



REPUBLIQUE TOGOLAISE

Travail-Liberté-Patrie

MINISTÈRE DE LA PLANIFICATION DU
DÉVELOPPEMENT ET DE LA COOPÉRATION



Institut Africain d'Informatique
Représentation du TOGO
(IAI-TOGO)

Tél : 22 20 47 00

Email : iaitogo@iai-togo.tg

Site Web : www.iai-togo.tg

07 BP 12456 Lomé 07, TOGO



Odda Technology

Tél : 90 39 76 86 / 92 28 61 27

Email : oddatechnology@outlook.com

Route CEET Agoè Cacaveli

Face CEG Cacavéli côté sud

PROJET DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME
DE LICENCE PROFESSIONNELLE EN INFORMATIQUE

OPTION : GENIE LOGICIEL & SYSTEMES D'INFORMATIONS
THEME :

Mise en place d'une plateforme web pour l'élaboration et
l'exécution des budgets des écoles publiques de
l'enseignement primaire et secondaire

Période : Du 22 Mai au 19 Août 2023

Rédigé et soutenu par :

KPEKPASSI Ibtihadj

Etudiant en troisième année

Année académique : 2022 – 2023

SUPERVISEUR

M. TCHANTCHO Leri D

Enseignant à IAI-TOGO

MAITRE DE STAGE

M. BILANTE Addo

DSI Odda Technology

DEDICACES

À mes chers parents

Le parcours scolaire et universitaire que j'ai traversé a été guidé par vos sacrifices incommensurables et votre confiance indéfectible en moi. Votre soutien moral et spirituel m'a accompagné tout au long de cette aventure, et vos encouragements constants ont été ma source d'inspiration pendant ces années d'études.

En Toute Reconnaissance

J'adresse également mes remerciements sincères à tous ceux et celles qui ont contribué, de près ou de loin, à la réalisation de ce projet. Votre présence engagée, vos conseils éclairés et vos encouragements quotidiens ont été les fondements de cette réussite.

Avec toute ma gratitude,

Ibtihadj

REMERCIEMENTS

Je commence par exprimer ma gratitude envers Dieu Tout-Puissant pour sa miséricorde continue dans nos vies. Mes sincères remerciements s'adressent également à toutes les personnes qui ont rendu possible la création de ce document et la réalisation de ce projet. Mes remerciements vont tout particulièrement à :

- ❖ **M. BILANTE Addo**, Directeur Système d'Information chez Odda Technology, pour son accueil chaleureux au sein de son entreprise, le partage de son expérience ainsi que la mise à disposition des outils essentiels à la réalisation de ce projet ;
- ❖ **M. BILANTE Tchapo**, Directeur Général d'Odda Technology, pour son soutien continu et ses apports précieux tout au long de ce projet ;
- ❖ **M. AGBETI Kodjo**, Représentant Résident de l'IAI-TOGO, pour avoir offert un cadre d'études propice et pour ses efforts déployés en vue de me garantir des conditions d'études optimales ;
- ❖ **M. AMEYIKPO Nicolas**, Directeur des Affaires Académiques et de la Scolarité de l'IAI TOGO, pour ses multiples contributions en faveur de l'institut ;
- ❖ **M. TCHANTCHO Léri D.**, mon superviseur, dont les conseils avisés ont grandement contribué à la réalisation de ce document et du projet dans son ensemble ;
- ❖ **Le Corps professoral et administratif de l'IAI-TOGO**, pour les enseignements riches et variés qu'ils m'ont transmis ;
- ❖ **Mes frères et sœurs**, qui ont toujours été là pour me soutenir et m'encourager durant toute la période de réalisation du projet.
- ❖ **A tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.**

Leurs efforts et leur soutien ont été essentiels pour mener à bien ce projet, et je leur en suis sincèrement reconnaissant.

AVANT-PROPOS

L'Institut Africain d'Informatique (IAI) est une institution régionale fondée le 29 janvier 1971 à Fort-Lamy, actuelle N'Djamena, capitale de la République du Tchad. Cette initiative a été mise en œuvre par onze pays, à savoir le Bénin, le Burkina Faso, le Cameroun, la Côte d'Ivoire, la Centrafrique, la République du Congo, le Gabon, le Niger, le Sénégal, le Tchad et le Togo. Son siège est situé à Libreville, au Gabon. L'IAI se consacre à la formation d'ingénieurs de travaux informatiques, dispensant ses enseignements à Libreville et dans ses antennes au Cameroun, au Niger et au Togo.

Depuis sa création en 2002, l'IAI-TOGO se consacre à la formation d'ingénieurs des travaux informatiques sur une période de trois ans. Trois filières y sont proposées : Génie Logiciel et Système d'Informations (GLSI), Administration Système & Réseaux (ASR) et Multimédia Web et Infographie (M-TWI). À la fin de cette formation triennale, un stage en entreprise d'une durée de trois mois est intégré au cursus, contribuant ainsi à l'obtention du Diplôme d'Ingénieur des Travaux Informatiques.

L'objectif de ce stage est de permettre aux étudiants d'appliquer les compétences acquises tout au long de leur formation, de façon à devenir des ingénieurs compétents et efficaces dans le monde professionnel. Pour notre part, nous avons effectué notre stage au sein des locaux de la société Odda Technology, en vue de mener à bien notre projet de fin de formation. Ce dernier se concentre sur la « **Mise en place d'une plateforme web pour l'élaboration et l'exécution des budgets des écoles publiques de l'enseignement primaire et secondaire** ».

RESUME

Dans le cadre de l'obtention du diplôme d'Ingénieur des Travaux Informatiques à l'IAI-TOGO, nous avons entrepris un stage au sein de l'entreprise Odda Technology. Notre projet de stage, intitulé « **Mise en place d'une plateforme web pour l'élaboration et l'exécution des budgets des écoles publiques de l'enseignement primaire et secondaire** », a été soumis à une étude approfondie.

L'objectif principal de ce projet était de développer une application web innovante, destinée aux acteurs financiers du secteur de l'enseignement primaire et secondaire des écoles publiques. Notre solution vise à simplifier l'élaboration et l'exécution des budgets alloués à ces établissements. Elle offre une plateforme conviviale permettant un suivi en temps réel des dépenses, tant au niveau des écoles qu'au niveau des instances telles que les Inspections, les Directions Régionales et la Direction Administrative et Financière.

Pour répondre à ces besoins complexes, nous avons adopté une approche méthodologique rigoureuse. La phase de modélisation a été abordée avec le langage de modélisation UML, couplé au processus 2TUP, afin de maîtriser la complexité du système à concevoir. En tant que fondement technique, nous avons utilisé le langage de programmation Java, associé au framework Spring Boot, au framework Angular pour l'interface utilisateur dynamique, ainsi qu'au système de gestion de bases de données MySQL.

ABSTRACT

As part of the requirements for obtaining a degree in Computer Engineering from IAI-TOGO, we embarked on an internship within the company Odda Technology. Our internship project, titled "**Implementation of a web platform for the development and execution of budgets for public primary and secondary schools**" underwent a comprehensive study.

The primary objective of this project was to develop an innovative web application aimed at the financial stakeholders in the primary and secondary education sector of public schools. Our solution aims to streamline the process of creating and executing budgets allocated to these institutions. It provides a user-friendly platform that enables real-time expense tracking, both at the school level and within entities such as Inspections, Regional Directorates, and the Administrative and Financial Directorate.

To address these complex requirements, we adopted a rigorous methodological approach. The modeling phase was executed using the UML modeling language, coupled with the 2TUP process, to manage the complexity of the system design. As the technical foundation, we utilized the Java programming language, along with the Spring Boot framework, the Angular framework for dynamic user interfaces, and the MySQL database management system.

SOMMAIRE

DEDICACES	i
REMERCIEMENTS	ii
AVANT-PROPOS	iii
RESUME	iv
ABSTRACT	v
SOMMAIRE	vi
GLOSSAIRE	viii
LISTE DE FIGURES	ix
LISTE DES TABLEAUX	xii
LISTE DES PARTICIPANTS AU PROJET	xiii
INTRODUCTION	1
PARTIE 1 : CAHIER DE CHARGES	2
1.1 PRESENTATIONS	3
1.2 THEME DU STAGE	8
1.3 ETUDE DE L'EXISTANT	11
1.4 CRITIQUE DE L'EXISTANT	11
1.5 PROPOSITIONS ET CHOIX DE SOLUTION	12
1.6 PLANNING PREVISIONNEL DE REALISATION	20
PARTIE 2 : ANALYSE ET CONCEPTION	23
2.1 CHOIX ET JUSTIFICATION DE LA METHODE D'ANALYSE	24
2.2 CHOIX ET JUSTIFICATION DE L'OUTIL DE MODELISATION	31
2.3 ETUDE DETAILLEE DE LA SOLUTION	32
PARTIE 3 : REALISATION ET MISE EN ŒUVRE	69
3.1 MATERIEL ET LOGICIELS UTILISES	70
3.2 ARCHITECTURES MATERIELLE ET LOGICIELLE DE L'APPLICATION	78

3.3 SECURITE DE L'APPLICATION.....	81
3.4 MISE EN PLACE DE LA BASE DE DONNEES.....	81
PARTIE 4 : GUIDE D'EXPLOITATION	87
4.1 CONFIGURATION MATERIELLE ET LOGICIEL	88
4.2 DEPLOIEMENT ET SUIVI.....	89
4.3 MAINTENANCE : ACTION A MENER EN CAS DE CERTAINES ERREURS.....	102
PARTIE 5 : GUIDE D'UTILISATION	104
5.1 DESCRIPTION TEXTUELLE DU LOGICIEL	105
5.2 PLAN DE NAVIGATION	106
5.3 PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION	108
5.4 PRESENTATION DES ETATS.....	111
CONCLUSION	113
BIBLIOGRAPHIE INDICATIVE.....	114
WEBOGRAPHIE INDICATIVE	115
TABLE DES MATIERES	116

GLOSSAIRE

Tableau 1: Glossaire

ABBREVIATION ET ACRONYMES	DEFINITION
IAI	Institut Africain d'Informatique
GL	Génie Logiciel
SR	Systèmes et Réseaux
MTWI	Multimédia Technologie Web et Infographie
UML	Unified Modeling Language
2TUP	2 Track Unified Process
API	Application Programming Interface
SGBD	Système de gestion de base de données
IDE	Integrated Development Environment
MVC	Modèle Vue Contrôleur
JRE	Java Runtime Environment
JVM	Java Virtual Machine
DAF	Directeur administratif et financier
CU	Cas d'utilisation

