

MST-RSI-02/2017

Mémoire de projet de fin d'étude

Présenté par

Ayoub MKHARBACH

Pour obtenir le diplôme du Master Sciences et Techniques

En Réseaux et Systèmes Informatiques

Sous le thème

Etude, conception et réalisation d'une application web
de gestion des processus métiers des demandes au sein
des organismes

Soutenu le 27/09/2017

Devant le jury

Pr. Bouchaib NASSERDDINE	FST Settat	Président
Pr. Abderahim MERZOUK	FST Settat	Encadrant
Pr. Amine BENMAKHLOUF	FST Settat	Examinateur
Pr. Abdelkrim HAQIQ	FST Settat	Examinateur
Mr. Khalid FARHI	UPSYs	Encadrant

Année Universitaire 2016 / 2017

Dédicaces

Je dédie ce travail,

Principalement aux êtres vivants les plus chers à mon cœur, à citer ma chère mère, **CHEIKH Fatiha**, avec son caractère tendre et ses conseils et critiques tout le temps constructif et à mon cher père, **MKHARBACH Reddad**, ma première référence dans cette vie et ma source de motivation.

Spécialement à mon grand-père **CHEIKH Abass**, qui est mort avant de me voir comme il a voulu. J'aurai bien aimé lui consacrer ce travail, ce que j'ai pu accomplir en suivant ses conseils. Avec ce travail je lui rends un grand hommage plein de sentiments d'estime et de respect.

A mes chers sœurs et frère **Nadia**, **Fatima** et **Yassine**, qui ont tant constitué pour moi une source de motivation pour aller en avant et ne cesser de rêver afin de leurs donner l'exemple et la référence.

A mes amis en témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble.

A tous les enseignants et les professeurs, qui ont sacrifié leurs ressources afin de m'inculquer les bons principes de la vie et me rendre fier d'appartenir à un tel pays qui est le MAROC.

Remerciements

C'est avec un réel plaisir que j'exprime ici mes reconnaissances à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

J'adresse les sincères remerciements à mon cher encadrant de projet de fin d'étude **Mr. MARZOUK Abderrahim**, pour leurs conseils précieux, leur soutien et leur disponibilité tout au courant de mon stage.

Je remercie vivement **Mr. FARHI Khalid** mon parrain de stage pour sa disponibilité sans condition, pour ses directives, qui m'a aidé tout au long de l'élaboration de ce projet et qui a tout fait pour me mettre dans les bonnes conditions de travail.

Je tiens à remercier également tous mes collègues dans UPSYS pour leurs accueils et leurs écoutes.

A notre chef de filière, **Mr. NASSERDDINE Bouchaib** qu'il trouve ici l'expression de ma profonde gratitude, pour sa veille sur le bon déroulement de notre formation.

Mes remerciements vont aussi à l'encontre de mes chers parents qui n'ont ménagés aucun effort et qui ne cesseront de le faire, pour me soutenir à tout moment de ma vie.

Enfin j'adresse mes plus sincères remerciements à tous les membres du jury qui m'ont honoré par leur présence et en acceptant de juger mon travail.

Résumé

Le présent rapport constitue le résultat d'un travail réalisé dans le cadre du projet de fin d'études, au sein de l'entreprise UPSYS, pour l'obtention du diplôme Master Sciences et Techniques en Réseaux et Systèmes Informatiques.

Mon stage de fin d'étude consiste à l'étude, la conception et la réalisation d'une application web qui permet la gestion des processus métiers des demandes, cette application est destinée à être utilisée par n'importe quel organisme pour gérer ses demandes.

L'application va être livré en premier à la commune de tiznit, de ce fait elle doit contenir des modules qui seront dédié pour gérer les demandes d'autorisations au sein de la commune.

Ce stage a été une expérience enrichissante, il m'a permis de découvrir des nouvelles technologies et des nouveaux concepts dans le monde de développement et de travailler avec une équipe créative et perspicace.

Mots clé : UPSYS – Développement – Processus métier -Gestion – Demande.

ملخص

العمل المقدم في هذا التقرير، يأتي في إطار موضوع التخرج للحصول على شهادة الماجستير في الاتصالات والنظم المعلوماتية، والمنجز في شركة أبسيد سيتيم.

تدريب نهاية دراستي يهدف إلى دراسة، تصميم وإنجاز تطبيق ويب لإدارة عمليات تداول الطلبات، المراد من التطبيق هو استخدامه من قبل أي منظمة لإدارة طلباتها الداخلية.

التطبيق سيتم تسليمه في الأول لبلدية تيزنيت، وبالتالي فسيحتوي على وحدات تتعلق فقط بإدارات طلبات الترخيص داخل البلدية.

هذا التدريب كان تجربة مفيدة، لقد مكنتني من اكتشاف تكنولوجيات ومفاهيم جديدة في عالم البرمجة والعمل مع فريق مليء بالحيوية وروح الفريق.

Abstract

This report is the result of a work done in the Project graduation within UPSYS company in order to obtain the master's degree in Networks and Informatic Systems.

My internship consists on design and realization of a web application that allows the management of business process, this application is intended to be used by any organization to manage its requests.

This application will firstly be delivered to the commune of Tiznit, therefore it must contain modules that will be dedicated to manage requests for authorization within the town.

This internship was a rewarding experience, it allowed me to discover new technologies and new concepts in the world of development and work with a creative team.

Key words: UPSYS – Development – Business process - Management – Request.

Sommaire

Dédicaces	2
Remerciements	3
Résumé.....	4
ملخص	5
Abstract	6
Sommaire.....	7
Liste des figures	10
Liste des tableaux.....	11
Liste des abréviations	12
Introduction générale.....	13
Chapitre 1 : Contexte générale du projet.....	14
Introduction	15
I. Présentation de la société :	15
1. Présentation d'UPSYS	15
2. Fiche Juridique :	16
3. Organigrammes	16
4. Les offres d'UPSYS :	17
II. Cahier des charges :	19
1. Problématique :	19
2. Objectif :	19
3. Processus et tâches :	20
4. Cas de gestion des autorisations de l'urbanisme :	21
III. Langage et méthodologie de conception :	22
1. Introduction :	22
2. La démarche UP-XP :	23
3. Phases de modélisation :	24
4. Le formalisme UML :	24
5. L'outil PowerAMC :	25
IV. Planning et conduite du projet :	26
1. Gestion de projet :	26
2. Diagramme de GANTT :	26
3. L'équipe de travail :	27
Chapitre 2 : Phase d'inception.....	28
Introduction	29

I.	Capture des besoins :	29
1.	Définition des besoins fonctionnels :	29
2.	Les besoins non fonctionnels :	30
3.	Regroupement des exigences ¹¹ par intentions d'acteurs :	31
II.	Diagramme de cas d'utilisation :	32
	Chapitre 3 : Phase d'analyse et de conception	33
	Introduction :	34
I.	Description textuelle des cas d'utilisation :	34
1.	Authentifier :	34
2.	Consulter le guide :	35
3.	Inscrire :	36
4.	Consulter l'état d'avancement d'une demande :	37
6.	Gérer :	38
7.	Planifier les réunions :	41
8.	Effectuer ses tâches :	42
II.	Diagramme de séquence « Boite noire » :	43
1.	Authentifier :	43
2.	Inscrire :	44
3.	Consulter le guide :	44
4.	Consulter l'état d'avancement d'une demande :	45
5.	Gérer :	45
6.	Planifier les réunions :	46
7.	Effectuer ses tâches :	46
III.	Diagramme de classe d'analyse :	47
1.	Gestion des processus :	48
2.	Gestion des demandes :	48
3.	Gestion des utilisateurs :	49
4.	Gestion des référentiels d'adresses :	49
5.	Gestion des intervenants :	50
6.	Gestion des dossiers :	50
7.	Gestion des réunions :	51
8.	Gestion des emails :	51
9.	Gestion des traçabilités :	51
IV.	Prototype des interfaces :	52
1.	Interface de gestion :	52
2.	Interface de d'accueil :	53

3. Interface de suivi :	53
Chapitre 4 : Phase de réalisation.....	54
Introduction :.....	55
I. Développement :.....	55
1. Choix de développement :	55
2. Choix de la plateforme Java EE :	57
3. L'architecture :.....	58
II. Environnement logiciel :	63
1. IntelliJ IDEA :	63
2. Apache tomcat :	64
3. Maven :.....	64
4. GitHub :.....	65
5. Projetcotor :.....	65
III. Les interfaces de l'application :.....	66
1. Interface d'authentification :.....	66
2. Interface de bienvenue :.....	66
3. Interface du niveau d'accueil :	67
4. Interface du niveau d'exploitation :	67
5. Interface du niveau de planification :	68
6. Interface des autres niveaux :	68
7. Interface de gestion des comptes :	69
8. Interface de gestion des entités :.....	69
9. Interface de gestion des processus :	70
10. Interface de gestion des types :.....	70
11. Interface de gestion des référentiels :	71
12. Interface de gestion des processus :	71
13. Interface de gestion des réunions :	72
Conclusion générale.....	73
Bibliographie et Webographie :.....	74
Annexes.....	75
Annexe 1 : Convention de nommage.	75
Annexe 2 : Choix des technologies.....	78
Annexe 3 : Diagramme de classe.....	80

Liste des figures

Figure 1 : Organigramme d'UPSYS	16
Figure 2 : Organigramme de service d'intégration logicielle et développement	17
Figure 3 : Diagrammes de UML	25
Figure 4 : Logo de PowerAMC	25
Figure 5 : Diagramme de Gantt.....	27
Figure 6 : Diagramme de cas d'utilisation.....	32
Figure 7 : Diagramme de séquence « Authentifier »	43
Figure 8 : Diagramme de séquence « Incrire ».....	44
Figure 9 : Diagramme de séquence « Consulter le guide ».....	44
Figure 10 : Diagramme de séquence « Consulter l'état d'avancement d'une demande ».....	45
Figure 11 : Diagramme de séquence « Gérer ».....	45
Figure 12 : Diagramme de séquence « Planifier les réunions ».....	46
Figure 13 : Diagramme de séquence « Effectuer ses tâches ».....	46
Figure 14 : Diagramme de classe « Gestion de processus ».....	48
Figure 15 : Diagramme de classe « Gestion des demandes ».....	48
Figure 16 : Diagramme de classe « Gestion des utilisateurs »	49
Figure 17 : Diagramme de classe « Gestion des référentiels d'adresses ».....	49
Figure 18 : Diagramme de classe « Gestion des intervenants ».....	50
Figure 19 : Diagramme de classe « Gestion des dossiers »	50
Figure 20 : Diagramme de classe « Gestion des réunions »	51
Figure 21 : Diagramme de classe « Gestion des emails ».....	51
Figure 22 : Diagramme de classe « Gestion des traçabilité ».....	51
Figure 23 : Prototype de l'interface de gestion.....	52
Figure 24 : Prototype de l'interface d'accueil	53
Figure 25 : Prototype de l'interface de suivi	53
Figure 26 : Modèle MVC (Modèle – Vue – Contrôleur).....	56
Figure 27 : Architecture de travail	58
Figure 28 : Interface d'authentification	66
Figure 29 : Interface de bienvenue.....	66
Figure 30 : Interface du niveau d'accueil.....	67
Figure 31 : Interface du niveau d'exploitation	67
Figure 32 : Interface du niveau de planification	68
Figure 33 : Interface des autres niveaux	68
Figure 34 : Interface de gestion des comptes	69
Figure 35 : Interface de gestion des entités	69
Figure 36 : Interface de gestion des processus.Figure 35 : Interface de gestion des entités	69
Figure 36 : Interface de gestion des processus	70
Figure 37 : Interface de gestion des types.Figure 36 : Interface de gestion des processus.....	70
Figure 37 : Interface de gestion des types	70
Figure 38 : Interface de gestion des référentiels.Figure 37 : Interface de gestion des types.....	70
Figure 38 : Interface de gestion des référentiels.....	71
Figure 39 : Interface de gestion des commissions.Figure 38 : Interface de gestion des référentiels.....	71
Figure 39 : Interface de gestion des commissions	71
Figure 40 : Interface de gestion des réunions.Figure 39 : Interface de gestion des commissions	71
Figure 40 : Interface de gestion des réunions	72
Figure 40 : Interface de gestion des réunions	72

Liste des tableaux

Tableau 1 : L'équipe de travail.....	27
Tableau 2 : Tableau des exigences.....	29
Tableau 3 : Tableau des exigences par intention d'acteur.....	31
Tableau 4 : Description textuelle de cas d'utilisation « Authentifier ».....	34
Tableau 5 : Description textuelle de cas d'utilisation « Consulter le guide ».....	35
Tableau 6 : Description textuelle de cas d'utilisation « Incrire ».....	36
Tableau 7 : Description textuelle de cas d'utilisation « Consulter l'état d'avancement d'une demande ».....	37
Tableau 8 : Description textuelle de cas d'utilisation « Ajouter ».....	38
Tableau 9 : Description textuelle de cas d'utilisation « Modifier ».....	39
Tableau 10 : Description textuelle de cas d'utilisation « Supprimer ».....	40
Tableau 11 : Description textuelle de cas d'utilisation « Planifier une réunion ».....	41
Tableau 12 : Description textuelle de cas d'utilisation « Effectuer ses tâches ».....	42

Liste des abréviations

HTML : HyperText Markup Language

UPSYS : UPSIDE SYSTEMS

UP-XP : Unified-Process eXtreme-Programming

UML : Unified Modeling Language

GED : Gestion électronique des documents

SaaS : Software as a Service

IHM : Interface Homme Machine

MVC : Module – Vue - Contrôleur

Java EE : Java Entreprise Edition

REST : REpresentational State Transfer

SQL : Structured Query Language

CSS : Cascading Style Sheets

IO : Input/ Output

XML : Extensible Markup Language

PDF : Portable Document Format

API : Application Programming Interface

JSON : JavaScript Object Notation

SGBD : Système de Gestion de Base de Données

SGBDRO : Système de Gestion de Base de Données Relationnel et Objet

CDMI : Cloud Data Management Interface

RFC : Request For Comments

IDE : integrated development environment

JSP : JavaServer Pages

ISO : International Organization for Standardization

ITIL : Information Technology Infrastructure Library

Introduction générale

Dans le cadre de ma formation à la Faculté des sciences et techniques de Settat, j'ai eu l'occasion d'effectuer un stage de fin d'études au sein de la société UPSYS. Ce dernier m'a permis de mettre en œuvre les informations et les idées théoriques acquises durant toute la période de formation, d'effectuer une analyse interne et externe et d'établir un contact réel et direct avec le milieu professionnel.

C'est déjà quelques années que les technologies d'informations et les activités des organisations et plus particulièrement le web ont évolué d'une façon croissante et remarquable. Aujourd'hui, le web est un secteur en perpétuelle expansion face à l'apparition du web 3.0¹ et les nouvelles technologies notamment le Spring Boot², AngularJS³, etc.

C'est dans ce contexte que plusieurs sociétés essayent de profiter au maximum possible de ces technologies afin d'améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression.

Dans ce cadre, UPSYS souhaite développer une application web qui permettant de gérer les processus métiers.

En guise de présenter notre travail d'une façon bien organisé, nous avons opté pour le plan suivant :

Le premier chapitre concerne la présentation du cadre générale du stage.

Le second chapitre porte sur la première phase de conception et de développement intitulé la phase d'inception, elle permet de déterminer les besoins attendus de l'application.

Le troisième chapitre porte sur la phase d'analyse et conception qui vise à identifier et décrire la majorité des besoins utilisateurs, construire l'architecture de base du système et lever les risques majeurs du projet.

Le quatrième chapitre porte sur la phase de réalisation qui sera consacré pour la présentation des langages et des outils de programmation utilisés pour la réalisation de mon application.

¹ Web 3.0: Mot utilisé en futurologie à court terme pour désigner le Web qui suit le Web 2.0 et constitue l'étape à venir du développement du World Wide Web.

² Spring Boot : Framework de développement Java, décrire dans le quatrième chapitre.

³ AngularJS : Framework Javascript, décrire dans le quatrième chapitre.

Chapitre 1 :

Contexte générale du projet.

« Un projet est un effort complexe pour atteindre un objectif bien spécifique, devant respecter un échéancier et un budget ... », Définition d'un projet.

Introduction :

L'étude du projet est une démarche stratégique visant à organiser le bon déroulement d'un projet et d'assurer la conduite de toutes les phases qui le constituent.

Une étude complète et efficace conduit généralement à la réussite d'un projet. Cette étude fera donc l'objet de notre premier chapitre qui sera consacré à la présentation du projet ainsi que la définition de notre langage et méthodologie de développement.

I. Présentation de la société :

1. Présentation d'UPSYS :

UPSYS est une société de services et d'intégration, spécialisée dans les technologies de l'information.

L'offre globale d'UPSYS s'appuie sur les dernières innovations, sa stratégie de croissance vise à renforcer ses domaines de compétences et à développer de nouvelles activités en adéquation avec l'évolution technologique.

Les solutions proposées par UPSYS sont basées sur les meilleures technologies du marché, pour un maintien soutenu du système d'informations à un niveau technologique et fonctionnel de qualité.

