

**UNIVERSITÉ JOSEPH KI-ZERBO**

**(UJKZ)**

\*\*\*\*\*

**INSTITUT BURKINABÈ DES ARTS ET MÉTIERS**

**(IBAM)**



**RAPPORT DE STAGE POUR L'OBTENTION DE LA LICENCE  
PROFESSIONNELLE**

**OPTION : Méthodes Informatiques Appliquées à la Gestion (MIAGE)**

Période de stage : 03 juin 2023 au 17 Septembre 2024

**THEME :**

**CONCEPTION ET REALISATION D'UNE PLATEFORME  
DE SUIVI DU PLAN DE PASSATION DE LA SONABEL.**

Présenté par : DANAYE David Eliezer

**Maitre de stage :**

M. NIKIEMA Laurent

Développeur web/data et Gérant

de Logo Services.

## SOMMAIRE

<b>SOMMAIRE.....</b>	<b>i</b>
<b>DEDICACES .....</b>	<b>ii</b>
<b>REMERCIEMENTS .....</b>	<b>iii</b>
<b>LISTE DES SIGLES ET ABREVIATIONS .....</b>	<b>iv</b>
<b>LISTE DES FIGURES ET GRAPHIQUES.....</b>	<b>vi</b>
<b>LISTE DES TABLEAUX .....</b>	<b>vii</b>
<b>RESUME .....</b>	<b>viii</b>
<b>INTRODUCTION GENERALE.....</b>	<b>1</b>
<b>CHAPITRE 1: PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL ....</b>	<b>2</b>
<b>I. Présentation de la structure de formation (IBAM).....</b>	<b>2</b>
<b>II. Présentation de la structure d'accueil (LOGO SERVICES).....</b>	<b>5</b>
<b>CHAPITRE 2: ANALYSE ET CONCEPTION .....</b>	<b>7</b>
<b>I. Etude préalable .....</b>	<b>7</b>
<b>II. Expressions des besoins.....</b>	<b>15</b>
<b>III. Conception globale .....</b>	<b>30</b>
<b>IV. Réalisation.....</b>	<b>36</b>
<b>CHAPITRE 3: BILAN DU STAGE .....</b>	<b>58</b>
<b>I. Présentation du déroulement du stage et des activités réalisées .....</b>	<b>58</b>
<b>II. Observations et suggestions .....</b>	<b>59</b>
<b>CONCLUSION GÉNÉRALE .....</b>	<b>60</b>
<b>BIBLIOGRAPHIE ET WEBOGRAPHIE .....</b>	<b>61</b>
<b>TABLE DES MATIERES .....</b>	<b>63</b>
<b>I. ANNEXES 1 : Le langage UML.....</b>	<b>I</b>
<b>II. ANNEXE 2 : Méthode COCOMO .....</b>	<b>II</b>
<b>III. ANNEXE 3 : Dictionnaire de données.....</b>	<b>III</b>

## DEDICACES

*À*

*Toute ma famille et à  
monsieur le Maître de  
Conférence Mahamoudou  
ZONGO.*

## REMERCIEMENTS

Nous tenons à exprimer notre gratitude envers toutes les personnes qui ont contribué à notre parcours scolaire et universitaire. Nous remercions particulièrement :

- **DIEU**, qui est la base de tout dans notre vie.
- **M. TOPAN**, propriétaire de LOGO SERVICES, pour son accueil chaleureux et son soutien précieux durant notre stage au sein de son entreprise.
- **M. Laurent NIKIEMA**, notre maître de stage, pour son aide continue et son attention généreuse, malgré les contraintes de son emploi du temps.
- **Dr Yacouba OUATTARA**, coordinateur de la filière MIAGE et notre superviseur, pour avoir apporté son expertise et son soutien constant, contribuant ainsi à notre développement professionnel.
- **Pr Gilbert BAYILI**, Directeur de l'IBAM, et l'ensemble des enseignants, pour les conseils avisés et l'enseignement de qualité que nous avons reçus pendant ces trois années à l'IBAM.
- Enfin, **nos amis et proches**, pour leur soutien indéfectible et leurs encouragements précieux tout au long de notre parcours.

## LISTE DES SIGLES ET ABREVIATIONS

**Tableau 1: liste des sigles et abréviations**

SIGLES ET ABREVIATIONS	SIGNIFICATION
2TUP	Two-Track Unified Process
ABF	Assurance-Banque-Finance
ADB	Assistant de Direction option Bilingue
ADC	Assistant de Direction option Comptable
ATOS	Administratifs, Techniques, Ouvriers, de Service
CCA	Comptabilité-Contrôle-Audit
COCOMO	COConstructive COst MOdel
CSAF	Chef Service Administratif et Financier
DGCMEF	Direction Générale du contrôle des Marchés Publics et des Engagements Financiers
IBAM	Institut Burkinabè des Arts et Métiers
IBF	Master en Ingénierie Bancaire et Financière
IFU	Identification Fiscale Unique
M2ISIE	Master en Informatique option Ingénierie des systèmes d'information en entreprise

MAGE	Master en Administration et Gestion des Entreprise
MCCA	Master en Comptabilité-Contrôle-Audit
MIAGE	Méthodes Informatiques Appliquées à la Gestion
MID	Marketing et innovation Digitale
MVT	Modèle Vue Contrôleur
PPM	Plan de passation de marchés
RCCM	Registre du Commerce et du Crédit Mobilier
RI-MT	Informatique option Réseau Informatique Multi media et Telecom
TIC	Technologies de l'Information et de la Communication
UJKZ	Université Joseph KI-ZERBO
UML	Unified Modeling Language
SARL	Société à Responsabilité Limitée
SONABEL	Société Nationale d'Électricité du Burkina Faso
SGBD	Système de Gestion des Bases de Données

## LISTE DES FIGURES ET GRAPHIQUES

Figure 1: Organigramme de l'IBAM .....	3
Figure 2: Schéma représentant le processus 2TUP [1] .....	13
Figure 3: La hiérarchie schématique pour UML [2] .....	14
Figure 4: Planning de réalisation du projet .....	15
Figure 5: Diagramme de cas d'utilisations .....	19
Figure 6: Diagramme de séquence vue de l'extérieur pour CU « s'authentifier ».....	24
Figure 7: Diagramme de séquence vue de l'extérieur pour CU « Importer un plan » .....	25
Figure 8: Diagramme de séquence vue de l'extérieur pour CU « Enregistrer un dossier ».....	26
Figure 9: Diagramme d'activité pour le CU « S'authentifier ».....	27
Figure 10: Diagramme d'activité pour le CU « Enregistrer un dossier » .....	28
Figure 11: Architecture 1 tier [3] .....	30
Figure 12: Schéma représentant l'architecture 2 tiers [4] .....	30
Figure 13: Schéma représentant l'architecture 3 tiers [5] .....	30
Figure 14: Diagramme de classes.....	36
Figure 15: Diagramme de déploiement.....	36
Figure 16: Structure de l'architecture Modèle-Vue-Template [6] .....	43
Figure 17: Résultat du test des modèles de notre application .....	46
Figure 18: Résultat du test de la vue .....	46
Figure 19: Ecran de connexion.....	55
Figure 20: Ecran d'import et de filtration .....	55
Figure 21: Ecran d'affichage des dossiers pour leurs suivis.....	56
Figure 22: Ecran d'évolution des dossiers .....	56
Figure 23: Ecran de réception des offres.....	57
Figure 24: Ecran d'ajout d'un avis de publication .....	57

## LISTE DES TABLEAUX

Tableau 1: liste des sigles et abréviations .....	iv
Tableau 2 : Comparaison des méthodologies ou processus de développement.....	9
Tableau 3: Liste des cas d'utilisations .....	16
Tableau 4:Description du cas d'utilisation « S'authentifier » .....	20
Tableau 5: Description du cas d'utilisation « importer un plan ».....	21
Tableau 6 : Description du cas d'utilisation « Enregistrer un dossier ».....	22
Tableau 7: Description du cas d'utilisation « Ajouter un lot » .....	23
Tableau 8 : Tableau comparatif des architectures de développement.....	29
Tableau 9: Description de la classe « Fournisseur » .....	31
Tableau 10: Description de la classe « Dossier » .....	32
Tableau 11: Description de la classe « Offre » .....	32
Tableau 12: Description de la classe « Plan » .....	33
Tableau 13: Description de la classe « Résultat » .....	33
Tableau 14: Etude comparative des SGBD.....	39
Tableau 15: Etude comparative des serveurs d'application.....	41
Tableau 16: Cout total du projet.....	53
Tableau 17: Formule de calcul COCOMO .....	II
Tableau 18: description de la classe « Marché ».....	VI
Tableau 19: description de la classe « Avis » .....	VI
Tableau 20: description de la classe « Lot » .....	VI
Tableau 21: description de la classe « Statut ».....	VII
Tableau 22: description de la classe « Utilisateur » .....	VII



## RESUME

Les marchés publics, qui regroupent les contrats entre les institutions publiques et les entreprises privées pour la fourniture de services, de matériel ou la réalisation de travaux, posent un enjeu majeur en matière de suivi. En effet, le grand nombre de dossiers à gérer et la dispersion des informations rendent le suivi des marchés difficile. C'est dans cette optique que notre projet de fin d'étude s'est orienté vers la conception d'une solution numérique visant à centraliser les données, suivre les projets en temps réel, et améliorer la visibilité sur les processus de passation des marchés publics.

Pour répondre à ces enjeux, nous avons suivi une méthodologie en plusieurs étapes. Tout d'abord, nous avons procédé à la spécification des besoins en menant une analyse approfondie des attentes des utilisateurs et des contraintes fonctionnelles. Ensuite, à l'aide d'outils de modélisation comme UML, nous avons conçu l'architecture du système en nous appuyant sur le modèle MVT (Modèle-Vue-Template). Enfin, nous avons développé et déployé la plateforme en intégrant PostgreSQL pour la gestion des données et Apache pour assurer une gestion efficace des requêtes.

Notre solution, désormais déployée, intègre des fonctionnalités clés telles que l'authentification sécurisée, la gestion des dossiers, des offres et des marchés, ainsi que le suivi détaillé de l'évolution du plan de passation. Grâce à ces outils, les institutions publiques peuvent améliorer la gestion des marchés, garantir une meilleure transparence et accroître l'efficacité des processus en automatisant certaines tâches autrefois gérées via des fichiers Excel.

## SUMMURY

Public procurement, which covers contracts between public institutions and private companies for the supply of services, equipment or works, poses a major challenge in terms of monitoring. The large number of files to be managed and the dispersal of information make it difficult to monitor contracts. With this in mind, our end-of-study project focused on designing a digital solution to centralise data, monitor projects in real time and improve visibility of the public procurement process.

To meet these challenges, we followed a multi-stage methodology. Firstly, we specified the requirements by carrying out an in-depth analysis of user expectations and functional constraints. Then, using modelling tools such as UML, we designed the system architecture based on the MVT (Model-View-Template) model. Finally, we developed and deployed the platform, integrating PostgreSQL for data management and Apache for efficient query management.

Our solution, which has now been deployed, incorporates key functionalities such as secure authentication, management of files, tenders and contracts, and detailed monitoring of the progress of the procurement plan. Thanks to these tools, public institutions can improve contract management, guarantee greater transparency and increase the efficiency of their processes by automating certain tasks previously managed via Excel files.

## INTRODUCTION GENERALE

Dans un monde en constante évolution et émergent considérablement sur la technologie et l'efficacité des processus, la modernisation des méthodes de gestion est devenue une priorité incontournable pour les entreprises et les institutions publiques. L'un des domaines où cette modernisation est particulièrement critique est le suivi des marchés publics. Ces marchés, qui comprennent tous les contrats entre une institution publique et des prestataires privés pour acheter des biens, des services ou réaliser des travaux, sont vraiment cruciaux pour que les institutions fonctionnent correctement. Pour assurer un suivi correct et conforme aux recommandations, il est indispensable que les marchés publics soient surveillés avec une rigueur et une transparence irréprochable.

Chaque année, la SONABEL, un pilier du secteur énergétique au Burkina Faso, doit gérer un grand nombre de marchés publics. Ce processus est assez complexe, comprenant des étapes qui vont de la préparation des dossiers techniques à l'attribution des marchés. Pour que ce suivi soit réalisé correctement, des outils informatiques performants sont indispensables pour centraliser les informations, suivre toutes les étapes, et garantir que tout se déroule de manière transparente et efficace.

C'est dans ce contexte que nous avons été accueillis par Logo-Services pour un stage de trois (03) mois, au cours duquel nous avons été chargés de faire la conception et le développement d'une application web dédiée au suivi des marchés publics pour le compte de la SONABEL.

Dans ce rapport, nous explorerons largement notre démarche, notre méthodologie et les résultats obtenus au cours de cette expérience professionnelle enrichissante. Pour mieux comprendre notre travail, ce rapport est structuré en quatre (04) grands chapitres, chacun abordant un aspect spécifique de notre projet. Le premier chapitre sera consacré à la présentation de nos structures de formation (IBAM) et d'accueil (LOGO SERVICES). Le deuxième chapitre se rapporte à l'analyse et à la conception du système. Le troisième chapitre se résume au bilan du stage effectué au sein de LOGO SERVICES.

---

## CHAPITRE 1

---

### PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL

---

Ce chapitre a pour objectif de présenter nos deux structures d'accueil et de formation. En premier temps nous présenterons la structure de formation qui est l'Institut Burkinabè des Arts et Métiers (IBAM<sup>1</sup>). En second lieu nous présenterons la structure d'accueil qui est LOGO SERVICES<sup>2</sup>. L'objectif est de montrer le cadre dans lequel s'est déroulé le stage.

#### **I. Présentation de la structure de formation (IBAM)**

L'Institut Burkinabè des Arts et Métiers (IBAM) est un établissement d'enseignement professionnel de renom au Burkina mais également dans la sous-région. Dans cette partie, nous exposerons une présentation générale de l'IBAM en faisant ressortir son historique, son objectif, son organisation et ses filières de formations.

##### **1. Historique**

Situé à Somgandé, l'Institut Burkinabè des Arts et Métiers (IBAM) est un institut d'enseignement professionnel. Il a été créé en janvier 2000 dans le cadre de la refondation de l'Université de Ouagadougou rebaptisée l'Université Ouaga I Pr Joseph KI-ZERBO, le 26 décembre 2015, puis plus simplement « l'Université Joseph KI-ZERBO » depuis le 12 avril 2019. L'IBAM est la matérialisation de l'engagement de l'Université Joseph KIZERBO (UJKZ) dans la professionnalisation des filières.

##### **2. Objectif**

L'objectif principal de l'IBAM est de répondre aux besoins du marché de l'emploi par la mise à disposition d'un potentiel humain de cadres moyens et supérieurs dans divers secteurs d'activités.

##### **3. Organisation**

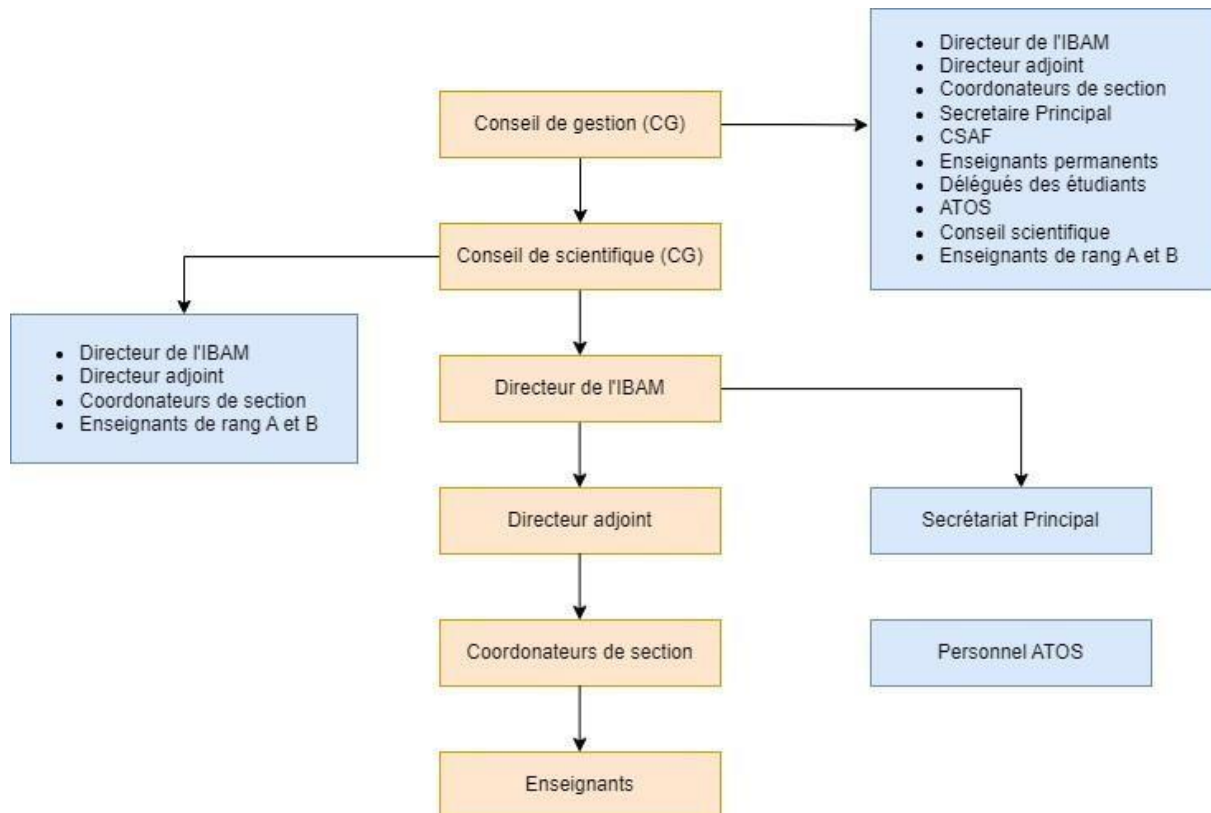
L'IBAM est organisé en deux organes à savoir l'organe statutaire et l'organe d'exécution. L'organisation des enseignants et l'organisation du contrôle des aptitudes et des connaissances sont fixées par arrêté ministériel. L'institut est dirigé par un Directeur assisté d'un adjoint, tous deux élus par le collège électoral de l'établissement pour un mandat de deux (02) ans

---

<sup>1</sup> IBAM : [https://ujkz.bf/u\\_department/ibam/](https://ujkz.bf/u_department/ibam/)

<sup>2</sup> LOGO SERVICES : <https://logo-services.com/>

renouvelables. La figure 1 présente l'organigramme de l'IBAM, à travers lequel on peut voir les liens hiérarchiques et organisationnels existants entre les personnels de l'IBAM.



**Figure 1: Organigramme de l'IBAM**

#### **a. Organe statutaire**

L'organe statutaire de l'IBAM est composé du Conseil de gestion et du Conseil scientifique. Le Conseil de gestion est l'organe qui gère l'institut. Il est composé du Directeur de l'IBAM, du Directeur adjoint, des Coordonnateurs de sections, du Secrétaire principal, du Chef de Service Administratif et Financier (CSAF), des Enseignants permanents, de deux (02) représentants des Délégués de syndicats des étudiants et d'un représentant du personnel ATOS. Il est présidé par le Directeur de l'IBAM et se réunit tous les trois (03) mois sauf en cas de session extraordinaire sur convocation de son président. Le Conseil de gestion a pour missions de :

- définir le règlement intérieur de l'institut ;
- proposer le budget de fonctionnement de l'institut ;

- décider des mesures administratives et financières conformément aux textes en vigueur ;
- proposer la modification des statuts de l'IBAM.

Le Conseil scientifique est également composé du Directeur de l'IBAM, du Directeur adjoint, des Coordonnateurs de section, des enseignants de rang A et les enseignants de rang B de l'Institut. Il est présidé par le Directeur de l'IBAM. Il se réunit et délibère dans les mêmes conditions que le Conseil de gestion, mais son secrétariat est assuré par un des Coordonnateurs de section. Le Conseil scientifique a pour mission de :

- Proposer une organisation générale des enseignements de l'institut ;
- Proposer l'ouverture, la fusion ou la fermeture de filières ;
- Créer des titres, diplômes et d'examiner les équivalences pédagogiques ;
- Proposer la création de nouveaux postes et ;
- Apprécier les orientations de la recherche au sein de l'institut.

#### **b. Organe d'exécution**

L'organe exécutif a pour mission de mettre en pratique les décisions prises par l'organe statutaire. Il est composé essentiellement de la direction de l'IBAM. La Direction est pilotée par le Directeur de l'IBAM dont les missions sont de diriger, de contrôler le fonctionnement des services et de définir la politique de l'institut. Elle coordonne les activités des différents services et veille au maintien de l'image de marque de l'Institut. En outre, elle veille à la mise en œuvre des décisions, des projets et des programmes de l'Institut. Enfin, elle veille à la mise en application des textes et règlements de l'établissement. Le Directeur de l'IBAM est assisté par une Secrétaire de direction. Rappelons que le Secrétaire principal et le CSAF pour leur part, assistent également le Directeur de l'IBAM dans l'accomplissement de certaines activités en relation avec leurs domaines de spécialisation. La Direction est composée du Secrétariat de direction, du Service administratif et financier et du Secrétariat principal.

#### **4. Filières de formation**

En vue d'atteindre les objectifs cités précédemment, l'IBAM offre des formations dans plusieurs filières. Celles-ci sont réparties en deux (02) groupes selon les diplômes, ces groupes sont : le groupe des Licences professionnelles et le groupe des Masters.

