



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'UN SYSTÈME
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
Enseignant-chercheur à IAI

Encadreur

Mr FEZE Séraphin
Ingénieur informaticien à Métrika

Année académique 2018-2019

DÉDICACE

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

AVANT-PROPOS

RÉSUMÉ

ABSTRACT

LISTE DES FIGURES

Figure 1 : Organigramme de METRIKA.....	5
Figure 2 : Échantillon de tablette sumérienne.....	8
Figure 3 : Aperçu d'un QR code.....	16
Figure 4 : Image paiement par NFC.....	17
Figure 5 : Processus de développement en Y.....	27
Figure 6 : Diagramme de contexte statique.....	43
Figure 7 : Diagramme des classes.....	44
Figure 8 : Diagramme des cas d'utilisation.....	45
Figure 9: DCP du cas d'utilisation « acheter document ».....	46
Figure 10: DCP du cas d'utilisation « uploader document ».....	47
Figure 11: DCP du cas d'utilisation « rechercher un document ».....	47
Figure 12: Diagramme de séquences du cas d'utilisation « Uploader document ».....	48
Figure 13: Diagramme de séquence du cas d'utilisation « achat documents ».....	49
Figure 14 : Diagramme d'activités de navigation globale.....	50
Figure 15: Architecture MVC.....	55
Figure 16: Illustration de la structure de l'application côté serveur.....	56
Figure 17: Illustration de la structure de l'application côté client mobile.....	57
Figure 18: Architecture 2-tiers.....	58
Figure 19: Architecture 3-tiers.....	59
Figure 20: Architecture client-serveur de notre solution.....	60

LISTE DES TABLEAUX

Tableau 1: Formats de documents numériques.....	10
Tableau 2 : Récapitulatif des besoins fonctionnels.....	30
Tableau 3 : Cas d'utilisation et acteurs associés.....	31
Tableau 4: Découpage des cas d'utilisation en itération.....	32
Tableau 5: Découpage du projet en itérations suivant Scrum.....	33
Tableau 6: Structuration des cas d'utilisation en packages.....	33
Tableau 7: Récapitulatif outils de modélisation et de présentation.....	42
Tableau 8: Récapitulatif des outils de développement.....	54

GLOSSAIRE

Abréviations	Signification
ANINF	Agence Nationale des Infrastructures Numériques et des Fréquences
API	Application Programming Interface
CD	Compact Disc
DRM	Digital Rights Management
DVD	Digital Versatile Disc
EDI	Environnement de développement intégré
HTTP	HyperText Transfer Protocol
IHM	Interface homme-machine
JEE	Java Enterprise Edition
JSON	JavaScript Object Notation
MST	Magnetic Secure Transmission
NFC	Near Field Communication
RFID	Radio Frequency Identification
SGBD	Systèmes de gestion de bases de données
SMS	Short Message Service
SQL	Structured Query Language
REST	Representational State Transfer
QR	Quick Response
XML	Extensible Markup Language

TABLE DES MATIÈRES

DÉDICACE.....	ii
ÉPIGRAPHE.....	iii
REMERCIEMENTS.....	iv
AVANT-PROPOS.....	v
RÉSUMÉ.....	vi
ABSTRACT.....	vii
LISTE DES FIGURES.....	viii
LISTE DES TABLEAUX.....	ix
TABLE DES MATIÈRES.....	xi
INTRODUCTION GÉNÉRALE.....	1
Partie I : CONTEXTE ET ÉTAT DE L'ART.....	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.....	6
I.2.3 Objectifs généraux.....	7
Conclusion.....	7
Chapitre II: CONCEPTS LIÉS AU DOMAINE D'ÉTUDE.....	8
Introduction.....	8
II.1 Historique.....	8
II.2 Principales notions.....	9
II.2.1 Qu'est ce qu'un document numérique?.....	9
II.2.1.1 Définition.....	9
II.2.1.2 Supports de diffusion des documents numériques.....	9
II.2.1.3 Formats de documents numériques.....	9
II.2.2 Distribution numérique.....	10
II.2.3 L'indexation.....	11
II.2.3.1 Procédure d'indexation de documents.....	11
II.2.3.2 Types d'indexation.....	12
II.2.4 La recherche.....	13
II.2.4.1 Recherche non-informée.....	13
II.2.4.2 Recherche informée ou éclairée.....	13
II.2.4.3 Recherche contradictoire ou avec adversaire.....	13
II.2.4.4 Recherche par interpolation.....	14
II.2.5 Gestion des droits numériques.....	14

II.2.6 Paiement mobile.....	15
II.2.6.1 Paiement à distance.....	15
II.2.6.2 Paiement de proximité ou sans contact.....	16
II.2.6.3 Paiement de mobile à mobile.....	18
Conclusion.....	18
Partie II : CONCEPTION.....	19
Chapitre III: AUDIT DE L'EXISTANT ET MÉTHODOLOGIE DE DÉVELOPPEMENT.....	20
III.1 Solutions existantes.....	20
III.1.1 Kiosque numérique de Sogapresse.....	20
III.1.2 Autres kiosques numériques.....	20
III.2 Méthodologie de développement.....	21
III.2.1 Tour d'horizon des méthodes de développement.....	21
III.2.1.1 Les méthodes cartésiennes ou fonctionnelles.....	21
III.2.1.2 Les méthodes systémiques.....	22
III.2.1.3 Les méthodes orientées-objet.....	23
III.2.1.4 Les méthodes agiles.....	24
III.2.1.5 Les méthodes formelles.....	25
III.2.2 Choix d'une méthode.....	25
III.2.2.1 SCRUM.....	25
III.2.2.2 Le langage UML.....	26
III.2.2.3 2TUP.....	26
Chapitre IV: ANALYSE ET MODÉLISATION.....	30
IV.1 Analyse des besoins.....	30
IV.1.1 Analyse des besoins fonctionnels.....	30
IV.1.1.1 Recueil des besoins fonctionnels.....	30
IV.1.1.2 Identification des acteurs.....	31
IV.1.1.3 Identification des cas d'utilisation.....	31
IV.1.1.4 Classification et découpage des cas d'utilisation en itération.....	32
IV.1.1.5 Découpage du projet suivant Scrum.....	32
IV.1.1.6 Structuration des cas d'utilisation en packages.....	34
IV.1.1.7 Description textuelle des cas d'utilisation.....	34
IV.1.2 Capture des besoins techniques.....	41
IV.2 Cahier de charges.....	41
IV.2.1 Collecte et distribution de documents.....	42
IV.2.2 Recherche de documents.....	42
IV.2.3 Gestion des droits numériques.....	42
IV.3 Modélisation.....	42
IV.3.1 Outils.....	42
IV.3.2 Modélisation statique.....	44
IV.3.2.1 Diagramme de contexte statique.....	44
IV.3.2.2 Diagramme de classes préliminaire.....	45
IV.3.2.3 Diagramme des cas d'utilisation.....	45

IV.3.2.4 Diagramme de classes participantes(DCP).....	46
IV.3.3 Modélisation dynamique.....	48
IV.3.3.1 Diagramme de séquences.....	48
IV.3.3.2 Diagramme d'activités de navigation.....	50
Partie III : MISE EN ŒUVRE DE LA SOLUTION.....	52
Chapitre V: DÉMARCHE DE RÉALISATION.....	53
V.1 Technologie et outils d'implémentation.....	53
V.2 Présentation de la solution.....	55
V.2.1 Architecture de la solution.....	55
V.2.1.1 Le MVC.....	55
V.2.1.2 Architecture client-serveur.....	58
V.2.1.2.1 Types d'architecture client-serveur.....	58
V.2.1.2.2 Types de client.....	60
V.2.1.2.3 Présentation notre l'architecture client-serveur.....	60
V.2.2 Diagramme de déploiement.....	61
V.2.3 Résultats obtenus.....	61
Chapitre VI: CONDUITE DE PROJET.....	62



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, des podcasts 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'un éditeur(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.

L'objectif de notre travail est de mettre en place 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.



Partie I : CONTEXTE ET ÉTAT DE L'ART

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 essentiellement les services et missions de l'entreprise. Nous présenterons les concepts clés soulevés par notre sujet.

CHAPITRES

CHAPITRE I : LA STRUCTURE D'ACCUEIL ET LE SUJET

CHAPITRE II : CONCEPTS LIÉS AU DOMAINE D'ÉTUDE



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); ce, pour la bonne option, de production et de vulgarisation des connaissances se rapportant aux méthodes formelles et heuristiques relevant de la mise en œuvre des systèmes d'information d'aide à la décision (systèmes d'information décisionnels).