Afin de répondre aux besoins de ses clients, UPSYS propose une palette diversifiée de services comme du stockage, de la sauvegarde, de l'archivage, de solution CLOUD⁴ sur MESURE, de le GED⁵, ainsi que la solution de travail collaboratif en mode SaaS⁶.

⁴ Cloud : Nuage en français, permet de données un accès à distance aux données sans savoir son emplacement physique, décrire dans le partie I-5 de ce chapitre.

⁵ GED : est un procédé faisant intervenir des moyens électroniques – typiquement des logiciels et/ou des matériels – pour prendre en charge la gestion des documents, décrire dans le partie I-5 de ce chapitre.

⁶ SaaS : Mode de cloud, permet à une entreprise de ne plus installer d'applications sur ses propres serveurs mais de s'abonner à des logiciels en ligne et de payer un prix qui variera en fonction de leurs utilisations effectives.

2. Fiche Juridique :

Raison sociale : Upside Systems

Sigle : UPSYS

Forme juridique : Société à responsabilité limitée.

Siège social : Apt N°1, Immeuble B, Koutoubia 2, Ave Med Vi – Marrakech.

TEL/FAX : +212 (0)5 24 42 36 00

+212 (0)5 24 43 41 11

Bureau de Casablanca : Technopark de Casablanca, Bureau 249, 2ème étage.

TEL/FAX: +212 (0)5 22 52 31 73

+212 (0)5 22 52 31 73

Site Web: <http://www.upsysmaroc.com/>

3. Organigrammes :

L'organigramme général d'UPSYS est le suivant :

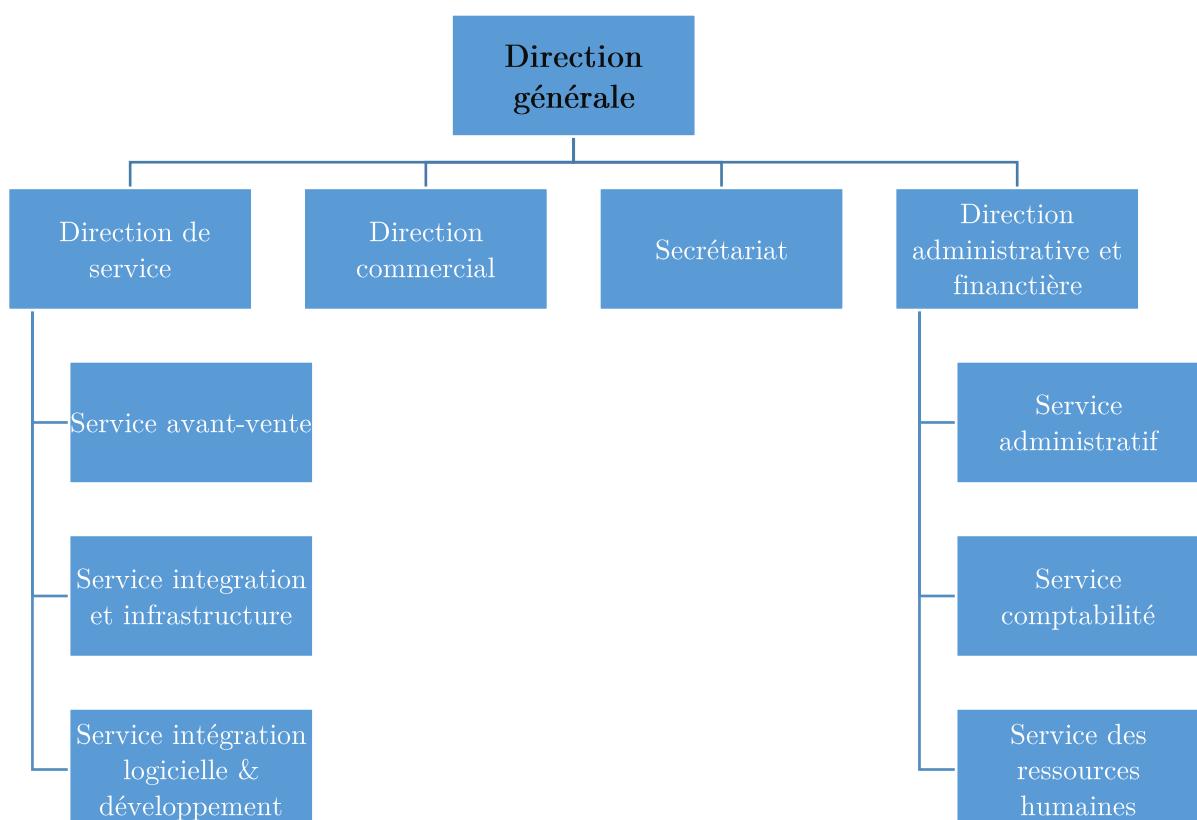


Figure 1 : Organigramme d'UPSYS.

Le stage se déroulera dans le « Service Intégration Logiciel & Développement » qui est organisé comme suit :

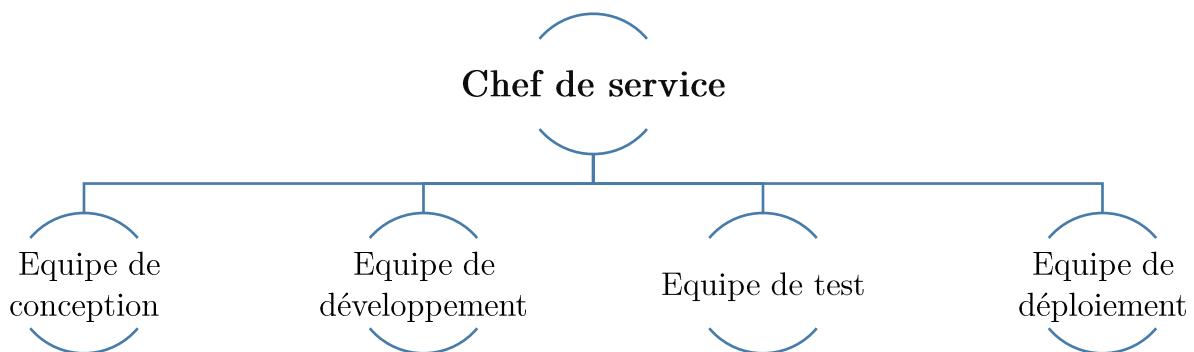


Figure 2 : Organigramme de service d'intégration logicielle et développement.

4. Les offres d'UPSYS :

Outre son savoir-faire technologique, l'un des atouts d'UPSYS réside dans l'originalité de son organisation : calquée sur la structure matricielle des systèmes informatiques, elle autorise une meilleure réactivité et fédère toutes les forces de l'entreprise vers les clients.

UPSYS occupe une position très forte au carrefour des technologies, des logiciels et des services spécialisés. L'offre globale d'UPSYS s'appuie sur les dernières innovations en termes d'e-technologies, couvre l'ensemble des besoins des clients durant toutes les étapes du cycle de vie de la donnée et permet une totale maîtrise de votre système d'information. Sa stratégie de croissance vise à renforcer ses domaines de compétences et à développer de nouvelles activités en adéquation avec les besoins de ses clients.

UPSYS est un acteur majeur dans la construction de systèmes d'information, et cela à tous les niveaux : Intégrateur global d'infrastructures stratégiques, performance, sécurité et développement de solutions de collaboration.

Les solutions de sauvegarde :

Le choix du système de sauvegarde est fait sur plusieurs critères :

- ✓ La capacité de stockage du support de sauvegarde.
- ✓ La vitesse de la sauvegarde.
- ✓ La fiabilité du support.
- ✓ La facilité à restaurer les données.

Les solutions de stockage objet :

Devant la prolifération exponentielle des données non structurées (vidéos, images haute définition, messagerie, GED, etc) qui touche pratiquement tous les secteurs d'activité, le stockage objet devient en effet un véritable besoin pour l'entreprise.

Il permet la fourniture d'un accès très rapide (de n'importe où et n'importe quand), une protection efficace contre la perte de données, et une évolutivité sans limites pour un budget raisonnable.

Virtualisation :

UPSYS propose la mise en œuvre de la solution d'infrastructures virtuelles complète intégrant :

- ✓ Le déploiement de produits de virtualisation.
- ✓ La migration des serveurs physiques candidats à la virtualisation vers des machines virtuelles.
- ✓ Les services associés.
- ✓ La formation du personnel.

L'archivage :

Limiter l'explosion des volumes en adoptant des solutions pour :

Réduire le volume des messageries et de la consommation d'espace de stockage, notamment en déportant les fichiers attachés.

- ✓ Archiver de façon sélective et appropriée les documents concernés.
- ✓ Limiter fortement les délais de sauvegarde.
- ✓ Permettre l'accès aux données (mails, contenus, rapports ...).

Cloud sur mesure :

UPSYS propose une solution leader du marché qui utilise les équipements modernes de mobilité dans un environnement professionnel.

- ✓ Le concept de base du Cloud sur mesure permet à ses clients de transformer leurs environnements existants en infrastructure prêtes pour le Cloud.
- ✓ La volonté est de mettre à la disposition de ses clients un environnement Cloud tangible où ils pourront construire leur Cloud sur mesure tout en bénéficiant d'une gestion de contenus, de fonctionnalités et de sécurité d'une grande qualité.

- ✓ Les collaborateurs de ses clients peuvent ainsi accéder aux données et travailler sur n'importe quel terminal et de n'importe où.
- ✓ UPSYS propose une architecture verticale complète du Cloud depuis l'agent sur le poste utilisateur Windows, MacOS, Smartphone et Tablette (iOS, Windows Phone et Android), jusqu'à l'équipement logiciel et matériel central de stockage et de partage sécurisé de données utilisateurs.

Gestion Electronique de Documents :

UPSYS propose une solution pour vous permettre de :

- ✓ Contrôler les versions de documents
- ✓ Réduire le coût et le temps des recherches.
- ✓ Accéder aux documents très rapidement.
- ✓ Accéder de partout et à partir de n'importe quel terminal, et permettre aux employés de l'entreprise de collaborer où qu'ils soient et à tout moment.

II. Cahier des charges :

1. Problématique :

De nos jours, le citoyen est trop occupé par son quotidien. De ce fait il ne dispose pas de suffisamment de temps pour assurer le suivi de ses affaires auprès des services administratifs. Par ailleurs, dans le cas d'une demande d'autorisation pour la construction par exemple, le citoyen marocain se voit heurté devant une multitude d'intervenants administratifs lui rendant difficile le suivi de l'état d'avancement de sa demande, ce qui provoque une perte de temps énorme pour trouver l'information souhaitée auprès de différents services de ladite administration.

2. Objectif :

Afin de remédier à cette problématique et rendre facile le suivi de la demande du citoyen, il m'a été demandé de réaliser un projet intitulé « Gestion des processus métiers des demandes au sein des organismes ».

L'objectif de ce projet est d'analyser, de concevoir et de réaliser une application destinée à :

- ✓ Dématérialiser les processus de validation des demandes,
- ✓ Faciliter le dépôt de la demande,
- ✓ Faciliter le suivi de sa requête,
- ✓ Assurer la traçabilité des tâches effectuées durant le processus,
- ✓ Trouver facilement l'information où qu'il soit.

3. Processus et tâches :

L'application consiste en premier lieu de créer des circuits d'activités (des workflows⁷ ou bien des processus) selon le service désiré. Donc un paramétrage préalable est nécessaire pour calquer sur les processus existant au sein de l'organisme.

L'organisme est constitué de plusieurs processus de travail, le processus lui-même est composé de plusieurs tâches.

Pour accomplir une activité relativement à un circuit donné, une tâche passe par plusieurs acteurs de ce circuit, les tâches sont liées entre elles par des actions à entreprendre. Chaque tâche est associée à un groupe d'agents qui vont les exécuter d'une façon circulaire.

Chaque tâche peut ou pas contenir des documents (un document est un ensemble de fichiers électroniques qui peuvent être physique au départ).

Une tâche est caractérisée par :

- ✓ Un numéro unique et un nom,
- ✓ Le processus dans lequel elle appartient,
- ✓ Une date de début et une date de fin, et la date réelle de sa fin,
- ✓ Manuelle ou automatique,
- ✓ Les tâches précédentes et les tâches suivantes,
- ✓ Sa priorité,
- ✓ L'état de la tâche,

Le processus lui-même est caractérisé par un certain nombre de propriétés :

- ✓ Un numéro unique et un nom,
- ✓ Sa date de début et sa date de fin,
- ✓ Progression du processus,
- ✓ Il est clôturé ou non,
- ✓ L'état de sa clôture,

⁷ Workflow : c'est la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier.

4. Cas de gestion des autorisations de l'urbanisme :

Objectif du projet :

Le principal objectif de ce projet est l'informatisation et l'automatisation du circuit de traitement de l'ensemble des autorisations d'un urbanisme.

Le but de ce projet est notamment l'amélioration de la gestion opérationnelle des autorisations, faciliter l'accès à l'information et le suivi du traitement des dossiers.

Descriptif détaillé des fonctionnalités :

Les utilisateurs de l'application sont :

- Administrateur d'application (paramétrages technique)
- Gestionnaire fonctionnel (paramétrage fonctionnel)
- Ressources humaines
- Organisateur
- Agent
- Utilisateurs externes

Chaque utilisateur a des fonctionnalités qui sont cité comme ceci :

a. LE CITOYEN :

- ✓ Incrire
- ✓ Consulter le guide des démarches administratives

b. LE DEMANDEUR :

- ✓ Suivre les demandes

c. GESTIONNAIRE TECHNIQUE (ADMINISTRATEUR) :

Ce gestionnaire aura la responsabilité de définir et valider la gestion des comptes, gestion des architectes et aussi des topographes, donc il peut :

- ✓ Gérer les comptes
- ✓ Gérer des architectes
- ✓ Gérer les topographes

d. ORGANISATEUR :

C'est lui qui aura la responsabilité de gérer les processus de validation des demandes.

- ✓ Gérer les processus

e. RESSOURCES HUMAINES :

Ce gestionnaire aura la responsabilité de définir et valider les utilisateurs et les entités (Les services et les communes).

- ✓ Gérer les utilisateurs
- ✓ Gérer des entités

f. GESTIONNAIRE FONCTIONNEL :

Le Paramétrage fonctionnel sera de la responsabilité du gestionnaire fonctionnel, ce gestionnaire aura la responsabilité de définir et valider :

- ✓ Gérer les services
- ✓ Gérer les demandes
- ✓ Gérer les pièces par type de demande
- ✓ Gérer les intervenants par type de demande
- ✓ Gérer les motifs par types de demande
- ✓ Gérer les commissions

g. AGENT D'EXECUTION :

- ✓ Effectuer ses tâches

III. Langage et méthodologie de conception :

1. Introduction :

Pour bien conduire notre projet et nous assurer du bon déroulement des différentes phases, nous avons opté la démarche UP-XP comme une méthodologie de conception et de développement.

Après le choix de la méthodologie, nous avons besoin d'un langage de modélisation comme UML.

Notre choix s'est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu'il propose. Aussi UML présente le meilleur outil pour schématiser des systèmes complexes sous un format graphique et textuel simplifié et normalisé.

En effet UML n'est ni un processus ni une démarche, d'où il fallait choisir une méthodologie de conception et de développement que nous devons l'adopter.

2. La démarche UP-XP :

L'objectif de la démarche UP-XP est de pouvoir analyser et concevoir un logiciel orienté objet⁸ évolutif et ayant l'aptitude à s'adapter aux éventuels changements que peut connaître le métier de l'entreprise à l'avenir en plusieurs niveaux : Fiscale, productif, organisationnel, juridique, etc.

UP- Unified Process :

La méthode du Processus Unifié (UP pour Unified Process) est un processus de développement itératif et incrémental, ce qui signifie que le projet est découpé en phases très courtes à l'issue de chacune desquelles une nouvelle version incrémentée est livrée.

Il s'agit d'une démarche s'appuyant sur la modélisation UML pour la description de l'architecture du logiciel (fonctionnelle, logicielle et physique) et la mise au point de cas d'utilisation permettant de décrire les besoins et exigences des utilisateurs.

XP-eXtreme Programming :

La méthode XP (pour eXtreme Programming) définit un certain nombre de bonnes pratiques permettant de développer un logiciel dans des conditions optimales en plaçant le client au cœur du processus de développement.

L'eXtreme Programming est notamment basé sur les concepts suivants :

- ✓ Les équipes de développement travaille directement avec le client sur des cycles très courts d'une à deux semaines maximum.
- ✓ L'équipe livraison de versions du logiciel intervient très tôt et à une fréquence élevée pour maximiser l'impact des retours utilisateurs.
- ✓ L'équipe de développement travaille en collaboration totale sur la base de binôme.
- ✓ Le code est testé et nettoyé tout au long du processus de développement.
- ✓ Des indicateurs permettent de mesure l'avancement du projet afin de permettre de mettre à jour.

⁸Orienté objet : Type de programmation, Il permet de manipuler toute sorte d'information sous forme d'un objet qui contient des propriétés et des méthodes.

3. Phases de modélisation :

Phase d'inception : Recensement et définition des besoins.

