les combiner au sein d'une grande variété de diagrammes correspondant à différents aspects des systèmes modélisés.

IV.2.2 2TUP

2TUP (2 Track Unified Process), appelé aussi modèle de développement en Y, est un processus de développement qui implémente le processus UP(Unified Process).

UP fournit un cadre au développement logiciel pour la construction de systèmes orientés objet. Il a les caractéristiques suivantes

- construit sur UML
- itératif et incrémental
- piloté par les cas d'utilisation
- orienté par les risques
- centré sur l'architecture du système

Le cycle de développement de UP organise la mise en œuvre d'un système en un ensemble sous-projets. Chaque sous-projet a un cycle de vie en quatre phases :

- **Création** : on définit les objectifs du sous-projet. Cela passe par un recueil des besoins fonctionnels et techniques. Cette phase permet aussi d'évaluer les risques et la faisabilité du sous-projet.
- **Élaboration** : elle sert à clarifier les besoins fonctionnels et techniques, et à poser les bases de l'architecture
- **Construction** : on construit des sous-ensembles livrables et stables du produit final.
- **Transition** : le produit final est déployé, testé et mis à la disposition des utilisateurs. Cette phase inclut la formation des utilisateurs.

Les activités représentent les actions à effectuer au cours d'une phase. Les activités de développement sont définies par des disciplines fondamentales suivantes : modélisation métiers, exigences, conception, implémentation, tests, déploiement, gestion de la configuration, gestion de projet, environnement. La

répartition de l'effort change avec l'évolution du projet dans le temps. Chaque phase a tendance à mettre l'accent sur certaines activités.

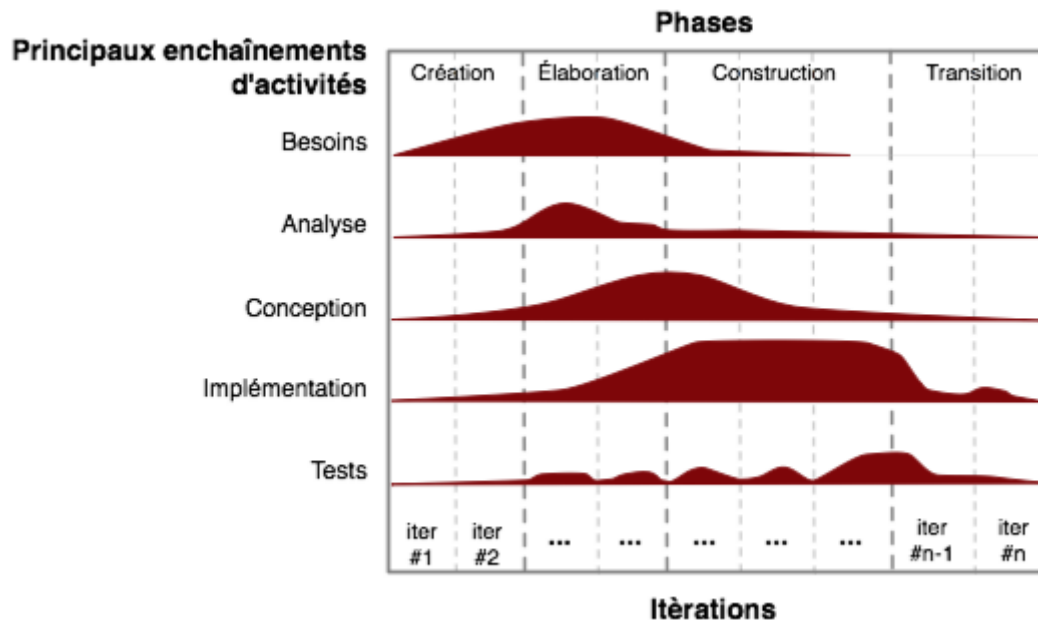


Figure 5: Phases et activités du Processus Unifié

Comme tout processus UP, 2TUP se base sur les formalismes de représentation UML. 2TUP apporte une réponse aux contraintes d'évolution continue des systèmes d'information en décomposant l'analyse du système suivant un axe fonctionnel et suivant un axe technique. Le processus de développement 2TUP comprend trois branches (voir Figure 6).

Ces branches sont précédées par une étude préalable qui est constituée des parties suivantes : présentation du projet, définition des choix techniques, recueil des besoins fonctionnels, recueil des besoins opérationnels, acteurs du système et diagrammes de contexte.

La branche de gauche (branche fonctionnelle) contient deux phases : **Capture des besoins fonctionnels** : 2TUP commence par une description exhaustive des besoins fonctionnels et opérationnels en modélisant les comportements attendus du système. Il s'agit d'une formalisation, par des cas d'utilisation, des exigences fonctionnelles du système sans imposer le mode de réalisation de ces fonctionnalités. La description des cas d'utilisation peut être complétée par des diagrammes dynamiques.

Analyse : cette phase consiste à étudier la spécification fonctionnelle au travers d'une analyse orientée objet pour trouver les classes fondamentales du système et leurs relations. Ces classes sont organisées suivant un regroupement logique sous forme de catégories (packages). Les classes sont complétées et détaillées itérativement au travers d'une analyse de la dynamique du système modélisée par des diagrammes de séquences, de collaboration et d'états.

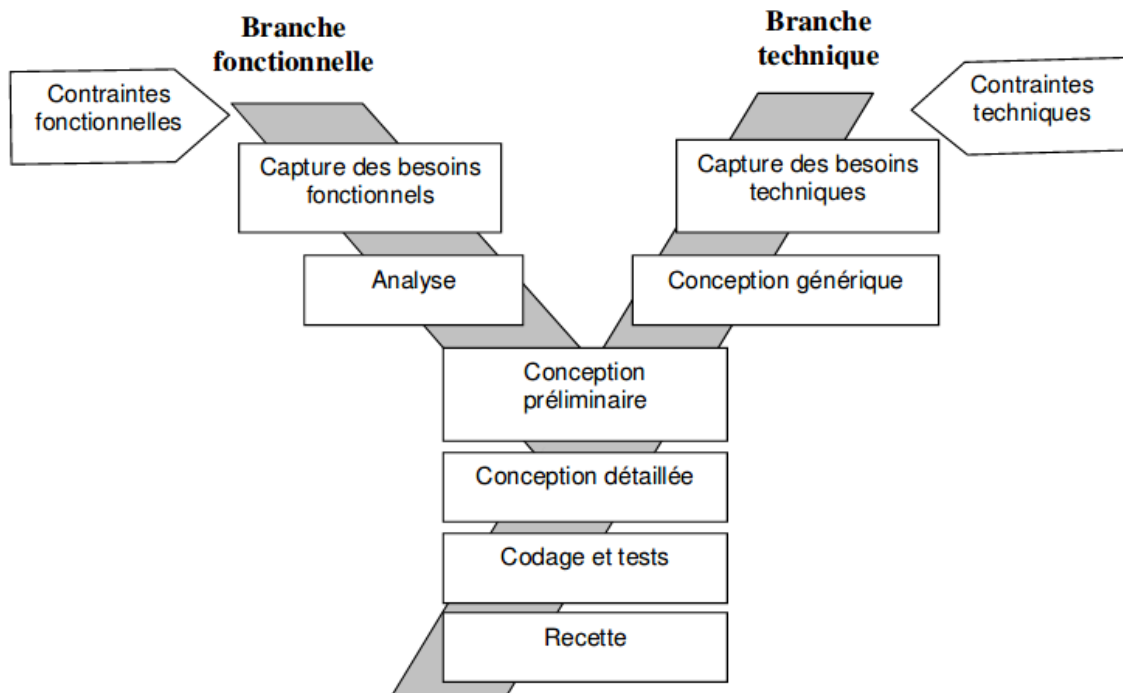


Figure 6 : Processus de développement en Y

La branche de droite (branche technique) intègre deux phases :

Capture des besoins techniques : cette phase recense toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description de l'application. Il s'agit de la spécification des outils (logiciels), de la structure des matériels à exploiter et des contraintes d'intégration avec l'existant. La spécification logicielle et la structure du matériel sont décrites au travers de diagrammes de cas d'utilisation, de packages, de composants et de déploiements.

Conception générique : cette phase définit les composants nécessaires à la construction de l'architecture technique indépendamment des aspects fonctionnels spécifiés dans la branche gauche. Cette conception générique peut aboutir à un



développement d'un prototype de manière à valider les principes par le codage et les tests.

La branche du milieu se compose de quatre phases :

Conception préliminaire : la conception préliminaire consiste à appliquer les concepts liés aux fonctionnalités du système et à intégrer les composants techniques au système. Il s'agit d'intégrer les fonctions métiers et applicatives dans l'architecture technique définie dans la phase de conception générique.

Conception détaillée : il s'agit d'étudier la manière de réaliser les composants. L'objectif d'une conception détaillée consiste à produire un modèle « prêt-à-coder ». Beaucoup de représentations UML sont utilisées dans cette phase : le diagramme de classes représente l'organisation des classes, les diagrammes d'interactions (séquence ou collaboration) montrent la dynamique d'échanges entre les classes en mettant en valeur l'utilité des différentes opérations, les diagrammes d'activités servent à détailler des méthodes, les diagrammes d'états représente les mécanismes d'une classe à état et le diagramme de composants modélisent de la configuration logicielle des sous-systèmes.

Codage et tests : cette phase consiste à la production des composants logiciels et à leurs tests au fur et à mesure de leur implémentation.

Recette : c'est la phase de validation des fonctionnalités du système développé.

Conclusion

Dans ce chapitre nous avons fait un tour d'horizon des méthodes de développement existantes. Ce qui nous a permis de choisir notre méthodologie. Dans le chapitre suivant, nous aborderons l'aspect conception avec les moyens offerts par ladite méthodologie.



Chapitre V: MISE EN ŒUVRE DE LA SOLUTION

Introduction

Dans ce chapitre, nous allons détailler la démarche, l'organisation nécessaire pour construire notre système en respectant les jalons de la méthode choisie précédemment. En effet, la démarche 2 TUP recommande la construction de notre système en une succession de sous-projets. Chaque sous-projet a un cycle de vie en 4 phases: création, élaboration, construction et transition. Pour les besoins de notre mémoire, nous allons présenter la mise en œuvre de 3 sous-projets ; le premier sous-projet étant la définition globale de l'architecture de notre système et ses fonctionnalités, et les autres, la mise en œuvre des incréments.

V.1 Projet 1 : Étude globale du système

V.1.1 Étude préalable

V.1.1.1 Projet à réaliser

Notre projet consiste en une étude préliminaire des fonctionnalités de notre système.

V.1.1.2 Recueil des besoins fonctionnels

Il s'agit des grandes fonctionnalités du système. Ces besoins fonctionnels expriment les principaux processus métiers que doit traiter le système :

- Collecte de documents
- Acquisition de documents
- Gestion des comptes éditeur
- Gestion des comptes client
- Recherche documentaire
- Gestion des abonnements
- Lecture de documents
- Suivi de l'évolution des ventes et des transactions de paiement
- Gestion des quote-part des éditeurs

V.1.1.3 Recueil des besoins opérationnels

- Les échanges entre les acteurs et le système doivent être sécurisées



- Le transfert de documents vers un appareil externe ou vers toute autre destination ne doit être possible
- La création d'un compte utilisateur se fait en fournissant une adresse email ou un numéro de téléphone valide
- Journalisation des actions effectuées dans le système
- Le site web doit être responsive

V.1.1.4 Identification des acteurs

Un acteur est rôle qu'un utilisateur ou une entité externe peut jouer en interagissant avec le système. Les acteurs peuvent consulter et/ou modifier directement l'état du système . Ceci dit, les entités revendiquant ces qualifications dans le cadre de notre application sont les suivantes :

- **Le visiteur** est toute personne qui se rend sur la plateforme.
- **Le client ou le lecteur** est toute personne pouvant se procurer un document.
- **L'éditeur** est celui qui est chargée de poster des contenus(document) sur la plateforme.
- **L'administrateur** qui est chargé de gérer les ressources du système ainsi que les comptes utilisateurs.
- La **passerelle de paiement** qui est chargé de récupérer les données de paiement (montant, numéro compte débiteur, numéro compte créditeur, etc) et de procéder au paiement proprement dit (débiter le compte client et créditer le compte marchand).

V.1.1.5 Identification des messages

Un message représente la spécification d'une communication unidirectionnelle entre les objets, il transporte de l'information avec l'intention de déclencher une activité chez le récepteur.

Le système émet les messages suivants	Le système reçoit les messages suivants
<ul style="list-style-type: none">• Historique de connexion• Les documents en vente• Liste des éditeurs• Liste des clients• Les documents achetés par un client• Les documents publiés par un éditeur• Les documents correspondant à un critère de recherche	<ul style="list-style-type: none">• Connexion• Chargement des documents• Création et modification de compte éditeur• Création et modification de compte client• Achats de documents• Demande liste des documents• Recherche de documents• Création des abonnements

- | | |
|--|---|
| <ul style="list-style-type: none">• Les abonnements auxquels a souscrit un client• Statistiques sous forme de diagrammes circulaire et à bandes• Envoi des données de paiement | <ul style="list-style-type: none">• Statistiques de ventes• Résultat de la transaction |
|--|---|

Tableau 2: Récapitulatif des messages émis et reçus par le système

V.1.1.6 Modélisation du contexte

La modélisation du contexte se fait à l'aide des diagrammes de contexte. Ils servent à délimiter le contour du système en cours d'étude, c'est-à-dire définir clairement ses frontières et les acteurs avec lesquels il communique. Il en existe deux types : Diagramme de contexte statique et diagramme de contexte dynamique.

V.1.1.6.1 Diagramme de contexte statique

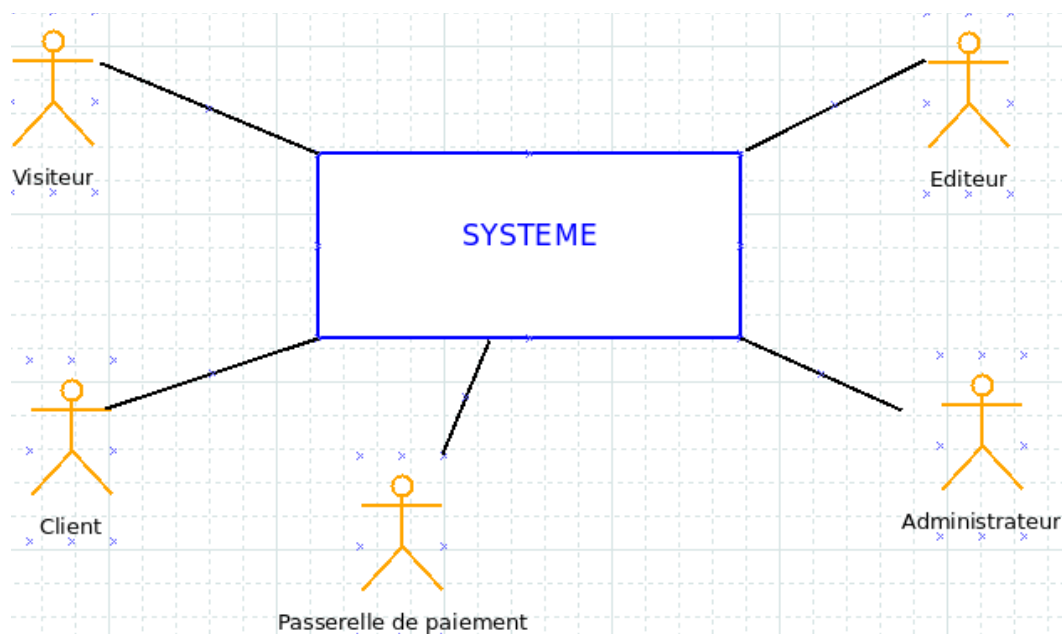


Figure 7: Diagramme de contexte statique



V.1.1.6.2 Diagramme de contexte dynamique

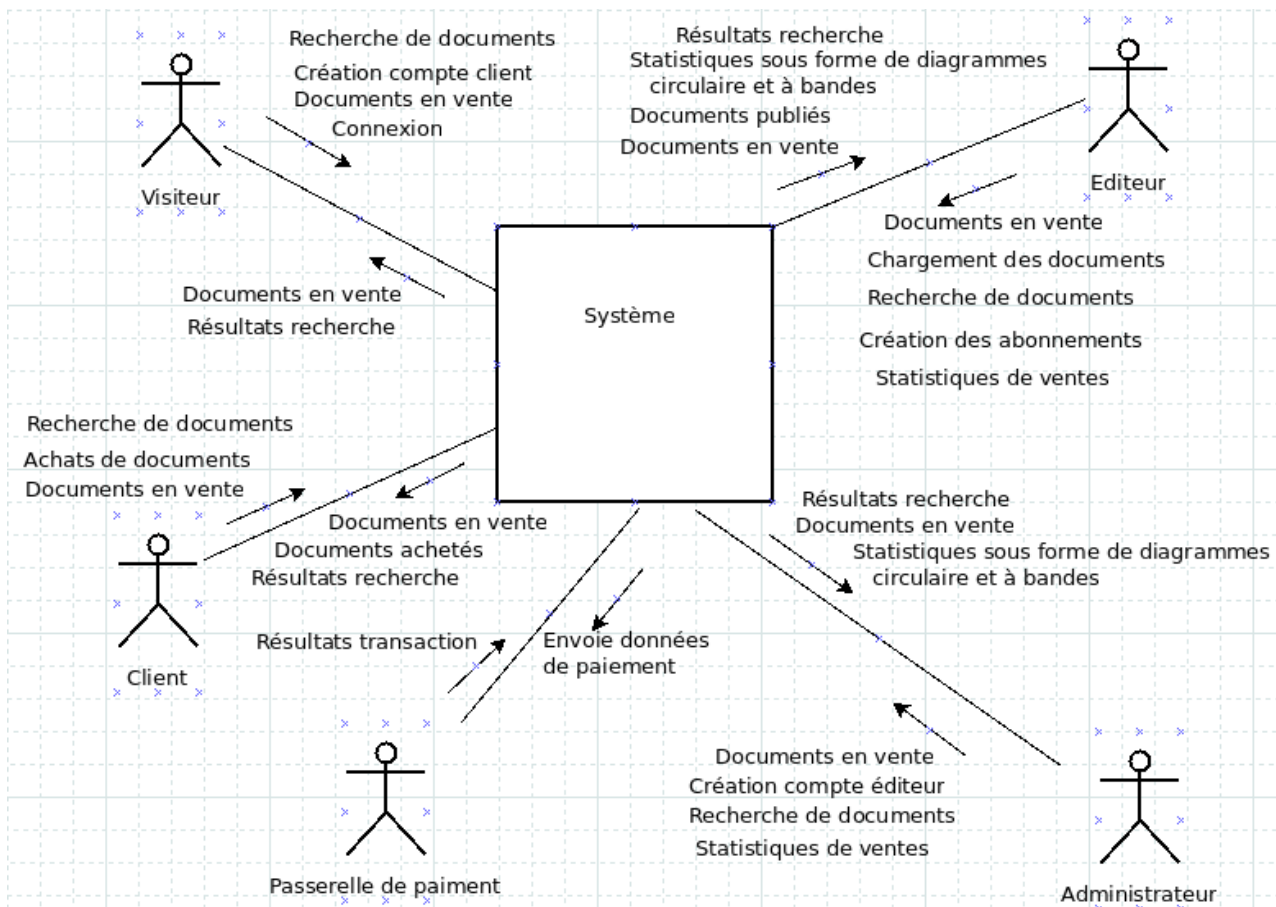


Figure 8: Diagramme de contexte dynamique

V.1.1.7 Définition des choix techniques

V.1.1.7.1 Architecture matérielle

Notre solution est pensée avant tout pour fonctionner dans les environnements réseaux de type TCP/IP basé sur ETHERNET ou WIFI (IEEE 802.11). L'architecture de configuration matérielle est constituée de :

- D'un poste de travail ou d'un smartphone.
- Serveur pour l'hébergement des composants de notre plateforme.



V.1.1.7.2 Langage de programmation

Le langage de programmation choisi doit supporter l'orientée-objet, doit avoir un type statique fort afin d'éviter les erreurs et de faciliter le débogage. Il doit être multi-plateforme, portable et supporter la programmation concurrente. Il doit être un langage pour lequel des compétences peuvent être facilement trouvées.

V.1.1.7.3 Architecture logicielle

L'architecture choisie devra faciliter le découplage des fonctionnalités du système. Elle devra permettre le découpage du système en composants par une séparation claire des préoccupations (présentation, logique applicative, logique métier, accès aux données).

Les données de notre système seront sauvegardées dans un SGBDR (système de gestion de bases de données relationnelles) car ce dernier offre un support fiable de transactions. Notre système sera déployé au travers d'un serveur d'applications. Un framework qui facilite le travail de développement en offrant un ensemble de composants pour gérer des problématiques tels que l'accès aux données, l'authentification, le développement des composants graphiques, la journalisation, etc.

