

**UNIVERSITÉ JOSEPH KI-ZERBO
(UJKZ)**

**INSTITUT BURKINABE 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 : 08 Juillet au __ __ 2025

**THÈME :
ANALYSE ET CONCEPTION D'UN SYSTEME DIGITAL DE
SUIVI DES FACTURES ET DES COMMANDES DANS UNE
PME : CAS DE ESOFTE BURKINA SARL.**

Présenté par **Abdoul Razack OUEDRAOGO**

Maitre de stage :

M. Vivian BOGNINI

Directeur générale de ESOFTE BURKINA

Directeur de rapport :

Dr Yaya TRAORE

*Maître de conférences en
Informatique / IBAM-UJKZ*

Année académique 2024-2024

SOMMAIRE

SOMMAIRE	i
DEDICACE	ii
REMERCIEMENT	ii
LISTE DES SIGLES ET ABREVIATIONS	iii
LISTE DES FIGURES ET GRAPHIQUES	iii
LISTE DES TABLEAUX	iii
INTRODUCTION GENERALE	1
CHAPITRE I : PRESENTATION DES STRUCTURES DE FORMATIONS ET D'ACCUEIL	2
I. Présentation de la structure de formation	2
II. Présentation de la structure d'accueil	6
CHAPITRE II : ANALYSE ET CONCEPTION	6
I. Étude préalable	10
II. Expression des besoins	22
III. Conception globale	47
IV. Réalisation	52
CHAPITRE III : BILAN DU STAGE	52
I. DÉROULEMENT DU STAGE ET ACTIVITÉS MENÉES	70
II. CONCLUSION GÉNÉRALE	72
BIBLIOGRAPHIE ET WEBOGRAPHIE	I
Table des matières	I

DÉDICACE

A toute notre
famille

REMERCIEMENTS

LISTE DES SIGLES ET ABREVIATIONS

LISTE DES FIGURES ET GRAPHIQUES

LISTE DES TABLEAUX

Tableau 1 : Tableau comparatif des méthodologies ou processus de développement	13
Tableau 2: Comparaison de quelques langages de modélisation	18
Tableau 3 :Présentation de solutions logicielles de gestion de projets existantes.....	25
Tableau 4: Listes des cas d'utilisation	29
Tableau 5: Description textuel du cas d'utilisation : s'authentifier.....	35
Tableau 6 : Description textuel du cas d'utilisation : créer un projet	36
Tableau 7 : Description textuelle du cas Créer des tâches	37
Tableau 8 : Description textuelle du cas Assigner des tâches	38
Tableau 9: Description textuelle du cas Mettre à jour le statut d'une tâche	39
Tableau 10 : Tableau descriptif et comparatif de l'architecture 3-tiers	44
Tableau 11 : Dictionnaire de données de la classe User	47
Tableau 12 : Dictionnaire de données de la classe Project.....	47
Tableau 13: Dictionnaire de données de la classe Task	48
Tableau 14 : Tableau de comparaison de SGBD	54
Tableau 15: Tableau comparatif des servers d'application.....	57
Tableau 16 : Tableau d'estimation du coup de notre projet	68

INTRODUCTION GENERALE

Dans un monde où le numérique occupe une place de plus en plus importante, bien gérer les projets et les tâches en équipe est devenu essentiel, notamment dans le domaine des technologies.

Avec la multiplication des projets et la nécessité de mieux coordonner les équipes, les entreprises cherchent des solutions pour centraliser leurs activités et améliorer leur productivité.

C'est dans ce contexte que s'inscrit notre stage chez DIGITECH BURKINA, une entreprise qui développe des solutions numériques. En pleine croissance, elle a du mal à gérer plusieurs projets en même temps, ce qui complique le travail d'équipe et le suivi des tâches. C'est pourquoi on nous a confié le thème « Analyse et conception d'une plateforme intégrée de gestion de tâches et de projets collaboratifs : cas de DIGITECH BURKINA ».

Notre but est de créer une solution simple qui permet d'organiser le travail, de suivre les projets et de faciliter la communication entre les membres des équipes. Ce rapport présente notre démarche, de l'analyse des besoins à la conception de la solution.

Afin de rendre compte de manière claire et structurée du travail réalisé, ce rapport a été organisé en trois chapitres, chacun répondant à une étape clé de notre démarche. Ce découpage permet non seulement de mieux situer le cadre du stage, mais aussi de suivre de manière logique l'évolution du projet, de la réflexion initiale jusqu'au bilan final.

Le premier chapitre présente le cadre de formation et l'entreprise d'accueil, en l'occurrence l'Institut Burkinabè des Arts et Métiers (IBAM) et DIGITECH BURKINA. Le deuxième chapitre est consacré à l'analyse et à la conception du système, en mettant en lumière la problématique, les objectifs visés, la méthodologie suivie, les acteurs concernés ainsi que le planning du projet. Enfin, le troisième chapitre propose un bilan du stage, à travers une présentation des activités réalisées, des compétences acquises et des recommandations pour la suite, ainsi qu'une conclusion générale.

CHAPITRE I : PRESENTATION DES

STRUCTURES DE FORMATIONS ET D'ACCUEIL

Dans ce chapitre, nous présenterons successivement notre structure de formation à savoir l'IBAM ainsi que notre structure d'accueil DIGITECH BURKINA.

I. Présentation de la structure de formation

Dans cette section, nous offrirons une vue d'ensemble de l'**IBAM** en mettant en exergue son historique, ses objectifs, son organisation institutionnelle ainsi que ses différentes offres de formation.

1. Historique

Situé à Somgandé, l'**Institut Burkinabè des Arts et Métiers (IBAM)** est un établissement d'enseignement professionnel. Il a été créé en janvier 2000 à la suite de la refondation de l'Université de Ouagadougou, devenue « Université Ouaga I Pr Joseph KI-ZERBO » le 26 décembre 2015, puis renommée « Université Joseph KI-ZERBO » le 12 avril 2019. L'IBAM constitue la concrétisation de l'engagement de l'Université Joseph KI-ZERBO (UJKZ) en faveur de la professionnalisation des filières de formation. Cette orientation vise à renforcer l'efficacité externe de l'institution.

2. Objectif

L'**Institut Burkinabè des Arts et Métiers (IBAM)** a pour principal objectif de former des cadres moyens et supérieurs compétents, capables de répondre de manière efficiente aux besoins du marché de l'emploi. Cette orientation vise à favoriser une meilleure insertion professionnelle des diplômés, en cohérence avec les missions de l'Université Joseph KI-ZERBO.

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 placé sous la direction d'un

Directeur, assisté d'un Directeur adjoint, tous deux élus par le collège électoral de l'IBAM pour un mandat de deux (2) ans renouvelables.

La figure ci-après présente l'organisation détaillée de l'IBAM.

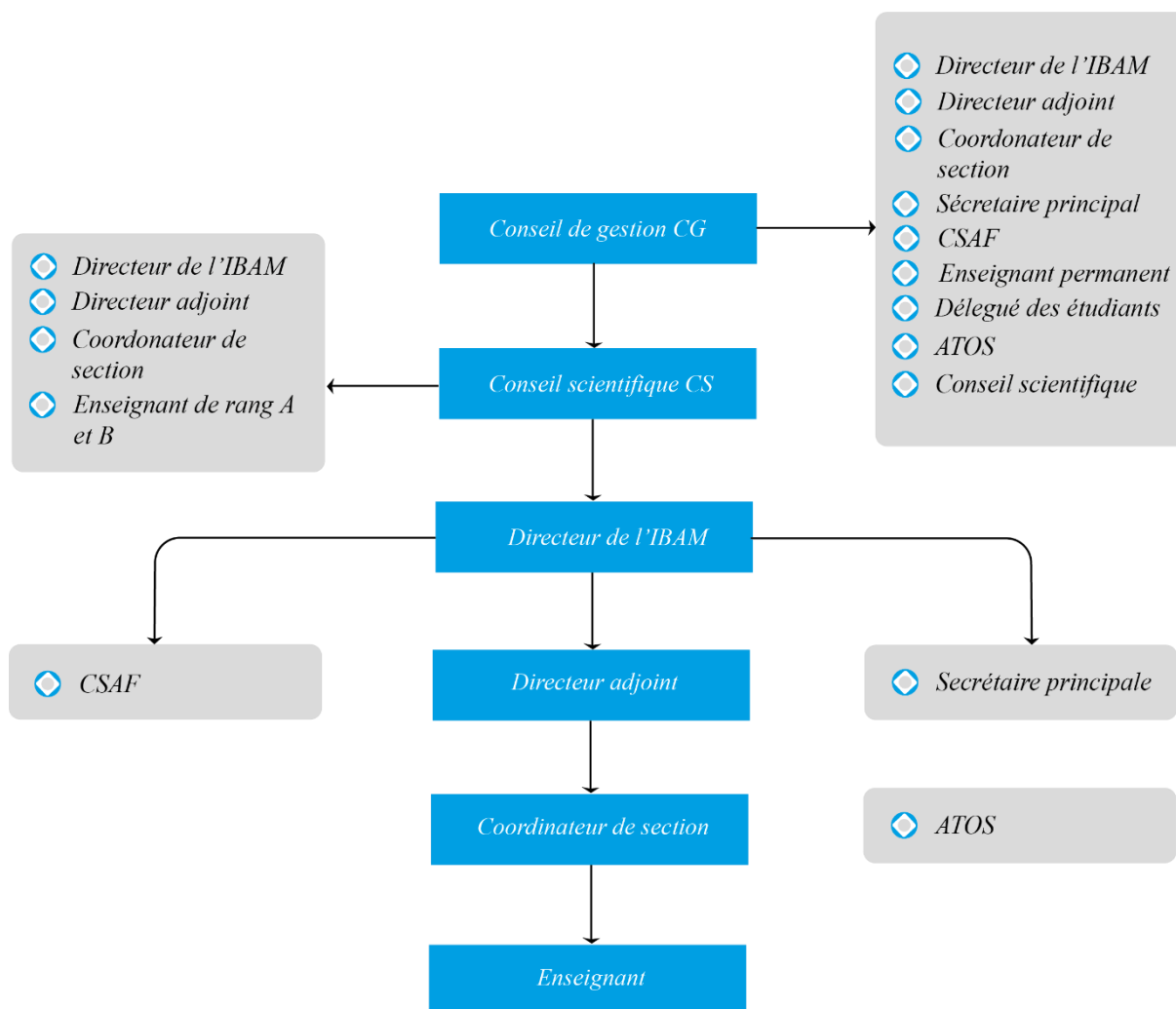


Figure 1: Organigramme de l'IBAM

a) Organe statuaire

L'organe statuaire de l'IBAM est constitué du Conseil de gestion et du Conseil scientifique.

Le **Conseil de gestion** constitue l'organe décisionnel de l'Institut. Il est composé du Directeur de l'IBAM, du Directeur adjoint, des Coordonnateurs de sections, du Secrétaire principal, du

chef du 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 convoquée par son président.

Le conseil de gestion a pour mission 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 composé également 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 d'apprécier les orientations de la recherche au sein de l'institut.

b) Organe d'exécution

L'organe d'exécution a pour rôle d'appliquer les décisions prises par l'organe statutaire. Il est essentiellement constitué de la Direction de l'IBAM.

La Direction est placée sous la responsabilité d'un Directeur, chargé de diriger et de superviser le fonctionnement des différents services, ainsi que de définir la politique générale 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.

Par ailleurs, 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 assistent également le Directeur dans l'accomplissement de certaines activités en lien 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.

a) Licences professionnelles

Les filières de formation initiale en licence professionnelle proposées sont les suivantes :

- ❖ Assurance Banque Finance (ABF) ;
- ❖ Assistant de Direction/Bilingue (ADB) ;
- ❖ Assistant de Direction Comptable (ADC) ;
- ❖ Comptabilité-Contrôle-Audit (CCA) ;
- ❖ Informatique, option Méthodes Informatiques Appliquées à la Gestion (MIAGE) ;
- ❖ Informatique, option Réseaux et Télécommunications (RI-MT) ;
- ❖ Marketing et Innovation Digitale (MID)

Il convient de noter que, depuis l'année académique 2021-2022, l'IBAM ne recrute plus d'étudiants dans la filière Marketing et Gestion. Ainsi, la filière MID a été introduite en 2022-2023 en remplacement et réadaptation de l'ancienne formation.

b) Master

En master, l'IBAM offre cinq (05) filières de formations qui sont :

- ❖ Master en Administration et Gestion des Entreprises (MAGE) ;
- ❖ Master en Banque-Finance (MBF) ;
- ❖ Master en Comptabilité-Contrôle-Audit (MCCA) ;
- ❖ Master en Informatique, option Ingénierie des Systèmes d'Informations en Entreprise (M2ISIE) ;
- ❖ Master en Informatique, option Sécurité Informatique (MISI).

La formation MAGE a été créée pour répondre aux besoins des entreprises en professionnels qualifiés dans les technologies informatiques. Elle accueille en première année les titulaires de baccalauréat des séries C, D et E ayant réussi le test d'admission ou acceptés comme auditeurs libres.

Pour compléter leur formation, les étudiants en fin de cycle MAGE doivent effectuer un stage pratique d'au moins trois mois, suivi d'une soutenance publique. Ce stage vise à les familiariser avec l'environnement professionnel. C'est dans ce cadre que nous avons été accueillis par l'entreprise **DIGITECH BURKINA**, que nous présenterons dans la suite de notre document.

II. Présentation de la structure d'accueil

Nous présentons à présent notre structure d'accueil **DIGITECH BURKINA**

1. Historique

DIGITECH BURKINA est une entreprise privée burkinabè fondée en 2017 par un groupe d'ingénieurs et d'entrepreneurs passionnés par l'innovation numérique. Basée à Ouagadougou, DIGITECH a vu le jour dans un contexte de forte demande pour la digitalisation des services et des entreprises au Burkina Faso.

Initialement spécialisée dans le développement de logiciels sur mesure et les services de conseils en transformation digitale, DIGITECH s'est progressivement imposée comme un acteur majeur dans le domaine des technologies de l'information (IT) et de la gestion de

projets numériques.

Depuis sa création, DIGITECH BURKINA a travaillé en partenariat avec plusieurs institutions publiques et privées pour moderniser leurs processus internes. En 2021, l'entreprise a lancé un vaste programme d'incubation de start-ups numériques burkinabè, témoignant de son engagement envers l'innovation locale.

Aujourd'hui, DIGITECH BURKINA est reconnue pour son expertise dans la conception de plateformes collaboratives, la gestion de projets informatiques, ainsi que dans l'accompagnement des entreprises dans leur transformation digitale.

2. Mission

La mission principale de DIGITECH BURKINA est de contribuer au développement numérique du Burkina Faso et de l'Afrique en :

- ❖ Développant des solutions digitales innovantes adaptées aux réalités locales ;
- ❖ Accompagnant les entreprises et les institutions dans leur transformation numérique ;
- ❖ Favorisant l'accès aux nouvelles technologies à travers des formations et du mentorat ;
- ❖ Encouragent l'entrepreneuriat numérique, notamment auprès des jeunes.

3. Fonctionnement

DIGITECH fonctionne selon un modèle agile et collaboratif :

- ❖ Chaque projet est confié à une équipe multidisciplinaire dirigée par un chef de projet.
- ❖ Les développements suivent principalement la méthodologie Scrum pour garantir une flexibilité et une amélioration continue.
- ❖ L'entreprise met un accent particulier sur le travail collaboratif, l'innovation et la formation continue de ses collaborateurs.

Les principales activités quotidiennes incluent :

- ❖ Le développement de logiciels (applications web, mobiles, cloud),
- ❖ Le conseil stratégique en digitalisation,
- ❖ Le développement de plateformes collaboratives,
- ❖ L'hébergement de solutions cloud sécurisées,

- ❖ Le suivi des projets IT.

4. Projets et partenaires

DIGITECH BURKINA est aujourd'hui à l'origine de plusieurs projets informatiques d'envergure, qui ont un impact significatif au Burkina Faso et rencontrent un large succès.

Parmi les projets les plus notables, on peut citer :

- ❖ **eAgroBurkina** : Plateforme de digitalisation des coopératives agricoles, développée en partenariat avec le ministère de l'Agriculture, visant à moderniser la gestion des coopératives et à faciliter l'accès aux marchés.
- ❖ **SmartSchool** : Application de gestion scolaire numérique, conçue en collaboration avec plusieurs établissements privés, permettant la gestion des inscriptions, des emplois du temps, des notes et de la communication école-parents;
- ❖ **eSanté Faso** : Développement d'un dossier médical électronique sécurisé pour les cliniques privées, afin de digitaliser la gestion des consultations et des dossiers patients.;
- ❖ **JobLinkO Faso**: Plateforme web et mobile de mise en relation entre jeunes diplômés et entreprises locales, facilitant l'insertion professionnelle et répondant aux besoins de recrutement des entreprises.

Certains de ces projets ont été rendus possibles grâce au soutien de partenaires institutionnels et d'organisations engagées dans la promotion du numérique au Burkina Faso. Parmi ces partenaires stratégiques, on peut citer :

- ❖ Ministère de la Transition Digitale, des Postes et des Communications Électroniques (MTDPCE), qui accompagne plusieurs initiatives de digitalisation portées par DIGITECH BURKINA.
- ❖ Le Ministère de la Fonction Publique, du Travail et de la Protection Sociale, impliqué notamment dans les projets liés à la modernisation administrative et à la dématérialisation des services publics.
- ❖ L'Agence Nationale de Promotion des Technologies de l'Information et de la Communication (ANPTIC), partenaire clé pour l'innovation numérique et la promotion de l'entrepreneuriat digital.

