

UNIVERSITE JOSEPH KI ZERBO

(UJKZ)

INSTITUT BURKINABE DES ARTS ET METIERS

(IBAM)



RAPPORT DE STAGE POUR L'OBTENTION DE LA LICENCE

PROFESSIONNELLE

OPTION : Méthodes Informatiques Appliquées à la Gestion (MIAGE)

Période de stage : 01 Juillet au 30 Septembre 2023

THEME :

**CONCEPTION ET REALISATION D'UNE PLATEFORME DE GESTION
INTEGREE ET DE SUIVI DES STARTUPS INCUBEES A SIRA LABS.**

Présenté par Rimalguedo Rahimata SAWADOGO

Maître de stage :

M. Laurent NIKIEMA

Consultant développeur web/data

à Sira Labs

Directeur de rapport :

Dr Ferdinand GUINKO

Maitre de Conférences en

Informatique à l'IBAM

Année académique 2022-2023

SOMMAIRE

SOMMAIRE	ii
DEDICACE.....	iii
REMERCIEMENTS	iv
LISTES DES SIGLES ET ABBREVIATIONS.....	v
LISTE DES FIGURES GRAPHIQUES.....	vi
LISTE DES TABLEAUX.....	vii
INTRODUCTION GENERALE	1
CHAPITRE I : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL	2
I. PRESENTATION DE LA STRUCTURE DE FORMATION	2
II. PRESENTATION DE LA STRUCTURE D'ACCUEIL	4
CHAPITRE II : ANALYSE ET CONCEPTION.....	6
I. ETUDE PREALABLE	6
II. EXPRESSIONS DES BESOINS	14
III. CONCEPTION GLOBALE.....	30
IV. REALISATION	38
CHAPITRE III : BILAN DU STAGE	49
I. DÉROULEMENT DU STAGE ET ACTIVITÉS MÉNÉES	49
II. OBSERVATIONS ET SUGGESTIONS.....	50
CONCLUSION GENERALE.....	51
BIBLIOGRAPHIE ET WEBOGRAPHIE.....	52
TABLES DES MATIERES	54
ANNEXES.....	57
ANNEXE 1 : LANGAGE UML.....	57
ANNEXE 2 : LA METHODE COCOMO.....	59

DEDICACE

Je dédie ce modeste
travail à toute ma
famille, elle qui se
sacrifie pour
mon bonheur.

REMERCIEMENTS

Nous ne saurons continuer notre travail sans d'abord présenter nos sincères remerciements à tous ceux qui ont contribué de près ou de loin à l'obtention ainsi qu'au bon déroulement de ce stage.

Nous saisissons cette occasion pour remercier :

- ❖ Dieu, qui par sa grande miséricorde, nous a donné la santé et la force de terminer ce modeste travail à temps ;
- ❖ M. Bathily DJANGO, Directeur Général de Sira Labs, qui nous a accueilli dans sa structure et qui n'a ménagé aucun effort, pour nous soutenir malgré ses multiples occupations, à nous écouter et à nous orienter vers la bonne direction en apportant des réponses et ressources adéquates à toutes nos préoccupations ;
- ❖ M. Laurent NIKIEMA, notre Maître de stage pour l'encadrement de ce travail, pour ses conseils, ses critiques, ses encouragements et sa disponibilité pour nous aider à accomplir ce stage dans les meilleures conditions ;
- ❖ Pr Gilbert BAYILI, Directeur de l'IBAM, et tout le corps professoral pour les conseils et la formation de qualité que nous avons reçue au cours de ces trois dernières années ;
- ❖ Dr Ferdinand G, notre Directeur de rapport, qui nous a témoigné son entière disponibilité à nous assister tout au long de la rédaction de ce document ;
- ❖ Tout le personnel de SIRA Labs pour leur disponibilité et leur collaboration ;
- ❖ Tous mes parents, amis, camarades dont le soutien a été inestimable ;
- ❖ A tous les soldats courageux tombés au front, nous nous souvenons de leur dévouement, de leur bravoure et de leur sacrifice ultime.

LISTES DES SIGLES ET ABBREVIATIONS

Abbréviation	Signification
2TUP	2 Track Unified Process
ABF	Assurance-Banque-Finance
ADB	Assistant de Direction/Bilingue
ADC	Assistant de Direction/Comptable
API	Application Programming Interface
ATOS	Administratif Technique Ouvrier et de Soutien
AWS	Amazon Web Service
AWS	Amazon Web Services
BPMN	Business Process Model and Notation
CCA	Comptabilité-Contrôle-Audit
CSAFC	Chef de Service Administratif Financier et Comptable
CSS	Cascading Style Sheet
CU	Cas d'utilisation
FCFA	Franc de la Communauté Financière Africaine
HTML	HyperText Markup Language
IBAM	Institut Burkinabè des Arts et Métiers
IBF	Master en Ingénierie Bancaire et Financière
M2ISIE	Master en Informatique option Ingénierie des systèmes d'informations des Entreprises
MAGE	Master en Administration et Gestion des Entreprises
MBT	Model Based Testing
MCCA	Master en Comptabilité-Contrôle-Audit
MG	Marketing et Gestion
MIAGE	Méthodes Informatiques Appliquées à la Gestion
MID	Marketing et Innovation Digitale
MISI	Master en Informatique option Sécurité informatique
RI-MT	Informatique, option Réseaux et Télécommunications
RUP	Rational Unified Process
SGBD	Système de Gestion de Base de données
SI	Système d'Information
SysML	System Modeling Language
UML	Unified Modeling Language
XP	eXtreme Programming

LISTE DES FIGURES GRAPHIQUES

Figure 1 : La hiérarchie schématique selon UML	12
Figure 2 : Planning de réalisation du projet.....	14
Figure 3 : Diagramme des cas d'utilisation	20
Figure 4 : Diagramme de séquence du cas d'utilisation « S'authentifier ».....	24
Figure 5 : Diagramme de séquence du cas d'utilisation « Ajouter un projet ».....	26
Figure 6 :Diagramme de séquence du cas d'utilisation « Valider une ressource »	27
Figure 7:Schéma représentant l'architecture deux tiers [4]	29
Figure 8:Schéma représentant l'architecture 3 tiers [4]	29
Figure 9 : Diagramme des classes du système de gestion et de collaboration des projets incubés à Sira Labs	33
Figure 10:Diagramme d'activité du cas d'utilisation « S'authentifier »	34
Figure 11:Diagramme d'activité du cas d'utilisation « Ajouter un projet ».....	35
Figure 12 : Diagramme du cas d'utilisation « Valider une ressource »	36
Figure 13 : Diagramme de déploiement	37
Figure 14: Liste comparative des SGBD [5]	39
Figure 15 : Ecran de connexion à l'interface administrateur	45
Figure 16 : Ecran de visualisation de l'interface administrateur.....	45
Figure 17 : Interface administrateur qui permet de créer les utilisateurs de l'application	46
Figure 18 : Interface de connexion de l'application	46
Figure 19: Liste des projets	47
Figure 20 : Ecran pour créer un nouveau projet	47
Figure 21 : Tableau de bord du projet	48
Figure 22 : Liste des ressources.....	48

LISTE DES TABLEAUX

Tableau 1: Tableau comparatif des méthodologies ou processus de développement	8
Tableau 2 : Comparaison entre quelques langages de modélisation [3]	11
Tableau 3 : Description de quelques solutions logicielles de gestion de projets existantes	16
.....	
Tableau 4 : Liste des cas d'utilisations	19
Tableau 5 : Description textuelle du cas d'utilisation « S'authentifier »	21
Tableau 6 : Description textuelle de cas d'utilisation : Ajouter un nouveau projet	22
Tableau 7 : Description du cas d'utilisation "Valider une ressource"	23
Tableau 8 : Comparaison des différentes architectures de développement.....	29
Tableau 9 : Dictionnaire de données	31
Tableau 10 : Tableau comparatif des serveurs d'applications	40
Tableau 11 : Comparaison des méthodes d'estimation de cout et de développement [6]	43
.....	
Tableau 12 : Cout total du projet.....	44
Tableau 13 : Formule de calcul COCOMO.....	59

INTRODUCTION GENERALE

Le secteur des startups connaît une croissance exponentielle, portant en son sein un potentiel d'innovation et de développement économique considérable. Cependant, ces jeunes entreprises font face à de nombreux défis pour transformer leurs idées en succès commercial durable. Pour les accompagner dans leur parcours, des coaches et des mentors jouent un rôle crucial en les guidant, en les conseillant et en les aidant à atteindre leurs objectifs.

Cependant, le suivi des startups par ces coaches et mentors peut s'avérer complexe, surtout lorsque ces derniers sont géographiquement dispersés. Dans le cadre de mon stage au sein de Sira Labs, une entreprise au fort engagement panafricain, j'ai eu l'opportunité de travailler sur le thème : « Mise en place d'un système de gestion intégré visant à faciliter le suivi et l'accompagnement des startups incubées à Sira Labs ».

Ce rapport de stage vise à présenter en détail le développement et l'implémentation d'une plateforme de gestion de projet permettant aux coaches et mentors de suivre en temps réel l'évolution des startups qu'ils accompagnent, et de faciliter la communication et les échanges d'informations entre les différentes parties prenantes.

Pour mener à bien notre travail, nous avons scindé notre étude en trois grands chapitres. Le premier chapitre sera consacré à la présentation de notre structure de formation et d'accueil. Le deuxième chapitre se rapporte à l'analyse et à la conception du système. Le troisième chapitre se résume au bilan du stage effectué au sein de Sira Labs.

CHAPITRE I : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL

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

I. PRESENTATION DE LA STRUCTURE DE FORMATION

L'objectif de l'IBAM est de répondre aux besoins du marché de l'emploi en mettant à sa disposition un potentiel humain de cadres moyens et supérieurs dans les divers secteurs d'activité. Une nouvelle orientation qui a pour objectif une meilleure insertion professionnelle des produits de l'Université Joseph KI-ZERBO.

1. Objectifs de l'IBAM

L'objectif de l'IBAM est de répondre aux besoins du marché de l'emploi en mettant à sa disposition un potentiel humain de cadres moyens et supérieurs dans les divers secteurs d'activité. Une nouvelle orientation qui a pour objectif une meilleure insertion professionnelle des produits de l'Université Joseph KI-ZERBO.

2. Organisation

L'organisation de l'IBAM est une structure hiérarchico-fonctionnelle. Nous avons :

- le conseil de gestion qui est l'organe suprême qui regroupe le directeur, le directeur adjoint, les coordonnateurs, les enseignants permanents, le CSAFC, la secrétaire principale et le représentant du personnel ATOS ;
- le Conseil Scientifique qui regroupe le Directeur, le Directeur Adjoint, les coordonnateurs et les enseignants de rang A de l'Institut ;
- le cabinet du directeur auquel sont rattachés directement le CSAFC, le Directeur Adjoint, et la Secrétaire Principale auquel est rattaché le personnel ATOS. Au directeur adjoint sont rattachés les coordonnateurs de filières et aux coordonnateurs de filières est rattaché le personnel enseignant de l'Institut.

3. Filières de formations

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

❖ Licences professionnelles

Les filières de formations initiales en licences professionnelles sont :

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

❖ Masters

En master, l'IBAM offre cinq (05) filières de formation que sont :

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

La filière MIAGE a été créée dans l'optique de répondre aux besoins croissants des entreprises en cadres compétents dans le domaine des Technologies de l'Information et de la Communication (TIC). Cette filière accueille en première année les bacheliers des séries C, D et E ayant passé avec succès le test de sélection ou ayant été acceptés en tant qu'auditeurs libres.

Les étudiants en fin de cycle en MIAGE doivent mettre en pratique les connaissances acquises en classe en effectuant un stage d'une durée d'au moins trois mois suivi d'une soutenance publique. Ce stage a pour objectif de leur permettre de se familiariser avec le monde

professionnel et d'appliquer leurs connaissances théoriques acquises au cours de leur formation.

C'est pour répondre à cette exigence académique que nous avons effectué notre stage pratique dans les locaux de Sira Labs que nous allons présenter dans les lignes suivantes.

II. PRESENTATION DE LA STRUCTURE D'ACCUEIL

Dans cette partie, nous présentons notre structure d'accueil Sira Labs qui est un acteur majeur de l'accompagnement à l'Innovation et à l'Entrepreneuriat dans la sous-région avec une présence dans 04 villes dont Ouagadougou, Bobo-Dioulasso, Dakar et Saint-Louis.

1. Histoire

Incubateur et Accélérateur d'entreprises, Sira Labs offre divers programmes riches allant de la structuration d'idées de projet innovantes à l'accélération de croissances d'entreprises. Sira Labs offre aussi aux institutions de développement un appui à la réalisation de programme d'appui à l'entrepreneuriat, l'insertion des jeunes et l'innovation. Nous offrons aussi divers programmes de formation et de renforcement de capacités des entreprises en gestion d'entreprises et élaboration de stratégies d'entreprises.

Depuis sa création, Sira Labs a mis un accent particulier sur le développement d'un réseau de partenaires nationaux et internationaux de qualités afin d'offrir aux entrepreneur(e)s qu'il accompagne, les meilleures méthodologies et expertises en matière d'accompagnement et de renforcement de capacités.

2. Objectifs

L'essor des technologies numériques a créé des environnements d'affaires très compétitifs, rendant assez précaire les nombreux projets de création d'entreprises naissant un peu partout en Afrique subsaharienne et bénéficiant de moyens limités.

Sira Labs offre un accompagnement adapté et une approche innovatrice pour faciliter la réussite des projets de création d'entreprises et une pérennisation de ces dernières.

- Vision : Les entrepreneurs africains évoluent dans un écosystème global dans lequel ils ont accès à toutes les compétences et aux outils dont ils ont besoin pour réaliser leurs projets avec succès.
- Mission : Créer un écosystème dynamique favorable au développement de projets entrepreneuriaux innovants durables en Afrique de l'Ouest.

3. Services

Sira Labs intervient dans les domaines d'incubation et d'accélération de startups dans la sous-région à travers cinq (02) activités phares : des programmes d'incubations et des programmes de mentorat d'excellence.

Les principaux services qu'elle offre sont les suivants :

❖ Une offre complète appelée : INCUBATION 1KREATION

Cette offre d'incubation est composée des programmes suivants :

- 1Kreation Digital, pour tout projet innovant intégrant du digital,
- 1Kreation DeepTech, pour tout projet issu des universités et laboratoires de recherche.
- 1Kreation Impact, pour les projets autour de l'environnement et de l'économie sociale et solidaire,
- 1Kreation Hardware, pour les projets d'objets connectés et les processus industriels, en partenariat avec start2prod, le laboratoire de Prototypage du groupe Michelin.

❖ Une offre destinée à l'accélération des startups appelée :1KROISSANCE

Sira Labs a développé un programme de mentorat d'excellence à destination des startups ayant déjà fait leur preuve de marché, leur preuve de produit et ayant généralement levé des fonds, autour de 3 thèmes qui leur sont nécessaire :

- Vente et marché
- Structure et croissance
- Financement et international.

Dans ce chapitre nous avons présenté notre structure de formation qu'est l'Institut Burkinabè des Arts et Métiers ainsi que de la structure d'accueil, l'incubateur Sira Labs. Nous allons dans la suite du travail présenté comment nous avons procédé pour l'analyse et la conception de notre système.

CHAPITRE II : ANALYSE ET CONCEPTION

Après avoir présenté les structures de formation et d'accueil, nous présentons dans ce chapitre, dans un premier volet le thème soumis à notre étude, la méthode d'analyse et de conception du projet, les acteurs impliqués ainsi que le planning de réalisation. Dans un deuxième volet, une étude technique et une conception détaillée du projet. Enfin, nous présentons les différents éléments entrant dans la réalisation du projet.

I. ETUDE PREALABLE

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 deux (02) axes principaux à savoir : la problématique et les résultats attendus.

a. Problématique

Sira Labs est un incubateur panafricain qui s'engage à accompagner les startups dans leur parcours vers la concrétisation de leurs projets de manière efficiente. Cependant, l'accompagnement de ces startups est un défi complexe, notamment en raison de la répartition géographique des coachs et mentors, qui se trouvent un peu partout en Afrique. Cette dispersion géographique crée des défis significatifs pour assurer un suivi cohérent et efficace des projets des entrepreneurs que nous accompagnons.

Afin de relever ce défi, Sira Labs a entrepris une réflexion approfondie pour répondre à la question fondamentale suivante : Comment faire pour permettre à nos coachs et porteurs de projets de pouvoir communiquer et travailler efficacement malgré l'éloignement et la dispersion géographique ?