##### **❖ Licences professionnelles**

L'IBAM offre des formations initiales dans la catégorie des licences professionnelles :

- Comptabilité-Contrôle-Audit (CCA) ;
- Assurance-Banque-Finance (ABF) ;
- Assistant de Direction option Bilingue (ADB) ;
- Assistant de Direction option Comptable (ADC) ;
- Informatique option Méthodes Informatique Appliquées à la Gestion (MIAGE) ;
- Informatique option Réseau Informatique Multimédia et Telecom (RI-MT) ;
- Marketing et innovation Digitale (MID).

#### ❖ Masters

En Master, l'IBAM offre cinq (05) filières de formation qui sont :

- Master en Administration et Gestion des Entreprises (MAGE) ;
- Master en Comptabilité-Contrôle-Audit (MCCA) ;
- Master en Ingénierie Bancaire et Financière (IBF) ;
- Master en Informatique option Ingénierie des systèmes d'information en entreprise (M2ISIE) ;
- Master en informatique option Sécurité Informatique.

La filière MIAGE a été créée dans l'optique de répondre aux besoins croissants des entreprises en cadres compétents dans le domaine des Technologies de l'Information et de la Communication (TIC). Cette filière accueille en première année les bacheliers des séries C, D et E ayant passé avec succès le test de sélection ou ayant été acceptés en tant qu'auditeurs libres. Les étudiants en fin de cycle en MIAGE doivent mettre en pratique les connaissances acquises en classe en effectuant un stage d'une durée d'au moins trois (03) mois suivi d'une soutenance publique.

C'est pour répondre à cette exigence académique que nous avons effectué notre stage pratique à LOGOS SERVICES que nous mettons en avant dans les lignes qui suivent.

## **II. Présentation de la structure d'accueil (LOGO SERVICES)**

LOGO SERVICES est la société au sein de laquelle nous avons effectué notre stage. Dans cette partie nous présenterons son histoire, ses services et son organisation de façon générale.

## 1. Histoire

LOGO SERVICES SARL est une société de services informatiques Burkinabè. Créée en Mars 2019, c'est une Société à Responsabilité Limitée (SARL) avec comme identifiant Registre du commerce (RCCM) N°BFOUA 2019B1837 et fiscale (N° IFU) : 00117182 B. LOGO SERVICES est reconnue intégrateur national de solutions informatiques à travers son agrément technique D5 : Intégrateur de solutions informatique. Elle est joignable aux adresses suivantes 01 BP 5988 Ouagadougou 01, Secteur 19 ex 25. Téléphones : Nous avons le fixe : +226 25 39 29 20 et les portables tels que le 76 40 32 14 et le 71 57 57 16. LOGO SERVICES est également joignable sur cette adresse email : [infos@logoservices.com](mailto:infos@logoservices.com).

## 2. Services

Logo Service offre de nombreuses prestations dans les domaines suivants :

- la conception et la mise en place de solutions logicielles ;
- la conception et la mise en place de solutions SIG et d'analyse de données spatiales ;
- la conception et la mise en place de solution d'aide à la décision (analyse et de reporting des données) ;
- l'ingénierie réseau et télécom ;
- la Sécurité informatique ;
- l'organisation, le traitement et la numérisation des documents et Archives ;
- le renforcement des capacités en informatique.

Elle dispose de compétences dans ses domaines d'actions et d'un réseau de partenaires qui lui ont permis de mener à bien des projets informatiques au profit de l'administration Burkinabè et des entreprises publiques et privées confère quelques références.

## 3. Organisation générale de LOGO SERVICES

LOGO SERVICES est une entreprise qui fonctionne grâce à la détermination de son Directeur Général, ses partenaires et ses différents consultants évoluant dans tous les domaines d'activité. Leurs efficacités et professionnalismes permettent de satisfaire les besoins des clients. Elle a à son actif, une quinzaine de projets exécutés à la satisfaction des clients.



---

## CHAPITRE 2

---

---

### ANALYSE ET CONCEPTION

---

Ce chapitre aborde le thème principal intitulé « **Conception et réalisation d'une plateforme de suivi du plan de passation de la SONABEL** ». Pour ce faire, nous débuterons par une présentation du sujet, en détaillant la méthode d'analyse et de conception choisie, les acteurs clés du projet ainsi que le calendrier de notre étude. Ensuite, l'expression des besoins sera traitée, suivie d'une étude technique approfondie et d'une conception détaillée du système à venir. Enfin, nous examinerons les différents aspects nécessaires à la réalisation de cette plateforme.

#### **I. Etude préalable**

Cette partie vise à apporter des précisions approfondies sur le sujet traité, en mettant en avant la méthode d'analyse et de conception que nous avons choisie, la composition de l'équipe de projet ainsi que l'organisation du calendrier pour mener à bien notre travail.

##### **1. Présentation du thème**

Cette présentation s'organise autour de deux aspects majeurs, à savoir la définition de la problématique et les résultats que nous visons à atteindre.

##### **a. Préliminaire**

- **Plateforme** : Système logiciel ou application créée pour gérer le suivi du plan de passation.
- **Suivi** : Action de suivre les étapes impliquées dans le processus du plan de passation.
- **Plan de passation** : Ensemble des procédures et des étapes pour attribuer des marchés publics.
- **SONABEL** : Société Nationale Burkinabè d'Électricité, l'entité pour laquelle la plateforme est développée.
- **Marchés publics** : Contrats passés entre la SONABEL et les entreprises pour la fourniture de biens et services.
- **Automatisation** : Utilisation de la technologie pour rendre le suivi plus efficace et réduire les tâches manuelles.
- **Transparence** : Garantir que les processus sont clairs et accessibles aux parties prenantes.
- **Efficacité** : Optimisation des ressources et des processus pour améliorer les résultats.

### **b. Problématique**

Le suivi des marchés publics au sein d'institutions comme la SONABEL devient de plus en plus complexe avec l'augmentation des projets et la nécessité de structurer les informations de manière plus organisée et accessible. Actuellement, le Plan de Passation des Marchés (PPM), est suivi à l'aide de fichiers Excel. Cette méthode de suivi entraîne souvent des insatisfactions, des incohérences et un manque de visibilité globale sur l'état d'avancement des différents projets. L'absence d'une plateforme centralisée pour regrouper toutes les informations complique la prise de décision et compromet la transparence tout au long du processus.

Une solution pour pallier ces lacunes est la digitalisation du suivi du PPM. Dans ce rapport, nous tentons de répondre à la question : comment concevoir une plateforme web qui centralise les informations du Plan de Passation des Marchés à la SONABEL, tout en assurant un suivi à chaque étape du processus de passation des marchés publics ?

### **c. Objectif**

Notre objectif est de proposer une solution numérique capable de centraliser les données, de suivre les projets en temps réel et de fournir une meilleure visibilité sur l'avancement des processus. Cela permettrait d'améliorer la précision, la transparence et l'efficacité du suivi des marchés publics en automatisant certaines tâches actuellement réalisées à l'aide de fichiers Excel, ce qui rendrait le processus plus fluide et plus fiable.

### **d. Résultats attendus**

Au regard des besoins exprimés par l'entreprise, les résultats attendus à l'issue de ce projet sont de mettre en place :

- ✚ Une application web qui doit permettre :
  - L'importation et la filtration des plans de passations ;
  - La création et l'affectation de dossiers techniques pour les lignes du plan ;
  - La réception des offres techniques assignée à chaque dossier ;
  - Le renseignement des avis de publication concernant chaque dossier ;
  - La publication des résultats liés à chaque offre ;
  - L'attribution des marchés pour chaque lot d'un dossier ;
  - La visualisation de l'état d'avancement de chaque dossier.

- La visualisation de l'état d'avancement globale de tous les dossiers
- La visualisation du plan de passation des marchés.

## 2. Méthodes d'analyse et de conception

Une méthode d'analyse et de conception est un procédé qui a pour but de permettre la formalisation des étapes préliminaires du développement d'un système afin de rendre ce développement plus conforme aux exigences du client.

### a. Processus de développement

Le processus de développement constitue la démarche fondamentale permettant d'obtenir un produit logiciel dans un délai raisonnable tout en minimisant les ressources.

Le tableau 2 ci-dessous est un récapitulatif d'une étude comparative entre les principales méthodologies de développement que nous avons choisi vu la diversité de ces méthodes en s'inspirant des travaux existants (voir les rapports dans la bibliographie).

**Tableau 2 : Comparaison des méthodologies ou processus de développement**

Méthodologies	Description	Atouts	Insuffisances
<b>Cascade<sup>3</sup></b>	<ul style="list-style-type: none"> <li>- Les phases sont déroulées d'une manière séquentielle.</li> </ul>	<ul style="list-style-type: none"> <li>- Distingue clairement les phases du projet.</li> </ul>	<ul style="list-style-type: none"> <li>- Non itératif.</li> <li>- Pas de modèles pour les documents</li> </ul>
<b>RUP<sup>4</sup> (Rational Unified Process)</b>	<ul style="list-style-type: none"> <li>- Le RUP est à la fois une méthodologie et un outil prêt à l'emploi.</li> <li>- Cible des projets de plus de 10 personnes.</li> </ul>	<ul style="list-style-type: none"> <li>- Itératif.</li> <li>- Spécifie le dialogue entre les différents intervenants du projet.</li> </ul>	<ul style="list-style-type: none"> <li>- Assez flou dans sa mise en œuvre.</li> <li>- Ne couvre pas les phases en amont et en aval au développement.</li> </ul>
<b>2TUP<sup>5</sup> (Two Truck Unified Process)</b>	<ul style="list-style-type: none"> <li>- Il s'articule autour de l'architecture.</li> </ul>	<ul style="list-style-type: none"> <li>- Itératif.</li> <li>- Laisse une large place à la</li> </ul>	<ul style="list-style-type: none"> <li>- Plutôt superficiel sur les phases situées en</li> </ul>

<sup>3</sup> CASCADES : [www.ionos.fr](http://www.ionos.fr)

<sup>4</sup> RUT : [prezi.com](http://prezi.com)

<sup>5</sup> TUP : [prezi.com](http://prezi.com)

Méthodologies	Description	Atouts	Insuffisances
	<ul style="list-style-type: none"> <li>- Propose un cycle de développement en Y.</li> <li>- Cible des projets de toutes tailles.</li> </ul>	<ul style="list-style-type: none"> <li>technologie et à la gestion des risques.</li> <li>- Définit les profils des intervenants, les plannings, les prototypes</li> </ul>	<ul style="list-style-type: none"> <li>amont et en aval du développement.</li> <li>- Ne propose pas de documents types.</li> </ul>
<b>XP<sup>6</sup> (eXtreme Programming)</b>	<ul style="list-style-type: none"> <li>- Ensemble des bonnes pratiques de développement.</li> <li>- Cible : Moins de 10 personnes</li> </ul>	<ul style="list-style-type: none"> <li>- Itératif.</li> <li>- Donne une importance aux aspects techniques.</li> <li>- Innovant : programmation en duo.</li> </ul>	<ul style="list-style-type: none"> <li>- Assez flou dans sa mise en œuvre.</li> <li>- Ne couvre pas les phases en amont et en aval au développement.</li> </ul>
<b>Scrum<sup>7</sup></b>	<ul style="list-style-type: none"> <li>- Se base sur des itérations dites sprints de développement</li> </ul>	<ul style="list-style-type: none"> <li>- Donne toute confiance aux développeurs et les laisse effectuer leur travail.</li> <li>.</li> </ul>	<ul style="list-style-type: none"> <li>- La mise en œuvre du développement n'est pas précisée.</li> <li>- Développement rapide et répétitif</li> </ul>

L'étude comparative réalisée sur les processus de développement Cascade, RUP, 2TUP, XP et Scrum résumés dans le tableau 2 nous permet de tirer les conclusions suivantes :

- le processus RUP néglige les contraintes techniques qui sont indispensables dans notre projet, nous avons par conséquent choisi de l'écarter,
- le processus XP néglige la phase de capture de besoins fonctionnels et techniques et la phase de conception et donne une grande importance à la phase de développement, il est par conséquent écarté,
- CASCADE est un processus séquentiel et non itératif,

<sup>6</sup> XP : [prezi.com](http://prezi.com)

<sup>7</sup> SCRUM : [asana.com](http://asana.com)

- Scrum quant à lui subdivise les différentes phases du projet en sprint qui en théorie ne dépasse pas 30 jours,
- le processus 2TUP permet en particulier de séparer les contraintes fonctionnelles des contraintes techniques érigées sous forme de deux branches permettant de les exploiter parallèlement.

Après avoir étudié et comparé les processus de développement et pour pouvoir contrôler au mieux les risques afin de mener à bon terme notre projet, nous avons opté pour le processus 2TUP pour plusieurs raisons :

- d'une part, 2TUP donne une grande importance à la technologie ce qui est important pour notre projet ;
- d'autre part, 2TUP est un processus en Y qui contient une branche technique, une branche fonctionnelle et une branche réalisation. Les deux branches technique et fonctionnelle peuvent être exploitées en parallèle. De ce fait, si la technologie évolue ou, s'il arrive que lors du déroulement du projet, il y a modification d'un besoin technique, la branche technique peut être traitée puis réintégrée dans le projet facilement. De même si une nouvelle fonctionnalité se présente, seule la branche fonctionnelle va être traitée sans toucher à l'autre branche.

Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, à identifier les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte. Le processus comporte 2 branches :

❖ **Une branche fonctionnelle qui se présente comme suit :**

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité ;
- L'analyse qui consiste à étudier les spécifications fonctionnelles dans le but de connaître le métier qui sera effectué par le système ;

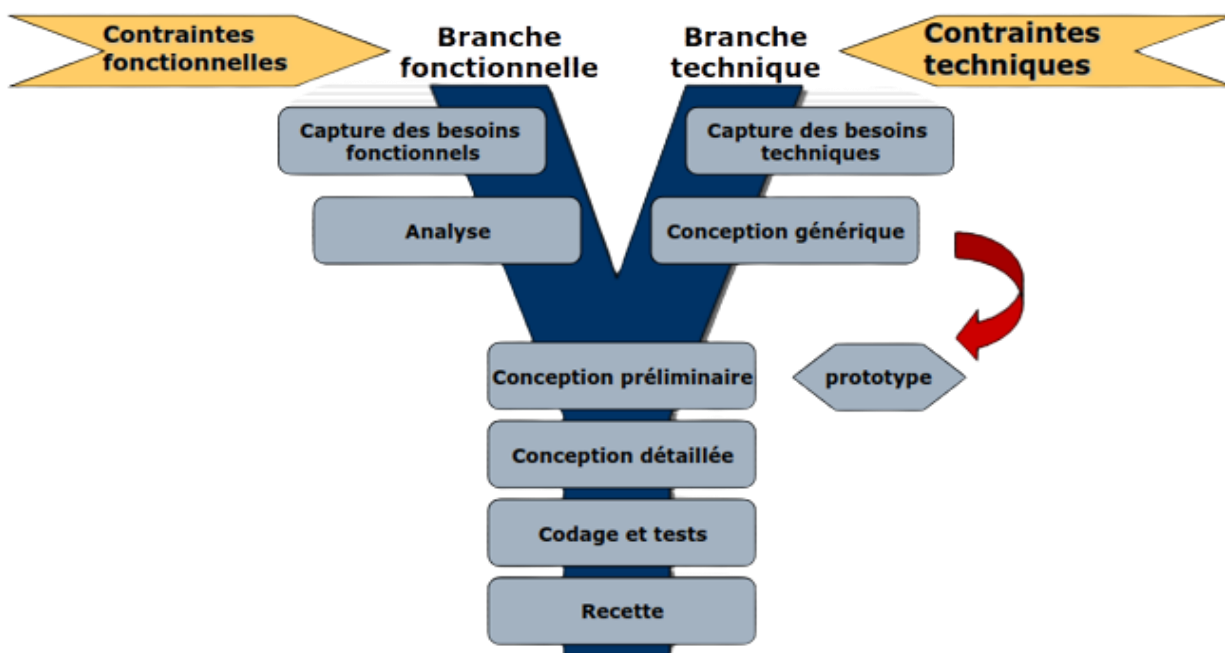
❖ **Une branche technique qui se présente comme suit :**

- La capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des prérequis d'architecture technique ;
- La conception générique qui consiste à indiquer les composants essentiels à l'élaboration de l'architecture technique. Elle assure la réponse aux exigences du système.

Ces deux branches se fusionnent pour donner la branche du milieu. Cette branche est constituée de :

- La conception préliminaire qui est un peu complexe. Elle doit intégrer le modèle d'analyse fonctionnelle dans l'architecture technique afin d'indiquer les composants du système à développer. Elle structure le système en composants, délivrant les services techniques et fonctionnels ;
- La conception détaillée qui permet d'exécuter chaque composant du système ;
- Le codage qui consiste à implémenter les fonctionnalités du système avec des tests effectués au fur et à mesure ;
- La recette qui consiste en la validation des fonctionnalités du système développé.

La figure suivante présente les grands axes de la méthode 2TUP :



**Figure 2: Schéma représentant le processus 2TUP [1]**

Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent en plus de leur facilité et de leur clarté, de bien modéliser le système à chaque étape.

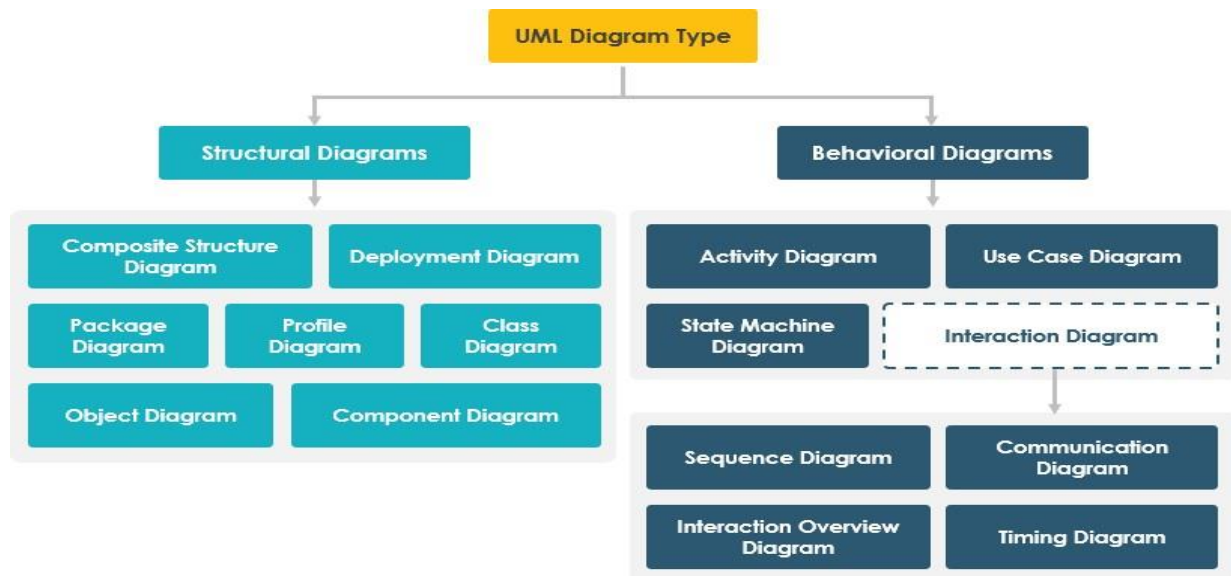
**b. Langage de modélisation**

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l'information ou de la connaissance des systèmes dans une structure qui est définie par un ensemble cohérent de règles. Pour développer une application, il faut d'abord organiser les idées, les documenter avant de commencer la réalisation tout en définissant les modules et les étapes. On appelle cette démarche « modélisation ».

Pour réaliser cette modélisation, nous avons utilisé le langage UML (ANNEXE 1) pour plusieurs raisons. Nous pouvons citer entre autres qu'il :

- présente l'avantage d'être le standard en matière de modélisation objet universellement reconnue ;
- est un langage visuel car sa notation graphique permet d'exprimer visuellement des solutions objets facilitant ainsi la comparaison et l'évaluation de celles-ci ;
- est un langage formel et normalisé doté d'un gain de précision et d'un gage de stabilité ;
- sert à formaliser tous les documents techniques d'un projet et permet d'affiner les détails de l'analyse au fur et à mesure de l'avancée du projet ;
- est capable d'utiliser le même atelier de génie logiciel, depuis l'expression des besoins des utilisateurs jusqu'à la génération de tout ou partie du code ;
- est un support de communication performant car il cadre l'analyse tout en facilitant la compréhension des représentations abstraites complexes.

Figure 3 illustre comment UML propose un ensemble standardisé de notations, de diagrammes et de symboles permettant de décrire les divers aspects structurels, comportementaux, interactifs et dynamiques des systèmes.



***Figure 3: La hiérarchie schématique pour UML [2]***

### 3. Présentation du groupe de travail

Le groupe de travail représente l'ensemble des personnes nécessaires à la réalisation du projet. Ainsi, nous avons trois groupes de travail à savoir le groupe projet, le groupe de pilotage et le groupe utilisateur.

#### a. Groupe de pilotage

C'est le groupe dirigeant chargé de veiller au bon déroulement du projet. Il planifie les dates clés du projet, examine les propositions du groupe de projet, et décide des orientations stratégiques. Ce groupe est constitué de

- M. NIKIEMA Laurent, notre Maître de stage
- M. TOPAN Bia, PDG de l'entreprise

#### b. Groupe de projet

Le groupe de projet est l'ensemble des personnes chargées de réaliser le projet. Il est l'intermédiaire entre le groupe de pilotage et le groupe des utilisateurs. Ce groupe est composé principalement de David Eliézer DANAYE, étudiant en 3ème année en Méthodes Informatiques Appliquées à la Gestion (MIAGE) à l'IBAM.