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 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

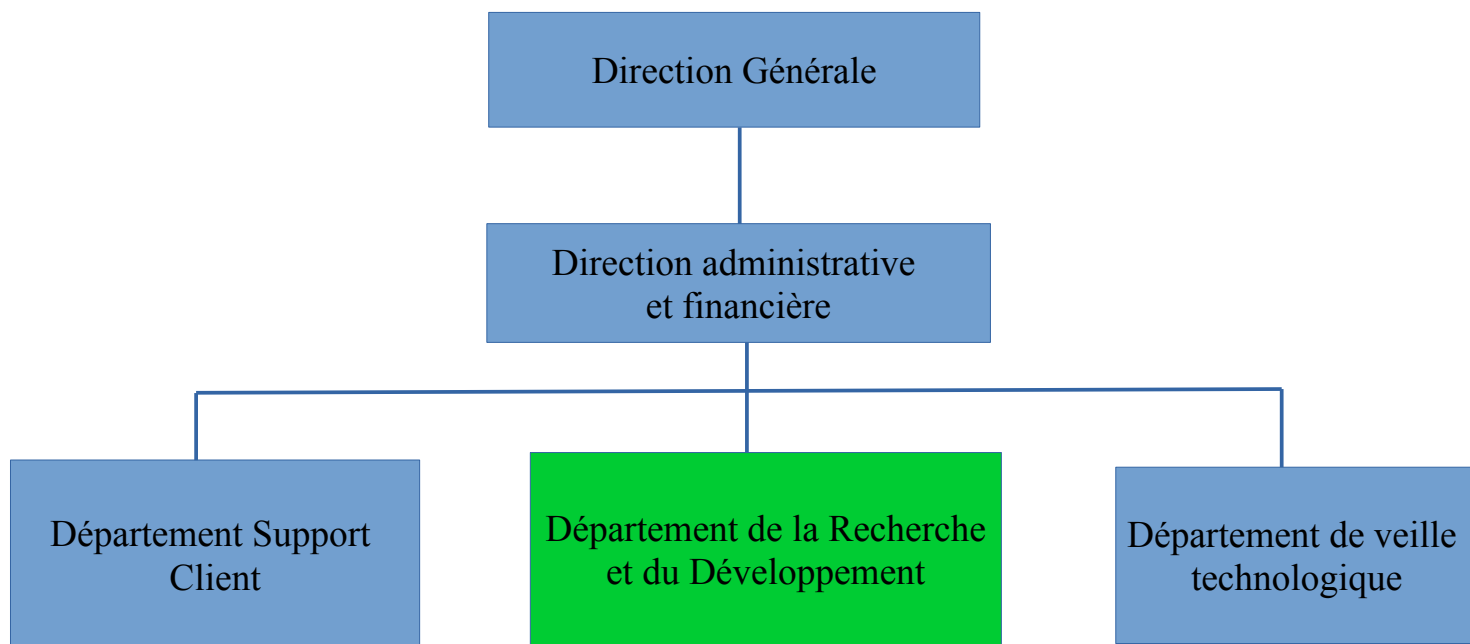


Figure 1 : Organigramme de METRIKA

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 contenus, 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 rencontrées très récemment par l'américain Blockbuster.

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

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 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).
- Offrir aux lecteurs un moyen d'acquisition de document par paiement électronique(mobile money).

Conclusion

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



Chapitre II: CONCEPTS LIÉS AU DOMAINE D'ÉTUDE

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 » qui 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. En 1963, Theodor Nelson invente le terme Hypertext, synonyme d'hypermédia. Par le biais du réseau électronique, le contenu du document numérique, textuel ou non, pourra être désormais affiché à l'écran. 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 ont 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 mobile

Le terme paiement mobile fait référence à tout paiement effectué à l'aide d'un appareil mobile. La dépendance croissante aux smartphones a engendré une multiplication de ces solutions de paiement simples et rapides. 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. Il existe plusieurs façons de payer avec son téléphone portable. Ces différents modes de paiement mobile seront développés dans les paragraphes suivants.

II.2.6.1 Paiement à distance

■ Paiement par Internet

Ce type de paiement est celui qui permet au consommateur de se connecter à Internet depuis son mobile 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 par SMS

Également appelés SMS+ ou premium SMS, les paiements par SMS sont disponible 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

II.2.6.2 Paiement de proximité ou sans contact

■ 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.

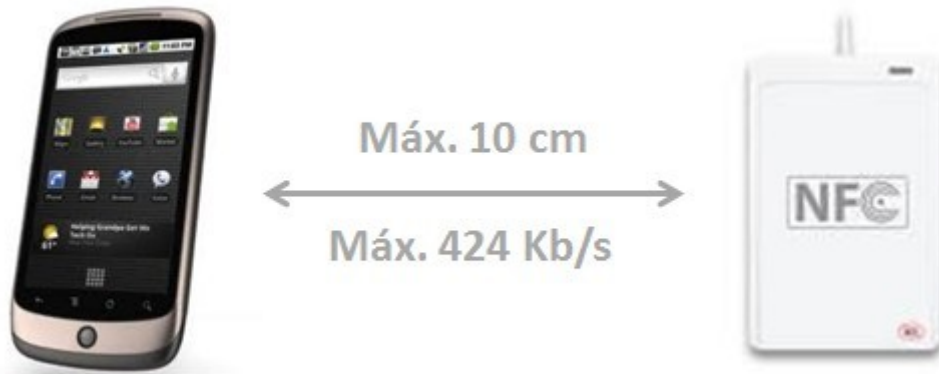


Figure 4 : Image paiement par NFC

■ Paiement par ondes sonores

Le paiement mobile 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.

II.2.6.3 Paiement de mobile à mobile

Aussi connu sous les noms Mobile Money et de portefeuille mobile, le paiement de mobile à mobile 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.

Conclusion

Par ce chapitre dédié aux concepts liés au sujet, nous mettons un terme à la première partie de notre travail. La partie suivante portera d'abord sur l'étude de l'existant et la méthodologie de développement. Puis nous terminerons par une analyse et une modélisation de notre problème.



Partie II : CONCEPTION

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 III: AUDIT DE L'EXISTANT ET MÉTHODOLOGIE DE DÉVELOPPEMENT

CHAPITRE IV: ANALYSE ET MODÉLISATION



Chapitre III: AUDIT DE L'EXISTANT ET MÉTHODOLOGIE DE DÉVELOPPEMENT

Introduction

Dans ce chapitre nous allons procéder séquentiellement à l'étude de l'existant et 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.

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 offerts respectivement par Libertis et Airtel qui sont des entreprises de téléphonie mobile implantées 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'ont vu le jour de nombreux sites de e-kiosque tels que www.ekiosque.cm et www.237kiosk.com, qui regroupent chacun une trentaine d'éditeurs. Ils permettent 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. En plus des moyens de paiements sus-cités, 237kiosk.com reçoit aussi les paiements par carte bancaire(Visa, Master Card, Discover Network, American Express). La recherche sur ces différents sites n'est assez performante ; ce qui constitue un frein pour un lecteur qui solliciterait un document traitant d'un sujet particulier. De plus, ils n'offrent pas la possibilité aux lecteurs de consulter les documents acquis hors ligne.

III.2 Méthodologie de développement

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.

III.2.1 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 :

III.2.1.1 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.

III.2.1.2 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.

III.2.1.3 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.

III.2.1.4 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).



III.2.1.5 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 de 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.

III.2.2 Choix d'une méthode

Au vu donc de la présentation des méthodes faites ci-dessus et au vu des contraintes de notre projet, nous avons choisi une démarche agile avec une méthodologie axée sur SCRUM et couplée au processus 2TUP. Ce choix s'explique notamment par les contraintes en termes de ressources humaines de notre projet (moins de 5 intervenants), le délai de livraison du projet (5 mois), le budget restreint. Nous préconisons en effet pour notre projet un processus de développement bien défini qui va de la détermination des besoins jusqu'à la conception, le codage final et la validation. Cette démarche exploite le langage UML, standard incontournable de la modélisation objet.

III.2.2.1 SCRUM

Scrum est une variante de méthode agile. Il est un framework itératif qui contient un ensemble de pratiques et de différents rôles. Il est composé d'un certain nombre d'itérations nommées sprints ayant une durée d'une à quatre semaines, et de trois rôles essentiels :



- **Scrum master** : celui qui maintient les processus ;
- **Product owner** : personne qui représente les investisseurs et entreprises ;
- **Team** : groupe de sept personnes au maximum, qui analysent, conçoivent et implémentent.

Durant chaque sprint, l'équipe crée un livrable. L'ensemble des fonctionnalités du sprint viennent du backlog du projet, qui est un ensemble priorisé d'exigences importantes à achever. Pendant une session de planification de sprint, les exigences du sprint sont définies. À la fin, l'équipe présente aux différentes parties prenantes du projet ce qui a été réalisé durant le sprint. Ils obtiennent de ce fait les différents retours lesquels doivent être intégrée au sprint suivant.

III.2.2.2 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.

III.2.2.3 2TUP

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

Process). 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 5).