LISTE DE FIGURES

Figure 1 : Localisation de l'IAI-TOGO	4
Figure 2 : Organigramme d'Odda Technology.....	6
Figure 3: Plan de localisation d'Odda Technology	7
Figure 4 : Logo de MyScol	13
Figure 5 : Diagramme de GANTT	22
Figure 6 : Logo de UML.....	26
Figure 7 : Illustration du processus 2TUP	30
Figure 8: Organigramme du futur système.....	34
Figure 9 : Diagramme de contexte statique	35
Figure 10 : Cas d'utilisation pour l'acteur "Administrateur"	39
Figure 11 : Cas d'utilisation pour l'acteur "Directeur administratif et financier" ..	40
Figure 12 : Cas d'utilisation pour l'acteur "Directeur régional"	41
Figure 13 : Cas d'utilisation pour l'acteur "Inspecteur".....	42
Figure 14 : Cas d'utilisation pour l'acteur "Directeur d'école".....	43
Figure 15 : Cas d'utilisation pour l'acteur "Comptable d'école"	44
Figure 16 : Diagramme d'activité du cas d'utilisation « Ajouter un utilisateur »	56
Figure 17 : Diagramme d'activité du cas d'utilisation « S'authentifier ».....	57
Figure 18 : Diagramme d'activité du cas d'utilisation « Initier la planification budgétaire ».....	58
Figure 19 : Diagramme d'activité du cas d'utilisation « Créer une révision budgétaire »	59
Figure 20 : Diagramme d'activité du cas d'utilisation « Créer une dépense»	60
Figure 21 : Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »	61
Figure 22 : Diagramme de séquence du cas d'utilisation « S'authentifier»	62
Figure 23 : Diagramme de séquence du cas d'utilisation « Initier la planification budgétaire».....	63
Figure 24 : Diagramme de séquence du cas d'utilisation « Créer une révision budgétaire ».....	64
Figure 25 : Diagramme de séquence du cas d'utilisation « Créer une dépense »....	65
Figure 26 : Diagramme de classe.....	67

Figure 27: Diagramme de déploiement	68
Figure 28 : Logo de PhpMyAdmin	71
Figure 29 : Logo de Spring boot.....	71
Figure 30 : Logo d'Angular	72
Figure 31 : Logo de Bootstrap	73
Figure 32 : Logo de PrimeNg	74
Figure 33 : Logo de Git.....	75
Figure 34 : Logo de Github	76
Figure 35 : Logo de MySQL	76
Figure 36 : Logo de IntelliJ IDEA	77
Figure 37 : Architecteure physique du projet.....	79
Figure 38 : Architecture MVC.....	80
Figure 39 : Structure de la table « rubriques_depenses »	82
Figure 40 : Structure de la table « inspections»	83
Figure 41 : Structure de la table « revisions_budgetaires »	84
Figure 42 : Structure de la table « depenses »	85
Figure 43 : Structure de la table « sources_de_financements »	86
Figure 44 : Commande de génération d'un fichier jar à partir d'une API Spring boot	90
Figure 45 : Génération du fichier .jar effectuée	90
Figure 46 : Dossier target contenant le fichier jar généré	91
Figure 47 : Fichier Dockerfile.....	91
Figure 48 : Repository Github de l'API du projet.....	93
Figure 49 : Interface de Render.....	93
Figure 50 : Accord des autorisation à Render sur notre compte Github.....	94
Figure 51 : Création d'un web service sur github	94
Figure 52 : Déploiement de la partie backend effectué sur Render	95
Figure 53: Installation de la CLI Firebase	96
Figure 54 : Connexion à Google	96
Figure 55 : Accord des autorisations à Firebase CLI	97
Figure 56 : Connexion à Firebase CLI réussie	97
Figure 57 : Création d'un projet sur Firebase	98
Figure 58 : Projet Firebase créé	98

Figure 59 : Liaison entre notre application front end et le projet créé sur Firebase	99
Figure 60 : Paramétrages des préférences	99
Figure 61 : Liaison effectuée avec succès	100
Figure 62 : Déploiement effectif sur Firebase	100
Figure 63 : Exemple illustrant la page d'erreur 404	103
Figure 64 : Page de connexion	105
Figure 65 : Page de demande de réinitialisation de mot de passe	106
Figure 66 : Plan de navigation de l'acteur : Administrateur	106
Figure 67 : Plan de navigation de l'acteur : Directeur administratif et financier	107
Figure 68 : Plan de navigation de l'acteur : Directeur régional	107
Figure 69 : Plan de navigation de l'acteur : Inspecteur	107
Figure 70 : Plan de navigation des acteurs : Directeur d'école et Comptable d'école	108
Figure 71 : Interface de gestion des sources de financements	109
Figure 72 : Interface de gestion des rubriques	109
Figure 73 : Interface de suivi des dépenses d'une école	110
Figure 74 : Interface de création d'une nouvelle révision budgétaire	110
Figure 75 : Tableau de bord de l'administrateur	111
Figure 76 : Etat d'execution de budget par sous rubrique : cas d'une ecole	112

LISTE DES TABLEAUX

Tableau 1: Glossaire	viii
Tableau 2: Liste des participants au projet	xiii
Tableau 3 : Coût du logiciel MyScol	16
Tableau 4: Coût de formation	16
Tableau 5 : Coût total	16
Tableau 6 : Coût matériel	17
Tableau 8: Coût de connectivité réseau	17
Tableau 7: Coût d'hébergement	18
Tableau 10: Coût de formation	18
Tableau 11 : Coût total	19
Tableau 12 : Planning prévisionnel de réalisation	20
Tableau 13 : Les cas d'utilisations du projet	36
Tableau 14 : Description textuelle du cas d'utilisation "Ajouter un utilisateur"	46
Tableau 15 : Description textuelle du cas d'utilisation « S'authentifier»	48
Tableau 16 : Description textuelle du cas d'utilisation « Initier la planification budgétaire»	50
Tableau 17 : Description textuelle du cas « Créer une révision budgétaire»	52
Tableau 18 : Description textuelle du cas « Enregistrer une dépense»	54
Tableau 19 : Matériel utilisé	70
Tableau 20 : Webographie indicative	115

LISTE DES PARTICIPANTS AU PROJET

Tableau 2: Liste des participants au projet

INTERVENANTS	FONCTIONS	ROLE
M. KPEKPASSI Ibtihadj	Etudiant en 3 ^{ème} année, option Génie Logiciel & Systèmes d'information à IAI-TOGO	Réalisateur
M. BILANTE Addo	Directeur Système d'Information d'Odda Technology	Maître de stage
M. TCHANTCHO Leri D.	Enseignant à IAI-TOGO	Superviseur

INTRODUCTION

L'éducation revêt un rôle fondamental dans le développement socio-économique d'une nation. L'efficacité et la transparence dans l'allocation des ressources aux écoles primaires et secondaires sont impératives pour garantir un enseignement de qualité et l'épanouissement des apprenants. Dans ce contexte, l'évolution des technologies de l'information offre une opportunité exceptionnelle de transformer et d'améliorer la manière dont les budgets sont élaborés et gérés dans ce secteur vital.

Ce mémoire se focalise sur un élément crucial de la gestion budgétaire des écoles publiques du primaire et du secondaire. Notre projet, intitulé « **Mise en place d'une plateforme web pour l'élaboration et l'exécution des budgets des écoles publiques de l'enseignement primaire et secondaire** », découle de la nécessité d'instaurer une uniformité, de rationaliser et d'optimiser la gestion des ressources financières au sein de ces institutions, en capitalisant sur les avancées technologiques actuelles.

Notre étude met l'accent sur les enjeux de planification, d'allocation et de suivi budgétaire dans les écoles publiques. Avec la complexité grandissante des systèmes éducatifs, répondre à ces défis exige une approche novatrice, visant une gestion budgétaire transparente et performante. Dans cette perspective, notre projet se place ainsi au cœur de la transformation en cours, visant à moderniser et optimiser la gestion budgétaire dans le domaine de l'éducation.

À travers ce mémoire, nous scruterons en détail les cinq (5) phases majeures du projet : le cahier des charges, l'analyse et la conception, la réalisation et la mise en œuvre, ainsi que la création d'un guide d'exploitation et d'utilisation de l'application.



PARTIE 1 : CAHIER DE CHARGES



Le cahier des charges est un document contractuel, matérialisant l'accord entre les différents acteurs du projet. De plus, il sert de document détaillé contenant toutes les informations nécessaires à la réalisation du projet. Tous les acteurs se réfèrent à ce document pour trouver leurs responsabilités respectives dans le projet. Validé par le maître de stage et les futurs utilisateurs, il sert de document de référence lors de l'évaluation du produit final. Les grandes lignes de notre cahier des charges portent sur les présentations (de l'IAI-TOGO et du centre d'accueil), le thème du stage, l'étude et la critique de l'existant, les propositions et choix de solution ainsi que le planning prévisionnel de réalisation.

1.1 PRESENTATIONS

1.1.1 Présentation de l'IAI-TOGO

L'Institut Africain d'Informatique (IAI) est un institut Inter-Etats, créé en 1971 à Fort Lamy actuel N'Djamena, au Tchad. Son siège se trouve à Libreville au Gabon. Il compte onze (11) Etats membres à savoir le Bénin, le Burkina Faso, le Cameroun, la République Centrafricaine, la Côte d'Ivoire, la République du Congo, le Gabon, le Niger, le Sénégal, le Tchad et le Togo. Une représentation nationale pour le cycle d'Analyste Programmeur, a été ouverte dans certains pays membres, notamment le Cameroun, le Niger et le Togo. Ces représentations ont été mises en place dans le but de mettre à la disposition des états membres un centre de formation adéquat et conforme aux normes internationales. L'IAI forme des ingénieurs informaticiens à Libreville et des ingénieurs des travaux informatiques dans ses représentations. L'Institut Africain d'Informatique, représentation du Togo (IAI-TOGO), forme en trois ans des ingénieurs des travaux informatiques depuis 2002. Il a élargi son programme de formation en y intégrant les Systèmes et Réseaux Informatiques et la filière MultiMedia Technologie Web et Infographie. Dans le but de permettre à l'étudiant de pouvoir s'intégrer facilement dans le milieu professionnel, l'IAI dans son programme de fin de 2ème et de 3ème année, a prévu un stage pratique en entreprise, à la fin de l'année académique. Ce stage est sanctionné par la production d'un mémoire qui représente un aperçu du travail effectué durant cette période et a pour finalité l'obtention du diplôme d'Ingénieur des Travaux Informatiques. L'IAI-Togo se trouve à Lomé dans le quartier administratif sur la rue de la Kozah derrière l'immeuble de SUNU Assurances dans les locaux du Centre National

d'Etudes et de Traitements Informatiques (CENETI) non loin de la CEB (Communauté Electrique du Bénin) comme l'indique le plan de localisation ci-dessous.