V.1.1.7.4 Outils de travail

Pour le développement notre système, nous aurons besoin :

- d'un environnement de développement intégré pour faciliter l'édition de code, de déploiement, d'écriture de tests
- d'un modèleur UML
- d'un outil de gestion de version de code source
- d'un outil de traitement de texte pour l'édition de nos rapports
- d'un outil d'édition et de présentation de diapositives

V.1.2 Capture des besoins fonctionnels

V.1.2.1 Identification des cas d'utilisation

Les cas d'utilisations décrivent les fonctionnalités fournies par le système à un acteur. Ils permettent de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Dans le cadre notre travail, nous avons pu identifier les use cases listés dans le tableau ci-après.



Cas d'utilisation	Acteurs
S'authentifier	Client, Éditeur, Administrateur
Créer compte client	Visiteur
Désactiver un compte client	Administrateur
Modifier un compte client	Client
Uploader un document	Éditeur
Rechercher un document	Visiteur, Client, Éditeur, Administrateur
Modifier un document	Éditeur
Acheter un document	Client
Lire ou ouvrir un document	Client
Consulter les documents en vente	Visiteur, Client, Éditeur, Administrateur
Créer un compte administrateur	Administrateur
Créer un compte éditeur	Administrateur
Désactiver un compte éditeur	Administrateur
Modifier un compte éditeur	Éditeur
Créer abonnement	Éditeur
Visualiser les statistiques de vente	Administrateur, éditeur
Souscrire à un abonnement	Client
Modifier un abonnement	Éditeur

Tableau 3: Cas d'utilisation et acteurs associés

V.1.2.2 Description textuelle des cas d'utilisation

■ Cas d'utilisation «S'authentifier»

➤ Sommaire d'identification

- ◆ Titre : **S'authentifier**
- ◆ Résumé : Ce cas d'utilisation permet à un utilisateur d'ouvrir une session sur le système.
- ◆ Acteurs : Client, Éditeur, Administrateur
- ◆ Date de création : 09-10-2019
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions



- Le système est accessible.
- ◆ Post-conditions
 - L'événement est enregistré dans le journal du système.

➤ Scénario nominal

1. L'utilisateur clique sur le bouton de connexion.
2. Le système affiche le formulaire d'identification.
3. L'utilisateur remplit le formulaire avec son email et son mot de passe, puis valide.
4. Le système vérifie qu'il existe un utilisateur dans la base de données dont l'email et le mot de passe correspondent à ceux renseignés dans le formulaire.
5. Le système signale que l'opération s'est déroulée avec succès.

➤ Scénario alternatif

A1 : Le système ne trouve pas d'utilisateur correspondant aux identifiants saisis.

L'enchaînement A1 démarre au point 4 du scénario nominal.

1. Le système renvoie un message d'erreur et signale à l'utilisateur de recommencer.
2. Retour au point 2 du scénario nominal.

■ Cas d'utilisation «Créer compte client»

➤ Sommaire d'identification

- ◆ Titre : **Créer compte client**
- ◆ Résumé : Ce cas d'utilisation permet à un visiteur de s'inscrire sur la plateforme.
- ◆ Acteur : Visiteur
- ◆ Date de création : 10-10-2019
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible.
- ◆ Post-conditions
 - Le nouvel client est enregistré en base de données.
 - Cette opération est enregistrée dans le journal du système.

➤ Scénario nominal

1. L'utilisateur clique sur le bouton d'inscription.
2. Le système affiche le formulaire d'inscription.



3. L'utilisateur remplit le formulaire avec les informations suivantes : noms, prénoms, email, mot de passe, numéro de téléphone et puis, valide.
4. Le système vérifie que les informations soumises sont valides : noms, email, mot de passe et numéro de téléphone sont obligatoires.
5. Le système renvoie une notification de succès.

➤ Scénario alternatif

A2 : Les informations renseignées par le visiteur ne sont pas valides.

L'enchaînement A2 démarre au point 4 du scénario nominal.

1. Le système renvoie un message d'erreur et demande à l'utilisateur de recommencer.
2. Retour au point 2 du scénario nominal.

■ **Cas d'utilisation «Uploader un document»**

➤ Sommaire d'identification

- ◆ Titre : **Uploader un document**
- ◆ Résumé : Ce cas d'utilisation permet de charger un document sur la plateforme afin de le rendre accessible aux lecteurs.
- ◆ Acteur : Éditeur
- ◆ Date de création : 10-10-2019
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible.
 - L'utilisateur est connecté.
- ◆ Post-conditions
 - Cette opération est enregistrée dans le journal du système.

➤ Scénario nominal

1. L'utilisateur navigue jusqu'à la page de chargement du fichier.
2. Le système affiche le formulaire de chargement du document.
3. L'éditeur le remplit avec les informations suivantes : le document, son prix, son type, le résumé du document et sa date de parution.
4. Le système vérifie que les données envoyées sont correctes.
5. Le système renvoie une notification de succès de l'opération.

➤ Scénario alternatif

A3 : Les données envoyées ne sont pas valides.

L'enchaînement A3 démarre au point 4 du scénario nominal.

1. Le système renvoie un message d'erreur et demande à l'utilisateur de recommencer.



2. Retour au point 2 du scénario nominal.

➤ Contraintes non-fonctionnels

- ◆ La taille du document numérique doit pas être inférieure ou égale à 20 mégaoctets.
- ◆ Le document doit être au format PDF.

■ Cas d'utilisation «Rechercher un document»

➤ Sommaire d'identification

- ◆ Titre : **Rechercher un document**
- ◆ Résumé : Ce cas d'utilisation permet de retrouver un document sur le système en spécifiant les critères auxquels il doit répondre.
- ◆ Acteurs : Visiteur, Client, Éditeur, Administrateur.
- ◆ Date de création : 11-10-2019.
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible.
 - L'utilisateur est connecté.

➤ Scénario nominal

1. L'utilisateur navigue jusqu'à la page de recherche.
2. Le système affiche le formulaire de recherche.
3. L'utilisateur renseigne ses critères(type de document, date de parution, éditeur) de recherche et les soumet.
4. Le système lui renvoie le résultat de son opération.

■ Cas d'utilisation «Acheter un document»

➤ Sommaire d'identification

- ◆ Titre : **Acheter un document**
- ◆ Résumé : Ce cas d'utilisation permet d'acquérir un document.
- ◆ Acteur : Client
- ◆ Date de création : 11-10-2019.
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible.
 - L'utilisateur est connecté.
- ◆ Post-conditions
 - L'utilisateur a désormais accès au document acheté.



- Cette opération est enregistrée dans le journal du système.

➤ Scénario nominal

1. L'utilisateur sélectionne le ou les document(s) à acheter et valide.
2. Le système affiche le récapitulatif(document et prix) de son achat.
3. L'utilisateur vérifie sa liste d'achat et clique sur «Commander».
4. Le système affiche les moyens de paiement qu'il supporte.
5. L'utilisateur choisi le moyen qui lui convient et valide.
6. Le système l'invite à renseigner ses données(numéro de téléphone ou numéro de carte bancaire) de paiement.
7. L'utilisateur renseigne lesdites données et valide.
8. Le système confirme le succès de l'opération.

➤ Scénarios alternatifs

A4.1: Les données de paiement sont erronées.

L'enchaînement A4.1 commence au point 7 du scénario nominal.

1. Le système renvoie un message d'erreur et demande à l'utilisateur de recommencer.
2. Retour au point 4 du scénario nominal.

A4.2: L'utilisateur veut ajouter des articles à son panier.

L'enchaînement A4.2 commence au point 2 du scénario nominal.

3. L'utilisateur clique sur le bouton « continuer les achats».
4. Le système affiche la page de listing des documents en vente.
5. L'utilisateur remplit son panier et clique sur le bouton « commander ».
6. Retour au point 3 du scénario nominal.

➤ Scénario d'exception

E1 : L'utilisateur annule l'achat et l'opération se termine par un échec.

➤ Contraintes non-fonctionnels

- ◆ Le solde du client doit être strictement supérieure au montant de l'achat.

■ Cas d'utilisation «Ouvrir un document»

➤ Sommaire d'identification

- ◆ Titre : **Ouvrir un document**
- ◆ Résumé : Ce cas d'utilisation permet d'ouvrir un document contenu dans la bibliothèque personnelle de l'utilisateur.
- ◆ Acteur : Client
- ◆ Date de création : 12-10-2019
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible.



- L'utilisateur est connecté.

➤ Scénario nominal

1. L'utilisateur navigue jusqu'à sa bibliothèque.
2. Le système affiche la liste des documents déjà achetés par l'utilisateur.
3. L'utilisateur sélectionne celui qu'il souhaite visualiser.
4. Le système ouvre le fichier et affiche son contenu à l'utilisateur.

■ **Cas d'utilisation «Modifier les informations de connexion»**

➤ Sommaire d'identification

- ◆ Titre : **Modifier les informations de connexion**
- ◆ Résumé : Ce cas d'utilisation permet de modifier les informations de connexion tels que le mot de passe.
- ◆ Acteurs : Client, éditeur, administrateur.
- ◆ Date de création : 12-10-2019.
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible.
 - L'utilisateur est connecté.
- ◆ Post-conditions
 - Les informations de connexion de l'utilisateur sont mises à jour.
 - Cette action est inscrite dans le journal du système.

➤ Scénario nominal

1. L'utilisateur clique sur le bouton de modification des informations.
2. Le système affiche le formulaire de modification.
3. L'utilisateur remplit le formulaire avec les informations nécessaires et les soumet.
4. Le système vérifie que les informations soumises sont valides.
5. Le système renvoie une notification de succès.

➤ Scénario alternatif

A5 : Les informations renseignées par l'utilisateur ne sont pas valides.

L'enchaînement A5 démarre au point 4 du scénario nominal.

1. Le système renvoie un message d'erreur et demande à l'utilisateur de recommencer.
2. Retour au point 2 du scénario nominal.

➤ Contraintes non-fonctionnelles

- ◆ Aucun utilisateur ne doit pouvoir modifier les informations de connexion d'un autre.



■ **Cas d'utilisation «Créer un compte éditeur»**

➤ Sommaire d'identification

- ◆ Titre : **Créer un compte éditeur**
- ◆ Résumé : Ce cas d'utilisation permet d'enregistrer un nouvel éditeur dans le système.
- ◆ Acteur : Administrateur
- ◆ Date de création : 12-10-2019
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions
 - Le système est accessible
 - L'utilisateur est connecté
- ◆ Post-conditions
 - Le nouvel éditeur est enregistré en base de données.
 - Cette opération est enregistrée dans le journal du système.

➤ Scénario nominal

1. L'utilisateur clique sur le bouton de création d'un nouvel éditeur.
2. Le système affiche le formulaire de création.
3. L'utilisateur remplit le formulaire avec les informations suivantes : noms, prénoms, email, numéro de téléphone, maison d'édition.
4. Le système vérifie que les informations soumises sont valides.
5. Le système renvoie une notification de succès.

➤ Scénario alternatif

A6 : Les informations renseignées par l'utilisateur ne sont pas valides.
L'enchaînement A6 démarre au point 4 du scénario nominal.

1. Le système renvoie un message d'erreur et demande à l'utilisateur de recommencer.
2. Retour au point 2 du scénario nominal.

■ **Cas d'utilisation «Supprimer un utilisateur»**

➤ Sommaire d'identification

- ◆ Titre : **Supprimer un utilisateur**
- ◆ Résumé : Ce cas d'utilisation a pour de supprimer un utilisateur du système.
- ◆ Acteur : Administrateur
- ◆ Date de création : 12-10-2019
- ◆ Version | Responsable : 1.1 | Kenmegne

➤ Description des enchaînements

- ◆ Pré-conditions



- Le système est accessible
- L'acteur est connecté
- ◆ Post-conditions
 - Le nouvel éditeur est enregistré en base de données.
 - Cette opération est enregistrée dans le journal du système.
- Scénario nominal
 1. L'acteur navigue jusqu'à la page de listing des utilisateurs.
 2. L'acteur sélectionne l'utilisateur et valide sa suppression.
 3. Le système demande confirmation de la suppression de l'utilisateur.
 4. L'acteur confirme la suppression.
 5. Le système confirme le succès de l'opération.
- Contraintes non-fonctionnels
 - ◆ Un utilisateur ne doit s'auto-supprimer.

■ Cas d'utilisation «Affecter un rôle à un utilisateur»

- Sommaire d'identification
 - ◆ Titre : **Affecter un rôle à un utilisateur**
 - ◆ Résumé : Ce cas d'utilisation permet d'assigner un rôle à un utilisateur afin qu'il puisse avoir accès aux fonctionnalités réservées à ce rôle.
 - ◆ Acteur : Administrateur.
 - ◆ Date de création : 12-10-2019.
 - ◆ Version | Responsable : 1.1 | Kenmegne
- Description des enchaînements
 - ◆ Pré-conditions
 - Le système est accessible.
 - L'acteur est connecté.
 - ◆ Post-conditions
 - L'utilisateur a désormais accès aux fonctionnalités du rôle nouvellement assigné.
- Scénario nominal
 1. L'acteur navigue jusqu'à la page qui affiche le listing des utilisateurs et leurs rôles courants.
 2. L'acteur sélectionne l'utilisateur et le rôle à être assigné, puis valide.
 3. Le système enregistre cette modification.

V.1.2.3 Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, produisant un résultat sur un acteur, et

ceci indépendamment de son fonctionnement interne. Des cas d'utilisation identifiés plus haut, nous avons construit le diagramme de cas d'utilisation suivant :

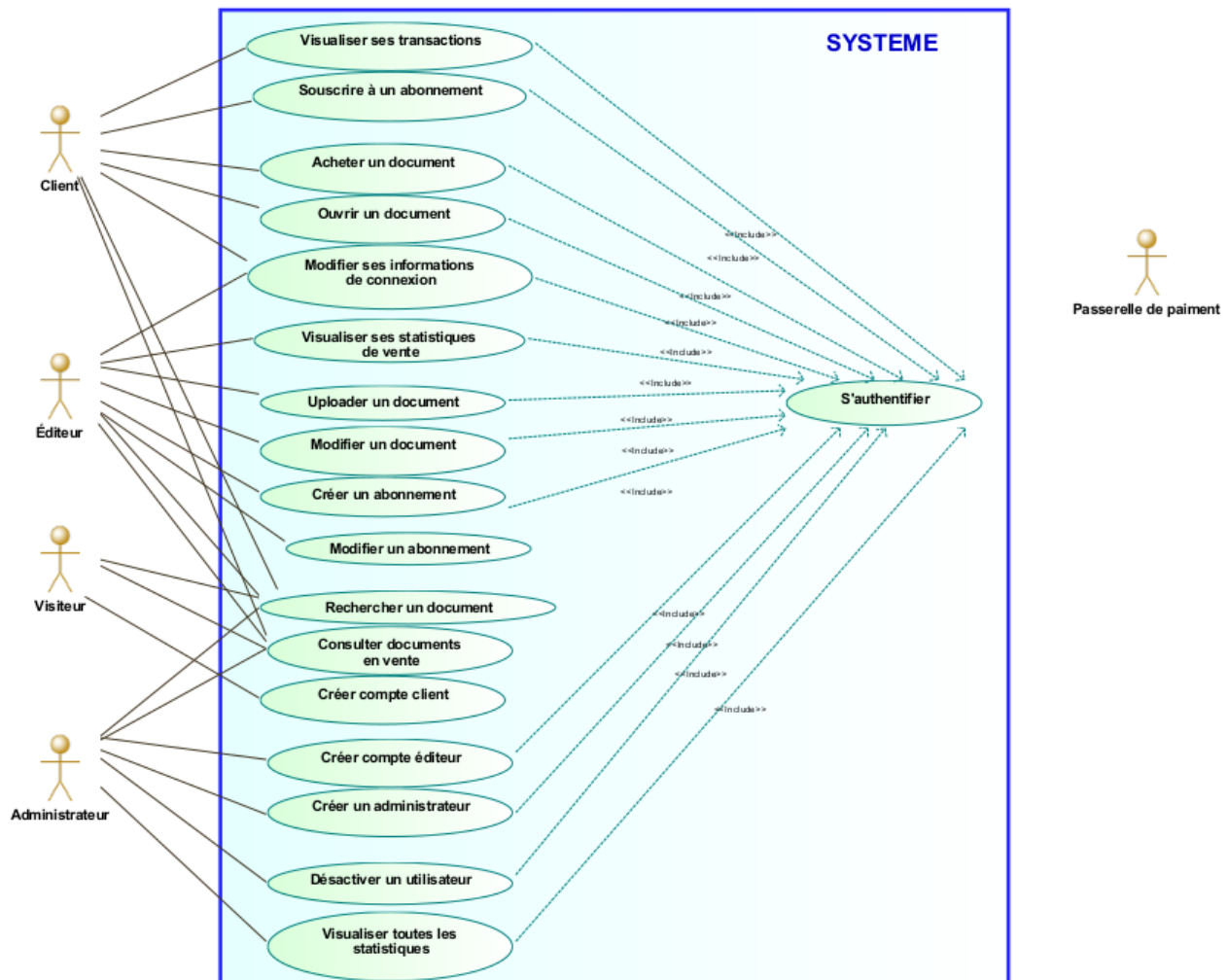


Figure 9: Diagramme des cas d'utilisation

V.1.2.4 Structuration des cas d'utilisation en package

Cas d'utilisation	Package
Créer compte client	Gestion des comptes client
Désactiver un compte client	
Modifier un compte client	
Uploader un document	Collecte de documents
Modifier un document	



Rechercher un document	Acquisition de documents
Acheter un document	
Lire ou ouvrir un document	
Consulter les documents en vente	
Créer un compte administrateur	Administration
Créer un compte éditeur	Gestion des comptes éditeur
Désactiver un compte éditeur	
Modifier un compte éditeur	
Créer abonnement	Gestion des abonnements
Souscrire à un abonnement	
Modifier un abonnement	
Visualiser les statistiques de vente	Statistiques

Tableau 4: Structuration des cas d'utilisation en package

V.1.2.5 Planification du travail en incréments et en itérations

2 TUP étant une démarche itérative et incrémentale, notre planification sera faite sur la base des cas d'utilisation identifiés plus haut. La planification des cas d'utilisation est exposée aux risques suivants :

- ★ Information : ce risque traduit la difficulté à comprendre les vocabulaires métiers du domaine. Pour chaque itération comportant ce type de risque, les concepts métiers méconnus ou mal appréhendés doivent être clarifiés dans le moindre détail.
- ★ Conception : pour ce type de risque, une bonne réflexion sur les aspects conceptuels concernant la réalisation des cas d'utilisation s'impose. C'est le risque le plus important puisque la conception constitue le fondement de l'implémentation.
- ★ Implémentation : les techniques, concepts et outils de programmation choisis doivent être maîtrisés et approfondis afin de satisfaire tous les cas d'utilisation constituant l'itération.

En outre, les différents cas d'utilisation seront classés en tenant compte des deux facteurs suivants : la priorité fonctionnelle (déterminée avec les valeurs : haut, moyen, bas) et les risques techniques (estimé par le chef de projet avec les valeurs : haut, moyen, bas). Pour cela, le chef de projet doit prendre en compte de façon combinée la priorité fonctionnelle et l'estimation du risque selon la logique suivante :



- Si la priorité est haute et le risque également, il faut planifier le cas d'utilisation dans les premières incréments.
- Si la priorité est basse et le risque également, on peut reporter le cas d'utilisation à un des tout derniers incréments.
- Les choses se compliquent lorsque les deux critères sont antagonistes. Le chef de projet doit alors décider en pesant le pour et le contre.

Pour l'élaboration de notre système, nous proposons la planification(en incréments et en itérations) suivante :

Cas d'utilisation	Priorité fonctionnelle	Risques techniques	Niveau risque	du Incrément	Nombre d'itérations
S'authentifier	Haute	Haut	Implémentation	1	3
Créer compte client	Haute	Haut	Information, conception	1	
Désactiver un compte client	Basse	Bas	Implémentation	1	
Modifier un compte client	Basse	Bas	Implémentation	1	
Uploader un document	Haute	Haut	Information, conception	4	3
Modifier un document	Moyenne	Bas	Implémentation	4	
Rechercher un document	Moyenne	Moyen	Information, implémentation	4	
Acheter un document	Haute	Haut	Implémentation	5	3
Ouvrir un document	Haute	Haut	Implémentation	5	
Consulter document en vente	Haute	Haut	Implémentation	5	
Créer un compte administrateur	Moyenne	Moyen	Implémentation	3	3
Créer un	Moyenne	Moyen	Implémentation	3	



compte administrateur					
Désactiver un compte administrateur	Moyenne	Moyen	Implémentation	3	
Créer compte éditeur	Haute	Haut	Information, conception	2	3
Désactiver un compte éditeur	Bas	Bas	Implémentation	2	
Modifier un compte éditeur	Basse	Bas	Implémentation	2	
Créer abonnement	Basse	Bas	Information, conception	6	3
Souscrire à un abonnement	Basse	Bas	Conception, implémentation	6	
Modifier un abonnement	Basse	Bas	Implémentation	6	
Visualiser les statistiques	Bas	Bas	Information, implémentation	7	2
Lister les transactions	Bas	Bas	Implémentation	7	

Tableau 5: Découpage des cas d'utilisation en itérations

V.1.3 Capture des besoins techniques

Il s'agit des contraintes techniques qui caractérisent le système. Ce sont des besoins en matière de performance, de disponibilité, d'utilisabilité, de type de matériels, type d'implémentation (langage de programmation, SGBD, système d'Exploitation...). Parmi les contraintes techniques, nous avons les contraintes suivantes :

- Le traitement et la sauvegarde de fichiers uploadés par les éditeurs : ceci est possible grâce au système de fichiers et SFL4J.
- Le stockage et la manipulation fiables des données. Pour cela nous allons faire usage d'un SGBDR supportant les transactions. Pour notre, nous avons fait le choix de MySQL pour l'application web et de SQLite pour l'application mobile.