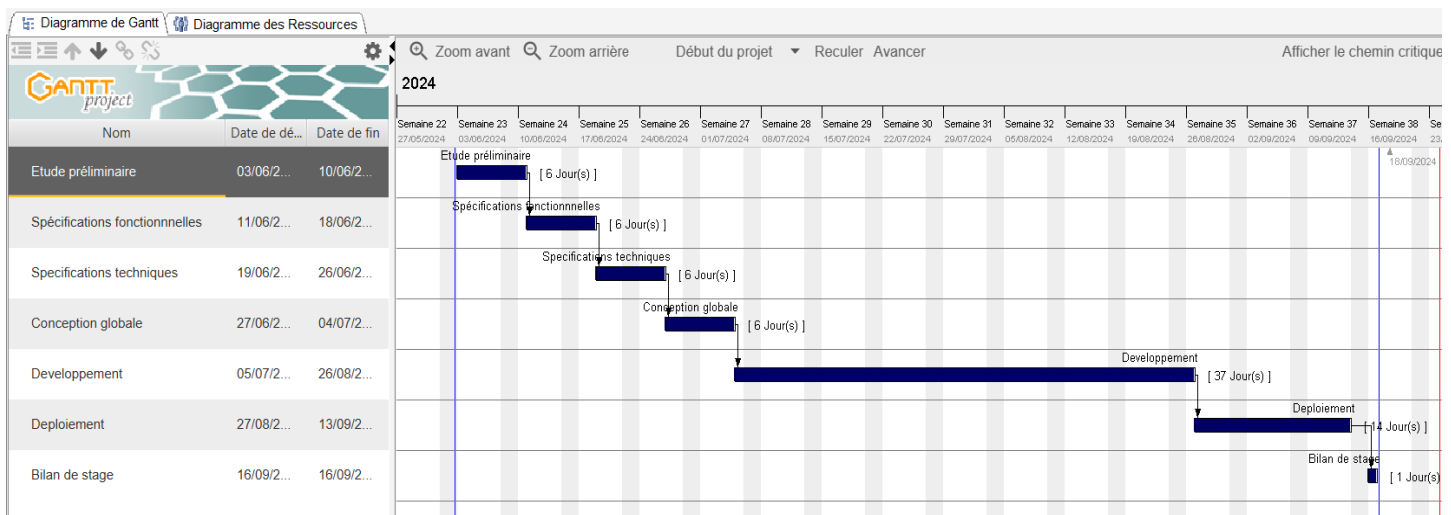
#### c. Groupe utilisateur

Ce groupe est constitué de l'ensemble de tous ceux qui vont utiliser le futur système. Il participe à la capture des besoins fonctionnels. Il est composé de principalement des employés de la SONABEL.



#### d. Planning de travail

Pour mener à bien notre étude et la réaliser dans les délais, nous avons subdivisé le projet en tâches. Pour cela, nous avons utilisé le diagramme de Gantt qui est un outil utilisé en ordonnancement et en gestion de projet ; il permet de visualiser (à l'aide d'un graphe) dans le temps les différentes tâches liées à un projet. Le planning prévisionnel est résumé dans le diagramme de Gantt ci-dessous :



**Figure 4: Planning de réalisation du projet**

Après cette phase d'étude préalable du système, nous allons passer à la phase d'étude détail

## **II. Expressions des besoins**

Cette partie consiste à identifier les différents acteurs qui vont interagir dans le futur système ainsi que les différents messages qu'ils échangeront entre eux.

### **1. Description du fonctionnement attendu de l'application**

L'application que nous mettrons en place devra permettre à l'entreprise de suivre l'évolution des marchés publics depuis l'expression des besoins jusqu'à l'attribution des marchés. Cette application sera fournie sous une version web. Le processus sera détaillé comme suit :

L'administrateur enregistre les employés dans le système et leur crée des comptes utilisateurs. Une fois authentifiés les agents pourront importer les plans dans le logiciel, et pour chaque ligne du plan de passation ils pourront créer des dossiers qui leurs seront affectés. Une fois les dossiers créés les employés auront la possibilité d'effectuer un suivi sur chaque dossier qui consistera pour chacun des dossiers dans un premier temps à enregistrer les avis de publication

suit à un retour de la DGCMEF, ensuite à réceptionner l'ensemble des offres liés à chaque lot des différents dossiers, en outre il auront également la possibilité de publier les résultats liés à chaque lot des dossiers en leurs affectant des attributeurs provisoires et enfin il seront capable d'attribuer des marchés à ces lots.

Une fois ces étapes réunies le directeur pourra visualiser l'état de chaque dossier afin de connaître si ces derniers sont soit **non entamés** soit **lancés pas encore attribués** ou **en cours d'exécutions** qui sont les trois principaux états auxquels ces dossiers peuvent appartenir. Le directeur pourra également visualiser l'évolution globale de tous les dossiers afin d'avoir une idée sur l'avancement générale des dossiers. Et pour terminer il aura en sortie une vue sur l'état complet du plan de passation.

## 2. Spécifications fonctionnelles

La spécification fonctionnelle est la description des fonctionnalités d'un logiciel en vue de sa réalisation. Dans cette partie, nous décrirons dans les détails les exigences du futur système à travers l'identification des acteurs, des cas d'utilisation et les différents diagrammes.

### a. Identification des acteurs

Un acteur représente un ensemble cohérent de rôles qu'un utilisateur ou une entité externe peut jouer en interagissant avec le système. Dans notre cas il s'agit bien :

- de l' employé ou agent ;
- du directeur;
- de l'administrateur;

### b. Identifications des cas d'utilisations

Les cas d'utilisations désignent l'ensemble des interactions qui vont permettre à un acteur d'atteindre un objectif dans un système. Le tableau suivant liste les cas d'utilisation de notre futur.

**Tableau 3: Liste des cas d'utilisations**

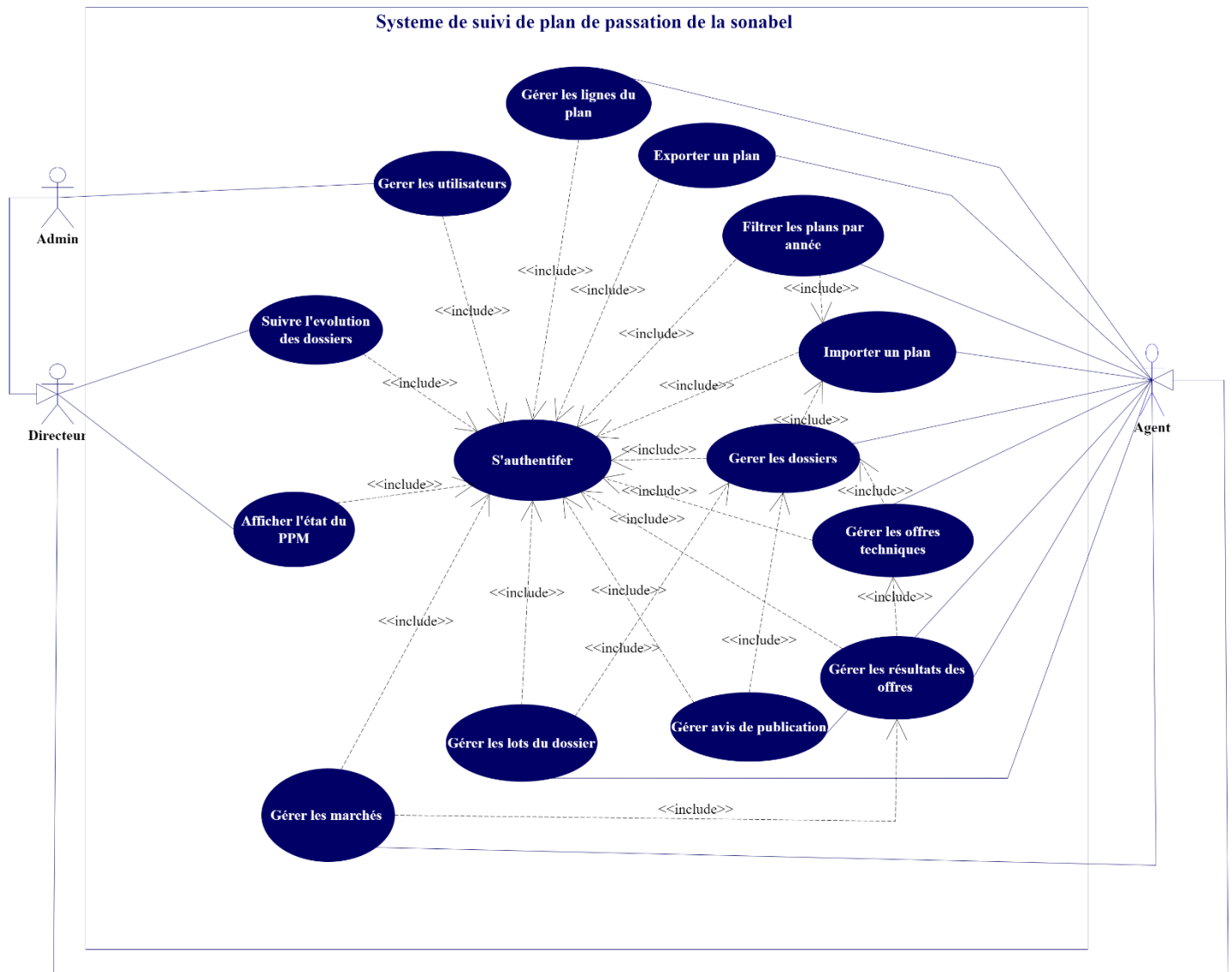
Numéro	Cas d'utilisation	Description	Auteur (s)
CU01	S'authentifier	Permet aux employés d'accéder au système	Tous les acteurs

<b>CU02</b>	Gérer les employés	Permet d'enregistrer, modifier, supprimer un employé	Administrateur
<b>CU03</b>	Importer les plans	Permet d'importer les plans par année dans le logiciel depuis un fichier csv	Tous les acteurs
<b>CU04</b>	Gérer les lignes des plan	Permet d'ajouter, de modifier, de supprimer et d'afficher les lignes de chaque plan	Tous les acteurs
<b>CU05</b>	Filtrer les plans	Permet d'afficher les plans par année sélectionnée	Tous les acteurs
<b>CU06</b>	Gérer les dossiers	Permet de créer, de modifier, de supprimer, et de lister les dossiers.	Tous les acteurs
<b>CU07</b>	Gérer les lots	Permet d'ajouter, de modifier, de supprimer, et de lister les lots	Tous les acteurs
<b>CU08</b>	Gérer les avis de publication	Permet d'enregistrer, de modifier, de supprimer et de lister les avis de publication	Tous les acteurs
<b>CU09</b>	Gérer les offres	Permet de créer, de modifier, de supprimer et de visualiser les offres associées à chaque lot	Tous les acteurs
<b>CU10</b>	Gérer résultats	Permet de créer, de modifier, de supprimer, d'afficher les résultats des offres	Tous les acteurs

<b>CU11</b>	Gestion des marchés	Permet de créer, de modifier, de supprimer, de lister le marché attribué à chaque lot.	Tous les acteurs
<b>CU12</b>	Suivre l'évolution des dossiers	Permet de connaître l'état de chaque dossier et de tout les dossiers	Administrateur, Directeur
<b>CU13</b>	Afficher le plan de passation	Permet de voir l'état global du plan de passation	Administrateur, Directeur
<b>CU14</b>	Exporter les plans	Permet d'exporter les plans de passation	Administrateur, Directeur

### c. Diagramme des cas d'utilisation

Un diagramme de cas d'utilisation est un diagramme UML qui illustre les interactions entre les utilisateurs (ou acteurs) et le système. Il décrit les différentes façons dont les utilisateurs peuvent interagir avec le système pour accomplir une tâche spécifique. Il permet d'une part de modéliser les besoins des utilisateurs en les clarifiant et en les organisant et d'autre part d'identifier les acteurs et les fonctionnalités du système comme le montre la figure 5 ci-dessous :



**Figure 5: Diagramme de cas d'utilisations**

#### d. Description textuelle de certains cas d'utilisation

Contrairement au diagramme de cas d'utilisation qui décrit les grandes fonctionnalités d'un système du point de vue des acteurs, la description textuelle permet d'exposer de façon détaillée le dialogue entre les acteurs et le système. Pour faciliter la lecture de notre rapport nous nous limiterons à la description de quatre (4) cas d'utilisations. Les cas d'utilisations qui sont décrits dans cette partie sont essentiellement : s'authentifier, importer un plan, enregistrer un dossier et ajouter un lot.

### ✚ CU01 : S'authentifier

Le tableau 4 décrit les interactions entre les acteurs et le système pour le cas d'utilisation « S'authentifier ». D'abord, l'utilisateur demande à se connecter, puis il reçoit un formulaire de connexion qu'il doit remplir en fournissant son nom et son mot de passe. Une fois soumis, le système vérifie les informations saisies. Si les informations sont incorrectes le système lui renvoie le même formulaire pour qu'il réessaie. Au cas où les informations sont correctes l'utilisateur est dirigé vers la page d'accueil qui le concerne. La procédure est valable pour tous les acteurs du système.

**Tableau 4:Description du cas d'utilisation « S'authentifier »**

Résumé	Ce cas permet à l'utilisateur de s'authentifier au système
<b>Acteurs</b>	Tous les acteurs
<b>Précondition</b>	Néant
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur demande à se connecter</li> <li>2. Le système l'affiche un formulaire de connexion</li> <li>3. L'acteur remplit le formulaire et soumet au système</li> <li>4. Le système vérifie les données saisies [A]</li> <li>5. Le système redirige l'acteur vers la page principale</li> </ol>
<b>Scénario Alternatif</b>	[A] : Identifiants ou mot de passe incorrect
	<ol style="list-style-type: none"> <li>1. Le système notifie l'acteur de l'échec de la connexion</li> <li>2. Le système renvoie le formulaire de connexion et le scénario reprend à partir du point 3 du scénario nominal.</li> <li>3. Le système renvoie le formulaire de connexion et le scénario reprend à partir du point 3 du scénario nominal.</li> </ol>
<b>Post conditions</b>	L'acteur accède à son tableau de bord

### ✚ CU03 : Importer un plan

Le tableau 5 décrit les interactions entre les acteurs et le système pour le cas d'utilisation « Importer un plan ». D'abord, l'utilisateur doit réaliser le cas d'utilisation s'authentifier pour pouvoir importer un plan, une fois authentifié il demande à importer un plan. Le système lui envoie le formulaire correspondant puis il charge le fichier qu'il veut importer depuis le disque. Le système vérifie les données. Si les données sont invalides le système lui notifie de l'échec de l'importation au cas où les données sont valides le système importe le fichier et notifie que le plan a été importé.

**Tableau 5: Description du cas d'utilisation « importer un plan »**

Résumé	Ce cas permet d'importer un plan
<b>Acteurs</b>	Administrateur, Directeur, Agent.
<b>Précondition</b>	S'authentifier
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur demande à importer un plan</li> <li>2. Le système lui envoie le formulaire d'importation de plan</li> <li>3. Il sélectionne un fichier dans le disque.</li> <li>4. Le système vérifie les données [A]</li> <li>5. Le système notifie que le plan a été importé</li> </ol>
<b>Scénario Alternatif</b>	<p>[A] : Un format de fichier non adéquat</p> <ol style="list-style-type: none"> <li>1. Le système notifie l'acteur de l'échec de l'importation en lui précisant d'importer un fichier de type csv.</li> <li>2. Le système renvoie le formulaire d'importation et le scénario reprend à partir du point 3 du scénario nominal.</li> </ol>
<b>Post conditions</b>	Le système importe le fichier

#### ❖ CU06 : Enregistrer un dossier

Le tableau 6 décrit les interactions entre les acteurs et le système pour le cas d'utilisation « Enregistrer un dossier ». Bien avant de réaliser ce cas d'utilisation, l'acteur doit nécessairement passer par les cas d'utilisations s'authentifier et importer un plan. Une fois ces cas d'utilisation effectués il demande à enregistrer un dossier. Le système lui renvoie le formulaire

d'enregistrement, formulaire qu'il remplira et soumettra au système. Une fois soumis, le système vérifie les informations saisies. Si les informations sont incorrectes le système lui renvoie le même formulaire pour qu'il réessaie. Au cas où les informations sont correctes le système enregistre le dossier et notifie que le dossier a été ajouté.

**Tableau 6 : Description du cas d'utilisation « Enregistrer un dossier »**

Résumé	Ce cas permet d'enregistrer un dossier
<b>Acteurs</b>	Administrateur, Directeur, Agent.
<b>Précondition</b>	S'authentifier, importer un plan.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur demande à enregistrer un dossier pour une ligne de plan concernée.</li> <li>2. Le système lui envoie le formulaire</li> <li>3. Il remplit le formulaire d'enregistrement</li> <li>4. Le système vérifie les données [A]</li> <li>5. Le système enregistre les données</li> <li>6. Le système envoie une notification de succès</li> </ol>
<b>Scénario Alternatif</b>	[A] : Champs mal remplis
	<ol style="list-style-type: none"> <li>1. Le système notifie l'acteur de l'échec de l'enregistrement en lui précisant les champs manquants</li> <li>2. Le système renvoie le formulaire d'enregistrement et le scénario reprend à partir du point 3 du scénario nominal.</li> </ol>
<b>Post conditions</b>	Le système enregistre le dossier et notifie que le dossier a bien été enregistré.

**❖ CU07 : Ajouter un lot**

Le tableau 7 décrit les interactions entre les acteurs et le système pour le cas d'utilisation « Ajouter un lot ». Bien avant de réaliser ce cas d'utilisation, l'acteur doit nécessairement passer par les cas d'utilisations s'authentifier, importer un plan et enregistrer un dossier. Une fois ces cas d'utilisation effectués il demande à ajouter un lot. Le système lui renvoie le formulaire



d'enregistrement, formulaire qu'il remplira et soumettra au système. Une fois soumis, le système vérifie les informations saisies. Si les informations sont incorrectes le système lui renvoie le même formulaire pour qu'il réessaie. Au cas où les informations sont correctes le système enregistre le lot et notifie que le lot a été ajouté.

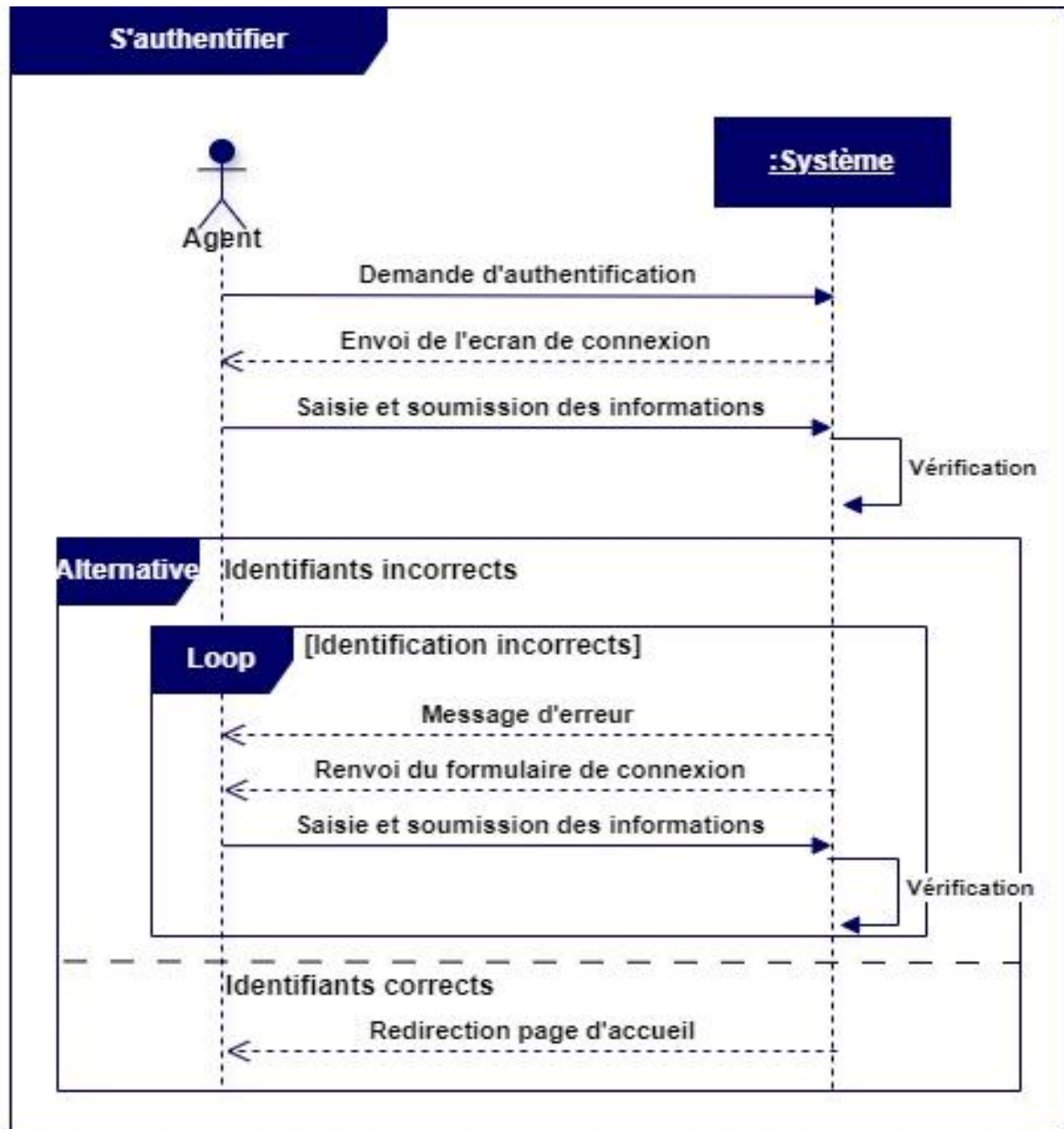
**Tableau 7: Description du cas d'utilisation « Ajouter un lot »**