- ❖ Des ONG locales et internationales, œuvrant pour l'inclusion numérique, l'éducation technologique et le développement de solutions innovantes adaptées aux besoins des populations.

5. Organisation

DIGITECH BURKINA est organiser suivant l'organigramme suivant :

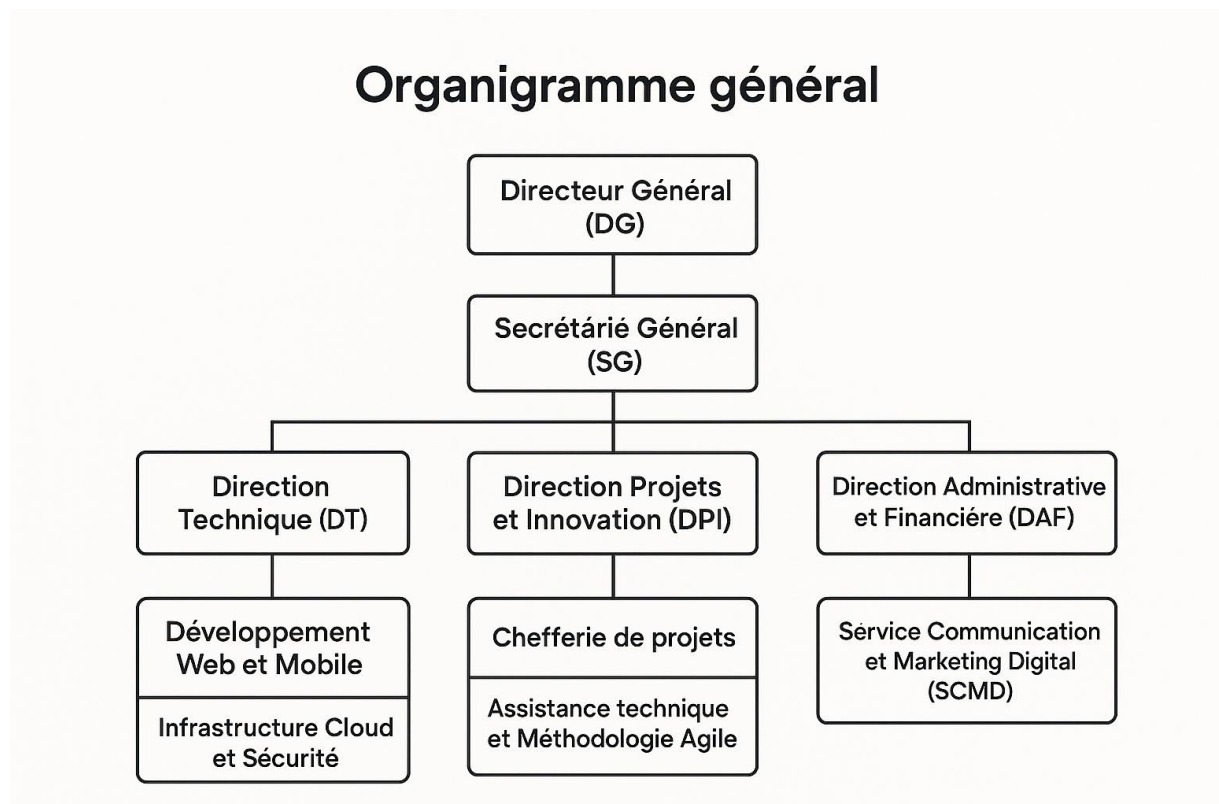


Figure 2: Organigramme de DIGITECH BURKINA

C'est au sein de la **Direction Technique**, et plus précisément au sein du Département **Développement Web et Mobile**, que nous avons été accueillis pour effectuer notre stage. Nous y avons participé à la conception et au développement d'applications innovantes.

Dans le chapitre suivant, nous présenterons l'analyse et la conception du projet qui constituent le cœur de notre thème et de ce rapport.

CHAPITRE II : ANALYSE ET CONCEPTION

Suite à la présentation des structures de formation et d'accueil. Dans ce chapitre, nous commencerons par présenter notre thème, notre méthode d'analyse, les acteurs du projet et le planning prévu. Nous poursuivrons avec l'expression des besoins, suivie d'une étude technique et d'une conception détaillée du système. Enfin, nous détaillerons les différents éléments nécessaires à la réalisation du projet.

I. Étude préalable

Cette partie a pour but de fournir les explications détaillées sur notre thème, la méthode d'analyse et conception utilisée, le groupe de travail du projet et le planning de réalisation.

1. Présentation du thème

La présentation de notre thème s'articule autour de trois (03) axes principaux, à savoir : la problématique, les objectifs et les résultats attendus.

a) Problématique

DIGITECH BURKINA est une entreprise spécialisée dans le développement de solutions numériques qui s'engage à fournir des services de qualité à ses clients dans divers secteurs d'activité. L'entreprise connaît une croissance significative, avec un portefeuille de projets en constante augmentation. Cependant, cette croissance s'accompagne de défis complexes en matière de gestion de projets et de coordination de ses équipes.

En effet, au sein de DIGITECH BURKINA, la gestion des projets et des tâches est actuellement réalisée à travers différents outils non intégrés tels : tableurs excel, documents partagés, messageries électroniques comme Telegram, WhatsApp, etc, ce qui engendre plusieurs problèmes :

- ❖ Difficulté à suivre l'état d'avancement réel des projets en temps réel ;
- ❖ La mauvaise coordination entre les membres d'une équipe ;

- ❖ L'absence de traçabilité et de capitalisation sur les actions menées ;
- ❖ La difficulté d'assurer un suivi rigoureux des délais et des responsabilités ;
- ❖ L'inefficacité des outils de communication et de rappel des échéances

Ces défis sont d'autant plus importants que DIGITECH BURKINA travaille sur des projets multidisciplinaires nécessitant l'intervention de différents profils de compétences tels que les développeurs, designers, chefs de projets, les commerciaux.

Face à ces constats, l'entreprise a entrepris une réflexion approfondie pour répondre à la question fondamentale suivante : « ***Comment DIGITECH BURKINA peut-elle optimiser la gestion de ses projets et la collaboration entre ses équipes afin d'améliorer l'efficacité opérationnelle et la qualité de ses livrables ?*** »

C'est dans cette optique que le sujet suivant nous a été confié : « **ANALYSE ET CONCEPTION D'UNE PLATEFORME INTÉGRÉE DE GESTION DE TÂCHES ET DE PROJETS COLLABORATIFS : CAS DE DIGITECH BURKINA** » Cette initiative vise à créer un environnement numérique unifié permettant une gestion fluide, centralisée et collaborative des projets et des tâches au sein de l'entreprise.

b) Objectifs

Les objectifs fixés pour notre projet sont les suivants :

- ❖ **Analyser en profondeur le système actuel de gestion de projets à C** pour identifier les points forts à consolider et les faiblesses à corriger ;
- ❖ **Concevoir une solution intégrée** répondant aux besoins spécifiques de l'entreprise en matière de gestion de projets collaboratifs ;
- ❖ **Mettre en pratique les connaissances acquises** durant notre formation académique et les enrichir par une expérience concrète en entreprise

c) Résultats attendus

Au regard des besoins exprimés par l'entreprise et notre institution de formation, les résultats attendus à l'issue de ce projet sont :

- ❖ Une plateforme fonctionnelle de gestion collaborative de projets et de tâches qui répond au besoin de DIGITECH BURKINA ;
- ❖ Un rapport de stage complet documentant l'ensemble de la démarche, des analyses et des solutions proposées ;
- ❖ Une soutenance devant un jury présentant les résultats de notre travail et les compétences acquises.

2. Méthodologie

Pour parvenir à la mise en place de cette solution, il sera question pour nous de suivre une méthode de conception et d'analyse de notre solution, afin de fonder une base solide à la réalisation.

Une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client. La phase d'analyse permet de mettre en exergue les résultats attendus et la phase de conception permet, le plus souvent en utilisant un langage de modélisation de mettre en évidence le fonctionnement du système futur.

a) Cycle de développement

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

Le tableau suivant résume une étude comparative entre les principales méthodologies de développement que nous avons choisi vu la diversité de ces méthodes.

Tableau 1 : Tableau comparatif des méthodologies ou processus de développement

METHODOLOGIES	DESCRIPTION	ATOUTS	INSUFFISANCES
RUP (Rational Unified Process)	<p>-A la fois une méthodologie et un outil prêt à l'emploi.</p> <p>-Méthodologie itérative et incrémentale développée par IBM (ex Rational Software) - Focus sur la gestion des risques et la modélisation. - Structurée autour de quatre phases : Inception, Elucidation, Construction et transition</p>	<p>-Approche flexible et Itérative -</p> <p>Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes -</p> <p>Propose des documentations complètes et un bon encadrement des processus</p>	<p>- Assez flou dans sa mise en œuvre. - Ne couvre pas les phases en amont et en aval au développement.</p> <p>- Difficile à adapter pour les équipes moins expérimentées. - Mise en œuvre pouvant nécessiter d'énormes ressources</p>
CASCADES	<p>- Propose de dérouler les phases de projet de manière séquentielle en suivant les étapes linéaires : analyse des exigences, conception, implémentation, tests, déploiement et maintenance - Cité</p>	<p>- Distingue clairement les phases projet</p> <p>- Simple à comprendre et à gérer.</p> <p>Bien adapté pour les projets avec des exigences claires et stables.</p>	<p>- Non itératif : Difficile de revenir en arrière lorsqu'une phase est terminée -</p> <p>Les erreurs détectées tardivement sont coûteuses à corriger.</p> <p>- Moins adapté aux projets complexes ou en évolution rapide.</p>

	pour des raisons historiques		- Ne propose pas de modèles de documents
Two-Track Unified Process (2TUP)	<p>Concentré sur deux tracks (A et B) :</p> <p>Gestion des exigences et des utilisateurs,</p> <p>Construction et déploiement techniques</p> <ul style="list-style-type: none"> - S'articule autour de l'architecture - Propose un cycle de développement en Y - Cible des projets de toutes tailles - Assure l'alignement des besoins métier avec le développement technique 	<ul style="list-style-type: none"> - Flexible et itératif, favorisant une réponse au changement. - Fait une large place à la technologie et à la gestion du risque - Définis les profils des intervenants, les livrables, les plannings, les prototypes - Bonne équilibre en les exigences métiers et le développement technique 	<ul style="list-style-type: none"> - Nécessite un bon niveau de discipline et de communication entre les membres de l'équipe - Ne propose pas de documents types
XP (eXtreme Programming)	<ul style="list-style-type: none"> - Ensemble des bonnes pratiques de développement. - Développement par petits incréments, automatisation des tests, 	<ul style="list-style-type: none"> - Excellente qualité du logiciel grâce aux tests rigoureux - Adaptation aux changements grâce aux petits incréments 	<ul style="list-style-type: none"> - Assez flou dans sa mise en œuvre. - Ne couvre pas les phases en amont et en aval au développement. - Elude la phase d'analyse, si bien

	<p>programmation en binôme, l'intégration continue</p> <p>-Axé sur la qualité du logiciel et la réactivité au changement</p>	<p>- Itératif et donne une importance aux aspects techniques.</p> <p>- Innovant : programmation en duo.</p>	<p>qu'on peut dépenser son énergie à faire et défaire</p>
Scrum	<p>- Se base sur des itérations dites sprints de développement -Met l'accent sur l'auto organisation des équipes, les réunions quotidiennes (Daily standups) et les revues de sprints</p>	<p>- Donne toute confiance aux développeurs et les laisser effectuer leur travail.</p> <p>Chaque itération a un objectif bien précis et fournit une fonctionnalité testée.</p>	<p>- La mise en œuvre du développement n'est pas précisée. -Peux devenir chaotique sans une bonne gestion.</p> <p>Développement rapide et répétitif - Moins de documentation formelle, ce qui pose des problèmes</p>

Après avoir examiné et comparé les processus de développement, dans le but de gérer les risques et de réussir notre projet, nous avons sélectionné le processus 2TUP pour de multiples raisons :

- ❖ 2TUP accorde une grande importance aux aspects technologiques. C'est un critère crucial pour nous étant donné les enjeux techniques de notre projet.
- ❖ la structure en Y de 2TUP permet de faire évoluer séparément les volets fonctionnels et techniques du projet. Cette indépendance nous donne l'agilité nécessaire pour nous adapter aux changements d'une branche indépendamment de l'autre.

Le processus 2TUP 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 :

➤ **La branche fonctionnelle qui comporte :**

- la capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier. Cette capture des besoins permet d'éviter au plus tôt le risque de produire un système inadapté aux utilisateurs.
- l'analyse, qui consiste à étudier les spécifications fonctionnelles dans le but d'avoir une idée de ce que va réaliser le système en termes de métier.

Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

➤ **La branche technique qui comporte :**

- **la capture des besoins techniques** qui consiste à énumérer les outils, matériels et technologies dont nous avons besoin. Elle permet de prendre en compte des contraintes (intégration avec l'existant). L'ensemble permet de construire une 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 qui comporte :

- **la conception préliminaire**, une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à 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, tests** qui consistent à implémenter et à tester les fonctionnalités du système au fur et à mesure.

- **la recette**, qui consiste en la validation des fonctionnalités du système développé.

La figure ci-dessous présente le processus 2TUP :

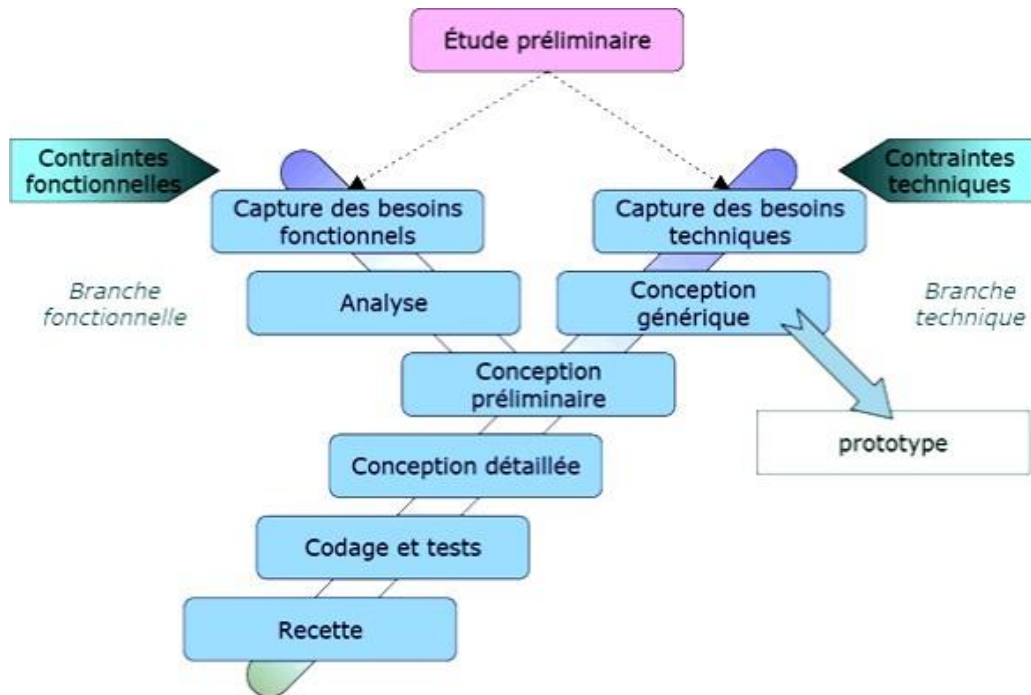


Figure 3: Schéma représentant le processus 2TUP

Source : <https://www.memoireonline.com/05/13/7195>

Le processus 2TUP s'appuie sur le langage UML tout au long du cycle de développement.

b) Langage de modélisation

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer l'information ou 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 ». Il existe plusieurs langages de modélisation parmi lesquels nous pouvons citer : BPMN , MBT, SysML et UML.

Dans le tableau 2 nous faisons une étude comparative entre les langages de modélisation cités ;

Tableau 2: Comparaison de quelques langages de modélisation

Langages de modélisation	Caractéristiques
BPMN (Business Process Model and Notation)	<ul style="list-style-type: none"> • Modèle de processus pour représenter le déroulement des processus internes ou publique d'une organisation ; • vise l'analyse et la conception des processus métiers qui font intervenir et interagir des systèmes ; • son but principal est de fournir une notation qui soit facilement compréhensible par tous les utilisateurs de l'entreprise.
MBT (Model Based Testing)	<ul style="list-style-type: none"> • Déclinaison des solutions d'ingénierie des modèles appliquées au test ; • permet de produire des cas de tests à partir d'un modèle ; • diminue les efforts de maintenance des jeux de tests ; <p>renforce la qualité de la documentation des exigences.</p>
SysML (Système Modeling Langage)	<ul style="list-style-type: none"> • Langage de modélisation spécifique au domaine de l'ingénierie système ; • sémantique de SysML est plus riche et flexible ; • langage plus réduit que UML.
BPMN (Business Process Model and Notation)	<ul style="list-style-type: none"> • Modèle de processus pour représenter le déroulement des processus internes ou publique d'une organisation ; • vise l'analyse et la conception des processus métiers qui font intervenir et interagir des systèmes ; • son but principal est de fournir une notation qui soit facilement compréhensible par tous les utilisateurs de l'entreprise.
UML (Unified Modeling Langage)	<ul style="list-style-type: none"> • Langage de représentation d'un système d'information ; • système de notation orienté objet ; • plus orientée vers la conception ;

- | | |
|--|--|
| | <ul style="list-style-type: none">• international. |
|--|--|

Après notre analyse comparative, nous avons opté pour le langage UML en raison de ses nombreux avantages :

- ❖ Il représente la norme internationale reconnue pour la modélisation objet
- ❖ Sa notation graphique permet de visualiser clairement les solutions objet, facilitant leur comparaison et évaluation
- ❖ Son caractère formel et normalisé assure précision et stabilité dans le processus de modélisation
- ❖ Il permet de formaliser tous les documents techniques d'un projet, affinant l'analyse au fil de l'évolution
- ❖ Il offre un ensemble cohérent d'outils de génie logiciel, depuis l'expression des besoins jusqu'à la génération de code
- ❖ Il constitue un excellent support de communication, encadrant l'analyse tout en simplifiant la compréhension des concepts abstraits complexes

UML propose un ensemble standardisé de notations, diagrammes et symboles pour décrire la structure, le comportement, les interactions et les aspects dynamiques des systèmes, comme illustré dans la Figure 4.