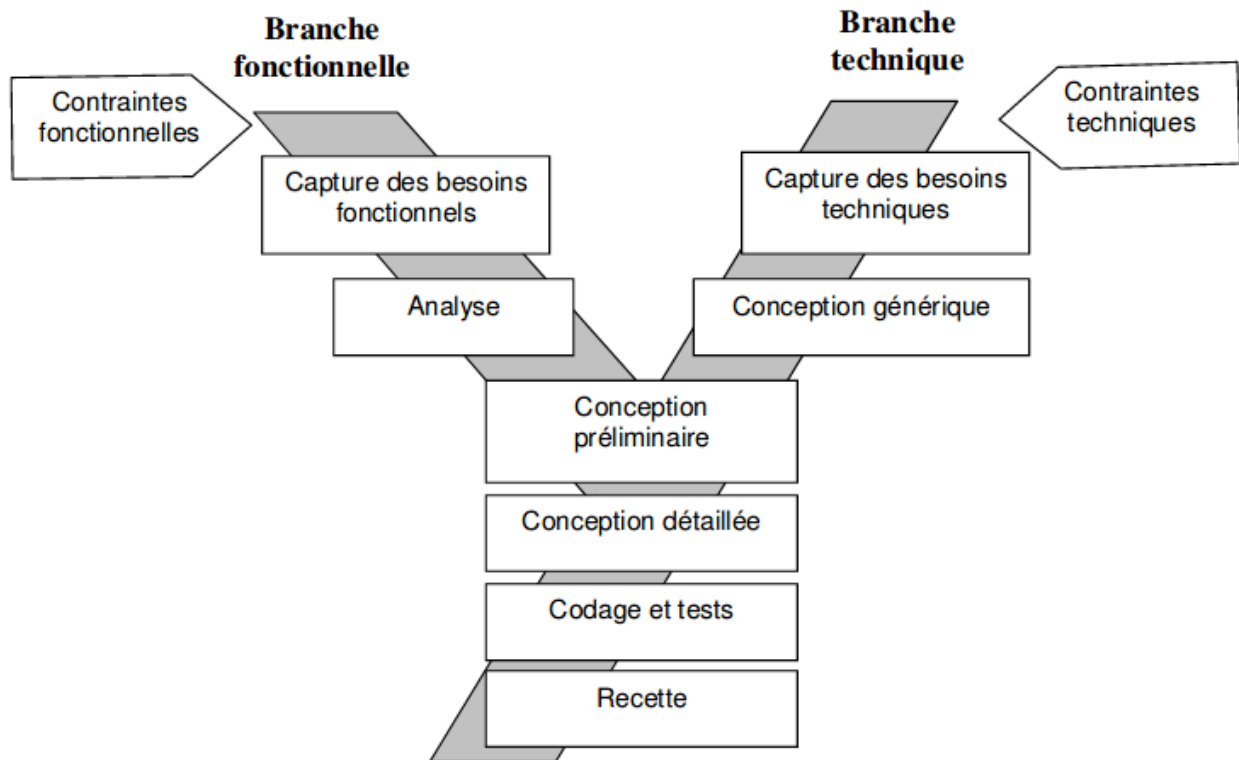


Figure 5 : Processus de développement en Y

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.

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 représentent 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 solutions et 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 IV: ANALYSE ET MODÉLISATION

Introduction

Dans ce chapitre, nous ferons d'abord une analyse des besoins afin d'établir notre cahier de charges. Nous terminerons par la construction des modèles du système à réaliser.

IV.1 Analyse des besoins

L'analyse des besoins est l'étape préalable à la réalisation d'un projet technologique d'envergure. Elle consiste à analyser le problème posé pour tenir compte des contraintes, des risques et de tout autre élément pertinent et assurer un ouvrage répondant aux besoins du client. Les besoins sont scindés en deux grands groupes : les besoins fonctionnels et les besoins non-fonctionnels.

IV.1.1 Analyse des besoins fonctionnels

IV.1.1.1 Recueil des besoins fonctionnels

Il s'agit des fonctionnalités du système. Ces besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées) :

Besoins	Fonctionnalités
Uploader(charger) un document	Collecter un document
Acheter un document	Distribuer un document
Enregistrer les informations d'identification d'un éditeur	Enregistrer une information
Enregistrer les informations d'identification d'un lecteur	Enregistrer une information
Empêcher le transfert d'un document vers n'importe quelle destination	Gérer les droits d'auteur
Consulter les documents hors connexion	Distribuer un document
Effectuer des recherches dans un document	Distribuer un document

Tableau 2 : Récapitulatif des besoins fonctionnels



IV.1.1.2 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.

IV.1.1.3 Identification des cas d'utilisation

Encore appelé use cases, 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 identifié les use cases listés dans le tableau ci-après. Il est à noter que les acteurs seront souvent désignés par le nom générique d'utilisateur.

Cas d'utilisation	Acteurs
S'authentifier	Visiteur
Créer compte client	Visiteur
Uploader un document	Éditeur
Rechercher un document	visiteur
Acheter un document	Client
Lire ou ouvrir un document	Client
Modifier les informations de connexion	Client, éditeur
Gérer les profils utilisateurs <ul style="list-style-type: none">• Créer un compte éditeur• Supprimer un compte utilisateur• Affecter un rôle à un utilisateur	Administrateur
Consulter les documents en vente	Visiteur
Créer abonnement	Administrateur



Visualiser les statistiques de vente

Administrateur, éditeur

Tableau 3 : Cas d'utilisation et acteurs associés

IV.1.1.4 Classification et découpage des cas d'utilisation en itération

Cas d'utilisation	Priorité fonctionnelle	Risques techniques	Itération
S'authentifier	Haute	Haut	2
Créer compte client	Haute	Haut	2
Consulter document en vente	Haute	Moyen	2
Uploader un document	Haute	Haut	2
Acheter un document	Haute	Haut	3
Ouvrir un document	Haute	Moyen	3
Modifier les informations de connexion	Moyen	Moyen	4
Gérer les profils utilisateur	Moyen	Moyen	4
Rechercher un document	Moyen	Bas	3
Visualiser les statistiques	Basse	Bas	3
Créer abonnement	Basse	Bas	4

Tableau 4: Découpage des cas d'utilisation en itération

IV.1.1.5 Découpage du projet suivant Scrum

➤ Sprints

En accord avec la méthode choisie, le projet contient quatre (4) itérations pouvant être exposées 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.

Numéro itération	Description	Risque
1	Pré-étude du problème	Information
2	Accès à la plateforme et publication de documents	Information, conception, implémentation
3	Acquisition des documents, suivi des opérations de publication et d'achat/vente	Information, conception, implémentation
4	Gestion des comptes et options d'acquisition	Information, conception, implémentation

Tableau 5: Découpage du projet en itérations suivant Scrum

➤ **Participants au projet et leurs rôles**

Plusieurs intervenants ont participé à la réalisation de ce projet, et chacun jouant son rôle tel que recommandé par Scrum :

- ✓ **Product Owner** : Métrika(Maître de stage)
- ✓ **Scrum master** : Encadreur académique
- ✓ **Team** : Stagiaire



IV.1.1.6 Structuration des cas d'utilisation en packages

Package	Cas d'utilisation
Authentification	S'authentifier
Acquisition	Consulter documents en vente
	Rechercher un document
	Uploader un document
	Acheter un document
	Ouvrir un document
Administration	Modifier les informations de connexion
	Gérer les profils utilisateurs
	Créer compte client
	Visualiser les statistiques
	Créer abonnement

Tableau 6: Structuration des cas d'utilisation en packages

IV.1.1.7 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 acteur d'ouvrir une session sur le système.
- ◆ Acteurs : Visiteur
- ◆ 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 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 les informations adéquates et valide.
4. Le système vérifie qu'il existe un utilisateur dont les informations d'identification correspondent à celles renseignées.
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 nécessaires à son inscription et les soumet.
4. Le système vérifie que les informations soumises sont valides.
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'utilisateur le remplit et le soumet.
4. Le système vérifie 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 ne doit pas être supérieure à 50 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 : 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 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 document à acheter.
 2. Le système affiche les moyens de paiement qu'il supporte.
 3. L'utilisateur choisi le moyen qui lui convient et valide.
 4. Le système affiche le récapitulatif de son achat et l'invite renseigner ses données de paiement.
 5. L'utilisateur renseigne lesdites données et valide.
 6. Le système confirme le succès de l'opération.
- Scénario alternatif

A4 : Les données de paiement sont erronées.
L'enchaînement A4 commence au point 5 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.

➤ 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 contenus 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 éditeur»

➤ Sommaire d'identification

- ◆ Titre : **Créer un 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 nécessaires à la création et les soumet.
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.