C'est la phase ayant comme principal objectif l'identification des principaux acteurs interagissant avec le système et les fonctionnalités attendues de celui-ci. Dans un premier temps, nous allons lister l'ensemble des exigences issues du cahier des charges. Chaque exigence sera recensée et elle pourra être étudiée ultérieurement. Ces exigences seront regroupées par intentions d'acteur, puis un diagramme de contexte statique sera établi.

A ce stade, nous pouvons déjà construire un diagramme de cas d'utilisation de notre projet.

Phase d'analyse et de conception :

Il s'agit d'une phase itérative qui vise à analyser et comprendre chaque fonctionnalité attendue du système en examinant toutes les scénarios courants des scénarios exceptionnels et des erreurs.

Une description détaillée est faite pour chaque use case.

Nous allons réaliser un diagramme de séquence boîte noir correspond au système (Ici c'est un diagramme de séquence par scénario, où la boîte noire correspond au système informatique à développer).

A partir des diagrammes de séquences et des descriptions textuelles, nous réaliserons le diagramme de classe d'analyse.

Phase de réalisation :

Dans cette phase nous déduirons à partir du diagramme de classe d'analyse le modèle logique de donnée pour la création de la base de données et ceci en appliquant les règles de passages.

4. Le formalisme UML :

UML est considéré comme le langage standard de conception orienté objet, il est un formalisme et pas une méthode. Il s'en suit qu'il définit un ensemble d'éléments de modélisation et une notation graphique pour modéliser les systèmes et ne décrit pas les étapes à suivre pour le faire.

Les raisons qui nous ont poussés à adopter UML dans notre projet se résume en :

UML offre un outil prêt à l'emploi basé sur une modélisation visuelle qui permet d'échanger des modèles compréhensibles.

Si on développe avec des langages orienté objet, il est plus approprié de concevoir avec des formalismes Orienté Objet.

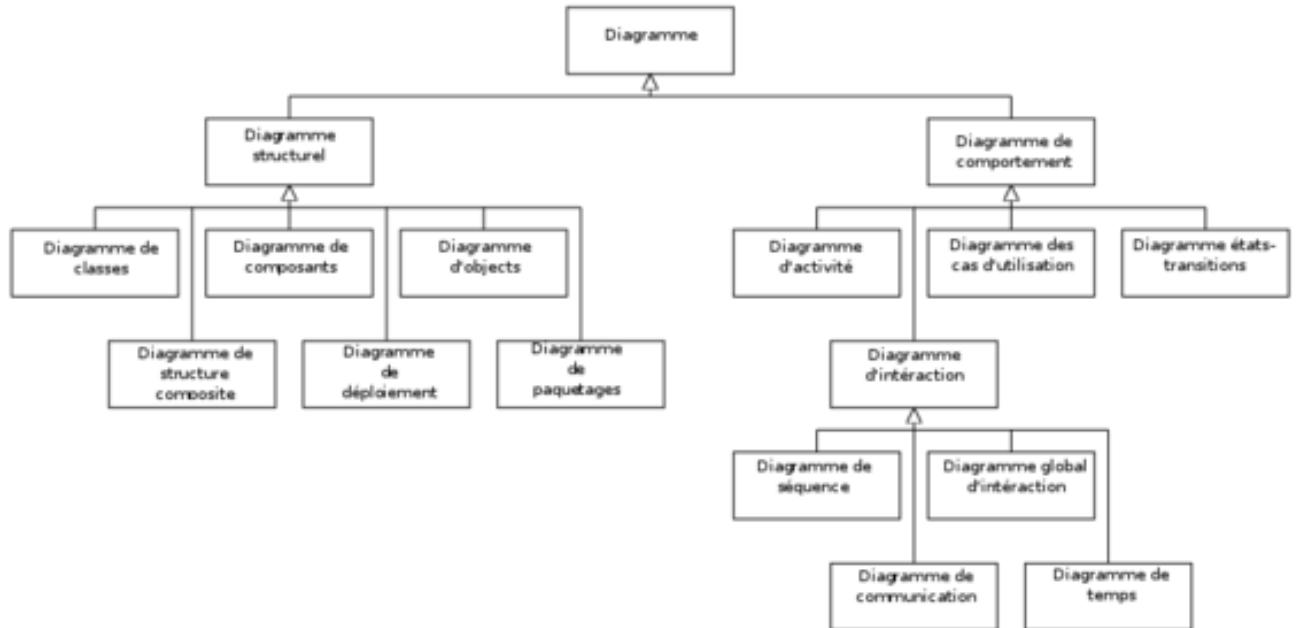


Figure 3 : Diagrammes de UML.

5. L'outil PowerAMC :

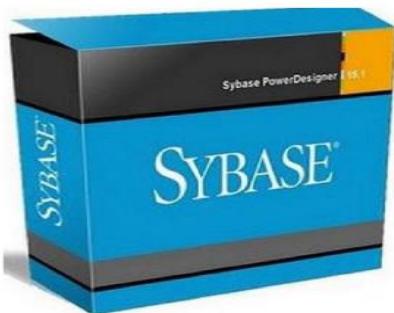


Figure 4 : Logo de PowerAMC.

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques en n'importe quels modèles informatiques.

PowerAMC offre des méthodes efficaces d'analyse d'impact, de gestion des changements et des techniques avancées de gestion des métadonnées.

La combinaison des techniques de modélisation et de la gestion des données confère à PowerAMC des fonctions uniques lui permettant de prendre en charge tous les environnements architecturaux. Le référentiel de métadonnées de PowerAMC permet également à toutes les parties prenantes de l'entreprise de collaborer et de communiquer efficacement. Ces dernières peuvent ainsi réagir plus rapidement face aux changements et garantir une meilleure capacité d'adaptation de l'entreprise.

IV. Planning et conduite du projet :

1. Gestion de projet :

La gestion de projet est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Lorsque la gestion de projet porte sur un ensemble de projets concourant à un même objectif, on parle de gestion de programme.

Le management de projet est un management qui se caractérise par :

- ✓ L'irréversibilité des opérations des participants.
- ✓ Un fort degré de liberté des actions des participants.
- ✓ Une organisation vouée à être évolutive et temporaire.
- ✓ Des flux de trésorerie d'investissement pouvant être négatifs.
- ✓ Une forte influence de variables exogènes sur le projet.

2. Diagramme de GANTT :

Le diagramme de Gantt (Harmonogram Adamieckiego⁹) est un outil utilisé en ordonnancement et gestion de projet et permettant de visualiser dans le temps les diverses tâches liées composant un projet. Il permet de représenter graphiquement l'avancement du projet.

Le premier diagramme de ce type fut réalisé par ingénieur polonais Karol Adamiecki en 1896. Il l'a décrit en 1931, mais la langue de publication n'a pas permis la reconnaissance internationale de son idée. Pour cette raison le concept a été nommé après Henry L. Gantt, ingénieur américain, qui a publié la description du diagramme en 1910.

Dans notre projet nous avons utilisé l'outil Gantt Project.

⁹Harmonogram Adamieckiego : La première nomination du diagramme de Gantt.

Diagramme de Gantt :

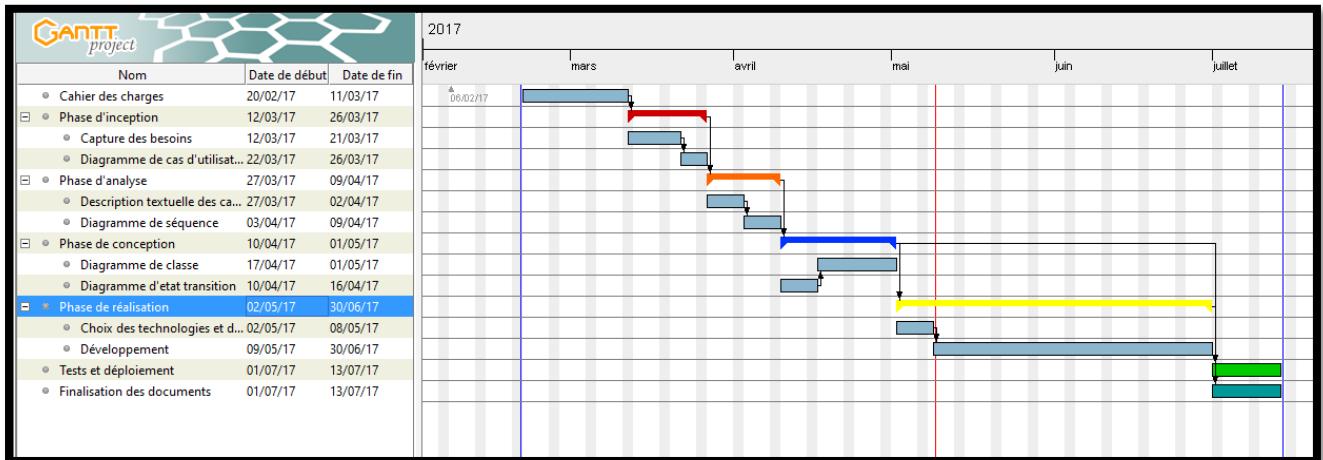


Figure 5 : Diagramme de Gantt.

3. L'équipe de travail :

Plusieurs types de ressources sont nécessaires à la bonne réalisation du projet.

Les ressources humaines se composent d'une équipe de sept personnes qui sont :

Personne	Rôle	Description
Mohammed EL ADLANI Abdelkrim DERRAG Amine HAQQAOUI	Client	C'est eux qui définissent les fonctionnalités attendues et qui pilotent le projet.
Khalid FARHI	Manager	C'est le patron de l'équipe, il s'assure qu'elle dispose des moyens nécessaires à son fonctionnement.
Said MOUNAJI	Chef de projet	C'est un coordonnateur de l'équipe.
Radouane ELARCHA Ayoub MKHARBACH	Développeur	Ce sont les personnes qui seront chargés de concevoir et développer les besoins du client.

Tableau 1 : L'équipe de travail.

Chapitre 2 :

Phase d'inception.

« Tout l'intérêt se trouve dans le commencement après le commencement c'est déjà la fin. », Pablo Picasso.

Introduction :

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d'identifier les cas d'utilisation initiaux. Ceci dit, notre description va sembler trop détaillée pour une première phase de processus.

I. Capture des besoins :

1. Définition des besoins fonctionnels :

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

Le système à concevoir doit permettre à l'utilisateur d'effectuer les opérations suivantes :

Référence	Fonction
R1	Gérer les entités/services
R2	Gérer les utilisateurs
R3	Gérer les comptes
R4	Gérer les demandes
R5	Gérer les pièces
R6	Gérer les champs
R7	Gérer les motifs de rejets
R8	Gérer les intervenants
R9	Gérer les processus
R10	Gérer les référentiels d'adresses
R11	Gérer les commissions
R12	Gérer les architectes
R13	Gérer les topographes
R14	Planifier les réunions
R15	Enregistrer de la demande
R16	Exploiter la demande
R17	Planifier une demande
R18	Suivre une demande
R19	Inscrire
R20	Consulter le guide
R21	Consulter l'état d'avancement des demandes

Tableau 2 : Tableau des exigences.

2. Les besoins non fonctionnels :

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des concepteurs des architectures dynamiques.

Notre application doit nécessairement assurer ces besoins :

L'extensibilité :

Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

La sécurité :

L'application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde, c'est-à-dire qu'il existe des fonctionnalités accessibles par un identifiant et un mot de passe attribué à une personne physique.

L'interface :

Avoir une application qui respecte les principes des IHM¹⁰ tels que l'ergonomie et la fiabilité.

La performance :

L'application devra être performante c'est-à-dire que le système doit réagir dans un délai précis, quel que soit l'action de l'utilisateur.

La convivialité :

L'application doit être simple et facile à manipuler même par des non experts.

¹⁰ IHM : Moyens (matériels ou logiciels) prévus pour la communication entre un être humain et une machine.

3. Regroupement des exigences¹¹ par intentions d'acteurs :

Référence	Fonction	Intention d'acteur	Acteur
R1	Gérer les entités/services	Gérer les entités/services	Ressources humaines
R2	Gérer les utilisateurs	Gérer les utilisateurs	Ressources humaines
R3	Gérer les comptes	Gérer les comptes	Gestionnaire technique
R4	Gérer les demandes	Gérer les demandes	Gestionnaire fonctionnel
R5	Gérer les pièces	Gérer les demandes	Gestionnaire fonctionnel
R6	Gérer les champs	Gérer les demandes	Gestionnaire fonctionnel
R7	Gérer les motifs de rejets	Gérer les demandes	Gestionnaire fonctionnel
R8	Gérer les intervenants	Gérer les demandes	Gestionnaire fonctionnel
R9	Gérer les processus	Gérer les processus	Organisateur
R10	Gérer les référentiels d'adresses	Gérer les référentiels d'adresses	Gestionnaire fonctionnel
R11	Gérer les commissions	Gérer les commissions	Gestionnaire fonctionnel
R12	Gérer les architectes	Gérer les personnels	Gestionnaire technique
R13	Gérer les topographes	Gérer les personnels	Gestionnaire technique
R14	Planifier les réunions	Planifier les réunions	Gestionnaire fonctionnel
R15	Enregistrer de la demande	Effectuer ses tâches	Agent d'exécution
R16	Exploiter la demande	Effectuer ses tâches	Agent d'exécution
R17	Planifier une demande	Effectuer ses tâches	Agent d'exécution
R18	Suivre une demande	Effectuer ses tâches	Agent d'exécution
R19	Inscrire	Inscrire	Citoyen
R20	Consulter le guide	Consulter le guide	Citoyen
R21	Consulter l'état d'avancement des demandes	Consulter l'état d'avancement des demandes	Demandeur/ Gestionnaire fonctionnel

Tableau 3 : Tableau des exigences par intention d'acteur.

¹¹ Exigence : Une exigence est un besoin ou une attente qu'une entité exige sur une autre de manière explicite ou implicite, dans notre cas les exigences sont les fonctionnalités attendues par les utilisateurs du système.

II. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation décrit les utilisations requises d'un système, ou ce qu'un système est supposé faire. Les principaux concepts de ce diagramme sont les acteurs, cas d'utilisation et sujets. Un sujet représente un système avec lequel les acteurs et autres sujets interagissent.

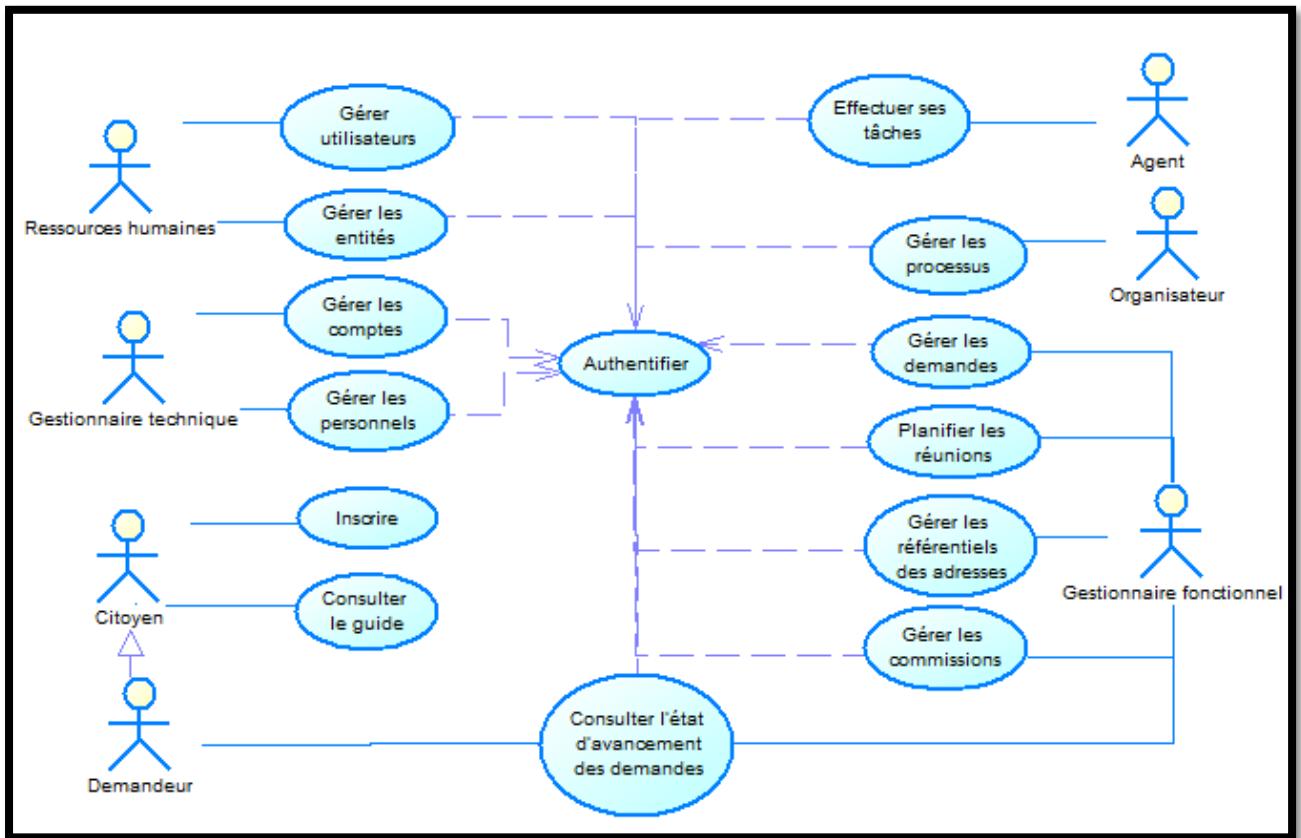


Figure 6 : Diagramme de cas d'utilisation.