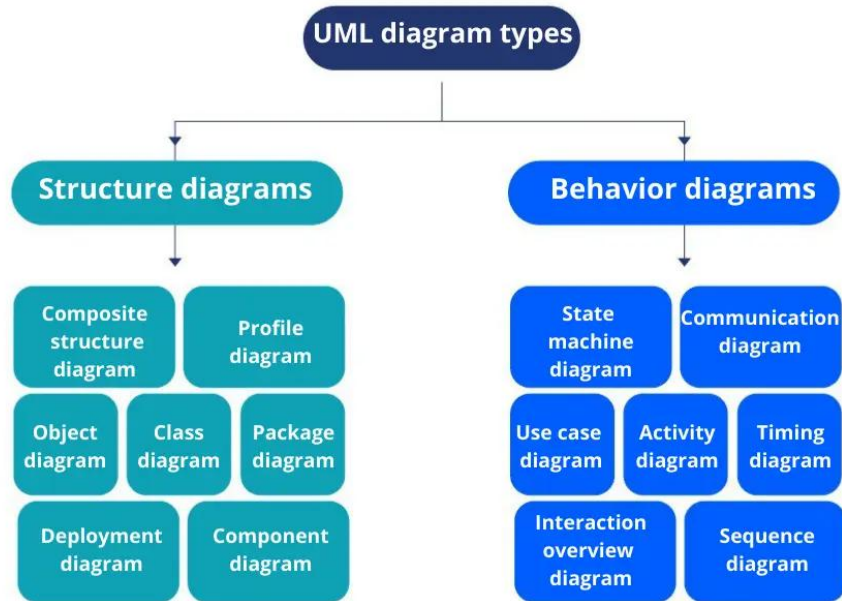


Figure 4 : Illustration de la hiérarchie schématique selon UML

Source <https://www.gleek.io/blog/uml-diagram-types>

3. Présentation du groupe de travail

Le groupe de travail comprend l'ensemble des personnes impliquées dans la réalisation du projet. Nous distinguons trois groupes principaux à savoir : le groupe projet, le groupe de pilotage et le groupe utilisateur.

a) Groupe de pilotage

Ce groupe dirigeant supervise le bon déroulement du projet. Il est responsable de la planification des étapes clés, de l'évaluation des propositions du groupe de projet et des décisions stratégiques. Il est constitué de :

- ❖ Dr Ferdinand GUINKO, notre enseignant encadrant.

b) Groupe des utilisateurs

Ce groupe rassemble les futurs utilisateurs du système. Il contribue à l'identification des besoins fonctionnels et comprend principalement les porteurs de projets et les employés de DIGITECH BURKINA.

c) Groupe de projet

Le groupe de projet est chargé de la réalisation effective du projet et sert d'intermédiaire entre le groupe de pilotage et les utilisateurs. Ce groupe est composé essentiellement de nous-même Abdoul Razack OUEDRAOGO, étudiants en Troisième année de MIAGE à l'IBAM.

Après avoir identifié les différents intervenants du projet de système de gestion des tâches et projets collaboratifs au Burkina, voici notre planning de réalisation.

d) Planning de travail

Pour organiser notre étude efficacement, nous avons décomposé le projet en tâches spécifiques et élaboré un diagramme de Gantt. Cet outil nous permet de visualiser le calendrier prévisionnel des différentes phases du projet, comme détaillé dans la figure ci-dessous.

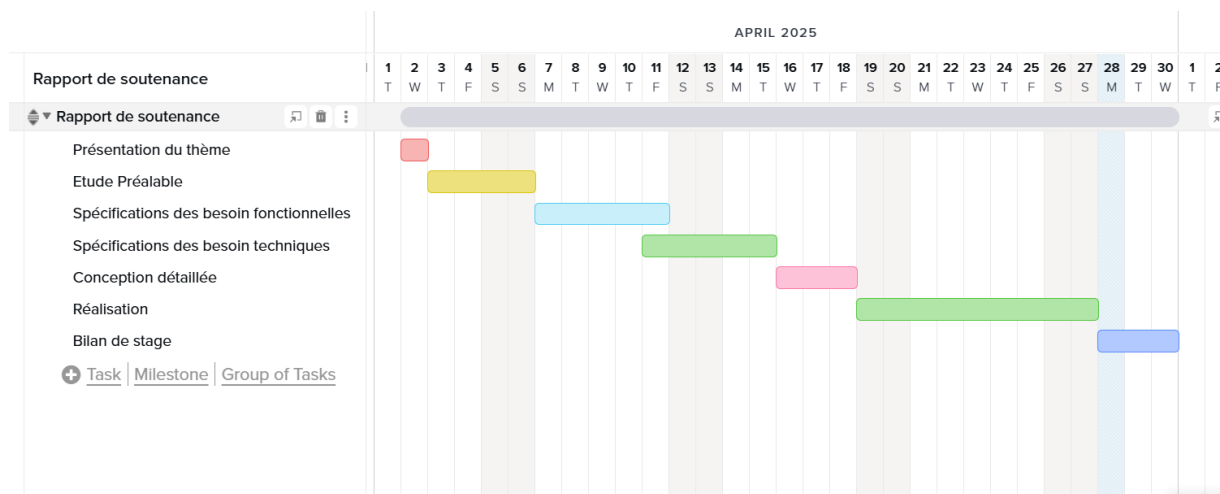


Figure 5 : Planning de réalisation du projet

La Figure 5 illustre le déroulement de notre stage fictif, qui s'est étendu du 02 au 30 avril 2025. Durant cette période, nous avons méthodiquement réalisé les phases suivantes : la présentation du thème, l'étude préliminaire, l'analyse des besoins, la conception globale, la réalisation, le bilan et la réalisation des livrables.

À l'issue de la présentation du planning de réalisation, il convient désormais de s'intéresser à l'expression des besoins, en mettant en évidence les acteurs concernés et les échanges de messages avec le système.

II. Expression des besoins

Cette partie est consacrée, d'une part, à l'analyse du fonctionnement des outils existants pour la gestion des projets au sein de DIGITECH BURKINA et aux améliorations envisagées. D'autre part, elle présente les acteurs concernés par la gestion des projets et des tâches, ainsi que les types d'échanges réalisés entre ces acteurs et le système en cours de conception.

1. Étude de l'existant

L'étude de l'existant constitue une démarche méthodique permettant d'analyser de manière approfondie le fonctionnement actuel d'une organisation, d'un processus ou d'un système. Dans ce contexte, nous allons examiner la situation actuelle de la gestion de projets et de tâches au sein de DIGITECH BURKINA.

a) Présentation du système utilisé à DIGITECH BURKINA

La présentation du système de gestion en place au sein de DIGITECH BURKINA vise à identifier ses points forts et ses limites, afin de définir précisément les besoins de l'entreprise pour la conception de la solution proposée.

La présentation du système actuel de gestion des projet et taches en place au sein de DIGITECH BURKINA vise à identifier ses points forts et ses limites, afin de définir précisément les besoins de l'entreprise pour la conception de la solution proposée.

Actuellement, DIGITECH BURKINA ne dispose pas d'une plateforme unifiée pour la planification, le suivi et la coordination de ses projets. La gestion des projets s'appuie sur des outils hétérogènes tels que des tableurs Excel, des documents partagés, ainsi que des applications de messagerie instantanée comme Telegram et WhatsApp.

Les chefs de projet, développeurs, designers et commerciaux échangent donc principalement via ces canaux pour communiquer l'état d'avancement des projets, attribuer des tâches et assurer la coordination entre les équipes.

Cependant, cette organisation, bien qu'elle permette une communication rapide, révèle

plusieurs limites majeures :

- ❖ La difficulté à obtenir une vision d'ensemble en temps réel de l'avancement des projets ;
- ❖ Une mauvaise coordination entre les membres des équipes projet ;
- ❖ L'absence de traçabilité structurée des actions menées ;
- ❖ Le manque de suivi rigoureux des échéances et responsabilités ;
- ❖ Une inefficacité notable dans la communication et le rappel des tâches.

À l'issue de cette analyse, nous pouvons dresser une synthèse des forces et faiblesses du système existant :

❖ **Forces du système actuel :**

- ✓ Communication rapide et immédiate via les applications de messagerie ;
- ✓ Renforcement des liens informels entre les membres des équipes.

❖ **Faiblesses du système actuel :**

- Manque d'intégration des outils utilisés ;
- Difficulté de suivi structuré et centralisé des projets ;
- Perte d'efficacité dans la gestion du temps et des ressources ;
- Faible capitalisation des données et historique des projets.

Ces constats démontrent la nécessité, pour DIGITECH BURKINA, de se doter d'une solution numérique adaptée, capable de centraliser la gestion des projets, d'améliorer la coordination des équipes, et de garantir une meilleure traçabilité des activités.

b) Exigences fonctionnelles

Afin de pallier les insuffisances identifiées, DIGITECH BURKINA a défini un ensemble d'exigences fonctionnelles destinées à guider la conception de la nouvelle plateforme de gestion de projets.

Ces exigences comprennent :

- ❖ La mise en place d'interfaces distinctes pour les administrateurs, chefs de projets,

développeurs, designers et commerciaux ;

- ❖ La personnalisation des tableaux de bord en fonction du profil utilisateur ;
- ❖ La centralisation et la traçabilité des informations relatives aux projets et aux tâches ;
- ❖ La possibilité d'assurer un suivi rigoureux des échéances, des responsabilités et des avancements ;
- ❖ L'intégration de rappels et de notifications automatiques pour les échéances importantes ;
- ❖ Une solution économiquement viable, avec un coût d'exploitation maîtrisé à long terme.

Par cette démarche, DIGITECH BURKINA affirme sa volonté de disposer d'une solution performante, évolutive et adaptée à son mode de fonctionnement interne.

c) Présentation d'outils logiciels de gestion de projet existants

La gestion de projets constitue aujourd'hui un défi majeur pour les entreprises, ce qui a favorisé l'émergence de nombreux outils dédiés sur le marché.

Après une analyse approfondie du marché, nous avons sélectionné plusieurs outils performants et populaires qui sont : **Monday.com, ClickUp, Wrike, Asana et Notion.**

Nous allons évaluer ces plateformes selon les critères suivants :

- ❖ **Gestion des tâches et sous-tâches** : Possibilité de créer, attribuer et suivre des tâches ;
- ❖ **Automatisation des processus** : Capacité à automatiser certaines actions dont les notifications, les assignations et les rappels.
- ❖ **Collaboration en temps réel** : Outils de communication interne intégrés à savoir les chats et les commentaires sur les tâches.
- ❖ **Personnalisation** : Adaptabilité des tableaux de bord et des vues de projet.
- ❖ **Suivi des temps** : Fonction de mesure du temps passé sur chaque tâche ou projet.
- ❖ **Intégrations externes** : Capacité à se connecter avec d'autres outils collaboratifs tels que Google Drive, Slack, Zoom
- ❖ **Modèle de tarification** : Coût par utilisateur et flexibilité du modèle économique.

À partir de ces critères, nous allons établir un tableau comparatif permettant d'évaluer les différentes solutions identifiées

Tableau 3 :Présentation de solutions logicielles de gestion de projets existantes

Critères	Monday.com	ClickUp	Wrike	Asana	Notion
Gestion des tâches et sous-tâches	Oui	Oui	Oui	Oui	Oui
Automatisation des processus	Oui	Oui	Oui	Limité	Limité
Collaboration en temps réel	Oui	Oui	Oui	Limité	Limité
Personnalisation	Oui	Oui	Limité	Oui	Oui
Suivi des temps	Oui	Oui	Oui	Non	Non
Intégrations externes	Non	Oui	Oui	Oui	Oui
Modèle de tarification	À partir de 5 200 FCFA	À partir de 3 250 FCFA	À partir de 5 850 FCFA	À partir de 6 500 FCFA	À partir de 2 600 FCFA

Bien que les outils présentés offrent des fonctionnalités variées et puissantes, ils ne répondent pas pleinement aux besoins spécifiques de DIGITECH BURKINA. En effet, bien que ces solutions soient adaptées à des structures de taille variable, elles ne permettent pas toujours une personnalisation suffisante pour les exigences particulières de l'entreprise, notamment en termes de gestion multi-utilisateurs, d'intégration complète dans les processus internes, et de coûts adaptés à une entreprise en pleine croissance.

2. Présentation du processus de fonctionnement amélioré

Dans le but d'améliorer la coordination et la productivité au sein de DIGITECH BURKINA, un nouveau système de gestion de projets a été mis en place à travers une plateforme web nommée TaskFlow.

Cette solution innovante permet aux différentes équipes de collaborer en temps réel, de suivre l'avancement des projets et de centraliser toutes les informations clés nécessaires au pilotage des activités.

TaskFlow a été pensée pour répondre aux besoins spécifiques de DIGITECH BURKINA en matière de suivi de projet, de planification stratégique et d'efficacité opérationnelle. Elle facilite la communication entre les chefs de projet, les membres des équipes (développeurs, designers, commerciaux) et les responsables, tout au long du cycle de vie d'un projet.

La plateforme comporte deux interfaces principales : une interface administrateur et une interface dédiée aux collaborateurs à savoir les développeurs, designers, chefs de projets et les commerciaux de DIGITECH BURKINA.

❖ Interface Administrateur

Depuis l'interface web, l'administrateur peut :

- Créer et gérer les comptes des utilisateurs associés aux différents projets ;
- Accorder les permissions nécessaires selon les rôles de chacun ;
- Créer de nouveaux projets avec leurs descriptions détaillées ;
- Générer des codes d'invitation uniques pour permettre aux collaborateurs de rejoindre les projets ;
- Consulter l'ensemble des projets de l'entreprise et leurs états d'avancement ;
- Assigner des tâches spécifiques aux membres d'équipe et suivre leur progression.

❖ Interface Collaborateurs

Après la création de leur compte par l'administrateur ou leur intégration via un code d'invitation, les collaborateurs peuvent, depuis la version web, après s'être authentifiés :

- Consulter les projets auxquels ils sont associés
- Vérifier l'état d'avancement de leurs projets en temps réel
- Prendre connaissance des tâches qui leur sont assignées
- Mettre à jour le statut de leurs tâches (À faire, En cours, Bloqué, Terminé)
- Fournir des descriptions détaillées des solutions apportées pour les tâches accomplies
- Consulter les détails de chaque tâche, incluant les dates d'échéance et les responsables
- Collaborer avec les autres membres du projet via un système centralisé

Notre solution permet une gestion fluide des projets selon les étapes suivantes :

1. **Création du projet** : Le chef de projet ou l'administrateur crée un nouveau projet avec un nom, une description et génère un code d'invitation unique.
2. **Constitution de l'équipe** : Les collaborateurs rejoignent le projet soit par assignation directe, soit en utilisant le code d'invitation fourni.
3. **Planification des tâches** : Les responsables définissent les tâches nécessaires à la réalisation du projet, avec des descriptions claires, des dates d'échéance et les assignent aux membres appropriés.
4. **Suivi et exécution** : Chaque membre accède à son tableau de bord personnel montrant ses tâches et leur statut, et peut mettre à jour l'avancement de son travail en temps réel.
5. **Collaboration** : Tous les membres peuvent visualiser l'état global du projet, facilitant ainsi la coordination et la communication entre les équipes multidisciplinaires.
6. **Finalisation et documentation** : Lorsqu'une tâche est terminée, le responsable peut documenter la solution implémentée, permettant ainsi une capitalisation des connaissances pour l'entreprise.

Ce nouveau processus de fonctionnement permet de résoudre les problématiques identifiées en offrant une visibilité en temps réel sur l'état des projets, en améliorant la coordination entre les équipes, en assurant une traçabilité complète des actions, en facilitant le suivi rigoureux des délais et des responsabilités, et en intégrant des outils efficaces de communication et de rappel des échéances.

Grâce à cette plateforme intégrée, DIGITECH BURKINA peut désormais optimiser sa gestion de projets et la collaboration entre ses équipes, contribuant ainsi à l'amélioration de son efficacité opérationnelle et de la qualité de ses livrables.

Suite à cette présentation du modèle amélioré de gestion des activités, nous abordons désormais l'analyse détaillée des spécifications fonctionnelles.