IV.1.2 Capture des besoins techniques

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de disponibilité, d'utilisabilité, de type de matériel ou le type de conception. Ces besoins peuvent également concerner les contraintes d'implémentation (langage de programmation, SGBD, système d'Exploitation...). Dans cet ordre d'idées, notre plateforme devra satisfaire les besoins suivants :

- Avoir une partie mobile et une partie web
- Offrir une interface utilisateur conviviale
- Supporter un nombre élevé de requêtes simultanées
- Avoir un temps de réponse assez faible (moins de 15 secondes)
- Être accessible à tout moment

IV.2 Cahier de charges

Après une étude de la problématique, des objectifs globaux, des concepts clés du domaine, de l'analyse des besoins, des limites des solutions existantes et des ambitions de METRIKA, nous avons établi notre cahier de charge. Le travail consistera à développer notre 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.



IV.2.1 Collecte et distribution de documents

Notre solution va permettre aux différents éditeurs de publier leurs documents sur la plateforme. Lesquels documents pourront être consultés puis acquis par les lecteurs au moyen du paiement mobile, notamment le Mobile Money.

IV.2.2 Recherche de documents

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

IV.2.3 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.

IV.3 Modélisation

IV.3.1 Outils

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).







Catégorie	Nom	Description	Logo
Modeleur UML	Modelio v3.8	Modelio est un outil de modélisation open source, gratuit et disponible sur les plates-formes 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 logiciel 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 Office. Il permet de créer, modifier et personnaliser les présentations et diaporamas.	

Tableau 7: Récapitulatif outils de modélisation et de présentation

- **Modeleur UML** : Comme outil de modélisation, notre choix s'est porté sur **Modelio**(version 3.8) et **Dia**(version 0.97.3-1). Ils sont des outils de



modélisation UML, open source, gratuit et disponible sur les plates-formes Windows, Linux et Mac. Prise en main facile, conforme à UML 2 et peu de ressources nécessaires à son fonctionnement, tels sont les atouts dont dispose Modelio et Dia et qui ont guidé notre choix.

- **Édition du mémoire et de la présentation : LibreOffice Writer** (Traitement de texte) et **WPS Presentation** (éditeur de diapositives).

IV.3.2 Modélisation statique

Nous allons y parvenir en respectant les jalons de la méthode de développement choisi précédemment.

IV.3.2.1 Diagramme de contexte statique

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

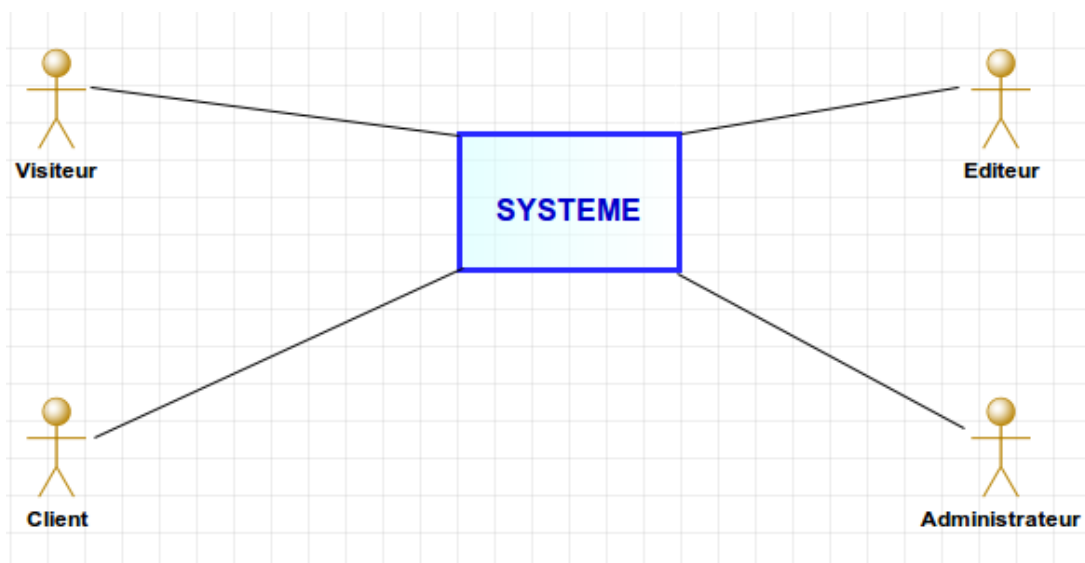


Figure 6 : Diagramme de contexte statique



IV.3.2.2 Diagramme de classes préliminaire

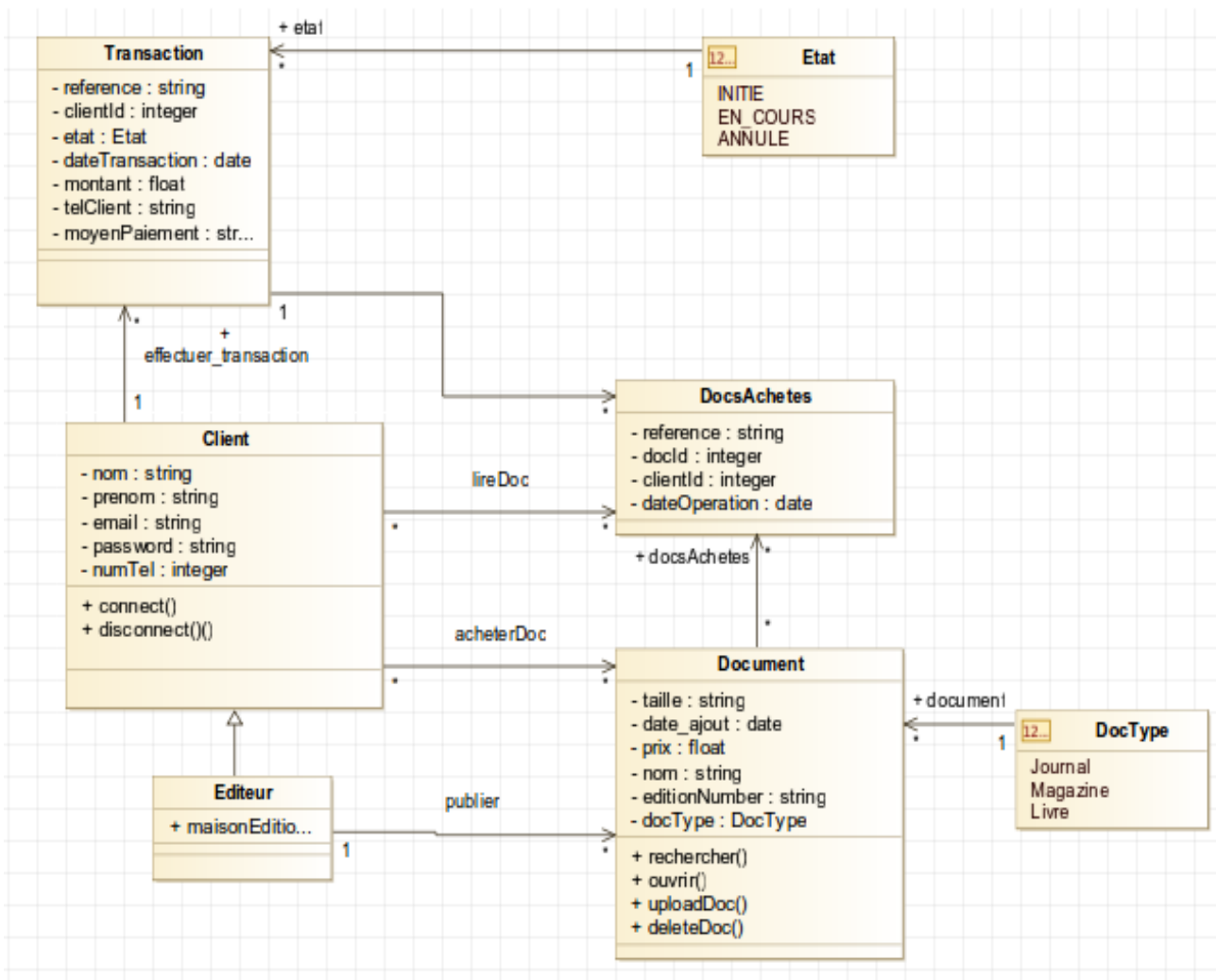


Figure 7 : Diagramme des classes

IV.3.2.3 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 :