- Utilisation du paradigme objet : pour la mise en œuvre de ce paradigme, nous avons opté pour le langage Java (v.1.8). Il est multi-plateforme, portable et supporte la programmation concurrente. HTML (v.5), CSS (v.3) et JavaScript seront aussi utilisés.
- La manipulation des données stockées doit selon les préceptes de l'orientée-objet. Pour cela nous mettrons en œuvre une couche de mappage. La solution adoptée est l'ORM EclipseLink.
- Protection des documents contre la copie-privée
 - ✓ Côté mobile: pour cette contrainte, nous passerons par le chiffrement. Nous utiliserons l'algorithme de chiffrement symétrique AES. Pour chaque document téléchargé sera généré et stocké un clé de chiffrement avec lequel il sera crypté.
 - ✓ Coté web : les documents achetés seront feuilletés et lu en ligne grâce au framework FlipBook3D.
- Sécurité de communication : notre solution sera déployé en intégrant les chiffrements des communications par le protocole SSL.
- Prototypage rapide de fonctionnalités et fiable : pour la mise en œuvre de cette contrainte, nous ferons usage des framework JSF (v2.3) et Primefaces (v7.0).
- L'architecture matérielle répondre ou exigences suivantes :
 - ✓ Le poste de travail doit être doté de capacités (connectique réseau, CPU, Disque dur ...), pour faire tourner un système d'exploitation moderne (doté d'un environnement graphique et d'une pile réseau TCP/IP) comme Linux, Windows NT, UNIX/BSD, etc. Ce poste de travail devra être aussi doté d'un navigateur récent supportant les technologies HTML 5 et CSS 3. Pour répondre aux exigences de performances, le poste doit être multiprocesseurs avec minimum 2 Ghz de fréquence, 2 Go de mémoire vive, des connectiques Ethernet 100 Mbit/s au moins, des connectiques WIFI 802.11 a/b/g, 50 Go minimum de disque dur.
 - ✓ Le smartphone doit avoir les caractéristiques minimales suivantes : système d'exploitation Android version 4.4, processeur dual core de fréquence 1.0 GHz, 5 Go de disque dur, 512 Mo de RAM, des connectiques WiFi IEEE 802.11 b/g/n.
 - ✓ Le serveur sur lequel sera hébergé notre plateforme doit être une machine multiprocesseurs possédant : au minimum 2 Ghz de fréquence, 4 Go de mémoire vive, des connectique Ethernet 100 Mbit/s au moins, 500 Go minimum de disque dur



Le reste de nos besoins techniques est résumé dans le tableau ci-dessous.

Catégorie	Nom	Description	Logo
Environnements de développement	Eclipse photon v4.8	Environnement de développement intégré gratuit pour Java et d'autres langages.	
	Android studio v3.5	Android Studio est l'environnement de développement intégré officiel du système d'exploitation Android, construit sur le logiciel IntelliJ IDEA et conçu spécifiquement pour le développement Android.	
	Gedit v3.18.3	Gedit est l'éditeur de texte par défaut de l'environnement de bureau GNOME. Conçu comme un éditeur de texte à usage général, gedit met l'accent sur la simplicité et la facilité d'utilisation, avec une interface graphique claire et simple.	
Langages de programmation	Java v1.8	Java est un langage de programmation à usage général basé sur des classes, orienté objet et conçu pour avoir le moins de dépendances d'implémentation possible.	
	Java EE v8	Java EE est une plate-forme construite sur le langage Java. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'application.	



Formats de données	XML	XML est un langage qui définit un ensemble de règles pour l'encodage de documents dans un format à la fois lisible par l'homme et lisible par la machine. Son but est de faciliter l'échange de données entre systèmes d'information hétérogènes.	
	JSON	JSON est un format minimal et lisible pour structurer les données. Il est utilisé principalement pour transmettre des données entre un serveur et une application Web, comme alternative au XML.	
Serveur Web	Glassfish v5.0	Glassfish est le serveur d'applications qui implémente à la lettre les spécifications de Java EE.	
SGBD	MySQL v5.7.28	MySQL est la base de données open source la plus populaire au monde. Il est réputé pour être un serveur de base de données SQL rapide, stable, multi-utilisateurs et multi-thread.	
	SQLite	SQLite est une bibliothèque écrite en C qui propose un moteur de base de données SQL. Elle est directement intégrée aux programmes en utilisant des fichiers de bases de données. SQLite est intégré à tous les téléphones mobiles.	



Langage d'interrogation de base de données	SQL	SQL est un langage conçu pour la gestion et la manipulation de bases de données.	
Gestion du Versionning	Git	Git est un système décentralisé de gestion de code. Il est conçu pour suivre les modifications du code source pendant le développement du logiciel et pour coordonner le travail entre les programmeurs. Il peut aussi être utilisé pour suivre les modifications dans n'importe quel ensemble de fichiers.	
Test	JUnit	JUnit est un framework de tests unitaires pour le langage de programmation Java. Junit joue un rôle important dans le développement du développement piloté par les tests.	
Bibliothèques et Frameworks	EclipseLink	EclipseLink est un framework open source de mapping objet-relationnel pour Java.	
	Primefaces v7.0	Primefaces est un framework de composants graphiques pour JSF.	
	Apache Lucene v8.3.0	Lucene est une bibliothèque open source écrite en Java qui permet d'indexer et de chercher du texte.	
	API Airtel Money (Formulaire & REST)	API permettant d'effectuer des paiements mobiles via Airtel Money. Ce dernier étant le service de paiement offert par Airtel(Entreprise de téléphonie implanté au Gabon).	







Modeleur UML	Modelio v3.8	Modelio est un outil de modélisation open source, gratuit et disponible sur les plateformes Windows, Linux et Mac. Prise en main facile, il est conforme à UML 2 et peu de ressources nécessaires à son fonctionnement.	
	Dia v0.97.3-1	Dia est un logiciel de création de diagrammes polyvalent gratuit et open source. Il est écrit en C et est multi-plateforme. Dia peut exporter des diagrammes vers différents formats.	
Édition mémoire et présentation	LibreOffice Writer v5	LibreOffice Writer est le composant de traitement de texte gratuit et open-source du progiciel LibreOffice. Writer est un logiciel de traitement de texte similaire à Microsoft Word avec de nombreuses fonctionnalités similaires et une compatibilité de format de fichier.	
	WPS Presentation v11.1.0.8865	WPS Presentation est l'un des composants de la suite WPS (Writer, Presentation, Spreadsheets) Office. Il permet de créer, modifier et personnaliser les présentations et diaporamas.	

Tableau 6: Récapitulatif des outils de développement

Notre plateforme de développement est bâtie sur une machine ACER, dotée de 8 Go de RAM, de 750 Go de disque dur et d'un microprocesseur Intel core i3 64 bits. Cette machine tourne sous un système Linux, dans sa distribution Ubuntu (version 16.04 LTS).

V.1.4 Conception générique

L'objectif de cette partie est de décrire l'architecture logique de notre système. L'architecture d'un logiciel décrit la manière dont seront agencés les différents composants d'une application et comment ils interagissent entre eux. Le modèle d'architecture décrit comment un système informatique doit être conçu de manière à répondre aux spécifications. Nous avons opté pour une architecture en 3-tiers associée à une architecture MVC.

V.1.4.1 Le MVC

MVC(modèle-Vue-Contrôleur) est un motif d'architecture logicielle composé de trois types de modules ayant trois responsabilités différentes :

- Un **modèle** qui contient les données ainsi que de la logique en rapport avec les données: validation, lecture et enregistrement.
- Une **vue** qui contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle.
- Un **contrôleur** qui est le module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

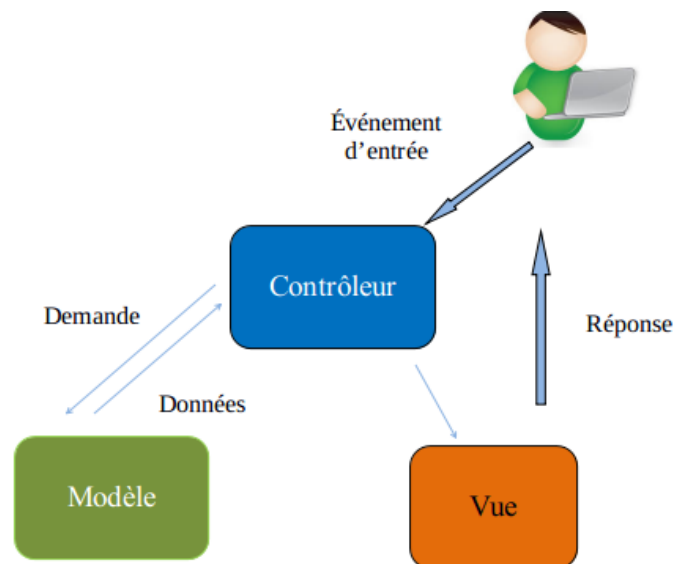


Figure 10: Architecture MVC

V.1.4.2 Architecture client-serveur

L'environnement client-serveur désigne un mode de communication à travers un réseau entre plusieurs programmes : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur ou la machine virtuelle sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur ou la machine virtuelle sur lequel est exécuté le logiciel serveur.

V.1.4.2.1 Types d'architecture client-serveur

➤ Architecture pair à pair

C'est un environnement client-serveur où chaque programme connecté est susceptible de jouer tour à tour le rôle de client et celui de serveur.

➤ Architecture à deux niveaux

Encore appelée architecture deux tiers, c'est un environnement client-serveur où le client demande une ressource au serveur qui la fournit à partir de ses propres ressources.

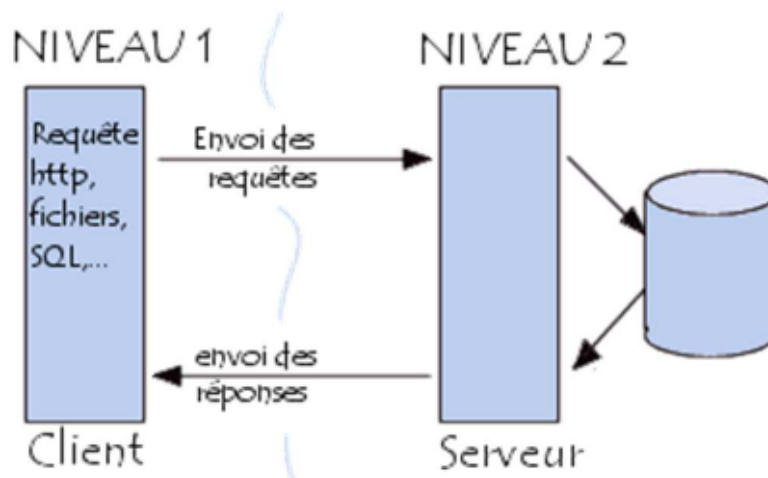


Figure 11: Architecture 2-tiers

➤ Architecture à trois niveaux

Une architecture à trois niveaux ou une architecture trois tiers ajoute un niveau supplémentaire à l'architecture à 2 niveaux, permettant de spécialiser les serveurs dans une tâche précise, ce qui donne un avantage de flexibilité, de sécurité et de performance :

- un **client** qui demande une ressource via une interface utilisateur (généralement un navigateur web) chargée de la présentation de la ressource ;

- un **serveur d'application** (appelé middleware) qui fournit la ressource, mais en faisant appel aux ressources d'un autre serveur ;
- un **serveur de données** qui fournit au serveur d'application les ressources requises pour répondre au client.

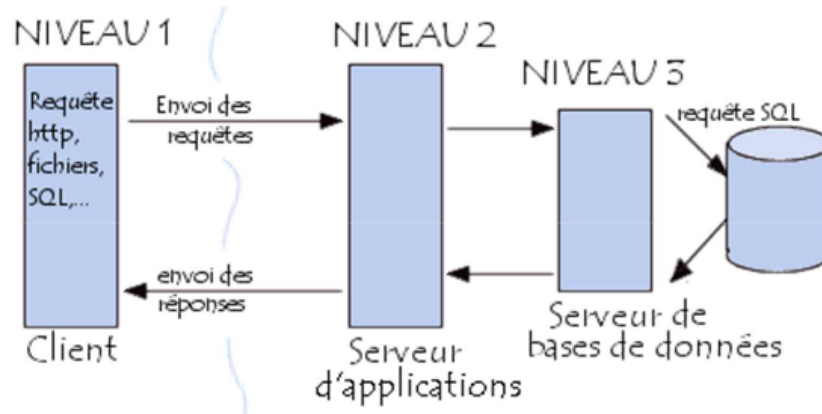


Figure 12: Architecture 3-tiers

➤ Architecture à N niveaux

Une architecture à N niveaux ou architecture N tiers n'ajoute pas encore des niveaux supplémentaires à l'architecture à 3 niveaux, mais introduit la notion des objets qui offre la possibilité de distribuer les services entre les 3 niveaux selon N couches, permettant ainsi de spécialiser les serveurs davantage.

V.1.4.2.2 Types de client

- **Client léger** : application où le traitement des requêtes du client est entièrement effectué par le serveur, le client recevant les réponses « toutes faites ».
- **Client lourd** : application où le traitement des requêtes du client est partagé entre le serveur et le client.
- **Client riche** : application où le traitement des requêtes du client est effectué majoritairement par le serveur, le client recevant les réponses « semi-finies » et les finalisant.

V.1.4.3 Présentation de notre architecture

L'architecture client-serveur à trois niveaux est l'architecture technique correspondant au mieux à notre projet. Sa description est la suivante :

- ✓ Le client (premier tiers) est soit une application mobile Android (client lourd) ou un ensemble de composants (client léger) HTML, CSS, JavaScript s'exécutant

dans un navigateur web. En effet, le lecteur aura le choix entre les plateformes mobile et web pour accéder à notre système. L'appareil mobile envoie des requêtes HTTP au serveur et ce dernier lui transmet une réponse au format JSON. Tandis que le client web enverra des requêtes HTTP au serveur et recevra des fichiers HTML, CSS et JavaScript de ce dernier en guise de réponse.

- ✓ Le second tiers qui constitue le cœur de notre système, implémente la logique applicative et métier. Il est basé sur une architecture MVC implémenté par le framework Primefaces + JSF. Ce composant est exécuté par le serveur d'application Glassfish qui est l'implémentation de référence de la technologie JEE. Ce composant implémente l'API REST(WebServices) utilisé par le client mobile. Il implémente également les objets métiers et la couche d'accès aux données. Ceci est fait à l'aide de l'ORM (Object Relational Mapping) EclipseLink qui est l'implémentation de référence de JPA(Java Persistence API). En définitive, l'architecture MVC de ce composant est la suivante : la vue est assurée par des pages HTML générées à l'aide Primefaces ; le modèle est assuré par des entités JavaEE (JavaBean) et par l'ORM EclipseLink ; le contrôleur par JSF (un unique aiguilleur central, la FacesServlet servant de point d'entrée à toute requête et un objet particulier, l'EJB déclaré via une annotation particulière).

- ✓ Le troisième tiers est le serveur de données MySQL.

Le schéma si-dessous résume ce qui a été dit précédemment.

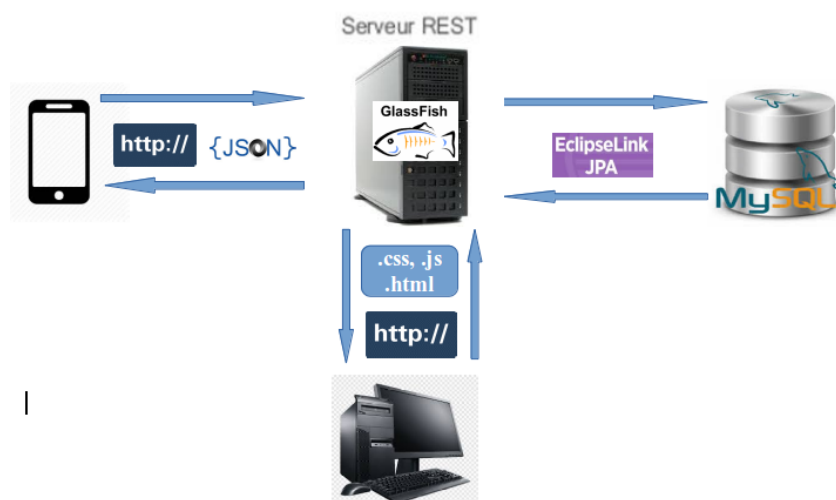


Figure 13: Architecture logique de notre solution



V.1.5 Analyse

V.1.5.1 Classes candidates

Il s'agit d'identifier les principales classes métier de notre système. Ces classes métier sont les suivantes :

- Document
- Transaction
- Utilisateur
- Abonnement
- Revue

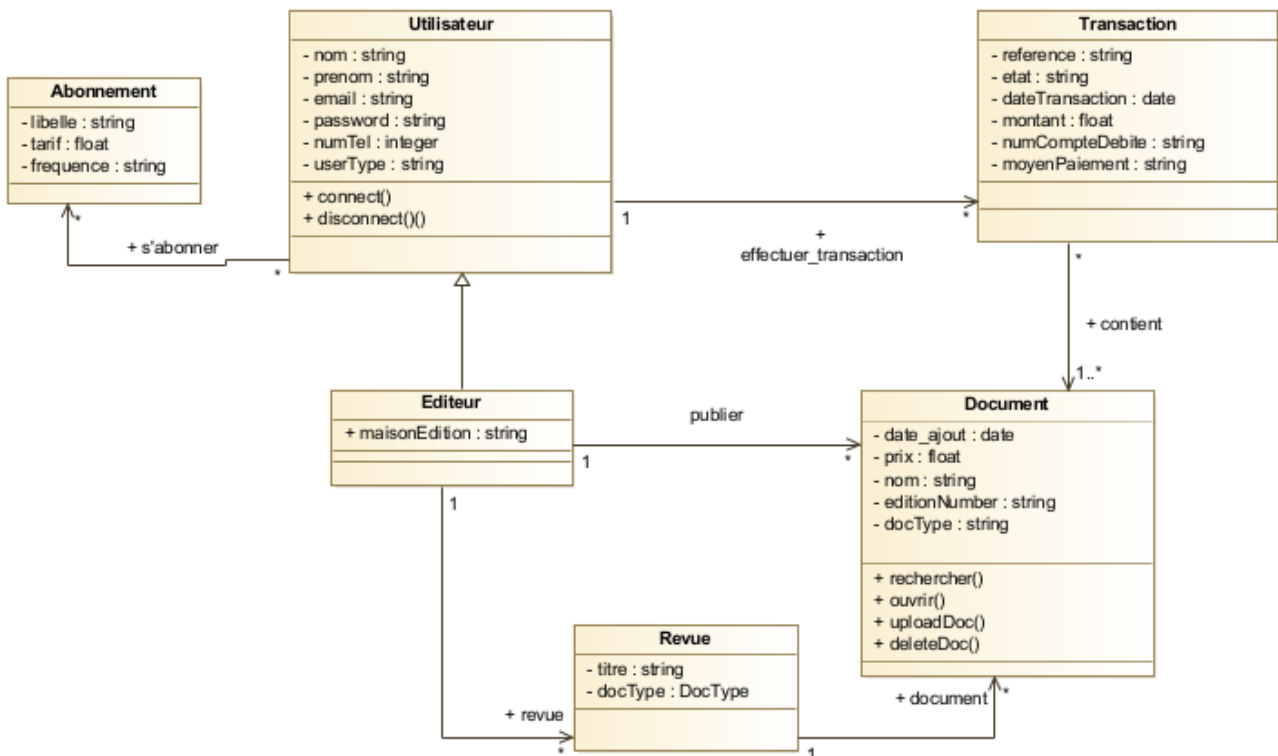


Figure 14: Diagramme de classes préliminaire

V.1.5.2 Diagramme d'activités de navigation

Un diagramme d'activités représente graphiquement la suite d'étapes qui constituent un processus, le comportement d'une méthode ou le déroulement d'un cas d'utilisation et les contraintes de séquençement. Il permet de mettre l'accent sur les traitements. Ci-dessous, le diagramme d'activités de navigation global de notre système.



The diagram illustrates a multi-tier architecture with three main components:

- <<serveur d'application>>** (Application Server):
 - Contains **<<application web>>** (Web Application):
 - <<modele>>** (Model): JavaBean, EclipseLink
 - <<vue>>** (View): PrimeFaces, JSF
 - <<contrôleur>>** (Controller): FacesServlet, EJB
- <<serveur de base de données>>** (Database Server):
 - Contains **MySQL**
- <<client>>** (Client):
 - Contains **<<smartphone>>** (Smartphone):
 - Application mobile**
 - <<database>> SQLite**
 - Contains **<<poste de travail>>** (Workstation):
 - Navigateur Web**

Connections:

- The **<<client>>** connects to the **<<serveur d'application>>** via a network connection (represented by a line with a circle and a plug).
- The **<<serveur d'application>>** connects to the **<<serveur de base de données>>** via a network connection (represented by a line with a circle and a plug).