3. Spécification fonctionnelle

La spécification fonctionnelle constitue l'exposé méthodique des capacités et comportements attendus du logiciel avant sa mise en œuvre. Dans cette section, nous allons explorer en profondeur les exigences du système proposé à travers l'identification précise des utilisateurs concernés, la définition des scénarios d'utilisation et l'élaboration des diagrammes explicatifs nécessaires.

a) Identification des acteurs

Un acteur définit un ensemble cohérent de rôles qu'un utilisateur ou une entité externe peut jouer en interagissant avec le système. Dans le cadre de notre plateforme TaskFlow pour DIGITECH BURKINA, les acteurs identifiés sont :

- ❖ **L'administrateur** : Il est responsable de la gestion globale de la plateforme. Il dispose des droits les plus étendus, lui permettant de paramétrer le système, de créer des comptes utilisateurs et de superviser l'ensemble des projets.
- ❖ **Le responsable** : Il assure la coordination et le suivi des projets. Il définit les objectifs, planifie les tâches, assigne les travaux aux employés et veille au respect des délais.
- ❖ **L'employé** : Collaborateur technique de l'entreprise, tel qu'un développeur ou un designer, il exécute les tâches spécifiques qui lui sont confiées dans son domaine d'expertise et contribue activement à la réalisation des projets.

b) Les cas d'utilisation

Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.

Les cas d'utilisation désignent l'ensemble des interactions qui permettent à un acteur d'atteindre son objectif en utilisant le système. Après l'analyse des processus décrits plus haut et des entrevues effectuées auprès des acteurs, nous avons identifié quinze (14) cas d'utilisation inscrits dans le Tableau ci-après .

Tableau 4: Listes des cas d'utilisation

Numéro	Cas d'utilisation	Description	Acteurs
CU_01	S'authentification		Tous les acteurs
CU_02	Gérer les utilisateurs		Responsable
CU_03	Gérer fournisseurs		Responsable
CU_04	Gérer Clients		Employé
CU_05	Gérer les types de retenus.		Responsable, Employé
CU_06	Gérer garanties de bonne exécution.		Responsable
CU_07	Enregistrer Facture Fournisseur		Responsable
CU_08	Enregistrer Facture Client		Responsable
CU_09	Associer une Facture Client à une garantie		Responsable, Employé
CU_10	Importer facture client depuis Odoo		Responsable, Employé

<i>CU_11</i>	Enregistrer un règlement fournisseur		Employé	
<i>CU_12</i>	Enregistrer un règlement client		Employé	
<i>CU_13</i>	Mentionner des retenues sur un règlement client			
<i>CU_14</i>	Gérer les utilisateur		Administrateur	
<i>CU_15</i>	Lister les factures selon un critère			
<i>CU_16</i>	Consulter l'historique des règlements			
<i>CU_17</i>	Consulter l'état d'une facture			
<i>CU_18</i>	Surveiller échéances et événements d'alerte			
<i>CU_19</i>	Évaluer statut des factures			
<i>CU_20</i>	Envoyer notifications			

CU_21	Recevoir & consulter les notifications			
-------	--	--	--	--

INTRODUCT

Pour une meilleure expérience de visualisation des cas d'utilisation, nous représenterons ces derniers dans un diagramme de cas d'utilisation, qui représentera le diagramme de cas d'utilisation de notre plateforme

c) Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation UML est un outil mis en place pour faciliter la représentation fonctionnelle d'un système informatique. Les cas d'utilisations étant les objets clés de ce diagramme, permettent de représenter chaque action que chaque acteur peut émettre dans le système. En bref, le diagramme de cas d'utilisation permet une meilleure expérience de vue d'ensemble d'un système futur. La figure ci-dessous représente celui de notre plateforme :

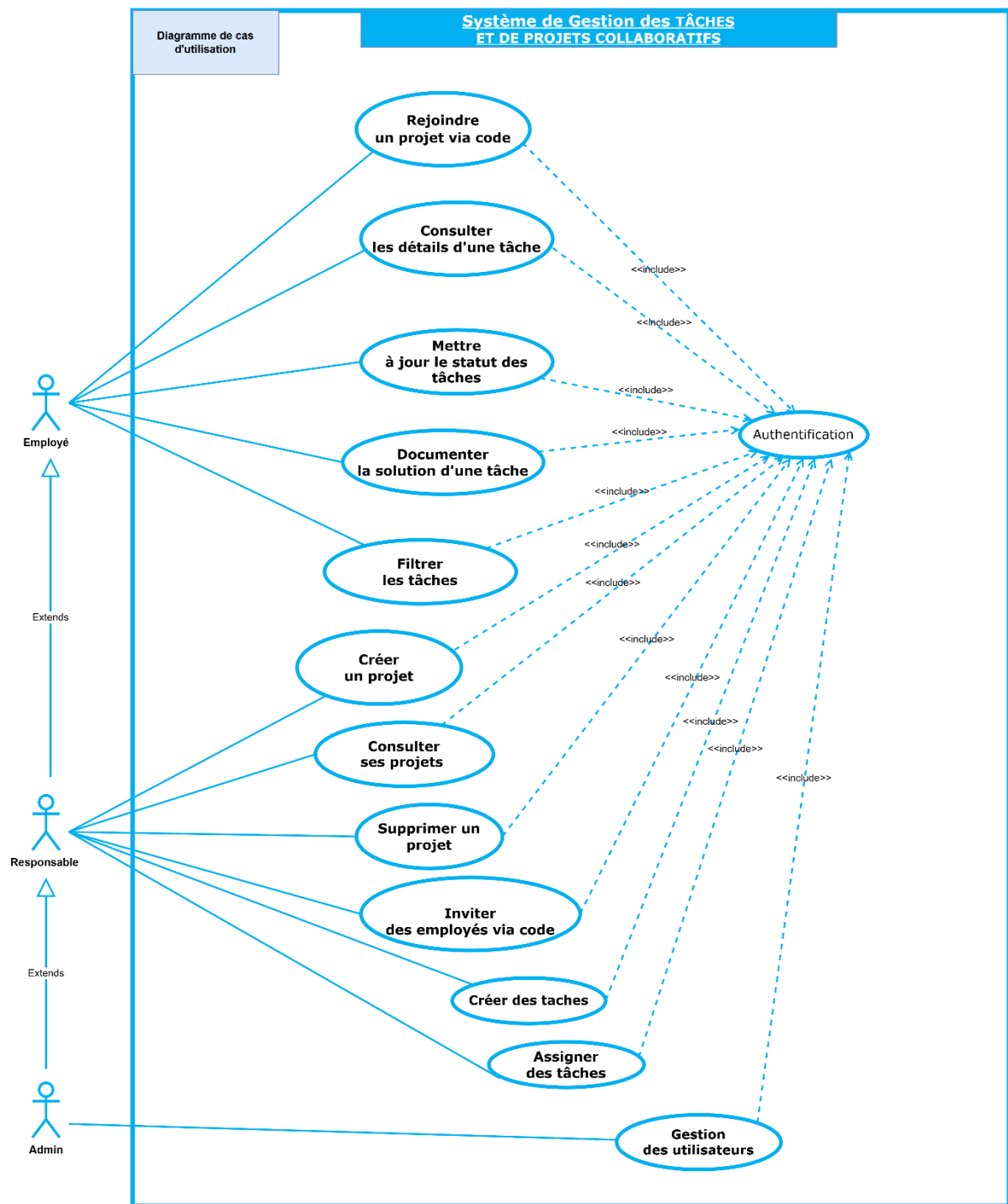
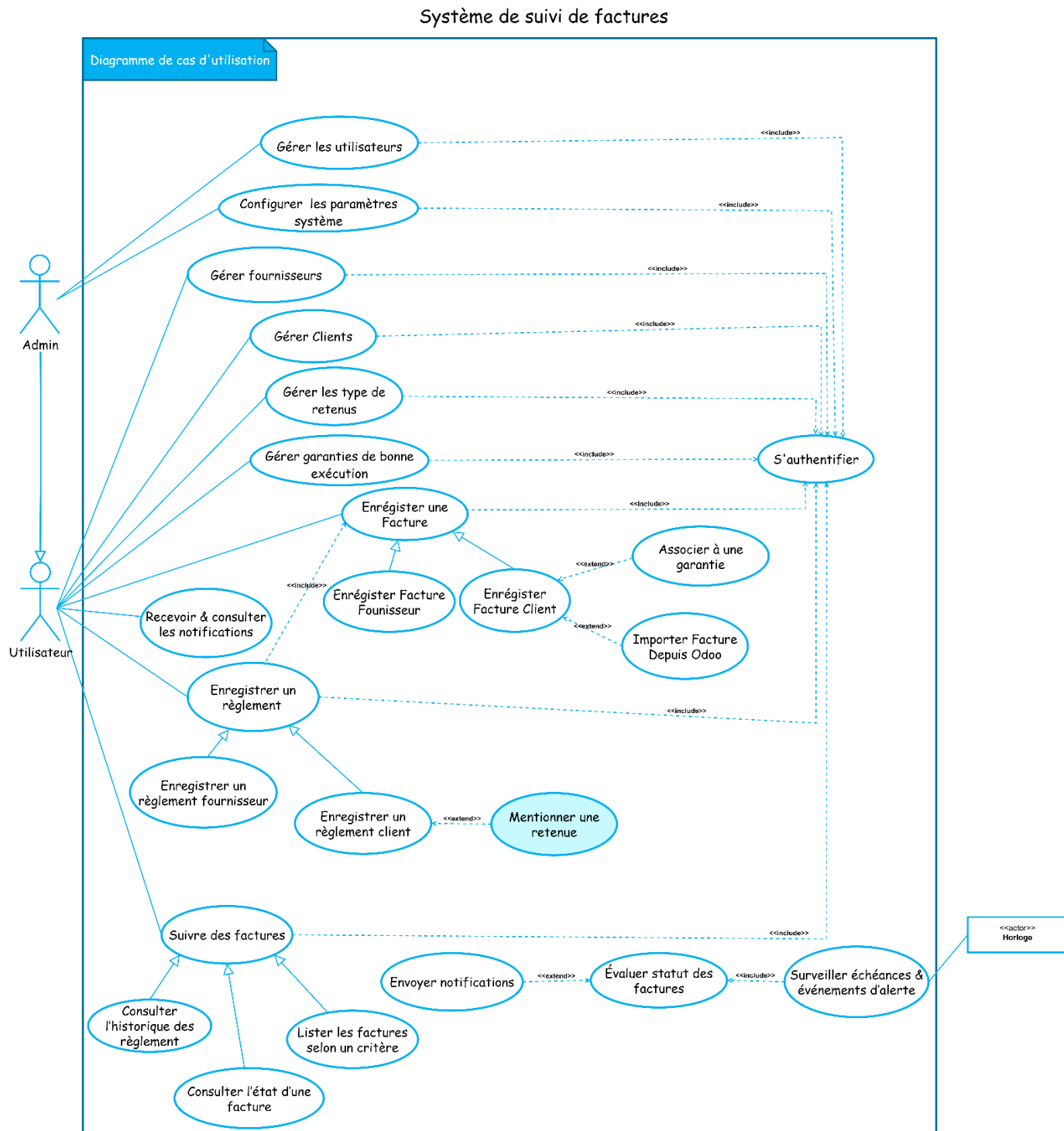


Figure 6 : Diagramme de cas d'utilisation

Remarque : Tous les cas d'utilisation inclut le cas d'utilisation « Authentification ».

L'authentification est gérée par Clerk authentification.





d) Description textuelle de certains cas d'utilisation

La description textuelle des cas d'utilisation permet d'exposer de manière détaillée les interactions entre les acteurs et le système TaskFlow. Elle vise à documenter précisément

comment le système réagit aux actions des utilisateurs pour accomplir des fonctions spécifiques liées à la gestion des projets.

Chaque description textuelle est organisée sous forme de tableau comportant plusieurs sections :

- ❖ **Acteurs** : liste des utilisateurs ou entités externes impliqués dans le cas d'utilisation ;
- ❖ **Précondition** : état requis du système avant le déclenchement du cas d'utilisation ;
- ❖ **Scénario nominal** : déroulement normal des interactions entre l'acteur et le système sans erreur ;
- ❖ **Scénario alternatif** : variantes du scénario nominal en cas d'erreurs ou d'événements particuliers ;
- ❖ **Postcondition** : état final du système après l'exécution du cas d'utilisation.

Pour garantir la clarté et la lisibilité du document, seuls certains cas d'utilisation considérés comme essentiels à la compréhension du fonctionnement de la plateforme TaskFlow ont été détaillés dans cette section. Il s'agit notamment des cas suivants : « **Authentification** », « **Créer un projet** », « **Créer des tâches** », « **Assigner des tâches** » et « **Mettre à jour le statut d'une tâche** ».

□ **CU01 : Authentification**

Tableau 5: Description textuel du cas d'utilisation : s'authentifier

Résumé	Ce cas d'utilisation permet à un utilisateur (administrateur, responsable ou employé) de se connecter à la plateforme TaskFlow.
Acteurs	Administrateur, Responsable, Employé
Précondition	<ul style="list-style-type: none"> - L'utilisateur doit disposer d'un compte enregistré sur la plateforme TaskFlow. - Il doit avoir ses identifiants d'accès (nom d'utilisateur et mot de passe).

Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de connexion de TaskFlow. 2. Il saisit son nom d'utilisateur et son mot de passe. 3. Le système vérifie les informations d'identification [A]. 4. Le système authentifie l'utilisateur. 5. L'utilisateur est redirigé vers la page principale de l'application.
Scenario alternatif	<p>[A] : Nom d'utilisateur ou mot de passe incorrect</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur indiquant que les informations sont incorrectes. • Le système présente à nouveau la page d'authentification. • L'utilisateur peut ressaisir ses informations et reprendre le scénario nominal à partir du point 2
Postcondition	L'utilisateur est connecté à la plateforme TaskFlow et accède à son tableau de bord principal.

□ **CU02 : Créer un projet**

Tableau 6 : Description textuel du cas d'utilisation : créer un projet

Résumé	Ce cas d'utilisation permet au responsable de créer un nouveau projet dans TaskFlow.
Acteurs	Responsable
Précondition	<ul style="list-style-type: none"> - Le responsable doit être connecté à la plateforme TaskFlow. - Le responsable doit avoir les droits nécessaires pour créer un projet.
Scenario nominal	<ol style="list-style-type: none"> 1. Le responsable accède à l'interface de création de projet. 2. Il saisit les informations requises (nom du projet, description, date de début, date de fin, etc.). 3. Il valide la création du projet. 4. Le système enregistre le projet dans la base de données.

	5. Le responsable est redirigé vers la page de gestion du projet nouvellement créé.
Scenario alternatif	<p>[A] : Informations manquantes ou invalides</p> <ul style="list-style-type: none"> Le système affiche un message d'erreur précisant les champs à corriger. Le système présente à nouveau le formulaire de création de projet. Le responsable corrige les informations et reprend le scénario nominal à partir du point 2.
Postcondition	Le nouveau projet est enregistré et visible dans la liste des projets du responsable.

□ **CU07 : Créer des tâches**

Tableau 7 : Description textuelle du cas Créer des tâches

Résumé	Ce cas d'utilisation permet au responsable de créer des tâches pour un projet donné.
Acteurs	Responsable
Précondition	<ul style="list-style-type: none"> Le responsable doit être connecté à la plateforme TaskFlow. Un projet doit déjà exister.
Scenario nominal	<ul style="list-style-type: none"> Le responsable accède à la section "Tâches" du projet. Il clique sur "Créer une tâche". Il saisit les informations de la tâche (titre, description, priorité, échéance, etc.). Il valide la création de la tâche.

	<ul style="list-style-type: none"> - Le système enregistre la tâche et l'associe au projet.
Scenario alternatif	<p>[A] : Informations de tâche incomplètes</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur précisant les champs manquants. • Le responsable complète les informations et reprend à partir du point 3.
Postcondition	La nouvelle tâche est ajoutée à la liste des tâches du projet.

□ **CU08 : Assigner des tâches**

Tableau 8 : Description textuelle du cas Assigner des tâches

Résumé	Ce cas d'utilisation permet au responsable d'attribuer une tâche à un ou plusieurs employés.
Acteurs	Responsable
Précondition	<ul style="list-style-type: none"> - Le responsable doit être connecté. - La tâche doit exister dans le projet. - Les employés doivent être membres du projet.

Scenario nominal	<ol style="list-style-type: none"> 1. Le responsable accède à la liste des tâches du projet. 2. Il sélectionne une tâche à assigner. 3. Il choisit un ou plusieurs employés parmi les membres du projet. 4. Il confirme l'assignation. 5. Le système enregistre l'attribution et notifie les employés concernés.
Scenario alternatif	<p>[A] : Aucun employé sélectionné</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur invitant à sélectionner au moins un employé. • Le responsable sélectionne un employé et reprend à partir du point 3.
Postcondition	La tâche est assignée aux employés choisis et visible dans leur liste de tâches.

□ **CU011 : Mettre à jour le statut d'une tâche**

Tableau 9: Description textuelle du cas Mettre à jour le statut d'une tâche

Résumé	Ce cas d'utilisation permet à un employé de changer le statut d'une tâche selon son avancement.
Acteurs	Employé
Précondition	<ul style="list-style-type: none"> - Le responsable doit être connecté. - La tâche doit lui être assignée.

Scenario nominal	<ol style="list-style-type: none"> 1. L'employé accède à sa liste de tâches. 2. Il sélectionne la tâche concernée. 3. Il modifie le statut de la tâche (ex : en cours, terminée, en attente). 4. Il valide la mise à jour. 5. Le système enregistre le nouveau statut de la tâche.
Scenario alternatif	<p>[A] : Problème de mise à jour</p> <ul style="list-style-type: none"> • Si le système rencontre une erreur technique, il affiche un message d'erreur. • L'employé peut réessayer l'opération.
Postcondition	Le nouveau statut de la tâche est pris en compte dans la gestion du projet.