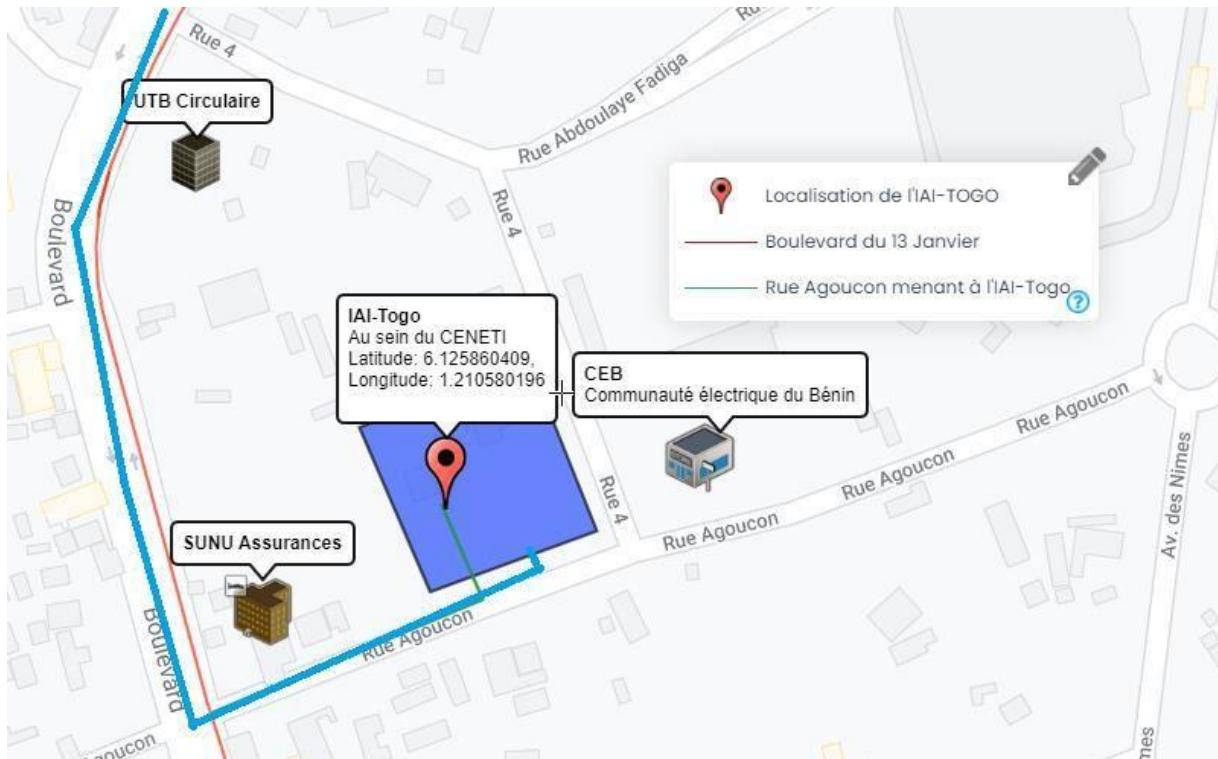


Figure 1 : Localisation de l'IAI-TOGO

L'IAI-TOGO peut être contacté aux adresses suivantes :

Tel: (+228) 22 20 47 00

E-mail: iaitogo@iai-togo.tg

1.1.2 Présentation du cadre de stage

La société informatique qui nous offre son cadre pour notre stage est **Odda Technology**. Nous verrons comment elle est organisée du point de vue de sa structure externe à travers son statut, sa mission, ses activités, ses réalisations, et dans sa structure interne à travers l'organigramme et le service d'accueil.

❖ Statut

ODDA Technology est une Société Informatique SARL (Société Anonyme à Responsabilité Limité). Elle est considérée comme une Société de Services en Ingénierie Informatique (SSII). C'est un carrefour d'esprits créatifs où les problèmes locaux sont objet d'analyse, et leur résolution de la manière la plus innovante devient un défi. Elle dispose aussi des cellules de maintenance et d'audit afin d'accompagner les sociétés de par leur expertise et leur expérience en la matière. Elle est aussi une référence en matière de vente de matériels informatiques de qualité. Tout pour une cohabitation agréable entre l'informatique et la population. **Odda Technology** s'est spécialisée en outre dans le développement web (Site web et Application), mobile et en système réseau pour répondre au besoin fonctionnel des systèmes d'information.

❖ Mission

Odda Technology s'est donné pour mission d'être la référence dans les domaines suivants : Conseil, audit, développement des systèmes et des solutions dans le domaine informatique et des réseaux.

❖ Activités

Depuis sa création, elle s'est positionnée comme partenaire de choix dans les solutions informatiques. Afin d'apporter des solutions complètes et de haute qualité aux multiples besoins des entreprises soucieuses d'accroître leur productivité et leur compétitivité, elle a bâti son activité autour de l'informatique, le réseau et la vente de matériels Informatiques.

❖ Quelques réalisations

- ✓ Une application dédiée à la gestion des écoles publiques et privées, et est utilisé dans plus de 100 écoles sur toute l'étendue du territoire : **SCHOOL MANAGER WEB 2.0**

- ✓ Application pour la gestion des écoles supérieures : **UNIVERSITY MANAGER**
 - ✓ Un système de calcul actuarial des employés (Calcul des indemnités de licenciement et des indemnités de retraite) : **staffACTUARY**

❖ Organigramme

La structure interne de la structure **Odda Technology** se présente comme suit :

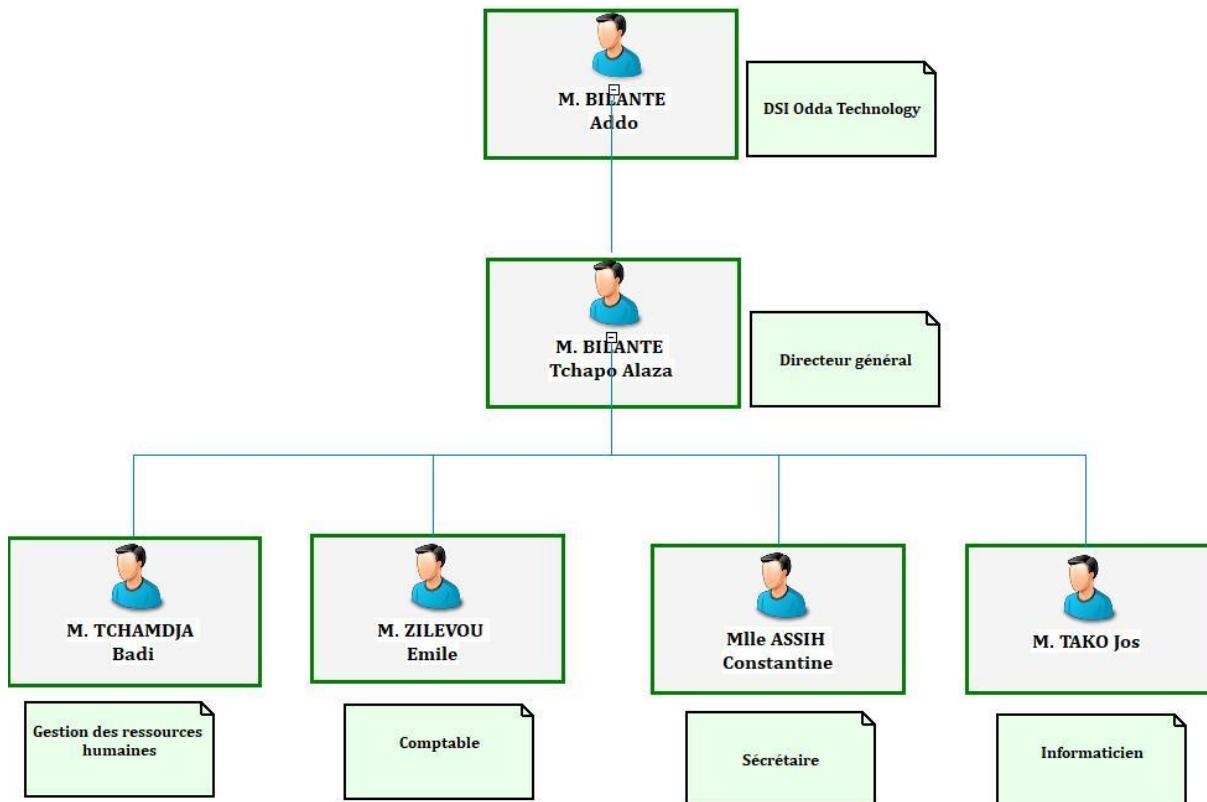
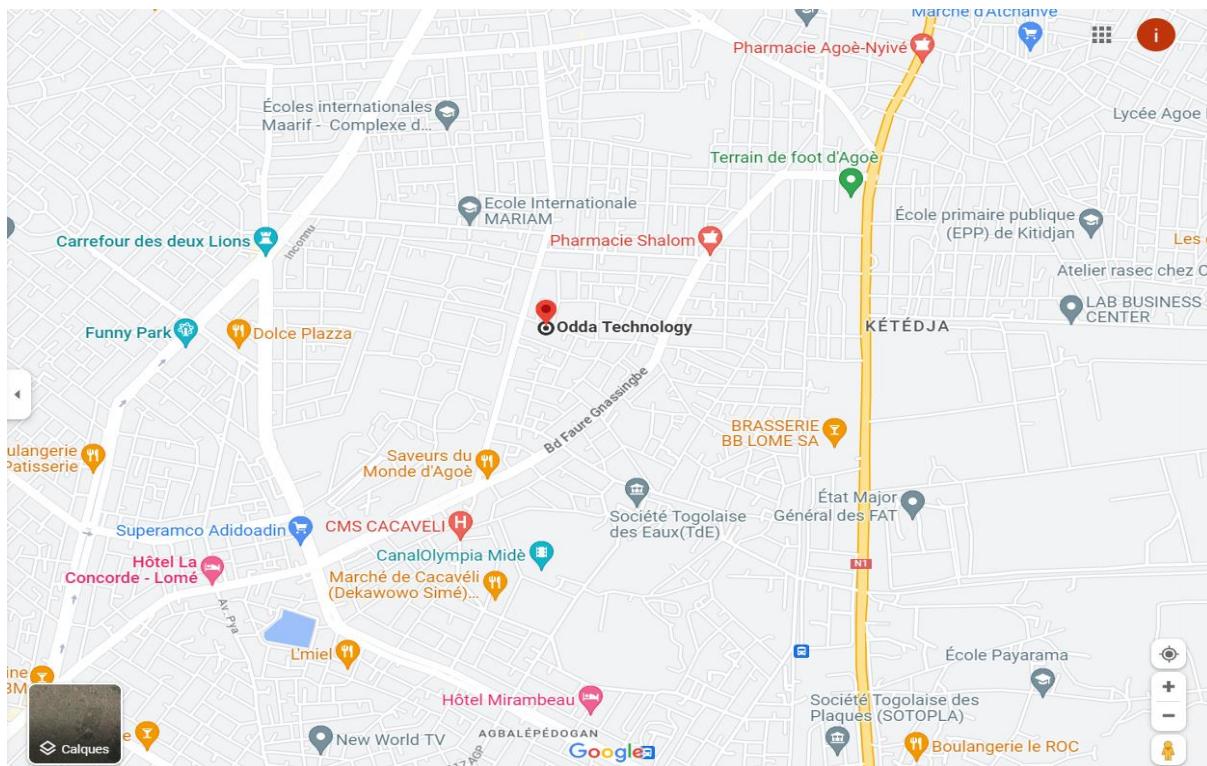