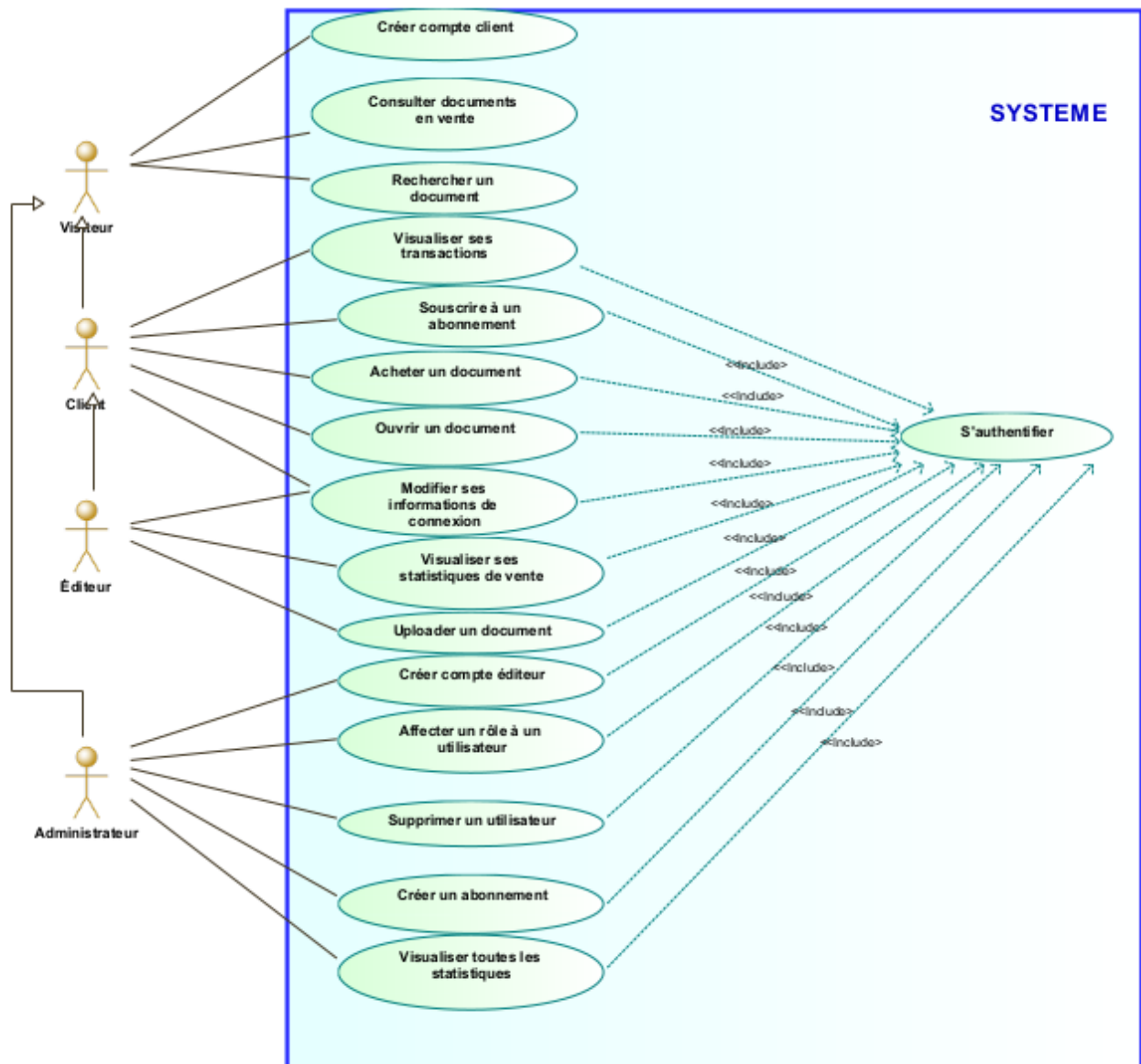


Figure 8 : Diagramme des cas d'utilisation

IV.3.2.4 Diagramme de classes participantes(DCP)

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.

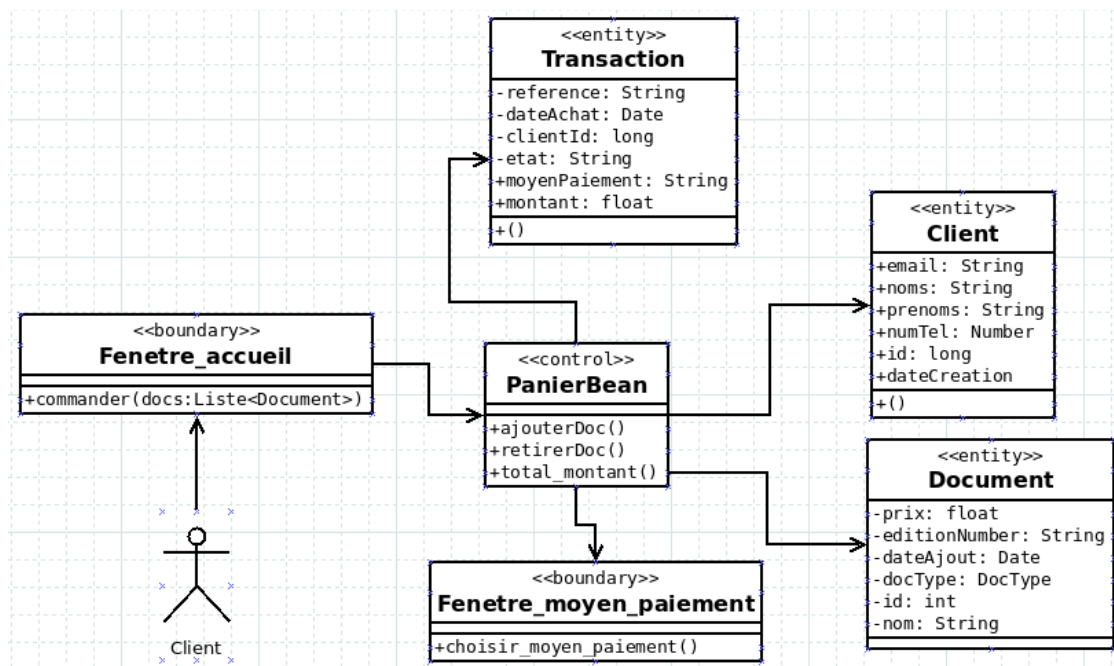


Figure 9: DCP du cas d'utilisation « acheter document »

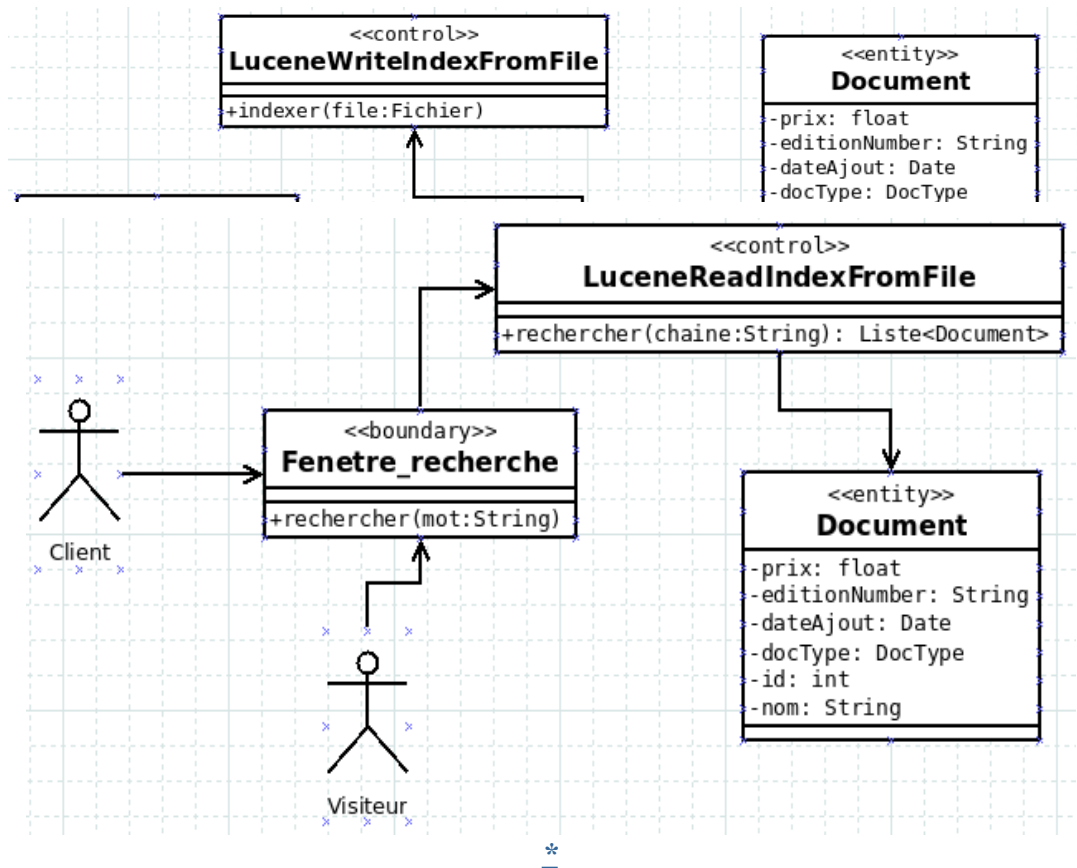


Figure 11: DCP du cas d'utilisation « rechercher un document »

IV.3.3 Modélisation dynamique

IV.3.3.1 Diagramme de séquences

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 « Uploader un document »