4. Spécifications techniques

a) Mise à disposition des conditions de travail

Dans le cadre stage au sein de DIGITECH BURKINA, un ordinateur performant et adapté a été mis à notre disposition. Nous avons également bénéficié de locaux confortables et accessibles en journée, favorisant un environnement de travail serein et propice à la concentration.

Ces conditions matérielles nous ont permis de mener à bien nos activités, en assurant une meilleure organisation et une plus grande efficacité dans la réalisation de nos missions.

La mise à disposition de ces ressources a contribué à créer un cadre favorable à la planification et à l'exécution du projet. Passons maintenant à l'examen de l'architecture de développement qui joue un rôle central dans la réalisation de notre projet.

b) Architecture de développement

L'architecture logicielle définit l'organisation et les interactions entre les divers composants d'une application. Elle a un impact sur la stabilité, l'efficacité et la pérennité du logiciel. Une conception architecturale inadéquate peut entraîner des vulnérabilités dans certaines applications, ne correspondant pas au contexte. Les architectures les plus courantes sont celles à deux (2) et trois (3) niveaux. Dans cette présentation, nous explorerons ces diverses architectures et justifierons notre choix en fonction des caractéristiques de chacune. Un tableau comparatif ci-dessous en s'inspirant des travaux existants (voir les rapports dans la bibliographie) mettra en opposition ces deux architectures :

□ **Architecture 2-tiers**

Description	Avantages	Inconvénients
<p>L'architecture deux (2) tiers encore appelée architecture client/serveur de première génération peut être séparée en deux types :</p> <ul style="list-style-type: none"> ❖ Un client, responsable de la gestion de la présentation et de la logique applicative, assure la prise en charge des aspects visuels et des fonctionnalités de l'application ; ❖ Un serveur, quant à lui, se charge de stocker les données de manière cohérente et peut également contenir une portion de la logique applicative, en plus de garantir la gestion et la conservation des informations de manière fiable. <p>Dans ce modèle, la communication utilise le langage SQL entre les deux entités. L'architecture</p>	<p>L'architecture en deux (2) tiers offre les avantages suivants :</p> <ul style="list-style-type: none"> ✓ Elle permet une utilisation multi-utilisateur de l'application, autorisant plusieurs utilisateurs à accéder à la même base de données ; ✓ Elle répartit les charges entre les machines, où le client est responsable de l'interface graphique et des opérations sur les données, tandis que le serveur se charge de la recherche des données pour répondre aux requêtes des clients ; ✓ Elle garantit une solide sécurité des données grâce à des mécanismes de sécurité intégrés au système de gestion de la base de données. 	<p>L'architecture en deux (2) tiers présente les inconvénients suivants :</p> <ul style="list-style-type: none"> ⊗ L'installation et la configuration de l'application sur l'ensemble des postes clients engendrent des coûts de déploiement élevés ; ⊗ Mettre à jour l'application requiert un redéploiement sur tous les postes clients, ce qui génère des coûts supplémentaires et rend l'évolution du système complexe ; ⊗ Complexité de la maintenance, la maintenance de l'application sur tous les postes clients peut devenir fastidieuse et nécessite une coordination rigoureuse pour garantir que toutes les versions soient à jour.

démontre une indépendance du client par rapport au serveur. Chaque poste client doit avoir l'application installée, tandis que la base de données est centralisée et hébergée sur un serveur dédié. La figure ci-dessous représente cette structure.

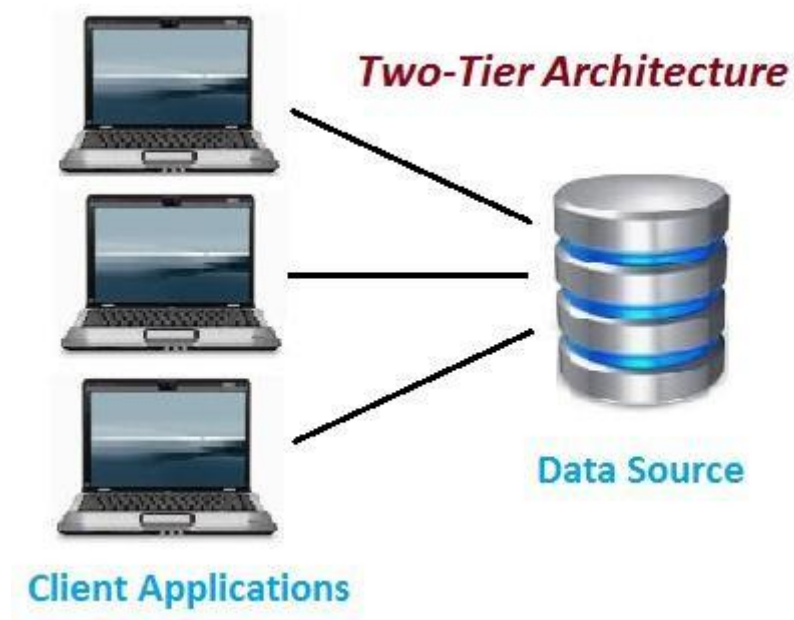


Figure 7: Représentation de l'architecture 2-tiers

Source : <https://www.softwaretestingclass.com/wp-content/uploads/2013/01/two-tier-architecture.png>

□ Architecture 3-tiers

Tableau 10 : Tableau descriptif et comparatif de l'architecture 3-tiers

Description	Avantages	Inconvénients
<p>L'architecture trois (3) tiers est une structure de conception bien établie pour les applications logicielles, qui divise ces applications en trois niveaux logiques et physiques distincts :</p> <ul style="list-style-type: none"> ❖ Le client léger, également connu sous le nom de niveau de présentation, représente l'interface utilisateur et la couche de communication de l'application. C'est ici que l'utilisateur final interagit avec l'application, que ce soit à travers un navigateur Web, une application de bureau ou une interface graphique utilisateur. Son rôle principal est d'afficher des informations à l'utilisateur et de recueillir ses entrées ; ❖ Le serveur d'application, ou niveau application, 	<p>La caractéristique essentielle des applications trois (3) tiers réside dans leur disponibilité élevée et leur adaptabilité aisée :</p> <ul style="list-style-type: none"> ✓ Facilité d'utilisation et évolutivité : Ces applications se distinguent par leur simplicité d'exploitation et leur grande souplesse en matière d'évolution. Les mises à jour sont principalement effectuées au niveau du serveur d'application, sans nécessiter de redéploiement sur les postes clients ; ✓ Déploiement simplifié : L'application est déployée exclusivement sur le serveur d'application. Les clients n'ont besoin que d'un navigateur web compatible avec l'application pour y accéder ; ✓ Sécurité renforcée : ces types d'applications 	<p>Cette technologie est relativement récente et exige un personnel informatique compétent pour sa mise en place. En outre, son utilisation optimale est recommandée au sein d'un réseau haut débit. Parmi les autres éventuels inconvénients à considérer, on peut noter :</p> <p>Complexité de la gestion : La gestion et la configuration des trois niveaux logiques (présentation, application, données) peuvent être plus complexes par rapport à des architectures plus simples ;</p> <p>Coût potentiel : La mise en place initiale d'une infrastructure trois tiers peut nécessiter un investissement en matériel et en logiciel plus important que des solutions plus basiques ;</p> <p>Maintenance requise : la maintenance d'une architecture trois tiers peut être plus exigeante, car il y a plusieurs composants à surveiller et à mettre à jour.</p>

<p>est le cœur de l'application. Dans cette couche, les données collectées au niveau de la présentation sont traitées, parfois en relation avec d'autres données, en utilisant la logique applicative ;</p> <p>❖ Le serveur de base de données, ou niveaux donnés, parfois appelé système central, est l'endroit où les informations traitées par l'application sont stockées et gérées. Cette architecture trois tiers est couramment utilisée pour organiser et concevoir des applications logicielles, car elle offre une séparation claire des préoccupations et permet une évolutivité, une maintenance et une gestion efficaces des applications</p>	<p>bénéficient d'une sécurité élevée des données.</p> <p>L'accès à la base de données est uniquement autorisé par le serveur d'application, contrairement aux applications deux tiers où tous les utilisateurs sont directement connectés à la base de données. Ainsi, la gestion de la sécurité se concentre au niveau du serveur d'application.</p>	
--	---	--

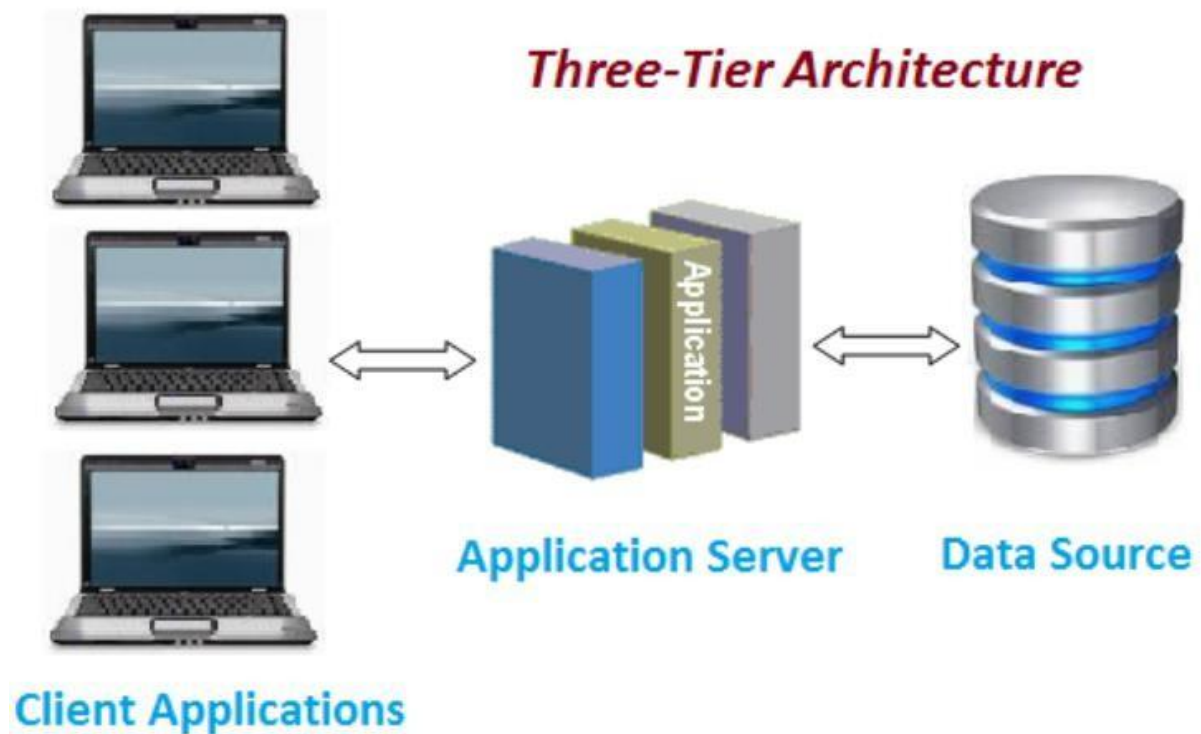


Figure 8 ; Représentation de l'architecture 3-tiers

Source : https://miro.medium.com/max/1104/1*sCV45G9qP1vD8G3_x9Dt7A.png

Choix d'architecture pour notre système

À la suite de la comparaison des architectures 2-tiers et 3-tiers présentée dans le tableau précédent, notre choix s'est porté sur l'architecture trois tiers.

En effet, cette structure s'aligne parfaitement avec les exigences du projet, notamment en matière de séparation des responsabilités, de fiabilité, d'évolutivité et de centralisation des traitements. Elle permet également d'assurer une meilleure organisation du système, une plus grande souplesse dans les mises à jour et une sécurité renforcée des éléments cruciaux dans un contexte de gestion collaborative de projets et de tâches, tel que celui de DIGITECH BURKINA.

Ce choix nous permettra ainsi de concevoir une solution robuste, durable et adaptée aux réalités de l'entreprise.

III. Conception globale

Dans cette partie de notre document nous aborderons les dictionnaires de données ainsi que le diagramme de classe de notre application.

1. Diagramme de classes

a) Dictionnaire de données

Le dictionnaire des données contient l'ensemble des descriptions des attributs des données utilisées dans le système.

Classe : User

Tableau 11 : Dictionnaire de données de la classe User

Attributs	Description	Type de données
id	Identifiant unique de l'utilisateur	Entier
name	Nom de l'utilisateur	Chaîne de caractères
email	Adresse email de l'utilisateur	Chaîne de caractères

Cette classe permet de recueillir les données sur les utilisateurs, notamment leur nom, leur adresse email et leurs relations avec les tâches et projets.

Classe : Project

Tableau 12 : Dictionnaire de données de la classe Project

Attributs	Description	Type de données
id	Identifiant unique du projet	Entier
name	Nom du projet	Chaîne de caractères

description	Brève description du projet	Chaîne de caractères
createdAt	Date de création du projet	Date
UpdatedAt	Date de la dernière mise à jour du projet	Date
inviteCode	Code d'invitation unique permettant de rejoindre le projet	Chaîne de caractères
createdById	Identifiant de l'utilisateur ayant créé le projet	Chaîne de caractères

Cette classe permet de recueillir les données sur les projets, comme le nom, la description, le créateur et les utilisateurs associés.

Classe : Task

Tableau 13: Dictionnaire de données de la classe Task

Attributs	Description	Type de données
id	Identifiant unique du projet	Entier
name	Nom ou titre de la tâche	Chaîne de caractères
description	Description détaillée de la tâche	Chaîne de caractères
status	Statut de la tâche	Chaîne de caractères
dueDate	Date limite de réalisation de la tâche	Date
projectId	Identifiant du projet auquel appartient la tâche	Entier
userId	Identifiant de l'utilisateur assigné à la tâche	Entier
createdById	Identifiant de l'utilisateur ayant créé la tâche	Entier

solutionDescription	Description de la solution apportée à la tâche	Chaîne de caractères
---------------------	---	----------------------

Cette classe permet de recueillir les données sur les tâches, incluant le nom, le statut, le projet lié, l’auteur de la tâche et éventuellement l’utilisateur assigné.

b) Diagramme de classes

Dans cette section, nous présentons le diagramme de classe de notre système. Le diagramme de classe est la représentation visuelle de la structure d'un système, qui met en évidence les classes, leurs attributs et les relations entre elles. Il constitue un élément fondamental de la modélisation orientée objet et permet de visualiser l'architecture statique du système.

Notre diagramme est composé de trois classes principales - **User**, **Project** et **Task** - qui représentent les entités essentielles de notre système. Chaque classe est représentée par un rectangle divisé en sections contenant le nom de la classe et ses attributs. Les relations entre ces classes sont matérialisées par des lignes avec des multiplicités qui définissent le nombre d'instances possibles dans chaque relation.

Ces relations permettent de structurer efficacement l'organisation et la gestion des projets et des tâches au sein du système, offrant ainsi une vision claire des responsabilités et des liens fonctionnels entre les différentes entités.

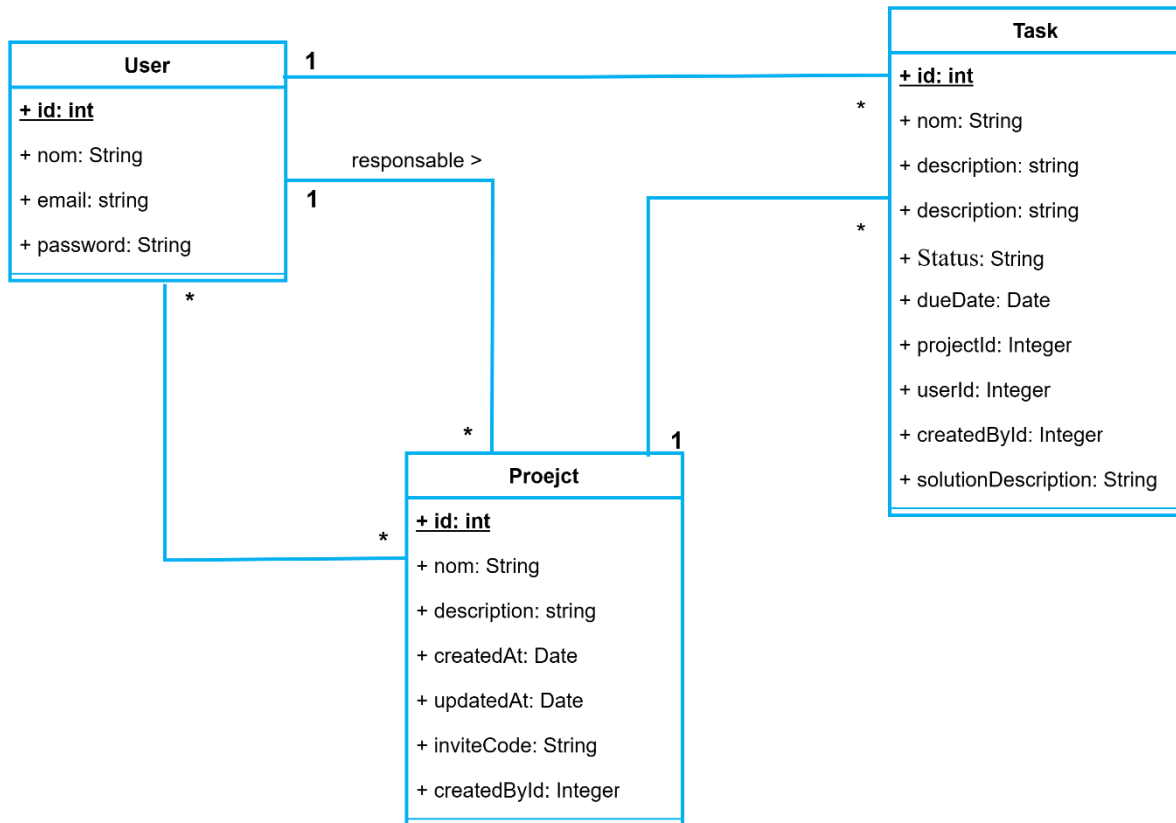


Figure 9 : Diagramme de classes

Ce diagramme montre les relations essentielles entre les entités : un utilisateur peut créer et participer à plusieurs projets, un projet contient plusieurs tâches, et un utilisateur peut créer et être assigné à différentes tâches. Les cardinalités « 1, 0..*, 1..*, * » indiquent précisément le nombre d'instances possibles dans chaque relation, structurant ainsi l'organisation fonctionnelle du système.