C'est dans cette optique que le sujet suivant nous a été confié : « *Conception et réalisation d'une plateforme de gestion intégrée et de suivi des startups incubées à Sira Labs* ». Cette initiative vise à créer un environnement numérique propice à la collaboration transparente entre les porteurs de projets, les coachs et les mentors, quel que soit leur emplacement géographique.

b. Objectifs

Les objectifs assignés à ce projet sont les suivants :

- analyser le système de gestion de projets à Sira Labs ;

- automatiser le système de gestion des projets incubés à Sira Labs ;
- mettre en pratique les connaissances acquises durant notre formation à l'IBAM.

c. Résultats attendus

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

le rapport de l'analyse détaillée et améliorée du fonctionnement du système de gestion des projets incubés à Sira Labs ;

- le logiciel de gestion de projets de Sira Labs ;
- la mise en pratique des connaissances apprises au cours des 3 années d'études ;
- le rapport de stage est rédigé et soutenu devant un jury.

2. Méthodologie

Par le terme méthodologie, nous faisons allusion au processus de développement et au langage de modélisation utilisé dans le cadre du processus de développement de système de gestion des programmes d'activités. Dans cette partie, nous présentons d'abord le processus de développement que nous avons utilisé et ensuite le langage de modélisation également.

a. Processus de développement

Le processus de développement est l'ensemble des activités à réaliser de manière structurée pour atteindre l'objectif d'un projet de développement d'un Système d'Information (SI). Il existe plusieurs types de processus de développement parmi lesquels les cinq (5) plus populaires sont CASCADE¹, RUP², 2TUP³, SCRUM⁴ et XP⁵. Le tableau 1 est un résumé de l'étude comparative entre le processus de développement CASCADE, SCRUM, 2TUP, RUP et XP.

¹ CASCADES : www.ionos.fr

² RUP : prezi.com

³ 2TUP : prezi.com

⁴ SCRUM : asana.com

⁵ XP : prezi.com

Processus de développement	Description	Atouts	Insuffisance
CASCADES	Les phases sont déroulées d'une manière séquentielle	Distingue clairement les phases du projet	<ul style="list-style-type: none"> • Non itératif. • Pas de modèles pour les documents
RUP (Rational Unified Process)	<ul style="list-style-type: none"> • Le RUP est à la fois une méthodologie et un outil prêt à l'emploi. • Cible des projets de plus de 10 personnes. 	<ul style="list-style-type: none"> • Itératif. • Spécifie le dialogue entre les différents intervenants du projet. 	<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.
2TUP (Two Truck Unified Process)	<ul style="list-style-type: none"> • Il s'articule autour de l'architecture. • Propose un cycle de développement en Y. • Cible des projets de toutes tailles. 	<ul style="list-style-type: none"> • Itératif. • Laisse une large place à la technologie et à la gestion des risques. • Définit les profils des intervenants, les plannings, les prototypes 	<ul style="list-style-type: none"> • Plutôt superficiel sur les phases situées en amont et en aval du développement. • Ne propose pas de documents types.
SCRUM	Se base sur des itérations dites sprints de développement	<ul style="list-style-type: none"> • Donne toute confiance aux développeurs et les laisser effectuer leur travail. • Chaque itération a un objectif bien précis et fournit une fonctionnalité testée 	<ul style="list-style-type: none"> • La mise en œuvre du développement n'est pas précisée. • Développement rapide et répétitif
XP (eXtreme Programming)	<ul style="list-style-type: none"> • Ensemble des bonnes pratiques de développement. • Cible : Moins de 10 personnes 	<ul style="list-style-type: none"> • Itératif. • Donne une importance aux aspects techniques. • Innovant : programmation en duo. 	<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.

Tableau 1: Tableau comparatif des méthodologies ou processus de développement

À la suite de l'étude et à la comparaison des processus de développement et aux fins de contrôler les risques et de mener à bon terme notre projet, nous avons opté pour le processus 2TUP pour les raisons suivantes :

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

Le processus 2TUP comporte 2 branches :

- ❖ une branche fonctionnelle qui se présente comme suit :
 - la capture des besoins fonctionnels, Cette étape vise à identifier les besoins spécifiques des utilisateurs en se concentrant sur leur métier ;
 - l'analyse qui consiste à étudier les spécifications fonctionnelles dans le but de connaître le métier qui sera effectué par le système ;
- ❖ une branche technique qui se présente comme suit :
 - la capture des besoins techniques qui consiste à énumérer les outils, matériels et technologies dont on aura besoin.
 - la conception générique qui consiste à indiquer les composants essentiels à l'élaboration de l'architecture technique.

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

- la conception préliminaire qui doit intégrer le modèle d'analyse fonctionnelle dans l'architecture technique afin d'indiquer les composants du système à développer.
- la conception détaillée qui permet d'exécuter chaque composant du système ;
- le codage qui consiste à implémenter les fonctionnalités du système avec des tests effectués au fur et à mesure ;

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

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

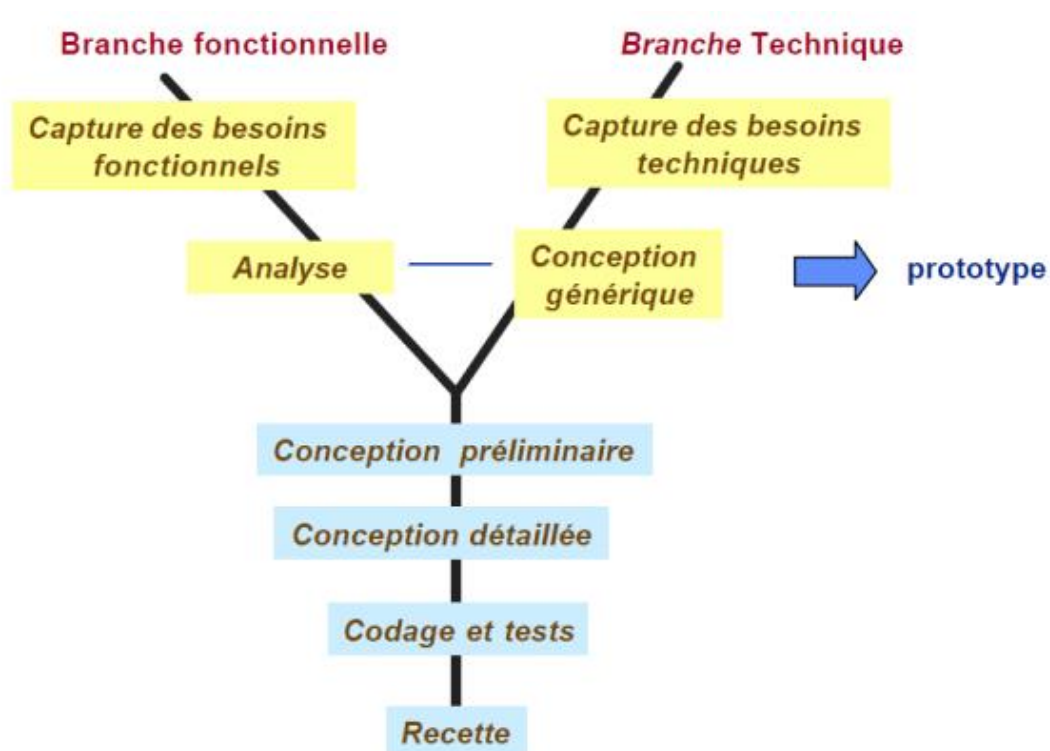


Figure 2 : Schéma représentant le processus 2TUP [1]

Le processus de développement 2TUP suggère également la présentation des différents groupes de travail du projet à savoir : le groupe de pilotage, le groupe des utilisateurs et le groupe de projets qui sera faite après avoir abordé le langage de modélisation utilisé dans notre travail.

b. Langage de modélisation

Un langage de modélisation est un langage informatique spécialement conçu pour représenter des modèles abstraits de systèmes, de processus, de données ou de concepts. Ces modèles peuvent être utilisés pour comprendre, analyser, concevoir, communiquer ou documenter des aspects complexes de l'information ou de la réalité. Il existe plusieurs langages de modélisation parmi lesquels nous pouvons énumérer : UML⁶, SysML⁷, BPMN⁸ et MBT⁹.

⁶ UML : <https://www.lucidchart.com/pages/fr/langage-uml>.

⁷ SysML : [OMG SysML Home | OMG Systems Modeling Language](#)

⁸ BPMN : [Spécification BPMN - Modèle de processus métier et notation](#)

⁹ MBT : [MaTeLo - Le Model-Based Testing \(matelo-testing-software.com\)](#)

Le tableau 2 présente une étude comparative entre les langages de modélisation cités précédemment.

Langages de modélisation	Caractéristiques
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 ; • international.
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.
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 ; • renforce la qualité de la documentation des exigences.

Tableau 2 : Comparaison entre quelques langages de modélisation [3]

Suite à l'étude comparative, nous avons choisi d'utiliser le langage UML pour diverses raisons majeures, à savoir :

- il présente l'avantage de représenter la norme universellement reconnue en matière de modélisation objet.;
- il se caractérise par son aspect visuel, avec une notation graphique qui permet de visualiser de manière claire les solutions objet, facilitant ainsi leur comparaison et leur évaluation.
- il possède un caractère formel et normalisé, garantissant une précision accrue et une stabilité dans le processus de modélisation.

- il sert à formaliser l'ensemble des documents techniques d'un projet, ce qui permet d'affiner les détails de l'analyse au fur et à mesure de l'évolution du projet.
- il permet d'utiliser un même ensemble d'outils de génie logiciel, de la phase d'expression des besoins des utilisateurs jusqu'à la génération partielle ou totale du code.
- il constitue un excellent support de communication, encadrant l'analyse tout en simplifiant la compréhension des représentations abstraites complexes.

UML offre un ensemble de notations standardisées, de diagrammes et de symboles pour décrire la structure, le comportement, les interactions et les aspects dynamiques des systèmes que nous illustrons dans la figure 1 :

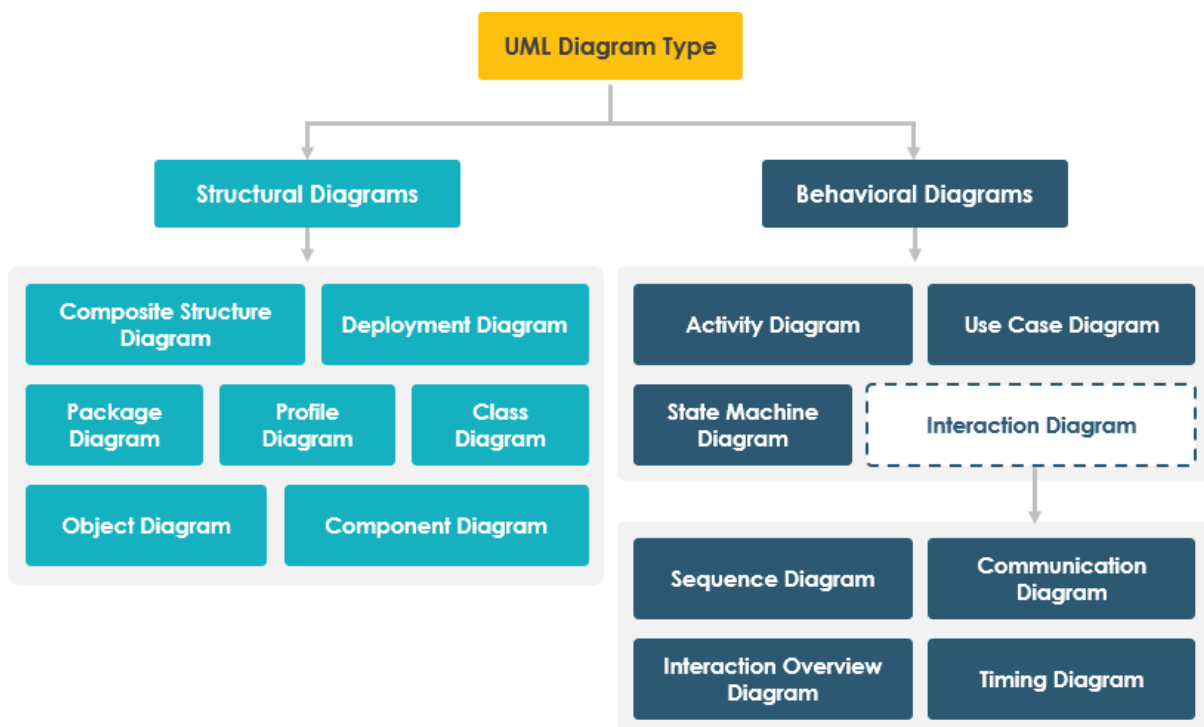


Figure 1 : La hiérarchie schématique selon UML [2]

Après avoir exposé la méthodologie de développement, abordons maintenant la présentation de l'équipe de travail chargée de mener à bien ce projet.

3. Présentation du groupe de travail

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

a. Groupe de pilotage

C'est le groupe dirigeant chargé de veiller au bon déroulement du projet. Il a à sa charge la planification des dates clés du projet, l'examen des propositions du groupe de projet et décide des orientations stratégiques. Ce groupe est constitué de :

- M. Bathily DJANGO, Directeur Général de Sira Labs ;
- M. Laurent NIKIEMA, Maître de notre stage ;
- Dr Ferdinand GUINKO, Directeur de notre rapport de stage.

b. Groupe des utilisateurs

Ce groupe est constitué de l'ensemble de ceux qui vont utiliser le futur système. Il participe à la capture des besoins fonctionnels. Il est composé principalement des porteurs de projets, des coachs, et des mentors de l'incubateur Sira Labs.

c. Groupe de projet

Le groupe de projet est l'ensemble des personnes chargées de réaliser le projet. Il est l'intermédiaire entre le groupe de pilotage et le groupe des utilisateurs. Ce groupe est composé principalement de Rimalguedo Rahimata SAWADO, étudiante en Méthodes Informatiques Appliquées à la Gestion à l'IBAM.

Nous venons de définir l'ensemble des personnes qui vont intervenir dans la réalisation du système de gestion des projets incubés à Sira Labs. Présentons maintenant le planning de réalisation établi pour la réalisation du projet.

d. Planning de travail

Pour mener à bien notre étude et la réaliser dans les délais, nous avons subdivisé le projet en tâches. Pour cela, nous avons utilisé le diagramme de Gantt qui est un outil utilisé en ordonnancement et en gestion de projet ; il permet de visualiser dans le temps les différentes tâches liées à un projet. Le planning prévisionnel est résumé dans la figure 2 :

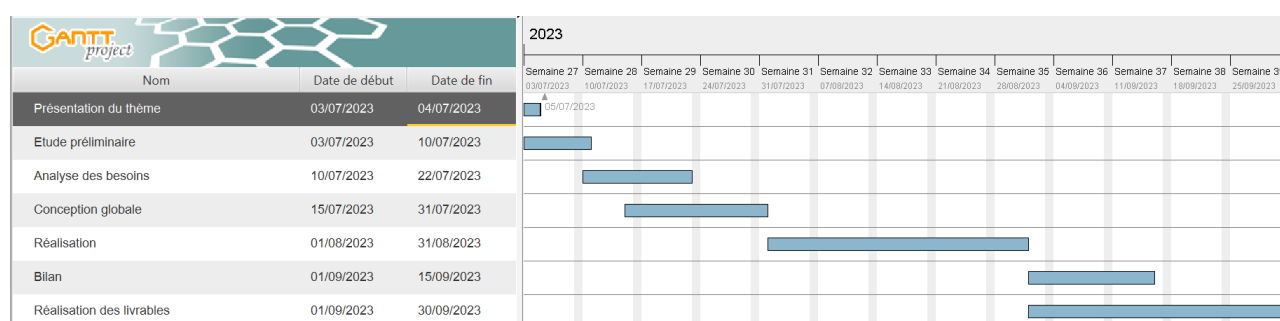


Figure 2 : Planning de réalisation du projet

Faire un commentaire de l'image : début et fin de stage et étape effectuées

Après avoir présenté le planning de réalisation, passons maintenant à la présentation de l'expression des besoins qui concerne les acteurs et les messages qui sont échangés entre eux et le système de gestion des projets incubés à Sira Labs.

II. EXPRESSIONS DES BESOINS

Cette partie consiste d'une part à présenter le fonctionnement des modèles existants et améliorés du système de gestion des projets incubés à Sira Labs. D'autre part, elle consiste à présenter les différents acteurs qui interagissent avec le système de gestion des programmes d'activités ainsi que les différents messages qu'ils échangent entre eux et ledit système.

1. Étude de l'existant

L'étude de l'existant est une démarche méthodique et analytique visant à examiner et comprendre en détail les éléments actuels d'un système, d'une organisation, ou d'un processus. Nous allons mener cette étude dans les lignes qui suivent pour le cas de Sira Labs.

a. Présentation du système utilisé à Sira Labs

Le but de la présentation du système utilisé pour la gestion des projets incubés à Sira Labs et la collaboration entre les acteurs concernés par ce projet en Septembre 2023 est de déterminer les points faibles et les points forts du système de gestion de projets actuellement utilisé à Sira Labs pour pouvoir déterminer les besoins du client, en vue d'en prendre en considération lors de la conception et de la réalisation de la solution. Dans cette partie, nous présentons le système existant pour la gestion des projets et la collaboration entre les acteurs du projet de Sira Labs.