© 2020



V.1.6 Autres activités

Dans l'optique de respecter la démarche incrémentale préconisé par les processus unifiés, nous ne mettons pas un accent sur les activités de conception préliminaire, conception détaillée, codage, tests et recette dans ce sous-projet. Ces activités seront les activités phares des autres sous-projets.

V.2 Projet 2 : Gestion des comptes client

V.2.1 Étude préalable

V.2.1.1 Projet à réaliser

Ce projet consiste à mettre en œuvre les fonctionnalités liées à la gestion des clients.

V.2.1.2 Recueil des besoins fonctionnels

Les principaux besoins fonctionnels sont :

- Créer un compte client
- Modifier les informations d'un compte client
- Lister les clients
- Désactiver un compte client

V.2.1.3 Recueil des besoins opérationnels

- L'adresse email fourni par l'utilisateur doit être validé ceci afin d'éviter la création des comptes fictifs dans notre système par les robots.
- En cas d'oubli de son mot de passe, le client doit pouvoir récupérer son compte à partir de son adresse email.
- Le mot de passe doit être protégé avant le stockage.
- L'envoi des informations au serveur doit être sécurisé.
- L'email doit être unique.

V.2.1.4 Identification des acteurs

- Visiteur : il peut créer un compte client
- Client : il peut modifier son profil
- Administrateur : il peut lister les comptes client et désactiver un compte client

L'administrateur et le client doivent s'authentifier avant de réaliser les besoins listés ci-dessus.

V2.1.5 Identification des messages

Le système émet les messages suivants	Le système reçoit les messages suivants
<ul style="list-style-type: none"> Liste des clients Résultat de l'action de désactivation 	<ul style="list-style-type: none"> Création de compte client modification de compte client Désactivation de compte client

Tableau 7: Messages émis et reçu par le système dans le projet «Gestion des comptes client»

V2.2 Capture des besoins fonctionnels

V2.2.1 Identification des cas d'utilisation

Cas d'utilisation	Acteur	Interface d'accès
Créer un compte client	Visiteur	Web et mobile
Modifier un compte client	Client	Web et mobile
Désactiver un compte client	Administrateur	Web
Lister les comptes client	Administrateur	Web

Tableau 8: Cas d'utilisation du projet « Gestion des comptes client »

V2.2.2 Diagramme des cas d'utilisation

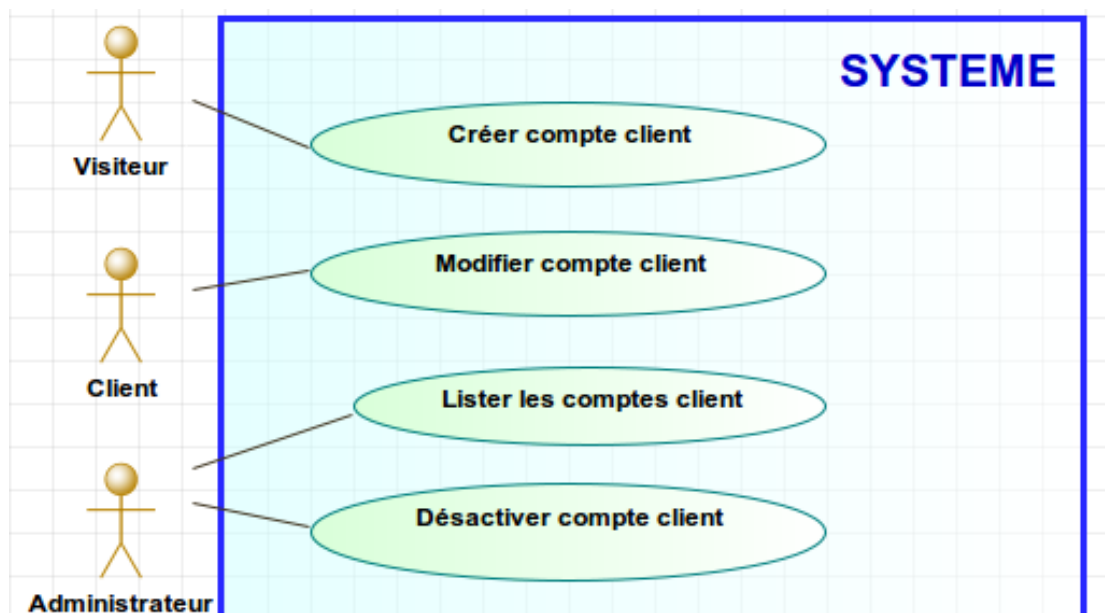


Figure 17: Diagramme des cas d'utilisation du projet «Gestion des comptes client»



V.2.3 Capture des besoins techniques

Les principaux besoins techniques sont :

- Hachage du mot de passe à l'aide de l'algorithme SHA-256 implémenté par la bibliothèque Jasypt.
- L'unicité de l'email est assurée par une contrainte du SGBD et par programmation.
- La protection des informations envoyées au serveur sera garanti par le protocole HTTPS.
- La validation de l'email va consister en l'envoi d'un mail à l'utilisateur contenant un lien et un jeton de validation. Ledit jeton est une chaîne de caractères générée de manière aléatoire. On utilisera une bibliothèque de génération de chaînes pseudo-aléatoires.

V.2.4 Analyse

L'analyse consiste à décrire (dynamiquement et structurellement) le système vu comme une boîte noire.

V.2.4.1 Diagramme de classes d'analyse

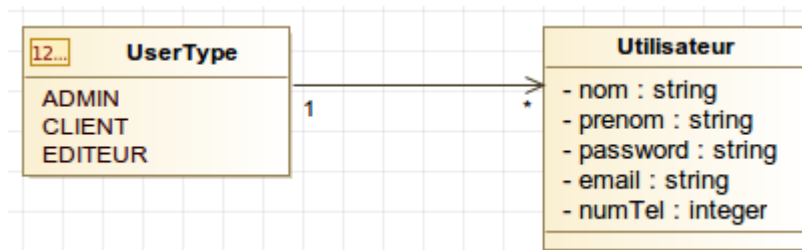


Figure 18: Diagramme classe du projet «Gestion des comptes client»

V.2.4.2 Diagrammes de séquence système (DSS)

Le diagramme de séquences permet de montrer les interactions entre les objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation, en mettant l'accent sur l'aspect temporel (la chronologie des envois de messages).

➤ **Diagramme de séquence du cas d'utilisation «Lister les comptes client»**



Figure 19: DSS du cas d'utilisation «Liste les comptes client»

➤ **Diagramme de séquence du cas d'utilisation «Créer compte client»**

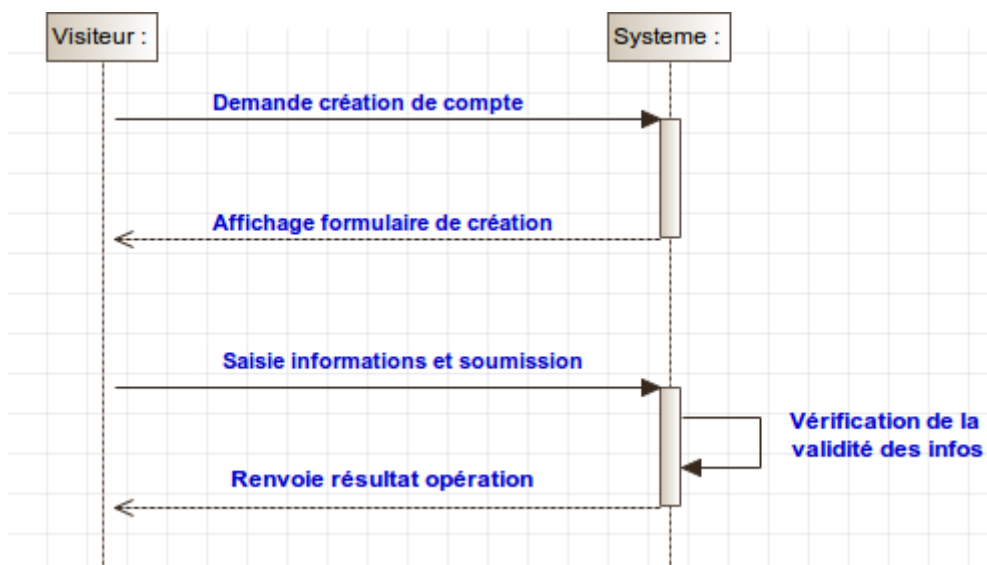


Figure 20: DSS du cas d'utilisation «Créer compte client»

➤ Diagramme de séquence du cas d'utilisation «Modifier un compte client»

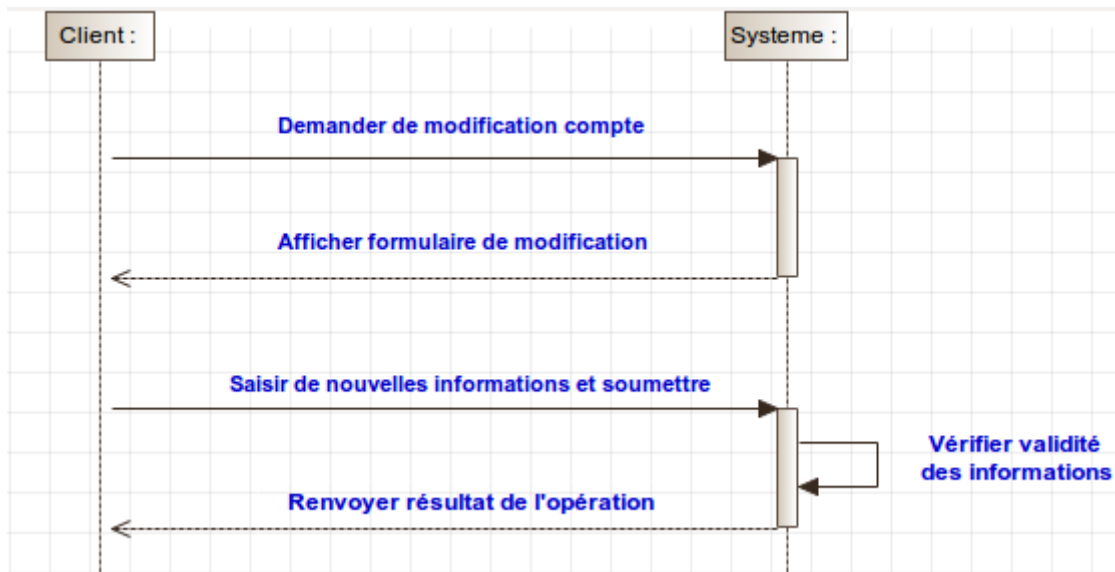


Figure 21: DSS du cas d'utilisation «Modifier un compte client»

➤ Diagramme de séquence du cas d'utilisation «Désactiver un compte client»

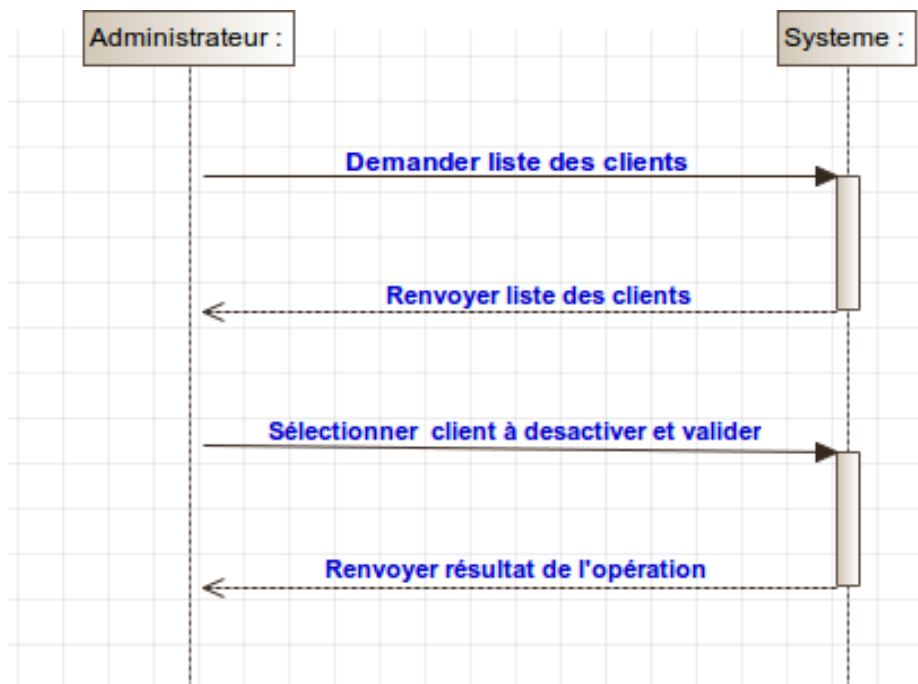


Figure 22: DSS du cas d'utilisation «Désactiver un compte client»

V.2.5 Conception préliminaire

Il est question de décrire le système vu comme une boîte blanche en le décomposant en ses différents composants de dialogues, contrôles et entités. De plus la description doit de se faire de façon générique en étant le plus éloigné possible des mécanisme liés à la plateforme d'exécution ou de programmation.

V.2.5.1 Diagramme de classes participantes

Le diagramme des classes participantes modélise trois types de classe d'analyse :

- ★ Les classes de type **dialogue**(« **boundary** ») qui permettent les interconnexions entre l'IHM et ses utilisateurs. Ce sont typiquement les écrans proposés à l'utilisateur(les formulaires de saisie, les résultats de recherche).
- ★ Les classes de type **métier**(« **entity** »): elles représentent les règles métier et proviennent directement du modèle du domaine mais sont confirmées et complétées pour chaque cas d'utilisation.
- ★ Les classes de type **contrôle**(« **control** »): elles contiennent la cinématique de l'application et font la transition entre les dialogues et les classes métier, en permettant aux écrans de manipuler des informations détenues par un ou plusieurs objets métier.

Nous allons ajouter aux entités, les dialogues et les contrôles, puis réaliser les diagrammes des classes participantes (DCP) des cas d'utilisation majeurs.

➤ DCP du cas d'utilisation «Créer un compte client»

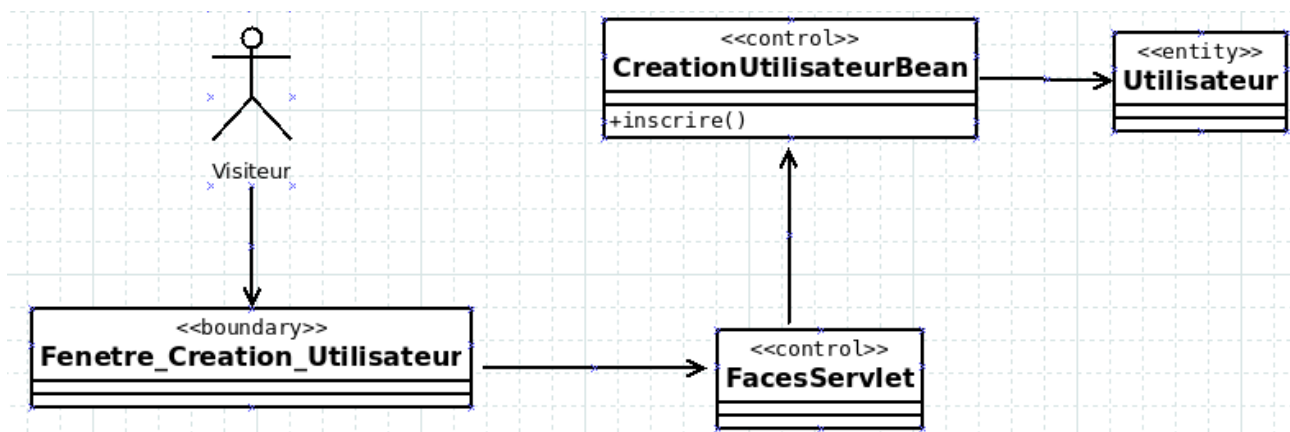


Figure 23: DCP du cas d'utilisation «Créer compte client»



➤ DCP du cas d'utilisation «Modifier un compte client»

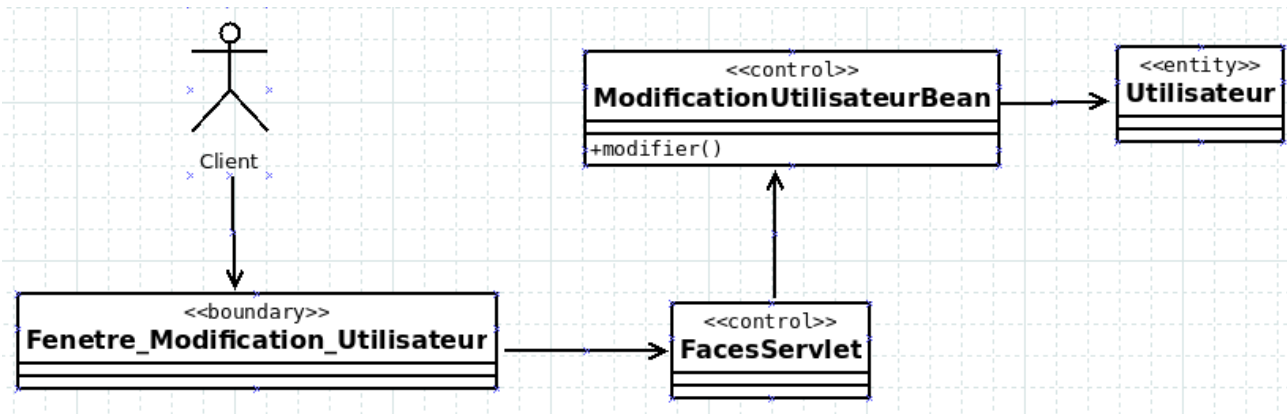


Figure 24: DCP du cas d'utilisation «Modifier compte client»

➤ DCP du cas d'utilisation «Désactiver un compte client»

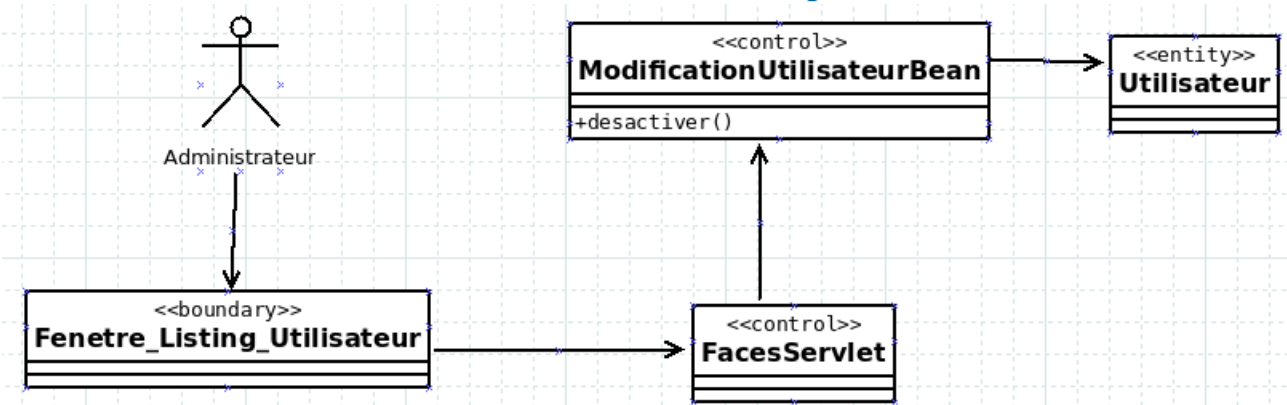


Figure 25: DCP du cas d'utilisation «Désactiver compte client»

➤ DCP du cas d'utilisation «Lister les clients»

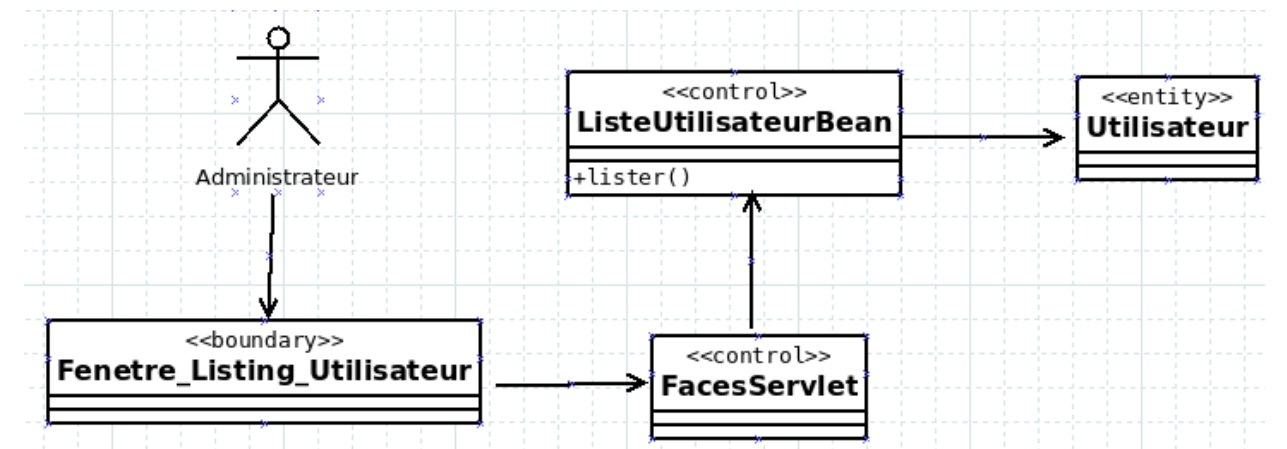


Figure 26: DCP du cas d'utilisation «Lister les comptes clients»

V.2.5.2 Diagrammes de séquence-objet préliminaire (DSOP)

Dans la suite de ce projet, nous présenterons uniquement les cas d'utilisation «Créer un compte client» et «Lister les comptes clients».