Chapitre 3 :

Phase d'analyse et de conception.

« Qui cherche la perfection obtient l'excellence. », Jori Cazilhac.

Introduction :

Cette phase poursuit la tâche entamée dans la phase d'inception, cette phase vise à identifier et décrire la majorité des besoins utilisateurs, construire l'architecture de base du système et lever les risques majeurs du projet.

Au niveau de cette phase, nous analysons les différents cas d'utilisation puis nous essayons de les détaillées pour comprendre chaque fonctionnalité attendue du système en examinant toutes les scénarios compréhensifs des scénarios exceptionnels et des erreurs.

I. Description textuelle des cas d'utilisation :

Pour rendre notre diagramme des cas d'utilisation plus lisible et afin de décrire le comportement d'un système, les concepteurs d'UML proposent l'utilisation d'une technique nommée la description textuelle des cas d'utilisation.

1. Authentifier :

Nom de cas d'utilisation :	Authentifier.	
Objectif :	Accès aux fonctionnalités attribuées.	
Acteurs principaux :	Ressources humaines, Organisateur, Gestionnaire technique, Gestionnaire fonctionnel, l'agent et demandeur.	
Rôle de cas d'utilisation	Ce scénario permet aux utilisateurs de s'identifier auprès du système et d'accéder aux fonctionnalités qui lui sont attribuées.	
Pré-condition :	Existence des données dans la base de données.	
Post-condition :	Affichage de la page désirée.	
Scénario nominal :	Action des acteurs	Action du système
	1. Consulter page d'authentification.	2. Afficher page d'authentification.
	3. Introduire Login et Mot de passe.	4. Vérifier Login et mot de passe. 5. Valider et afficher la page désirée.

Tableau 4 : Description textuelle de cas d'utilisation « Authentifier ».

Les enchainements alternatifs :

A1 : Login ou Mot de passe erroné

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

5-le système indique à l'utilisateur que le login ou le mot de passe est erroné.

Le scénario nominal reprend au point 3.

A2 : Les champs de saisie sont vides

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

5-le système demande aux utilisateurs de remplir les champs.

Le scénario nominal reprend au point 3.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

2. Consulter le guide :

Nom de cas d'utilisation :	Consulter le guide.	
Objectif :	Consulter les guides d'effectuation des demandes.	
Acteurs principaux :	Citoyen.	
Rôle de cas d'utilisation	Ce scénario permet aux citoyens de consulter les réglementations en vigueur et les démarches administratives.	
Pré-condition :	-	
Post-condition :	Les guides sont bien affichées.	
Scénario nominal :	Action des acteurs	Action du système
	1. Demander la page de guide.	2. Afficher les différents guides existants.
	3. Choisir le guide désiré.	4. Afficher les informations nécessaires.

Tableau 5 : Description textuelle de cas d'utilisation « Consulter le guide ».

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

3. Incrire :

Nom de cas d'utilisation :	Inscrire.	
Objectif :	Inscrire à la plateforme.	
Acteurs principaux :	Citoyen.	
Rôle de cas d'utilisation	Ce scénario permet aux citoyens d'inscrire et de créer un compte.	
Pré-condition :	-	
Post-condition :	Création du compte pour le citoyen.	
Scénario nominal :	Action des acteurs	Action du système
	1. Demander la page d'inscription.	2. Afficher le formulaire.
	3. Remplir les champs.	4. Vérifier les données saisies. 5. Enregistrer dans la base de données. 6. Afficher un message de succès.

Tableau 6 : Description textuelle de cas d'utilisation « Incrire ».

Les enchainements alternatifs :

A1 : Données manquantes

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

5- le système affiche un message d'erreur.

Le scénario nominal reprend au point 4.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

4. Consulter l'état d'avancement d'une demande :

Nom de cas d'utilisation :	Consulter l'état d'avancement des demandes.	
Objectif :	Effectuer un suivi des demandes.	
Acteurs principaux :	Demandeur/ Gestionnaire fonctionnel.	
Rôle de cas d'utilisation	Ce scénario permet au demandeur de savoir l'état d'avancement de ses demandes et le gestionnaire de savoir l'état d'avancement de toutes les demandes.	
Pré-condition :	Authentification préalable.	
Post-condition :	Les demandes sont bien affichées.	
Scénario nominal :	Action des acteurs	Action du système
	1. Authentification	3. Afficher les demandes.
	2. Demande la liste des demandes.	4. Choisir une demande.
		5. Afficher l'état d'avancement de la demande.

Tableau 7 : Description textuelle de cas d'utilisation « Consulter l'état d'avancement d'une demande ».

Les enchainements alternatifs :

A1 : Aucune demande trouvée

L'enchainement A1 démarre au point 3 du scénario nominal.

4- le système arrête le scénario nominal.

Le scénario s'arrête.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

6. Gérer :

- Ajouter :

Nom de cas d'utilisation :	Ajouter.	
Objectif :	Effectuer une gestion des ressources.	
Acteurs principaux :	Gestionnaire fonctionnel / Organisateur / Ressource humain.	
Rôle de cas d'utilisation	Ce scénario permet à l'utilisateur d'ajouter une ressource.	
Pré-condition :	Authentification préalable.	
Post-condition :	Une nouvelle ressource est bien ajoutée.	
Scénario nominal :	Action des acteurs	Action du système
	1. Authentification. 2. Demander le formulaire d'ajout.	3. Afficher le formulaire.
	4. Remplir les champs.	5. Vérifier les données saisies. 6. Enregistrer dans la base de données. 7. Afficher un message de succès.

Tableau 8 : Description textuelle de cas d'utilisation « Ajouter ».

Les enchainements alternatifs :

A1 : Données manquantes

L'enchainement A1 démarre au point 5 du scénario nominal.

6- le système affiche un message d'erreur.

Le scénario nominal reprend au point 4.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

- **Modifier :**

Nom de cas d'utilisation :	Modifier.	
Objectif :	Effectuer une gestion des ressources.	
Acteurs principaux :	Gestionnaire fonctionnel / Organisateur / Ressource humain.	
Rôle de cas d'utilisation	Ce scénario permet au gestionnaire de modifier une ressource.	
Pré-condition :	Authentification préalable.	
Post-condition :	Une ressource est bien modifiée.	
Scénario nominal :	Action des acteurs	Action du système
	1. Authentification. 2. Sélectionner une ressource.	3. Afficher le formulaire avec les informations nécessaires.
	4. Modifier les champs.	5. Vérifier les données saisies. 6. Mise à jour de la base de données. 7. Afficher un message de succès.

Tableau 9 : Description textuelle de cas d'utilisation « Modifier ».

Les enchainements alternatifs :

A1 : Aucune ressource trouvée

L'enchainement A1 démarre au point 2 du scénario nominal.

3- le système arrête le scénario nominal.

Le scénario s'arrête.

A2 : Données manquants

L'enchaînement A2 démarre au point 5 du scénario nominal.

6- le système affiche un message d'erreur.

Le scénario nominal reprend au point 4.

Les enchaînements d'exception :

E1 : Le système ne répond pas

L'enchaînement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

- Supprimer :

Nom de cas d'utilisation :	Supprimer.	
Objectif :	Effectuer une gestion des ressources.	
Acteurs principaux :	Gestionnaire fonctionnel / Organisateur / Ressource humain.	
Rôle de cas d'utilisation	Ce scénario permet au gestionnaire de supprimer une ressource.	
Pré-condition :	Authentification préalable.	
Post-condition :	Une ressource est bien supprimée et archivée.	
Scénario nominal :	Action des acteurs	Action du système
	1. Authentification. 2. Sélectionner une ressource.	3. Afficher un message de confirmation.
	4. Valider son choix.	5. Supprimer la ressource. 6. Afficher un message de succès.

Tableau 10 : Description textuelle de cas d'utilisation « Supprimer ».

Les enchainements alternatifs :

A1 : Aucune ressource trouvée

L'enchainement A1 démarre au point 2 du scénario nominal.

4- le système arrête le scénario nominal.

Le scénario s'arrête.

A2 : L'utilisateur annule la suppression

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

5- le système affiche arrête le scénario nominal.

Le scénario s'arrête.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

7. Planifier les réunions :

Nom de cas d'utilisation :	Planifier les réunions.	
Objectif :	Effectuer un suivi des demandes.	
Acteurs principaux :	Gestionnaire fonctionnel.	
Rôle de cas d'utilisation	Ce scénario permet au gestionnaire fonctionnel de planifier les réunions des commissions.	
Pré-condition :	Authentification préalable.	
Post-condition :	Les demandes sont bien affichées.	
Scénario nominal :	Action des acteurs	Action du système
	1. Authentification	2. Afficher la page de planification.
	3. Planifier une réunion pour une commission.	4. Afficher un message de succès.

Tableau 11 : Description textuelle de cas d'utilisation « Planifier une réunion ».

Les enchainements alternatifs :

A1 : Données manquants

L'enchainement A1 démarre au point 3 du scénario nominal.

4- le système affiche un message d'erreur.

Le scénario nominal reprend au point 3.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

8. Effectuer ses tâches :

Nom de cas d'utilisation :	Effectuer ses tâches.	
Objectif :	Effectuer une tâche.	
Acteurs principaux :	Agent d'exécution.	
Rôle de cas d'utilisation	Ce scénario permet à l'agent d'exécution d'effectuer une tâche au sein du processus de validation de la demande.	
Pré-condition :	Authentification préalable.	
Post-condition :	Une tâche est bien effectuée.	
Scénario nominal :	Action des acteurs	Action du système
	1. Authentification	2. Afficher la liste des profils.
	3. Choisir un profil.	4. Afficher la page du profil.
	5. Effectuer sa tâche.	6. Enregistrer les modifications. 7. Afficher un message de succès.

Tableau 12 : Description textuelle de cas d'utilisation « Effectuer ses tâches ».

Les enchainements alternatifs :

A1 : Aucune profil relié à l'agent.

L'enchainement A1 démarre au point 2 du scénario nominal.

3- le système arrête le scénario nominal.

Le scénario s'arrête.

Les enchainements d'exception :

E1 : Le système ne répond pas

L'enchainement E1 démarre au point 1 du scénario nominal.

2- le système ne répond pas.

Le cas d'utilisation se termine en échec.

II. Diagramme de séquence « Boite noire » :

Pour schématiser la vue comportementale de notre système informatique, nous faisons recours au diagramme de séquence d'UML. Ce diagramme permet de préciser les interactions entre l'acteur et le système avec des messages présentés dans un ordre chronologique. Le diagramme de séquence système traite le système informatique comme étant une boite noire. Le comportement du système est décrit vu de l'extérieur sans avoir d'idée sur comment il le réalisera.

1. Authentifier :

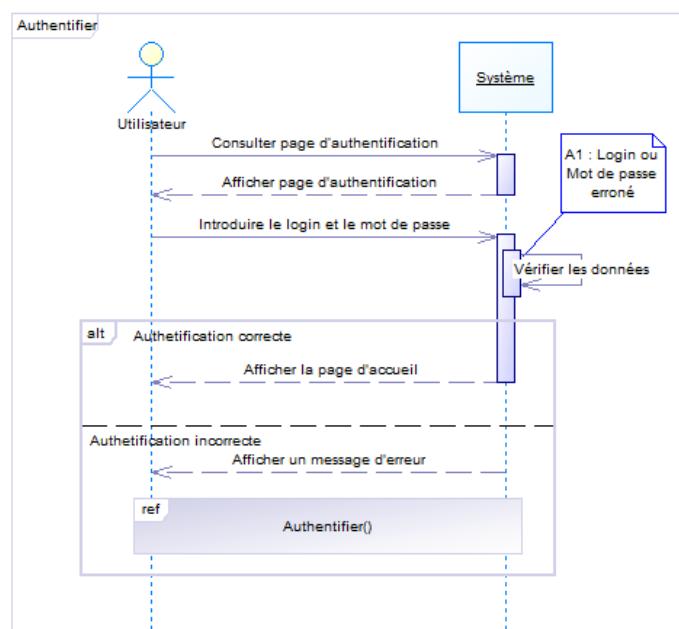


Figure 7 : Diagramme de séquence « Authentifier ».

2. Incrire :

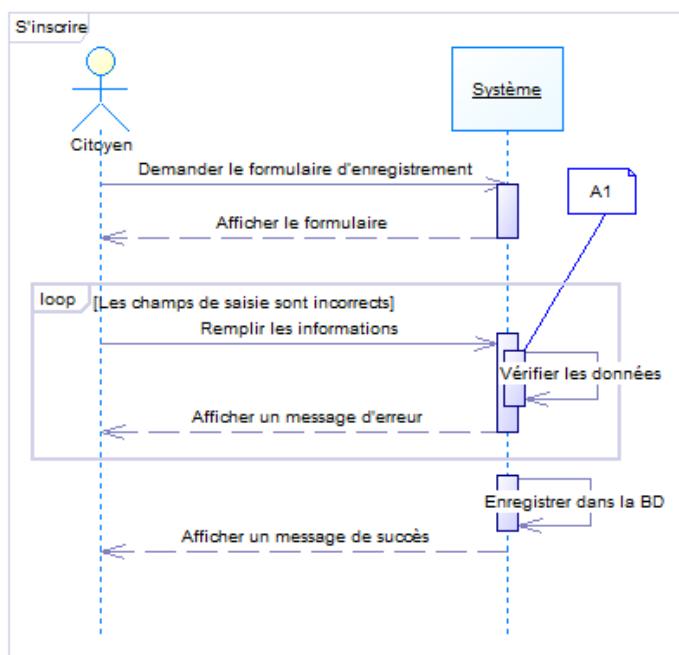


Figure 8 : Diagramme de séquence « Incrire ».

3. Consulter le guide :

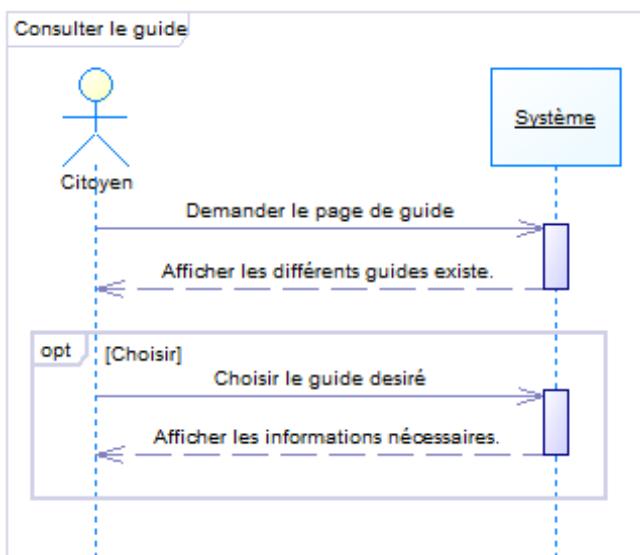


Figure 9 : Diagramme de séquence « Consulter le guide ».

4. Consulter l'état d'avancement d'une demande :

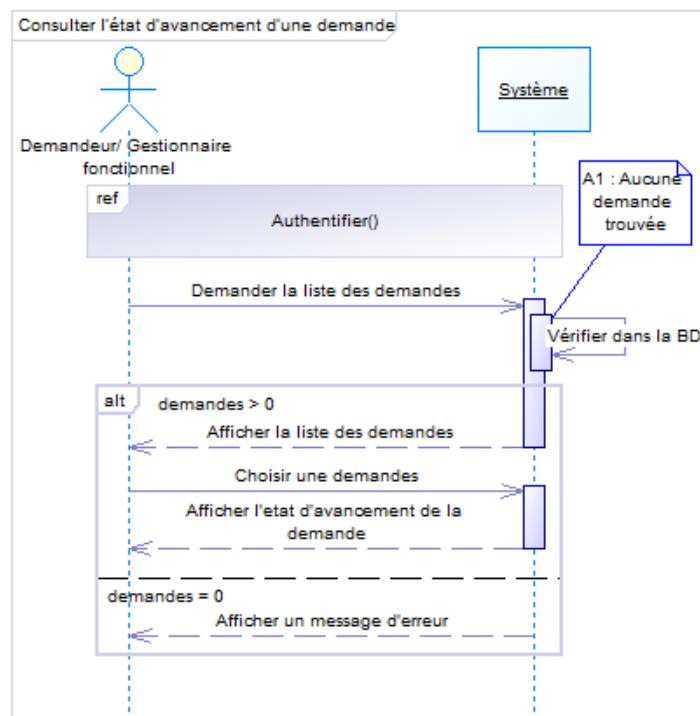


Figure 10 : Diagramme de séquence « Consulter l'état d'avancement d'une demande ».