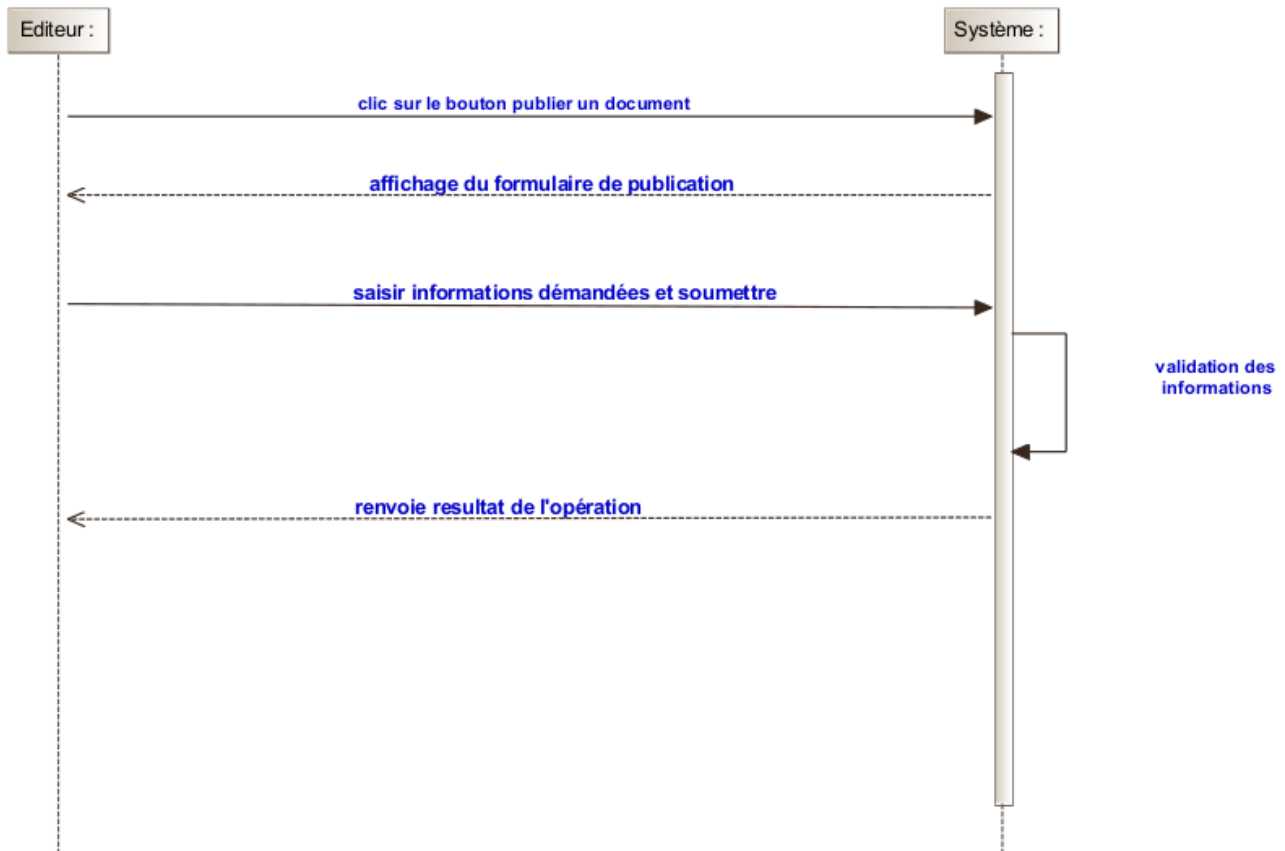


Figure 12: Diagramme de séquences du cas d'utilisation « Uploader document »

➤ **Diagramme de séquence du cas d'utilisation «Acheter documents »**

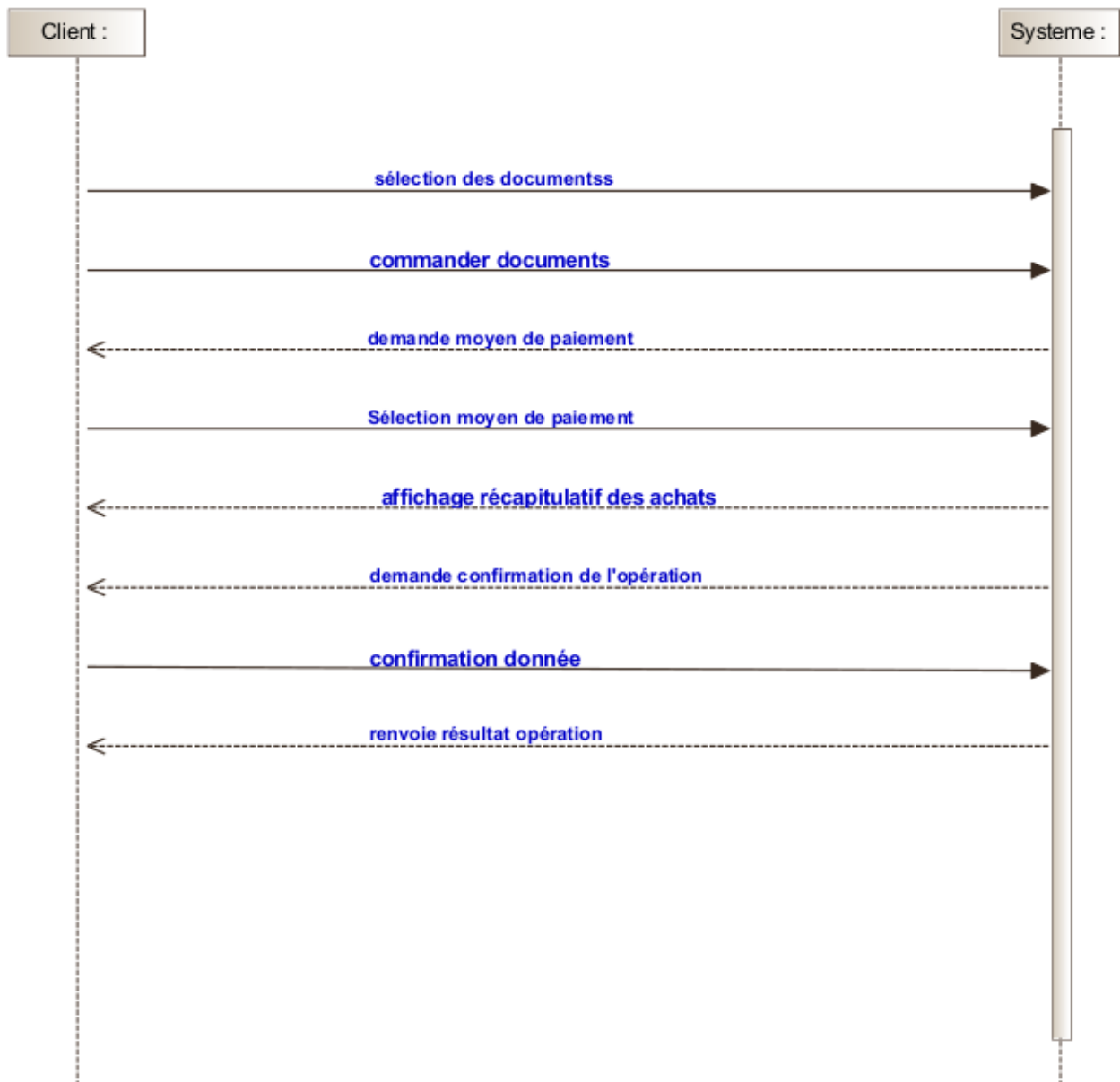


Figure 13: Diagramme de séquence du cas d'utilisation « achat documents »

IV.3.3.2 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.