Sira Labs ne possède pas de plateforme dédiée pour faire la planification et le suivi de ses projets incubés. Pour enregistrer et suivre l'état d'avancement des projets à Sira Labs, la responsable administrative de Sira Labs, enregistre les données des porteurs de projet dans un fichier Excel qu'elle confie aux coachs et aux mentors de Sira Labs. Une fois les mentors et les coachs de Sira Labs en possession des informations du projet, ils communiquent sur l'application de messagerie WhatsApp de l'évolution du projet et ils attribuent aux porteurs de projets des tâches via WhatsApp.

Une fois le contact établi entre le porteur de projet, son coach et son mentor, les coachs et mentors fournissent un rapport détaillé de l'évolution du projet à Sira Labs.

A l'issue de cette présentation de fonctionnement du système existant à Sira Labs nous décelons les forces et les faiblesses du système que nous allons énumérer dans les lignes suivantes :

❖ forces du système

Le système existant possède quelques forces à savoir :

- facilité et simultanéité des discussions ;
- amélioration des relations entre les acteurs du projet.

❖ faiblesses du système

Le système existant pose plusieurs problèmes à savoir :

- perte de temps et de ressources ;
- problème du suivi de l'avancement des projets ;
- problème de centralisation de l'information.

Ces avantages et inconvénients que nous avons déterminés du système existant de Sira Labs prouvent que l'incubateur a besoin de digitaliser et d'utiliser un outil personnalisé pour pouvoir suivre efficacement l'avancement des projets incubés en leur sein.

Maintenant, nous avons pris connaissance du système de gestion existant à Sira Labs. Étudions maintenant les solutions logicielles de gestion de projet existantes.

b. Présentation d'outils logiciels de gestion de projet existants

Gérer les projets de toutes tailles des entreprises est un problème auquel plusieurs startups ont trouvé des solutions. Et ces solutions sont variées en fonction des besoins du client et de la taille du projet. Après une recherche approfondie sur Google, voici un tableau récapitulatif de quelques solutions logicielles existantes qui sont : Trello¹⁰, Assana¹¹, Notion¹², Jira¹³ et leurs caractéristiques :

Mettre ce tableau sur la page suivante

Caractéristiques	Trello	Assana	Notion	Jira
Tableaux de bord	Non	Oui	Oui	Oui
Tâches et sous-tâches	Oui	Oui	Oui	Oui
Vues de projet	Non	Oui	Oui	Oui

¹⁰ Trello : [Gérez les projets de votre équipe où que vous soyez | Trello](#)

¹¹ Assana : [Gérez en ligne le travail, les projets et les tâches de votre équipe • Asana](#)

¹² Notion : [Your connected workspace for wiki, docs & projects | Notion](#)

¹³ Jira : [Optimisez le travail de votre équipe avec Jira Software \(atlassian.com\)](#)

Calendrier	Non	Oui	Oui	Oui
Personnalisation	Limitée	Oui	Oui	Oui
Prix mensuel par utilisateur	À partir de 6190 FCFA	À partir de 6190 FCFA	À partir de 2476 FCFA	À partir de 4333 FCFA

Tableau 3 : Description de quelques solutions logicielles de gestion de projets existantes

Suite à la comparaisons des solutions logicielles de gestion de projet existantes dans le tableau 3, nous avons constaté que toutes ces plateformes sont payantes pour une utilisation professionnelle, et la capacité de personnalisation est restreinte pour une entreprise comme Sira Labs et le tarif augmente en fonction du nombre d'utilisateurs ce qui ne convient pas à Sira Labs qui a plusieurs types d'utilisateurs sur un projet quelconque, d'où le besoin de développer un système de gestion adapté à leur entreprise.

Maintenant que nous avons une idée sur le fonctionnement du système de gestion des projets existant de Sira Labs, des outils, logiciels existants également pour la gestion des projets et leurs limites, nous présentons le système de gestion de projets amélioré qui prend en compte les limites du système existant. **Je n'ai pas trouvé d'outils qui correspondent aux exigences fonctionnelles**

2. Présentation du processus de fonctionnement amélioré

Le système amélioré permet aux différents acteurs d'un projet de Sira Labs de travailler simultanément sur un projet et de voir l'avancement du projet sur une même plateforme appelée SIRAIIS.

SIRAIIS est une application web de gestion de projets innovante conçue pour simplifier la collaboration entre les porteurs de projets, les coachs et les mentors tout au long du cycle de vie d'un projet à Sira Labs. Elle offre un ensemble complet de fonctionnalités pour faciliter la planification, la coordination et le suivi des projets, tout en favorisant le partage d'informations.

Elle comporte deux interfaces, l'interface administrateur, et l'interface dédiée aux porteurs de projets, aux mentors et aux coachs.

Depuis, l'interface Web, l'administrateur crée les comptes des utilisateurs associés à un projet, et leur accorde les permissions nécessaires au bon déroulement du projet. Ensuite il crée les projets des porteurs de projets, crée des tâches et les assigne aux porteurs de projets sur la plateforme.

Après leur compte créé par l'administrateur, les porteurs de projets, peuvent depuis la version web, après s'être authentifiés, vérifier leur état d'avancement, ajouter des ressources à leur projet, prendre connaissance des tâches qui leur ont été assignées et soumettre des commentaires au projet.

Après leur compte créé par l'administrateur, les coachs et les mentors, peuvent depuis la version web, après s'être authentifiés, vérifier l'état d'avancement des projets qui leur sont confiés, ajouter des ressources aux différents projets, ajouter des tâches aux porteurs de projets, changer le statut de ses tâches (En cours, terminé, bloqué) et soumettre des commentaires au projet.

Après avoir présenté le fonctionnement du modèle amélioré du système de gestion des programmes d'activités, nous présentons maintenant les exigences fonctionnelles de Sira Labs.

3. Exigences fonctionnelles

SIRA Labs a défini des exigences fonctionnelles claires pour le développement de leur logiciel. Ces exigences comprennent la création d'interfaces distinctes pour l'administration et les clients, avec une emphase particulière sur la personnalisation de ces interfaces. Un critère majeur est l'efficacité financière à long terme du logiciel, soulignant la nécessité d'une solution économique sur la durée. De plus, le besoin de créer et de gérer des groupes d'utilisateurs via l'interface d'administration a été spécifié. Cette approche systématique démontre la vision stratégique de SIRA Labs pour un logiciel qui allie fonctionnalité, personnalisation et rentabilité.

Après avoir présenté le fonctionnement du modèle amélioré du système de gestion des programmes d'activités, nous présentons maintenant la spécification fonctionnelle.

Mettre les exigences fonctionnelles juste après la présentation de l'existant

4. Spécification fonctionnelle

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

a. Identification des acteurs

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

- de l'administrateur;
- du mentor : celui qui est chargé de suivre le projet jusqu'au bout;
- du coach : celui qui est chargé d'intervenir sur une partie spécifique du projet ;
- du porteur de projet : celui chargé de travailler son projet.

b. Identification des cas d'utilisations

Les cas d'utilisation désignent l'ensemble des interactions qui permettent à l'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 (15) cas d'utilisation inscrits dans le tableau 3. Nous avons par exemple le cas d'utilisation « S'authentifier ». Ce cas d'utilisation ou fonctionnalité permet aux coaches, aux porteurs de projets et aux mentors de se connecter au système via l'interface administrateur ou l'interface client. Un deuxième exemple est celui du cas d'utilisation « Gérer les projets », qui permet à l'administrateur uniquement de créer de nouveaux projets, de mettre à jour les détails du système existant et de supprimer des projets du système. Un troisième exemple est celui du cas d'utilisation « Valider une ressource », qui permet au coach de valider les ressources soumises par les porteurs de projet, garantissant ainsi leur pertinence et leur conformité.

Maintenant que nous avons une vue détaillée de ces cas d'utilisation, examinons de plus près leurs caractéristiques et comparons-les dans le tableau 4 :

Numéro	Cas d'utilisation	Description	Acteur(s)
CU01	Authentification	Permet aux utilisateurs de se connecter à l'application	Tous les acteurs
CU02	Gérer les utilisateurs	Permet aux administrateurs de créer, modifier et supprimer des utilisateurs	Administrateur
CU03	Gérer les projets	Permet de créer, modifier, supprimer, afficher les détails, voir les ressources des projets	Administrateur
CU04	Visualiser les projets	Permet de modifier, afficher les détails et voir les ressources des projets	Tous les acteurs du projet
CU05	Suivre l'état d'un projet	Permet aux utilisateurs de suivre l'avancement d'un projet	Tous les acteurs du projet
CU06	Gérer les ressources	Permet d'ajouter, de modifier, de supprimer et de voir les ressources associées à un projet	Tous les acteurs du projet
CU07	Valider une ressource	Permet au coach de valider une ressource soumise	Coach

CU08	Gérer les tâches	Permet aux coachs d'ajouter, de modifier, de supprimer des tâches aux projets existants	Coach
CU09	Visualiser les tâches	Permet aux porteurs de projets de voir et de valider les tâches a de son projet.	Coach
CU10	Ajouter un Business Model Canva	Permet au porteur de projet de remplir son business model sur la plateforme	Porteur de Projet
CU11	Gérer les agendas	Permet aux utilisateurs de vérifier, créer, planifier l'agenda sur la plateforme	Tous les acteurs du projet
CU12	Gérer les notifications	Permet aux utilisateurs de recevoir et d'ouvrir les notifications	Tous les acteurs du projet
CU13	Visualiser les données	Permet aux utilisateurs de voir les données associées au projet	Tous les acteurs du projet
CU14	Gérer les groupes	Permet de créer, modifier ou supprimer des groupes d'utilisateurs	Administrateur
CU15	Octroyer des permissions	Permet d'accorder ou révoquer des permissions aux utilisateurs	Administrateur

Tableau 4 : Liste des cas d'utilisations

Pour une meilleure visualisation globale de l'ensemble des cas d'utilisation du système nous présentons le diagramme des cas d'utilisation du système dans la suite de notre travail.

c. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Il permet d'une part de modéliser les besoins des utilisateurs en les clarifiant et en les organisant et d'autre part d'identifier les acteurs et les fonctionnalités du système comme le montre la figure 3. **Enlever la flèche entre le porteur et l'admin**

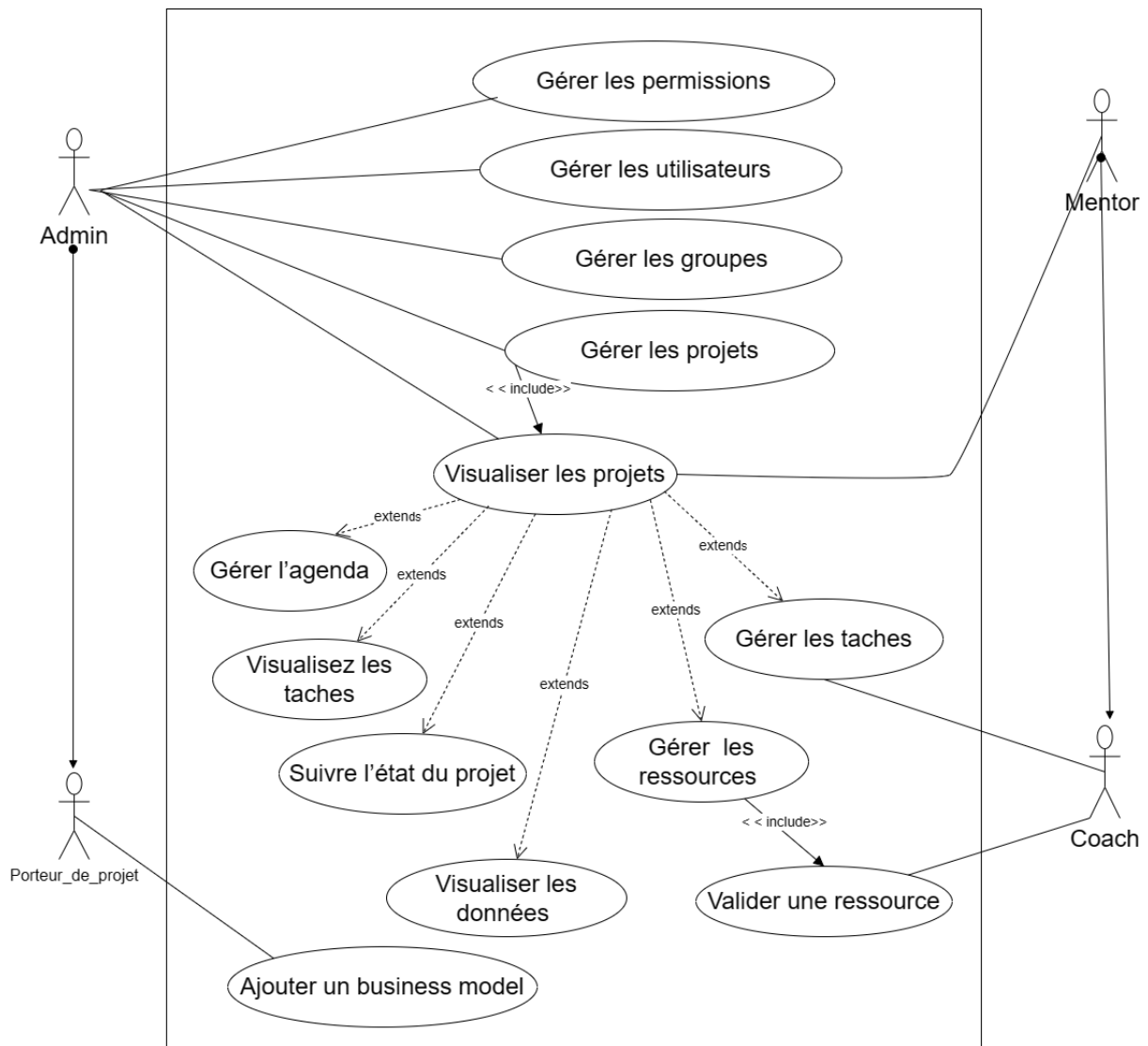


Figure 3 : Diagramme des cas d'utilisation

Remarque : Tous les cas d'utilisations inclut le cas d'utilisation « S'authentifier ». L'authentification est gérée par le système d'authentification de Django.

Pour mieux comprendre les cas d'utilisation illustrés dans le diagramme des cas d'utilisation et le tableau 4, nous présentons à titre d'exemple les descriptions textuelles de quelques cas d'utilisation.

d. Description textuelle de quelques cas d'utilisation

La description textuelle permet d'exposer de façon détaillée le dialogue entre les acteurs et le système. Elle sert à documenter en détail comment un système interagit avec des acteurs (utilisateurs, systèmes externes, etc.) pour accomplir une fonction spécifique. Préciser que pour un besoin de lisibilité, le reste des cas est dans l'annexe

La description textuelle des cas d'utilisation est un tableau composé de plusieurs parties : la partie acteur, elle présente les différents acteurs pouvant utiliser le cas d'utilisation ; la partie précondition qui décrit dans quel état doit être le système avant que le cas d'utilisation puisse être déclenché ; la partie scénario nominal qui décrit le déroulement normal du cas d'utilisation quand il n'y a pas d'erreur ; la partie scénario alternatif qui est une variante du scénario nominal et enfin la partie post condition qui décrit l'état du système à l'issue des différents scénarios .

Les cas d'utilisation qui sont décrits dans cette partie sont essentiellement les cas d'utilisation « S'authentifier », « Ajouter un projet », et « Valider une ressource » respectivement dans le tableau 5, le tableau 6, et le tableau 7.

❖ CU01 : S'authentifier

Le résumé de ce cas d'utilisation se trouve dans le tableau 5.

Résumé	Ce cas permet à l'utilisateur de se connecter au système en s'identifiant
Acteurs	Tous les acteurs
Précondition	<ul style="list-style-type: none"> Le système est opérationnel. L'utilisateur possède un compte enregistré.
Scénario Nominal	<ul style="list-style-type: none"> L'utilisateur accède à l'interface de connexion du système. L'utilisateur saisit son nom d'utilisateur (identifiant) et son mot de passe dans les champs correspondants. Le système vérifie les informations d'identification fournies par l'utilisateur. Si les informations sont correctes et correspondent à un compte enregistré, l'utilisateur est authentifié et dirigé vers le tableau de bord du système. Si les informations sont incorrectes, un message d'erreur est affiché, invitant l'utilisateur à réessayer.
Scénario Alternatif	<p>[A] : L'identifiant et/ou le mot de passe sont incorrects</p> <ul style="list-style-type: none"> Le système notifie l'utilisateur de l'échec de la connexion Le système renvoie le formulaire de connexion et le scénario reprend à partir du point 2 du scénario nominal.
Post conditions	L'utilisateur accède à son tableau de bord

Tableau 5 : Description textuelle du cas d'utilisation « S'authentifier »

Le tableau 5 décrit les interactions entre les acteurs et le système pour le cas d'utilisation « S'authentifier ». D'abord, l'utilisateur demande à se connecter, puis il reçoit un formulaire de connexion qu'il doit remplir en fournissant son nom d'utilisateur et son mot de passe. Une fois soumis, le système vérifie les informations saisies. Si les informations sont incorrectes le

système lui renvoie le même formulaire pour qu'il réessaie. Au cas où les informations sont correctes l'utilisateur est dirigé vers le tableau de bord qui le concerne. La procédure est valable pour tous les acteurs du système.