Résumé	Ce cas permet d'ajouter un lot
<b>Acteurs</b>	Administrateur, Directeur, Agent.
<b>Précondition</b>	S'authentifier, importer un plan, enregistrer un dossier
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur demande à ajouter un lot pour un dossier concerné.</li> <li>2. Le système lui envoie le formulaire</li> <li>3. Il remplit le formulaire d'enregistrement</li> <li>4. Le système vérifie les données [A]</li> <li>5. Le système enregistre les données</li> <li>6. Le système envoie une notification de succès</li> </ol>
<b>Scénario Alternatif</b>	[A] : Champs mal remplis
	<ol style="list-style-type: none"> <li>1. Le système notifie l'acteur de l'échec de l'enregistrement en lui précisant les champs manquants</li> <li>2. Le système renvoie le formulaire d'enregistrement et le scénario reprend à partir du point 3 du scénario nominal.</li> </ol>
<b>Post conditions</b>	Le système ajoute le lot et affiche la liste des lots

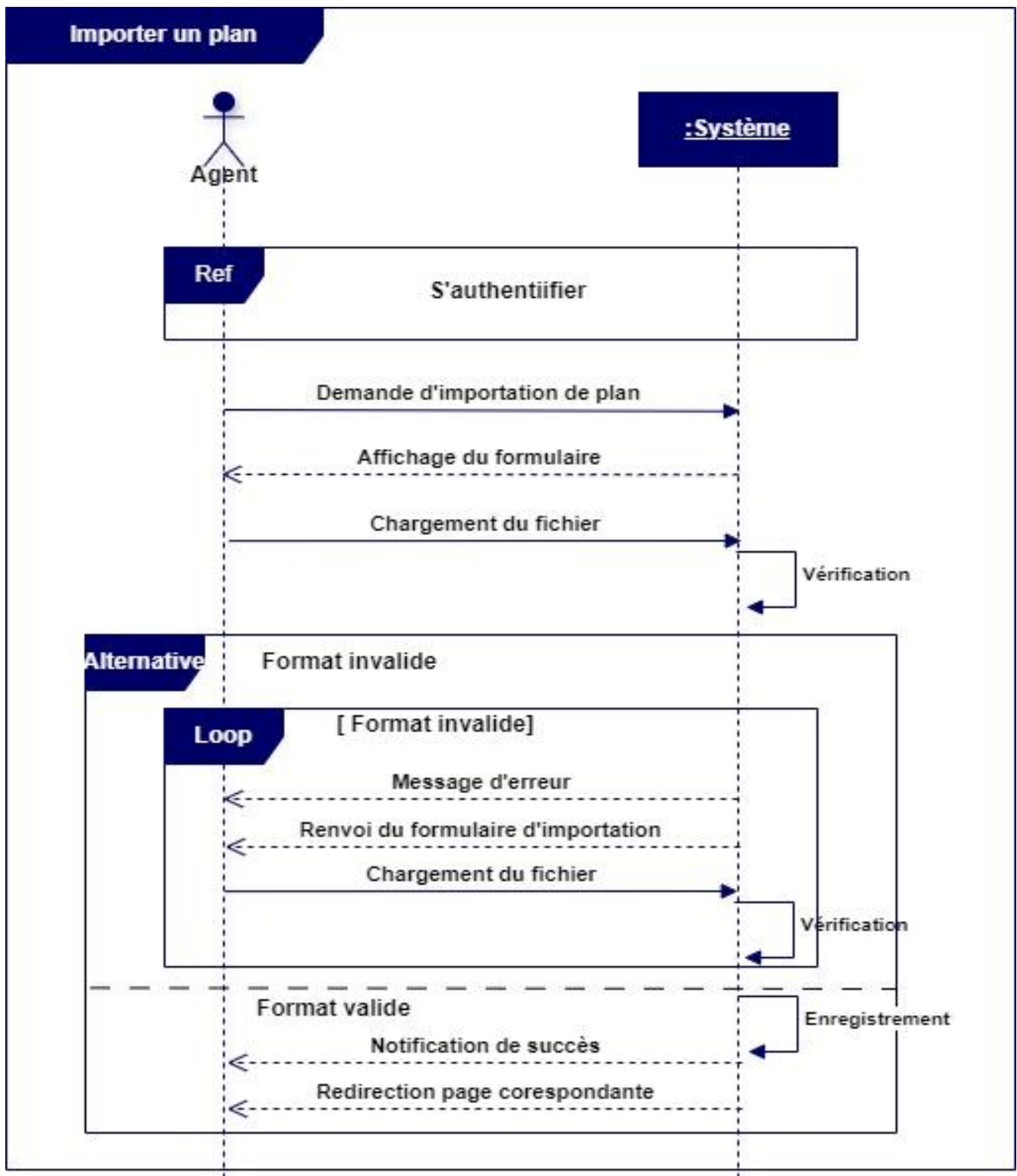
#### e. Diagramme de séquence

Les diagrammes de séquences représentent les interactions entre le système et les acteurs en montrant sous forme de scénarios la chronologie des envois de messages issus d'un cas d'utilisation donné. Les diagrammes de séquences qui sont décrits dans cette partie sont essentiellement : s'authentifier, importer un plan et enregistrer un dossier.

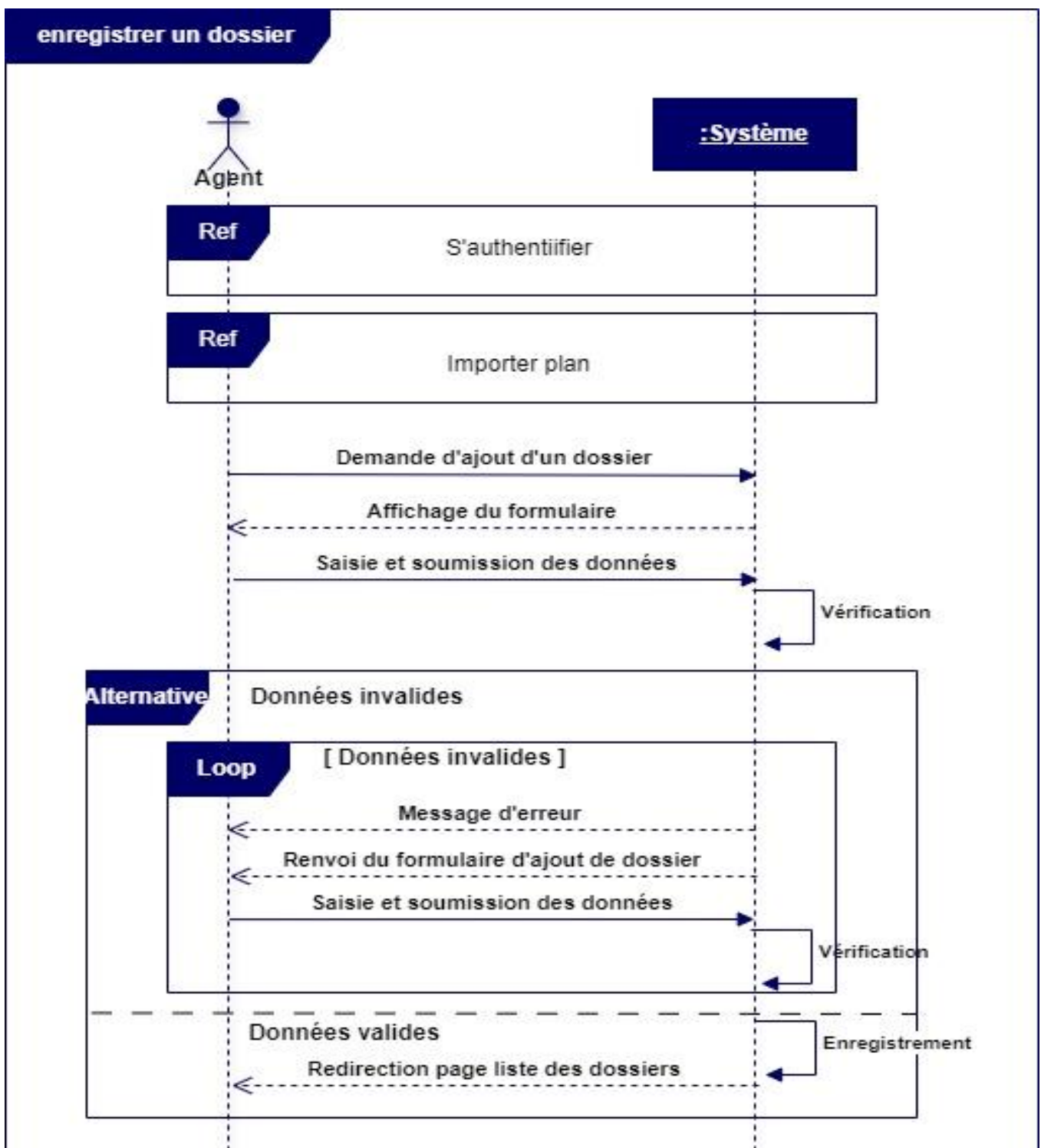
**NB :** La lecture des différents diagrammes de séquence se fait comme suit : les flèches orientées vers la droite présentent les messages de demande effectués par les acteurs et celles orientées vers la gauche présentent les messages de réponse renvoyés par le système. Les lignes verticales présentent les temps d'activités des acteurs et la lecture se fait du haut en bas.



**Figure 6: Diagramme de séquence vue de l'extérieur pour CU « s'authentifier »**



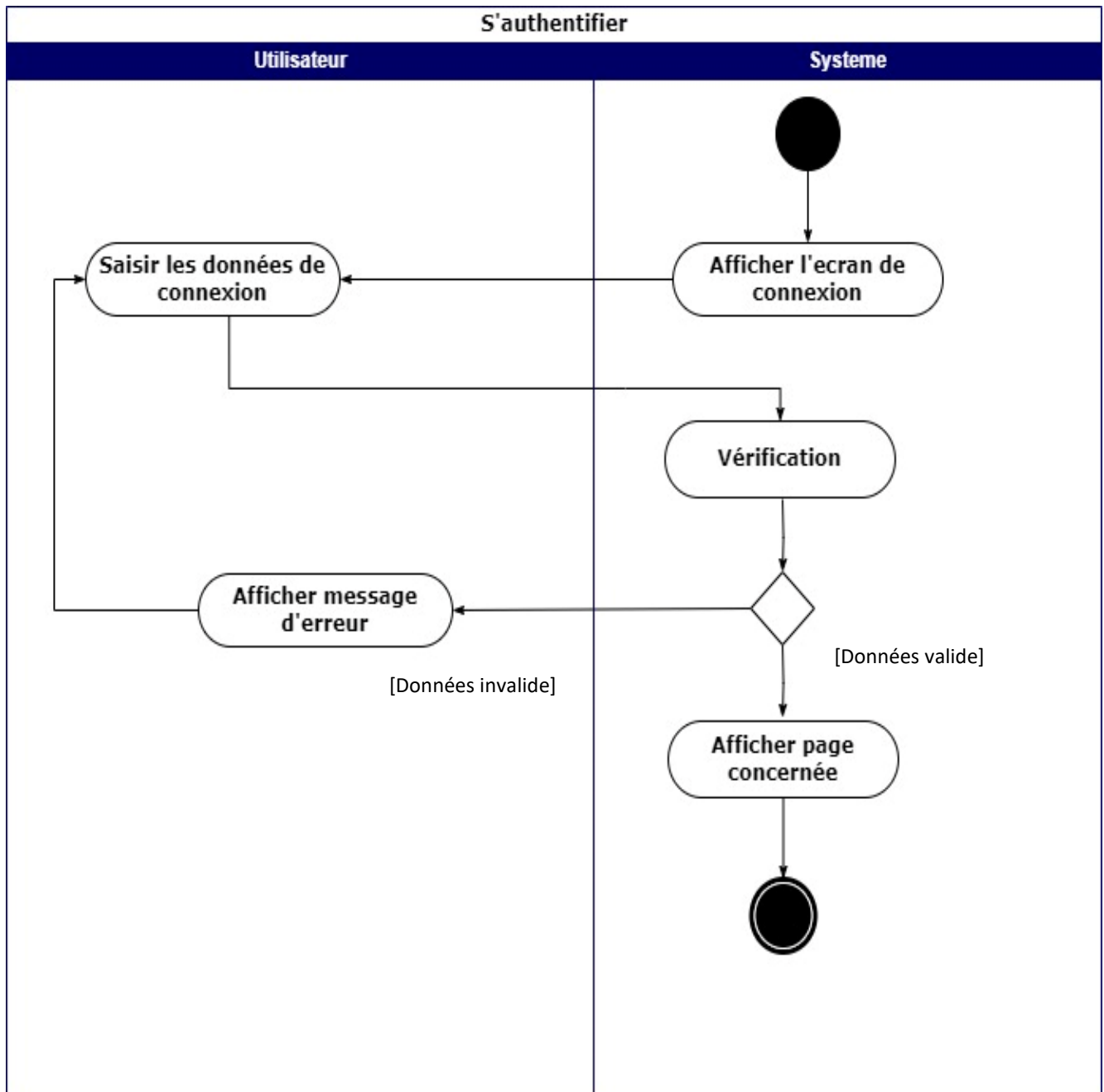
***Figure 7: Diagramme de séquence vue de l'extérieur pour CU « Importer un plan »***



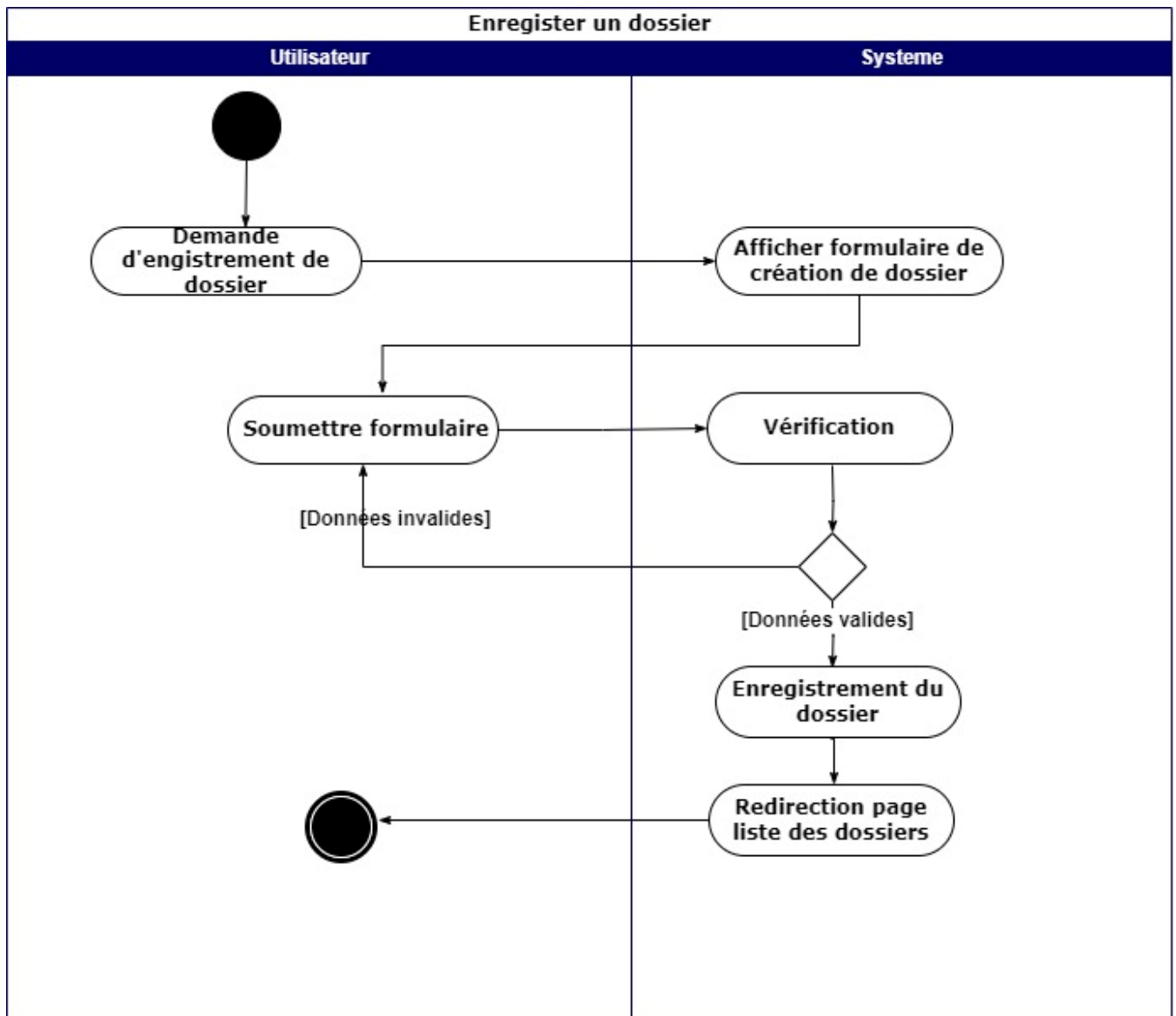
**Figure 8: Diagramme de séquence vue de l'extérieur pour CU « Enregistrer un dossier »**

### f. Diagramme d'activité

Le diagramme d'activité offre une visualisation graphique et détaillée du déroulement d'un cas d'utilisation donné entre les acteurs et le système. Les cas d'utilisation spécifiques abordés dans cette section sont : "S'authentifier" et "Enregistrer un dossier".



***Figure 9: Diagramme d'activité pour le CU « S'authentifier »***



**Figure 10:Diagramme d'activité pour le CU « Enregistrer un dossier »**

### 3. Spécifications techniques

La spécification technique est la description des choix techniques pris pour satisfaire les spécifications fonctionnelles et les besoins.

#### a. Mise à disposition des conditions de travail

LOGO SERVICES a mis à notre disposition un serveur pour effectuer nos tests ainsi que la connexion internet pour nos différents travaux et recherches dans le but de parfaire notre travail. Pour le développement, nous avons utilisé notre ordinateur portable personnel.

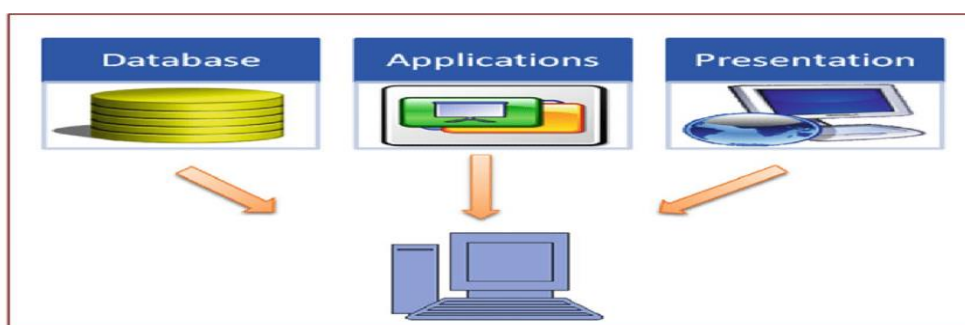
### b. Architecture de développement

Une architecture de développement désigne un mode de communication à travers un réseau entre plusieurs éléments physiques et/ou logiques. Il en existe plusieurs types dont les plus répandus et performants sont du type un tiers (1) deux (2) tiers et trois (3) tiers. Le tableau suivant est un tableau comparatif de ses deux architectures.

**Tableau 8 : Tableau comparatif des architectures de développement**

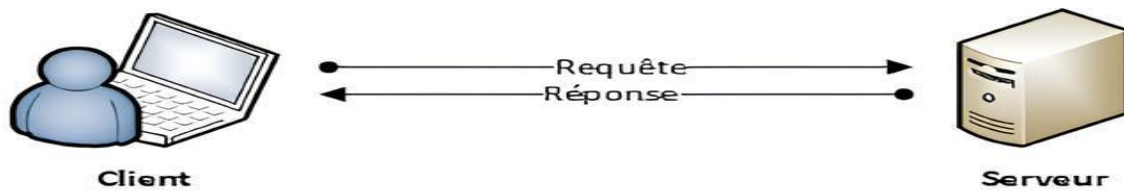
Architecture	Avantages	Inconvénients
<b>Architecture un tiers</b>	<ul style="list-style-type: none"> <li>• Simplicité de mise en œuvre</li> <li>• Les tables des données sont directement accessibles.</li> <li>• Coûts réduits</li> </ul>	<ul style="list-style-type: none"> <li>• Maintenance difficile du système</li> <li>• Sécurité faible</li> </ul>
<b>Architecture deux tiers</b>	<ul style="list-style-type: none"> <li>• Séparation des données et de la logique métier</li> <li>• Charges réparties entre le client et le serveur</li> </ul>	<ul style="list-style-type: none"> <li>• Coût de déploiement élevé</li> <li>• Évolution difficile du système</li> </ul>
<b>Architecture trois tiers</b>	<ul style="list-style-type: none"> <li>• Évolution facile</li> <li>• Déploiement aisé</li> <li>• Une sécurité accrue</li> </ul>	<ul style="list-style-type: none"> <li>• nécessite un personnel informatique initié pour sa mise en œuvre</li> <li>• Complexité de mise en œuvre</li> </ul>

La Figure 11 ci-dessous illustre les informations données dans le tableau 8 sur l'architecture un tiers.