5. Gérer :

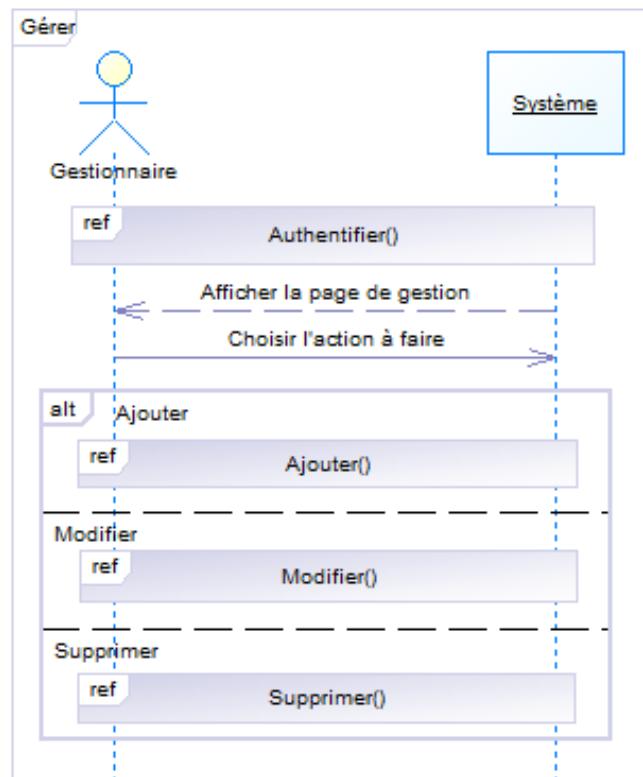


Figure 11 : Diagramme de séquence « Gérer ».

6. Planifier les réunions :

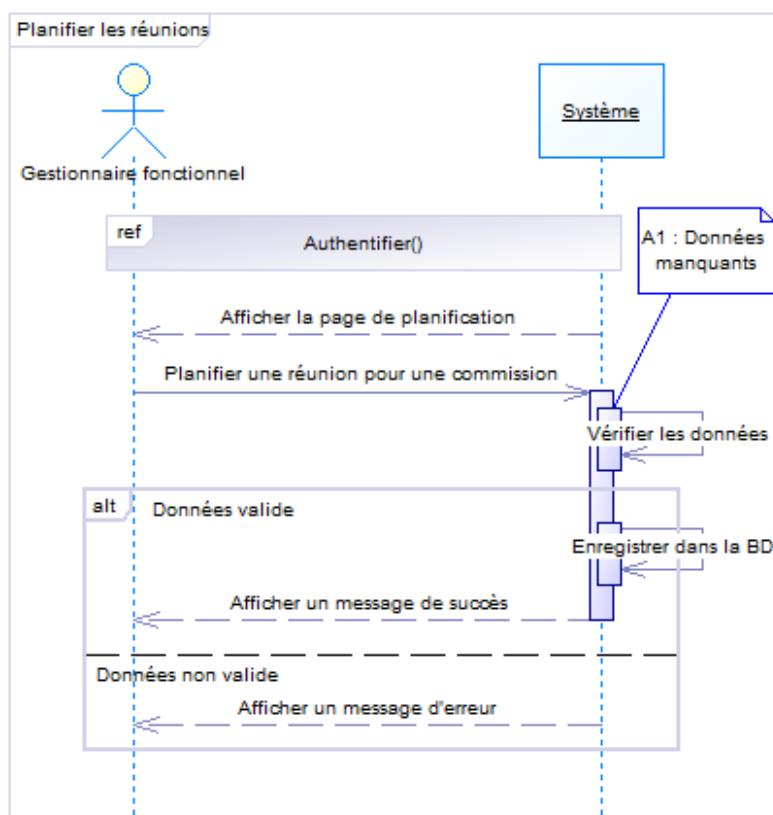


Figure 12 : Diagramme de séquence « Planifier les réunions ».

7. Effectuer ses tâches :

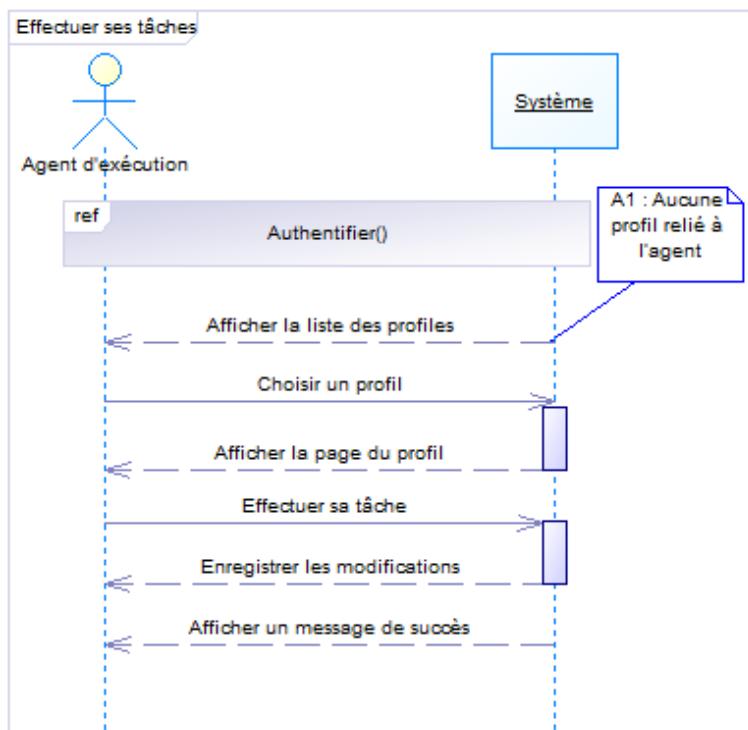


Figure 13 : Diagramme de séquence « Effectuer ses tâches ».

III. Diagramme de classe d'analyse :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation. Une classe-association possède les caractéristiques des associations et des classes : elle se connecte à deux ou plusieurs classes et possède également des attributs et des opérations. Une classe-association est caractérisée par un trait discontinu entre la classe et l'association.

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations ou méthodes) et de relations avec d'autres objets (les associations et les agrégations).

Une classe de conception est composée par :

Attribut

Chaque attribut d'une classe est le même pour chaque instance de cette classe.

Méthodes

Elle définit le comportement d'une classe elle-même, et non le comportement de ses instances qui peut être différent.

Pour assurer la lisibilité de notre diagramme de classe, nous avons le découpé sous forme des modules comme ceci :

1. Gestion des processus :

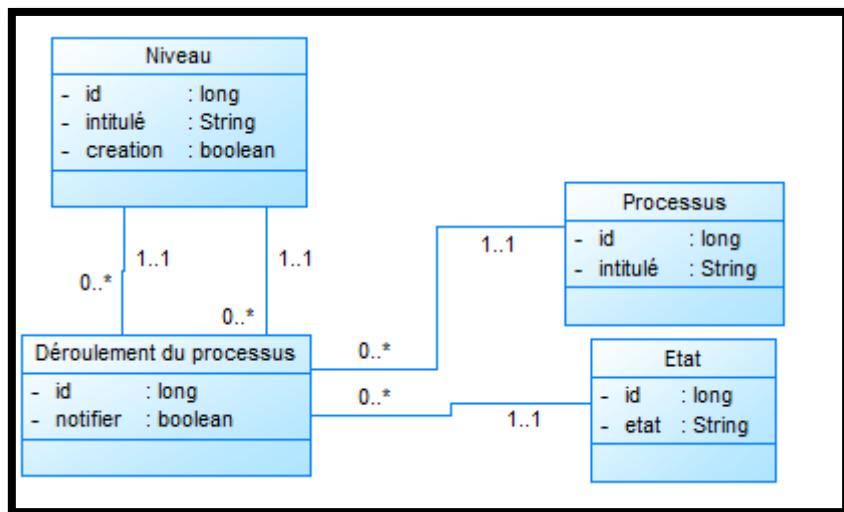


Figure 14 : Diagramme de classe « Gestion de processus ».

2. Gestion des demandes :

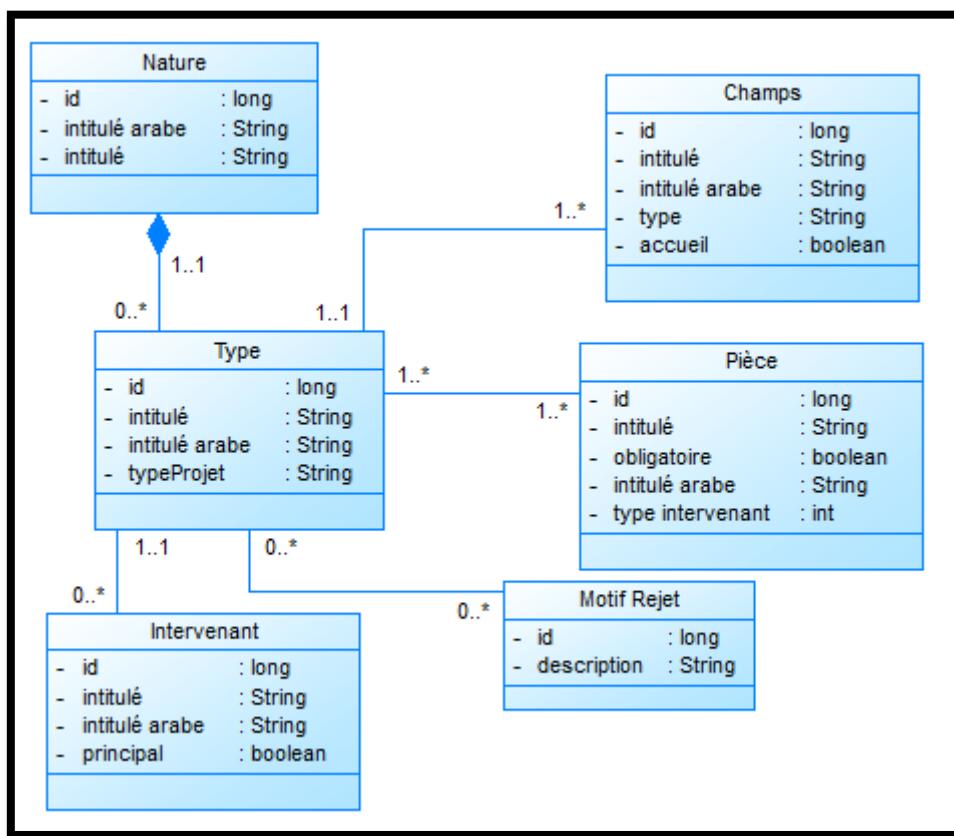


Figure 15 : Diagramme de classe « Gestion des demandes ».

3. Gestion des utilisateurs :

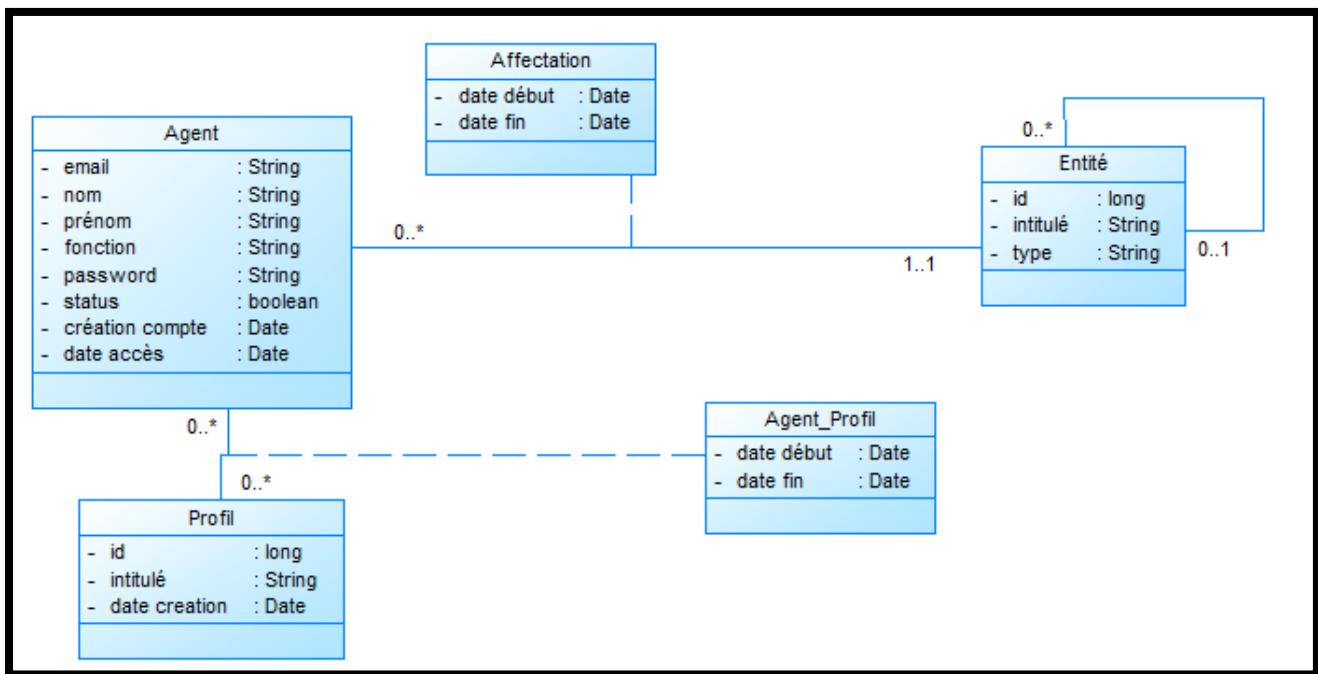


Figure 16 : Diagramme de classe « Gestion des utilisateurs ».

4. Gestion des référentiels d'adresses :

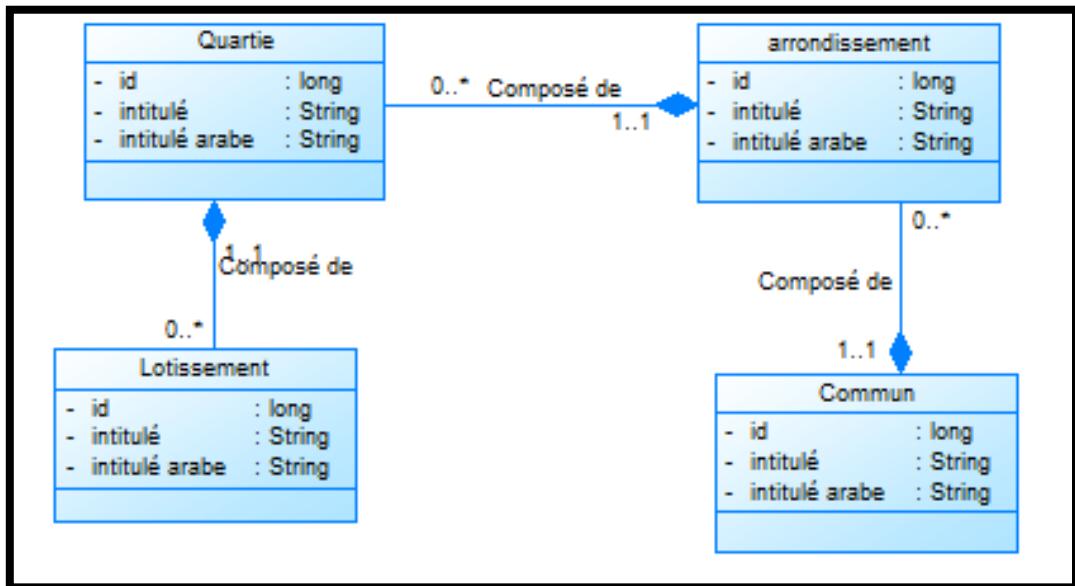


Figure 17 : Diagramme de classe « Gestion des référentiels d'adresses ».

5. Gestion des intervenants :

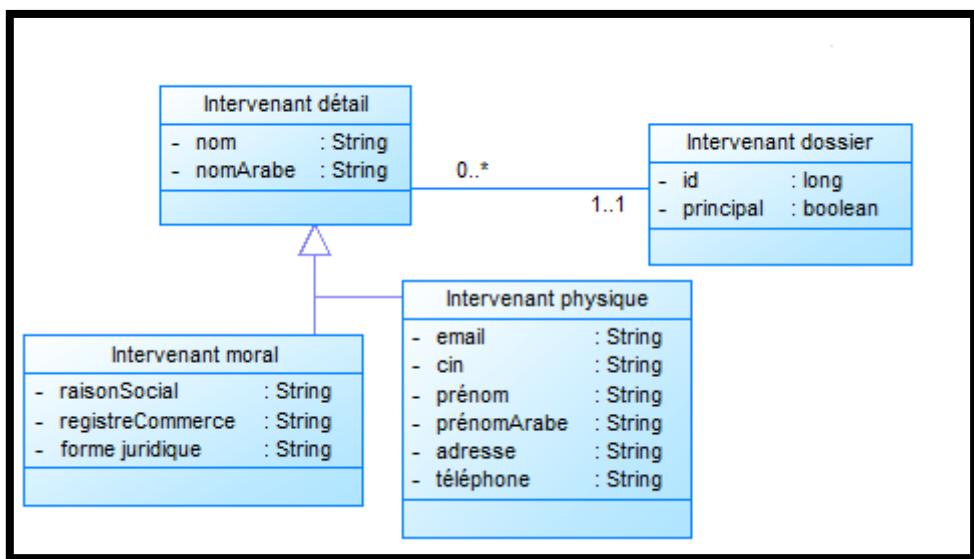


Figure 18 : Diagramme de classe « Gestion des intervenants ».