Figure 2 : Organigramme d'Odda Technology

❖ Service d'accueil

Nous avons effectué notre stage au sein du service des Ressources Humaines qui est chargé du recrutement et de la rémunération des agents

❖ Plan de localisation



1

Figure 3: Plan de localisation d'Odda Technology

**Route CEET Agoè Cacaveli, Face CEG cacaveli Coté Sud,
Lomé**

Tel: (+228) 90 39 76 86

E-mail: oddatechnology@outlook.com

¹ Source : Capture d'écran de Google maps

1.2 THEME DU STAGE

1.2.1 Présentation du sujet

Le thème qui a été soumis à notre étude est : « **Mise en place d'une plateforme web pour l'élaboration et l'exécution des budgets des écoles publiques de l'enseignement primaire et secondaire** ».

Il s'agit de concevoir une plateforme web permettant aux écoles publiques de l'enseignement primaire et secondaire d'élaborer leur budget et de suivre en temps réel le rapport de leurs dépenses par rapport au budget établi.

Cette plateforme vise à centraliser les dépenses des institutions scolaires, ce qui en fera un outil essentiel pour les décideurs directement ou indirectement concernés par l'aspect financier des écoles publiques de l'enseignement primaire et secondaire.

1.2.2 Problématique

La quantité d'informations décisionnelles découlant des dépenses effectuées par les écoles publiques de l'enseignement primaire et secondaire dans le pays est considérable. Cependant, en l'absence d'un système informatique centralisé visant à consolider ces dépenses et à améliorer la gestion des ressources financières dans le domaine, l'accès à ces informations est quasi-impossible. Le défi majeur réside donc dans l'évaluation des dépenses financières engagées chaque année par ces écoles.

Afin de répondre de manière satisfaisante à ce besoin, les questions suivantes méritent être posées :

- Comment collecter et enregistrer de manière centralisée les données des dépenses pour toutes les écoles publiques de l'enseignement primaire et secondaire au Togo ?
- Comment élaborer les budgets scolaires de manière efficace et précise, en prenant en compte les besoins spécifiques de chaque école ?
- Comment permettre aux écoles d'élaborer et exécuter leurs budgets de manière collaborative, en impliquant tous les acteurs concernés, tels que les directeurs d'école et les responsables financiers ?

- Comment assurer le suivi en temps réel des dépenses par rapport aux budgets établis, afin de détecter les écarts éventuels et de prendre des mesures correctives rapidement ?
- Comment générer des rapports clairs et détaillés sur les dépenses et les budgets, permettant une analyse approfondie des données financières ?
- Comment garantir la sécurité des données financières et assurer la confidentialité des informations sensibles liées aux budgets des écoles ?
- Comment rendre la plateforme web conviviale et facile à utiliser pour les écoles, avec une interface intuitive et des fonctionnalités adaptées à leurs besoins spécifiques en matière d'élaboration et de suivi des budgets ?

La mise en place d'une plateforme web dédiée à l'élaboration et à l'exécution des budgets scolaires permettrait de répondre à ces questions et de faciliter la gestion des dépenses et des budgets pour les écoles, offrant ainsi une solution efficace pour suivre et contrôler les dépenses tout en assurant une meilleure transparence et une utilisation plus efficiente des ressources financières.

1.2.3 Intérêt du sujet

Divisé en objectifs et résultats attendus, il est question de montrer dans cette partie de notre mémoire le motif et les attentes de ce projet.

1.2.3.1 Objectifs

Objectif général

L'objectif général de notre projet est de mettre en place une plateforme web complète et intégrée permettant aux écoles publiques de l'enseignement primaire et secondaire au Togo d'élaborer, suivre et exécuter leurs budgets de manière efficace.

Objectif spécifiques

De façon spécifique, ce projet va permettre de :

- ✓ **Centraliser les données financières** : Mettre en place un système centralisé pour collecter et consolider les données financières des écoles publiques de l'enseignement primaire et secondaire, permettant une gestion plus efficace des informations ;

- ✓ **Suivre en temps réel des dépenses** : Permettre aux écoles de suivre en temps réel leurs dépenses par rapport aux budgets établis, facilitant ainsi la détection rapide des écarts et des problèmes potentiels ;
- ✓ **Générer des rapports financiers détaillés** : Fournir des rapports financiers clairs et complets pour aider les écoles à analyser leurs dépenses, évaluer les performances et prendre des décisions éclairées ;
- ✓ **Assurer la sécurité des données** : Mettre en place des mesures de sécurité robustes pour protéger les données financières des écoles et garantir la confidentialité des informations sensibles ;
- ✓ **Fournir un soutien technique et une formation aux écoles** : Assurer un accompagnement technique pour garantir une utilisation optimale de la plateforme, ainsi que des sessions de formation visant à aider les écoles à exploiter pleinement les fonctionnalités et les avantages offerts par la plateforme.

1.2.3.2 Résultats attendus

Les principaux résultats attendus de ce projet sont :

- ✓ **La centralisation des données est effective** ;

La centralisation des données sera réalisée avec succès, permettant ainsi une gestion unifiée de toutes les informations financières de l'entreprise.

- ✓ **Le suivi en temps réel des dépense est possible** ;

Le suivi en temps réel des dépenses sera mis en place, offrant aux différents acteurs du système la possibilité de surveiller et de gérer les dépenses au fur et à mesure qu'elles surviennent.

- ✓ **La génération des rapports financiers est possible à partir de la plateforme** ;

La plateforme permettra la génération automatisée de rapports financiers, offrant ainsi aux utilisateurs une vue claire et précise de la situation financière en fonction de l'institution à laquelle ils appartiennent.

- ✓ **La sécurité et la confidentialité des données sont garanties** ;

Des mesures de sécurité rigoureuses seront mises en place pour assurer la protection et la confidentialité des données financières, garantissant que seules les personnes autorisées y ont accès.

- ✓ **Les futurs utilisateurs de la solution sont formés et le support technique est disponible.**

Les futurs utilisateurs de la solution bénéficieront d'une formation complète pour maximiser leur compréhension et leur utilisation efficace de la plateforme.

Un support technique sera accessible pour aider les utilisateurs en cas de problèmes techniques ou de besoins d'assistance, garantissant ainsi un fonctionnement fluide de la solution.

1.3 ETUDE DE L'EXISTANT

Avant de commencer notre travail, il est essentiel d'avoir une compréhension claire et précise de la situation actuelle. Ainsi, nous avons entrepris une analyse approfondie des différentes méthodes utilisées par les écoles pour gérer leurs budgets annuels.

Cette analyse a révélé l'existence de deux cas distincts :

1^{er} Cas : Gestion manuelle

Dans ce cas, les acteurs financiers des écoles remplissent des formulaires physiques pré-imprimés comportant différentes rubriques. Lors des révisions budgétaires, ces formulaires sont modifiés en fonction des montants précédemment enregistrés.

2^{ème} Cas : Gestion assistée par ordinateur

Dans ce cas, les acteurs financiers utilisent principalement le logiciel Excel pour faciliter les calculs et la gestion budgétaire.

Face à cette diversité de pratiques et afin d'harmoniser le processus budgétaire, l'entreprise ODDA Technology a pris l'initiative de développer une plateforme web centralisée. Cette plateforme permettra aux acteurs financiers des écoles de collaborer sur une même base, en utilisant des outils et des rubriques bien définies pour l'élaboration et l'exécution de leurs budgets.

1.4 CRITIQUE DE L'EXISTANT

L'étude approfondie de l'existant a révélé certaines lacunes dans le processus actuel de gestion des budgets des écoles publiques de l'enseignement primaire et secondaire au Togo. Malgré les efforts déployés par les acteurs financiers, il existe des problèmes majeurs qui entravent une gestion budgétaire efficace et transparente.

Tout d'abord, le cas de gestion manuelle pose des défis importants. Les formulaires physiques utilisés pour collecter les données budgétaires sont sujets à des erreurs humaines et à des difficultés de mise à jour. De plus, la consolidation et l'analyse des données provenant de différentes écoles et régions deviennent extrêmement laborieuses, voire impossibles. Il est donc difficile, voire impossible, d'avoir une vue d'ensemble des dépenses engendrées pour une rubrique ou une sous-rubrique à l'échelle régionale ou nationale.

De même, le recours à des logiciels tels qu'Excel présente également des limitations. Bien que ces outils puissent faciliter les calculs, ils ne permettent pas une collaboration en temps réel ni une centralisation des données budgétaires. Cela entraîne une fragmentation des informations, rendant difficile la coordination entre les acteurs financiers et la consolidation des données.

Ces lacunes dans le processus actuel soulignent la nécessité d'une plateforme web dédiée à l'élaboration et à l'exécution des budgets des écoles publiques. Une telle plateforme permettrait une gestion budgétaire plus efficace, transparente et harmonisée. En centralisant les données, elle faciliterait la collaboration entre les acteurs financiers, garantirait l'intégrité des données et offrirait une vue d'ensemble des dépenses engagées pour chaque rubrique ou sous-rubrique, tant au niveau régional que national.

En mettant en place cette plateforme, nous visons à résoudre les problèmes actuels et à améliorer la gestion des budgets des écoles publiques de l'enseignement primaire et secondaire. Cela permettra une utilisation plus efficiente des ressources, une meilleure planification et une prise de décision éclairée, contribuant ainsi à l'amélioration globale du système éducatif au Togo.