❖ CU02 : Ajouter un projet

Le résumé de ce cas d'utilisation se trouve dans le tableau 6.

Résumé	Ce cas permet à l'administrateur d'ajouter un nouveau projet au système
Acteurs	Administrateur
Précondition	<ul style="list-style-type: none"> • L'utilisateur est authentifié et connecté au système. • L'utilisateur a les autorisations nécessaires pour créer un projet
Scénario Nominal	<ul style="list-style-type: none"> • L'utilisateur accède à la page de création de projet. • L'utilisateur remplit le formulaire de création de projet. • L'utilisateur soumet le formulaire. • Le système enregistre les informations du projet dans la base de données. • Le système affiche un message de confirmation indiquant que le projet a été créé avec succès. • L'utilisateur est redirigé vers la page de la liste des projets
Scénario Alternatif	<p>[A] : Le nom du projet existe déjà :</p> <ul style="list-style-type: none"> • Le système détecte que le nom du projet existe déjà dans la base de données, ce qui entraînerait un conflit. • Le système affiche un message d'erreur indiquant que le nom du projet est déjà utilisé et demande à l'utilisateur de choisir un nom de projet différent. • L'utilisateur révisé les informations du projet et modifie le nom du projet. • L'utilisateur soumet à nouveau le formulaire. • Le système valide les informations, crée le nouveau projet avec le nom modifié, et l'ajoute à la base de données.
Post conditions	<ul style="list-style-type: none"> • Un nouveau projet est créé avec les informations fournies par l'utilisateur. • L'utilisateur peut accéder aux détails du projet et le gérer.

Tableau 6 : Description textuelle de cas d'utilisation : Ajouter un nouveau projet

Le tableau 6 détaille le cas d'utilisation qui permet à un utilisateur authentifié, d'ajouter un nouveau projet dans le système. Pour cela, l'utilisateur doit fournir des informations essentielles telles que le nom du projet, sa description, les dates de début et de fin prévues, ainsi que les personnes associées au projet, comme le porteur de projet et le coach. De plus, l'utilisateur peut spécifier l'état actuel du projet, qu'il soit en cours, terminé ou en attente. Une fois le formulaire rempli, l'utilisateur peut soumettre les informations, et le système enregistre le nouveau projet dans la base de données. Cette fonctionnalité est essentielle pour la gestion

et le suivi des projets incubés au sein de l'organisation, permettant ainsi aux parties concernées de collaborer efficacement et de suivre l'avancement des projets.

❖ CU03 : Valider une ressource

Le résumé de ce cas d'utilisation se trouve dans le tableau 7.

Résumé	Ce cas d'utilisation permet à l'utilisateur de valider une ressource du projet
Acteurs	Coach, Mentor
Précondition	<ul style="list-style-type: none"> • L'utilisateur est connecté en tant que coach ou mentor. • Un projet existe avec des ressources ajoutées par un porteur de projet.
Scénario Nominal	<ul style="list-style-type: none"> • L'utilisateur se connecte à la plateforme en tant que coach ou mentor. • L'utilisateur accède au projet pour lequel il agit en tant que coach ou mentor. • L'utilisateur accède à la liste des ressources du projet. • L'utilisateur sélectionne une ressource à valider. • Le système affiche les détails de la ressource, y compris son type, son titre, sa description, son lien, etc. • L'utilisateur examine la ressource et décide de la valider. • L'utilisateur sélectionne l'option "Valider" sur la ressource. • Le système enregistre la validation de la ressource et met à jour son statut en tant que "validée". • Le système génère une notification pour informer le porteur de projet que sa ressource a été validée.
Post conditions	<ul style="list-style-type: none"> • La ressource est validée et affiche son statut comme "validée" dans la liste des ressources du projet. • Le porteur de projet est notifié de la validation.

Tableau 7 : Description du cas d'utilisation "Valider une ressource"

Le tableau 7 décrit le cas d'utilisation permet aux coaches et aux mentors de valider les ressources ajoutées par les porteurs de projet dans le système. Les ressources peuvent être des documents, des liens ou d'autres types de fichiers. L'acteur principal, qui est un coach ou un mentor, peut examiner chaque ressource, décider de sa validation, et si nécessaire, laisser des commentaires. Une fois validée, la ressource est considérée comme approuvée et peut être utilisée dans le cadre du projet.

La description textuelle expose les interactions entre les différents acteurs du système de manière statique, mais l'on a aussi les diagrammes de séquence qui décrivent dynamiquement ces interactions. Nous présentons ci-dessous les diagrammes de séquence illustrés respectivement dans les figures 4,5 et 6 qui décrivent les aspects dynamiques des cas d'utilisation « S'authentifier », « Ajouter un projet » et « Valider une ressource ».

e. Diagramme de séquence

Les diagrammes de séquences représentent les interactions entre le système et les utilisateurs en montrant sous forme de scénarios la chronologie des envois de messages issus d'un cas d'utilisation donné. Les diagrammes de séquences qui sont décrits dans cette partie sont essentiellement : authentification, ajouter un projet, valider une ressource.

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

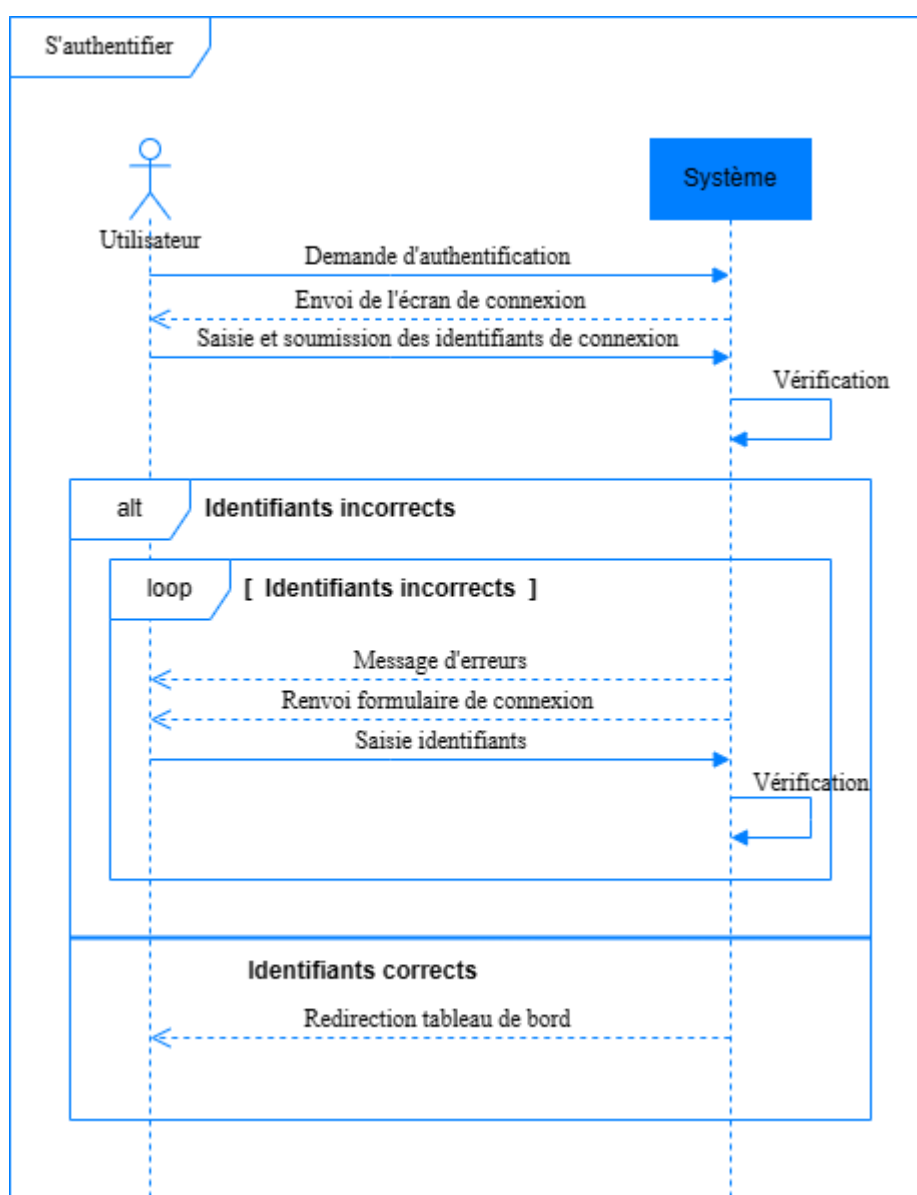


Figure 4 : Diagramme de séquence du cas d'utilisation « S'authentifier »

PROBLÈME AU NIVEAU DU PDF La figure 4 présente graphiquement les messages échangés au niveau du cas d'utilisation « S'authentifier ». L'utilisateur demande à se connecter au système, qui lui renvoie un formulaire pour saisir son email et son mot de passe pour se connecter. Les informations saisies sont vérifiées à deux (02) niveaux par le système. Il faut que les informations saisies respectent la syntaxe d'une adresse email et que le mot de passe soit composé au minimum de quatre (04) caractères. Si les normes ne sont pas respectées, l'utilisateur reçoit une notification d'erreur avec le message « Syntaxe non valide ». Au cas où les informations saisies sont syntaxiquement valides, une deuxième vérification est faite concernant la compatibilité des informations dans le système. En cas d'informations incompatibles, l'utilisateur reçoit un message d'erreur « Identifiants incorrectes ». Dans le cas contraire ; l'utilisateur s'est authentifié avec succès et accède à la page d'accueil qui lui est réservée.

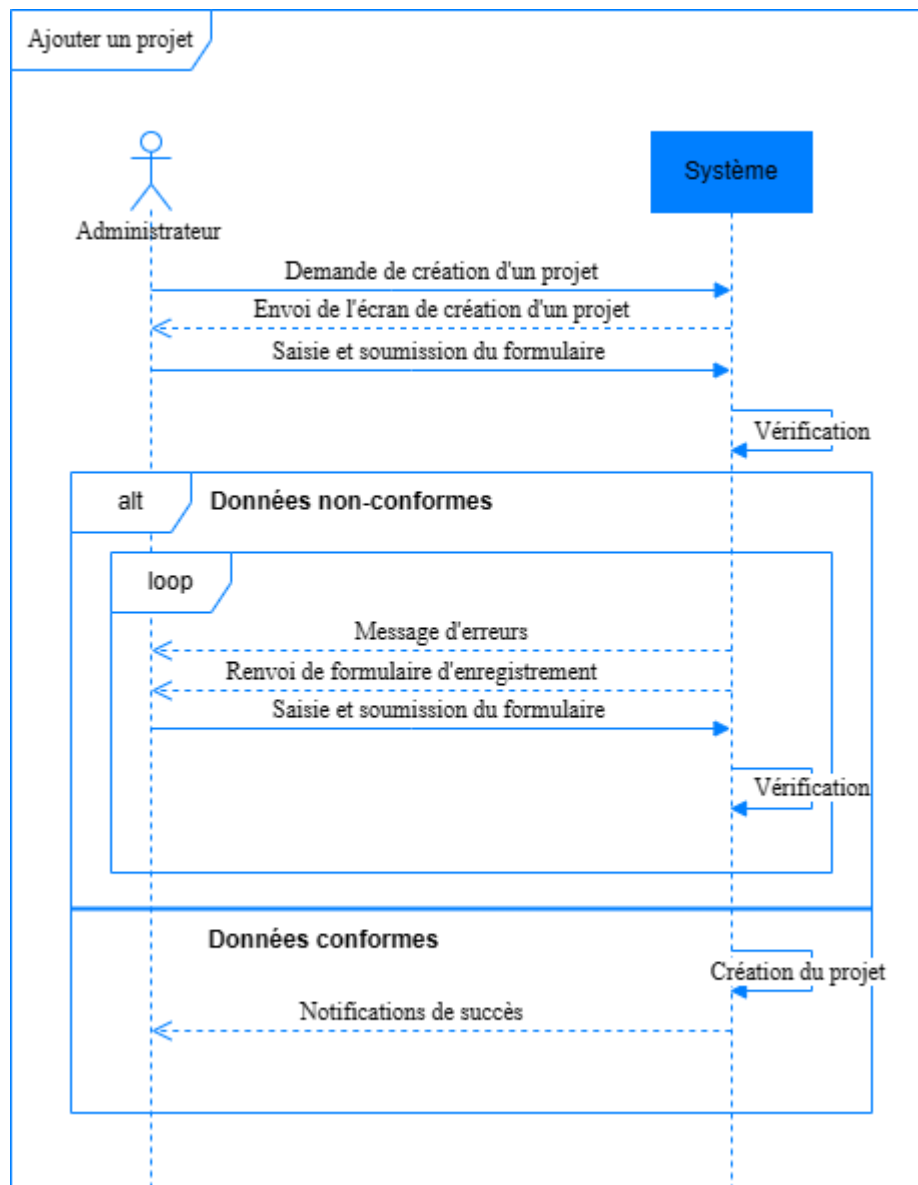


Figure 5 : Diagramme de séquence du cas d'utilisation « Ajouter un projet »

La figure 5 présente graphiquement les messages échangés au niveau du cas d'utilisation « Ajouter un projet ». L'utilisateur déjà authentifié en tant qu'administrateur demande à ajouter un projet et le système lui renvoie un formulaire à remplir. Après la soumission du formulaire, le système fait une vérification du nom du projet. Au cas où les informations saisies sont compatibles aux informations existant dans le système, l'utilisateur reçoit une notification d'erreur « Données existent dans le système » et le formulaire d'ajout d'un projet lui est renvoyé. Dans le cas contraire, le nouveau projet est enregistré et l'utilisateur reçoit un message de succès et est redirigé vers la page actualisée de la liste des utilisateurs

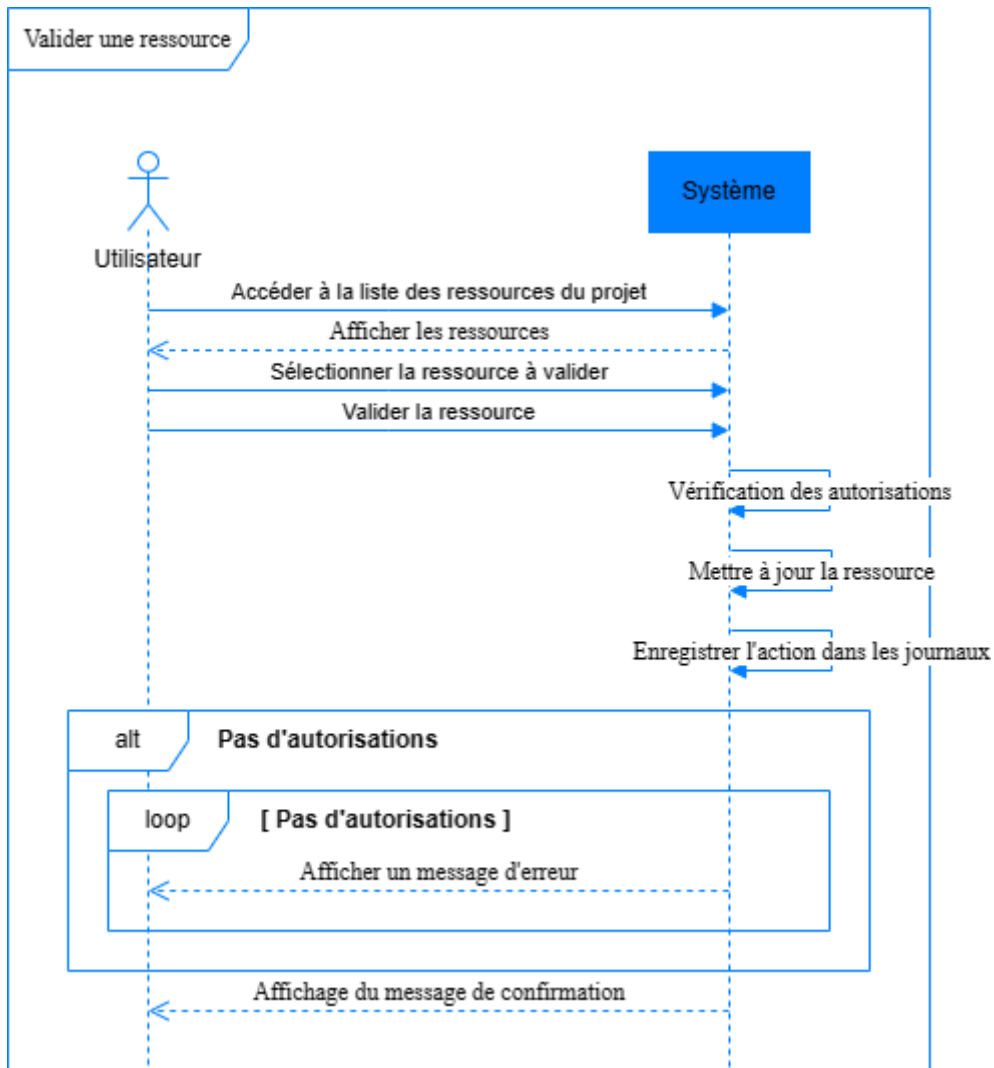


Figure 6 :Diagramme de séquence du cas d'utilisation « Valider une ressource »

La figure 6 illustre le processus par lequel un utilisateur, qu'il s'agisse d'un coach, ou d'un mentor vérifie et valide une ressource envoyée par un porteur de projet dans le système. Le flux commence par l'utilisateur sélectionnant la ressource à valider, puis le système affiche les détails de cette ressource. Une fois les détails affichés, le système fait une vérification des autorisations du projet, met à jour l'état de la ressource et enregistre l'action dans les journaux et passe à l'étape de validation de la ressource. Si la ressource est marquée comme valide, le système affiche un message de confirmation. Si la ressource est marquée comme invalide, le système affiche un message d'erreur. Dans les deux cas, un enregistrement de l'action est conservé pour la traçabilité. Ce diagramme de séquence détaille le processus de validation des ressources, garantissant ainsi leur qualité au sein de la plateforme.