**Figure 11: Architecture 1 tier [3]**

La Figure 12 ci-dessous illustre les informations données dans le tableau 8 sur l'architecture deux tiers.

**Figure 12: Schéma représentant l'architecture 2 tiers [4]**

La Figure 13 ci-dessous illustre parfaitement les informations données dans le tableau 8 sur l'architecture trois tiers.

**Figure 13: Schéma représentant l'architecture 3 tiers [5]**

**Solution retenue :** A la suite de l'étude comparative que nous avons menée sur les trois (3) types d'architectures, notre choix s'est porté sur l'architecture trois (3) tiers. En effet, elle offre une facilité de déploiement, une meilleure sécurité en séparant les couches présentation, logique et persistance des données et permettant également la maintenance et la réutilisation code.

### III. Conception globale

Dans cette partie, nous mettrons en avant le dictionnaire de données, le diagramme des classes ainsi que celui du déploiement.

#### 1. Diagramme de classe

Dans cette partie nous mettrons en évidence le dictionnaire de données puis notre diagramme des classes.

##### a. Dictionnaire de données

Le dictionnaire de données contient l'ensemble des descriptions des attributs de chaque classe qui est utilisé dans le système. Pour faciliter la lecture, nous ferons la description classe par classe. Nous décrirons quelques classes. Les autres seront décrites plus bas (**ANNEXE 3**).



### † Fournisseur

Cette classe permet de recueillir les informations d'un fournisseur.

**Tableau 9: Description de la classe « Fournisseur »**

Attribut	Description	Type de donnée
<b>nom_four</b>	Le nom du fournisseur	Chaine de caractères
<b>recm</b>	Le numéro du Registre du Commerce et du Crédit Mobilier	Chaine de caractères
<b>ifu</b>	Le numéro d'identification fiscale unique	Chaine de caractères
<b>telephone1</b>	Le premier numéro de téléphone	Chaine de caractères
<b>telephone2</b>	Le deuxième numéro de téléphone	Chaine de caractères
<b>adresse</b>	L'adresse physique ou postale du fournisseur.	Chaine de caractères
<b>email</b>	L'adresse e-mail du fournisseur	Chaine de caractères
<b>domaine</b>	Le domaine ou secteur d'activité	Chaine de caractères
<b>type</b>	Le type de fournisseur	Chaine de caractères

### † Dossier

Cette classe permet de recueillir les informations d'un dossier.

**Tableau 10: Description de la classe « Dossier »**

Attribut	Description	Type de donnée
<b>numero_doss</b>	Le numéro unique attribué au dossier	Entier
<b>intitule_doss</b>	L'intitulé ou le titre du dossier	Chaine de caractères
<b>date_trans_sign</b>	La date à laquelle le dossier a été envoyé pour signature	Date
<b>date_retour_sign</b>	La date de retour du dossier après signature	Date
<b>date_retour_dgcmef</b>	La date à laquelle le dossier a été transmis à la DGCMEF	Date
<b>fichier</b>	Un fichier joint au dossier	Fichier

### ✚ Offres

Cette classe permet de recueillir les informations d'une offre.

**Tableau 11: Description de la classe « Offre »**

Attribut	Description	Type de donnée
<b>off_doss_id</b>	Un identifiant spécifique à l'offre	Entier
<b>date_prevu_reception</b>	La date prévue pour la réception des offres.	Date
<b>date_reel_reception</b>	La date réelle à laquelle l'offre a été reçue.	Date
<b>offre_technique</b>	La date réelle à laquelle l'offre a été reçue.	Fichier

### ✚ Plan

Cette classe permet de recueillir les informations d'un plan.

**Tableau 12: Description de la classe « Plan »**

Attribut	Description	Type de donnée
<b>id</b>	Identifiant du plan	Entier
<b>année</b>	L'année du plan	Date
<b>date_plan</b>	La date de création du plan	Date

### † Résultat

Cette classe permet de recueillir les informations d'un plan.

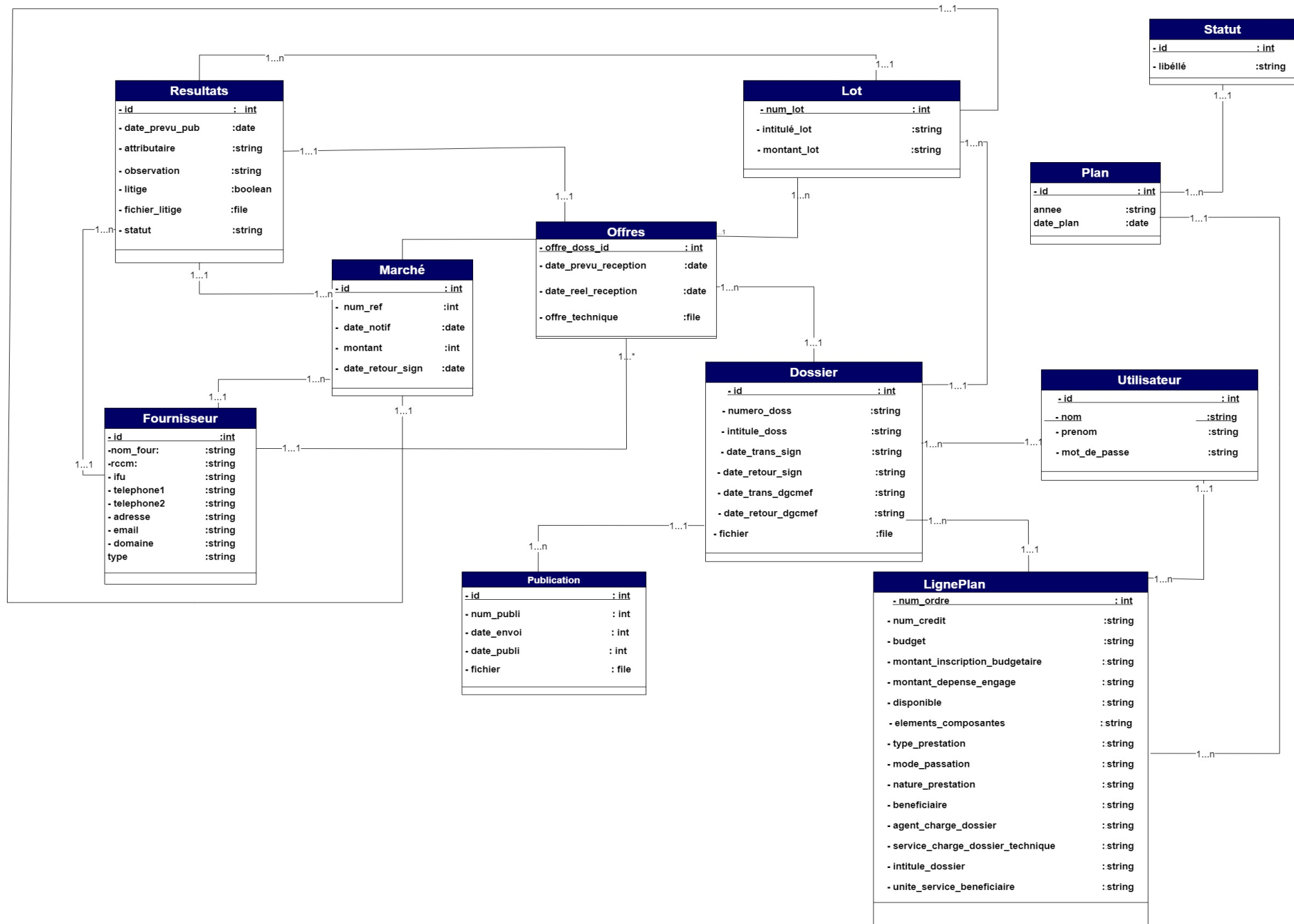
**Tableau 13: Description de la classe « Résultat »**

Attribut	Description	Type de donnée
<b>num_par_res</b>	Numéro du resultat	Chaine de caractères
<b>litige</b>	Indique s'il y a un litige lié au résultat du marché	Booléen
<b>fichier_litige</b>	Un fichier concernant un litige	Chaine de caractères
<b>date_prevu_pub</b>	La date prévue de publication des résultats	Date
<b>date_reelle_pub</b>	La date réelle à laquelle les résultats ont été publiés	Date
<b>observation</b>	Des remarques ou observations éventuelles sur le résultat	Chaine de caractères
<b>statut</b>	Le statut du résultat de l'offre	Booléen

### b. Diagramme de classe

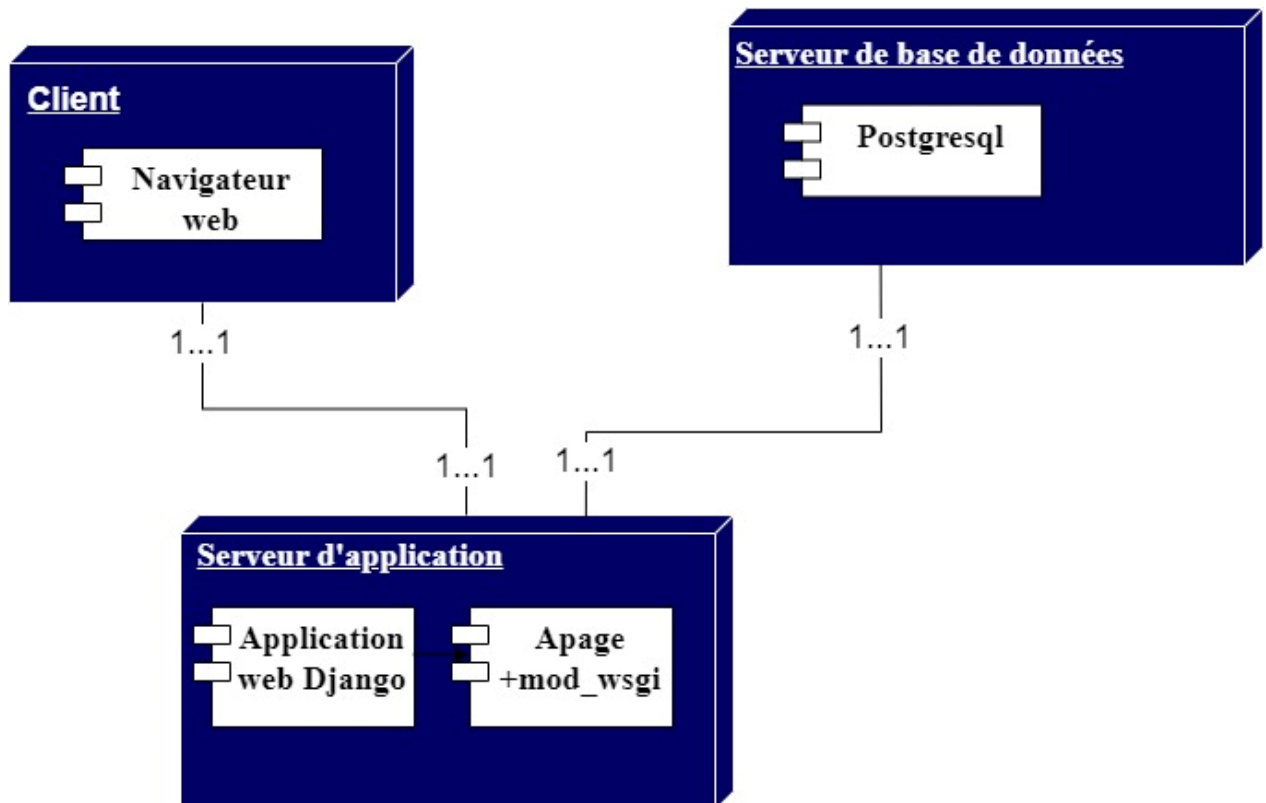
Le diagramme de classe exprime la structure statique du système en ce qui concerne les classes et la relation entre ces classes. Nous présentons notre diagramme de classe à la figure 14. Il possède onze (11) classes, à l'intérieur desquelles nous retrouvons les données ou attributs

issus du dictionnaire des données. Nous remarquons que les classes sont représentées par des rectangles arrondis et identifiées par des noms. Les classes sont reliées entre elles par des associations portant des cardinalités telles que : « 1...\* » qui signifie un ou plusieurs, « 0...\* » qui signifie zéro ou plusieurs et « 1...1 » ou « 1 » qui signifie un et un seul. Considérons les classes « Dossier » et « Offre ». Nous remarquons que sur la liaison entre « Dossier » et « Offre », nous avons les cardinalités « 1...n » du côté Offre, « 1...1 » du côté Dossier. Cela signifie qu'une instance de la classe « Dossier » peut avoir plusieurs instances de la classe « Offre » et une instance de la classe « Offre » concerne une et une seule instance de la classe « Dossier ».



**Figure 14: Diagramme de classes****2. Diagramme de déploiement**

Le diagramme de déploiement est un type de schéma UML qui illustre la configuration matérielle des différents éléments nécessaires à l'exécution du système, ainsi que les instances de composants qu'ils abritent. Ce diagramme est constitué de "nœuds" reliés entre eux par des connexions physiques.

**Figure 15: Diagramme de déploiement**

Notre diagramme de déploiement comprend quatre (3) nœuds qui représentent le serveur de données, le serveur d'application et le poste client.

Ce diagramme de déploiement laisse entrevoir l'architecture logicielle trois tiers qui est utilisée pour notre système. Après cette partie réservée à la conception globale du système, nous passons maintenant à la réalisation du logiciel en question.

**IV. Réalisation**

Pour concevoir ce système, nous avons utilisé plusieurs outils. Nous commencerons par détailler ces outils, suivis par une présentation de l'architecture MVT. Par la suite, nous aborderons les tests réalisés et la politique de sécurité adoptée. Enfin, nous procéderons à une

analyse des coûts de développement et concluons par une présentation de quelques écrans de la plateforme.

## 1. Présentation des outils de réalisations

Les outils de réalisation sont des éléments qui aident un développeur dans le déroulement d'une activité de développement. Parmi ces éléments, nous étudierons notamment les langages utilisés, le framework de développement et les outils de conception.

### a. Langages

Un langage informatique désigne un ensemble de règles et de conventions utilisé pour écrire des programmes qui ordonnent à un ordinateur d'exécuter des tâches précises. L'utilisation de langages informatiques est essentielle car ils permettent aux développeurs de communiquer efficacement avec les machines, de créer des applications variées et de résoudre des problèmes complexes de manière structurée et compréhensible.

Nous avons utilisé plusieurs langages pour le développement de notre application. Ce sont :

- **Python<sup>8</sup>** : Ce langage de programmation interprété, polyvalent et de haut niveau est apprécié pour sa simplicité d'utilisation, sa grande flexibilité et son application dans de nombreux domaines, tels que le développement d'applications web, l'analyse de données ou encore l'automatisation des processus.
- **JavaScript<sup>9</sup>** : Ce langage de script est principalement utilisé pour créer des pages web interactives, constituant ainsi un élément clé dans le développement d'applications web.
- **HTML5<sup>10</sup> (HyperText Markup Language)** : (HyperText Markup Language) : Il s'agit d'un langage de balisage permettant de concevoir et d'écrire des pages pour le World Wide Web.
- **CSS 3<sup>11</sup> (Cascading Style Sheets)** : Le CSS est utilisé pour styliser et mettre en forme les pages web.

### b. Framework de développement

Un framework constitue un ensemble structuré de bibliothèques, d'outils et de pratiques recommandées qui fournissent une architecture préconçue pour le développement

---

<sup>8</sup> Python : [Bienvenue chez Python.org](https://www.python.org/)

<sup>9</sup> JavaScript : [JavaScript - Apprendre le développement web | MDN \(mozilla.org\)](https://developer.mozilla.org/fr/docs/Web/JavaScript)

<sup>10</sup> HTML : [Tutoriel HTML \(w3schools.com\)](https://www.w3schools.com/html/)

<sup>11</sup> CSS : [Tutoriel CSS \(w3schools.com\)](https://www.w3schools.com/css/)

d'applications. Contrairement aux bibliothèques classiques, qui offrent des fonctionnalités à la demande, un framework impose une organisation standardisée des composants du logiciel.

Nous avons utilisé plusieurs framework pour le développement de notre application. Ce sont :

- **Django**<sup>12</sup> : Django est un framework web open-source écrit en Python qui permet de développer rapidement des applications web robustes, sécurisées et évolutives.
- **Bootstrap**<sup>13</sup> **4** : Bootstrap est un Framework front-end gratuit pour un développement Web plus rapide et plus facile. Il comprend des modèles de conception basés sur HTML et CSS.

### c. Outils de conception

- **Draw io**<sup>14</sup> : Étant un outil de conception, il nous a offert la possibilité de réaliser nos différents diagrammes UML.
- **GanttProject**<sup>15</sup> : Nous l'avons utilisé pour construire notre diagramme de GANTT.
- **Figma**<sup>16</sup> : Utilisé pour le prototypage de sites web et d'applications, il nous a aidé à élaborer les maquettes de certains écrans de notre application.
- **PgAdmin**<sup>17</sup> version 4 : il nous a été utile pour l'administration notre base de données.
- **Git**<sup>18</sup> : Git, un système de gestion de versions distribué, nous a servi à organiser et à archiver le code source de notre plateforme.
- **Visual Studio Code**<sup>19</sup> : c'est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Cet éditeur open source et extensible nous a permis de développer efficacement les différents composants de notre solution.

### d. Choix du système de gestion de base de données (SGBD)

En Informatique, il existe une diversité de système de gestion de base de données

Parmi lesquelles nous pouvons retenir :

<sup>12</sup> Django : [Le framework web pour les perfectionnistes avec des délais | Django \(djangoproject.com\)](https://www.djangoproject.com/)

<sup>13</sup> Bootstrap: [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](https://getbootstrap.com/)

<sup>14</sup> Draw io: [Créateur d'organigrammes et logiciel de diagramme en ligne \(diagrams.net\)](https://draw.io/)

<sup>15</sup> GanttProject: [GanttProject - Application gratuite de gestion de projet](https://www.ganttproject.biz/)

<sup>16</sup> Figma: <https://www.figma.com/fr-fr/design/>

<sup>17</sup> PgAdmin: <https://www.pgadmin.org/docs/>

<sup>18</sup> Git : [GitHub: Construisons à partir d'ici · GitHub](https://github.com/)

<sup>19</sup> Visual Studio Code : [Télécharger Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/)



- MySQL<sup>20</sup>
- PostgreSQL<sup>21</sup>
- Oracle<sup>22</sup>

Nous avons réalisé une étude comparative qui est la suivante :

**Tableau 14: Etude comparative des SGBD**

SGBD	Avantages	Inconvénients
<b>Oracle</b>	<ul style="list-style-type: none"> <li>- Oracle est conçu pour traiter de grandes quantités de données et est optimisé pour des environnements à haute charge.</li> <li>- Il peut facilement gérer des bases de données de grande taille avec une grande quantité d'utilisateurs simultanés.</li> <li>- Oracle est réputé pour sa robustesse et sa fiabilité, offrant une disponibilité élevée grâce à la technologie de clustering et de redondance.</li> <li>- Il propose une gamme étendue de fonctionnalités avancées pour la gestion des données, telles que le partitionnement, les index avancés, la gestion des droits d'accès, etc.</li> </ul>	<ul style="list-style-type: none"> <li>- Oracle est une solution coûteuse, en particulier pour les grandes entreprises, car il implique des frais de licence et de maintenance élevés.</li> <li>- En raison de ses fonctionnalités étendues, Oracle peut être complexe à gérer et à administrer, nécessitant des compétences spécialisées.</li> <li>- Oracle peut nécessiter des ressources matérielles importantes pour fonctionner efficacement, ce qui peut être un défi pour les petites entreprises.</li> </ul>

<sup>20</sup> MySQL : [En savoir plus sur MySQL](#)

<sup>21</sup> PostgreSQL : [En savoir plus sur PostgreSQL](#)

<sup>22</sup> Oracle : [En savoir plus sur Oracle](#)

<b>MySQL</b>	<ul style="list-style-type: none"> <li>- MySQL est une base de données open-source, ce qui signifie qu'elle est gratuite à utiliser et modifiable selon les besoins.</li> <li>- Il est relativement simple à installer, configurer et administrer, ce qui en fait un choix populaire pour les petites applications et les développeurs débutants.</li> </ul> <p>MySQL offre de bonnes performances pour les applications de taille moyenne et les sites web à trafic modéré.</p> <ul style="list-style-type: none"> <li>- En tant que projet opensource populaire, MySQL bénéficie d'une large communauté d'utilisateurs et de développeurs qui fournissent un support et des ressources.</li> </ul>	<ul style="list-style-type: none"> <li>- Bien qu'il soit évolutif, il peut avoir des limitations lorsqu'il s'agit de gérer de grandes charges de travail ou des applications très complexes.</li> </ul> <p>La gestion des transactions peut être moins robuste par rapport à certaines autres bases de données.</p>
<b>PostgreSQL</b>	<ul style="list-style-type: none"> <li>- PostgreSQL est réputé pour sa fiabilité, sa stabilité et sa conformité aux normes SQL. - Il offre de nombreuses fonctionnalités avancées, telles que le support de JSON, les transactions ACID, les vues matérialisées, etc.</li> <li>- Comme MySQL,</li> <li>- PostgreSQL est open-source, ce qui le rend gratuit et personnalisable.</li> </ul>	<ul style="list-style-type: none"> <li>- Bien que de plus en plus populaire, PostgreSQL est moins répandu que MySQL ou Oracle, ce qui peut signifier moins de ressources de support et de documentation. - La complexité des fonctionnalités avancées peut rendre son</li> <li>- Administration plus difficile pour les utilisateurs novices.</li> </ul>

À l'issue de notre étude comparative que nous avons menée nous a conduit à choisir PostgreSQL comme SGBD. Ce dernier, open source et gratuit, est soutenu par une large communauté. A sa facilité de déploiement et d'utilisation s'ajoute à son adoption au sein de l'entreprise où nous avons effectué notre stage, ce qui renforce notre décision.