1.5 PROPOSITIONS ET CHOIX DE SOLUTION

1.5.1 PROPOSITIONS DE SOLUTION

Afin de trouver une solution sûre et durable aux difficultés relevées plus haut, nous proposons deux(2) solutions :

- ❖ Acquisition d'un logiciel existant : **MyScol**²,
- ❖ Conception et implémentation d'une solution développée en interne : **EduFundsManager**

1.5.1.1 MyScol

MyScol est un logiciel de gestion scolaire complet qui vise à faciliter la gestion administrative et financière des écoles. Il propose une multitude de fonctionnalités destinées à optimiser les processus internes des établissements éducatifs. L'un des points forts de MyScol réside dans sa capacité à gérer efficacement les budgets des écoles, en fournissant des outils et des fonctionnalités spécifiques pour cette tâche cruciale.

Le logiciel MyScol permet aux écoles de gérer leurs budgets de manière transparente et efficace. Voici quelques points saillants concernant la gestion budgétaire offerte par MyScol :

- ✓ Création et Planification Budgétaire
- ✓ Suivi des Dépenses et des Revenus
- ✓ Gestion des Écarts et des Prévisions
- ✓ Rapports Budgétaires



Figure 4 : Logo de MyScol

1.5.1.2 EduFundsManager

Cette solution consiste à développer une plateforme web qui permettrait la centralisation des données de manière efficace, le suivi en temps réel des dépenses, la génération de rapports financiers à partir de la plateforme, ainsi que la garantie de la sécurité et de la confidentialité des données, tout en assurant la formation des futurs utilisateurs et en mettant à leur disposition un support technique disponible.

² Lien du site internet de MyScol : <https://myscol.com/gestion-de-finance/>

1.5.2 Evaluation technique des solutions

Nous analyserons les deux solutions en tenant compte de leurs avantages et de leurs inconvénients.

1.5.2.1 Evaluation technique de : MyScol

❖ Avantages

- ✓ **Déploiement rapide** : MyScol est une solution clé en main, permettant une mise en route rapide, évitant ainsi le développement d'une solution personnalisée.
- ✓ **Support technique** : Nous pouvons bénéficier du support technique fourni par MyScol pour résoudre les problèmes et obtenir de l'aide en cas de besoin.
- ✓ **Interface utilisateur convivial** : Les modules conviviaux et intuitifs de MyScol favorisent une adoption aisée par les utilisateurs.

❖ Inconvénients

- ✓ **Personnalisation restreintes** : Les fonctionnalités prédefinies de MyScol limitent les adaptations spécifiques à nos besoins.
- ✓ **Dépendance envers le fournisseur** : En utilisant MyScol, nous dépendons du fournisseur pour les mises à jour, le support technique et les fonctionnalités futures.
- ✓ **Coûts** : Malgré des tarifs compétitifs, les coûts à long terme associés à l'utilisation de MyScol peuvent devenir considérables.

1.5.2.2 Evaluation technique de : EdufundsManager

❖ Avantages

- ✓ **Adaptation aux besoins spécifiques** : En développant une plateforme sur mesure, nous pouvons répondre précisément aux exigences et aux particularités du système éducatif togolais, assurant ainsi une meilleure adéquation avec les besoins des écoles publiques.
- ✓ **Flexibilité et évolutivité** : En concevant la plateforme de A à Z, nous avons la possibilité de la rendre flexible et facilement adaptable à l'évolution des besoins des écoles et du contexte éducatif.
- ✓ **Intégration optimisée** : En construisant la plateforme en interne, nous pouvons l'intégrer de manière transparente avec les systèmes et processus existants, minimisant ainsi les perturbations et les frictions lors de la mise en œuvre.

- ✓ **Contrôle total sur la sécurité** : En gérant la sécurité de bout en bout, nous pouvons garantir la confidentialité des données et mettre en place des mesures de protection adaptées aux besoins spécifiques et aux exigences légales en matière de protection des informations.
- ✓ **Apprentissage et expertise** : Le développement de la plateforme représente une opportunité d'apprentissage et de renforcement de notre expertise en matière de gestion financière des écoles, nous permettant ainsi de fournir un produit de qualité et adapté aux besoins du domaine.

❖ **Inconvénients**

- ✓ **Temps de développement** : La conception et le développement d'une plateforme sur mesure peuvent prendre plus de temps par rapport à l'achat d'une solution existante. Cela peut retarder la mise en œuvre et l'adoption de la plateforme par les écoles.
- ✓ **Coûts initiaux** : Le développement de la plateforme implique des coûts initiaux plus élevés en termes de main-d'œuvre, de ressources et de technologie, par rapport à l'achat d'une solution préexistante.
- ✓ **Risques de développement** : Tout projet de développement comporte des risques, tels que les retards, les erreurs de conception ou les imprévus techniques, qui pourraient affecter la qualité et le calendrier du produit final.
- ✓ **Besoin de compétences techniques** : Le développement de la plateforme nécessite des compétences techniques avancées, ce qui peut être un défi si nous n'avons pas déjà une certaine expertise dans les technologies qui seront utilisées.

1.5.3 Evaluation financière des solutions

1.5.3.1 Evaluation financière de : MyScol

- ❖ Coût du logiciel

Tableau 3 : Coût du logiciel MyScol

DESIGNATION	DESCRIPTION	COUT UNITAIRE	QUANTITÉ	COUT TOTAL (FCFA)
Coût du logiciel	Offre « MEDIUM » ³	120 € HT 73.057,56 FCFA	n ⁴	73.057,56 X n
TOTAL				73.057,56 X n

✚ Coût de formation

Tableau 4: Coût de formation

DESIGNATION	NOMBRE D'HEURES	PRIX PAR HEURE (FCFA)	COUT TOTAL (FCFA)
Formation	4 heures x 5 jours	10.000	200.000
Total			200.000

✚ Coût total

Tableau 5 : Coût total

DESIGNATION	COUT (FCFA)
Coût du logiciel	73.057,56 X n
Coût de la formation	200.000
Total	933.057,56 X n

³ Source : <https://myscol.com/tarifs/> à la date du 18/08/2023

⁴ Nombre de mois d'utilisation de la solution

1.5.3.2 Evaluation financière de : EduFundsManager

✚ Coût matériel

Tableau 6 : Coût matériel

DESIGNATION	DESCRIPTION	COUT UNITAIRE	QUANTITE	COUT TOTAL (FCFA)
Ordinateur	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz 8Go RAM 128Go SSD 15.6"	325 € ⁵ 200,044.98 FCFA (conversion du 01/08/2023)	1	325€ 200,044.98 FCFA (conversion du 01/08/2023)
Total				200.044,98 FCFA

✚ Coût de connectivité réseau

Tableau 7: Coût de connectivité réseau

DESIGNATION	DESCRIPTION	COUT UNITAIRE (FCFA)	QUANTITE	COUT TOTAL (FCFA)
Abonnement internet	FAI (Fournisseur d'accès Internet) = Canalbox Premium 50 Mb/s	30.000/Mois	3	90.000
Total				90.000

⁵ Source : <https://www.fnac.com/mp48492719/PC-portables-Dell-Latitude-E5570-i5-8-250SSD-Full-HD-Poids-1-30-kg/w-4> au 01/08/2023

✚ Coût d'hébergement

Tableau 8: Coût d'hébergement

DESIGNATION	DESCRIPTION	COUT UNITAIRE	QUANTITE	COUT TOTAL (FCFA)
Serveur d'hébergement HEROKU (Pour la partie Backend)	Offre «Production » ⁶	25 \$ 14.986,25 FCFA (conversion du 14/08/2023)	n ⁷	14.986,25 X n
Serveur d'hébergement NETLIFY (Pour la partie frontend)	Offre «PRO» ⁸	19 \$/mois 11.842,51 FCFA (Conversion du 15/08/2023)	n ⁹	11.842,51 X n
Total				26.828,76 X n

✚ Coût de formation

Tableau 9: Coût de formation

DESIGNATION	NOMBRE D'HEURES	PRIX PAR HEURE (FCFA)	COUT TOTAL (FCFA)
Formation	4 heures x 5 jours	10.000	200.000
Total			200.000

⁶ Source : <https://www.heroku.com/pricing> à la date du 14/08/2023

⁷ Le nombre de mois d'utilisation de la solution

⁸ Source : <https://www.netlify.com/pricing/> à la date du 15/08/2023

⁹ Le nombre de mois d'utilisation de la solution

Coût total

Tableau 10 : Coût total

DESIGNATION	COUT (FCFA)
Coût matériel	200.044,98
Coût d'hébergement	26.828,76 X n
Coût de connectivité réseau	90.000
Coût de conception	6.080.000
Coût formation	200.000
Total	6.596.873,74

1.5.4 Choix de la solution

Après une évaluation approfondie des deux options disponibles, à savoir l'achat de la solution préexistante MyScol et le développement de notre propre plateforme nommée EduFundsManager, Odda Technology a pris la décision de ne pas se tourner vers l'acquisition d'une solution déjà existante, étant donné qu'elle ne répondrait pas de manière précise aux exigences spécifiques énoncées dans notre contexte.

Au lieu de cela, nous avons opté pour le développement sur mesure de la plateforme web EduFundsManager. Cette décision représente une occasion unique d'apprentissage et de développement pour nous. En choisissant de développer en interne cette plateforme, nous serons en mesure de créer une solution parfaitement adaptée aux besoins particuliers des écoles publiques de l'enseignement primaire et secondaire au Togo.

Cette approche sur mesure nous permettra non seulement de relever les défis spécifiques liés à la gestion des budgets scolaires, mais également de renforcer notre expertise dans le domaine de la gestion financière des institutions éducatives. Cette expérience d'apprentissage enrichissante renforcera notre capacité à fournir des solutions numériques innovantes et spécifiquement adaptées à des besoins précis.

1.6 PLANNING PREVISIONNEL DE REALISATION

Le planning prévisionnel représente le chronogramme nous permettant de structurer et de planifier les activités durant la période de notre stage, de définir des échéances et de faciliter le suivi et la coordination des différentes étapes jusqu'à la réalisation du projet de stage.