Nous venons d'élucider les différentes fonctionnalités du projet. Mais dans quel environnement technique le logiciel va-t-il fonctionner ? Dans la suite du travail, nous abordons la spécification technique du système de gestion des programmes d'activités de Sira Labs.

5. Spécifications techniques

Tandis que la spécification fonctionnelle aborde les fonctionnalités développées pour la réalisation du logiciel, la spécification technique concerne le moyen technique tel que l'architecture, la technologie ou le matériel pour mettre en œuvre ces fonctionnalités.

a. Mise à disposition des conditions de travail

Sira Labs a mis à notre disposition :

- une connexion internet haut débit pour nos recherches ;
- 15 ordinateurs : Sira Labs possède environ plusieurs ordinateurs de service dans ses différents sièges ;
- des locaux ouverts et disponibles 24h /24 ;
- serveurs : hébergement mutualisé sur AWS¹⁴ ;
- site web de Sira Labs ;
- document descriptif du fonctionnement de Sira Labs ;

La mise à disposition des conditions de travail crée un environnement propice à la planification et à la mise en œuvre 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

Une architecture en informatique désigne un mode de communication à travers un réseau entre plusieurs éléments physiques et/ou logiques. Il existe plusieurs types d'architectures de développement dont nous faisons une étude comparative de quelques-uns dans le tableau 8.

¹⁴ [Services et produits de cloud Amazon | AWS](#)

Architecture	Avantages	Inconvénients
Architecture un tiers	les tables des données sont directement accessibles.	<ul style="list-style-type: none"> • Maintenance difficile du système • surcharge la machine hôte
Architecture deux tiers	<ul style="list-style-type: none"> • applications multi-utilisateurs • charges réparties entre le client et le serveur • bonne sécurité des données 	<ul style="list-style-type: none"> • coût de déploiement élevé • évolution difficile du système
Architecture trois tiers	<ul style="list-style-type: none"> • évolution facile • déploiement aisé • une sécurité accrue deux tiers. 	<ul style="list-style-type: none"> • nécessite un personnel informatique initié pour sa mise en œuvre • recommandé dans un réseau avec une connexion haut débit

Tableau 8 : Comparaison des différentes architectures de développement

La figure 7 ci-dessous illustre parfaitement les informations données dans le tableau sur l'architecture deux tiers.

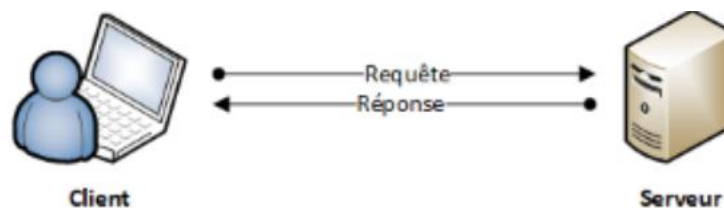


Figure 7:Schéma représentant l'architecture deux tiers [4]

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

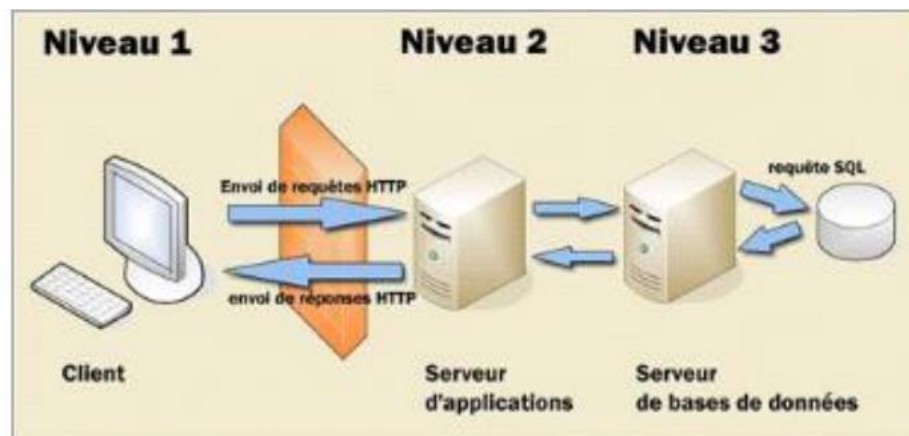


Figure 8:Schéma représentant l'architecture 3 tiers [4]

A la suite de l'étude comparative menée dans le tableau 8, notre choix s'est porté sur l'architecture trois tiers. En effet, cette dernière offre une facilité de déploiement, une expérience utilisateur enrichie et est celle que nous avons le mieux assimilée et que nous maîtrisons. Dans les pages suivantes du document, nous présentons la conception globale du système de gestion des projets incubés à Sira Labs.

III. CONCEPTION GLOBALE

Dans cette partie, nous mettrons en évidence le dictionnaire de données, le diagramme des classes, le diagramme d'activité des cas d'utilisation « S'authentifier », « Ajouter un projet » et « Valider une ressource », ainsi que celui du déploiement.

1. Diagramme des classes

Dans cette partie, nous présentons le dictionnaire des données puis le diagramme des classes.

a. Dictionnaire des données

Le dictionnaire des données contient l'ensemble des descriptions des attributs des données utilisées dans le système. Le tableau 9 montre l'ensemble des attributs utilisés dans le développement du système de gestion des projets incubés à Sira Labs.

Attribut	Description	Type de données
idProjet	Identifiant du projet	Entier
name	Nom du projet	Chaine de caractère
desc	Description du projet	Chaine de caractère
start_date	Date du début	date
end_date	Date du fin	date
Porteur_de_projet	Nom du porteur de projet	Chaine de caractère
coach	Nom du coach	Chaine de caractère
Is_active	Vérifie si le projet est activé	booléen
Project_state	L'état du projet	Chaine de caractère
Current_phase	Phase actuelle du projet	Chaine de caractère
idResource	Identifiant de la ressource	Entier
title	Titre de la ressource	Chaine de caractère
desc	Description de la ressource à ajouter	Chaine de caractère
type	Type de ressource	Chaine de caractère
file	Le lien du fichier à uploader	Chaine de caractère
link	Le lien du fichier à ajouter à la ressource	Chaine de caractère
validated	Vérifie si le a ressource est validée	Booléen
created_at	Date de création de la ressource	date

validation_phase	Détermine la phase de validation du projet	Chaine de caractère
idTask	Identifiant de la tâche	Entier
name	Nom du tâche	Chaine de caractère
desc	Description de la tâche à ajouter	Chaine de caractère
status	Vérifie le statut de la tâche	Chaine de caractère
deadline	Date de fin de la tâche	date
validation_phase	Détermine la phase de validation du projet	Chaine de caractère
username	Nom de l'utilisateur	string
email	Mail de l'utilisateur	string
phone	Numéro de l'utilisateur	
address	Adresse de l'utilisateur	string
expertise	Domaine d'expertise de l'utilisateur	string
isadmin	Vérifie si l'utilisateur est un administrateur	booléen
iscoach	Vérifie si l'utilisateur est un coach	booléen
ismentor	Vérifie si l'utilisateur est un mentor	booléen
isprojectowner	Vérifie si l'utilisateur est un porteur de projet	booléen
Key-segment	Détermine les segments clés	string
valueproposition	Détermine les canaux de diffusion	string
channels	Détermine les canaux de diffusion	string
Customer-relation	Détermine la relation client	string
moneysource	Détermine la source des revenus du projet	string
Key-resource	Détermine les ressources clés du projet	string
Key-activities	Détermine les activités clés du projet	string
Key-partners	Détermine les partenaires clés du projets	string
money-structure	Détermine le modèle économique	string
content	Contenu d'un commentaire	string
date	date d'un commentaire	date

Tableau 9 : Dictionnaire de données

Le dictionnaire des données présente l'ensemble des attributs utilisés dans le développement du système. Mais comment ces données sont-elles organisées dans notre base de données ? Nous abordons dans les lignes qui suivent, le diagramme des classes qui concerne l'organisation logique des données de notre système.

b. Diagramme des classes

Les données présentées au niveau du dictionnaire des données peuvent être organisées en entités encore appelées classes. Nous avons par exemple les classes « Projet », « Ressource » et « Phase ».

La classe « Projet » représente les informations générales sur un projet, telles que son nom, la date de début et de fin prévue, le porteur de projet associé, et l'état actuel du projet. La classe "Phase" représente les différentes étapes du projet, comme la phase de conception, de développement, ou de validation. La classe "Ressource" représente les éléments ajoutés au projet par le porteur de projet, comme des documents, des liens, ou d'autres fichiers. Les relations entre ces entités sont définies par des associations. Un projet peut avoir plusieurs phases, mais une phase appartient à un seul projet (relation un-à-plusieurs). De même, un projet peut contenir plusieurs ressources, mais une ressource est spécifique à un seul projet (relation un-à-plusieurs). Ces relations décrivent les relations entre quelques entités clés du système et sont cruciales pour le logiciel de gestion et de collaboration des projets incubés à Sira Labs.

Le diagramme de classe est la représentation visuelle de la structure d'un système logiciel, mettant en évidence les classes, les attributs et les relations entre elles. Chaque classe est représentée par un rectangle divisé en trois parties. La première partie affiche le nom de la classe, la deuxième liste ses attributs, et la troisième ses méthodes. Les flèches indiquent les relations entre les classes, avec des annotations pour préciser la nature de ces relations (comme l'agrégation, la composition, ou l'association). Les multiplicateurs (*, 1) indiquent le nombre d'objets associés. Un diagramme de classe fournit une vue structurée et visuelle du système, facilitant la compréhension de ses composants et de leurs interactions.

Pour avoir une visualisation globale des classes, nous présentons le diagramme des classes du système de gestion de projets à la figure 9. **Diagramme non lisible**

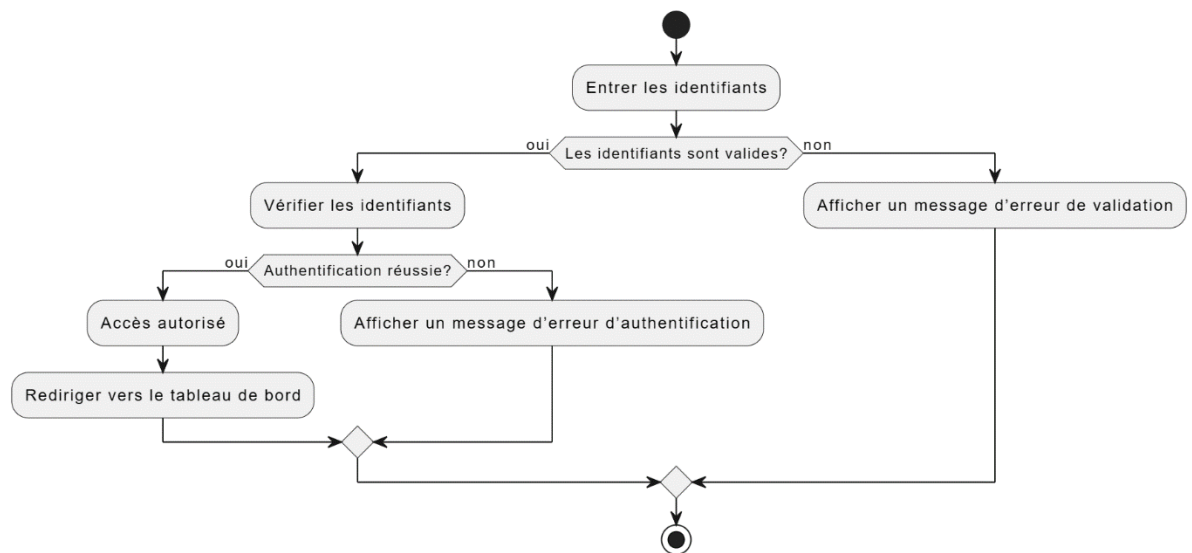


Figure 10:Diagramme d'activité du cas d'utilisation « S'authentifier »

Ce diagramme d'activité détaille le flux complet du cas d'utilisation « S'authentifier » et illustre les différentes étapes impliquées dans le processus.

L'utilisateur démarre en entrant ses identifiants. Ces identifiants sont ensuite validés, vérifiant s'ils sont corrects. En cas de validation réussie, le système procède à la vérification de l'authentification. Si l'authentification est réussie, l'utilisateur obtient un accès autorisé et est redirigé vers le tableau de bord. En revanche, si la validation des identifiants échoue ou si l'authentification échoue, des messages d'erreur appropriés sont affichés pour informer l'utilisateur. Le processus se termine à ce stade, illustrant le flux complet de l'authentification, de la vérification des identifiants à l'accès autorisé ou à l'affichage d'un message d'erreur.

Après avoir authentifié l'utilisateur avec succès, le système le redirige vers le tableau de bord où il peut ajouter un nouveau projet. La figure 11 illustre le cas d'utilisation « Ajouter un nouveau projet ».

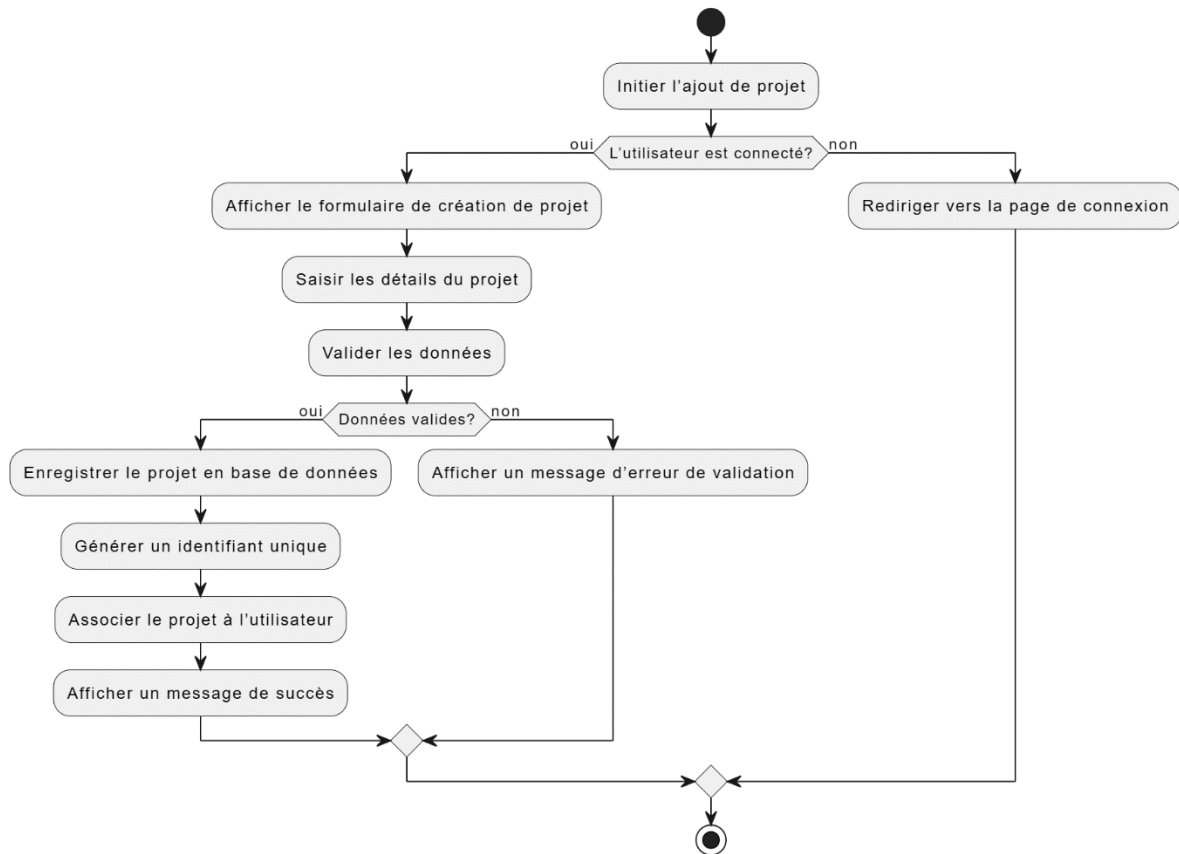


Figure 11: Diagramme d'activité du cas d'utilisation « Ajouter un projet »

Ce diagramme d'activité présenté dans la figure 11 commence par l'initiation de l'ajout de projet par l'utilisateur. Il vérifie d'abord si l'utilisateur est connecté. Si c'est le cas, il affiche le formulaire de création de projet, permet à l'utilisateur de saisir les détails du projet, puis valide les données. Si les données sont valides, le projet est enregistré en base de données, un identifiant unique est généré, le projet est associé à l'utilisateur, et un message de succès est affiché.