#### e. Serveur d'application

Le serveur d'application assure l'hébergement d'une application et la gestion des interactions avec la base de données. L'objectif d'un serveur d'application est de permettre à partir d'un client aussi léger que possible, d'effectuer des traitements sur une machine distante. Les utilisateurs y accèdent principalement à travers un navigateur web. Voici quelques serveurs d'application :

- Tomcat<sup>23</sup>
- GlassFish<sup>24</sup>
- Oracle WebLogic Server<sup>25</sup>
- Nginx (proxy inverse)<sup>26</sup>
- Apache + mod\_wsgi<sup>27</sup>

Nous avons réalisé une étude comparative qui est la suivante :

**Tableau 15: Etude comparative des serveurs d'application**

Serveur d'application	Mémoire	Démarrage	Configuration	Communauté	Prix	Composants
<b>Tomcat</b>	Modérée	Rapide	Modérée	Forte	Gratuit	Serveur de servlets et JSP, support TLS/SSL.
<b>GlassFish</b>	Élevée	Modéré	Complexe	Moyenne	Gratuit	Serveur http, conteneur de servlets
<b>Oracle WebLogic Server</b>	Légère	Rapide	Facile	Très forte	Payant	Serveur complet avec tous les composants

<sup>23</sup> Tomcat : [Apprendre plus sur Tomcat](#)

<sup>24</sup> Glassfish: <https://glassfish.org/docs/>

<sup>25</sup> Oracle WebLogic Server: [Serveur Weblogic | Oracle](#)

<sup>26</sup> Nginx (proxy inverse): [Proxy inverse NGINX | NGINX Documentation](#)

<sup>27</sup> Apache + mod\_wsgi: [mod\\_wsgi — mod\\_wsgi 4.9.4 documentation \(modwsgi.readthedocs.io\)](#)

<b>Nginx (proxy inverse)</b>	Modérée	Rapide	Facile	Forte	Gratuit	HTTP Server, Proxy Inverse, Load Balancer
<b>Apache + mod_wsgi</b>	Modérée	Rapide	Très facile	Très forte	Gratuit	Serveur complet

Suite à l'étude comparative du Tableau 15, nous avons choisi la configuration Apache + mod\_wsgi. Apache, serveur web robuste, couplé au module mod\_wsgi, assure une gestion fiable des requêtes HTTP et des applications Python. Cette solution permet d'obtenir des performances optimales tout en assurant une gestion efficace des processus et une bonne intégration entre le serveur web et l'application. Par ailleurs, Apache est le serveur HTTP le plus utilisé sur le WWW (World Wide Web), soutenu par une communauté active de développeurs.

Nous allons à présent examiner l'architecture MVT de notre application, une composante centrale qui soutient la structuration et le bon déroulement du système.

## 2. Présentation de l'architecture MVT de notre application

Le développement de l'application web s'appuie sur l'architecture Model-View-Template (MVT), une caractéristique fondamentale du framework Django. Cette architecture favorise une séparation claire des préoccupations, facilitant ainsi la gestion du code, la maintenance de l'application, et la collaboration entre les développeurs. Voici les détails des composants principaux :

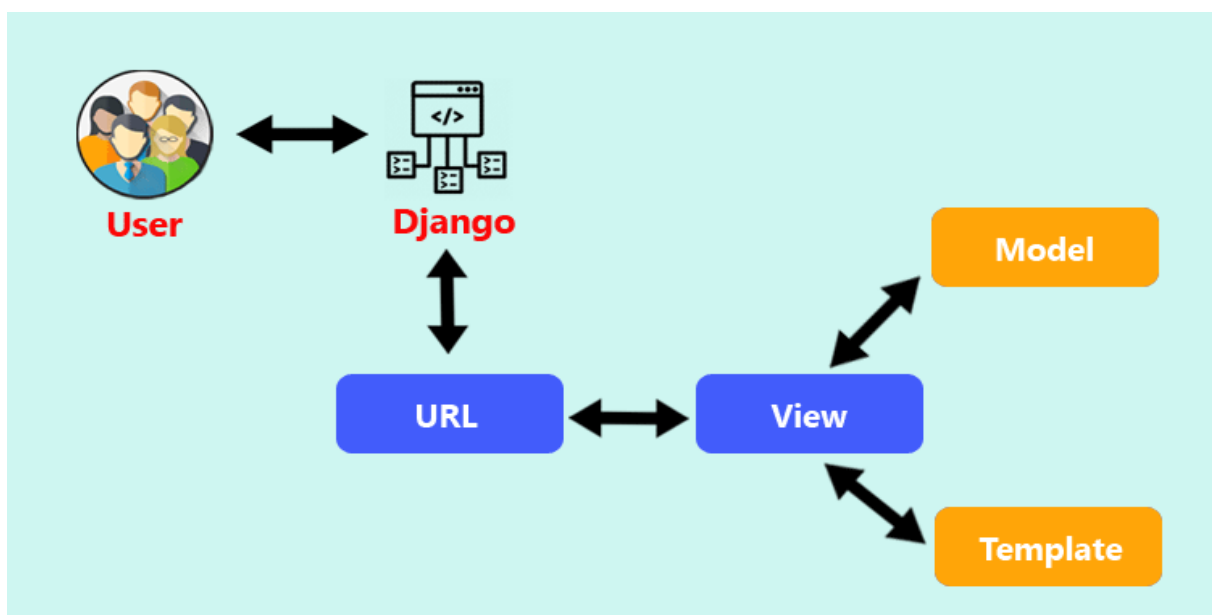
- **Model (Modèle) :**

- **Fonction principale :** Le modèle gère la structure des données et la logique de manipulation des informations. Dans Django, les modèles définissent la forme des données sous forme de classes Python, qui sont directement liées à des tables dans une base de données relationnelle.
- **Structure et relations :** Les modèles permettent de définir les tables, les champs, et les relations entre ces tables. Chaque classe de modèle correspond à une table, et chaque instance de la classe représente une ligne dans cette table.

- **View (Vue) :**

- **Fonction principale** : La vue est responsable de la logique métier de l'application. Elle traite les requêtes HTTP, interagit avec les modèles pour récupérer ou modifier les données, et renvoie une réponse au client.
- **Interaction avec les modèles et templates** : Les vues récupèrent les informations à partir des modèles et les transmettent aux templates pour le rendu. Elles peuvent également gérer des actions spécifiques, telles que la création, la mise à jour, ou la suppression d'entrées dans la base de données.
- **Template (Gabarit)** :
  - **Fonction principale** : Le template est chargé de la présentation visuelle des données. Il s'agit de fichiers HTML avec des balises Django qui permettent d'afficher dynamiquement les informations provenant des vues.
  - **Séparation de la logique et de la présentation** : Les templates offrent un moyen de séparer la logique métier (gérée par les vues) de la présentation (gérée par les templates), ce qui rend le code plus modulaire et maintenable.

Django utilise cette architecture MVT pour assurer une séparation des préoccupations, permettant ainsi une gestion plus efficace du code. Ce découpage en trois couches distinctes facilite également la collaboration entre les développeurs et les designers, tout en rendant l'application plus facile à maintenir et à évoluer.



**Figure 16: Structure de l'architecture Modèle-Vue-Template [6]**

### **3. Politique de sécurité**

Une politique de sécurité correspond à l'ensemble des stratégies et mesures mises en place pour minimiser les failles d'un système informatique face aux risques, qu'ils soient accidentels ou malveillants. La protection de notre application est une priorité essentielle. Elle s'appuie sur des mesures de sécurité rigoureuses dès les premières étapes du développement, en passant par l'intégration de protections spécifiques lors de la mise en production, ainsi la sensibilisation des utilisateurs.

#### **a. Mesures de sécurité prises au cours du développement**

Le mécanisme d'authentification, illustré dans la figure 8 sous forme de diagramme de séquence pour le cas d'utilisation « S'authentifier », constitue l'une des principales mesures de sécurité de notre application. L'accès à l'application a été protégé par un nom d'utilisateur et un mot de passe assurant une meilleure confidentialité pour garantir que seuls les utilisateurs autorisés puissent accéder aux données de la plateforme. En outre, un utilisateur qui se connecte ne peut exécuter que les fonctionnalités du système que lui confère son profil.

#### **b. Mesures de sécurité prises au cours la mise en production**

Dans le cadre de la mise en production de l'application, plusieurs mesures de sécurité ont été mises en place pour garantir la protection des données et des utilisateurs. Tout d'abord, le mode DEBUG a été désactivé afin d'éviter la divulgation d'informations sensibles en cas d'erreur, et seuls les domaines spécifiés ont été autorisés à accéder à l'application via la configuration d'ALLOWED\_HOSTS. Ensuite, la sécurisation des communications a été renforcée avec la redirection automatique du trafic HTTP vers HTTPS, l'activation du HSTS (HTTP Strict Transport Security) pour forcer l'utilisation de HTTPS (HyperText Transfer Protocol Secure). Les cookies de session et CSRF (Cross-Site Request Forgery) ont également été sécurisés pour garantir qu'ils ne soient transmis que via des connexions chiffrées. De plus, des protections contre les attaques courantes ont été intégrées, notamment la prévention des attaques de clickjacking avec X\_FRAME\_OPTIONS, le blocage des attaques XSS (Cross-Site Scripting) avec un filtre activé dans les navigateurs, et la prévention du type de contenu interprété à tort avec SECURE\_CONTENT\_TYPE\_NOSNIFF. Pour les fichiers statiques, White Noise a été utilisé pour améliorer leur gestion et sécurité. Enfin, un système de journalisation a été mis en place pour suivre les erreurs et anomalies sans exposer d'informations sensibles à l'interface utilisateur, avec un fichier de log dédié à cet effet.

### **c. La sensibilisation des utilisateurs**

Dans notre stratégie de sécurité, il est crucial de souligner l'importance du rôle humain. Ainsi, il est indispensable de fournir à nos utilisateurs des conseils concrets pour optimiser l'utilisation de notre logiciel. Par exemple, en évitant de partager leurs mots de passe, de laisser leur ordinateur sans surveillance et en veillant à la mise à jour régulière de leurs antivirus, ils peuvent significativement réduire les risques d'incidents. Il est également recommandé de restreindre l'accès à des sites web non fiables ou dépourvus de certificats SSL. En éduquant les utilisateurs aux bonnes pratiques de sécurité, nous renforçons la protection globale de notre plateforme.

## **4. Tests logiciels**

Le test logiciel est une procédure systématique et contrôlée visant à évaluer et à vérifier les fonctionnalités, la performance et la fiabilité d'une application logicielle, garantissant ainsi que celle-ci répond de manière précise et conforme aux spécifications fonctionnelles établies. Dans le cadre de notre projet, nous avons réalisé des tests unitaires et des tests fonctionnels pour les fonctionnalités de notre application, que nous détaillons ci-dessous.

### **a. Test unitaire**

Un test unitaire est conçu pour évaluer le bon fonctionnement d'une portion précise du code, généralement une fonction ou une méthode. Son objectif est de garantir que chaque unité de code opère correctement de manière indépendante. Dans notre contexte, nous procéderons à des tests de modèles ainsi qu'à des tests de vues.

#### **✚ Environnement de test**

Pour notre étude, nous avons utilisé le module `django.test`, qui inclut des classes comme `TestCase` pour configurer un environnement de test adéquat. Ce module permet de générer des objets de modèles fictifs et de gérer les transactions en base de données, facilitant ainsi l'examen de certaines fonctionnalités isolées de la logique métier.

#### **✚ Test des modèles**

Le test des modèles est essentiel pour valider la création d'objets dans la base de données et pour garantir que les contraintes, comme les champs obligatoires, soient respectées. Dans le cadre de notre projet, nous avons effectué des tests sur les classes fournisseurs et plan. La Figure

17 montre les résultats des tests effectués sur ces classes, présentées comme exemples.

```
• (env) PS C:\Users\Danaye David\Desktop\new\sonabel-main - Copie> python manage.py test suivi
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 0.004s

OK
Destroying test database for alias 'default'...
○ (env) PS C:\Users\Danaye David\Desktop\new\sonabel-main - Copie> █
```

**Figure 17: Résultat du test des modèles de notre application**

Le message "OK" indique que les tests ont été exécutés avec succès sans aucune erreur, ce qui témoigne de la fiabilité de notre code.

### ✚ Test des vues

Le test des différentes vues du projet permet la vérification du bon rendu des pages et vérifie l'accès aux pages en fonction des autorisations comme dans la figure 18 où nous montrons le résultat du test de la vue des tests des vues sur les vues home pour la page d'accueil, itemsliste pour lister lignes des plans de passation et addplanitems pour ajouter une ligne au plan.

```
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 4.845s

OK
Destroying test database for alias 'default'...
○ (env) PS C:\Users\Danaye David\Desktop\new\sonabel-main - Copie> █
```

**Figure 18: Résultat du test de la vue**

Le message "OK" indique que les tests ont été exécutés avec succès sans aucune erreur, ce qui témoigne de la fiabilité de notre code.

### b. Test fonctionnel

Les tests fonctionnels visent à valider les fonctionnalités en simulant des scénarios métiers basés sur les besoins fonctionnels. Ils constituent une approche habituelle pour s'assurer que les spécifications du logiciel fonctionnent correctement.



Fonctionnalités	Scénarios-Tests	Résultats
<b>S'authentifier</b>	L'utilisateur entre ses identifiants sur le formulaire de connexion et se trouve redirigé sur l'interface qui lui revient.	Fonctionnel
<b>Importer un plan</b>	L'utilisateur sélectionne l'année pour laquelle il désire importer un plan puis il clique sur choisir un fichier pour sélectionner le plan sur son disque, une fois le fichier sélectionné il appuie sur le bouton importer un plan et reçoit une notification de succès.	Non complet
<b>Filtrer les plans par année</b>	L'utilisateur sélectionne l'année pour laquelle il souhaite afficher le plan puis il clique sur le bouton pour filtrer et le plan en question s'affiche	Fonctionnel
<b>Afficher les lignes du plan</b>	L'utilisateur sélectionne le nombre de lignes qu'il veut afficher par exemple 10 ou 20 et ces lignes du plan s'affiche	Fonctionnel
<b>Ajouter une nouvelle ligne au plan</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier une ligne du plan</b>	L'utilisateur clique sur le bouton d'édition pour une ligne du plan et le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer une ligne du plan</b>	L'utilisateur clique sur le bouton de suppression pour une ligne du plan et le formulaire correspondant s'affiche, il confirme	Fonctionnel

Fonctionnalités	Scénarios-Tests	Résultats
	et reçoit une notification de succès de l'opération	
<b>Lister les dossiers</b>	L'utilisateur clique sur le bouton pour lister les dossiers et la liste s'affiche	Fonctionnel
<b>Ajouter un dossier</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier un dossier</b>	L'utilisateur clique sur le bouton d'édition, le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer un dossier</b>	L'utilisateur clique sur le bouton de suppression le formulaire correspondant s'affiche, il confirme et reçoit une notification de succès de l'opération	Fonctionnel
<b>Afficher les lots</b>	L'utilisateur clique sur le bouton pour voir les lots et la liste s'affiche	Fonctionnel
<b>Ajouter un lot</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier un lot</b>	L'utilisateur clique sur le bouton d'édition, le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer un lot</b>	L'utilisateur clique sur le bouton de suppression le formulaire correspondant s'affiche, il confirme et reçoit une notification de succès de l'opération	Fonctionnel

Fonctionnalités	Scénarios-Tests	Résultats
<b>Lister avis de publication</b>	L'utilisateur clique sur le bouton pour lister les avis de publication et la liste s'affiche	Fonctionnel
<b>Ajouter un avis de publication</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier un avis de publication</b>	L'utilisateur clique sur le bouton d'édition, le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer un avis de publication</b>	L'utilisateur clique sur le bouton de suppression le formulaire correspondant s'affiche, il confirme et reçoit une notification de succès de l'opération	Fonctionnel
<b>Lister les offres</b>	L'utilisateur clique sur le bouton pour lister les offres et la liste s'affiche	Fonctionnel
<b>Ajouter une offre</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier une offre</b>	L'utilisateur clique sur le bouton d'édition, le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer une offre</b>	L'utilisateur clique sur le bouton de suppression le formulaire correspondant s'affiche, il confirme et reçoit une notification de succès de l'opération	Fonctionnel
<b>Lister les résultats</b>	L'utilisateur clique sur le bouton pour lister les résultats et la liste s'affiche	Fonctionnel

Fonctionnalités	Scénarios-Tests	Résultats
<b>Ajouter un résultat</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier un résultat</b>	L'utilisateur clique sur le bouton d'édition, le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer le résultat</b>	L'utilisateur clique sur le bouton de suppression le formulaire correspondant s'affiche, il confirme et reçoit une notification de succès de l'opération	Fonctionnel
<b>Lister les marchés</b>	L'utilisateur clique sur le bouton pour lister les marchés et la liste s'affiche	Fonctionnel
<b>Ajouter un marché</b>	L'utilisateur clique sur le bouton d'ajout, le formulaire correspondant s'affiche, il saisit les données puis il soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Modifier un marché</b>	L'utilisateur clique sur le bouton d'édition, le formulaire correspondant s'affiche, il modifie les données, soumet le formulaire et reçoit une notification de succès de l'opération	Fonctionnel
<b>Supprimer un marché</b>	L'utilisateur clique sur le bouton de suppression le formulaire correspondant s'affiche, il confirme et reçoit une notification de succès de l'opération	Fonctionnel
<b>Suivre l'évolution des dossiers</b>	L'utilisateur clique sur le bouton de suivi et la page correspondante avec l'ensemble des dossiers et leurs états	Fonctionnel

Fonctionnalités	Scénarios-Tests	Résultats
<b>Afficher l'état du PPM</b>	L'utilisateur clique sur le bouton d'affichage et le plan de passation s'affiche.	Fonctionnel
<b>Exporter le PPM</b>	L'utilisateur clique sur le bouton d'exportation et le plan de passation est exporté et télécharger sous format Excel	Fonctionnel

Au regard des résultat des tests des 34 fonctionnalités nous avons attribué un pourcentage équitable de 2,94 % à chaque fonctionnalité et vu que la fonctionnalité importer plan n'est pas totalement terminer nous estimons le taux de réalisation du logiciel en fonction des cas d'utilisation à taux de réalisation =  $2,94 \% \times (34 - 1) = 97,06\%$ .

Après la présentation des tests effectués sur notre projet, voyons à présent à combien est estimé le coût de réalisation de notre système de gestion des projets.

### 5. Estimation du cout de développement

En ce qui concerne la conception d'un logiciel, une phase essentielle est souvent l'estimation des coûts de réalisation. Cette opération permet d'identifier les ressources humaines, financières et matérielles requises pour le projet. Parmi les méthodes d'estimation, on trouve la méthode des points de fonction, l'estimation par analogie, les réseaux de neurones, les arbres de régression, le jugement d'experts et la méthode COCOMO.

Méthodes	Types	Avantages	Inconvénients
<b>Point de fonction</b>	Paramétrique ou algorithmique	<ul style="list-style-type: none"> <li>- Ne dépend pas du métier mais de la sémantique générique ;</li> <li>- Métrique objective car ne prend pas en compte les internes informatiques dont l'utilisateur final n'a ni connaissance ni utilité.</li> </ul>	Élaborée pour des systèmes d'information de gestion pure.

<b>COCOMO</b>	Paramétrique ou algorithmique	<ul style="list-style-type: none"> <li>- Approche permet de réduire des erreurs de l'estimation finale de l'effort ;</li> <li>- Ses estimations sont plus fiables si les paramètres du projet sont bien connus.</li> </ul>	<ul style="list-style-type: none"> <li>- Elle ne concerne que le développement informatique ;</li> <li>- Il existe une difficulté de calculer de façon fiable la taille des projets à un stade peu avancé.</li> </ul>
<b>Estimation par analogie</b>	Non paramétriques (non algorithmique)	<ul style="list-style-type: none"> <li>- Rapidité d'estimation du cout du projet du à l'expérience de la personne en charge de l'estimation du cout ;</li> <li>- Les différences entre les projets achevés et le projet proposé peuvent être identifiés et les impacts estimés.</li> </ul>	<ul style="list-style-type: none"> <li>- Ne traite pas convenablement le cas des projets logiciels décrits par des valeurs linguistiques telles que « bas », « élevé », et « complexe » ;</li> <li>- Elle est imprécise et non déterministe ;</li> </ul>
<b>Réseaux de neurones</b>	Non paramétriques (non algorithmique)	<ul style="list-style-type: none"> <li>- Permettent l'apprentissage automatique à partir des situations et des résultats précédents ;</li> <li>- Permettent un traitement parallèle de l'information.</li> </ul>	<ul style="list-style-type: none"> <li>- Difficultés d'explication et d'interpréter du processus ;</li> <li>- Problèmes de classification et de catégorisation.</li> </ul>
<b>Arbre de régression</b>	Non paramétriques (non algorithmique)	<ul style="list-style-type: none"> <li>- Comportement facilement compréhensible ;</li> <li>- Permet une modélisation des relations complexes entre les différents facteurs affectant le coût.</li> </ul>	<ul style="list-style-type: none"> <li>- Ils ne sont pas faciles à développer ;</li> <li>- Il implique des calculs très intensifs pour générer l'estimation.</li> </ul>

Nous utilisons la méthode COCOMO SIMPLIFIÉE (Constructive Cost Model) pour la fiabilité de ses estimations et sa popularité. Les estimations de COCOMO sont basées sur la complexité des applications classées en plusieurs niveaux. Le niveau « S » qui concerne les applications simples et déterministes, ayant peu de cas particuliers. Le niveau « P » qui regroupe les applications plus complexes que les applications de niveau « S » ils restent tout de même déterministes. Enfin, les applications de niveau « E » qui sont très complexes et non déterministes. Pour plus de détail sur cette méthode voir (ANNEXE 2).