➤ DSOP du cas d'utilisation «Créer un compte client»

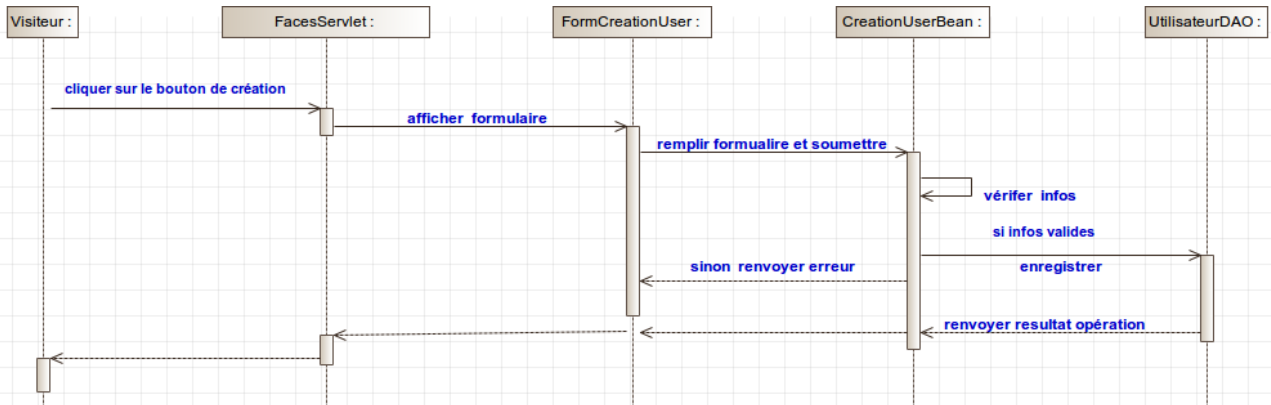


Figure 27: DSOP du cas d'utilisation «Créer compte client»

➤ DSOP du cas d'utilisation «Lister les comptes clients»

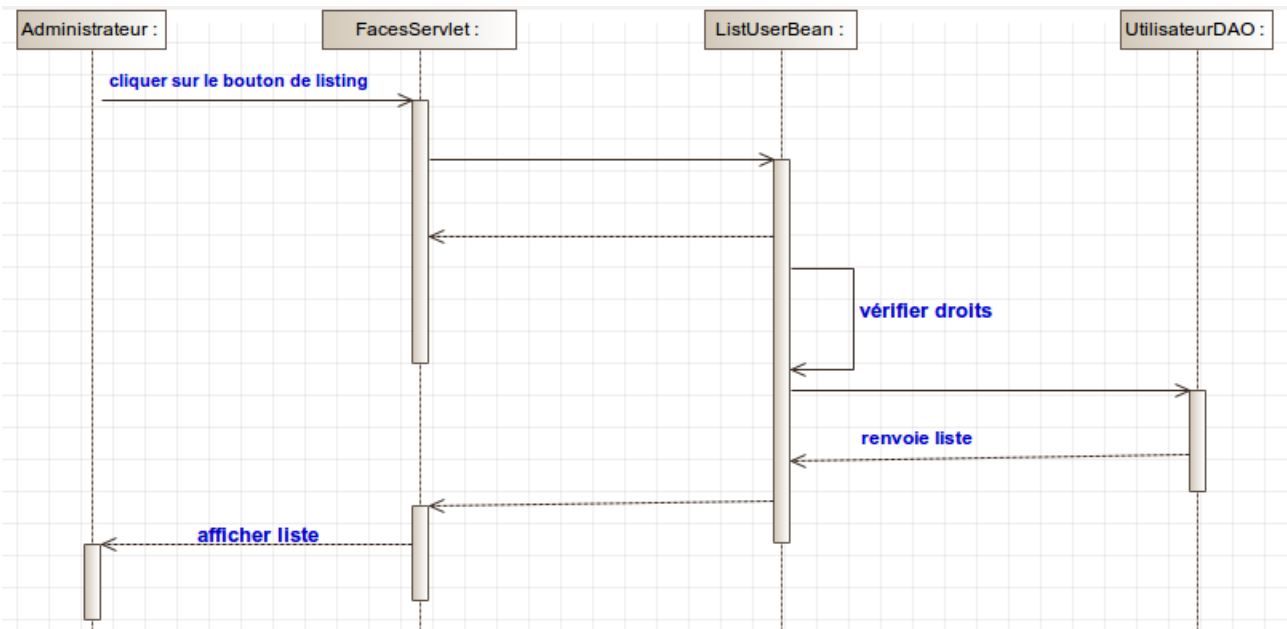


Figure 28: DSOP du cas d'utilisation «Lister les comptes clients»



V.2.6 Conception détaillée

Il s'agit ici de décrire de façon détaillée les fonctionnalités à implémenter en introduisant les éléments de conception (design pattern, interface) qui vont donner une image «prête à coder» de la solution.

V.2.6.1 Diagramme de classes détaillées (DCD)

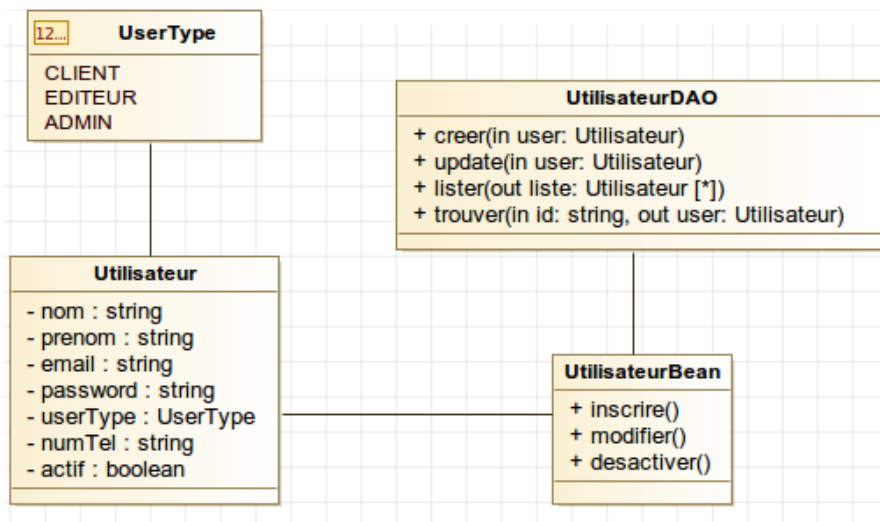


Figure 29: DCD du projet «gestion des comptes client»

V.2.6.2 Diagramme de séquences-objets détaillés (DSOD)

➤ DSOD du cas d'utilisation «Créer compte client»

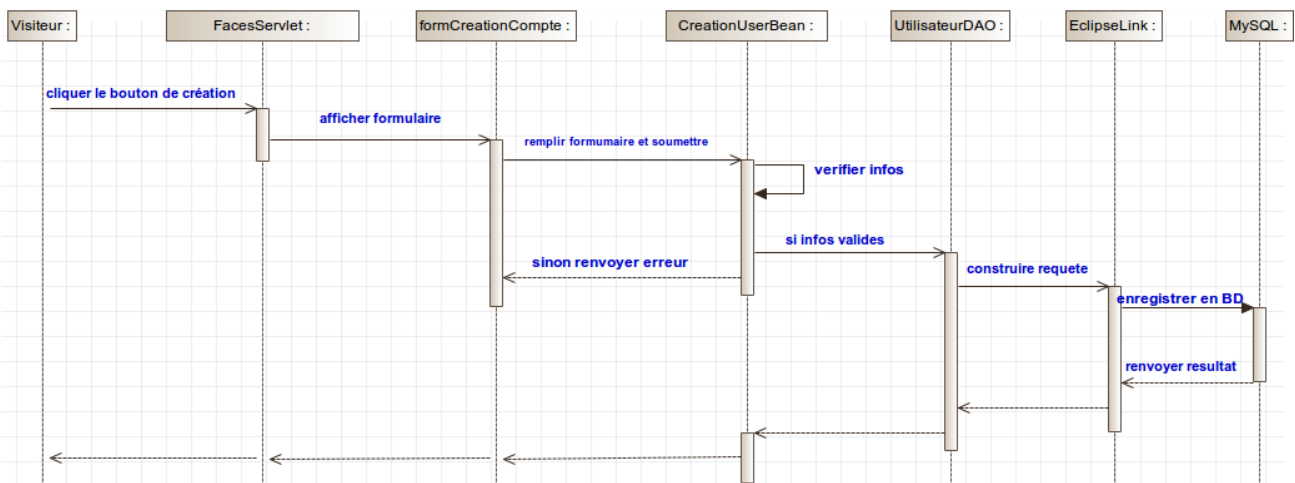


Figure 30: DSOD du cas d'utilisation «Créer compte client»



➤ DSOD du cas d'utilisation «Lister les comptes clients»

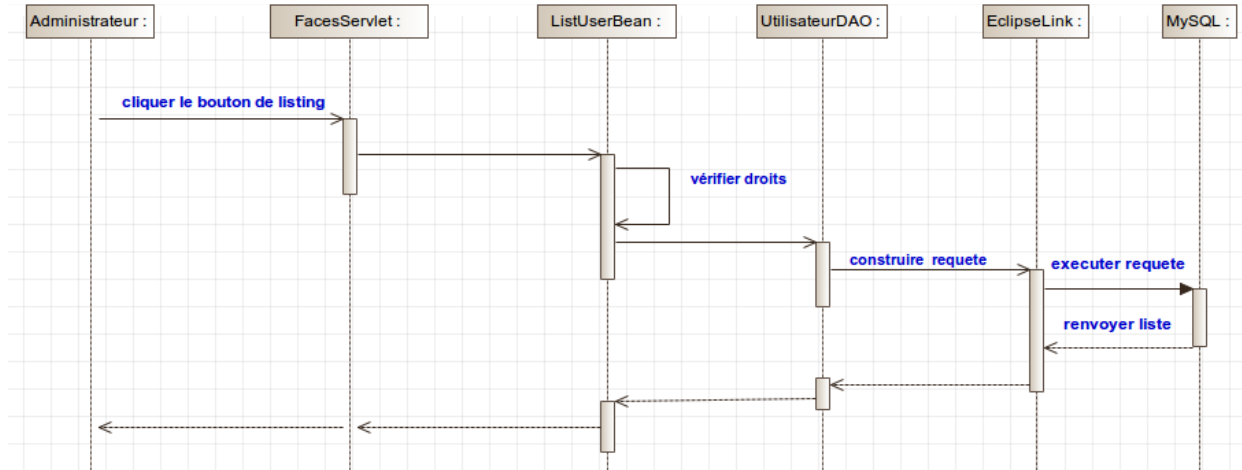


Figure 31: DSOD du cas d'utilisation «Lister les comptes clients»

V.2.7 Codage

Nous présentons les captures de l'implémentation de certaines classes.

```
1 package beans;
2
3 import java.io.IOException;
4
5 @Named
6 @ConversationScoped
7 public class CreationUserBean implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10    private final String MODULE = CreationUserBean.class.getName();
11    private Utilisateur user;
12    @EJB
13    private UtilisateurDao userDao;
14
15    public CreationUserBean() {
16        user = new Utilisateur();
17    }
18    public Utilisateur getUser() {
19        return user;
20    }
21    public void setUser(Utilisateur user) {
22        this.user = user;
23    }
24
25    public void inscrire() {
26        initialiserDateInscription();
27        user.setUserType(UserType.CLIENT.name());
28        String pass = user.getMotDePasse();
29        user.setMotDePasse(Utilitaire.chiffrer(pass));
30        try {
31
```

Figure 32: Implémentation de la classe «CreationUserBean»



```
1 package dao;
2
3 import java.util.ArrayList;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 @Stateless
20 public class UtilisateurDao {
21     private final String MODULE = UtilisateurDao.class.getName();
22     private static final String JPQL_SELECT_PAR_EMAIL = "SELECT u FROM Utilisateur u";
23     private static final String PARAM_EMAIL = "email";
24     // Injection du manager, qui s'occupe de la connexion avec la BDD
25     @PersistenceContext(unitName = "distdocs_PU")
26     private EntityManager em;
27
28     // Enregistrement d'un nouvel utilisateur
29     public void creer(Utilisateur utilisateur) throws DAOException{
30         try {
31             em.persist(utilisateur);
32         } catch (Exception e) {
33             Logger.getLogger(MODULE).log(Level.SEVERE, e.getMessage(), e);
34             e.printStackTrace();
35             throw new DAOException(e);
36         }
37     }
38
39     // Recherche d'un utilisateur à partir de son adresse email
40     public Utilisateur trouver(String email){
41         Utilisateur utilisateur = null;
42         Query requete = em.createQuery(JPQL_SELECT_PAR_EMAIL);
43         requete.setParameter(PARAM_EMAIL, email);
44         try {
45             utilisateur = (Utilisateur) requete.getSingleResult();
46         } catch (NoResultException e) {
47             return null;
48         } catch (Exception e) {
49             Logger.getLogger(MODULE).log(Level.SEVERE, e.getMessage(), e);
50             e.printStackTrace();
51         }
52     }
53 }
```

Figure 33: Implémentation de la classe «UtilisateurDao»

V3 Projet 3 : Acquisition des documents

V3.1 Étude préalable

V3.1.1 Projet à réaliser

Le but de ce projet est de mettre en œuvre les fonctionnalités liées à la gestion des acquisitions des documents.

V3.1.2 Recueil des besoins fonctionnels

Les principaux besoins fonctionnels sont :

- Acheter un document
- Rechercher un document
- Ouvrir un document
- Consulter les documents en vente

V3.1.3 Recueil des besoins opérationnels

- L'envoi des informations au serveur doit être sécurisé
- La recherche de document est une recherche plein texte



- L'ouverture d'un document doit se faire au sein de l'application
- Seuls les documents achetés par les clients peuvent être lu par ces derniers
- Les documents achetés ne doivent pas être partagé à des tiers
- Au sein de l'application mobile, le téléchargement des documents acquis doit être possible et les fichiers résultants doivent être protégés avant le stockage.
- La recherche peut se faire soit dans la bibliothèque personnelle de l'utilisateur(ensemble des documents acquis) ou à la page d'accueil dans l'ensemble des documents existants dans le système.

V3.1.4 Identification des acteurs

- Client
- Éditeur
- Administrateur
- Visiteur
- Passerelle de paiement

Ces acteurs doivent s'authentifier avant de réaliser les besoins ci-dessus.

V3.1.5 Identification des messages

Messages reçus par le système	Messages émis par le système
<ul style="list-style-type: none">• Soumettre une requête de recherche• Ouvrir un document	<ul style="list-style-type: none">• Liste des documents correspondant aux critères de la recherche• Afficher le contenu du document• Liste des documents achetés par un utilisateur

Tableau 9: Messages émis et reçu par le système dans le projet «Acquisition des documents»

V3.2 Capture des besoins fonctionnels

V3.2.1 Identification des cas d'utilisation

Cas d'utilisation	Acteur	Interface d'accès
Acheter un document	Client	Web et mobile
Consulter les documents achetés	Client	Web et mobile
Consulter les documents en vente	Visiteur, Client, Administrateur	Web et mobile
Ouvrir un document	Client	Web et mobile
Rechercher un document	Visiteur, Client,	Web et mobile

Tableau 10: Cas d'utilisation du projet «Acquisition des documents»

V3.2.2 Diagramme des cas d'utilisation

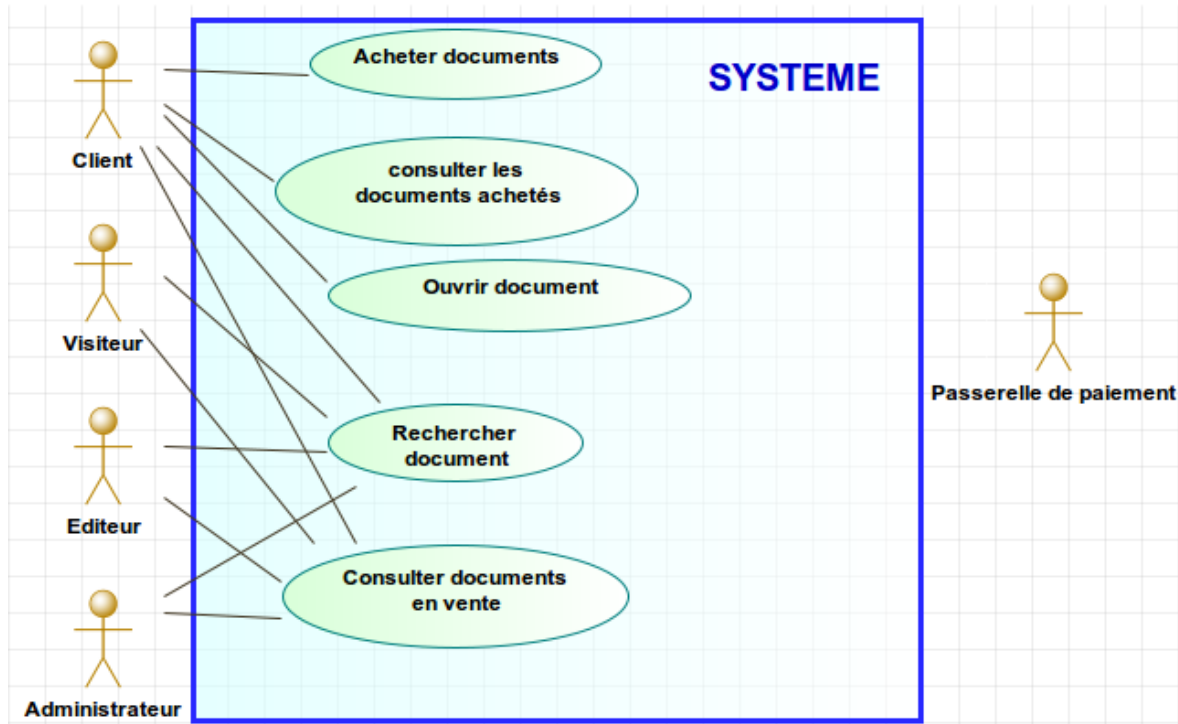


Figure 34: Diagramme des cas d'utilisation du projet «Acquisition des documents»

V3.3 Capture des besoins techniques

Les principaux besoins techniques sont :

- La protection des informations envoyées au serveur sera garanti par le protocole HTTPS.
- Au sein de l'application web, la lecture des documents acquis et leur protection seront assurées à l'aide de la bibliothèque FlipBook3D.
- Au sein de l'application mobile, la protection des documents téléchargés sera mis en œuvre à l'aide de l'algorithme AES fourni par Java.
- La recherche de documents se fera à l'aide des algorithmes implémentés dans la bibliothèque Lucene.
- Le paiement AIRTEL MONEY sera implémenté en priorité.



V3.4 Analyse

V3.4.1 Diagramme de classes d'analyse

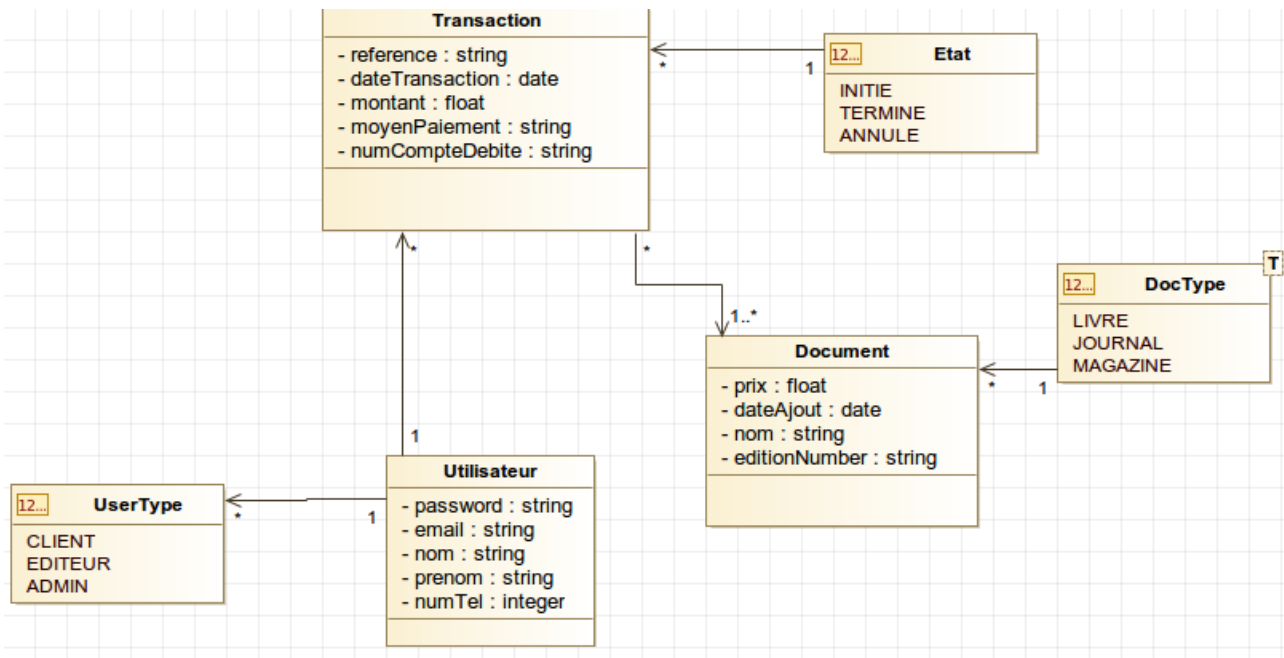


Figure 35: Diagramme de classes du projet «Acquisition de documents»

V3.4.2 Diagramme de séquence système (DSS)

➤ DSS du cas d'utilisation «Rechercher un document»

Cette recherche est celle qui se fait dans l'ensemble des documents du système.