Tout comme l'ajout d'un nouveau projet, la validation des ressources est un élément clé pour le système de gestion. La figure 12 illustre le cas d'utilisation « Valider une ressource ».

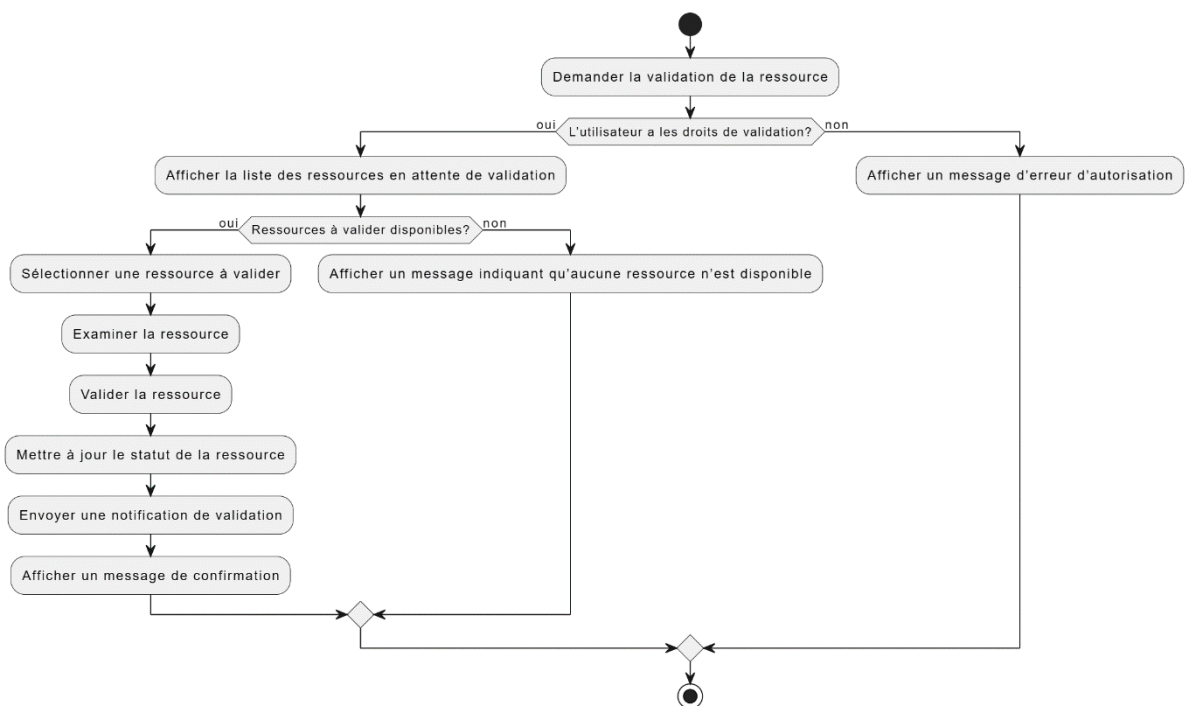


Figure 12 : Diagramme du cas d'utilisation « Valider une ressource »

Ce diagramme d'activité détaille le flux complet du cas d'utilisation « Valider une ressource » et illustre les différentes étapes impliquées dans le processus.

Il commence par vérifier d'abord si l'utilisateur possède les droits de validation de la ressource, si c'est le cas, l'acteur explore la liste des ressources associées au projet et sélectionne une ressource à valider. Le système affiche les détails de la ressource, permettant à l'acteur d'examiner son contenu. L'acteur prend alors la décision de valider la ressource, sélectionne l'option « Valider », et le système enregistre cette validation, mettant à jour le statut de la ressource comme « validée ». Une notification est générée pour informer le porteur de projet que sa ressource a été validée. Si l'utilisateur ne possède pas les droits de validations du projet, le système affiche un message d'erreur d'autorisation. Ce processus assure une validation transparente et efficace des ressources au sein de la plateforme, facilitant la collaboration entre les différents intervenants du projet.

Après avoir illustré les flux dynamiques du projet à travers le diagramme d'activité, plongeons maintenant dans l'infrastructure physique qui soutient cette interaction avec le passage au diagramme de déploiement

3. 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.

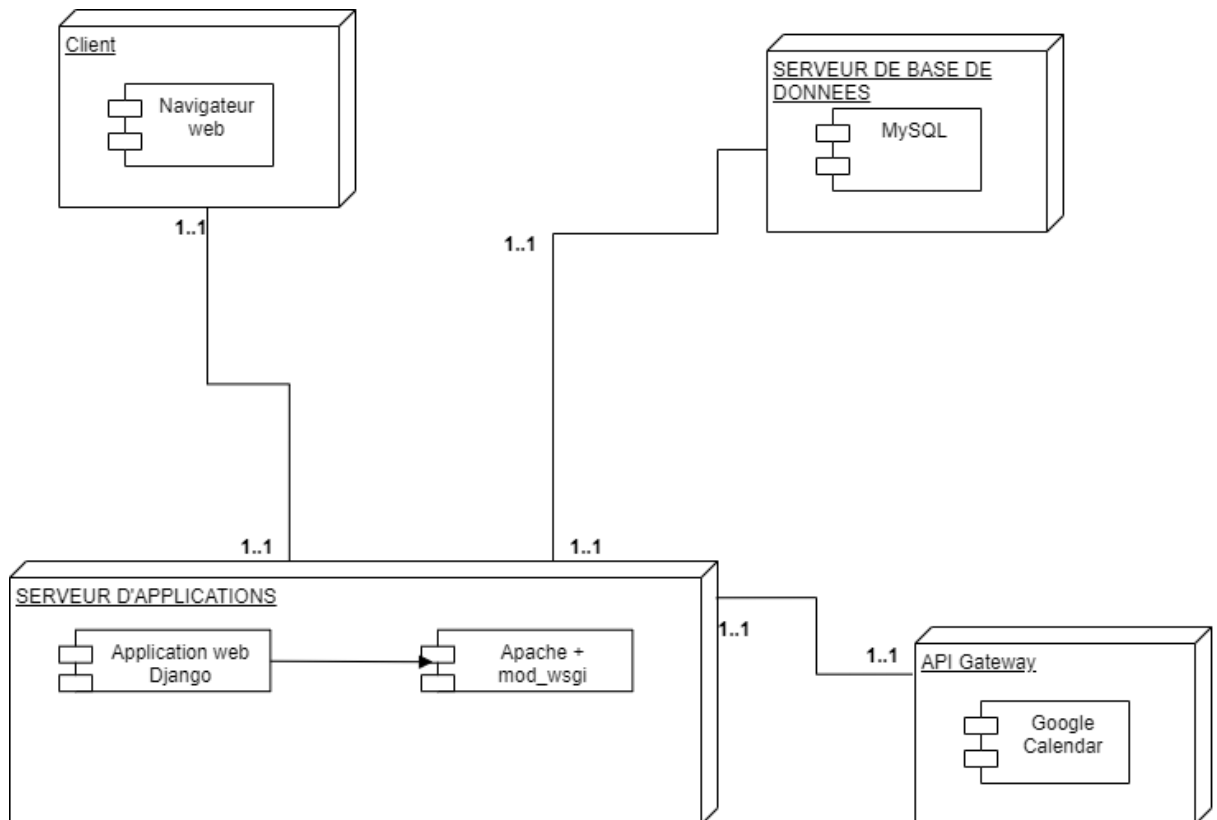


Figure 13 : Diagramme de déploiement

Notre diagramme de déploiement comprend quatre (4) nœuds qui représentent le serveur de données, le serveur d'application, le poste client et l'API Gateway ou passerelle API. En effet l'API Google Calendar est une interface de programmation applicative (API) fournie par Google qui permet aux développeurs d'accéder et de manipuler les données du service Google Agenda [7].

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

IV. REALISATION

Dans le cadre de la réalisation de ce système informatisé, 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 de réalisation

Les outils de réalisation sont des éléments qui aident un développeur dans le déroulement d'une activité de développement.

a. Choix du Système de Gestion de Base de Données

Il existe plusieurs SGBD dont :

- Microsoft Office Access ¹⁵;
- Microsoft SQL Server ¹⁶;
- MySQL ¹⁷ ;
- Oracle Database ¹⁸;
- PostgreSQL ¹⁹.

La liste comparative de ces SGBD est dans le tableau suivant :

SGBD	Avantages	Inconvénients
Microsoft Office Access	<ul style="list-style-type: none">• Facile à utiliser et à apprendre• Intégré à la suite Microsoft Office• Supporte les requêtes SQL	<ul style="list-style-type: none">• Limité en termes de capacité, de performance et de sécurité• Non adapté aux applications web ou aux environnements multi-utilisateurs• Peu de support technique et de documentation
Microsoft SQL Server	<ul style="list-style-type: none">• Puissant, fiable et évolutif• Compatible avec les normes SQL et les langages de programmation courants• Offre des outils d'analyse, de reporting et d'intégration de données	<ul style="list-style-type: none">• Coûteux et complexe à installer, configurer et maintenir• Dépendant du système d'exploitation Windows• Vulnérable aux attaques et aux failles de sécurité
MySQL	<ul style="list-style-type: none">• Gratuit, open source et populaire• Rapide, performant et flexible	<ul style="list-style-type: none">• Moins de fonctionnalités avancées que d'autres SGBD

¹⁵ [Logiciels et applications de base de données | Microsoft Access](#)

¹⁶ [Téléchargements SQL Server | Microsoft](#)

¹⁷ [MySQL](#)

¹⁸ [Base de données | Oracle](#)

¹⁹ [PostgreSQL : Téléchargements](#)

	<ul style="list-style-type: none"> • Supporte les applications web et les environnements multi-utilisateurs 	<ul style="list-style-type: none"> • Moins conforme aux normes SQL que d'autres SGBD • Moins de support technique et de documentation que d'autres SGBD
Oracle Database	<ul style="list-style-type: none"> • Leader du marché des SGBD relationnels • Robuste, sécurisé et évolutif • Offre des fonctionnalités avancées et innovantes 	<ul style="list-style-type: none"> • Très coûteux et complexe à utiliser • Nécessite des ressources matérielles importantes • Peut être difficile à migrer vers d'autres SGBD
PostgreSQL	<ul style="list-style-type: none"> • Gratuit, open source et respectueux des normes SQL • Supporte les données relationnelles et non relationnelles • Offre des extensions pour étendre ses fonctionnalités 	<ul style="list-style-type: none"> • Moins rapide et performant que d'autres SGBD • Moins compatible avec les langages de programmation courants que d'autres SGBD • Moins populaire et moins soutenu que d'autres SGBD

Figure 14: Liste comparative des SGBD [5]

A travers l'étude comparative que nous venons de mener au Tableau 14, nous portons notre choix sur le SGBD MySQL. En effet, MySQL le SGBD utilisé dans l'entreprise dans laquelle nous avons effectué notre stage. Aussi, il offre une interface web, « phpMyAdmin » pour l'administration de ses bases de données. Il est aussi celui que nous maîtrisons le mieux et qui est convenable à notre projet.

Dans la continuité de la présentation des outils de réalisation, nous aborderons maintenant un aspect fondamental : les langages de programmation.

b. Langages

Les langages de programmation sont les outils essentiels qui permettent aux développeurs de donner vie à des idées, de créer des logiciels innovants et de résoudre une vaste gamme de problèmes informatiques. Pour le développement de notre application nous avons eu recours à plusieurs langages informatiques dont :

- **Python**²⁰ est un langage de programmation interprété, polyvalent et de haut niveau. Il est populaire en raison de sa simplicité, de sa polyvalence et de sa large utilisation dans divers domaines, de la création d'applications web à l'analyse de données en passant par l'automatisation des tâches.
- **JavaScript**²¹ est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.

²⁰ [Bienvenue chez Python.org](#)

²¹ [JavaScript - Apprendre le développement web | MDN \(mozilla.org\)](#)

- **HTML5**²² (HyperText Markup Language) : est un langage de marquage et de balisage servant à écrire des pages pour le World Wide Web. (Pour les pages web)
- **CSS3**²³ (Cascading Style Sheet) : le CSS permet de faire la mise en forme des pages web.

c. Serveur d'application

Un serveur d'application est chargé d'héberger une application. Il permet la communication avec la base de données. L'objectif d'un serveur d'application est de permettre à partir d'un client aussi léger que possible d'effectuer des traitements sur une machine distante. Les utilisateurs y accèdent par le biais d'un navigateur.

- uWSGI²⁴
- Oracle WebLogic Server²⁵
- CherryPy²⁶
- Nginx (proxy inverse)²⁷
- Apache + mod_wsgi²⁸

La liste comparative de ces serveurs d'applications se trouve dans le tableau 10.

Serveur d'application	Mémoire	Démarrage	Configuration	Communauté	Prix	Composants
uWSGI	Modérée	Rapide	Modérée	Forte	Gratuit	HTTP Server, Load Balancer, Cache
Oracle WebLogic Server	Légère	Rapide	Facile	Très forte	Payant	Serveur complet avec tous les composants
CherryPy	Modérée	Modérée	Facile	Petite	Gratuit	HTTP Server, Framework Web
Nginx (proxy inverse)	Modérée	Rapide	Facile	Forte	Gratuit	HTTP Server, Proxy Inverse, Load Balancer
Apache + mod_wsgi	Modérée	Rapide	Très facile	Très forte	Gratuit	Serveur complet

Tableau 10 : Tableau comparatif des serveurs d'applications

²² [Tutoriel HTML \(w3schools.com\)](http://w3schools.com/html/html5_tutorial.asp)

²³ [Tutoriel CSS \(w3schools.com\)](http://w3schools.com/css/css3_tutorial.asp)

²⁴ [Le projet uWSGI — Documentation uWSGI 2.0 \(uwsgi-docs.readthedocs.io\)](http://uwsgi-docs.readthedocs.io/en/latest/)

²⁵ [Serveur Weblogic | Oracle](http://www.oracle.com/technetwork/weblogic/index.html)

²⁶ [CherryPy — Un framework Web Python minimaliste](http://cherrypy.org/en/latest/)

²⁷ [Proxy inverse NGINX | NGINX Documentation](http://nginx.org/en/docs/)

²⁸ [mod_wsgi — mod_wsgi 4.9.4 documentation \(modwsgi.readthedocs.io\)](http://modwsgi.readthedocs.io/en/4.9.4/)

A travers l'étude comparative que nous venons de mener au Tableau 10, notre choix porte sur Apache + mod_wsgi car le serveur Apache est un logiciel libre et gratuit, ce qui est idéal pour nous étant donné que nos ressources sont limitées. Aussi, il est celui que nous maîtrisons le mieux et qui est convenable à notre projet sur le plan technique.