Nous estimons à 7000 le nombre moyen de lignes de code de notre application qui est de complexité S. En supposant que le salaire d'un Ingénieur-informaticien soit égal à 250 000 F CFA, on peut alors estimer le coût de développement comme suit :

- Effort :

$$HM = 2,4 * KLS^{1,05} = 2,4 * (7000/1000)^{1,05} = 18,51 \text{ Homme/Mois}$$

- Temps de développement :

$$TDEV = 2,5 * Effort^{0,38} = 2,5 * 18,51^{0,38} = 7,57 \text{ mois}$$

- Coût de développement :

$$CDEV = 15,75 * X \text{ FCFA} = 18,51 * 250\,000 \text{ FCFA} = 4\,627\,500 \text{ FCFA}$$

Même si la méthode COCOMO évalue le coût d'un projet en fonction des lignes de code, et que l'utilisation d'un framework génère des lignes supplémentaires, nous avons choisi d'utiliser COCOMO SIMPLIFIÉE pour sa simplicité d'implémentation et son adaptabilité à différents types de projets. En outre, elle est largement reconnue pour offrir une estimation rapide et cohérente, ce qui permet de tenir des délais serrés tout en garantissant des résultats proches de la réalité.

En prenant en considération le coût du développement ainsi que celui du besoin en matériel informatique et logiciel, on obtient le coût total du projet tel que présenté dans le Tableau 16.

**Tableau 16: Cout total du projet**

Désignation	Quantité	Coût unitaire (En FCFA)	Disponibilité ou coût total (en FCFA)
-------------	----------	----------------------------	--

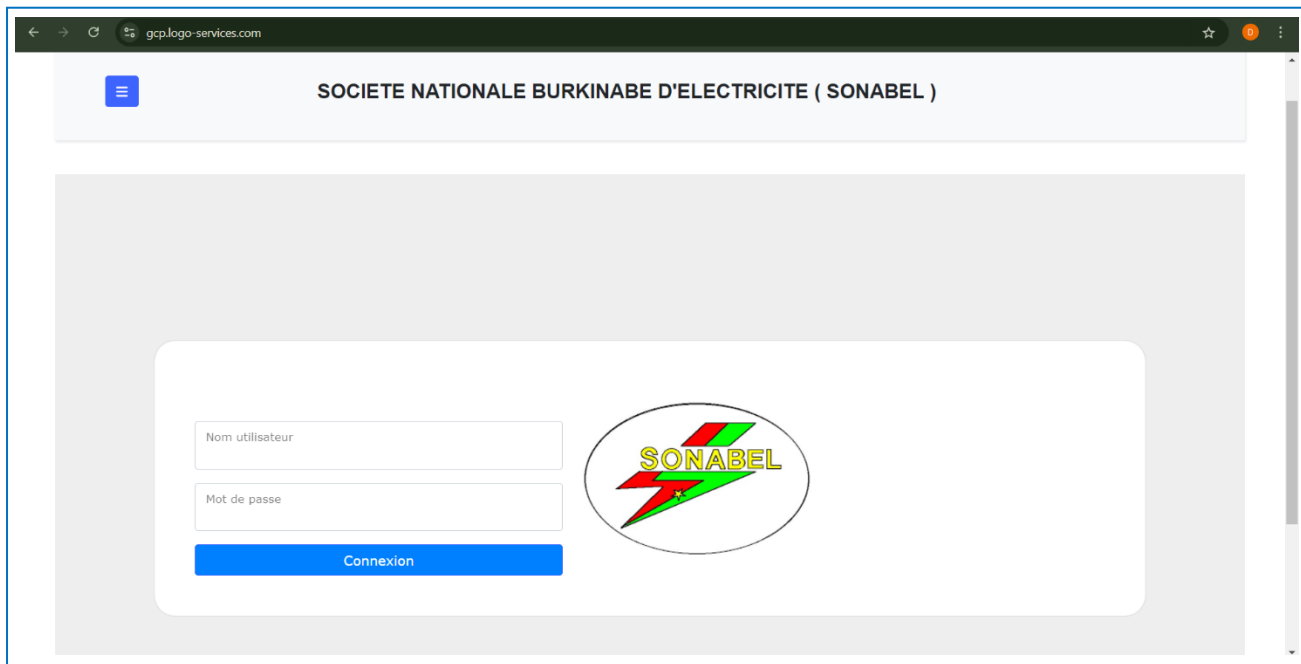
<b>Ordinateur portable</b>	1	400 000	400 000
<b>GanttProject</b>	1	-	Licence gratuite
<b>Postgre SQL</b>	1	-	Licence gratuite
<b>Visual Studio Code</b>	1	-	Licence gratuite
<b>Développement</b>	-	-	4 627 500
<b>Total.....</b>			<b>5 027 500</b>

## 6. Quelques maquettes IHM de notre application

La figure suivante représente la partie connexion de notre système. En effet, seul l'administrateur est autorisé à enregistrer un utilisateur. Tous les utilisateurs ayant un compte, peuvent se connecter à partir de ce formulaire de connexion.

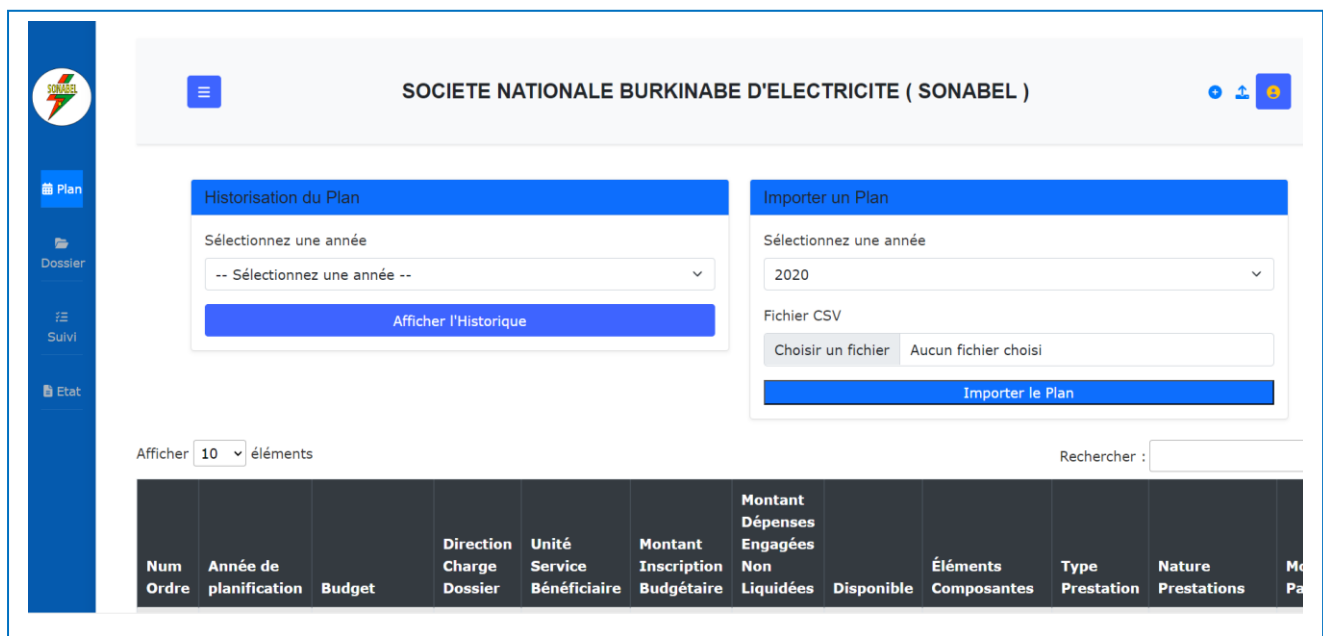


L'interface ci-dessous à figure19 nous permet de nous connecter à notre application afin de pouvoir accéder aux différentes pages du système.



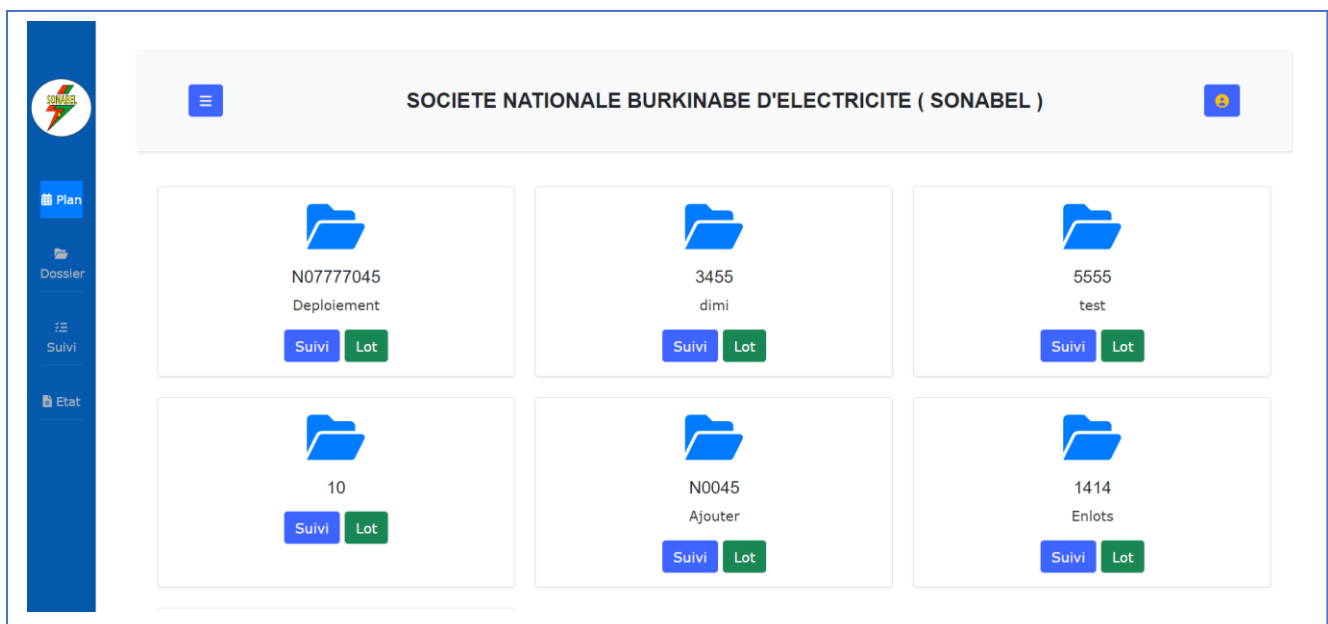
**Figure 19: Ecran de connexion**

L'interface ci-dessous à la figure 20 nous permet d'importer et de filtré les plans par année.



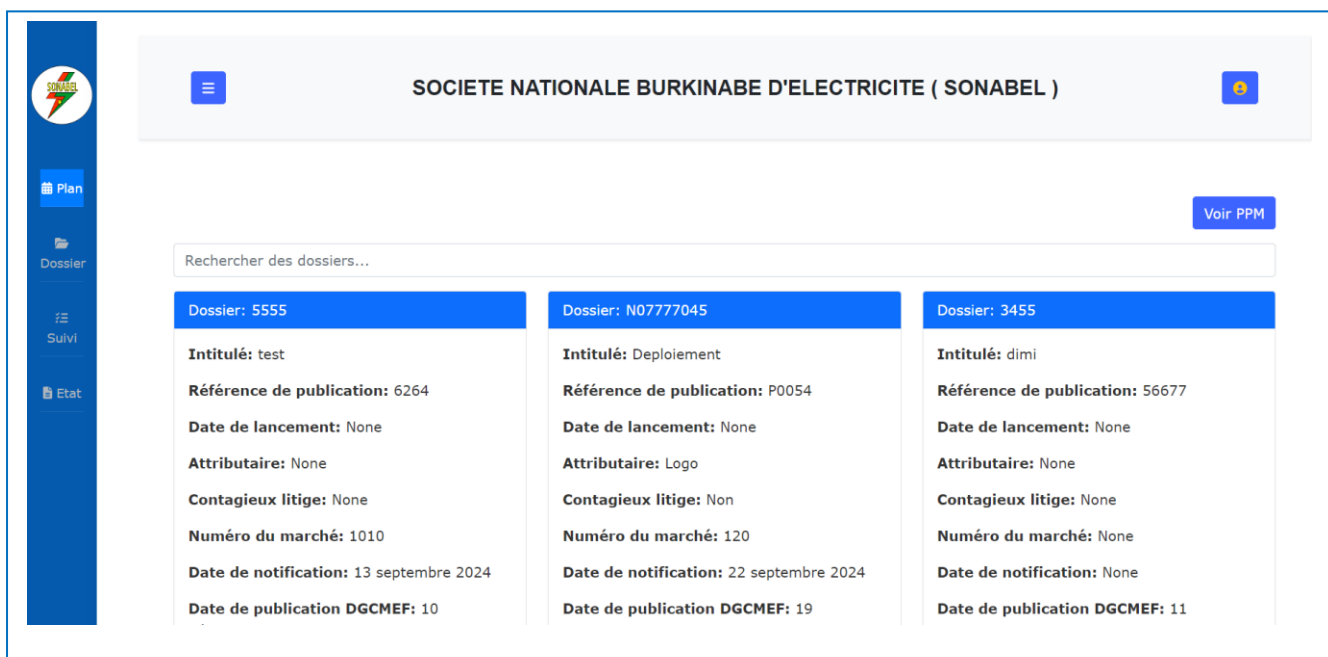
**Figure 20: Ecran d'import et de filtration**

L'interface ci-dessous à figure 21 permet d'afficher chaque pour effectuer le suivi par dossier.



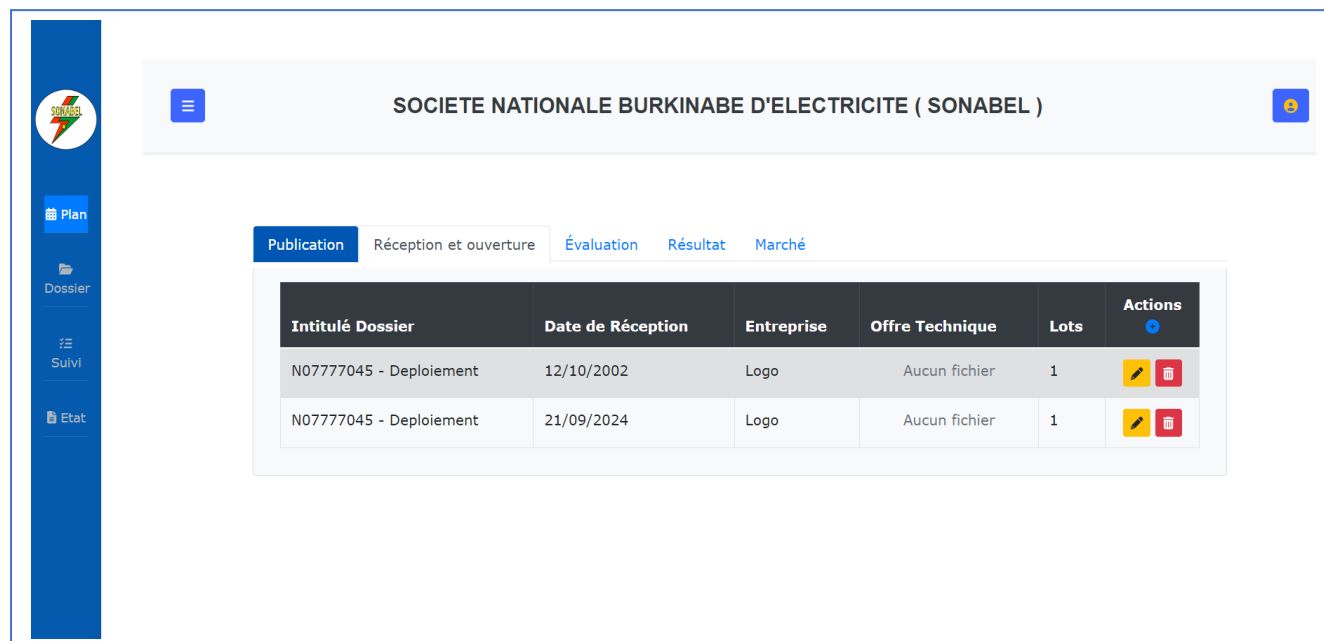
**Figure 21: Ecran d'affichage des dossiers pour leurs suivis**

L'interface ci-dessous, figure 22 permet de suivre l'évolution des dossiers.







**Figure 22:Ecran d'évolution des dossiers**

L'interface ci-dessous, figure 23 permet de réceptionner les offres techniques.



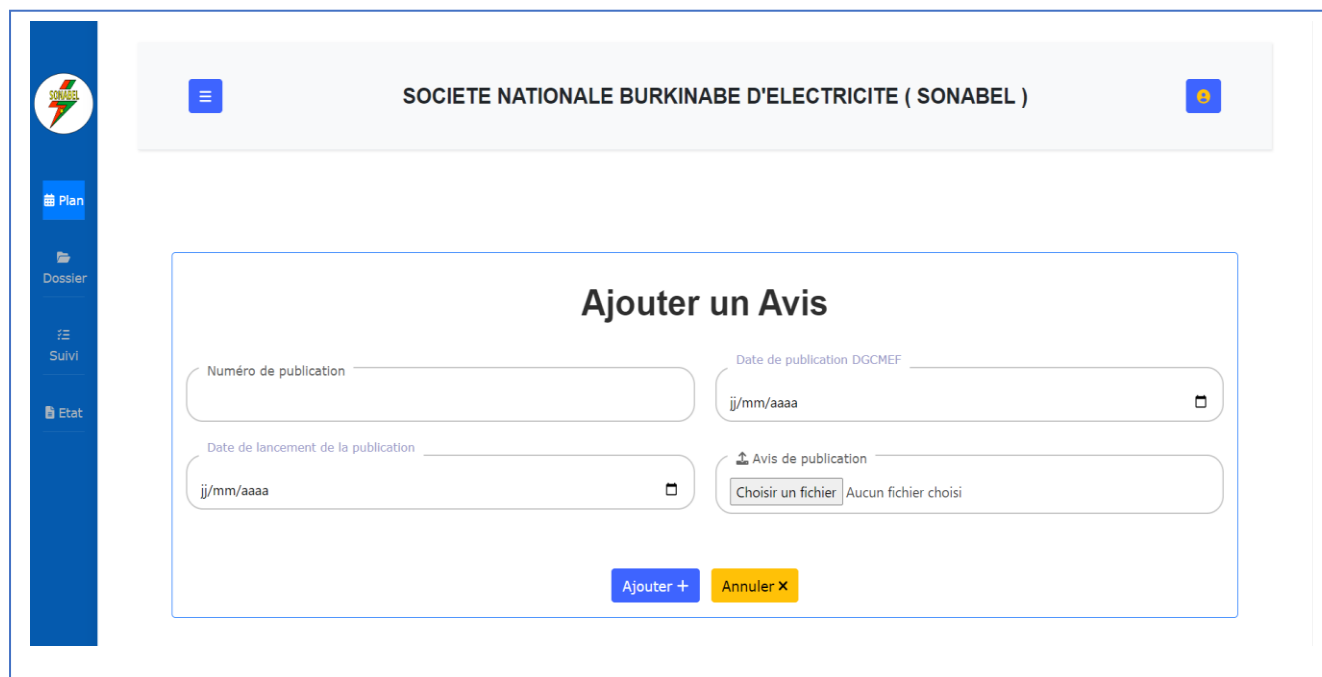
**SOCIETE NATIONALE BURKINABE D'ELECTRICITE ( SONABEL )**

Publication Réception et ouverture Évaluation Résultat Marché

Intitulé Dossier	Date de Réception	Entreprise	Offre Technique	Lots	Actions
N07777045 - Deploiement	12/10/2002	Logo	Aucun fichier	1	 
N07777045 - Deploiement	21/09/2024	Logo	Aucun fichier	1	 

**Figure 23: Ecran de réception des offres**

L'interface ci-dessous, figure 25 permet d'ajouter un avis de publication.



**SOCIETE NATIONALE BURKINABE D'ELECTRICITE ( SONABEL )**

### Ajouter un Avis

Numéro de publication

Date de publication DGCMEF

Date de lancement de la publication

Avis de publication  Aucun fichier choisi

**Figure 24: Ecran d'ajout d'un avis de publication**

---

## CHAPITRE 3

---

### BILAN DU STAGE

---

Dans ce chapitre final, nous aborderons le bilan de notre stage. Pour ce faire, nous présenterons d'une part le déroulement du stage ainsi que les activités menées au cours de cette période. D'autre part, nous ferons part de nos observations et suggestions.