6. Gestion des dossiers :

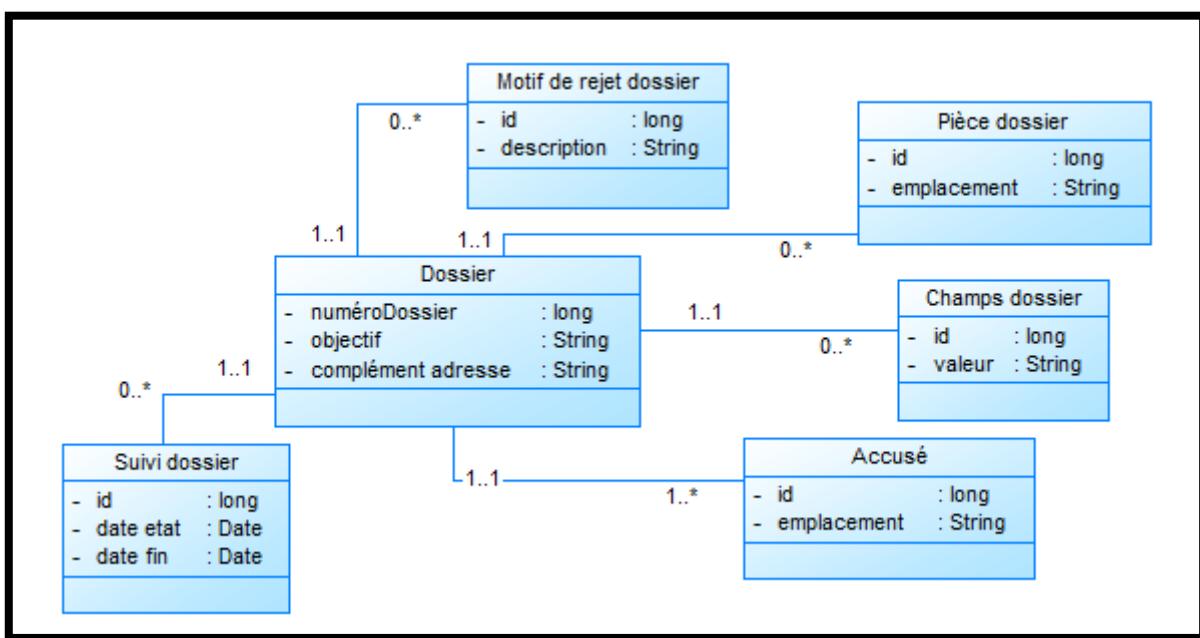


Figure 19 : Diagramme de classe « Gestion des dossiers ».

7. Gestion des réunions :

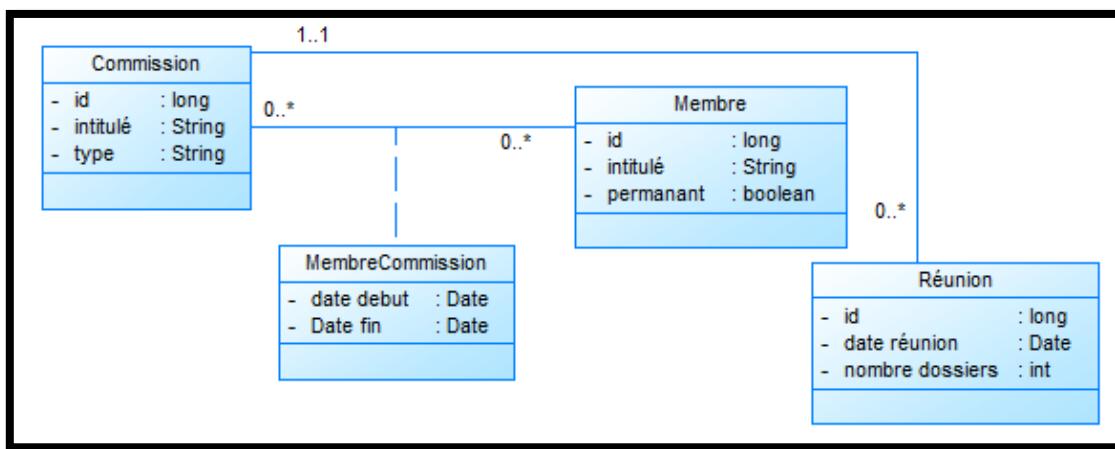


Figure 20 : Diagramme de classe « Gestion des réunions ».

8. Gestion des emails :

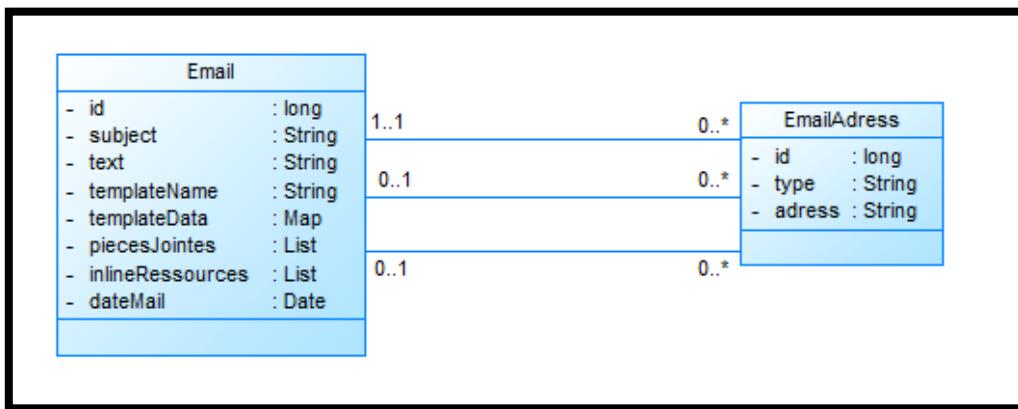


Figure 21 : Diagramme de classe « Gestion des emails ».

9. Gestion des traçabilités :

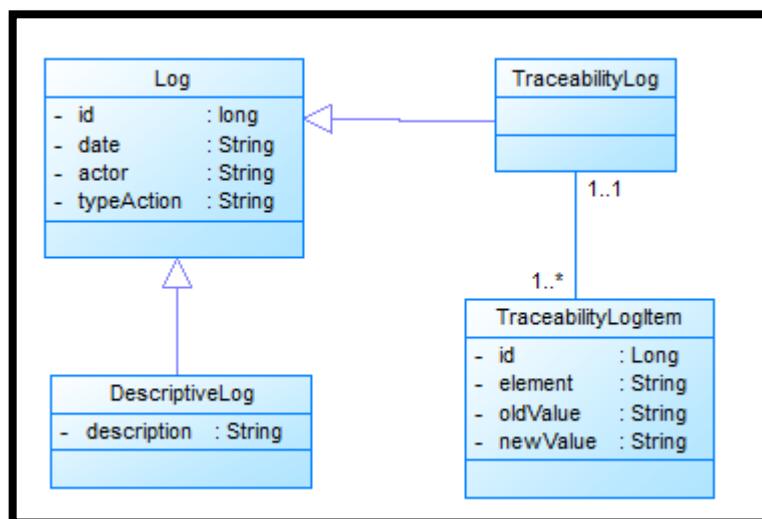


Figure 22 : Diagramme de classe « Gestion des traçabilité ».

IV. Prototype des interfaces :

Dans le domaine du web, une technique est apparue et prend une place très importante dans le développement des applications Web, il s'agit de prototypage. Cette technique consiste préparer quelques interfaces graphiques de l'application en utilisant un outil de conception de prototypage afin de mesurer le degré de satisfaction du client par rapport à la compréhension du projet par le développeur. L'interaction qui se produit entre l'utilisateur final et le développeur, à la suite de la discussion sur ces interfaces, permet d'ajuster les besoins et de les concevoir de manière précise et exacte. En effet les interfaces graphiques font que l'utilisateur final soit plus interactif, précis et le poussent à mieux s'exprimer quant à ses attentes. Ci-dessus quelques interfaces réalisées avec l'outil Balsamiq Mockups ¹².

1. Interface de gestion :

The screenshot shows a web browser window titled 'A Web Page' with the URL 'http://localhost:8080/ressource'. The page has a header with a logo (a square with an 'X' and 'Logo' in the center), user information ('Ayoub MKHARBACH' and 'Logout'), and a dropdown menu 'Choisir un profil'. On the left, there's a sidebar with a tree view containing nodes like 'Gestion des processus', 'Gestion des demandes', etc. The main content area is titled 'Liste des ressources' and contains a search bar 'Rechercher un ressource' and a button 'Ajouter un ressource'. Below is a table with columns 'Id', 'Nom', 'Champs 1', 'Champs 2', and 'Opération'. The table has 7 rows, each representing a resource with values A-N and '(M) (S)' in the operation column. At the bottom is a navigation bar with numbers 1-9.

ID	Nom	Champs 1	Champs 2	Opération
1	Ressource 1	A	B	(M) (S)
2	Ressource 2	C	D	(M) (S)
3	Ressource 3	E	F	(M) (S)
4	Ressource 4	G	H	(M) (S)
5	Ressource 5	I	J	(M) (S)
6	Ressource 6	K	L	(M) (S)
7	Ressource 7	M	N	(M) (S)

Figure 23 : Prototype de l'interface de gestion.

¹² Balsamiq Mockups : est un outil permettant de créer facilement des prototypes d'IHM électronique. Avec Balsamiq Mockups il est ainsi possible de prototyper tout type d'applications (desktop, web, smartphone, ...).

2. Interface de d'accueil :

The screenshot shows a web browser window titled "A Web Page" with the URL <http://localhost:8080/accueil>. The page has a header with a logo, user information (Ayoub MKHARBACH, Logout), and a dropdown menu (Accueil). On the left is a vertical sidebar with a logo and a list of management modules: Gestion des processus (highlighted in blue), Gestion des demandes, Gestion des intervenants, Gestion des entités, Gestion des utilisateurs, Gestion des comptes, Gestion des personnels, Gestion des commissions, and Gestion des référentiels. The main content area contains three sections: "Informations sur le dossier" with fields for "Nature de demande" and "Type de demande"; "Informations sur l'intervenant" with fields for "Nom de l'intervenant" and "Prénom de l'intervenant"; and "Validation du dossier" with a checkbox group for "Etat 1", "Etat 2" (checked), and "Etat 3". A "Enregistrer" button is at the bottom.

Figure 24 : Prototype de l'interface d'accueil.

3. Interface de suivi :

The screenshot shows a web browser window titled "A Web Page" with the URL <http://localhost:8080/suivi>. The layout is similar to Figure 24, with a header, sidebar, and main content area. The sidebar highlights "Gestion des processus". The main content includes a search bar ("Rechercher un dossier"), a file upload field ("Document de sortie", "Choisir un fichier: Upload file ..."), and a validation section ("Validation du dossier", "Choisir l'état de sortie: Etat 1, Etat 2 (checked), Etat 3"). An "Enregistrer" button is at the bottom.

Figure 25 : Prototype de l'interface de suivi.

Chapitre 4 :

Phase de réalisation.

« Chaque bonne réalisation, grande ou petite, connaît ses périodes de corvée et de triomphe ; un début, un combat et une victoire. », Ghandi.

Introduction :

Cette phase est la dernière phase dans le cycle de développement d'un logiciel avec la démarche UP-XP. Cette phase est souvent appelée phase d'implémentation. Les tâches effectuées pendant cette phase ne sont pas claires, et ils dépendent fortement du type de déploiement du logiciel.

Pour notre projet, ce chapitre sera consacré pour la présentation des langages et outils de programmation utilisés pour la réalisation de notre application, et nous finirons par quelques interfaces graphiques du logiciel fourni.

I. Développement :

1. Choix de développement :

Pour le développement, nous avons appliqué le modèle MVC.

Ce paradigme divise l'IHM (Interface Homme Machine) en un modèle (M pour modèle de données) une vue (V pour la présentation, l'interface utilisateur) et un contrôleur (C pour la logique de contrôle, et la gestion des événements / synchronisation), chacun ayant un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. L'architecture MVC ne résout pas tous les problèmes. Elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Ce patron d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

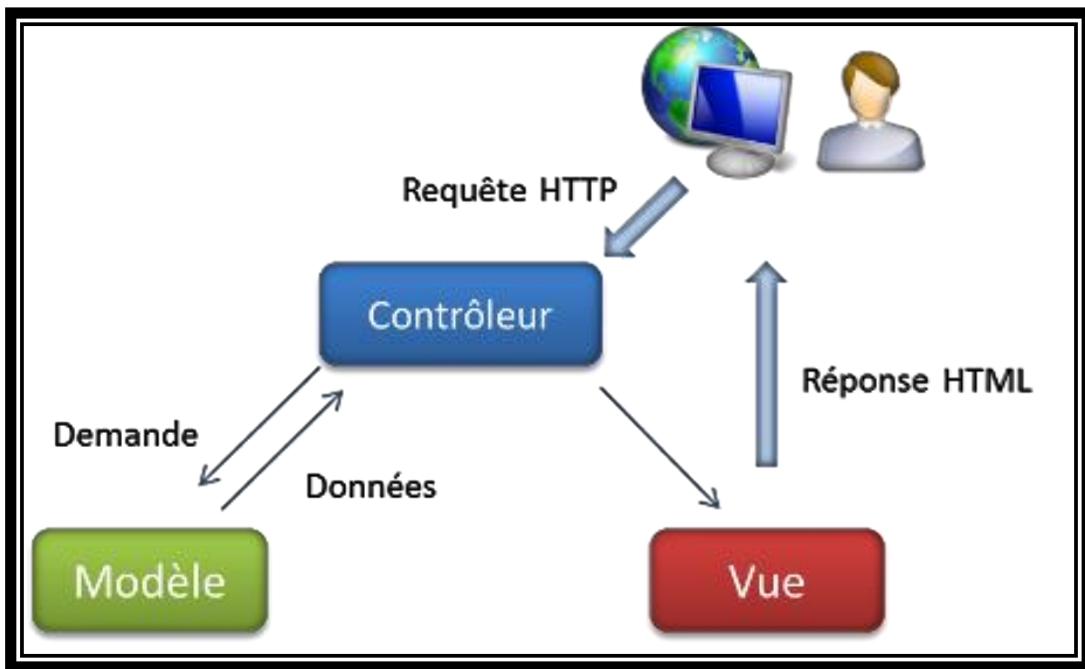


Figure 26 : Modèle MVC (Modèle – Vue – Contrôleur).

Le modèle :

Le modèle représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation.

La vue :

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

Le contrôleur :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle, et ce dernier notifie la vue que les données ont changée pour qu'elle les mette à jour.

2. Choix de la plateforme Java EE :

Pour le développement de l'application, nous avons utilisé la plateforme Java EE (Java Edition Entreprise).

Parmi les avantages de la plateforme Java EE, on peut citer :

Technologie sans frais :

Java (sur lequel est basé Java EE) est une technologie Open Source¹³, les outils de développement JEE sont disponibles gratuitement.

Maintenabilité :

Les applications Java EE sont plus faciles à entretenir, dans la plupart des cas, ils sont conçus en plusieurs couches. Il est facile d'ajouter de nouvelles fonctionnalités tierces pour les applications Java EE en raison de sa fonction d'évolutivité.

Indépendance :

Les applications développées avec Java EE peuvent être déployées sur la plupart des matériels disponibles. Elles offrent une flexibilité de matériel à l'utilisateur final. Ainsi, l'utilisateur peut déployer et exécuter des applications Java EE sur le système d'exploitation et le matériel de son choix.

¹³ Open source : s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. Mis à la disposition du grand public

3. L'architecture :

L'architecture utilisé tout au long du développement de l'application est divisé en trois parties.