Dans la continuité de la présentation des outils de réalisation, nous présenterons dans la partie suivante, les plateformes de développement.

d. Framework

Un framework appelé aussi infrastructure logicielle désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel. Pour atteindre notre objectif, nous avons utilisé plusieurs frameworks qui sont :

- **Django**²⁹ : Django est un framework web open-source écrit en Python qui permet de développer rapidement des applications web robustes, sécurisées et évolutives.
- **Bootstrap 4**³⁰ : Bootstrap est un Framework front-end gratuit pour un développement Web plus rapide et plus facile. Il comprend des modèles de conception basés sur HTML et CSS.

e. Autres outils

Les outils de conception désignent l'ensemble des outils utilisés depuis la conception jusqu'à la réalisation du projet. Ce sont :

- **Draw io**³¹ : c'est un outil de conception ; il nous a donc permis de concevoir nos différents diagrammes UML.
- **GanttProject**³² : il a servi à mettre en place notre diagramme de GANTT.
- **Figma**³³ : c'est un outil de prototypage de site web et d'application ; il nous a permis de concevoir les maquettes des différents écrans de notre application
- **PhpMyAdmin version 4**³⁴ : il nous a été utile pour administrer notre base de données.

²⁹ [Le framework web pour les perfectionnistes avec des délais | Django \(djangoproject.com\)](https://djangoproject.com/)

³⁰ [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](https://getbootstrap.com/)

³¹ [Créateur d'organigrammes et logiciel de diagramme en ligne \(diagrams.net\)](https://diagrams.net/)

³² [GanttProject - Application gratuite de gestion de projet](https://ganttproject.com/)

³³ [Avec Figma, le webdesign prend une dimension collaborative](https://figma.com/)

³⁴ [phpMyAdmin - Téléchargements](https://www.phpmyadmin.net/)

- **Git³⁵** : Git est un logiciel de gestion de versions décentralisé. Il nous a permis de gérer et archiver le code source de la plateforme.
- **Visual Studio Code ³⁶**: c'est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré
- **Microsoft Office Word et Power Point 2016 ³⁷**: ce sont des outils efficaces pour le traitement des textes. Ils nous ont été utiles dans la rédaction de la documentation nécessaire à la connaissance et à l'utilisation de l'application.

2. Politique de sécurité

La politique de sécurité est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système informatique contre les menaces accidentelles ou intentionnelles auxquelles il peut être confronté. Pour le cas de notre système, cette politique s'articule autour des mesures de sécurité mises en œuvre dans le processus de développement du système, des mesures à implémenter lors de sa mise en production et après sa mise en production.

a. Mesures de sécurité prises pendant le développement

Le service de sécurité implémenté pour notre application est le mécanisme d'authentification présenté au niveau de la figure 4, diagramme de séquence du cas d'utilisation « S'authentifier » L'accès à l'application a été protégé par un nom d'utilisateur et un mot de passe. En outre, un utilisateur qui se connecte ne peut exécuter que les fonctionnalités du système que lui confère son profil. L'authentification des utilisateurs sera gérée en utilisant le système d'authentification de Django.

En plus de cela, des restrictions sont faites sur un certain nombre de pages et d'informations. De ce fait, un utilisateur qui se connecte ne peut accéder qu'aux données auxquelles ils disposent les droits nécessaires.

b. Mesures de sécurité à prendre pour l'exploitation

Le système de gestion de projets incubés sera hébergé chez un hébergeur externe et les mesures de sécurité sur le serveur de l'hébergeur sont celles implémentées par l'hébergeur

³⁵ [GitHub: Construisons à partir d'ici · GitHub](#)

³⁶ [Télécharger Visual Studio Code - Mac, Linux, Windows](#)

³⁷ [Édition de documents en ligne gratuite avec Microsoft Word | Microsoft 365](#)

externe. Cette approche garantit une infrastructure robuste et sécurisée, tirant parti des normes de sécurité de pointe appliquées par l'hébergeur externe. Ainsi, les données et les opérations du système seront soumises aux protocoles de sécurité rigoureux établis par cet hébergeur professionnel. Cette configuration vise à assurer la confidentialité, l'intégrité et la disponibilité des informations, offrant ainsi un environnement fiable pour le système de gestion des projets incubés.

Après la présentation de la politique de sécurité, voyons à présent à combien est estimé le coût de réalisation de notre système de gestion des projets.

c. Autres mesures de sécurité

Le facteur humain étant important dans la démarche sécuritaire alors il est nécessaire de fournir à nos clients des conseils pratiques pour un meilleur rendement du service fournie par la plateforme. Ainsi, le client peut se prémunir de plusieurs tords en adoptant les démarches suivantes : éviter de communiquer ses identifiants à un tiers, ne pas céder son ordinateur sans aucune surveillance de l'usage que l'on en fait, se procurer d'anti-virus régulièrement mis à jour et éviter au maximum d'accéder aux sites web frauduleux ou non muni d'un certificat SSL.

3. 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 dont on peut retenir la méthode d'estimation par analogie, le jugement d'experts et la méthode COCOMO. Le tableau 12 présente une étude comparative des différentes méthodes d'estimation de coût de développement logiciel.

Méthodes d'estimations	Principes	Avantages	Inconvénients
Estimation par analogie	Le coût du projet est calculé en le comparant à d'autres projets similaires dans le même domaine d'application.	Estimations précises si les données concernant des projets similaires sont disponibles.	Impossible si des projets similaires n'étaient pas réalisés
Jugement d'experts	Un ou plusieurs experts à la fois utilisent leur expertise pour faire des prévisions de coût.	Estimations précises si les experts ont de l'expérience dans des projets similaires.	Estimations non précises s'il y a peu d'expertise.
La méthode COCOMO	Il s'appuie uniquement sur la taille estimée du logiciel et sur le type de logiciel à développer.	Estimations précises, très populaires et faciles à mettre en œuvre avec divers facteurs.	Ignore les contraintes matériels

Tableau 11 : Comparaison des méthodes d'estimation de cout et de développement [6]

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

Nous estimons à 6000 le nombre moyen de lignes de code de notre application qui est de complexité S. 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 :

$$HM = 2,4 * KLS^{1,05} = 2,4 * (6000/1000)^{1,05} = 15,75 \text{ Homme/Mois}$$
- Temps de développement :

$$TDEV = 2,5 * \text{Effort}^{0,38} = 2,5 * 15,75^{0,38} = 7,12 \text{ mois}$$
- Coût de développement :

$$CDEV = 15,75 * X \text{ FCFA} = 15,75 * 200\,000 \text{ FCFA} = 3\,150\,000 \text{ FCFA}$$

Voir vocal 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 dans le tableau 12.

Désignation	Quantité	Coût unitaire (FCFA)	Coût total (CFA)
Ordinateur	1	-	3 00000
Draw io	1	-	Licence gratuite
GanttProject	1	-	Licence gratuite
Visual studio code	1	-	Licence gratuite
MySQL	1	-	Licence gratuite
Développement	-	-	3150 000
Cout total			3 450 000

Tableau 12 : Cout total du projet

4. Présentation de quelques interfaces graphiques de l'application

Dans cette section nous faisons la description de quelques interfaces du système de gestion des projets de Sira Labs.

a. Présentation de l'interface administrateur

Dans cette partie nous faisons la description de quelques interfaces de l'administration du système de gestion des projets de Sira Labs.

La figure 15 nous montre l'écran qui permet de se connecter à l'interface administrateur de l'application.



Figure 15 : Ecran de connexion à l'interface administrateur

La figure 16 nous montre l'écran permettant de visualiser l'application dans sa globalité. Depuis cette page, l'administrateur peut voir les activités effectuées récemment sur la plateforme et les différents utilisateurs et projets.

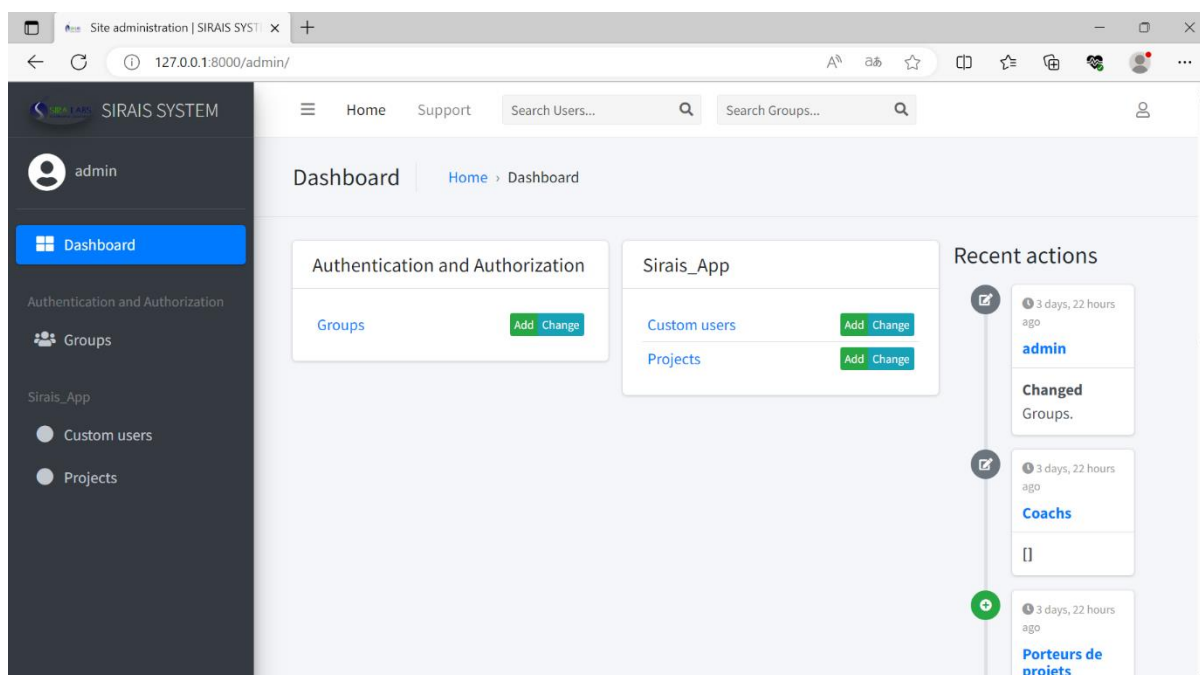


Figure 16 : Ecran de visualisation de l'interface administrateur

La figure 17 nous montre l'écran qui permet de créer les utilisateurs à partir de l'interface administrateur.

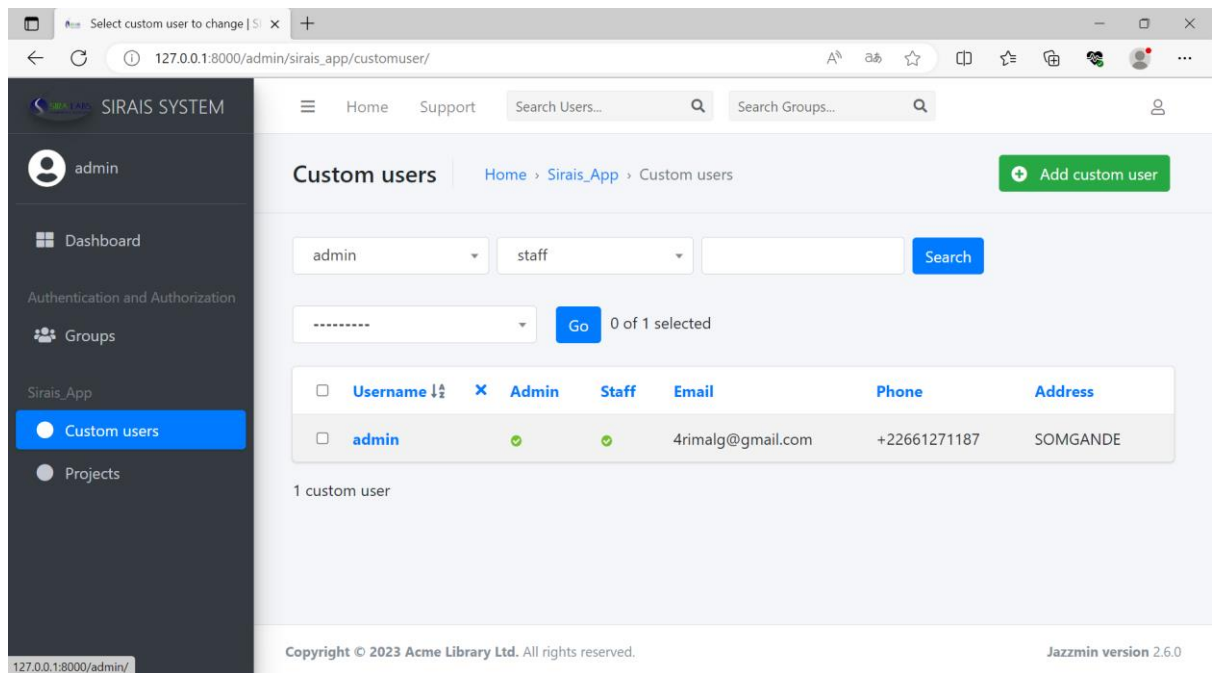


Figure 17 : Interface administrateur qui permet de créer les utilisateurs de l'application

b. Présentation de l'interface client

Dans cette partie nous faisons la description de quelques interfaces destinées au porteur de projet, au coach et au mentor d'un projet du système de gestion des projets de Sira Labs.

La figure 18 nous montre l'écran qui permet de se connecter à l'interface client de l'application.



Figure 18 : Interface de connexion de l'application

La figure 19 nous présente l'écran qui affiche la liste des projets sur l'application client. A partir de cette interface, le porteur / coach /mentor, peut voir la liste des projets auxquels il est associé et vérifier les détails de celui-ci, il peut également supprimer et voir les ressources associées au projet au question.

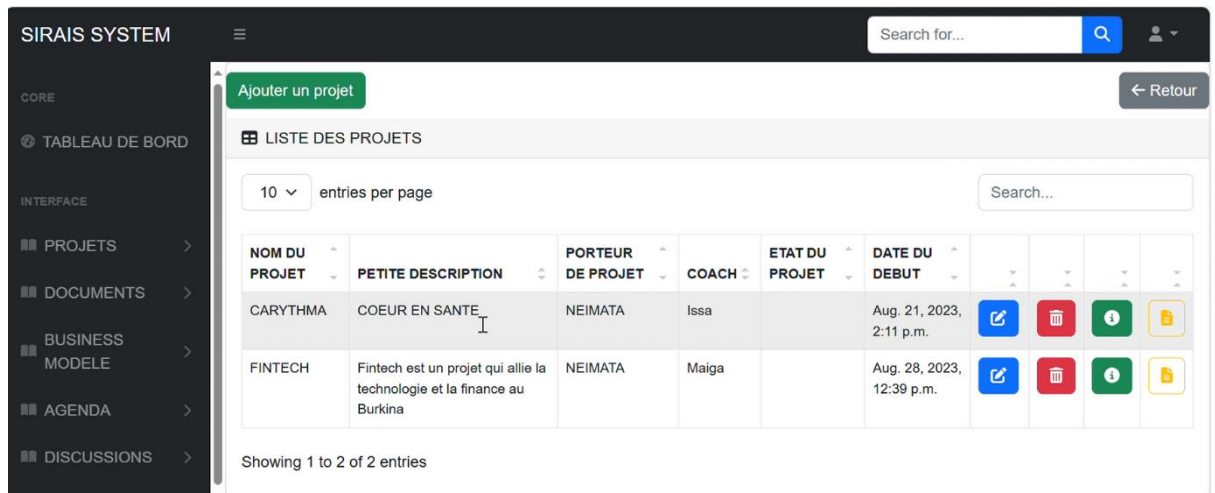


Figure 19: Liste des projets

La figure 20 montre le formulaire de création d'un projet incluant les date de début, de fin , les porteurs de projet associés , le coach associé au projet ainsi que l'étape actuelle du projet.