Par exemple une instance de la classe **Project (1)** peut contenir plusieurs **Task (*)**, tandis qu'une tâche appartient à un seul projet (1).

Un **User (1)** peut **créer** plusieurs **Project (*)** et **participer** à plusieurs projets (*), *tandis qu'un projet peut avoir plusieurs utilisateurs participants (*)*.

Un **User (1)** peut **créer** plusieurs **Task (*)** et **se voir assigner** plusieurs tâches (*), bien qu'une tâche ne puisse être assignée qu'à un seul utilisateur (1).

2. Diagramme de déploiement

Le diagramme de déploiement est un diagramme UML qui montre la configuration physique des différents éléments qui participent à l'exécution du système, ainsi que les instances de composants qu'ils supportent. Il est constitué de « nœuds » connectés par des liens physiques.

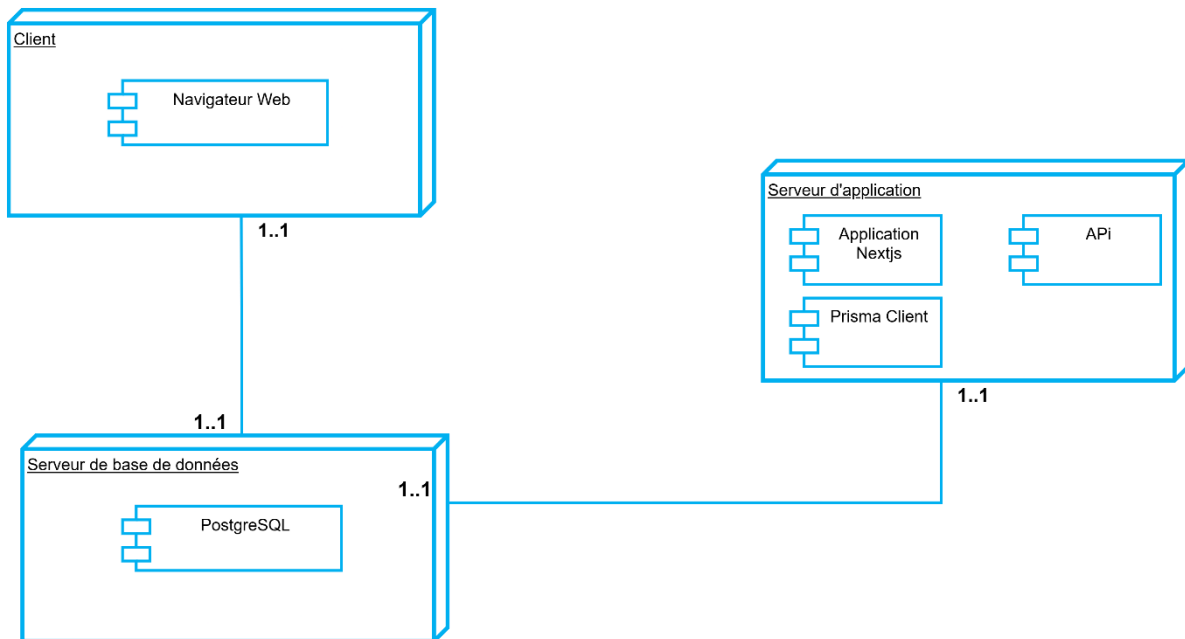


Figure 10 : Diagramme de déploiement

Notre diagramme de déploiement met en évidence une architecture simple mais efficace, composée principalement de trois nœuds : le **poste client** dont le navigateur, le **serveur applicatif** où s'exécute le code métier, et le **serveur de base de données**. Cette structure reflète une approche classique en architecture trois-tiers.

Le serveur applicatif gère les opérations principales de l'application, telles que la gestion des utilisateurs, des projets et des tâches. L'utilisation d'un ORM permet une abstraction efficace de la base de données, facilitant les requêtes complexes, tout en assurant la cohérence des données. Le système inclut également une gestion des autorisations implicite, en vérifiant systématiquement les droits des utilisateurs sur les projets ou les tâches.

Cette organisation offre un bon compromis entre simplicité de mise en œuvre et maintenabilité, tout en permettant une éventuelle montée en charge grâce à la séparation claire entre le backend et la base de données. À la suite de cette présentation de l'architecture de déploiement, nous pouvons désormais passer à la phase de réalisation concrète de l'application.

IV. Réalisation

Dans le cadre de la réalisation de notre projet, nous avons utilisé plusieurs outils et techniques. Ainsi, nous présentons d'abord les outils de développement utilisés et les politiques de test et de sécurité du système. Ensuite, nous présentons quelques écrans de celle-ci. Enfin, nous évoquons l'estimation financière de la mise en place de la plateforme.

1. Présentation des outils développement

Dans le cadre de la réalisation de notre projet, nous avons utilisé plusieurs outils indispensables au développement. Ces outils nous ont permis de concevoir, développer et gérer efficacement notre application. Il s'agit notamment des langages de programmation, des plateformes de développement, des outils de conception, ainsi que du système de gestion de base de données (SGBD).

a) Les langages de programmation

Les langages de développement sont des outils qui permettent de traduire des algorithmes en instructions compréhensibles et exécutables par une machine. Un algorithme représente une suite ordonnée d'opérations logiques destinées à atteindre un objectif précis.

Les langages de programmation que nous avons utilisés sont les suivants :

TypeScript : Le langage de programmation type script est un langage de programmation développé par Microsoft, en surcouche à Java Script. C'est un langage descriptif qui permet d'écrire les scripts orientés utilisateur. Il nous a permis de mettre en place la vue de notre application.

HTML : est un langage de marquage et de balisage servant à écrire des pages pour le World Wide Web. (Pour les pages web)

CSS3 : le CSS permet de faire la mise en forme des pages web.

b) Plateforme de développement (Framework)

Les plateformes de développement, ou *frameworks*, sont des environnements logiciels fournissant des composants réutilisables et des structures prédéfinies qui facilitent et accélèrent la création d'applications. Elles permettent de se concentrer sur les fonctionnalités métier tout en respectant de bonnes pratiques de développement.

Next.js : Next.js est un framework React moderne qui permet de construire des applications web performantes avec rendu côté serveur (SSR) et génération statique. Il a constitué la base de notre application côté client (*frontend*) et a facilité l'organisation du code, la gestion des routes, ainsi que l'intégration avec le backend grâce aux fonctions serverless.

React : Bibliothèque JavaScript utilisée via Next.js pour construire l'interface utilisateur dynamique et réactive de notre application. Elle nous a permis de gérer l'état de l'application et de créer des composants réutilisables pour les vues.

Prisma : Prisma est un ORM (Object-Relational Mapping) moderne pour Node.js qui simplifie l'accès à la base de données. Il a été utilisé pour interagir efficacement avec notre base de données (PostgreSQL), en facilitant la création, la récupération et la gestion des données liées aux utilisateurs, projets et tâches.

Clerk : Clerk est une plateforme d'authentification moderne utilisée dans notre projet pour gérer l'inscription, la connexion et l'authentification des utilisateurs. Elle s'intègre facilement avec Next.js et offre une gestion sécurisée des sessions utilisateur.

Tailwind CSS (via DaisyUI) : Tailwind est un framework CSS utilitaire qui nous a permis de concevoir rapidement une interface esthétique et responsive. Combiné à DaisyUI, une bibliothèque de composants UI basée sur Tailwind, il a facilité la création d'une interface utilisateur cohérente et moderne.

c) Les outils de conception

Draw.io : Outil de modélisation en ligne qui nous a permis de concevoir les différents diagrammes UML (cas d'utilisation, classes, séquence, etc.) nécessaires à la phase d'analyse.

TeamGantt : Utilisé pour l'élaboration du diagramme de Gantt, il nous a permis de planifier les tâches du projet et de suivre l'évolution du développement.

Git : Système de gestion de versions indispensable, il nous a permis de sauvegarder, suivre les modifications, collaborer efficacement en équipe et maintenir un historique propre du code source.

Microsoft Word : Outil de traitement de texte qui nous a servi à la rédaction de la documentation du projet, notamment les rapports, cahiers des charges et présentations fonctionnelles.

Visual Studio Code : Environnement de développement léger et polyvalent, utilisé pour le développement côté frontend, avec le support des extensions TypeScript, React et Tailwind CSS.

d) Système de Gestion de Base de données (SGBD)

Nous distinguons plusieurs SGBD dont les plus connus sont :

- ❖ Microsoft Office Access;
- ❖ MySQL;
- ❖ Microsoft SQL Server;
- ❖ Oracle Database ;
- ❖ PostgreSQL.

Pour un choix plus raisonnable et adapter à notre besoin, nous ferons une étude comparative de ces SGBD dans le tableau suivant :

Tableau 14 : Tableau de comparaison de SGBD

SGBD	Avantages	Inconvénients
------	-----------	---------------

Microsoft Office Access	<ul style="list-style-type: none"> ✓ Très puissant et très ludique ✓ Une grande série d'outils de conversion de données ▪ Une forte intégration de ✓ Microsoft Office/VBA 	<ul style="list-style-type: none"> ⊗ Prix élevé ⊗ Une gestion des verrous mortels mal conçue ⊗ Une administration complexe ⊗ L'architecture présente de nombreuses failles de sécurité intrinsèques
MySQL	<ul style="list-style-type: none"> ❖ Rapide et simple à utiliser ❖ Multiplateforme ❖ (Windows, Unix, Linux, ❖ etc.), gratuit et facile à utiliser ❖ S'intègre facilement dans l'environnement Apache 	<ul style="list-style-type: none"> ⊗ Non convenable aux grosses bases de données ⊗ Les fonctionnalités limitées ⊗ Présente une sécurité moyenne ⊗ Ne permet pas la reprise à chaud
Microsoft SQL Server	<ul style="list-style-type: none"> ❖ Administration aisée ❖ Fonction d'audit évolué ❖ Optimiseur statistique enrichi à flux tendu ❖ Langage T-SQL très convivial, intégration de ❖ CLR ❖ Services Web ❖ Support XML ❖ Ordonnanceur intégré 	<ul style="list-style-type: none"> ⊗ Une licence très couteuse du produit ⊗ Toujours pas de cluster (hormis en actif passif, en se basant sur le cluster OS) ⊗ Pas certifié SQLJ, pas d'intégration Java,

		<ul style="list-style-type: none"> ⊗ orientation C# et pas de contraintes d'unicité multi nulle
Oracle	<ul style="list-style-type: none"> ❖ Fourniture de technologies de haut niveau et des solutions d'affaires intégrées. ❖ Fiabilité et intègre la technologie flashback. ❖ Récupération efficace des données incorrectement supprimées ou perdues grâce à flashback d'Oracle. ❖ Fournit des bases de données fiables et compétentes grâce aux quatre propriétés (Atomicité, cohérence, isolation et durabilité). 	<ul style="list-style-type: none"> ⊗ Prix élevé ⊗ Une administration complexe ⊗ Porosité entre les schémas ⊗ Une gestion des verrous mortels mal conçue ⊗ Nombreuses failles de sécurités liées à l'architecture elle-même
PostgreSQL	<p>Open Source, gratuit, fiable</p> <p>Supporte la majorité du standard SQL-92</p> <p>Possède de nombreuses fonctionnalités d'extensions (Java, Ruby, PL-SQL)</p> <p>Très riche fonctionnellement</p> <p>Simple d'utilisation et d'administration</p>	<p>Pas possible de requêter sur plusieurs bases à la fois</p> <p>Sauvegardes peu évoluées</p> <p>Pas de services Web</p> <p>Pas d'ordonnanceur intégré</p> <p>Pas de fonctions d'agrégat OLAP</p> <p>Pas de requêtes récursives</p>

Suite à la comparaison effectuée, notre choix s'est porté sur **PostgreSQL** en raison de sa fiabilité, de sa richesse fonctionnelle et de sa gratuité, ce qui en fait une solution idéale pour notre projet.

Serveur d'application

Un serveur d'application joue un rôle central dans une architecture logicielle. Il exécute la logique métier, assure la sécurité, communique avec la base de données et expose des services pour permettre à l'interface utilisateur d'accéder aux données. Plusieurs serveurs d'application existent sur le marché tel que Nginx , Wildfly, Oracle WebLogic server, Apache , chacun avec ses avantages selon les cas d'usage.

Tableau 15: Tableau comparatif des servers d'application

Serveur d'application	Supports HTTPS	Console d'administration	Communauté	Prix	Plateformes supportées
Nginx	Disponible	Via Plugin	Moyenne	Gratuit	Linux, Unix, MacOS, Windows, BSD, Raspbian
Wildfly	Disponible	Intégré	Forte	Gratuit	Windows, Linux, MacOS
Oracle WebLogic Server ;	Disponible	Intégré	Moyenne	Licence payante	Windows, Linux,
Serveur d'application	Supports HTTPS	Console d'administration	Communauté	Prix	Plateformes supportées

					Solaris, macOS
Apache	Disponible	Intégré	Moyenne	Gratuit	Windows, Linux, Solaris, macOS

Choix retenu :

Plutôt que d'opter pour un serveur d'application traditionnel, nous avons fait le choix de **Next.js**¹, un framework JavaScript moderne basé sur **Node.js**², qui permet d'unifier le frontend et le backend dans un seul projet. Ce choix s'explique par plusieurs avantages :

- ❖ Développement d'API REST directement intégrées via le dossier /api dans la structure des fichiers du framework.
- ❖ Rendu côté serveur appelé SSR et génération statique et SSG possibles dans un même environnement.
- ❖ Grande flexibilité et écosystème riche comme React, TypeScript, Clerk, Tailwind, et toutes les technologies modernes.
- ❖ Déploiement CI/CD facile sur des plateformes modernes comme Vercel, Node.js, Docker.

2. Présentation de l'architecture MVC de notre application

¹ <https://nextjs.org/>

² <https://nodejs.org/>

L'architecture MVC (Modèle – Vue – Contrôleur) est un paradigme de développement logiciel structurant, très utilisé dans la conception d'applications web. Elle permet de séparer les responsabilités au sein de l'application en trois composants distincts :

❖ Modèle (Model)

Le modèle représente la couche métier de l'application. Il contient :

- Les données manipulées par l'application.
- La logique métier, c'est-à-dire les règles de gestion.
- La gestion de la persistance, via les accès à la base de données ou à des services externes.
- Le modèle est indépendant de l'interface utilisateur : il ne se préoccupe pas de l'affichage.

❖ Vue (View)

La vue est en charge de l'affichage des données.

- Elle présente visuellement les informations du modèle.
- Elle se met à jour automatiquement lorsqu'il y a un changement dans le modèle.
- Elle capture les interactions utilisateur sans logique métier, et les transmet au contrôleur.

❖ Contrôleur (Controller)

Le contrôleur fait l'interface entre la vue et le modèle.

- Il traite les actions de l'utilisateur.
- Il modifie le modèle en fonction des besoins (ajout, mise à jour, suppression).
- Il définit la vue qui doit être affichée après chaque interaction.

Ainsi, l'utilisateur interagit avec la Vue, qui transmet l'action au Contrôleur. Celui-ci agit sur le Modèle, puis la Vue est mise à jour avec les nouvelles données. Ce fonctionnement assure une architecture claire et modulaire.

❖ Avantages de l'architecture MVC

Le choix de cette architecture s'est imposé naturellement pour notre plateforme en raison des nombreux avantages qu'elle offre :

- Séparation des préoccupations : chaque couche a un rôle bien défini, rendant le code plus clair et maintenable.
- Réutilisation du modèle : les mêmes données peuvent être affichées dans plusieurs interfaces.
- Testabilité : chaque composant peut être testé indépendamment.