Tableau 11 : Planning prévisionnel de réalisation

N°	Tâches	Date de début	Date de fin	Durée en jours
1	Définition du planning prévisionnel	22 Mai 2023	18 Août 2023	76
2	Insertion dans l'entreprise	22 Mai 2023	23 Mai 2023	1
3	Prise de connaissance sur le thème	24 Mai 2023	25 Mai 2023	1
4	Collecte d'informations sur le thème et analyse initiale des spécifications	26 Mai 2023	07 Juin 2023	12
5	Apprentissage des outils logiciels, langages et frameworks à utiliser	08 Juin 2023	18 Juin 2023	10
6	Programmation	19 Juin 2023	09 Juillet 2023	20
7	Test logiciel et corrections	10 Juillet 2023	17 Juillet 2023	7
8	Validation du mémoire et corrections	18 Juillet 2023	20 Juillet 2023	2
9	Dossier de réalisation	21 Juillet 2023	25 Juillet 2023	4
10	Dossier d'exploitation et guide d'utilisation	26 Juillet 2023	30 Juillet 2023	4

N°	Tâches	Date de début	Date de fin	Durée en jours
11	Correction du mémoire et de l'application	01 Août 2023	05 Août 2023	4
12	Validation à nouveau du mémoire et des corrections	06 Août 2023	16 Août 2023	10
13	Test de déploiement	17 Août 2023	18 Août 2023	1
14	Validation définitive avec les participants au projet	18 Août 2023	19 Août 2023	1

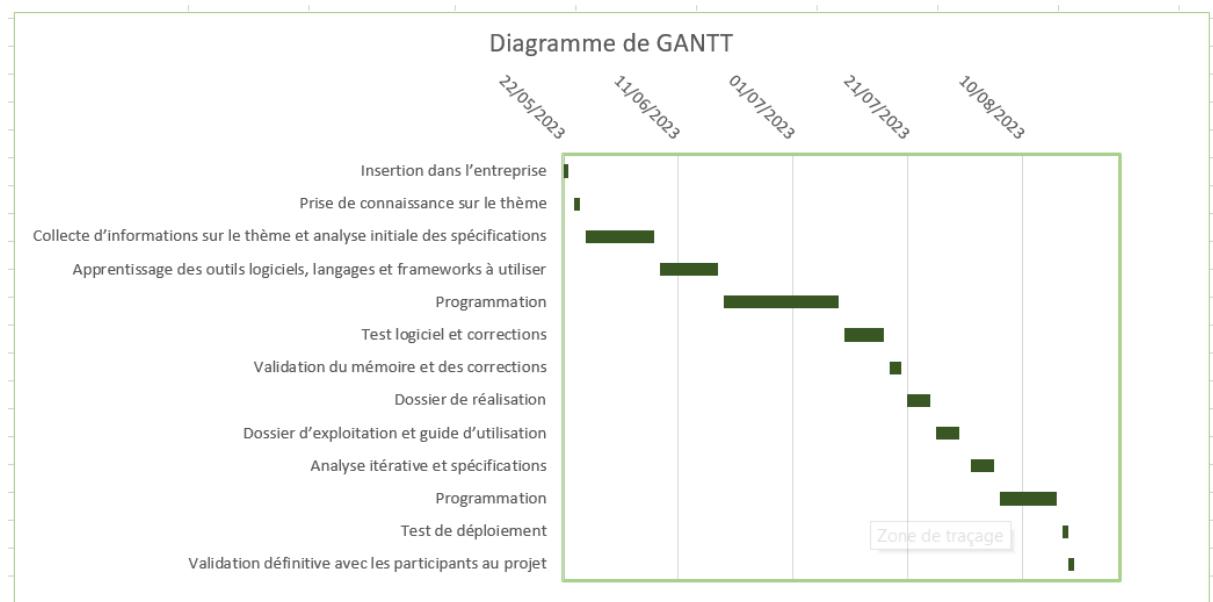
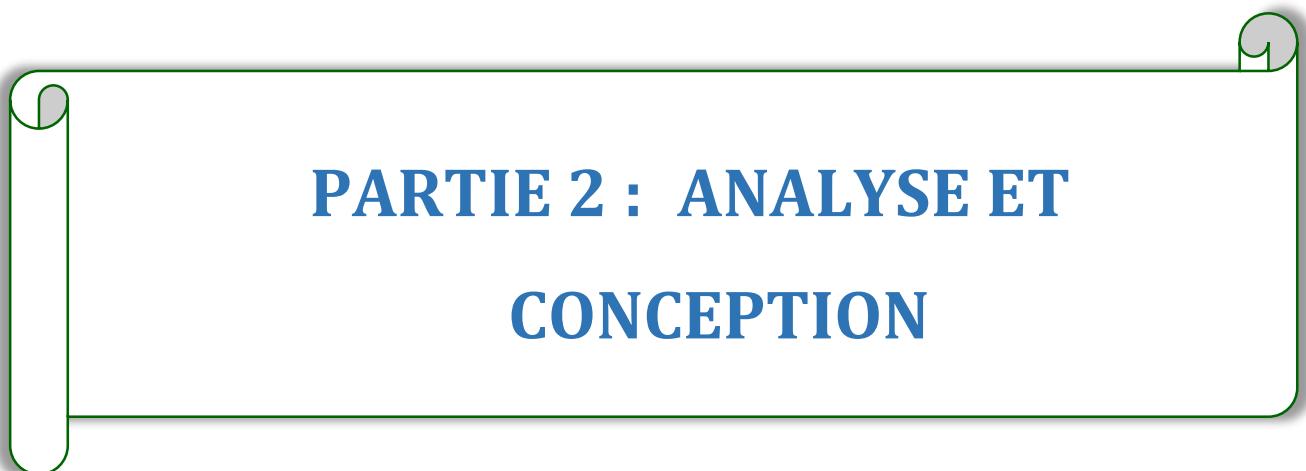


Figure 5 : Diagramme de GANTT



PARTIE 2 : ANALYSE ET CONCEPTION

L'analyse et la conception sont des étapes cruciales dans la réalisation d'un projet informatique. Pour mener à bien ce processus, un document clé est le dossier d'analyse et de conception. Il sert à décrire le fonctionnement du système d'information, à définir les démarches à suivre et à mettre en place les outils nécessaires à la réalisation du projet. Dans le cadre de notre projet, nous avons choisi d'utiliser comme langage de modélisation Unified Modeling Language (UML) couplé avec le processus 2TUP pour tirer parti de leurs nombreux avantages. Cette partie du mémoire présentera les outils d'analyse et de modélisation, les différentes étapes de modélisation et de conception, ainsi que les diagrammes statiques et dynamiques qui permettent de mieux comprendre le système et ses fonctionnalités.

2.1 CHOIX ET JUSTIFICATION DE LA METHODE D'ANALYSE

La présentation de la méthode d'analyse est consacrée essentiellement à la justification de l'approche objet et à la présentation du langage UML couplé avec le processus 2TUP.

2.1.1 L'approche objet

L'approche objet désigne l'ensemble des processus et langages utilisés au cours du cycle de vie de l'application, qui reposent sur la manipulation des objets. Ici, une application est vue comme un ensemble d'objets qui coopèrent. Un autre concept qui est très souvent lié à l'approche objet est celui de classe qui permet de regrouper les propriétés communes des objets. C'est un modèle structurel d'objets, à partir duquel, il est possible de fabriquer autant d'objets nécessaires. Ce concept est associé à la notion de composants logiciels réutilisables, facilitant la production des logiciels et diminuant l'effort de tests.

2.1.2 Pourquoi l'approche objet et non l'approche fonctionnelle ?

L'approche fonctionnelle, depuis toujours, a fait ses preuves et reste aujourd'hui un moyen efficace pour les solutions informatiques. Cependant, avec l'évolution des technologies, les solutions informatiques sont devenues de plus en plus complexes et évolutives. L'approche fonctionnelle fléchit devant certaines contraintes spécifiques qu'imposent les solutions informatiques. De nouvelles méthodes de cette approche ont vu le jour. Aussi, d'autres anciennes méthodes de l'approche fonctionnelle ont dû introduire

de nouveaux concepts pour s'adapter. C'est le cas de MERISE 2 qui est une évolution de MERISE. Toujours dans la recherche des moyens efficaces de résolution des solutions informatiques de plus en plus complexes, l'approche objet vit le jour. Cette approche intègre à la fois les aspects statiques et dynamiques des applications.

Elle favorise :

- La réutilisation des composants ;
- La stabilité dans le temps ;
- L'indépendance des données et des méthodes ;
- La construction par le couplage faible entre les composants ;
- L'aisance dans la maintenance des logiciels aidée par la facilité à circonscrire les tâches à effectuer.

L'approche objet pour pouvoir assurer ces tâches se base sur un langage (pour s'exprimer clairement à l'aide des concepts objets).

Le langage doit permettre :

- De représenter des concepts abstraits (graphiquement) ;
- De limiter les ambiguïtés (parler un langage commun ; indépendant des langages orientés objet) et faciliter l'analyse (simplifier la comparaison et l'évaluation de solutions) ;
- Une démarche d'analyse et de conception objet : Elle est nécessaire non pas pour effectuer une analyse fonctionnelle et se contenter d'une implémentation objet, mais pour penser objet dès le départ et définir les vues qui permettent de décrire tous les aspects d'un système avec des concepts objets. Pour notre étude nous opterons pour le langage UML (Unified Modeling Language) couplé avec le processus 2TUP.

2.1.3 Le langage de modélisation UML



Figure 6 : Logo de UML

Avant de présenter le langage de modélisation UML, il est important de comprendre le concept de modélisation et son rôle dans le processus de la conception logicielle.

La modélisation est un processus essentiel dans le développement d'un système informatique. Elle consiste à représenter de manière abstraite les éléments du monde réel pour faciliter leur étude et leur documentation. Les modèles permettent de clarifier les aspects complexes et de tracer les interactions entre les différentes parties du système. Dans le domaine de la conception logicielle, la modélisation est indispensable pour comprendre les processus des systèmes d'information et faciliter la communication entre les acteurs du projet.

UML propose trois modes de représentation pour modéliser un système de manière complète :

- **Les diagrammes structurels ou statiques :**
 - ✓ Le diagramme de classes ;
 - ✓ Le diagramme d'objets ;
 - ✓ Le diagramme de composants ;
 - ✓ Le diagramme de déploiement ;
 - ✓ Le diagramme de structures composites ;
 - ✓ Le diagramme de packages ;
 - ✓ Le diagramme de profils.

- **Les diagrammes de comportement :**
 - ✓ Le diagramme de cas d'utilisation ;
 - ✓ Le diagramme d'activité ;
 - ✓ Le diagramme d'état-transition.
- Les diagrammes d'interaction ou dynamiques :
 - ✓ Le diagramme de séquence ;
 - ✓ Le diagramme de communication ;
 - ✓ Le diagramme global d'interaction ;
 - ✓ Le diagramme de temps.

Ces trois modes de représentation sont complémentaires et nécessaires pour décrire la composition du système et le fonctionnement de ses composantes. Dans la modélisation des solutions informatiques, l'ordre des diagrammes n'est pas imposé par UML. Cependant, l'utilisation de méthodes spécifiques peut aider à définir une approche cohérente.

Ainsi, il est recommandé de combiner UML avec un processus de développement approprié afin de produire des applications de qualité répondant aux besoins des utilisateurs. UML peut être utilisé comme une méthode d'analyse et de conception intégrée au processus de développement, permettant de modéliser efficacement les solutions informatiques.