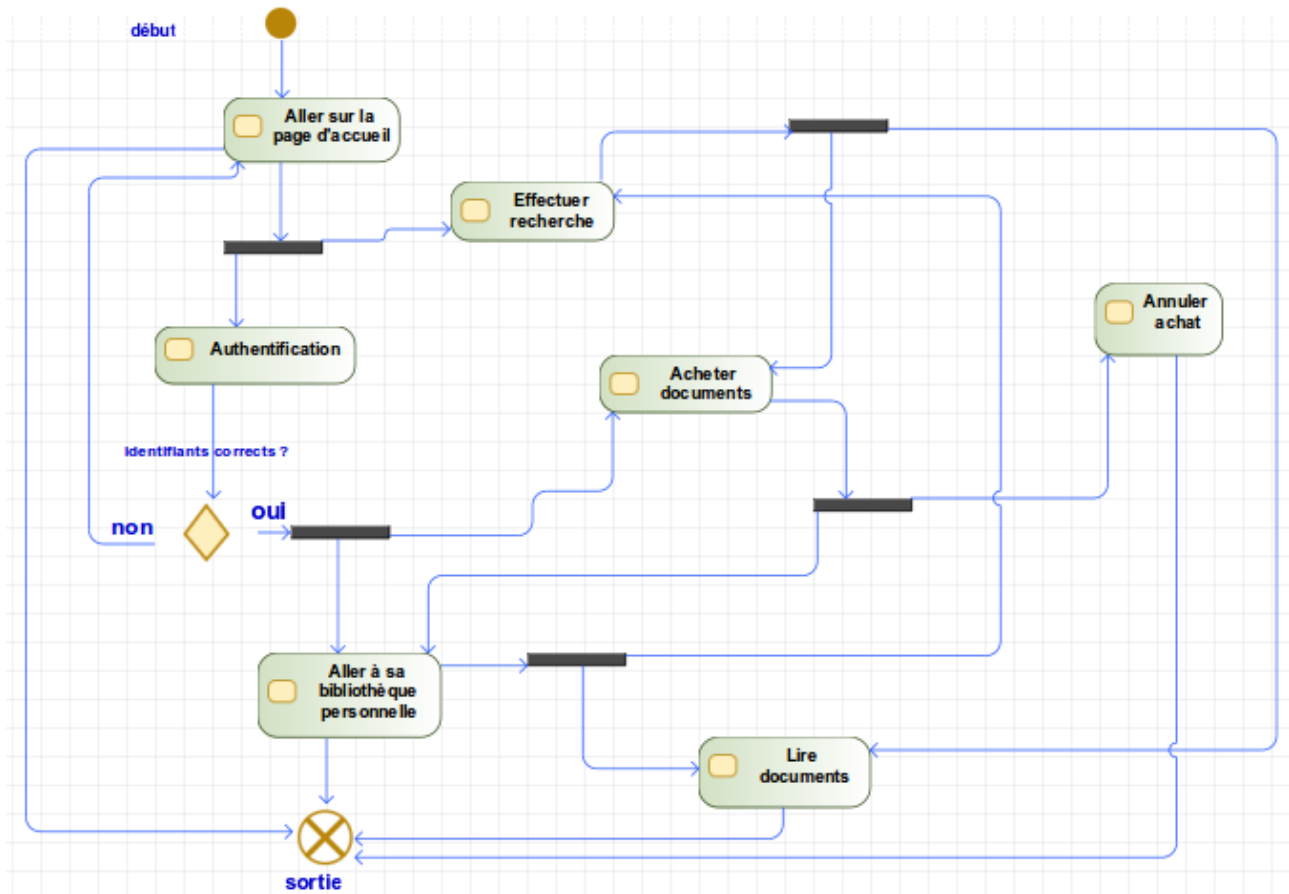


Figure 14 : Diagramme d'activités de navigation globale

Conclusion

Ce chapitre met un terme à cette partie dans laquelle nous avons présenté l'existant, choisi notre méthode de développement et procéder successivement 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 : MISE EN ŒUVRE 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 V: DÉMARCHE DE RÉALISATION

CHAPITRE VI: CONDUITE DE PROJET




Chapitre V: DÉMARCHE DE RÉALISATION

Introduction

V.1 Technologie et outils d'implémentation

Catégorie	Nom	Description	Logo
Environnements de développement	Eclipse photon v4.8		
	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. La première version stable a été publiée en décembre 2014.	
	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 distri-	



		buées, 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.	

Tableau 8: Récapitulatif des outils de développement

V.2 Présentation de la solution

V.2.1 Architecture de la solution

L'architecture d'un logiciel décrit la manière dont seront agencés les différents éléments 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 l'architecture MVC dans un environnement client-serveur.

V.2.1.1 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.

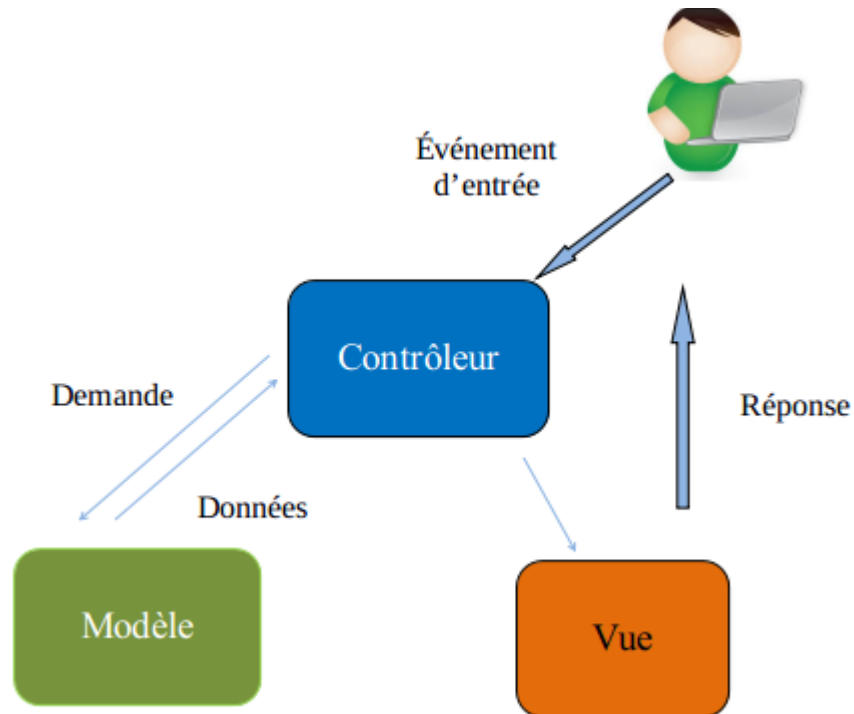


Figure 15: Architecture MVC

➤ Côté serveur

La structure globale de notre solution est celle d'une application web Java EE MVC décrite ci-après :

- la vue est assurée par des pages XHTML (Facelets)
- le modèle est assuré par des entités(JavaBeans)
- le contrôleur décomposé en deux éléments :
 - ✓ un unique aiguilleur central : la FacesServlet servant de point d'entrée à toute requête.
 - ✓ un objet particulier(JavaBean), déclaré via une annotation et désigné par le terme managed-bean.

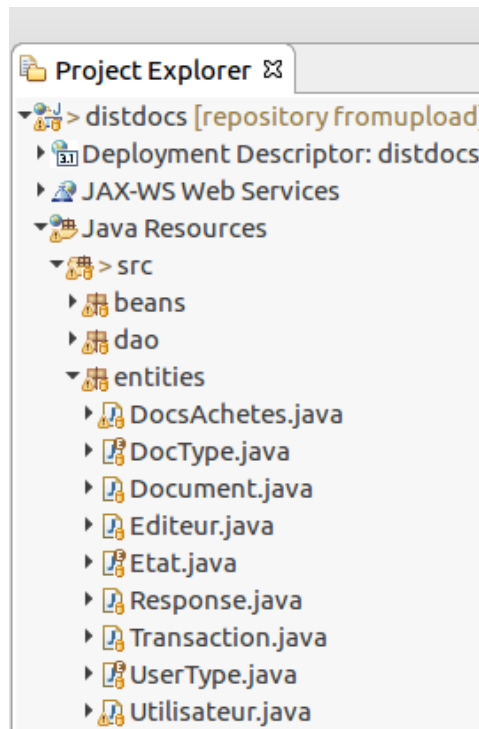


Figure 16: Illustration de la structure de l'application côté serveur

➤ Côté client

L'organisation de notre application est assez bien établie car l'outil de développement Android studio offre une configuration et une structuration par défaut, assez modulable et modifiable à souhait. La figure 14 qui va suivre est une vue partielle cette structuration.

Notons aussi que le client mobile(appareil Android) communiquera avec le serveur à l'aide des web services. Un service Web est un ensemble de protocoles et de normes utilisés pour l'échange de données entre des applications ou des systèmes hétérogènes. Les applications écrites dans divers langages de programmation et exécutées sur diverses plates-formes peuvent utiliser des services Web pour échanger des données sur des réseaux informatiques.

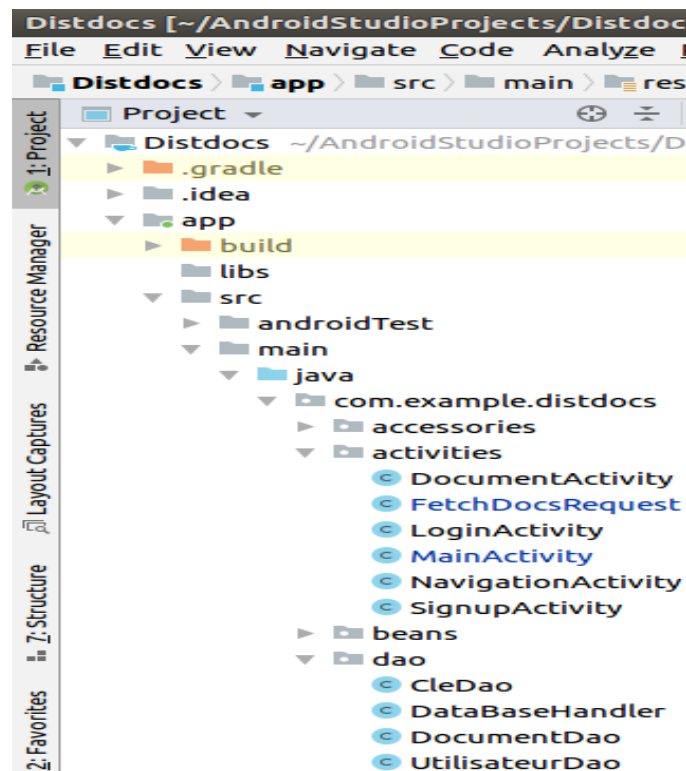


Figure 17: Illustration de la structure de l'application côté client mobile

V.2.1.2 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.2.1.2.1 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.