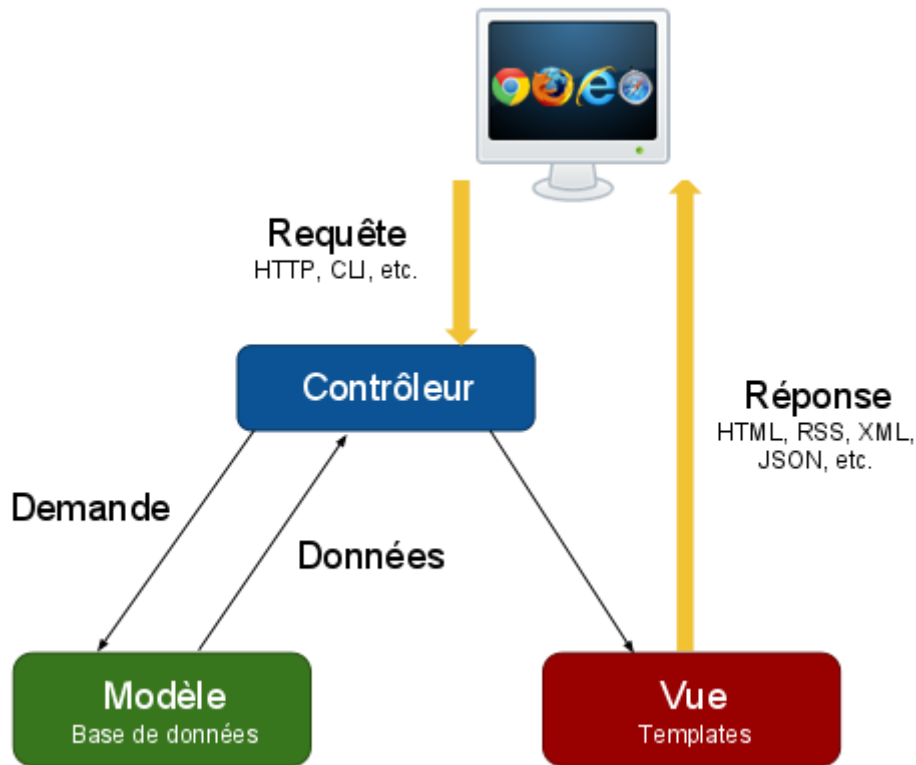


Figure 11: Schéma d'illustration de l'architecture MVC

Source : <https://raphaelbellon.wordpress.com/wp-content/uploads/2017/03/mvc.png>

3. Quelques interfaces de notre application

Dans cette section, nous faisons la description de quelques interfaces du système de gestion des projets et de tâche collaborative de DIGITECH BURBINA.

❖ Page de connexion

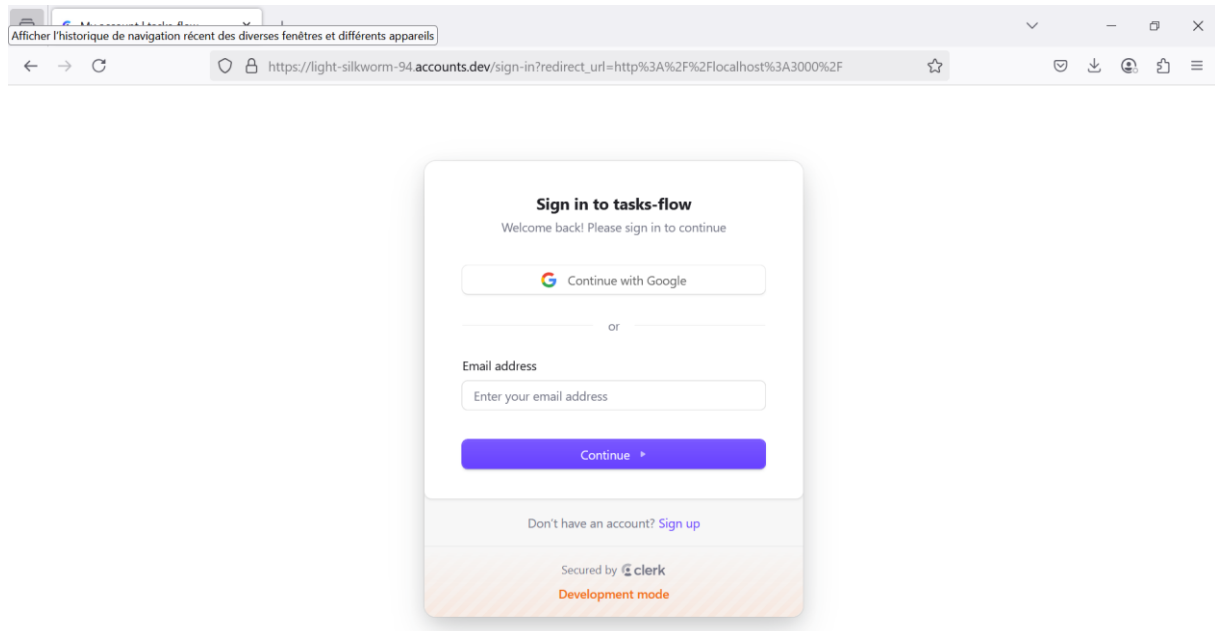


Figure 12 : Interface de connexion de TaskFlow

Un utilisateur à la possibilité de se connecter directement avec son compte google

❖ Page d'accueil

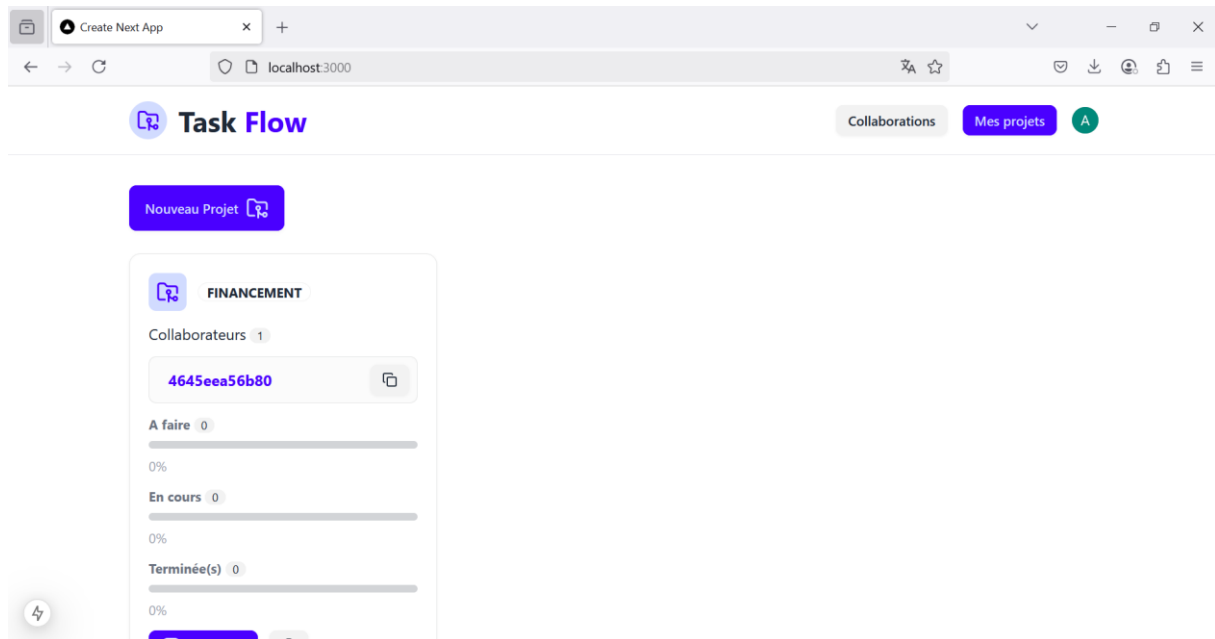


Figure 13 : Page d'accueil de TaskFlow

Une fois l'utilisateur connecter il perçoit automatiquement les projet auquel il est responsable et supervise

❖ Formulaire de création d'un nouveau projet

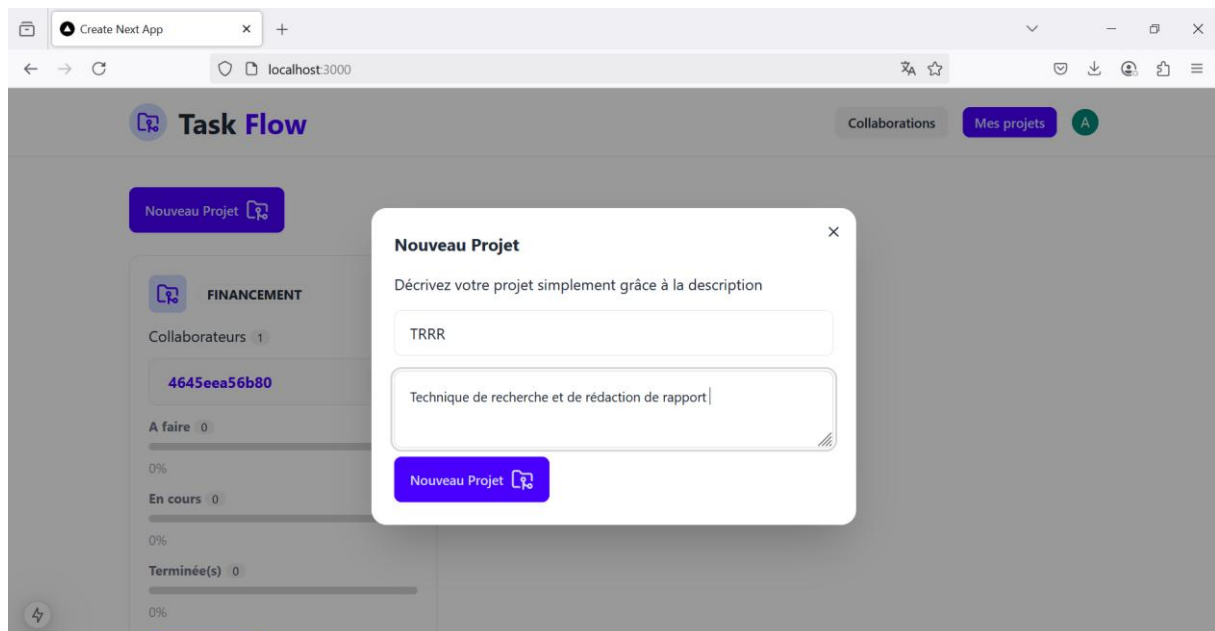


Figure 14 : Interface du formulaire de création d'un projet

❖ Interface de création de tâche et d'assignation

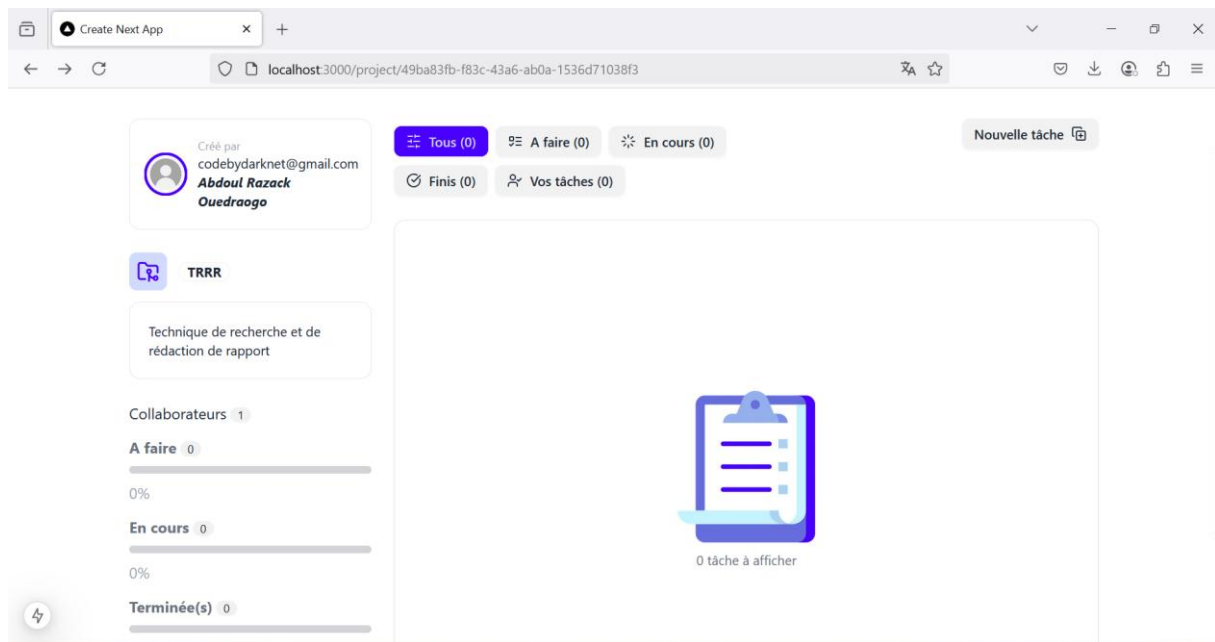


Figure 15 : Interface des différentes tâches associées à un projet

Cet écran permet de voir les tâches qui sont disponibles dans le projet ainsi que d'ajouter directement une tâche au projet

❖ Interface de création de tâche et d'assignation de tâche

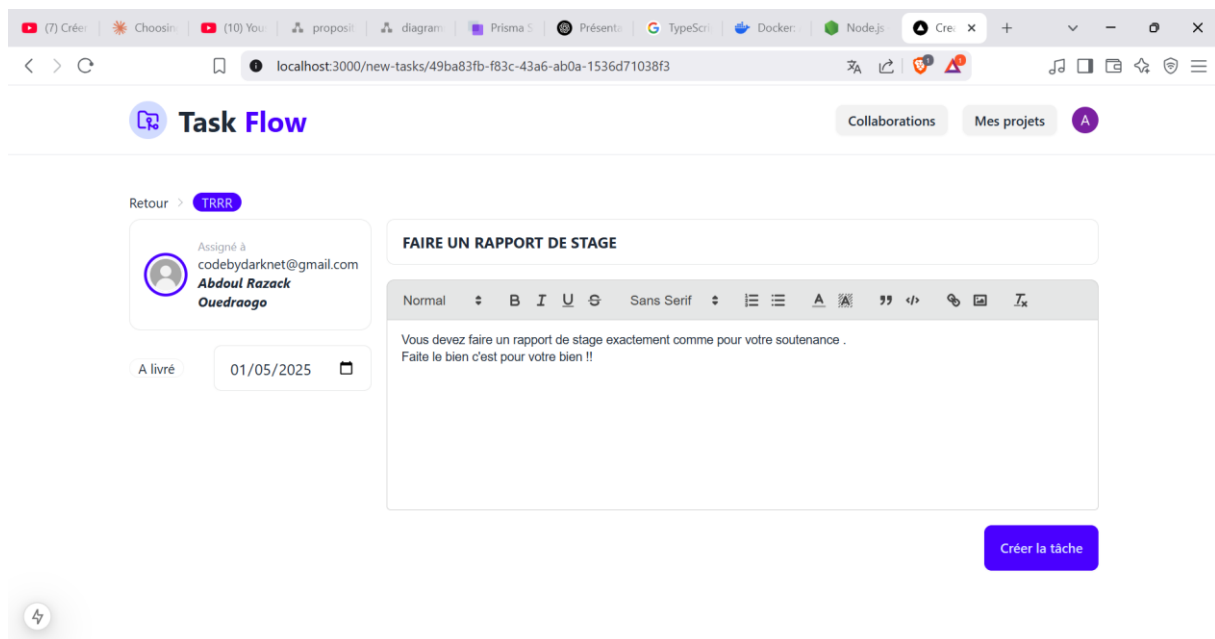


Figure 16 : interface de création et d'assignation de tâche

Ici on entre le nom de la tâche et les actions à mener, on le dédié a un utilisateur soit même.

❖ Interface de traitement de la tache

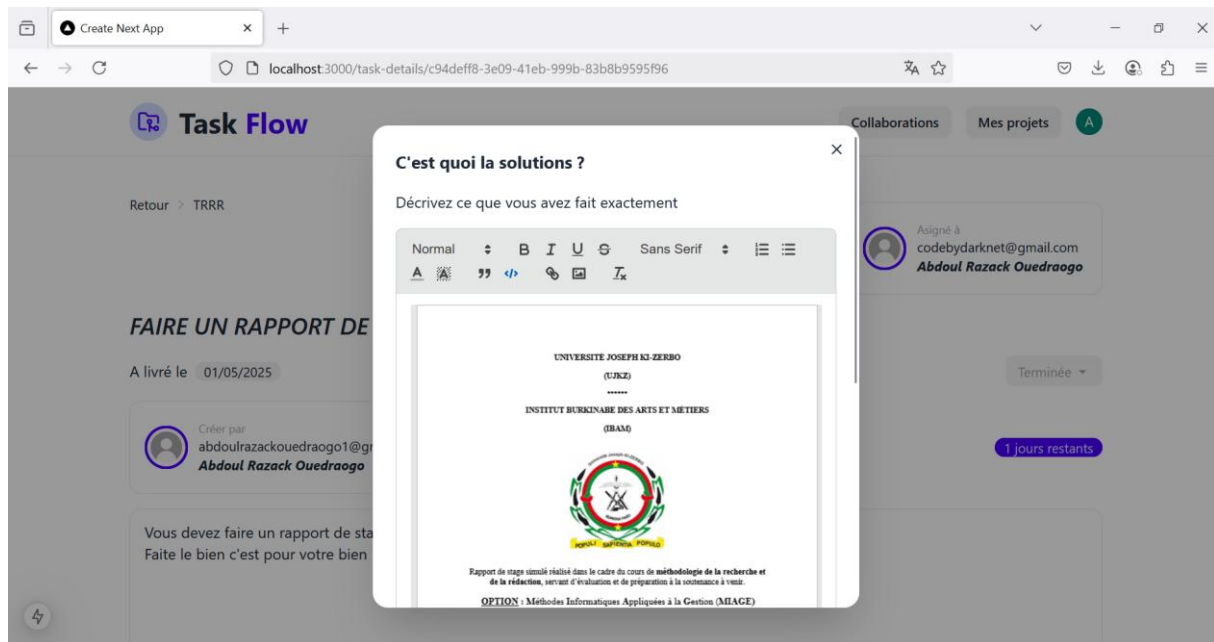


Figure 17: Interface de traitement d'une tache et apport de solution

La tâche *est traitée* et son statut change

❖ Interface des taches effectuer

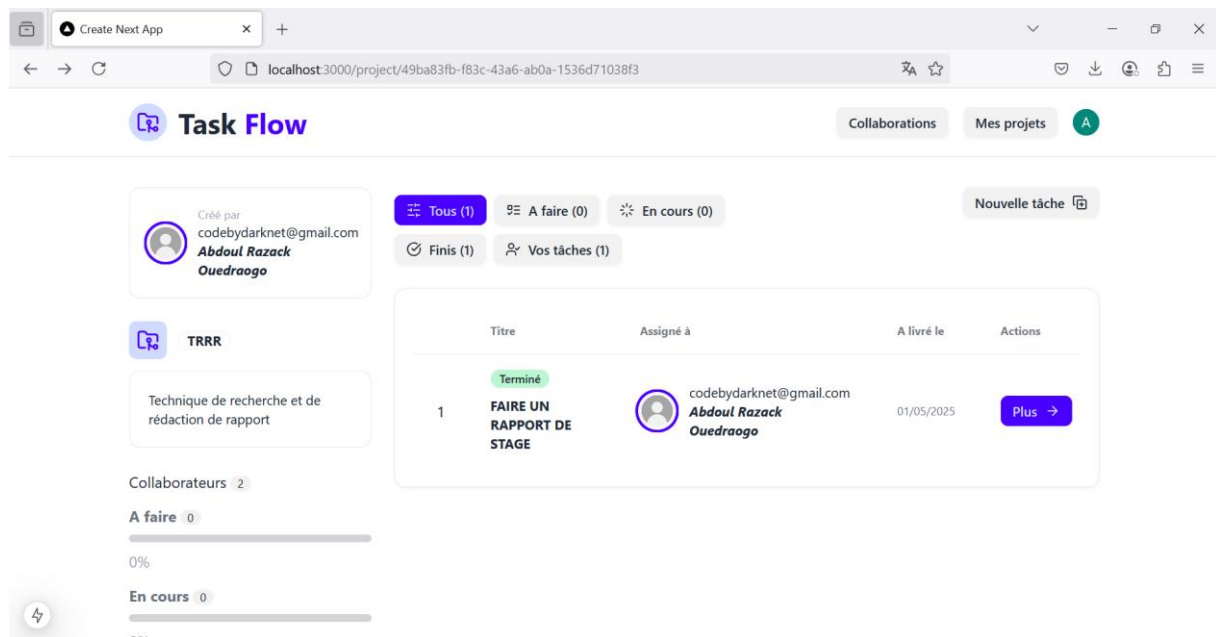


Figure 18: Interface de tâche effectuer avec un bon statut

Après avoir présenté quelques interfaces de notre plateforme, nous allons dans la suite faire des tests pour s'assurer du bon fonctionnement.