Parmi les processus qui peuvent être couplés à UML, nous avons identifié les suivants :

- UP (Unified Process) ;
- 2TUP (Two Track Unified Process) ;
- Méthode Agile (Scrum, XP, etc.) ;
- RUP (Rational Unified Process).

Dans notre cas, nous avons choisi d'utiliser UML en combinaison avec le processus 2TUP, en raison des avantages qu'il offre :

- Approche itérative et incrémentale, permettant une évolution progressive du système ;
- Focalisation sur l'architecture du système, assurant une base solide pour le développement.

- Séparation des besoins fonctionnels et techniques, permettant une meilleure compréhension des exigences.
- Approche orientée composant, favorisant la réutilisation et la modularité du système.
- Prise en compte des besoins et attentes des utilisateurs, assurant une meilleure satisfaction.
- Gestion des risques intégrée, permettant une identification précoce et une mitigation efficace.
- Utilisation de la modélisation graphique à l'aide d'UML, facilitant la communication et la compréhension entre les parties prenantes.

2.1.4 Présentation du processus 2TUP

Le processus **Two Track Unified Process** (2TUP) est une approche de développement logiciel qui met en œuvre le Processus Unifié. Le 2TUP propose un cycle de développement en forme de Y, qui sépare les aspects techniques des aspects fonctionnels.

Le processus débute par une étude préliminaire, où l'accent est mis sur l'identification des acteurs qui vont interagir avec le système à développer, ainsi que sur l'analyse des messages échangés entre les acteurs et le système. Cette étape permet de produire le cahier des charges et de modéliser le contexte du système (le système est considéré comme une boîte noire).

Ensuite, le processus se déroule autour de trois phases clés : une branche technique, une branche fonctionnelle, et une phase de réalisation.

1. **La branche technique** se concentre sur les aspects techniques du système, tels que l'architecture logicielle, la sélection des technologies appropriées, la définition des normes de codage, et la mise en place des infrastructures nécessaires. Cette branche permet de garantir la stabilité et la performance du système.
2. **La branche fonctionnelle** met l'accent sur les fonctionnalités du système. Elle vise à définir les cas d'utilisation, à spécifier les besoins des utilisateurs, à concevoir l'interface utilisateur, et à modéliser les différents scénarios d'utilisation. Cette branche permet de s'assurer que le système répond aux attentes fonctionnelles des utilisateurs.

3. Enfin, **la phase de réalisation** consiste à mettre en œuvre les fonctionnalités du système en utilisant les résultats des branches technique et fonctionnelle. Cette phase englobe le développement, les tests, la correction des erreurs, et la livraison du produit final.

Le processus 2TUP offre une approche structurée et itérative pour le développement de logiciels. En dissociant les aspects techniques et fonctionnels, il permet une meilleure gestion des risques, une meilleure compréhension des besoins utilisateurs, et une meilleure qualité du produit final.

Par ailleurs, ce processus unifié repose sur un cycle de développement en Y qui comprend sept (7) étapes principales :

Étape 1. Étude préliminaire

Au cours de cette étape, nous identifions les différents acteurs du futur système. Pour décrire cette étape, nous utilisons le diagramme de contexte d'UML.

Étape 2. Capture des besoins fonctionnels

Cette étape consiste à modéliser les aspects fonctionnels du système, notamment la spécification des fonctionnalités et des interactions entre les acteurs et le système. Pour cela, nous utilisons les diagrammes de cas d'utilisation, d'activité et de séquence système d'UML.

Étape 3. Capture des besoins techniques

Cette étape est consacrée à l'élaboration de l'architecture matérielle et logicielle du système.

Étape 4. Analyse

Dans cette étape, nous modélisons les aspects statiques de la solution, en déterminant les différents objets utilisés par le système. Pour cela, nous utilisons le diagramme de classes d'UML.

Étape 5. Conception

La conception se compose de trois (3) sous-étapes :

- ❖ **Conception générique** : détermination des différents composants dynamiques du système ;
- ❖ **Conception préliminaire** : établissement de la cartographie des composants ;
- ❖ **Conception détaillée** : détermination des différentes méthodes de réalisation des composants.

Pour décrire cette étape, nous utilisons le diagramme de déploiement d'UML.

Étape 6. Codage et tests

Pendant cette étape, nous produisons les composants du système et les testons progressivement.

Étape 7. Recette

Cette étape vise à valider les fonctionnalités du système développé.

Ces sept étapes du processus 2TUP offrent une approche structurée et itérative pour la réalisation d'un projet informatique, en garantissant la prise en compte des besoins fonctionnels et techniques, ainsi que la qualité du produit final.

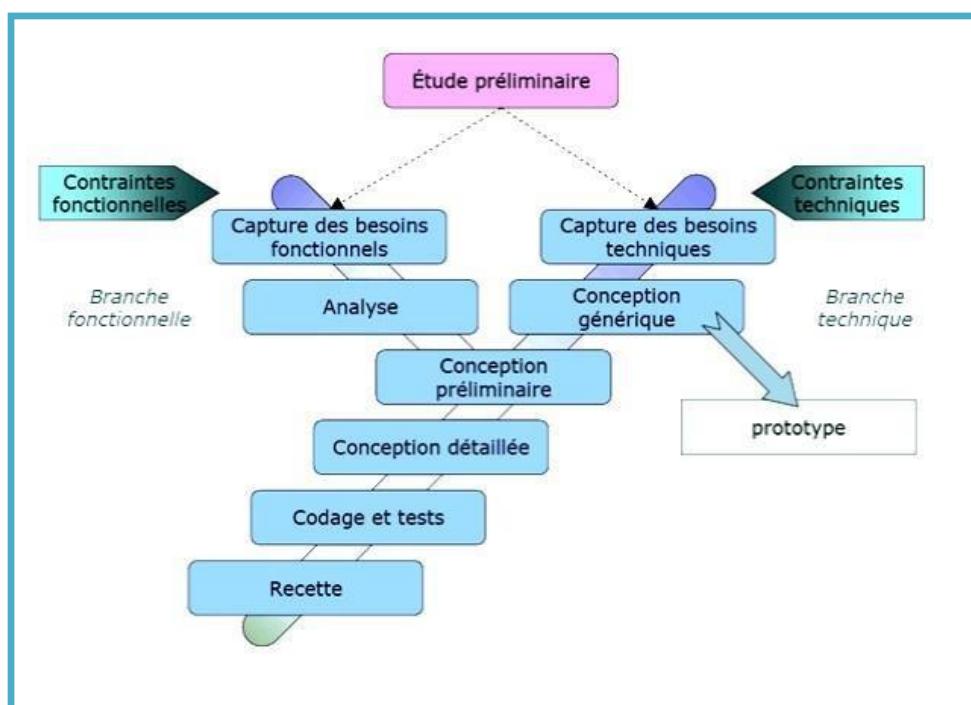


Figure 7 : Illustration du processus 2TUP

En combinant UML avec le processus 2TUP, nous visons à bénéficier des avantages de ces deux approches pour la réalisation de notre projet.

2.2 CHOIX ET JUSTIFICATION DE L'OUTIL DE MODELISATION

Dans le domaine de la modélisation UML, plusieurs outils sont disponibles pour nous aider à représenter notre système. Parmi ces outils, nous avons considéré une variété d'options telles que StarUML, BOUML, Modélio, ArgoUML, PowerAMC, Astah, PowerDesigner, Umbrello, CodeDesigner, UMLStudio et Enterprise Architect.

Après avoir évalué ces différents outils, notre choix s'est finalement porté sur PowerAMC. Ce dernier répond à nos besoins spécifiques et offre les fonctionnalités nécessaires pour mener à bien notre analyse et atteindre les objectifs de notre projet.

2.2.1 Présentation et justification du choix de PowerAMC

PowerAMC est un outil de modélisation et de conception de logiciels utilisé par de nombreux professionnels de l'informatique. Il offre une large gamme de fonctionnalités pour la création de modèles, tels que les diagrammes de classes, les diagrammes de flux de données, les diagrammes de séquence, etc. Il permet de visualiser et de représenter visuellement la structure, le comportement et les interactions des systèmes logiciels.

Voici quelques raisons pour lesquelles nous avons choisi PowerAMC :

- ✓ **Familiarité avec l'outil** : C'est l'outil avec lequel nous sommes le plus familiarisé ;
- ✓ **Interface conviviale** : PowerAMC dispose d'une interface utilisateur intuitive et conviviale, ce qui facilite la création et la modification des modèles. Il offre une expérience utilisateur agréable et permet une prise en main rapide ;
- ✓ **Large gamme de diagrammes** : PowerAMC propose une variété de diagrammes, ce qui permet de représenter différents aspects d'un système logiciel. Nous pouvons choisir parmi les diagrammes de classes, les diagrammes de flux de données, les diagrammes de séquence, les diagrammes d'état-transition, etc., en fonction de nos besoins de modélisation.

2.3 ETUDE DETAILLEE DE LA SOLUTION

Dans cette section, nous présenterons les diagrammes d'UML en suivant la structure recommandée par 2TUP.

2.3.1 Etude préliminaire

Dans cette première étape, nous identifions les acteurs qui interagiront avec le système. En utilisant le diagramme de contexte d'UML, nous représentons visuellement ces interactions, ce qui nous permet d'avoir une vue d'ensemble du système.

Identification des acteurs

Un acteur, est une entité externe qui interagit avec un système. Il peut s'agir d'une personne, d'un autre système, d'un appareil ou d'une organisation qui a un rôle défini dans le fonctionnement du système informatique. L'acteur peut initier des actions, recevoir des informations ou influencer le comportement du système.

On distingue :

➤ **Les acteurs principaux**

Il s'agit des acteurs qui utilisent directement le système pour atteindre des objectifs. Ce sont les utilisateurs principaux ou les bénéficiaires finaux du système. Les acteurs principaux interagissent activement avec le système et peuvent effectuer des actions qui déclenchent des fonctionnalités spécifiques.

➤ **Les acteurs secondaires**

Il s'agit des acteurs qui interagissent indirectement avec le système, généralement par le biais d'un acteur principal. L'acteur secondaire peut fournir des informations, des ressources ou des services au système, ou peut être affecté par les résultats du système. Il ne participe pas directement aux fonctionnalités principales du système mais peut avoir un rôle d'assistance ou de support.

Dans notre projet, nous avons identifié les acteurs suivants :

➤ **Les acteurs principaux**

- **L'administrateur**

Cet acteur porte le rôle de plus haut niveau dans la hiérarchie du système. Il est responsable de la configuration globale du système, impactant ainsi l'ensemble de son fonctionnement. En tant qu'acteur clé, il détient les pouvoirs décisionnels et influence les paramètres essentiels du système pour assurer son bon fonctionnement.