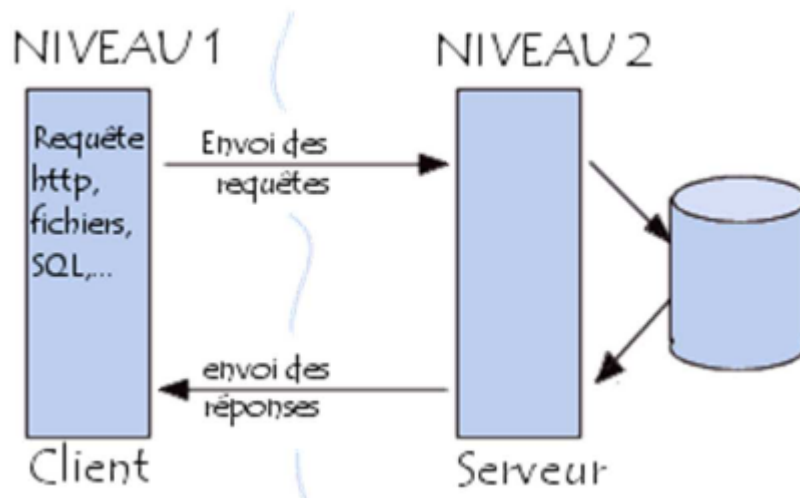


Figure 18: Architecture 2-tiers

➤ 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.

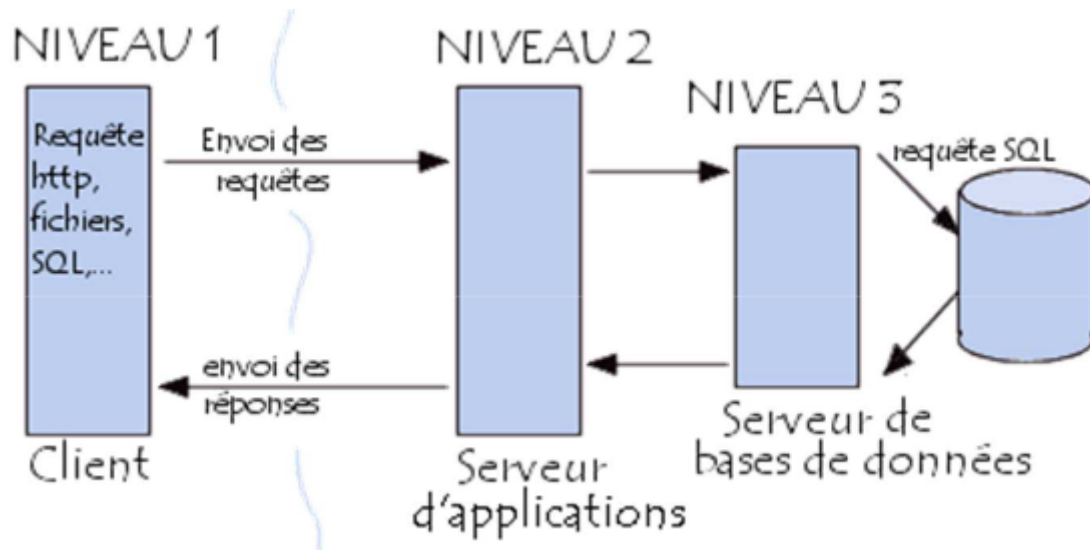


Figure 19: Architecture 3-tiers

➤ 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.2.1.2.2 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.2.1.2.3 Présentation notre l'architecture client-serveur

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



- ✓ Le client est soit un appareil mobile Android(client lourd) ou un navigateur web(client léger). 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 communiquera avec le serveur via uniquement des requêtes HTTP.
- ✓ Le serveur d'application est Glassfish car c'est l'implémentation de référence de la technologie(JEE) utilisée coté web et il prend également en charge l'API Java pour les Web services RESTful (JAX-RS). L'accès aux données sera faite à l'aide de l'ORM (Object Relational Mapping) EclipseLink qui est l'implémentation de référence de JPA(Java Persistence API) ;
- ✓ le serveur de données est le SGBD MySQL car léger, simple d'utilisation, installation et prise en main faciles.

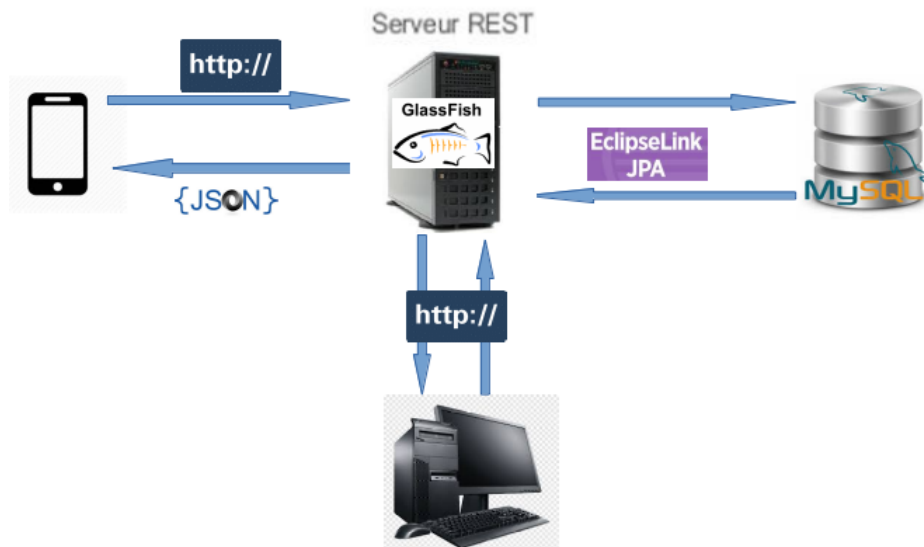


Figure 20: Architecture client-serveur de notre solution

V.2.2 Diagramme de déploiement

V.2.3 Résultats obtenus

Conclusion



Chapitre VI: CONDUITE DE PROJET



CONCLUSION GÉNÉRALE



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.



WEBOGRAPHIE

- [S1] https://fr.wikipedia.org/wiki/Document_num%C3%A9rique#Supports_de_diffusion, le 15-05-2019.
- [S2] <https://www.cairn.info/revue-les-enjeux-de-l-information-et-de-la-communication-2002-1-page-2.htm#>, le 18-05-2019.
- [S3] http://www1.rfi.fr/actufr/articles/075/article_42240.asp, le 22-05-2019.
- [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 30-05-2019.
- [S5] <https://fr.mobiletransaction.org/types-paiement-mobile/>, le 25-06-2019.
- [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 27-06-2019.
- [S7] <http://paiement-mobile.info/technologies/qr-code>, le 27-06-2019.
- [S8] https://fr.wikipedia.org/wiki/Distribution_num%C3%A9rique, le 29-07-2019.
- [S9] <http://www.infcom.ens.tn/ead/Chapitre1/ch1sec132.htm>, le 28-09-2019.
- [S10] <http://www.ordinateur.cc/programmation/Computer-Programming-Languages/87470.html>, le 28-09-2019.
- [S11] https://www.memoireonline.com/10/10/4000/m_Etude-et-mise-en-place-dun-systeme-informatise-de-transfert-dargent-inter-agences-COMECI4.html, le 28-09-2019.
- [S12] https://www.collectionscanada.gc.ca/obj/s4/f2/dsk1/tape7/PQDD_0020/MQ56896.pdf, le 28-09-2019.
- [S13] <https://www.supinfo.com/articles/single/6765-methodologies-developpement-logiciel>, le 29-09-2019.
- [S14] <https://fr.gadget-info.com/difference-between-informed>, le 06-11-2019.
- [S15] http://turing.cs.pub.ro/auf2/html/chapters/chapter3/chapter_3_2_1.html, le 06-11-2019.
- [S16] <https://www.softfluent.fr/blog/architecture-logicielle-pour-application/>, le 15-12-2019.
- [S17] https://fr.wikipedia.org/wiki/Architecture_logicielle, le 15-12-2019.
- [S18] [https://fr.wikipedia.org/wiki/Architecture_\(informatique\)](https://fr.wikipedia.org/wiki/Architecture_(informatique)), le 16-12-2019.
- [S19] <https://fr.wikipedia.org/wiki/Client%E2%80%93serveur>, le 19-12-2019.