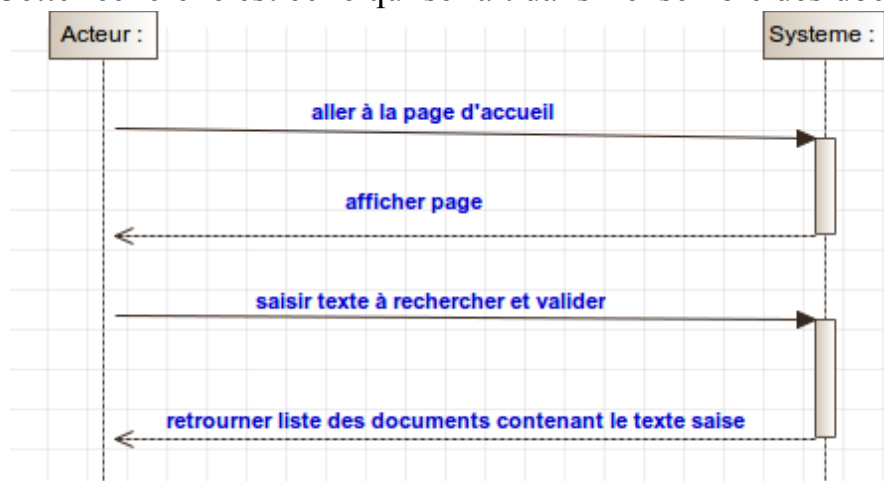


Figure 36: DSS du cas d'utilisation «Rechercher un document»



➤ DSS du cas d'utilisation «Acheter un document»

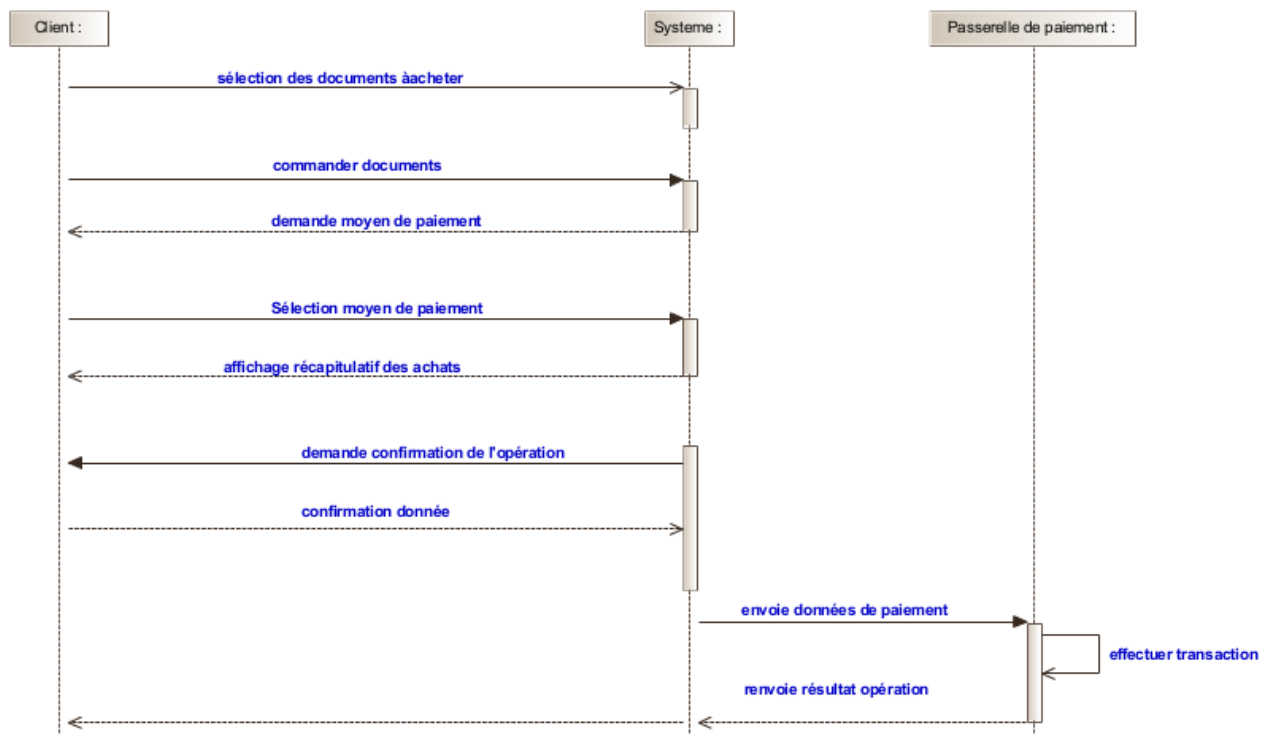


Figure 37: DSS du cas d'utilisation «Acheter un document»

➤ DSS du cas d'utilisation «Ouvrir un document»

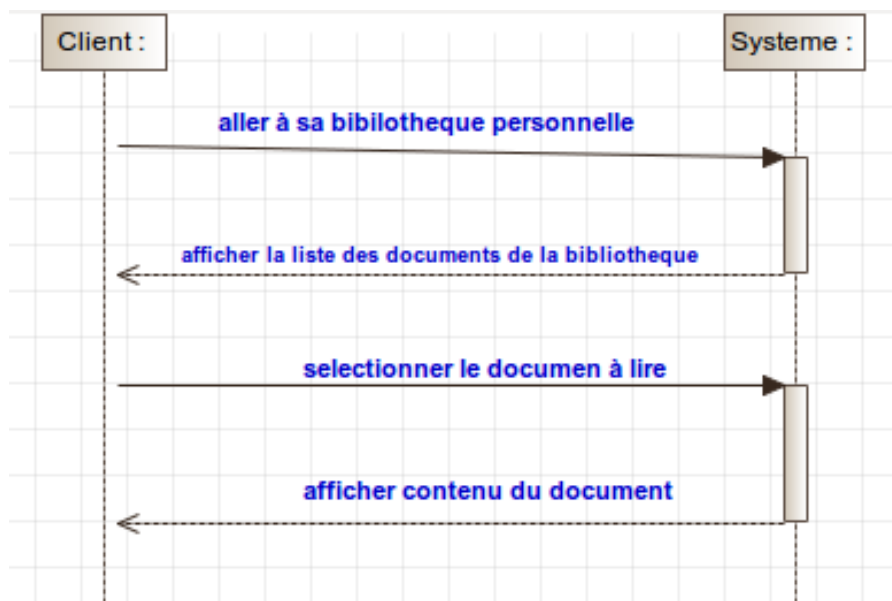


Figure 38: DSS du cas d'utilisation «Ouvrir un document»



V3.5 Conception préliminaire

V3.5.1 Diagramme de classes participantes (DCP)

➤ DCP du cas d'utilisation «Ouvrir un document»

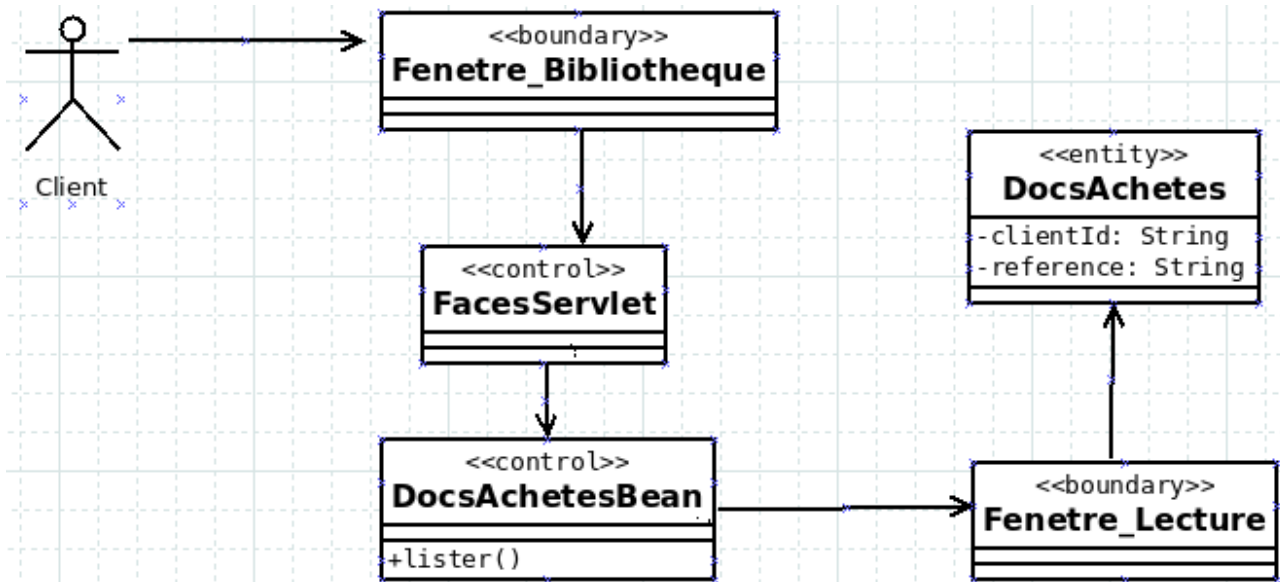


Figure 39: DCP du cas d'utilisation «Ouvrir un document»

➤ DCP du cas d'utilisation «Acheter un document»

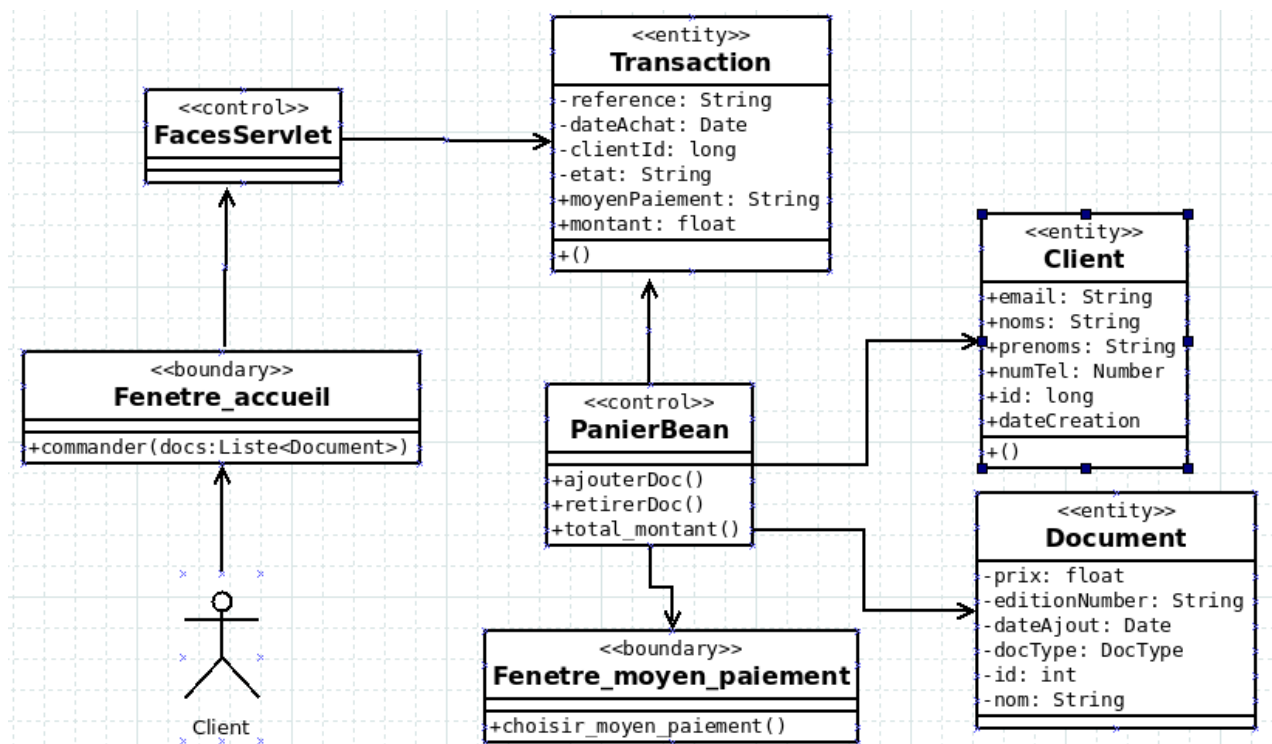


Figure 40: DCP du cas d'utilisation «Acheter un document»



➤ DCP du cas d'utilisation «Rechercher un document»

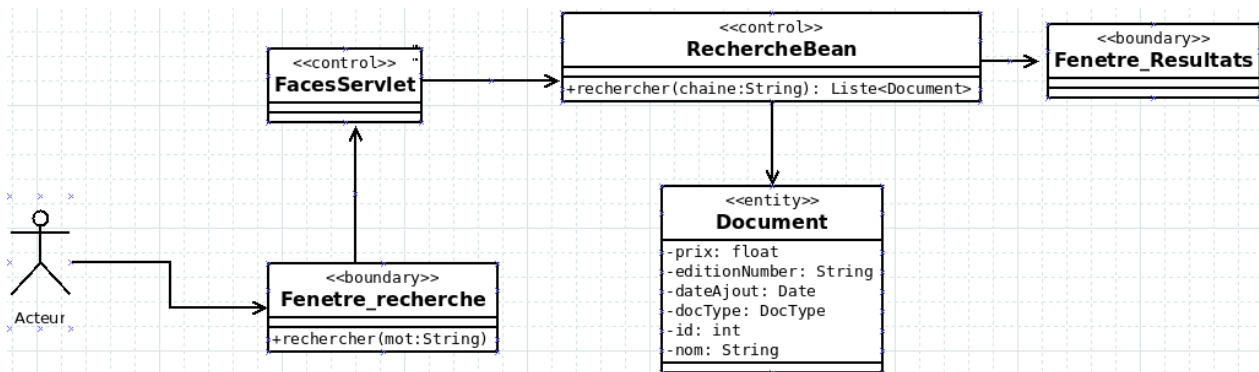


Figure 41: DCP du cas d'utilisation «Rechercher un document»

V3.5.2 Diagrammes de séquence-objet préliminaire(DSOP)

➤ DSOP du cas d'utilisation «Ouvrir un document»

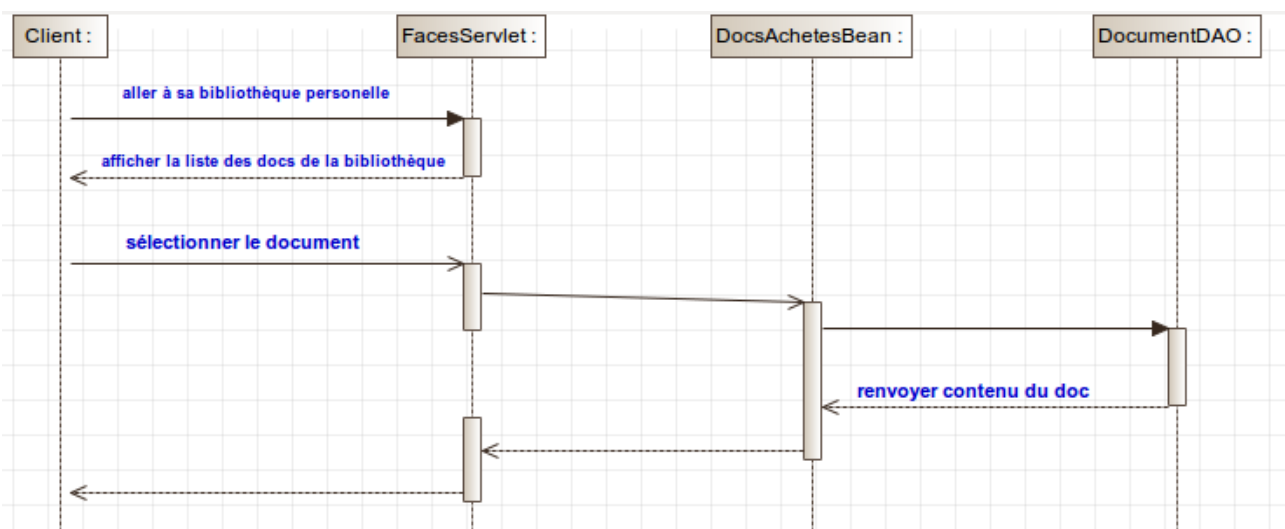


Figure 42: DSOP du cas d'utilisation «Ouvrir un document»



➤ DSOP du cas d'utilisation «Rechercher un document»

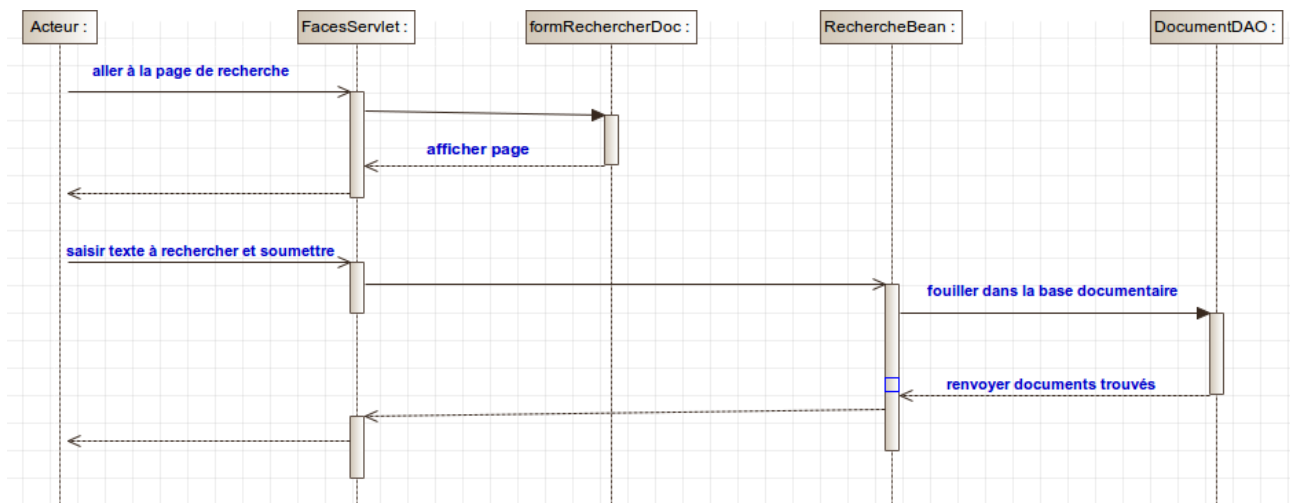


Figure 43: DSOP du cas d'utilisation «Rechercher un document»

➤ DSOP du cas d'utilisation «Acheter un document»