4. Politique de sécurité

La politique de sécurité désigne l'ensemble des moyens à mettre en place pour protéger le logiciel. La sécurité de notre plateforme représente un aspect essentiel à prendre en compte. Cette politique s'articule autour de la sensibilisation des utilisateurs, des mesures de sécurité mises en œuvre dans le processus de développement de la plateforme et des mesures à implémenter lors de sa mise en production.

❖ Sensibilisation des utilisateurs

Le facteur humain étant un élément important dans la démarche sécuritaire, il est nécessaire de fournir à nos utilisateurs des conseils pratiques pour une utilisation sécurisée de la plateforme. Ainsi, l'utilisateur peut se prémunir de plusieurs risques en adoptant les démarches suivantes :

- ✓ Éviter de communiquer ses données d'authentification (email et mot de passe) à un tiers
- ✓ Ne pas laisser sa session ouverte sur un poste de travail sans surveillance
- ✓ Utiliser des mots de passe forts et uniques pour accéder à la plateforme
- ✓ Vérifier régulièrement les membres associés à ses projets pour s'assurer qu'aucun utilisateur non autorisé n'y a accès
- ✓ Se déconnecter après chaque utilisation de la plateforme, surtout sur des postes partagés

❖ Mesures de sécurité implémentées pour le développement

➤ Gestion des accès et authentification

L'accès à l'application est protégé par un système d'authentification basé sur l'email de l'utilisateur. Notre application implémente un système de contrôle d'accès qui assure qu'un utilisateur ne peut exécuter que les fonctionnalités auxquelles son profil lui donne droit. En particulier :

- ✓ Seul le créateur d'un projet peut le supprimer
- ✓ Un utilisateur ne peut accéder qu'aux projets auxquels il est associé
- ✓ Un système de code d'invitation unique permet de contrôler l'ajout de membres à un projet

➤ Sécurisation des données

Pour garantir la sécurité des données manipulées par l'application :

- ✓ Les codes d'invitation sont générés de manière aléatoire et sécurisée ;
- ✓ Toutes les opérations de base de données sont réalisées à travers l'ORM Prisma, qui prévient les attaques par injection SQL ;
- ✓ Des vérifications systématiques sont effectuées avant chaque opération critique pour s'assurer que l'utilisateur a les droits nécessaires.

➤ Gestion des erreurs

Notre application implémente une gestion robuste des erreurs qui permet :

- De tracer les problèmes rencontrés via des messages d'erreur détaillés ;
- De protéger les informations sensibles en ne renvoyant pas les détails techniques des erreurs aux utilisateurs

- De valider les données d'entrée avant leur traitement pour éviter les injections et les erreurs de traitement

❖ Mesures de sécurité à prendre pour l'exploitation

➤ Sécurité SSL/TLS

Pour notre système, nous préconisons l'utilisation des protocoles sécurisés comme TLS (Transport Layer Security) afin d'assurer des fonctions de sécurité très élevées. Cette technologie est une norme dans le domaine de la sécurisation des transactions Internet. Le certificat TLS va être utilisé pour crypter l'ensemble des échanges entre les postes clients et le serveur hébergeant notre plateforme. En résumé, le certificat TLS assure :

- ✓ La confidentialité et l'intégrité des données échangées
- ✓ L'authentification du serveur, garantissant aux utilisateurs qu'ils accèdent bien à la plateforme légitime
- ✓ La protection contre l'interception des données sensibles, notamment les codes d'invitation aux projets

➤ Politique de sauvegarde des données

La politique de sauvegarde consiste à prendre des mesures nécessaires pour préserver l'intégrité des données en cas de dysfonctionnement du système. Pour notre plateforme de gestion de projets et de tâches collaborative, nous préconisons :

- ✓ Des sauvegardes journalières de la base de données qui seront conservées pendant une semaine
- ✓ Des sauvegardes hebdomadaires qui seront conservées pendant un mois
- ✓ Des sauvegardes mensuelles qui seront conservées pendant six mois
- ✓ Des sauvegardes semestrielles qui seront conservées pendant un an
- ✓ Des sauvegardes annuelles qui seront conservées définitivement

Ces sauvegardes concerneront principalement :

- ✓ Les données des utilisateurs
- ✓ Les informations sur les projets et leurs codes d'invitation
- ✓ Les tâches associées aux projets avec leurs statuts et descriptions
- ✓ Les relations entre utilisateurs et projets
- ✓ Des audits de sécurité périodiques pour identifier les vulnérabilités potentielles

- ✓ La mise en place d'un environnement de test permettant de valider les mises à jour avant leur déploiement en production

5. Estimation du coût de développement

La conception d'un logiciel inclut le plus souvent une phase très importante qui est l'estimation du coût de réalisation. En effet, cette opération permet de connaître les ressources humaines, financières et matérielles nécessaires pour mener à bien le projet. Il existe plusieurs méthodes pour faire cette estimation. Nous utiliserons la méthode COCOMO SIMPLIFIEE (Constructive Cost Model) pour la fiabilité de ces estimations. Les informations concernant cette méthode sont détaillées en Annexe ([ANNEXE 3 : La méthode COCOMO](#)).

De plus, nous estimons le nombre de lignes de code de notre application à neuf mille (9 000). En supposant que le salaire d'un Ingénieur-informaticien est égal à 200 000 F CFA, on peut alors estimer le coût de développement comme suit :

- $Effort = 2,4 * (9000/1000)^{1,05} = 24.10 \text{ Homme/Mois}$
- $TDEV = 2,5 * Effort^{0,38} = 2,5 * 24.10^{0,38} = 8.37 \text{ mois}$
- $CDEV = 24.10 * X \text{ FCFA} = 24.10 * 200\,000 \text{ FCFA} = 4\,820\,000 \text{ FCFA}$

En prenant en considération le coût du développement ainsi que celui du besoin en matériel informatique et logiciels, on obtient le coût total du projet tel que présenté comme suit :

Tableau 16 : Tableau d'estimation du coup de notre projet

Désignation	Quantité	Coût unitaire (en FCFA)	Disponibilité ou coût total (en FCFA)
Ordinateur portable	1	300 000	300 000
PostgreSQL	1	-	Licence gratuite

Visual Studio Code	1	-	Licence gratuite
Développement	-	-	4 820 000
Coût Total			5120000

CHAPITRE III : BILAN DU STAGE

Dans ce chapitre de notre document, nous aborderons le bilan de notre projet de développement. Pour ce faire, nous présenterons d'une part, le déroulement de notre projet ainsi que les activités menées et d'autre part, nous ferons des suggestions et des observations.

I. DÉROULEMENT DU STAGE ET ACTIVITÉS MENÉES

1. Activités menées

Durant la période de développement, notre activité était essentiellement basée sur l'étude du thème « **ANALYSE ET CONCEPTION D'UNE PLATEFORME INTÉGRÉE DE GESTION DE TÂCHES ET DE PROJETS COLLABORATIFS : CAS DE DIGITECH BURKINA** » suivant le planning de réalisation du projet présenté plus haut.

Nous avons effectué entre autres les activités suivantes :

- ❖ L'analyse approfondie des besoins de DIGITECH BURKINA en matière de gestion de projets
- ❖ La conception de l'architecture de la plateforme basée sur les technologies modernes
- ❖ Le développement des fonctionnalités de gestion des utilisateurs avec système d'authentification
- ❖ L'implémentation du système de gestion des projets avec codes d'invitation uniques
- ❖ Le développement du module de gestion des tâches avec attribution et suivi d'état
- ❖ La mise en place d'un système de sécurité robuste
- ❖ Les tests et déploiement de la solution

2. Connaissances acquises et compétences développées

Ce stage effectué au sein de l'entreprise DIGITECH BURKINA nous a été bénéfique à plusieurs niveaux. Il nous a permis de :

- ❖ Mettre en pratique les connaissances académiques engrangées durant les trois (03) ans de formation à l'IBAM ;
- ❖ Renforcer nos compétences en développement d'application web ;
- ❖ Développer nos compétences en sécurisation des données avec l'implémentation de mécanismes comme la génération de codes d'invitation sécurisés
- ❖ S'imprégner des réalités de la vie professionnelle

II. OBSERVATIONS ET SUGGESTIONS

Nous adressons nos remerciements à OKABI Technology Services pour le stage qu'il nous a accordé et qui a été vraiment bénéfique pour nous. Nous apprécions à sa juste valeur l'accueil chaleureux dont nous avons bénéficié de la part du personnel qui par ailleurs, a mis à notre disposition des locaux agréables et conviviaux pour le travail. De plus, avec OKABI Technology Services, l'effort est toujours récompensé. Ainsi, nous avons bénéficié d'une rémunération mensuelle.

Au regard des résultats palpables auxquels nous sommes parvenus, nous suggérons à OKABI Technology Services l'organisation de sessions de formations ouvertes et de hackathons afin de détecter des talents qu'elle pourra récupérer pour booster son équipe de développement. Nous encourageons OKABI Technology Services à continuer dans son élan et à être encore plus porteur de valeurs et de bienfaits.

CONCLUSION GÉNÉRALE

BIBLIOGRAPHIE ET WEBOGRAPHIE

1. Bibliographie

- ❖ [1] **SAWADODO Rimalguedo Rahimata (2023)**, « Conception Et Réalisation D'une d'une plateforme de gestion integree et de suivi des startups incubees à SIRA LAB » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 58p ;
- ❖ [2] **SAWADOGO Patounizambo Yves (2024)**, « ANALYSE ET CONCEPTION D'UNE PLATEFORME INTEGRÉE DE REGISTRE ELECTRONNIQUE DE VACCINTION » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 80p ;
- ❖ [3] **OUEDRAOGO P. David (2021)**, « Mise en place d'un système de communication base sur les sms pour les professionnels et les particuliers » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 64p ;
- ❖ [4] **Ismaël Faïssal GUIRE**), « 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), 84p ;

2. Webographie

- ❖ [1] Next.js sur [Next.js - Documentation officielle](#), Framework React pour le développement web, page consultée quotidiennement durant toutes les phases du développement.
- ❖ [2] Prisma sur [Prisma - Documentation officielle](#), ORM pour l'interaction avec la base de données PostgreSQL, consultée régulièrement pour la modélisation et les requêtes.
- ❖ [4] React sur [React - Documentation officielle](#)
Bibliothèque JavaScript pour la création d'interfaces utilisateur, consultée pour les composants et les hooks.
- ❖ [5] TypeScript sur [TypeScript - Documentation officielle](#), Langage de programmation typé basé sur JavaScript, consulté pour les types et les interfaces.
- ❖ [6] GitHub [GitHub](#), Plateforme de gestion de code source, utilisée pour le versionnage de notre code.
- ❖ [7] Stack OverFlow [Newest Questions - Stack Overflow](#), Forum en ligne, consultée plusieurs fois durant le développement, la résolution des bugs de notre plateforme
- ❖ [8] Youtube [YouTube](#), plateforme de streaming de média, page consultée plusieurs fois durant le développement.
- ❖ [9] Youtube [YouTube](#), plateforme de streaming de média, page consultée plusieurs fois durant le développement.
- ❖ [10] OpenAI GPT-4o sur [ChatGPT](#), modèle d'IA développé par OpenAI. Page consulté plusieurs fois pendant le développement.

- ❖ [11] **Claude** sur [Claude.ai](https://claude.ai), Modèle d'IA développé par Anthropic. Consulté plusieurs fois pendant le développement pour améliorer la structure du code et résoudre des problèmes complexes.

Table des matières

SOMMAIRE	i
DEDICACE.....	ii
REMERCIEMENT	iii
LISTE DES SIGLES ET ABREVIATIONS	iii
LISTE DES FIGURES ET GRAPHIQUES.....	iii
LISTE DES TABLEAUX.....	iii
INTRODUCTION GENERALE.....	1
CHAPITRE I : PRESENTATION DES STRUCTURES DE FORMATIONS ET D'ACCUEIL	2
I. Présentation de la structure de formation	2
1. Historique	2
2. Objectif.....	2
3. Organisation	2
a) Organe statuaire.....	3
b) Organe d'exécution	4
4. Filières de formation	5
a) Licences professionnelles.....	5
b) Master.....	5
II. Présentation de la structure d'accueil.....	6
1. Historique	6
2. Mission	7
3. Fonctionnement.....	7
4. Projets et partenaires	8
5. Organisation	9
CHAPITRE II : ANALYSE ET CONCEPTION.....	10
I. Étude préalable.....	10
1. Présentation du thème	10
a) Problématique.....	10
b) Objectifs	11
c) Résultats attendus.....	11
2. Méthodologie	12
a) Cycle de développement	12
b) Langage de modélisation.....	17
3. Présentation du groupe de travail	20
a) Groupe de pilotage	20

b)	Groupe des utilisateurs	20
c)	Groupe de projet.....	21
d)	Planning de travail.....	21
II.	Expression des besoins.....	22
1.	Étude de l'existant.....	22
a)	Présentation du système utilisé à DIGITECH BURKINA.....	22
b)	Exigences fonctionnelles.....	23
c)	Présentation d'outils logiciels de gestion de projet existants	24
2.	Présentation du processus de fonctionnement amélioré.....	25
3.	Spécification fonctionnelle.....	28
a)	Indentification des acteurs.....	28
b)	Les cas d'utilisation.....	28
c)	Le diagramme de cas d'utilisation	31
d)	Description textuelle de certains cas d'utilisation.....	34
4.	Spécifications techniques	40
a)	Mise à disposition des conditions de travail.....	40
b)	Architecture de développement.....	41
III.	Conception globale.....	47
1.	Diagramme de classes	47
a)	Dictionnaire de données	47
b)	Diagramme de classes	49
2.	Diagramme de déploiement	50
IV.	Réalisation.....	52
1.	Présentation des outils développement	52
a)	Les langages de programmation.....	52
b)	Plateforme de développement (Framework)	53
c)	Les outils de conception	53
d)	Système de Gestion de Base de données (SGBD)	54
2.	Présentation de l'architecture MVC de notre application	58
3.	Quelques interfaces de notre application	60
4.	Politique de sécurité	65
5.	Estimation du coût de développement	68
CHAPITRE III : BILAN DU STAGE		70
I.	DÉROULEMENT DU STAGE ET ACTIVITÉS MENÉES.....	70
1.	Activités menées	70
2.	Connaissances acquises et compétences développées.....	70
II.	OBSERVATIONS ET SUGGESTIONS	71

CONCLUSION GÉNÉRALE	72
BIBLIOGRAPHIE ET WEBOGRAPHIE	I
1. Bibliographie	I
2. Webographie	II
Table des matières	IV
ANNEXES	VII
I. ANNEXE 1 : Langage UML	VII
II. ANNEXE 2 : Méthode COCOMO	VIII

ANNEXES

I. ANNEXE 1 : 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 ou 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)
S	$\text{Effort} = 2,4 (\text{KLS})^{1,05}$	$\text{TDev} = 2,5 (\text{Effort})^{0,38}$
P	$\text{Effort} = 3 (\text{KLS})^{1,12}$	$\text{TDev} = 2,5 (\text{Effort})^{0,35}$
E	$\text{Effort} = 3,6 (\text{KLS})^{1,2}$	$\text{TDev} = 2,5 (\text{Effort})^{0,32}$

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

DS - Enregistrer Facture Client