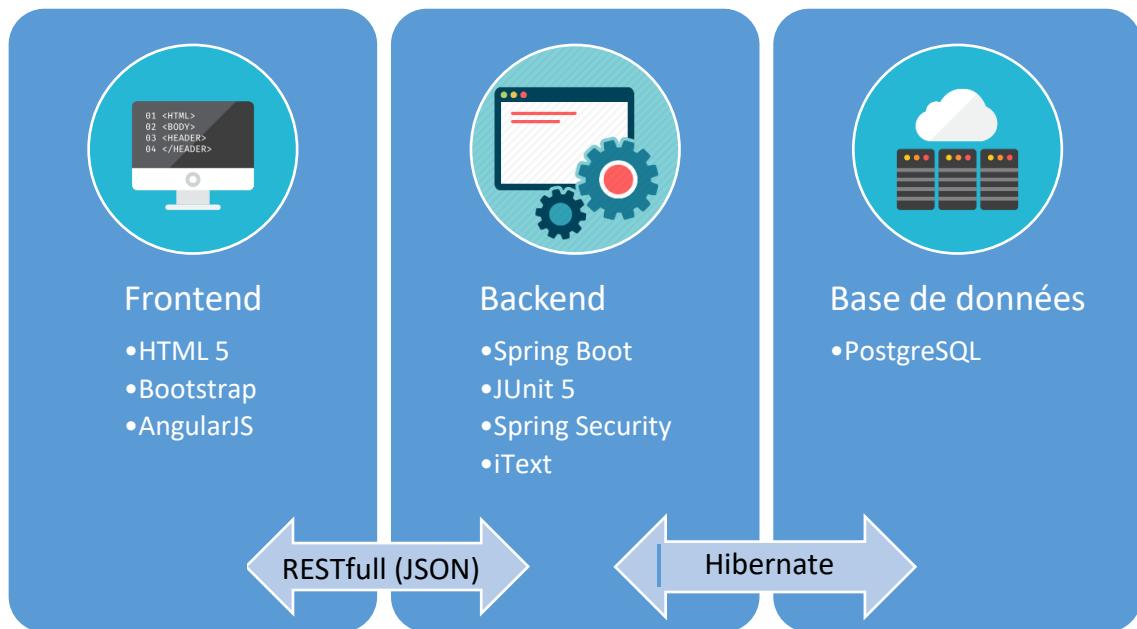


Figure 27 : Architecture de travail.

Frontend :

C'est la partie visible d'un site web. Autrement dit, tout ce que le visiteur télécharge depuis le serveur lorsqu'il charge une page. On y trouve donc le HTML, le CSS¹⁴ et bien sûr le Javascript¹⁵.

Dans notre application nous avons utilisé les langages et les Framework suivants :

a. HTML 5 :



HTML (Hyper Text Markup Language / langage hypertexte) est le langage dans lequel sont écrites les pages du web. Un site web est constitué d'un ou plusieurs documents HTML, appelées aussi pages.

¹⁴CSS : est un langage de feuille de style utilisé afin de décrire la présentation d'un document écrit en HTML. CSS décrit la façon dont les éléments doivent être affichés, à l'écran, sur du papier ou sur autre support.

¹⁵JavaScript : est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

Le HTML n'est pas un langage de programmation comme le C++¹⁶. Les langages dynamiques comme JavaScript vont d'ailleurs générer des pages HTML statiques.

b. BOOTSTRAP :



Bootstrap

Bootstrap est une collection d'outils utile à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

Bootstrap rend le développement web front-end plus rapide et plus simple. Il est conçu pour des personnes de tous niveaux, des appareils de toutes formes et des projets de toutes tailles.

c. ANGULARJS :



AngularJS est un framework¹⁷ JavaScript libre et open source développé par Google. AngularJS est fondé sur l'extension du langage HTML par de nouvelles balises (tags) et attributs pour aboutir à une définition déclarative des pages web, par opposition à l'utilisation systématique de l'élément div et à la définition des éléments de présentation en JavaScript.

Le code HTML étendu représente alors la partie « vue » du patron d'architecture MVC auquel AngularJS correspond, via des modèles en couche appelés « scopes » et des contrôleurs permettant de prototyper des actions en code JavaScript natif. AngularJS utilise une boucle de dirty-checking (qui consiste à surveiller et à détecter des modifications sur un objet JavaScript) pour réaliser un data-binding bidirectionnel permettant la synchronisation des modèles et des vues.

¹⁶ C++ : est un langage de programmation compilé, permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique.

¹⁷ Framework : Un framework désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. C'est le framework, ou structure logicielle en français, qui établit les fondations d'un logiciel ou son squelette.

Backend :

a. SPRING BOOT :



Spring Boot est un nouveau framework créé par l'équipe de chez Pivotal, conçu pour simplifier le démarrage et le développement de nouvelles applications Spring. Le framework propose une approche dogmatique de la configuration, qui permet d'éviter aux développeurs de redéfinir la même configuration à plusieurs endroits du code. Dans ce sens, Boot se veut d'être un acteur majeur dans le secteur croissant du développement d'applications rapide.

La plateforme Spring IO a vivement été critiquée au fil des années à cause de la configuration XML¹⁸ volumineuse et d'une gestion des dépendances complexe.

Spring boot offre une expérience de développement ne nécessitant aucune configuration XML. Boot relève le défi à l'extrême, ne se contentant pas uniquement de décharger les développeurs de l'utilisation de XML, mais aussi, dans certains cas, de leur permettre de se soulager de certains imports fastidieux.

b. JUNIT :

JUnit JUnit est un framework open source pour le développement et l'exécution de tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non-régression.

JUnit a été initialement développé par Erich Gamma et Kent Beck. Il propose :

- ✓ Un framework pour le développement des tests unitaires reposant sur des assertions qui testent les résultats attendus
- ✓ Des applications pour permettre l'exécution des tests et afficher les résultats.

¹⁸ XML : est un langage informatique de balisage générique, était utilisé par le framework Spring pour définir ses configurations.

c. SPRING SECURITY :



Spring Security est un module incontournable d'une application développée en Spring. Il apporte tout le nécessaire pour sécuriser une application et il a l'avantage d'être vraiment personnalisable. La notion de sécurité informatique n'est pas une mince affaire et sa

mise en place est parfois longue et demande d'être constamment adaptée au niveau réseau, serveurs, application... Spring Security n'intervient que sur le domaine applicatif.

Quand on parle sécurité applicative, nous avons deux notions essentielles : l'authentification (savoir qui je suis) et les autorisations (savoir ce que j'ai le droit de faire). Spring Security essaye d'apporter une solution à ces deux problématiques.

a. ITEXT :



iText est une API¹⁹ open source en java, grâce à iText vous aurez la possibilité de créer des PDFs avec votre code java. Vous pourrez faire tout ce qui vous plaira en ce qui concerne la mise en page de vos documents, tout en utilisant un format qui est largement répandu et utilisé par tous. Il est distribué sous licence AGPL²⁰ et propriétaire pour certaines parties.

Base de données :



PostgreSQL est un SGBDRO. C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MySQL

et Firebird), ou propriétaires (comme Oracle, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

¹⁹ API : En informatique, est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

²⁰ AGPL : Basée sur la licence GPL, elle répond à un besoin du projet Affero, qui souhaite que tout opérateur d'un service Web utilisant leur logiciel et l'améliorant publie ces modifications.

PostgreSQL peut stocker plus de types de données que les types traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type etc.

PostgreSQL fonctionne sur toutes sortes d'Unix²¹. Depuis la version 8.0, PostgreSQL fonctionne également nativement sur Windows. Avant la version 8.

PostgreSQL est largement reconnu pour son comportement stable, proche d'Oracle. Mais aussi pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/SQL²². Le traitement interne des données peut aussi être couplé à d'autres modules externes compilés dans d'autres langages.

Communication entre le frontend et le backend :

a. RESTFULL :



Utilisée par les navigateurs, la technologie REST (REpresentational State Transfer) peut être considérée comme la langue d'Internet.

Aujourd'hui, l'utilisation croissante du Cloud entraîne l'apparition de différentes API visant à présenter les services Web. REST constitue un

choix logique pour le développement d'API permettant à l'utilisateur final de se connecter à des services Cloud et d'interagir avec eux. De nombreux sites font appel à des API RESTful, notamment Google, Amazon, Twitter et LinkedIn.

Une API RESTful fragmente une transaction en plusieurs petits modules, chacun traitant une partie sous-jacente spécifique de la transaction.

Les API RESTful tirent explicitement parti des méthodologies HTTP définies par le protocole RFC²³ 2616.

Elles utilisent simplement une requête PUT pour modifier ou mettre à jour l'état d'une ressource (un objet, un fichier ou un bloc), GET pour récupérer une ressource, POST pour créer cette ressource, et DELETE pour la supprimer.

²¹ Unix : est un système d'exploitation multitâche et multi-utilisateur créé en 1969 par Kenneth Thompson. Il repose sur un interpréteur ou superviseur et de nombreux petits utilitaires, accomplissant chacun une action spécifique, commutables entre eux et appelés depuis la ligne de commande.

²² PL/SQL : est un langage, basé sur les paradigmes de programmation procédurale et structurée. Il est propriétaire, créé par Oracle et utilisé dans le cadre de bases de données relationnelles.

²³ RFC : sont une série numérotée de documents officiels décrivant les aspects techniques d'Internet, ou de différents matériels informatiques, RFC 2616 décrit le protocole HTTP.

b. JSON :



JavaScript Object Notation

JSON est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou générable par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué à n'importe quel type de langage. Ces propriétés font de JSON un langage d'échange de données idéal.

Hibernate - Communication entre le backend et la base de données :



Hibernate est un Framework permettant de résoudre le problème d'adaptation entre le paradigme objet et les SGBD.

Pour faire simple, les SGBD sont complexes d'utilisation et contraignant, Hibernate permet au programmeur, d'accéder à la base de données en créant des objets, ce qui simplifie grandement la tâche.

Ce type de technologie peut être appelé Framework de mapping objet-relationnel ou de persistance objet des données.

En effet, la couche applicative voit les données comme des classes dont le contenu reste en mémoire même après la fin d'exécution du programme. D'où persistance objet des données. De plus, le lien entre les classes exposées et la source physique des données (souvent une base de données relationnelle) est définie par un fichier xml. D'où mapping objet-relationnel.

II. Environnement logiciel :

Nous avons énuméré au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

1. IntelliJ IDEA :



IntelliJ IDEA est un IDE, autrement dit un ensemble d'outils destinés au développement logiciel. Contrairement à Eclipse qui est géré par un consortium (Fondation Eclipse), IntelliJ IDEA est édité par une société commerciale (JetBrains) qui en assure le développement et le support depuis janvier 2001.

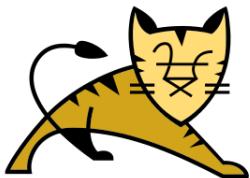
IntelliJ IDEA est décliné en deux éditions :

Community : l'édition communautaire, open source et gratuite, est le choix idéal pour travailler sur des applications pures Java / Groovy²⁴ ou des développements Swing²⁵.

Ultimate : l'édition complète (et payante) permet entre autres le développement d'applications Web JEE.

IDEA n'est pas décliné en plusieurs versions (flavors) comme Eclipse²⁶. JetBrains pousse directement ses différentes versions sur l'IDE et le développeur doit simplement accepter la mise à jour.

2. Apache tomcat :



Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP²⁷ du Java Community Process, est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

3. Maven :



Maven est un outil de gestion de projet qui comprend un modèle objet pour définir un projet, un ensemble de standards, un cycle de vie, et un système de gestion des dépendances. Il embarque aussi la logique nécessaire à l'exécution d'actions pour des phases bien définies de ce cycle de vie, par le biais de plugins. Lorsque vous utilisez Maven, vous décrivez votre projet selon un modèle objet de projet clair, Maven peut alors lui appliquer la logique transverse d'un ensemble de plugins (partagés ou spécifiques).

Il permet notamment :

- ✓ D'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet.
- ✓ De gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet.
- ✓ De générer des documentations concernant le projet.

²⁴ Groovy : est le nom d'un langage de programmation orienté objet destiné à la plate-forme Java. Il constitue une alternative au langage Java pour cette plate-forme et est inspiré de Python, Ruby.

²⁵ Swing : est une bibliothèque graphique pour le langage de programmation Java.

²⁶ Eclipse : est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre, en s'appuyant principalement sur Java.

²⁷ JSP : est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web.

4. GitHub :



GitHub (exploité sous le nom de GitHub, Inc.) est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git²⁸. Ce site est développé par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.

5. Projeqtor :



Projeqtor est un Logiciel de gestion de projet open source, qui regroupe dans un outil unique toutes les fonctionnalités nécessaires à l'organisation de vos projets. Il reste simple, facile à utiliser au quotidien, tout en couvrant un maximum de fonctionnalités de la gestion de projet.

Sa particularité, outre sa complétude, est d'être orienté qualité. Ceci signifie qu'il vous permet d'enregistrer tous les événements de vos projets, et de simplifier de ce fait la mise en conformité aux principaux standards de gestion de la qualité, que ce soit ISO, ITIL ou autre.

²⁸ Git : est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.

III. Les interfaces de l'application :

1. Interface d'authentification :

Cette interface permet aux utilisateurs d'accéder à leurs profils.



Figure 28 : Interface d'authentification.

2. Interface de bienvenue :

Cette interface constitue la fenêtre d'accueil de notre application. Elle contient deux menus de navigation, un permet de gérer les ressources dédiées aux administrateurs et l'autre permet de choisir un profil qui correspondant à un niveau au sein d'un processus.

A screenshot of the application's main interface. At the top, there is a dark header bar with the title "Guichet unique", the user's email "a.mkharbach@upsysmaroc.com", a "Logout" link, and a "Choisir profil" dropdown menu. The main content area has a light gray background. On the left, there is a sidebar with a logo and several navigation links: "Consultation", "Gestion des utilisateurs", "Gestion des entités", "Gestion des processus", "Gestion des types", "Gestion des référentiels", "Gestion des commissions", and "Gestion des réunions". In the center, the word "Bienvenue" is displayed above a card for a user profile. The card for "MKHARBACH Ayoub" shows the following details:

Service :	Service de l'urbanisme
Fonction :	Chef de service
Téléphone :	+212602628486
Email	a.mkharbach@upsysmaroc.com

The time "22:45:29" is shown at the bottom of the card. The overall design is clean and professional.

Figure 29 : Interface de bienvenue.

3. Interface du niveau d'accueil :

Cette interface est la première interface de n'importe quel processus, elle permet d'enregistrer une nouvelle demande toute en sauvegardant les informations générales de la demande à savoir la nature et le type de la demande, l'intervenant principale et les pièces nécessaires.

Guichet unique a.mkharbach@upsysmaroc.com Logout ACCUEIL

Accueil

Type de projet

Grand projet Petit projet

Information du dossier +

Information intervenant +

Validation du dossier -

Choisir l'état de sortie

Rejeté
 Réceptionné

Valider

Figure 30 : Interface du niveau d'accueil.

4. Interface du niveau d'exploitation :

Cette interface permet à l'utilisateur de rechercher un dossier qui se trouve dans le niveau de l'exploitation pour compléter les informations de la demande à savoir les autres intervenants, les pièces et les informations techniques.

Guichet unique a.mkharbach@upsysmaroc.com Logout EXPLOITATION

Exploitation

201700023

Informations sur le dossier

Numéro de dossier	201700023
Type de demande	Construction
Nature de demande	Authorisation
Type de projet	Grand projet

Suivi dossier

Niveau	Etat	Date	Heure	User
Accueil	Réceptionné	11/09/2017	11:15	MKHARBACH Ayoub

Information intervenant +

Les pièces

Figure 31 : Interface du niveau d'exploitation.

5. Interface du niveau de planification :

Cette interface permet à l'utilisateur de planifier automatiquement selon un algorithme une commission pour des dossiers, cette interface entre dans les modules qui sont dédiés à la gestion des autorisations au sein des communes.

The screenshot shows a web-based application interface titled 'Guichet unique'. The top right corner displays the user's email (a.mkharbach@upsysmaroc.com) and a 'Logout' link. A 'PLANIFICATION' button is also present. The main content area is titled 'Planification' and contains a sub-section titled 'Planification des dossiers'. Below this are two green buttons: 'Planifier PP' and 'Planifier GP'. At the bottom left of the main content area, there is a note: 'GP: Grand Projet' and 'PP: Petit Projet'. On the left side, there is a vertical sidebar with various navigation links: 'Consultation', 'Gestion des utilisateurs', 'Gestion des entités', 'Gestion des processus', 'Gestion des types', 'Gestion des référentiels', 'Gestion des commissions', and 'Gestion des réunions'. The logo of the commune de Tiznit is displayed at the top of the sidebar.

Figure 32 : Interface du niveau de planification.

6. Interface des autres niveaux :

Pour les autres niveaux des processus, une interface unique existe qui permet à l'utilisateur de chercher un dossier, si le dossier existe dans le niveau choisi leurs informations seront affiché avec la possibilité de passer à un autre niveau selon l'état choisie.

The screenshot shows a web-based application interface titled 'Guichet unique'. The top right corner displays the user's email (a.mkharbach@upsysmaroc.com) and a 'Logout' link. A 'RECYCLAGE' button is also present. The main content area is titled 'RECYCLAGE' and features a search bar containing the number '201700023' and a search icon. Below the search bar is a table titled 'Informations sur le dossier' with the following data:

Numéro de dossier	201700023
Type de demande	Construction
Nature de demande	Autorisation
Type de projet	Grand projet

Below this table is another table titled 'Suivi dossier' with the following data:

Niveau	Etat	Date	Heure	User
Accueil	Receptionné	11/09/2017	11:15	MIKHARBACH Ayoub
Exploitation	Rejeté	12/09/2017	1:48	MIKHARBACH Ayoub

At the bottom of the main content area is a blue bar titled 'Validation du dossier' with a button labeled 'Choisir le document de sortie...'. On the left side, there is a vertical sidebar with the same navigation links as Figure 32: 'Consultation', 'Gestion des utilisateurs', 'Gestion des entités', 'Gestion des processus', 'Gestion des types', 'Gestion des référentiels', 'Gestion des commissions', and 'Gestion des réunions'. The logo of the commune de Tiznit is displayed at the top of the sidebar.