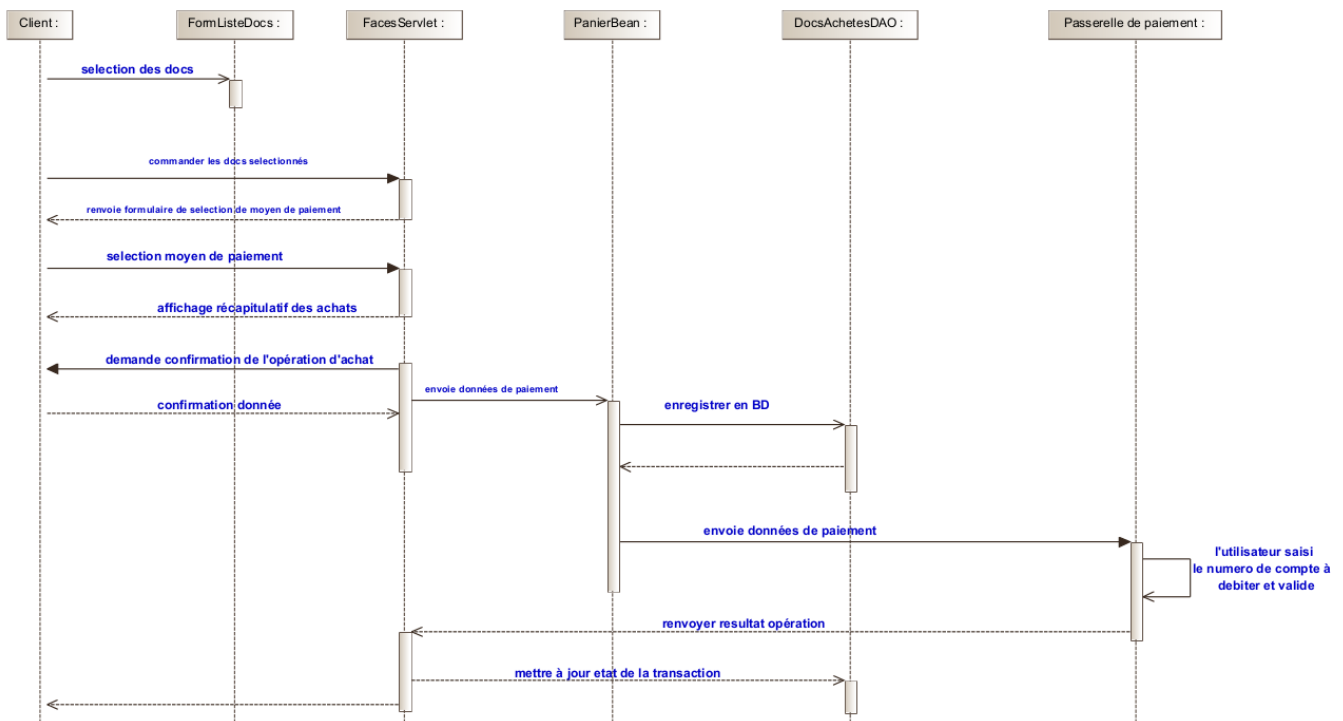


Figure 44: DSOP du cas d'utilisation «Acheter un document»



V3.6 Conception détaillée

V3.6.1 Diagramme de classes détaillées (DCD)

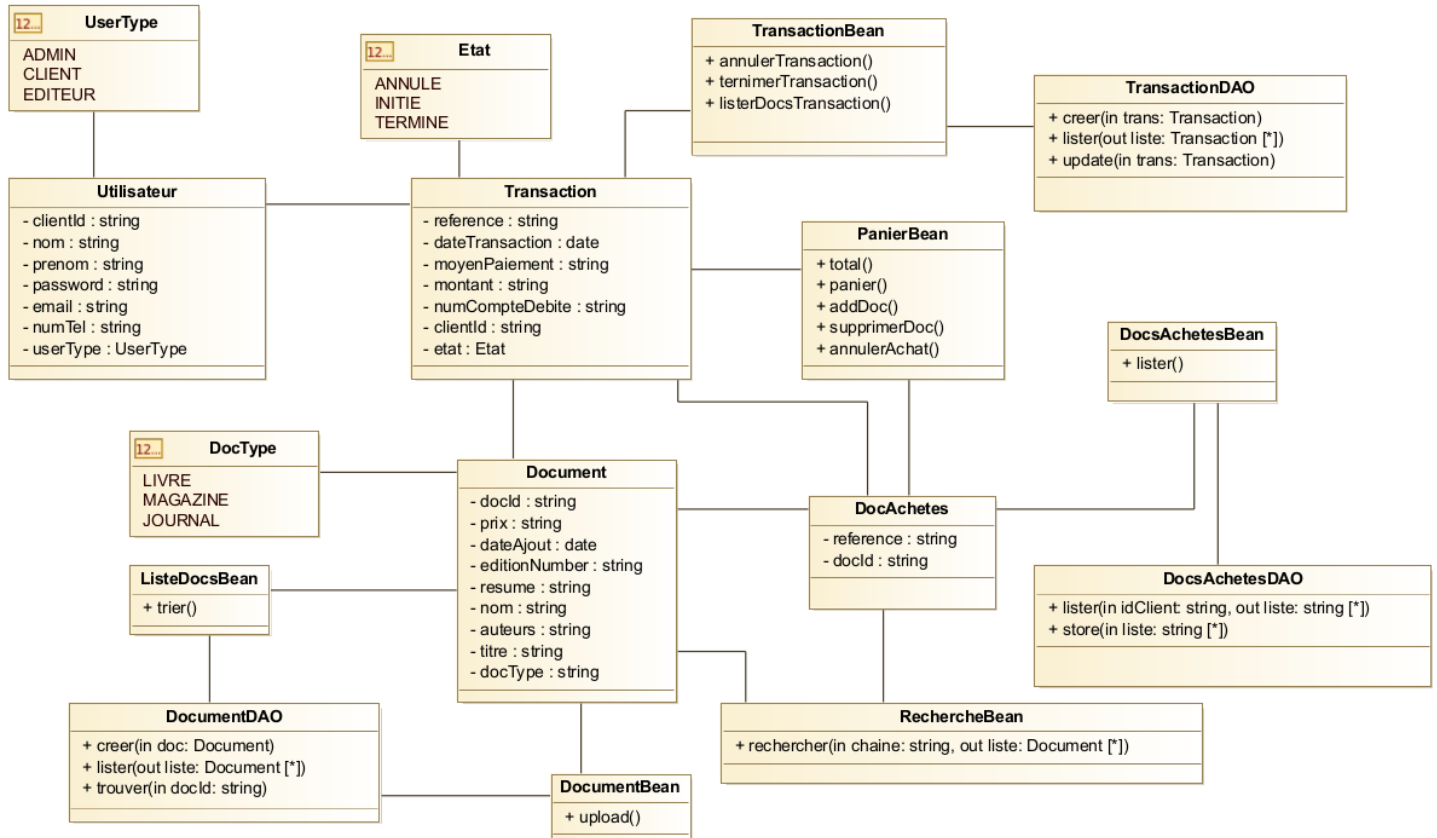


Figure 45: Diagramme de classes détaillées du projet «Acquisition des documents»

V3.6.2 Diagrammes de séquence-objet détaillés (DSOD)

➤ DSOD du cas d'utilisation «Ouvrir un document»

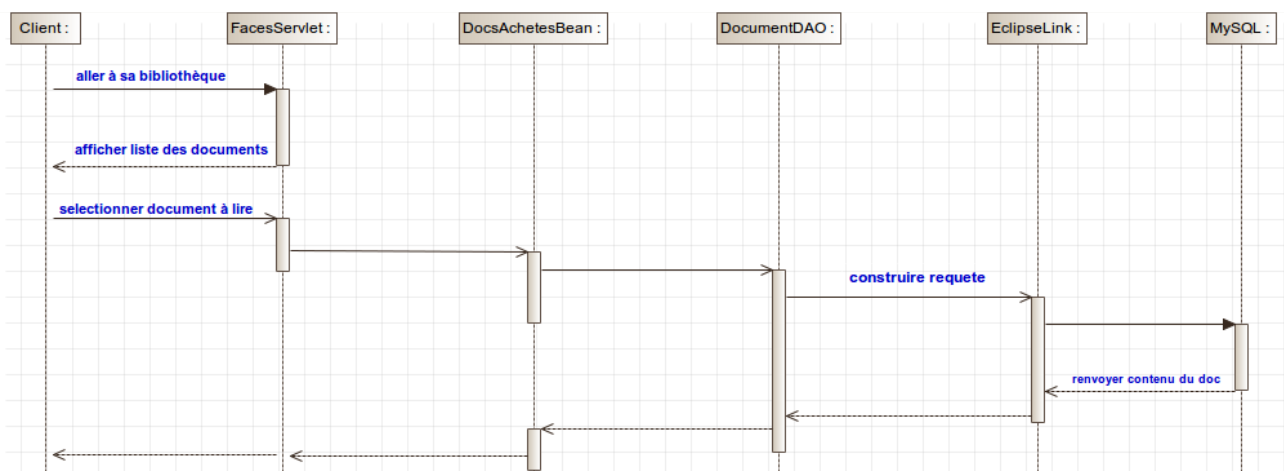


Figure 46: DSOD du cas d'utilisation «Ouvrir un document»



➤ DSOD du cas d'utilisation «Rechercher un document»

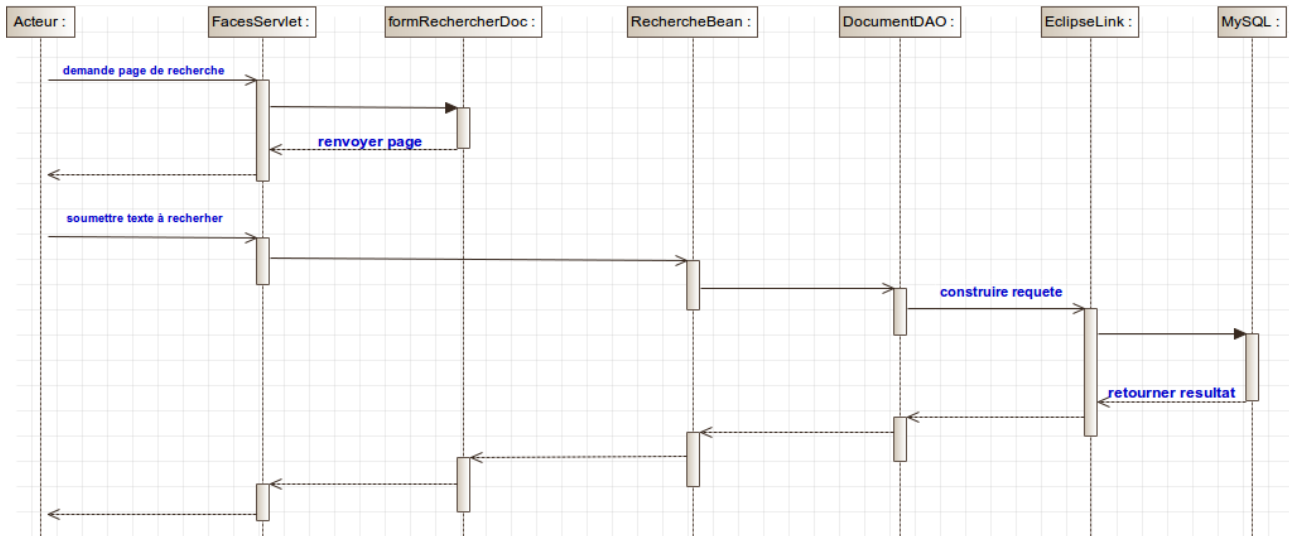


Figure 47: DSOD du cas d'utilisation «Rechercher un document»

V3.7 Codage

```
1 package beans;
2
3 import java.io.Serializable;
4
5 @Named
6 @SessionScoped
7 public class PanierBean implements Serializable{
8
9     private static final long serialVersionUID = 1L;
10    private final String MODULE = PanierBean.class.getName();
11    //liste des articles du panier
12    private List<Document> articles;
13    //document sélectionné par le client
14    private Document selectedDoc;
15    //moyen de paiement sélectionné par le client
16    private String moyenPaiement;
17    //lien où airtel va rediriger la réponse
18    private String redirect;
19    //tel marchand où l'argent sera versé
20    private String telMarchand;
21    //reference de la transaction maxi 13 caractères
22    private String reference;
23    //montant total des articles dans le panier
24    private float total;
25
26    @EJB
27    private DocsAchetesDao dao;
28
29    @EJB
30    private TransactionDao transDao;
31    private long clientId;
32
33    @PostConstruct
34    public void init() {
35        selectedDoc = new Document();
36        articles = new ArrayList<>();
37        redirect = Constante.url_redirect;
38        telMarchand = Constante.tel_marchand;
39    }
40 }
```

Figure 48: Implémentation de la classe «PanierBean»



```
1 package entities;
2
3 import java.sql.Timestamp;
4
5
6
7
8
9
10 @Entity
11 @Table(name="Transaction")
12 public class Transaction {
13
14     @Id
15     private String reference;
16     @Column(name="date_achat")
17     private Timestamp dateAchat;
18     @Column(name="last_update")
19     private Timestamp lastUpdate;
20     @Column(name="tel_client")
21     private String telClient;
22     @Column(name="moyen_paiement")
23     private String moyenPaielement;
24     @Column(name="client_id")
25     private long clientId;
26     private String etat;
27     private float montant;
28
29     public String toString() {
30         return "ref = "+reference+" tel = "+telClient+" moyen = "+moyenPaielement
31             +" etat = "+etat+" montant = "+montant;
32     }
33     public Timestamp getDateAchat() {
34         return dateAchat;
35     }
36     public void setDateAchat(Timestamp dateAchat) {
37         this.dateAchat = dateAchat;
38     }
39     public String getMoyenPaielement() {
40         return moyenPaielement;
41     }
42     public void setMoyenPaielement(String moyenPaielement) {
```

Figure 49: Implémentation de la classe «Transaction»

Conclusion

Ce chapitre met un terme à cette partie dans laquelle nous avons présenté l'existant, choisi notre méthode de développement et procéder à l'analyse des besoins et à la mise au point des différents modèles de notre système. La partie suivante est consacrée à la présentation de la solution obtenue et le suivi du projet.



Partie III : FINALISATION DE LA SOLUTION

APOSTILLE

Dans cette partie, nous présentons notre solution dans son entièreté et les éléments nécessaires à son déploiement. Nous terminerons par des commentaires sur la conduite du projet.

CHAPITRES

CHAPITRE VI: DÉPLOIEMENT ET RÉSULTATS

CHAPITRE VII: CONDUITE DE PROJET



Chapitre VI: DÉPLOIEMENT ET RÉSULTATS

Introduction

Dans ce chapitre, il sera essentiellement question de présenter les résultats de notre travail.