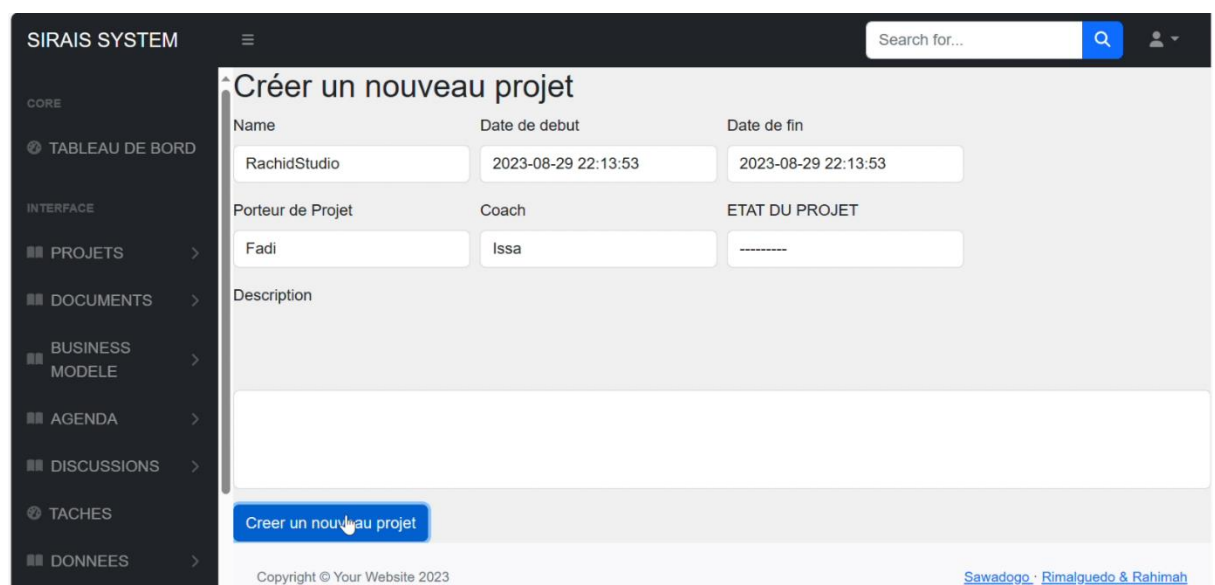


Figure 20 : Ecran pour créer un nouveau projet

La figure 21 montre le tableau de bord d'un projet, cette interface correspond à l'écran ou l'on aperçoit toutes les informations associées à un projet ainsi que les différentes phases du projet et leur niveau d'avancement.

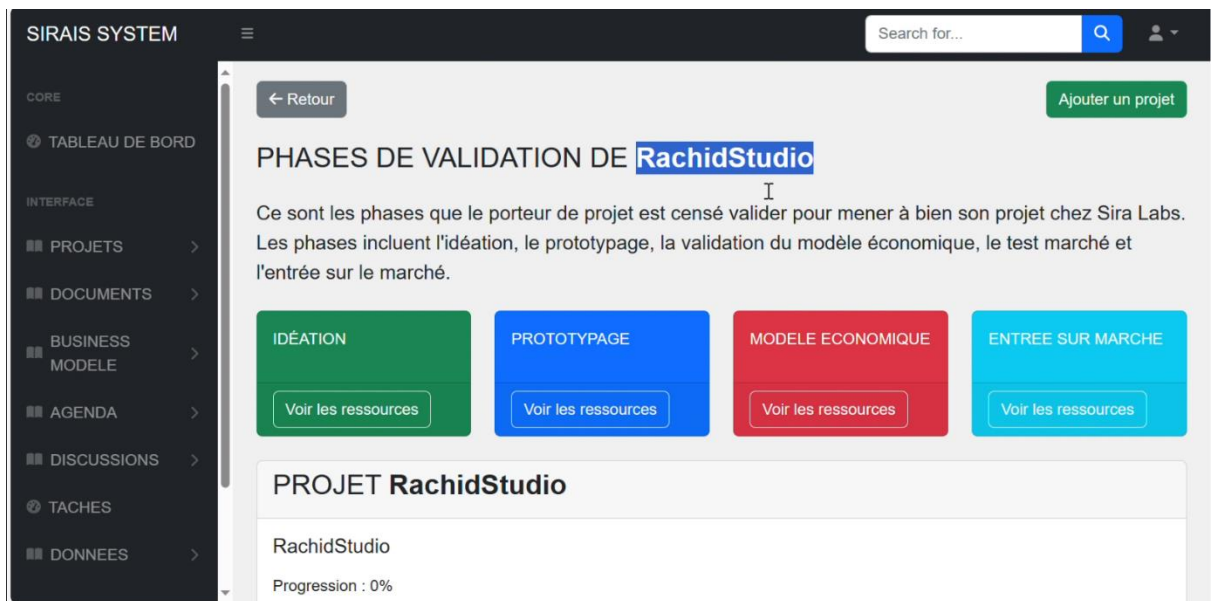


Figure 21 : Tableau de bord du projet

La figure 22 correspond à l'interface où on peut apercevoir les différentes ressources du projet ainsi que les commenter et les valider.

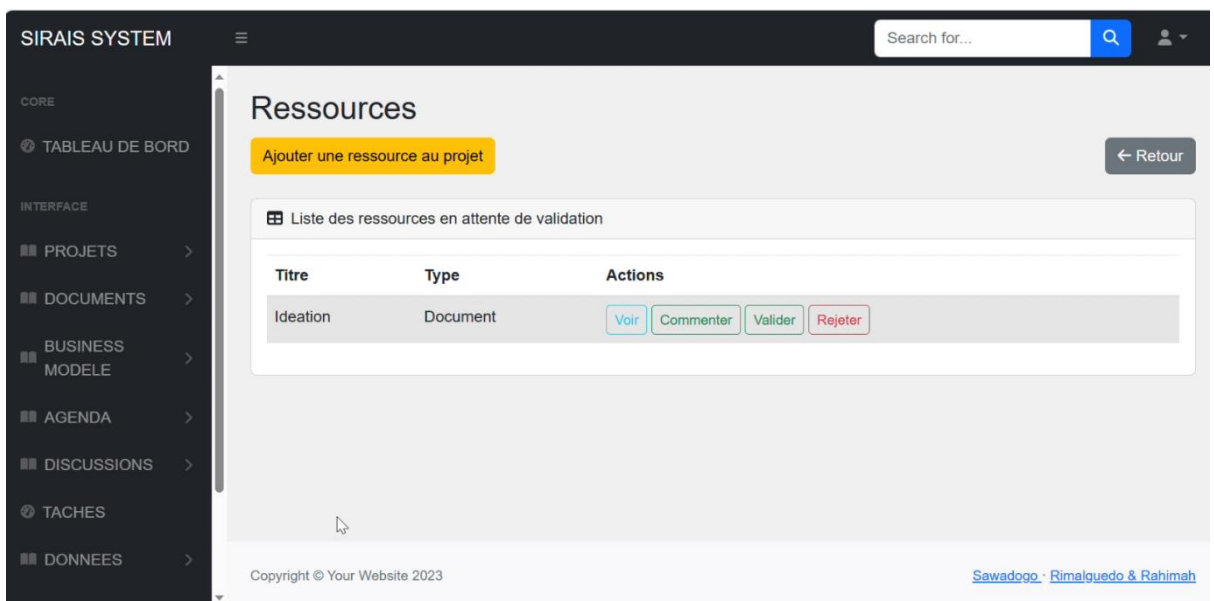


Figure 22 : Liste des ressources

CHAPITRE III : BILAN DU STAGE

Après avoir présenté les activités à mener dans le cadre de la mise en place d'un système de gestion de projets incubées à Sira Labs, nous faisons dans ce dernier chapitre le bilan de notre stage. Pour ce faire, nous allons d'abord, présenter le déroulement du stage ainsi que les activités menées au cours de son déroulement. Ensuite, nous apporterons nos critiques et nos suggestions.

I. DÉROULEMENT DU STAGE ET ACTIVITÉS MÉNÉES

Dans cette partie, nous décrirons comment s'est déroulé le stage dans les locaux de Sira Labs et les activités menées au cours de celui-ci.

1. Activités menées

Notre stage s'est déroulé sur la période allant du premier (1) juillet 2023 au trente (30) Septembre 2023 au sein de l'entreprise Sira Labs. Durant ces trois (3) mois, notre activité était essentiellement basée sur l'étude du thème « **Conception et réalisation d'une plateforme de gestion intégrée et de suivi des startups incubées à Sira Labs.** » suivant le planning de réalisation du projet présenté plus haut.

Nous avons effectué entre autres les activités suivantes :

- l'initiation aux outils de télétravail tels que Google Meet
- le travail dans une équipe pluridisciplinaire
- l'accompagnement et l'assistance de jeunes dans leur projet
- la réalisation de design UI avec Figma
- la découverte de jazzmin

2. Connaissances acquises

Ce stage effectué au sein de l'entreprise Sira Labs 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 accompagnement de projet
- renforcer nos compétences en développement d'application web ;
- renforcer nos compétences en intégration des API
- renforcer nos compétences en sécurité informatique

- s'imprégner des réalités de la vie professionnelle

II. OBSERVATIONS ET SUGGESTIONS

Nous adressons nos remerciements à Sira Labs 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 Sira Labs, 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 Sira Labs de recruter plus de développeurs afin de mener à bien leurs projets pour booster son équipe de développement. Nous encourageons Sira Labs à continuer dans son élan et à être encore plus porteur de valeurs et de bienfaits.

CONCLUSION GENERALE

Le stage effectué au sein de Sira Labs nous a permis d'approfondir nos connaissances acquises durant nos 3 années de formation en mettant en place un système de gestion des projets incubés à Sira Labs.

Dans notre processus de réalisation, nous avons opté pour la méthode 2TUP et le langage de modélisation UML à travers notamment les diagrammes de cas d'utilisation, de séquence, d'activité, de déploiement et de classes avant d'implémenter les différentes fonctionnalités identifiées.

Ce travail fut l'occasion pour nous d'approfondir nos connaissances en matière de développement d'applications avec des technologies telles que Python, Django et de mettre en pratique nos connaissances théoriques et méthodologiques acquises au cours de notre formation à l'IBAM. Ce stage nous a permis non seulement de nous familiariser avec le monde professionnel, mais aussi de renforcer notre capacité à travailler en équipe.

En perspective, nous prévoyons intégrer un service de messagerie instantanée dans notre application pour permettre aux coachs et aux porteurs de projet de s'envoyer des messages via l'application. **Dire qu'on a développé un logiciel fonctionnel**

BIBLIOGRAPHIE ET WEBOGRAPHIE

1. Bibliographie

- ❖ [3] **Abdoul Fayçal NANA (2022)** « CONCEPTION ET REALISATION D'UNE PLATEFORME INFORMATIQUE INTEGREE DE GESTION DES PROGRAMMES D'ACTIVITES : CAS DE ELITE AFRIQUE » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 66p.

2. Webographie

- ❖ [1] « Titre : 8-La méthode 2TUP [6]. | Download Scientific Diagram ». Consulté le 29 septembre 2023.
https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F303497462%2Ffigure%2Ffig6%2FAS%3A556507492487174%401509693133373%2FLa-methode-2TUP-6.png&tbid=aEHDJC0CY8HkpM&vet=1&imgrefurl=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FLa-methode-2TUP-6_fig6_303497462&docid=mxGcCYW32wdYqM&w=824&h=575&source=sh%2F%2Fm%2Fm4%2F1&shem=canimage.
- ❖ [2] « Présentation du langage de modélisation unifié (UML) ». *Cybermédiane* (blog), 4 mars 2022. <https://www.cybermedian.com/fr/unified-modeling-language-uml-introduction/>.
- ❖ [4] Ambrosy, Mathieu. « Architecture client-serveur ». Consulté le 1 octobre 2023. <https://www.geonov.fr/architecture-client-serveur/>.
- ❖ [5] Les SGBD. « Comparaison générale des SGBD », 3 octobre 2014. <https://sgbdslam5.wordpress.com/comparaison-generale-des-sgbd/>
- ❖ appvizer.fr. « Les 11 meilleurs logiciels de gestion de projets gratuits 2023 ». Consulté le 26 septembre 2023. <https://www.appvizer.fr/magazine/operations/gestion-de-projet/meilleurs-logiciels-gestion-de-projet-gratuits-2021>.
- ❖ [6] « Différences entre COCOMO, COSYSMO, REVIC, les points de fonction et WMFP ». Consulté le 26 septembre 2023. http://www.projectcodemeter.com/cost_estimation/help/GL_cocomodiff.htm.

- ❖ « outil open-source qui utilise des descriptions textuelles simples pour dessiner des diagrammes UML. » Consulté le 26 septembre 2023. <https://plantuml.com/fr/>.
- ❖ [7]« Présentation de l'API Google Calendar | Google for Developers ». Consulté le 26 septembre 2023. <https://developers.google.com/calendar/api/guides/overview?hl=fr>.
- ❖ « Qu'est-ce que le langage UML | Lucidchart ». Consulté le 26 septembre 2023. <https://www.lucidchart.com/pages/fr/langage-uml>.
- ❖ « Stack Overflow - où les développeurs apprennent, partagent et construisent des carrières ». Consulté le 26 septembre 2023. <https://stackoverflow.com/>.
- ❖ « Système de gestion de base de données ». In *Wikipédia*, 1 janvier 2023. https://fr.wikipedia.org/w/index.php?title=Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es&oldid=200045650.

TABLES DES MATIERES

<i>SOMMAIRE.....</i>	<i>ii</i>
<i>DEDICACE</i>	<i>iii</i>
<i>REMERCIEMENTS.....</i>	<i>iv</i>
<i>LISTES DES SIGLES ET ABBREVIATIONS</i>	<i>v</i>
<i>LISTE DES FIGURES GRAPHIQUES</i>	<i>vi</i>
<i>LISTE DES TABLEAUX.....</i>	<i>vii</i>
<i>INTRODUCTION GENERALE.....</i>	<i>1</i>
<i>CHAPITRE I : PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL.....</i>	<i>2</i>
I. PRESENTATION DE LA STRUCTURE DE FORMATION.....	2
1. Objectifs de l'IBAM	2
2. Organisation.....	2
3. Filières de formations.....	2
II. PRESENTATION DE LA STRUCTURE D'ACCUEIL	4
1. Histoire.....	4
2. Objectifs.....	4
3. Services	5
<i>CHAPITRE II : ANALYSE ET CONCEPTION</i>	<i>6</i>
I. ETUDE PREALABLE	6
1. Présentation du thème	6
a. Problématique.....	6
b. Objectifs.....	6
c. Résultats attendus	7
2. Méthodologie	7
a. Processus de développement	7
b. Langage de modélisation	10
3. Présentation du groupe de travail	12
a. Groupe de pilotage.....	13
b. Groupe des utilisateurs	13
c. Groupe de projet	13
d. Planning de travail	13
II. EXPRESSIONS DES BESOINS	14

1.	Étude de l'existant.....	14
a.	Présentation du système utilisé à Sira Labs	14
b.	Présentation d'outils logiciels de gestion de projet existants.....	15
2.	Présentation du processus de fonctionnement amélioré.....	16
3.	Exigences fonctionnelles.....	17
4.	Spécification fonctionnelle.....	17
a.	Identification des acteurs	17
b.	Identification des cas d'utilisations.....	18
c.	Diagramme de cas d'utilisation	19
d.	Description textuelle de quelques cas d'utilisation.....	20
e.	Diagramme de séquence	24
5.	Spécifications techniques	28
a.	Mise à disposition des conditions de travail	28
b.	Architecture de développement	28
III.	CONCEPTION GLOBALE.....	30
1.	Diagramme des classes	30
a.	Dictionnaire des données.....	30
b.	Diagramme des classes	32
2.	Diagramme d'activité.....	33
3.	Diagramme de déploiement	37
IV.	REALISATION	38
1.	Présentation des outils de réalisation	38
a.	Choix du Système de Gestion de Base de Données.....	38
b.	Langages de programmation	39
c.	Serveur d'application.....	40
d.	Framework utilisé.....	41
e.	Autres outils.....	41
2.	Politique de sécurité.....	42
a.	Mesure de sécurité prise pendant le développement.....	42
b.	Mesures de sécurité à prendre pour l'exploitation.....	42
c.	Autres mesures de sécurité	43
3.	Estimation du coût de développement	43
4.	Présentation de quelques écrans de l'application.....	44
a.	Présentation de l'interface administrateur	45
b.	Présentation de l'interface client	46
	CHAPITRE III : BILAN DU STAGE.....	49
I.	DÉROULEMENT DU STAGE ET ACTIVITÉS MÉNÉES	49
1.	Activités menées	49

2. Connaissances acquises.....	49
II. OBSERVATIONS ET SUGGESTIONS.....	50
<i>CONCLUSION GENERALE.....</i>	<i>51</i>
<i>BIBLIOGRAPHIE ET WEBOGRAPHIE</i>	<i>52</i>
1. Bibliographie.....	52
2. Webographie	52
<i>TABLES DES MATIERES.....</i>	<i>54</i>
<i>ANNEXES</i>	<i>57</i>
ANNEXE 1 : LANGAGE UML.....	57
ANNEXE 2 : LA METHODE COCOMO.....	59

ANNEXES

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.

ANNEXE 2 : LA METHODE COCOMO

Un grand nombre de méthodes est mis à la disposition des développeurs pour estimer le coût de leurs plateformes. Nous utiliserons la méthode Constructive Cost Model (COCOMO) pour l'estimation du cout 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 on ne peut envisager toutes les possibilités de saisies qu'un utilisateur pourrait effectuer. Elles ne sont pas déterministes.

La formule de calcul COCOMO est présentée dans le Tableau 9.

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	$\text{Effort} = 3,6 * \text{KLS}^{1,2}$	$\text{TDev} = 2,5 * \text{Effort}^{0,32}$

Tableau 13 : Formule de calcul COCOMO

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