Figure 33 : Interface des autres niveaux.

7. Interface de gestion des comptes :

Cette interface permet à l'administrateur technique de gérer les comptes.

Cin	Nom & Prénom	Username	Fonction	Téléphone	Date création	Actif	Affection	Roles	Opération
	ABDELKARIM DERRAGH	aderragh@yahoo.fr			14-07-2017	false			
J72884	AKHUCHAL LAHCEN	akhuchal.lahcen@gmail.com	Agent d'exploitation	+212661916415	14-07-2017	false			
JE 74008	BAKKA AICHA	bakka.aicha@gmail.com	Agent d'exploitation	+212661572072	14-07-2017	false			
A187088	BICHOUARINE ABDELHAMID	bichouarine.abdelhamid@gmail.com	Agent d'exploitation	+212661243781	14-07-2017	false			
JE 33136	CHAOURRI OMAR	chaourri.omar@gmail.com	Chef de service	+212662415509	14-07-2017	false			
	Chbourk Chbourk	o.chbourk@yourdev.com			14-07-2017	false			
JE140888	CHRAF MOHAMED	chref.mohamed@gmail.com	Agent d'exploitation	+212662196135	14-07-2017	false			
JE 16527	EL ADOUZI EL HASSAN	eladouzi.elhassan@gmail.com	Agent d'exploitation	+212661512443	14-07-2017	false			

Figure 34 : Interface de gestion des comptes.

8. Interface de gestion des entités :

Cette interface permet à la ressource humaine de gérer les entités.

Numéro	Nom	Type de l'entité	Date de création	Entité supérieure	Opération
1	Commune urbaine de tiznit		14-07-2017		
2	Division de l'urbanisme et de la construction		14-07-2017	Commune urbaine de tiznit	
3	Service de l'urbanisme		14-07-2017	Division de l'urbanisme et de la construction	
4	Service de la construction		14-07-2017	Division de l'urbanisme et de la construction	
5	yourdev	Division	26-07-2017		

Figure 35 : Interface de gestion des entités.

9. Interface de gestion des processus :

Cette interface permet à l'organisateur de gérer les processus.

Numéro	Nom	Date de création	Opération
1	Processus des autorisations	14-07-2017	
2	Test 1	16-07-2017	
3	Test amine	24-08-2017	

Figure 38 : Interface de gestion des processus.

10. Interface de gestion des types :

Cette interface permet au gestionnaire fonctionnel de gérer les types de demandes, ses pièces, ses types d'intervenants, ses champs et ses motifs de rejet.

Numéro	Nature	Nom	Type de projet	Processus	Afficher				
					Pièces	Champs	Intervenants	Motifs	Opération
1	Autorisation	Construction	Grand projet	Processus des autorisations					
2	Autorisation	Aménagement	Grand projet	Processus des autorisations					
3	Autorisation	Morcellement	Grand projet	Processus des autorisations					
4	Autorisation	Modification	Grand projet	Processus des autorisations					
5	Autorisation	Branchemet	Grand projet	Processus des autorisations					
6	Autorisation	Exploitation	Grand projet	Processus des autorisations					
7	Attestation	Permis d'habiter	Grand projet	Processus des autorisations					
8	Attestation	Certificat de conformité	Grand projet	Processus des autorisations					

Figure 41 : Interface de gestion des types.

11. Interface de gestion des référentiels :

Cette interface permet au gestionnaire fonctionnel de gérer les référentiels des adresses.

Numéro	Nom	Nom du quartier	Nom de l'arrondissement	Nom de la commune	Opération
1	Lotissement 1	Quartier 4	Arrondissement 2	Commune de tiznit	
2	Lotissement 2	Quartier 2	Arrondissement 1	Commune de tiznit	
3	Lotissement 3	Quartier 3	Arrondissement 1	Commune de tiznit	
4	Lotissement 4	Quartier 4	Arrondissement 2	Commune de tiznit	
5	Lotissement 5	Quartier 5	Arrondissement 2	Commune de tiznit	
6	Lotissement 6	Quartier 6	Arrondissement 2	Commune de tiznit	
7	Lotissement 7	Quartier 7	Arrondissement 3	Commune de tiznit	
8	Lotissement 8	Quartier 8	Arrondissement 3	Commune de tiznit	

Figure 44 : Interface de gestion des référentiels.

12. Interface de gestion des processus :

Cette interface permet au gestionnaire fonctionnel de gérer les commissions.

Numéro	Nom	Type	Membres	Opération
1	commission mardi	Commission de petit projet		
2	commission grand projet	Commission de grand projet		

Figure 47 : Interface de gestion des commissions.

13. Interface de gestion des réunions :

Cette interface permet au gestionnaire fonctionnel de gérer les réunions.

Numéro	Commission	Date	Nombre de dossiers	Opération
1	commission mardi	25-07-2017	20	
2	commission mardi	26-07-2017	50	
3	commission grand projet	02-08-2017	10	
4	commission grand projet	27-07-2017	5	

Figure 50 : Interface de gestion des réunions.

Conclusion générale.

Mon projet intitulé « Etude, conception et réalisation d'une application web de gestion des processus métiers des demandes au sein des organismes », est un carrefour, un point de croisement de plusieurs disciplines, à savoir, les domaines : développement informatique, gestion des projets, et la qualité. L'enjeu durant ma période de stage a été de mettre en œuvre une solution utilisable sur n'importe quel processus métier, ainsi que d'être flexible pour intégrer des modules spéciaux à un tel processus.

Le présent manuscrit détaille toutes les étapes par lesquelles nous avons passé pour arriver au résultat attendu. Nous avons essayé tout au long de ce travail de construire une application concrète en utilisant la démarche UP-XP, qui se décompose de quatre phases, chaque une entre elles a pour un objectif bien défini.

Au cours de la réalisation de ce projet, nous avons été astreintes par quelques limites notamment, la contrainte d'avoir une application 100% dynamique qui peut être fonctionnée sur n'importe quel processus métier qui était relativement un obstacle devant ce retard pour finir le projet. Cependant, il était une occasion pour mettre en évidence et déployer sur le plan pratique mes connaissances que ce soit en informatique ou dans la gestion des projets.

Ce projet a été une occasion en or pour mettre en pratique mon savoir théorique acquit durant les deux années de formation en master. Il a été un travail sérieux, acharné, et une expérience qui m'a procuré un savoir-faire que ce soit sur le plan technique, ou managérial. Avant de commencer cette aventure, j'ai fixé un objectif personnel, c'est de sculpter non seulement le profil d'un ingénieur, mais un ingénieur compétent, ambitieux est dévoué à son travail. Le présent travail en est, et en sera à jamais le témoin de cette réalisation.

Le projet est en cours de développement, il voit toujours des modifications dans la conception et plus précisément le diagramme de classe pour mieux l'adopter aux besoins de l'utilisateur et pour qu'il soit toujours fiable et au niveau des progrès atteint par la société.

Bibliographie et Webographie :

Ouvrages :

- Julien Dubois, Jean-Philippe Retaillé, Thierry Templier « Spring par la pratique », Eyrolles, 2007.
- Taoufiq GADI, Omar EL BEGGAR, Brahim BOUSETTA « Guide de la Modélisation en UML », 2013.
- Véronique Messager Rota « Vers la gestion des projets », Eyrolles, 2008.
- Brad Green, Shyam Seshadri « AngularJS », O'Reilly, 2013.

Articles :

- <https://www.ibm.com/developerworks/library/ws-restful/index.html>.
- <https://spring.io/blog/2015/07/14/microservices-with-spring>.

Sites :

- Site officiel de Gantt Project : <http://ganttproject.biz>
- Site officiel de Spring : <https://spring.io/>
- Site officiel d'IntelliJ : <https://www.jetbrains.com/idea/>
- Site officiel de ProjeQtor : <https://www.projeqtor.org/fr/>
- Site officiel de Postgresql : <https://www.postgresql.org/>
- Site officiel de Github : <https://github.com/>
- Site officiel d'Apache tomcat : <https://tomcat.apache.org/>
- Site officiel de Maven : <https://maven.apache.org/>
- Site officiel de Windows Server : <https://www.microsoft.com/fr-fr/cloud-platform/windows-server>
- Site officiel de Bootstrap : getbootstrap.com/
- Site officiel de AngularJS : <https://angularjs.org/>
- Site officiel de Hibernate : hibernate.org/

Annexes

Annexe 1 : Convention de nommage.

Entités	Règles	Exemple
Les packages	<p>Généralement les packages sont nommés avec un nom de domaine en commençant par la fin et en ajoutant ensuite le nom du projet.</p> <p>D'autre part Les packages permettent de grouper les classes sous une forme hiérarchisée. Le choix des critères de regroupement est laissé aux développeurs.</p> <p>Il est préférable de regrouper les classes par packages selon des critères fonctionnels.</p>	com.upsysmaroc.collab
Les classes, les interfaces et les constructeurs	<p>La première lettre est en majuscule.</p> <p>Si le nom est composé de plusieurs mots, la première lettre de chaque mot doit être en majuscule(camelCase), ne pas mettre de caractère underscore '_'</p> <p>Le nom d'une classe peut finir par Impl pour la distinguer d'une interface qu'elle implémente.</p> <p>Les classes qui définissent des exceptions doivent finir par Exception.</p>	MaClasse MonInterface MaClasse()
Les méthodes	<p>Leur nom devrait contenir un verbe.</p> <p>La première lettre est obligatoirement une minuscule.</p> <p>Si le nom est composé de plusieurs mots, la première lettre de chaque mot doit être en majuscule sans mettre de caractère underscore '_'.</p> <p>Les méthodes pour obtenir la valeur d'un champ doivent commencer par get suivi du nom du champ.</p> <p>Les méthodes comportent généralement un verbe d'action.</p> <p>Les méthodes pour mettre à jour la valeur d'un champ doivent commencer par set suivi du nom du champ.</p> <p>Les méthodes pour créer des objets (factory) devraient commencer par new ou create.</p> <p>Les méthodes de conversion devraient commencer par to suivi par le nom de la classe renvoyée à la suite de la conversion. Il existe également d'autres mots-clés fréquemment utilisés comme add et remove (ajout, supprimer) pour ajouter et supprimer quelque chose.</p>	public float calculerMontant() { addDocument(), removeUser(),
Les variables	<p>La première lettre est obligatoirement une minuscule et ne devrait pas être un caractère dollar '\$' ou underscore '_' même si ceux-ci sont autorisés.</p> <p>Pour les variables d'instances non publiques, certains recommandent de commencer par un underscore pour éviter la confusion avec le nom d'une variable fournie en paramètre d'une méthode telle que le setter.</p> <p>Si le nom est composé de plusieurs mots, la première lettre de chaque mot doit être en majuscule, ne pas mettre de caractère underscore '_'.</p> <p>Les noms de variables composés d'un seul caractère doivent être évités sauf pour des variables provisoires (index d'une boucle).</p> <p>Les noms communs pour ces variables provisoires sont i, j, k, m et n pour les entiers et c, d et e pour les caractères.</p>	String nomPersonne; Date dateDeNaissance; int i;
Les constantes	Toujours en majuscules, chaque mot est séparé par un underscore '_'. Ces variables doivent obligatoirement être initialisées lors de leur déclaration.	static final int VAL_MIN = 0; static final int VAL_MAX = 9;

L'utilisation des commentaires de documentation :

- La documentation est définie entre les caractères `/**` et `*/`.
- La première ligne de commentaires ne doit contenir que `/**`
- Les lignes de commentaires suivantes doivent obligatoirement commencer par un espace et une étoile. Toutes les premières étoiles doivent être alignées.
- La dernière ligne de commentaires ne doit contenir que `*/` précédé d'un espace.
- Un tel commentaire doit être défini pour chaque entité : une classe, une interface et chaque membre (variables et méthodes).
- Javadoc définit un certain nombre de tags qu'il est possible d'utiliser pour apporter des précisions sur plusieurs informations.
- Ces tags permettent de définir des caractéristiques normalisées. Il est possible d'inclure dans les commentaires des tags HTML de mise en forme (PRE, TT, EM ...) mais il n'est pas recommandé d'utiliser des tags HTML de structure tels que Hn, HR, TABLE ... qui sont utilisés par javadoc pour formater la documentation.
- Il faut obligatoirement faire précéder l'entité documentée par son commentaire car l'outil associe la documentation à la déclaration de l'entité qui le suit.

La déclaration des variables :

Il ne doit y avoir qu'une seule déclaration d'entité par ligne.

```
String nom;  
String prenom;
```

La déclaration des classes et des méthodes :

- Il ne doit pas y avoir d'espaces entre le nom d'une méthode et sa parenthèse ouvrante.
- L'accolade ouvrante qui définit le début du bloc de code doit être à la fin de la ligne de déclaration.
- L'accolade fermante doit être sur une ligne séparée dont le niveau d'indentation correspond à celui de la déclaration.
- Une exception tolérée concerne un bloc de code vide : dans ce cas les deux accolades peuvent être sur la même ligne.
- La déclaration d'une méthode est précédée d'une ligne blanche.

```

class MaClasse extends MaClasseMere {
    String nom;
    String prenom;

    MaClasse(String nom, String prenom) {
        this.nom = nom;
        this.prenom = prenom;
    }

    void neRienFaire() {}
}

```

Il faut éviter d'écrire des méthodes longues et compliquées : le traitement réalisé par une méthode doit être simple et fonctionnel. Cela permet d'écrire des méthodes réutilisables dans la classe et facilite la maintenance. Cela permet aussi d'éviter la redondance de code.

La déclaration des constructeurs :

- Elle suit les mêmes règles que celles utilisées pour les méthodes.
- Il est préférable de définir explicitement le constructeur par défaut (le constructeur sans paramètre). Soit le constructeur par défaut est fourni par le compilateur et dans ce cas il serait préférable de le définir soit il existe d'autres constructeurs et dans ce cas le compilateur ne définit pas de constructeur par défaut.
- Il est préférable de toujours initialiser les variables d'instance dans un constructeur soit avec les valeurs fournies en paramètres du constructeur soit avec des valeurs par défaut.
- Il est recommandé de toujours appeler explicitement le constructeur hérité lors de la redéfinition d'un constructeur dans une classe fille grâce à l'utilisation du mot clé super.

Annexe 2 : Choix des technologies.

Elément	Technologie	Description	Versio n	Alternative	Motifs de choix
Langage de programmation	JAVA	Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications.	8	PHP, Python, .NET	- Java est portable. - Java est orienté objet. - Java assure la gestion de la mémoire.
IDE	IntelliJ	Contrairement à Eclipse qui est géré par un consortium (Fondation Eclipse), IntelliJ IDEA est édité par une société commerciale (JetBrains) qui en assure le développement et le support depuis janvier 2001.	2017.1	Netbeans, STS (Spring Tools Suite)	- Contient tous les plugins nécessaires pour développer des application Spring. - Documentation.
Architecture	J2EE	Java EE (Java Enterprise Edition) est une plate-forme fondée sur le langage Java et son environnement d'exécution, qui regroupe un ensemble de technologies destinées à développer des applications d'entreprise.	7		
Framework J2EE	Spring	Spring est un socle pour le développement d'applications, principalement d'entreprises mais pas obligatoirement. Il fournit de nombreuses fonctionnalités parfois redondantes ou qui peuvent être configurées ou utilisées de plusieurs manières : ceci laisse le choix au développeur d'utiliser la solution qui lui convient le mieux et/ou qui répond aux besoins	4	EJB	- Il est très largement utilisé dans le monde Java, - Propose une très bonne intégration avec des Framework open source ou des standards de Java -Documentation complète.
Framework Spring	Spring boot	Spring Boot est un nouveau Framework conçu pour simplifier le démarrage et le développement de nouvelles applications Spring.	4		
Manipulation des PDFs	iText	iText est une API open source en java, grâce à iText vous aurez la possibilité de créer des PDFs avec votre code java.	7	PDFBox, jPDFWriter	-Facile à utiliser

La Sécurité de l'application	Spring Security	Spring Security est un Framework qui offre un système complet de la gestion de sécurité des applications. Il offre les mécanismes d'authentification et d'autorisation	4.2	Apache Shir	- Spring Security est portable, il ne dépendant pas d'un serveur d'applications - le support de solutions de solutions de Single Sign-On
Framework Mapping Objet Relationnel	Hibernate	Hibernate est le framework open source de mapping O/R le plus populaire. Cette popularité est liée à la richesse des fonctionnalités proposées et à ses performances.	5	IBatis, TopLink	- facile à mettre en oeuvre, efficace et fiable. -open source.
Moteur de production	Maven	Maven est un outil de construction de projets (build), Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java (compilation, tests unitaires et déploiement des applications qui composent le Projet)	3.5	Apache ant, Gradle,	- Maven est plus flexible. - le concept POM simple
Gestionnaire de code source	GitHub	GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git		SVN ;	-La gestion des branches. -La rapidité.

Annexe 3 : Diagramme de classe.