#### **I. Présentation du déroulement du stage et des activités réalisées**

##### **1. Activités menées**

Dans le cadre du projet de création et développement de l'application web pour le suivi des marchés publics à la SONABEL, nous avons participé à plusieurs tâches essentielles qui ont contribué à l'atteinte des objectifs. Ce stage, d'une durée de trois mois, m'a permis de mettre en pratique les connaissances acquises en classe tout en découvrant de nouveaux aspects du développement web. Les activités spécifiques auxquelles j'ai participé incluent :

- Initiation aux technologies de pointe telles que Django pour le développement web et PostgreSQL pour la gestion des bases de données, renforçant ainsi mes compétences techniques ;
- Participation à des réunions d'échange via Google Meet et en présentiel, ce qui m'a aidé à mieux comprendre les exigences du projet et à collaborer efficacement avec l'équipe de développement ;
- Utilisation d'outils de versioning comme GitHub pour gérer les versions du code, suivre les modifications et travailler en collaboration avec les autres développeurs ;
- Déploiement de l'application, ce qui m'a permis de comprendre les étapes cruciales de la mise en production d'un projet informatique.

##### **2. Connaissances acquises**

Ce stage chez Logo Services a été extrêmement enrichissant et m'a permis de renforcer et d'appliquer les connaissances théoriques acquises au cours de ma formation. Les apprentissages principaux incluent :

- Application des compétences en développement web, particulièrement avec Django, dans un environnement réel et professionnel ;
- Amélioration des compétences en gestion de bases de données, en utilisant PostgreSQL et sqlite3 pour gérer les données critiques du projet ;

- Renforcement des compétences en collaboration et en gestion de versions, grâce à l'utilisation de GitHub, un outil essentiel pour tout développeur ;
- Compréhension des processus de déploiement, un aspect essentiel pour assurer la mise en production réussie des projets.

## **II. Observations et suggestions**

L'expérience acquise durant ce stage a été très positive. Logo Services m'a offert un cadre de travail structuré et propice à l'apprentissage, où j'ai pu mettre en pratique mes connaissances tout en découvrant les réalités du monde professionnel.

Cependant, je pense que certaines améliorations pourraient être envisagées pour rendre l'expérience encore plus bénéfique pour les stagiaires futurs :

- En tenant compte de l'ampleur des projets au sein de LOGO SERVICES, nous suggérons d'augmenter l'effectif des développeurs afin d'assurer plus rapidement la réussite de ces projets.

## CONCLUSION GÉNÉRALE

Ce stage chez Logo Services a marqué une étape clé dans mon parcours en informatique. Il m'a permis de mettre en pratique les connaissances théoriques acquises durant ma licence tout en approfondissant mes compétences techniques grâce à l'utilisation de technologies modernes comme Django et PostgreSQL.

Dans le cadre de ce projet, nous avons adopté la méthode 2TUP et utilisé le langage de modélisation UML pour développer des diagrammes de cas d'utilisation, de séquence, d'activité, de déploiement et de classes, avant de passer à l'implémentation des fonctionnalités identifiées.

Cette expérience m'a également offert l'opportunité de me familiariser avec le fonctionnement réel d'une entreprise, en particulier en ce qui concerne la collaboration et la gestion de projet.

En perspective nous comptons finaliser le logiciel, élaborer un manuel d'utilisation et former les utilisateurs.

## BIBLIOGRAPHIE ET WEBOGRAPHIE

### 1. Bibliographie

- ✚ **KOUDA Ézéchiass Wendtoen (2022)**, « Conception Et Réalisation D'un Outil De Gestion Des Employés De Nafann Dans Un Contexte De Gestion De Projet » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 86p.
- ✚ **GUIRE Ismaël Faïssal (2023)**, « Automatisation des processus d'approbation dans l'administration des systèmes informatiques au sein de TICANALYSE » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 95p.
- ✚ **KAMBOU Yéri Sheila Mariane (2023)**, « Mise en place d'un système de gestion de missions : cas de A2SYS CONSULTING » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 87p.
- ✚ **TIAMA Bernard (2023)**, « Analyse et conception d'une plateforme de gestion de portefeuille clients destinée aux structures d'assurance et de mutuelle » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 80p.
- ✚ **SAWADOGO Rahimata Rimalguedo (2023)**, « Conception et réalisation d'une plateforme de gestion intégrée et de suivis startups incubées a SIRALABS » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 69p.
- ✚ **KONATE Salim Adam (2023)**, « Conception et réalisation d'un outil de gestion des demandes de terrains et d'autorisations d'aménagement pour le ministère de l'urbanisme, des affaires foncières et de l'habitat »

## 2. Webographie

- ✚ [1] « Cycle en Y, Processus de développement 2TUP ». Consulté le 04 Octobre 2024.  
<https://slideplayer.fr/slide/482289/1/images/22/Conception+pr%C3%A9liminaire.jpg/>.
- ✚ [2] « Présentation du langage de modélisation unifié (UML) ». *Cybermédiane* (blog), Consulté le 06 Octobre 2024. <https://www.cybermedian.com/fr/unified-modeling-language-umlintroduction/>.
- ✚ [3] « Architecture 1 tier » Consulté le 06 Octobre 2024.  
<https://www.researchgate.net/profile/RifaatAbdalla/publication/305631562/figure/fig3/0/AS:725762063757318@1550046568438/One-tier-architecture.png>.
- ✚ [4] « Architecture 2 tiers » Consulté le 06 Octobre 2024.  
<https://images.app.goo.gl/JctssrzMR6EVX4s38>.
- ✚ [5] « Architecture 3 tiers » Consulté le 06 Octobre 2024.  
<https://images.app.goo.gl/V7EJ1jG5Lze3wv2w8>.
- ✚ [6] « Architecture MVT de Django » Consulté le 06 Octobre 2024.  
<https://blog.stepskochi.com/python-django-mvt-pattern/>
- ✚ « Stack Overflow - où les développeurs apprennent, partagent et construisent des carrières ». Consulté plusieurs fois lors du développement et du déploiement. <https://stackoverflow.com/>.
- ✚ « La documentation de Django » consulté plusieurs fois lors du développement de la plateforme »  
<https://docs.djangoproject.com/en/5.0/>
- ✚ Open AI GPT-4 « Model d'IA développé par Open AI » Consulté plusieurs fois pendant le développement.  
<https://chat.openai.com/>
- ✚ « La documentation de Select2 » consultée à plusieurs reprises pour améliorer l'interactivité des champs de sélection dans l'application.  
Lien vers la documentation : <https://select2.org/>



## TABLE DES MATIERES

<b>SOMMAIRE.....</b>	<b>i</b>
<b>DEDICACES .....</b>	<b>ii</b>
<b>REMERCIEMENTS.....</b>	<b>iii</b>
<b>LISTE DES SIGLES ET ABREVIATIONS.....</b>	<b>iv</b>
<b>LISTE DES FIGURES ET GRAPHIQUES.....</b>	<b>vi</b>
<b>LISTE DES TABLEAUX .....</b>	<b>vii</b>
<b>RESUME.....</b>	<b>viii</b>
<b>INTRODUCTION GENERALE.....</b>	<b>1</b>
<b>CHAPITRE 1 : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL... 2</b>	
<b>I. Présentation de la structure de formation (IBAM) .....</b>	<b>2</b>
1. Historique .....	2
2. Objectif .....	2
3. Organisation.....	2
a. Organe statutaire .....	3
b. Organe d'exécution .....	4
4. Filières de formation.....	4
<b>II. Présentation de la structure d'accueil (LOGO SERVICES).....</b>	<b>5</b>
1. Histoire .....	6
2. Services.....	6
3. Organisation générale de LOGO SERVICES .....	6
<b>CHAPITRE 2: ANALYSE ET CONCEPTION .....</b>	<b>7</b>
<b>I. Etude préalable .....</b>	<b>7</b>
1. Présentation du thème.....	7
a. Préliminaire .....	7
b. Problématique .....	8
c. Objectif .....	8
d. Résultats attendus.....	8
2. Méthodes d'analyse et de conception .....	9
a. Processus de développement.....	9
b. Langage de modélisation .....	13

3. Présentation du groupe de travail .....	14
a. Groupe de pilotage .....	14
b. Groupe de projet .....	14
c. Groupe utilisateur .....	14
d. Planning de travail .....	15
<b>II. Expressions des besoins.....</b>	<b>15</b>
1. Description du fonctionnement attendu de l'application.....	15
2. Spécifications fonctionnelles.....	16
a. Identification des acteurs .....	16
b. Identifications des cas d'utilisations.....	16
c. Diagramme des cas d'utilisation .....	18
d. Description textuelle de certains cas d'utilisation .....	19
e. Diagramme de séquence.....	23
f. Diagramme d'activité .....	27
3. Spécifications techniques .....	28
a. Mise à disposition des conditions de travail .....	28
b. Architecture de développement.....	29
<b>III. Conception</b>	<b>globale</b>
.....	<b>30</b>
1. Diagramme de classe.....	30
a. Dictionnaire de données .....	30
b. Diagramme de classe.....	33
2. Diagramme de déploiement.....	36
<b>IV. Réalisation.....</b>	<b>36</b>
1. Présentation des outils de réalisations .....	37
a. Langages.....	37
b. Framework de développement .....	37
c. Outils de conception.....	38
d. Choix du système de gestion de base de données (SGBD) .....	38
e. Serveur d'application .....	41
2. Présentation de l'architecture MVT de notre application.....	42
3. Politique de sécurité .....	44
a. Mesures de sécurité prises au cours du développement .....	44
b. Mesures de sécurité prises au cours la mise en production .....	44
c. La sensibilisation des utilisateurs .....	45

4. Tests logiciels .....	45
a. Test unitaire.....	45
b. Test fonctionnel .....	46
5. Estimation du cout de développement.....	51
6. Quelques maquettes IHM de notre application .....	54
<b>CHAPITRE 3: BILAN DU STAGE .....</b>	<b>58</b>
<b>I. Présentation du déroulement du stage et des activités réalisées .....</b>	<b>58</b>
1. Activités menées.....	58
2. Connaissances acquises .....	58
<b>II. Observations et suggestions.....</b>	<b>59</b>
<b>CONCLUSION GÉNÉRALE.....</b>	<b>60</b>
<b>BIBLIOGRAPHIE ET WEBOGRAPHIE .....</b>	<b>61</b>
1. Bibliographie .....	61
2. Webographie.....	62
<b>TABLE DES MATIERES .....</b>	<b>63</b>
<b>I. ANNEXES 1 : Le langage UML.....</b>	<b>I</b>
<b>II. ANNEXE 2 : Méthode COCOMO.....</b>	<b>II</b>
<b>III. ANNEXE 3 : Dictionnaire de données.....</b>	<b>III</b>

## I. ANNEXES 1 : Le langage UML.

UML est l'acronyme anglais pour « Unified Modeling Language ». Il se traduit par « Langage de modélisation unifié ». C'est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée permettant de visualiser la conception d'un système. Ce langage est utilisé pour la spécification, la visualisation, la modification et la construction des documents nécessaires au bon développement d'un logiciel orienté objet. Il offre un standard de modélisation, pour représenter l'architecture logicielle. Grâce à UML, il est possible de générer tout ou une partie du code d'un logiciel à partir des divers documents réalisés. Ce langage de modélisation nous offre principalement 13 diagrammes (depuis sa deuxième version) pour modéliser un système. Ces diagrammes peuvent être utilisés selon la phase du développement d'un logiciel. En analyse, nous pouvons utiliser des :

- ✚ Diagrammes de cas d'utilisation : modélisent les besoins des utilisateurs ;
- ✚ Diagrammes de séquences vue de l'extérieur : présentent les scénarios entre les utilisateurs ;
- ✚ Diagrammes d'activités : c'est un enchaînement d'actions représentant un comportement du logiciel.

En phase de conception, le développeur peut utiliser des :

- ✚ Diagrammes de classes : pour représenter la structure interne du logiciel ;
- ✚ Diagrammes d'objets : pour présenter l'état interne du logiciel à un instant donné ;
- ✚ Diagrammes d'états-transitions : pour présenter l'évolution de l'état d'un objet ;
- ✚ Diagrammes de séquence vue de l'intérieur : pour montrer les scénarios d'interactions avec les utilisateurs au sein du logiciel ;
- ✚ Diagrammes de composants : pour présenter les composants physiques du logiciel ;
- ✚ Diagrammes de déploiement : pour l'organisation matérielle du logiciel.

Ces diagrammes sont rarement tous implémentés dans le cadre du même projet. Le choix des diagrammes à mettre en œuvre dans la modélisation est généralement fonction de la nature du projet et de sa taille.

## II. ANNEXE 2 : Méthode COCOMO

Un grand nombre de méthodes est mis à la disposition des développeurs pour estimer le coût de leurs plateformes. Notre choix se porte sur la méthode Constructive Cost Model (COCOMO) pour l'estimation du coût total du développement (CTDEV) de notre application, du fait de sa fiabilité. De plus, cette méthode permet également d'estimer le temps de développement (TDEV) du système correspondant au temps requis pour terminer le projet avec toutes les ressources disponibles. La méthode COCOMO se base principalement sur la complexité de l'application à développer qui correspond à l'un des trois (03) types suivants :

- ✦ S : ce sont des applications simples, n'ayant que peu de cas particuliers et de contraintes. Elles sont parfaitement déterministes ;
- ✦ P : ce sont des applications intermédiaires, plus complexes que les applications de type S. Elles restent tout de même déterministes bien que le nombre de leurs cas particuliers et de tests soit plus important que pour les applications de type S ;
- ✦ E : ce sont des applications très complexes, que ce soit au niveau de leurs contraintes, comme un système temps réel, où au niveau des données saisies, comme certaines interfaces graphiques où l'on ne peut envisager toutes les possibilités de saisies qu'un utilisateur pourrait effectuer. Elles ne sont pas déterministes.

**Tableau 17: Formule de calcul COCOMO**

Complexité	Effort (en Homme mois)	Temps de développement (TDEV en mois)
<b>S</b>	Effort = 2,4 (KLS) <sup>1,05</sup>	TDev = 2,5 (Effort) <sup>0,38</sup>
<b>P</b>	Effort = 3 (KLS) <sup>1,12</sup>	TDev = 2,5 (Effort) <sup>0,35</sup>
<b>E</b>	Effort = 3,6 (KLS) <sup>1,2</sup>	TDev = 2,5 (Effort) <sup>0,32</sup>

NB : HM est le nombre d'« homme mois » nécessaire à la réalisation du projet, et KLS est le nombre de Kilo Lignes Sources. Un homme mois correspond à 152 heures de travail effectif. Le nombre de personnes requis pour réaliser le projet dans cet intervalle de temps est donc :

$N = HM/TDEV$ . Etant donné que le salaire moyen d'un informaticien est de X FCFA, le coût total de développement pour ce projet est :  $CTDEV = HM * X$ .

### III. ANNEXE 3 : Dictionnaire de données

#### ‡ Classe : LignePlan

Cette classe permet de recueillir les informations des éléments du plan de passation depuis un fichier csv.

*Tableau 8: Description de la classe « LignePlan »*

Attribut	Description	Type de donnée
<b>num_ordre</b>	Le numéro d'ordre pour chaque ligne du plan	Chaine de caractères
<b>num_credit</b>	Le numéro de crédit alloué à cet élément du plan	Chaine de caractères
<b>budget</b>	Le budget global alloué pour cet élément du plan	Chaine de caractères
<b>direction_charge_dossier</b>	La direction ou département en charge de la gestion du dossier.	Chaine de caractères
<b>unite_service_beneficiaire</b>	L'unité ou service bénéficiaire du dossier	Chaine de caractères
<b>montant_inscription_budgetaire</b>	Le montant budgétisé alloué	Chaine de caractères
<b>montant_depenses_engagees_non_liquidees</b>	Montant des dépenses engagées mais non encore liquidées.	Chaine de caractères
<b>disponible</b>	Les fonds ou ressources encore disponibles	Chaine de caractères
<b>elements_composantes</b>	Les différents éléments ou composants liés au dossier	Chaine de caractères
<b>type_prestation</b>	Le type de prestation ou de service associé à l'élément.	Chaine de caractères

<b>nature_prestations</b>	La nature des prestations ou services fournies.	Chaine de caractères
<b>mode_passation</b>	Le mode de passation choisi pour l'appel d'offres ou le marché public.	Chaine de caractères
<b>agent_charge_dossier</b>	L'agent responsable de la gestion du dossier pour cet élément du plan.	Chaine de caractères
<b>date_prevue_reception_dossier_technique</b>	La date prévue pour la réception du dossier technique.	Chaine de caractères
<b>date_reelle_reception_dossier_technique</b>	La date réelle de réception du dossier technique.	Chaine de caractères
<b>service_charge_dossier_technique</b>	Le service responsable du dossier technique.	Chaine de caractères
<b>intitule_dossier</b>	L'intitulé ou le titre du dossier lié à cet élément du plan.	Chaine de caractères
<b>ref_dossier_appel_concurrence</b>	La référence du dossier d'appel à concurrence.	Chaine de caractères
<b>publication_dgcmef</b>	Détails de la publication par la DGCMEF	Chaine de caractères
<b>nom_attributaire_provisoire</b>	Nom de l'attributaire provisoire du marché.	Chaine de caractères
<b>contentieux_litiges</b>	Informations sur les contentieux ou litiges éventuels.	Chaine de caractères
<b>montant</b>	Le montant total du marché ou de l'offre.	Chaine de caractères
<b>numero_marche</b>	Le numéro du marché attribué.	Chaine de caractères
<b>date_notification_projet_contrat</b>	La date de notification du projet de contrat.	Chaine de caractères
<b>date_signature_contrat</b>	La date de signature du contrat final.	Chaine de caractères
<b>dossiers_marches_non_conclu_montant_alloue</b>	Montant alloué aux marchés non encore conclus.	Chaine de caractères
<b>taux_passation_marches</b>	Le taux de passation des marchés.	Chaine de caractères

<b>nombre_marches_conclure_par_ligne_ppm</b>	Nombre de marchés à conclure par ligne de PPM.	Chaine de caractères
<b>nombre_marches_conclus_par_ligne_ppm</b>	Nombre de marchés conclus par ligne de PPM.	Chaine de caractères
<b>nombre_ligne_fait_objet_marche</b>	Nombre de lignes faisant l'objet d'un marché.	Chaine de caractères
<b>date_remise_garantie_bonne_execution</b>	Date de remise de la garantie de bonne exécution.	Chaine de caractères
<b>date_transmission_contrat_enregistrement</b>	Date de transmission du contrat pour enregistrement.	Chaine de caractères
<b>date_retour_enregistrement_contrat</b>	Date de retour après l'enregistrement du contrat.	Chaine de caractères
<b>date_emission_ordre_service</b>	Date d'émission de l'ordre de service.	Chaine de caractères
<b>date_remise_site</b>	Date de remise du site.	Chaine de caractères
<b>date_demarrage_prevue_execution</b>	Date prévue pour le démarrage de l'exécution.	Chaine de caractères
<b>date_reelle_demarrage_execution</b>	Date réelle de démarrage de l'exécution.	Chaine de caractères
<b>delai_prevue_execution</b>	Délai prévu pour l'exécution.	Chaine de caractères
<b>delai_reelle_execution</b>	Délai réel de l'exécution.	Chaine de caractères
<b>date_butoir</b>	La date butoir pour l'exécution.	Chaine de caractères
<b>gestionnaire_credit</b>	Le gestionnaire du crédit associé à ce plan.	Chaine de caractères
<b>date_reelle_lancement</b>	La date réelle de lancement du projet ou du marché.	Chaine de caractères
<b>observations</b>	Observations diverses ou remarques.	Chaine de caractères

### ✚ Marché

Cette classe permet de recueillir les informations d'un plan.



**Tableau 18: description de la classe « Marché »**

Attribut	Description	Type de donnée
<b>id</b>	Identifiant du marché	Entier
<b>num_ref</b>	Le numéro de référence du marché	Date
<b>date_date_notif</b>	La date de notification du marché	Date
<b>date_retour_sign</b>	La date de retour de la signature du marché	Date

### ✚ Avis

Cette classe permet de recueillir les informations d'un avis de publication.

**Tableau 19: description de la classe « Avis »**

Attribut	Description	Type de donnée
<b>id</b>	Identifiant de l'avis	Entier
<b>num_publi</b>	Le numéro de publication	Chaine de caractères
<b>date_publi</b>	La date de publication de l'avis	Date
<b>fichier</b>	Le fichier correspondant a l'avis	Fichier
<b>date_lancement_pulication</b>	La date de lancement de la publication	Date

### ✚ Lots

Cette classe permet de recueillir les informations d'un lot.

**Tableau 20: description de la classe « Lot »**

Attribut	Description	Type de donnée
----------	-------------	----------------

<b>id</b>	Identifiant du plan	Entier
<b>num_lot</b>	Le numéro du lot	Entier
<b>intitule_lot</b>	L'intitulé du lot	Chaine de caractères
<b>montant_lot</b>	Le montant du lot	Entier

### ✚ Statut

Cette classe permet de recueillir les informations d'un statut.

**Tableau 21: description de la classe « Statut »**

Attribut	Description	Type de donnée
<b>id</b>	Identifiant du plan	Entier
<b>nom</b>	Le libellé du statut	Chaine de caractères

### ✚ Utilisateur

Cette classe permet de recueillir les informations d'un utilisateur.

**Tableau 22: description de la classe « Utilisateur »**

Attribut	Description	Type de donnée
<b>id</b>	Identifiant de l'utilisateur	Entier
<b>nom</b>	Le nom de l'utilisateur	Chaine de caractères
<b>mot_de_passe</b>	Le mot de passe de l'utilisateur	Mot de passe