VI.1 Diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

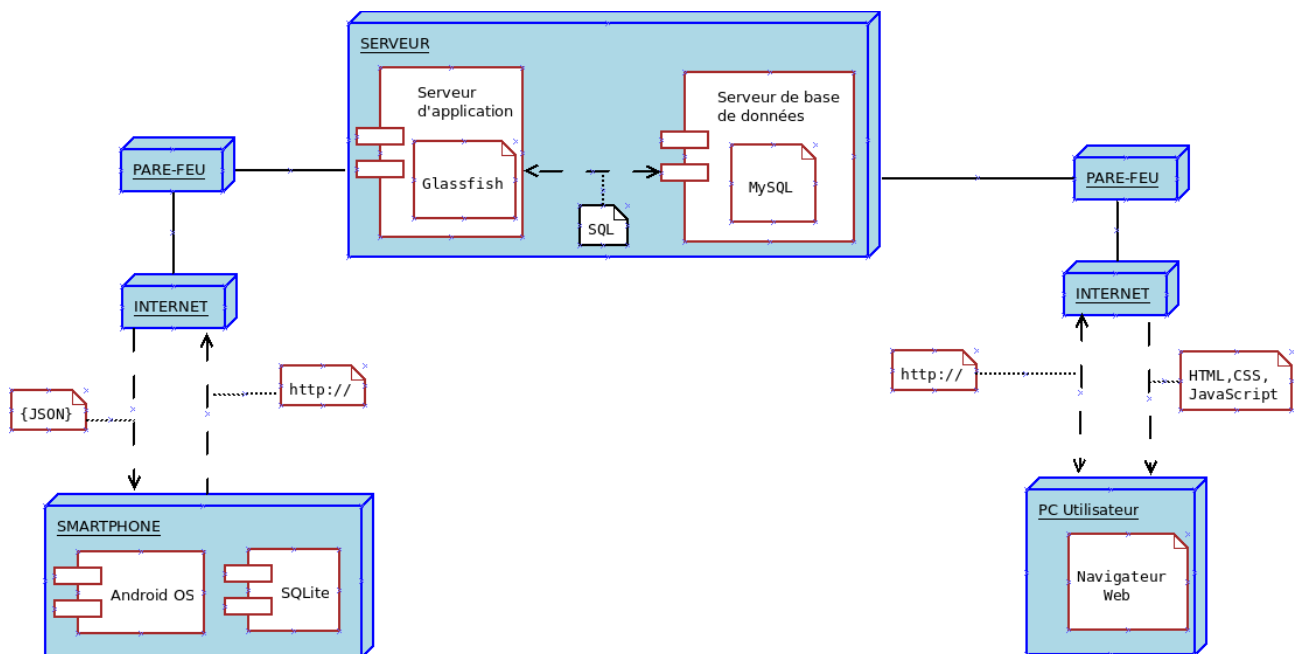


Figure 50: Diagramme de déploiement de la solution



VI.2 Résultats obtenus

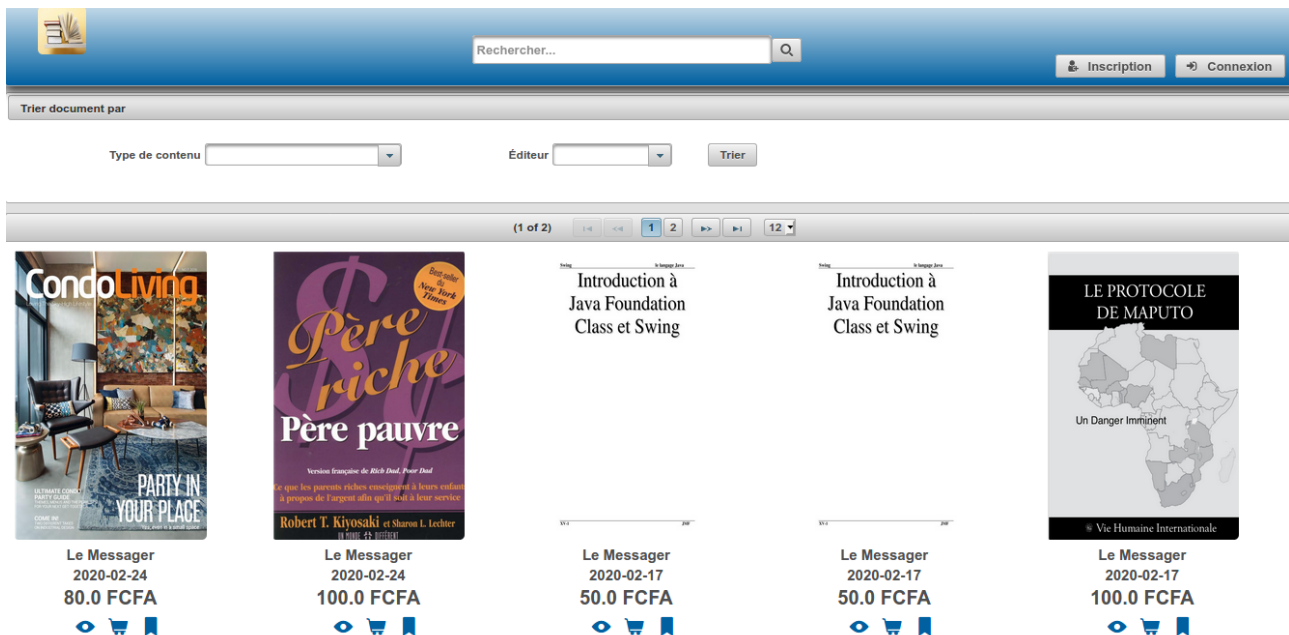


Figure 51: Fenêtre d'accueil application web

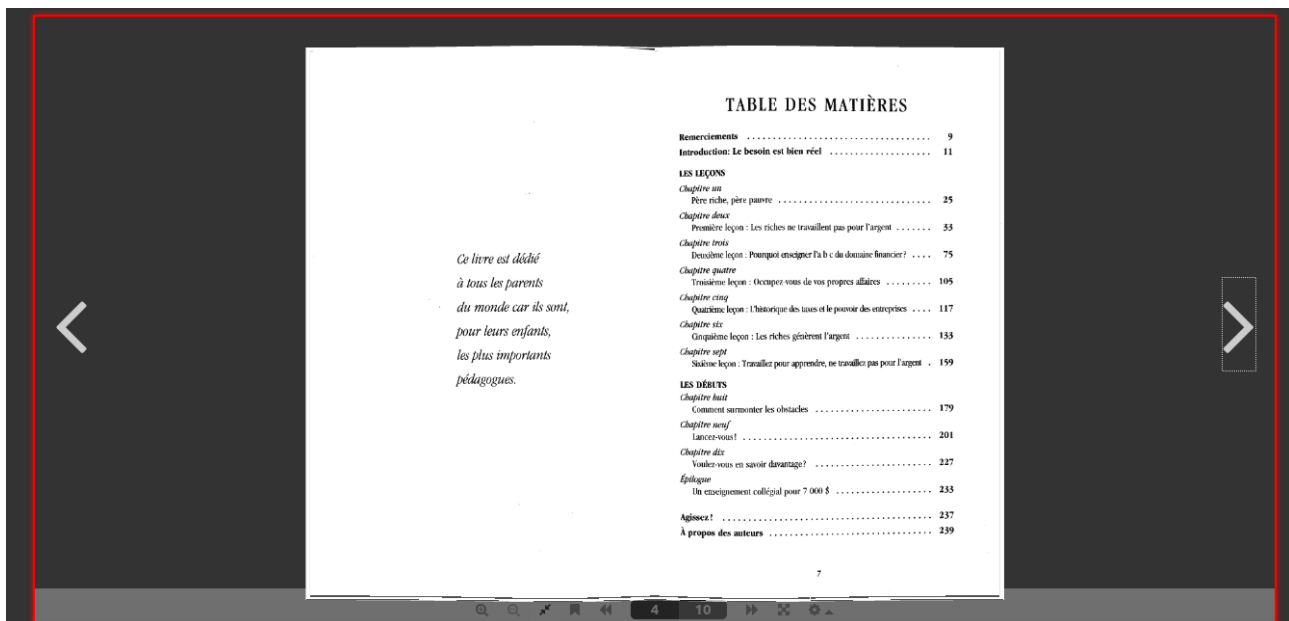


Figure 52: Fenêtre de lecture de PDF au sein de l'application web

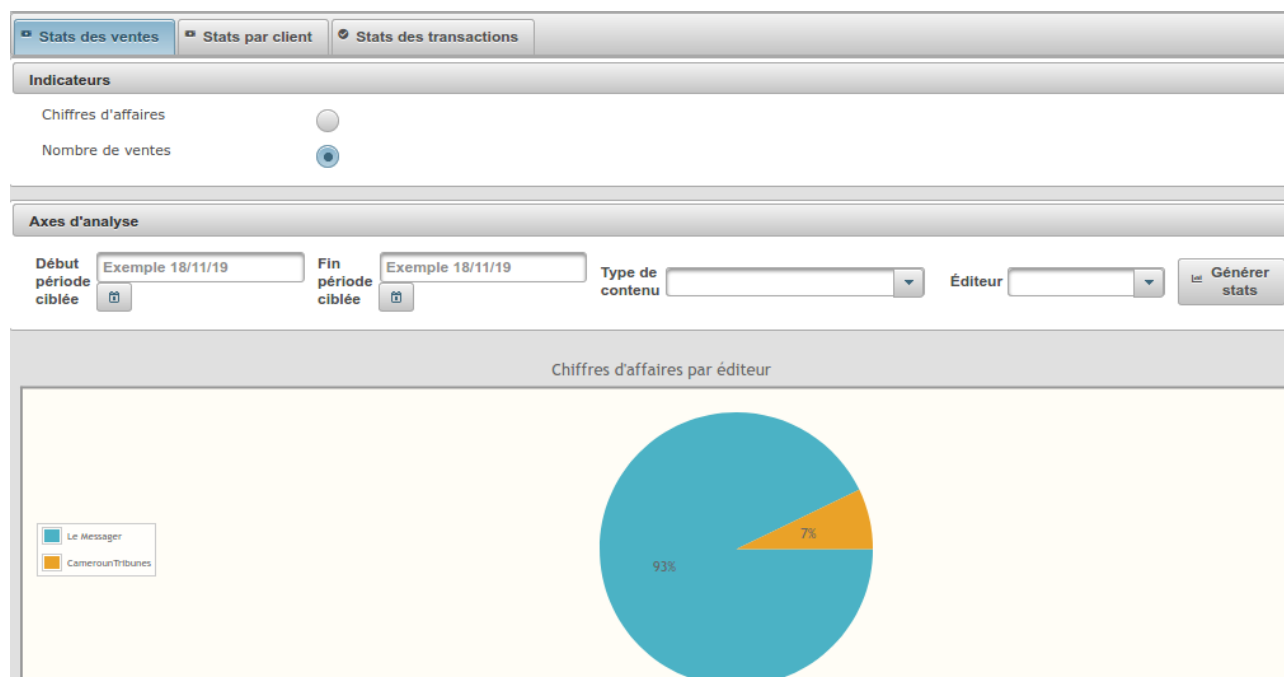


Figure 53: Fenêtre de statistiques de ventes - diagramme circulaire

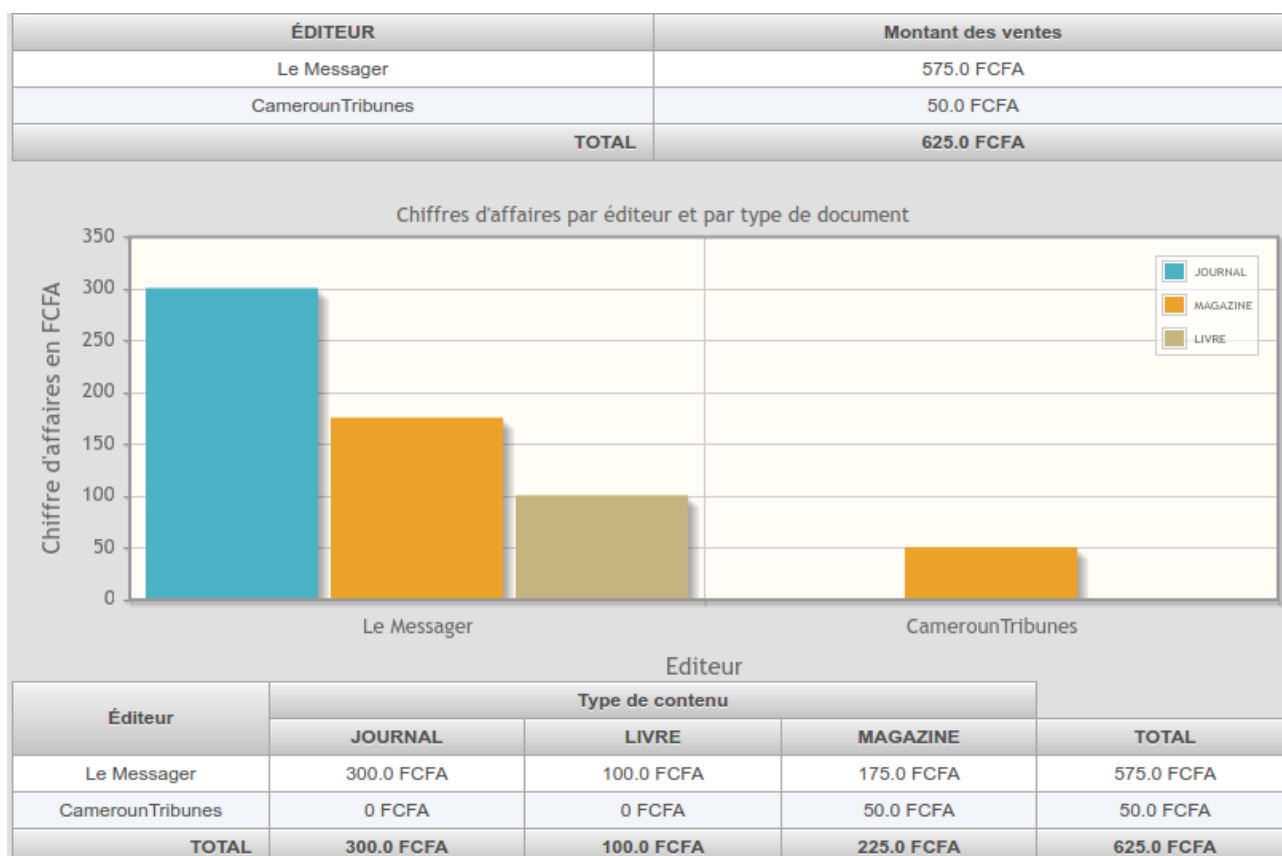


Figure 54: Fenêtre de statistiques de ventes - diagramme à bandes



Conclusion

Dans ce chapitre, nous avons présenté l'architecture de notre solution et quelques captures montrant les résultats obtenus. Le chapitre suivant sera consacré au déroulement du projet et les ressources utilisées.



Chapitre VII: CONDUITE DE PROJET

Introduction

Les caractéristiques communes à chaque projet sont les contraintes de délai, de coût et de ressources. Dans ce chapitre, nous présenterons l'ensemble des moyens humains et matériels mis à contribution pour atteindre nos objectifs dans le temps qui nous était imparti. Nous ferons un bilan de notre travail.

VII.1 Diagrammes de planification du projet

Le projet a débuté le 09 février 2019 et devrait s'achever le 10-02-2020. Mais il a réellement pris fin le 30 mars 2020.

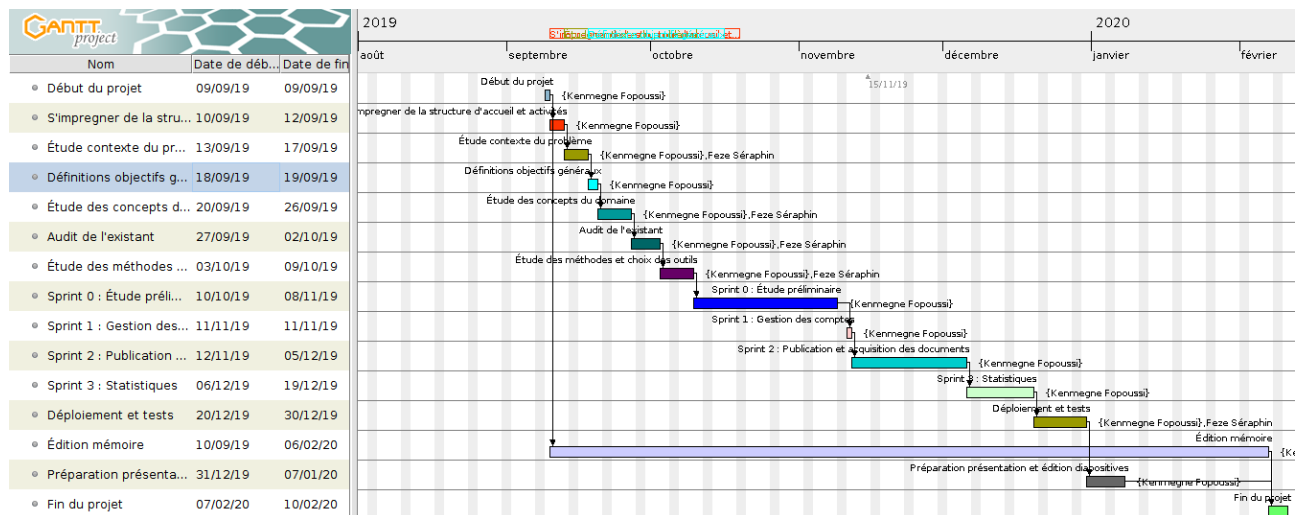


Figure 55: Diagramme de Gantt prévisionnel

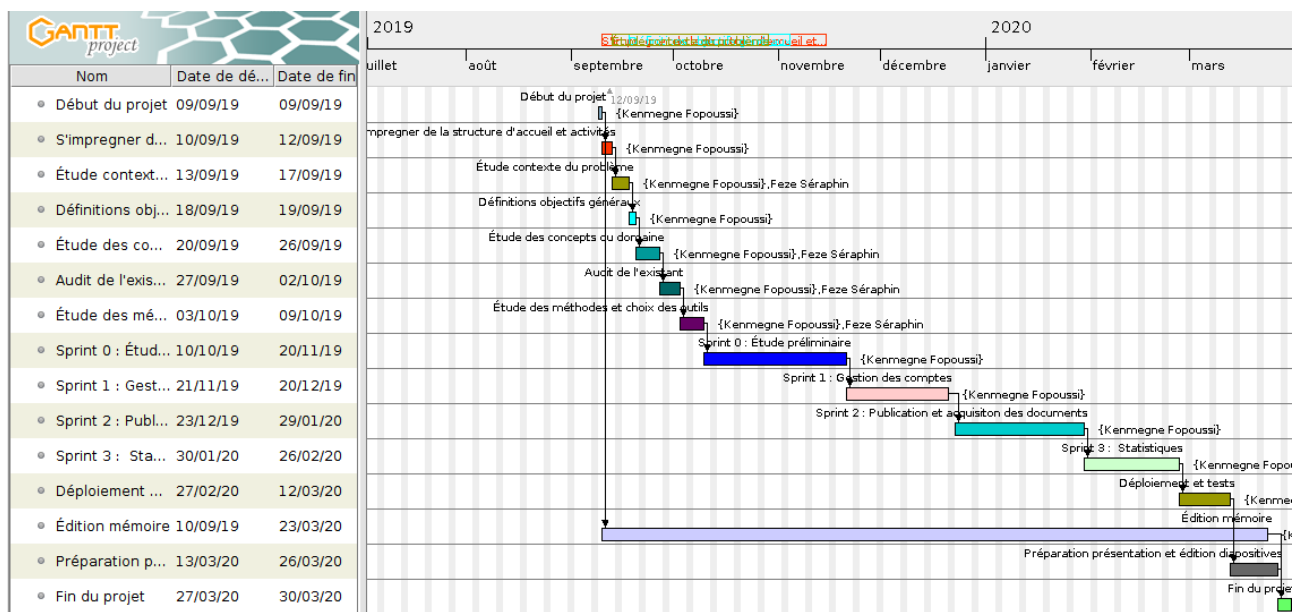


Figure 56: Diagramme de Gannt réel

VII.2 Évaluation du coût du projet

VII.2.1 Évaluation réelle

Points de dépense	Description	Quantité	Prix unitaire (FCFA)	Prix total (FCFA)
Ressources humaines	Ingénieur stagiaire en informatique	1	50 000/mois pendant 5 mois	250 000
Logiciels	Linux(Ubuntu), Eclipse, MySQL, Glassfish, FlipBook3D		0	0
Machine de développement	Laptop Acer Intel Core i3-6100U CPU 2.3 GHz	1	0	0
Téléphone(pour les tests de la partie mobile)	Téléphone portable Tecno LC6, Quad cor 2.0 GHz, Android 9	1	0	0
TOTAL				250 000 FCFA

Tableau 11: Estimation des coûts et charges du projet en tant que stagiaire



VII.2.2 Évaluation nominale

Points de dépense	Description	Quantité	Prix unitaire (FCFA)	Prix total (FCFA)
Ressources humaines	Ingénieur en informatique	1	5000fcfa/heure pour 8 heures de travail par jour	800000/mois pendant 5 mois = 4 000 000
Logiciels	Linux(Ubuntu), Eclipse, MySQL, Glassfish		0	0
Machine de développement	Laptop Acer Intel Core i3-6100U CPU 2.3 GHz	1	400 000	400 000
Téléphone(pour les tests de la partie mobile)	Téléphone portable Tecno LC6, Quad cor 2.0 GHz, Android 9	1	100 000	100 000
FlipBook3D	Framework permettant de créer des livres en retournant des pages 3D à partir de brochures PDF	1	60 000	60 000
Frais annuel d'hébergement	Déploiement	12	20 000 / mois	240 000
Frais Play Store	Déploiement	1	20 000	20 000
TOTAL				4 820 000 FCFA

Tableau 12: Coût pour un particulier voulant se procurer notre solution

VII.3 Bilan et perspectives

VII.3.1 Bilan

VII.3.1.1 Apports

Notre stage au sein de Métrika IDB nous a permis de côtoyer des ingénieurs et experts hautement qualifiés, possédant plusieurs années d'expérience professionnelle dans le domaine. Ce fut l'endroit idéal pour asseoir les connaissances apprises



pendant la formation et le lieu idoine pour effectuer un travail à la hauteur du titre d'ingénieur auquel nous prétendons.

Ce stage nous a permis :

- ✓ De nous familiariser avec le monde professionnel;
- ✓ D'explorer l'univers assez vaste et prometteur du développement mobile et des paiements électroniques.
- ✓ D'assister à la gestion des projets d'envergure et de nous rapprocher des experts du métier.
- ✓ De compléter nos connaissances sur la conception, le langage Java et bien d'autres outils et de techniques de programmation. Nous avons aussi améliorer notre approche commerciale en ce qui est des produits logiciels.

VII.3.1.2 Difficultés

Nous avons fait face à des difficultés au long de notre stage au rang desquels :

- ✓ Mise en pratique des connaissances théoriques apprises en cours.
- ✓ Implémentation du paiement électronique par Airtel Money : ce procédé était nouveau pour nous et a nécessité plusieurs essais pour arriver un résultat probant.

VII.3.2 Perspectives

Pour la suite de nos travaux, nous envisageons implémenter la notion de fidélisation des clients(abonnements), d'améliorer l'aspect sécurité de nos deux modules et d'intégrer d'autres moyens de paiement à notre solution.

Conclusion

Par ce chapitre dédié au suivi du projet, nous mettons un terme à troisième et dernière partie de notre travail.



CONCLUSION GÉNÉRALE

Tout au long de ce stage, notre objectif était de mettre sur pied une plateforme web et mobile(Android) permettant la collecte, puis la distribution des documents électroniques. Grâce à notre solution, l'éditeur peut publier ses contenus, visualiser ses statistiques de vente. Le client peut acquérir des documents par paiement mobile(Airtel money), les lire et les télécharger, tout ceci dans le respect des droits d'auteur.

Pour arriver à cette solution, nous sommes passés par diverses étapes qui allaient de la présentation du contexte du stage au déploiement de la solution ; en passant par l'étude des principaux concepts du domaine (document numérique, paiement mobile, droits d'auteur, etc), l'élaboration du cahier de charge, le choix de la méthode de développement (2TUP, UML), etc.

Après finalisation de notre plateforme web et mobile, nous estimons que l'objectif visé par notre travail a été atteint. En effet, l'application mobile que nous avons mise en œuvre permettra aux clients de télécharger leurs documents et les lire offline. Cependant, deux perspectives pour la suite de nos travaux seraient celles de : fidéliser les clients, en intégrant un système d'abonnement à notre solution afin de leurs proposer de recevoir périodiquement et automatiquement un produit ; étendre le module mobile à d'autres systèmes d'exploitation(Windows,iOS) ;



RÉFÉRENCES BIBLIOGRAPHIQUES

- [B1] Hassan Diab «Évaluation des méthodes formelles de spécification», mémoire présenté au Département de mathématiques et d'informatique en vue de l'obtention du grade de maître ès sciences, p 10-11.
- [B2] Sridevi Addagada «Indexing and Searching Document Collections using Lucene», University of New Orleans Theses and Dissertations, p15-16, 2007.
- [B3] Pascal Roques, «UML 2 par la pratique : Études de cas et exercices corrigés», 5e édition.
- [B4] Pascal Roques, «UML 2 : Modéliser une application web, 2e édition».
- [B5] Pascal Roques, Franck Vallée, «UML 2 en action : De l'analyse des besoins à la conception», 4e édition.
- [B6] Christian Soutou , «UML 2 pour les bases de données».
- [B7] Dr Ange Nambila, «Cours de gestion de projet» ; IAI 2019 ;
- [B8] Pr Souleymane KOUSSOUBE, «Cours de Conception Orienté Objet» ; IAI 2017.
- [B9] BakoAI, «Intégration de l'API REST de PVit Airtel Money».



WEBOGRAPHIE

- [S1] https://fr.wikipedia.org/wiki/Document_num%C3%A9rique#Supports_de_diffusion, le 15-09-2019 : Document numérique.
- [S2] <https://www.cairn.info/revue-les-enjeux-de-l-information-et-de-la-communication-2002-1-page-2.htm#>, le 18-09-2019 : Indexation des documents.
- [S3] http://www1.rfi.fr/actufr/articles/075/article_42240.asp, le 22-09-2019 : Boom des kiosques numériques.
- [S4] <https://www.agenceecofin.com/mobile/1202-63922-gabon-3-millions-d-abonnes-mobile-et-1-million-d-utilisateurs-d-internet-a-ce-jour>, le 30-09-2019 : Statistiques de l'ANINF sur Internet et la téléphonie mobile.
- [S5] <https://fr.mobiletransaction.org/types-paiement-mobile/>, le 25-10-2019 : Différents types de paiements mobiles.
- [S6] [https://fr.wikiversity.org/wiki/Paiement_mobile_\(m-paiement\)/Les_modes_de_paiement_et_leurs_technologies](https://fr.wikiversity.org/wiki/Paiement_mobile_(m-paiement)/Les_modes_de_paiement_et_leurs_technologies), le 27-10-2019 : Différents types de paiements mobiles.
- [S7] <http://paiement-mobile.info/technologies/qr-code>, le 27-10-2019 : Paiement mobile.
- [S8] https://fr.wikipedia.org/wiki/Distribution_num%C3%A9rique, le 28-10-2019 : Distribution numérique.
- [S9] <http://www.ordinateur.cc/programmation/Computer-Programming-Languages/87470.html>, le 28-11-2019 : Programmation orientée-objet.
- [S10] https://www.memoireonline.com/10/10/4000/m_Etude-et-mise-en-place-dun-systeme-informatise-de-transfert-dargent-inter-agences-COMECI4.html, le 28-11-2019 : Méthodes orientées-objet.
- [S11] <https://www.supinfo.com/articles/single/6765-methodologies-developpement-logiciel>, le 29-11-2019 : Méthodologies de développement logiciel.
- [S12] <https://fr.gadget-info.com/difference-between-informed>, le 06-12-2019 : Différence entre recherche informée et recherche non-informée.
- [S13] http://turing.cs.pub.ro/auf2/html/chapters/chapter3/chapter_3_2_1.html, le 06-12-2019 : Types de recherche.
- [S14] <https://www.softfluent.fr/blog/architecture-logicielle-pour-application/>, le 15-01-2020 : Architecture logicielle.
- [S15] https://fr.wikipedia.org/wiki/Architecture_logicielle, le 15-01-2020 : Architecture logicielle.
- [S17] <https://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-composants-deploiement#L8-2>, le 28-02-2020 : Diagramme de composants.
- [S18] <https://www.slideshare.net/IbenbrahimSoumaya/2-tup>, le 28-02-2020 : 2 TUP
- [S19] https://www.e-bancel.com/Processus_2TUP.php, le 29-02-2020 : 2TUP