- **Le directeur administratif et financier**

Cet acteur, situé directement après l'administrateur dans la hiérarchie du système, joue également un rôle consultatif en accédant aux informations des institutions de niveau inférieur. Selon l'organigramme de notre futur système, nous distinguons deux types de Directeurs Administratifs et Financiers (DAF) : l'un pour l'enseignement technique et l'autre pour l'enseignement général.

- **Le directeur régional**

Cet acteur, situé directement après le DAF joue un rôle consultatif en accédant aux informations des institutions de niveau inférieur dans la hiérarchie du système. De plus, il a le pouvoir de gérer des révisions budgétaires pour les écoles relevant de sa direction régionale.

- **L'inspecteur**

L'inspecteur joue principalement un rôle consultatif en accédant aux informations des écoles placées sous son autorité dans la hiérarchie du système. En plus de cela, il a le pouvoir d'approuver les demandes de modification des montants de dépenses effectuées par ces écoles.

- **Le directeur d'école**

Il joue un rôle de gestion des informations liées à de son école. Il est également chargé de définir les sources de financement et les imputations budgétaires pour l'exercice de chaque année scolaire.

- **Le comptable d'école**

Le comptable d'école est responsable de l'enregistrement de toutes les dépenses effectuées par l'établissement tout au long de l'année dans le système. De plus, il est

chargé de soumettre les demandes de modification des montants de dépenses de son école à l'inspection hiérarchique.

- **Organigramme du futur système**

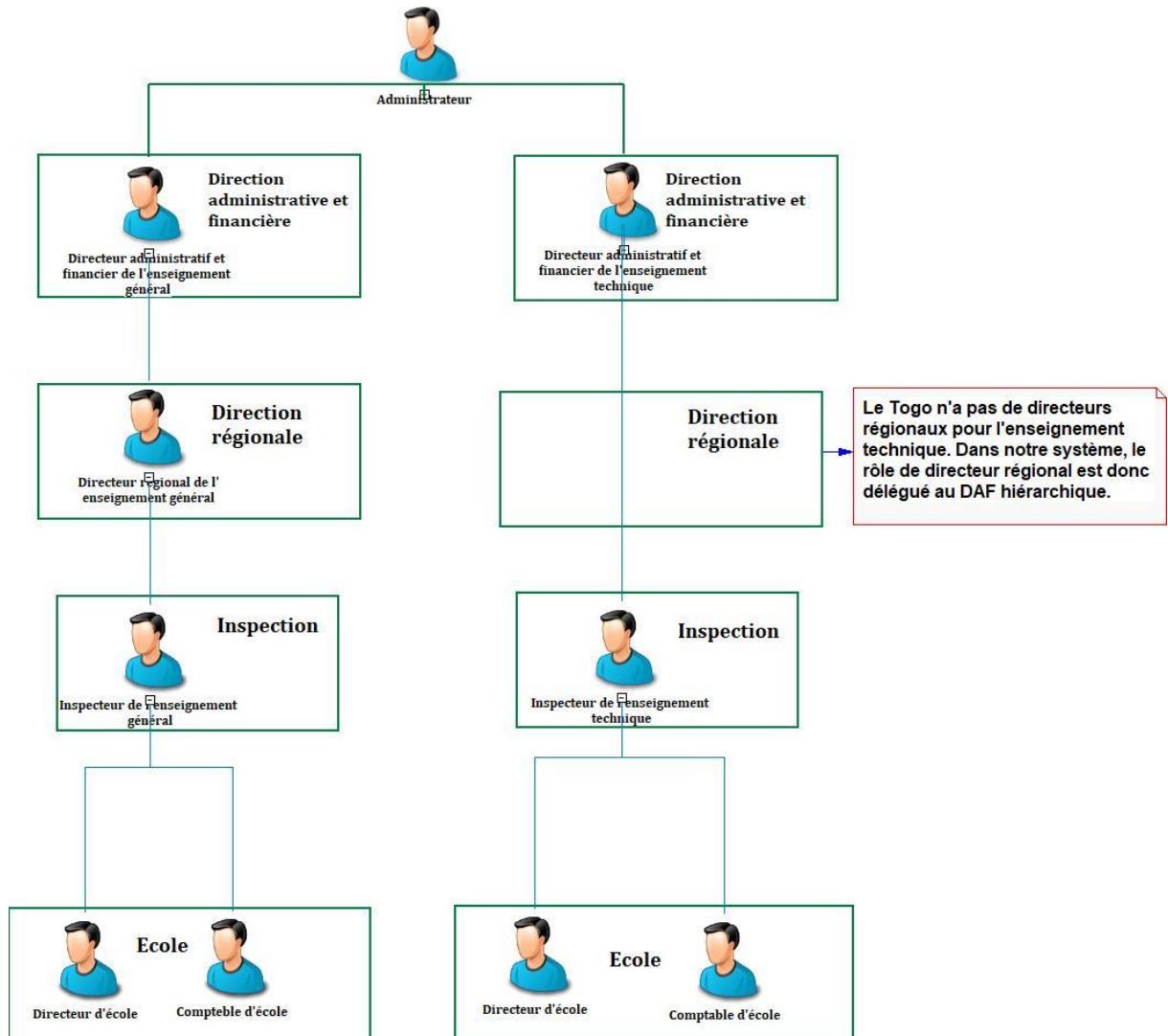


Figure 8: Organigramme du futur système

➤ **Les acteurs secondaires**

- **Serveur SMTP**

C'est le serveur mail qui intervient dans l'envoi des emails.

2.3.1.1 Diagramme de contexte statique

Le diagramme de contexte statique est un outil non officiel dans le cadre de UML, mais il est utilisé pour décrire le domaine d'étude de manière globale avant de se plonger dans les détails. Il permet de représenter le système dans son environnement et de visualiser

les acteurs externes et les interactions principales. Voici le diagramme de contexte statique de notre système :

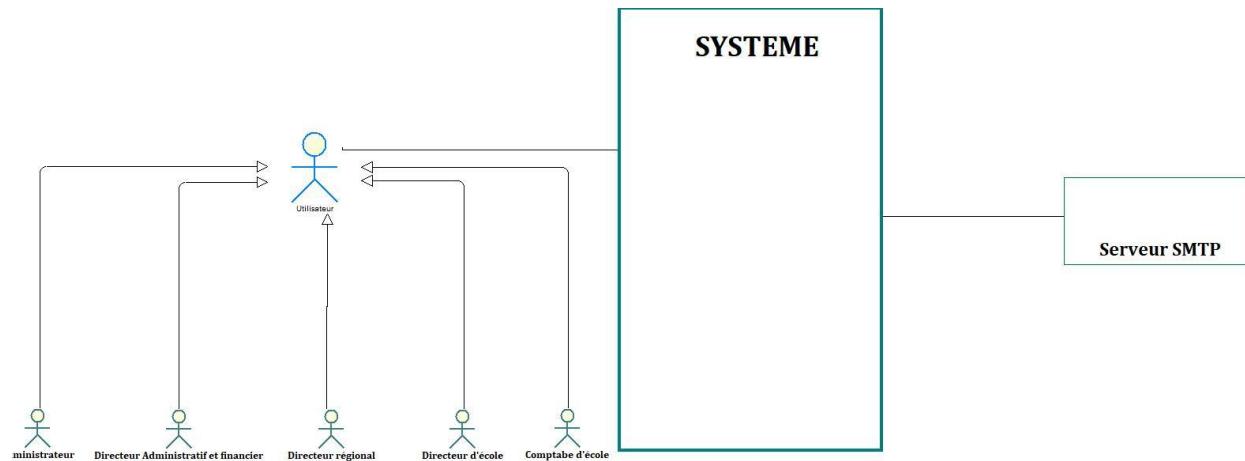


Figure 9 : Diagramme de contexte statique

2.3.2 Capture des besoins fonctionnels

Dans cette étape, la concentration est portée sur la modélisation des fonctionnalités du système. Les diagrammes de cas d'utilisation, d'activité et de séquence système sont utilisés pour représenter les interactions et le flux des actions dans le système.

2.3.2.1 Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation (ovales) reliés par des associations (lignes) à des acteurs (icône du "stick man" ou représentation graphique équivalente).

Il permet de décrire les fonctionnalités offertes par le système du point de vue des acteurs, en mettant en évidence les actions, les scénarios d'utilisation et les relations entre les acteurs et les cas d'utilisation. Cela permet de visualiser de manière claire et concise les interactions clés entre les utilisateurs et le système.

❖ Les cas d'utilisation

Les cas d'utilisation sont des descriptions succinctes des interactions entre les utilisateurs et le système. Ils identifient les fonctionnalités clés et décrivent les actions réalisées par les utilisateurs et les réponses du système. Les cas d'utilisation servent à comprendre les besoins des utilisateurs et à guider la conception du système.

Les cas d'utilisation de notre projet sont résumés dans le tableau suivant :

Tableau 12 : Les cas d'utilisations du projet

CAS D'UTILISATION	ACTEURS
Gérer les comptes	Créer un compte
	Retirer les droits
Gérer les sources	Ajouter
	Modifier
	Supprimer
Gérer les rubriques	Ajouter
	Modifier
	Supprimer
Gérer les années scolaires	Ajouter
	Modifier
	Supprimer
Gérer les directions financières	Ajouter
	Modifier
	Supprimer
Gérer les directions régionales	Directeur administratif et financier
	Ajouter
	Modifier
Gérer les inspections	Supprimer
	Ajouter
	Modifier
	Supprimer

CAS D'UTILISATION	ACTEURS
Gérer les écoles	Ajouter
	Modifier
	Supprimer
Gérer les comptes	Créer un compte
	Retirer les droits
Gérer budget	Ajouter
	Modifier
	Supprimer
Gérer les imputations	Ajouter
	Modifier le montant
	Supprimer
Gérer les financements	Ajouter
	Modifier
	Supprimer
Gérer les dépenses	Ajouter
	Modifier
	Supprimer
Gérer les factures	Ajouter
	Modifier
Gérer les demandes de modification de montant	Faire une demande
	Annuler une demande

CAS D'UTILISATION	ACTEURS
Modifier dépense	Inspecteur
Consulter les statistiques	Administrateur, Directeur administratif et financier, Directeur régional, Inspecteur, Directeur d'école, Comptable d'école
Exporter les statistiques	Administrateur, Directeur administratif et financier, Directeur régional, Directeur d'école, Inspecteur, Comptable d'école
Gérer profil	Modifier Profil Changer son mot de passe Récupérer son compte

❖ **Représentation des diagrammes de cas d'utilisation par acteur**

➤ Pour l'administrateur

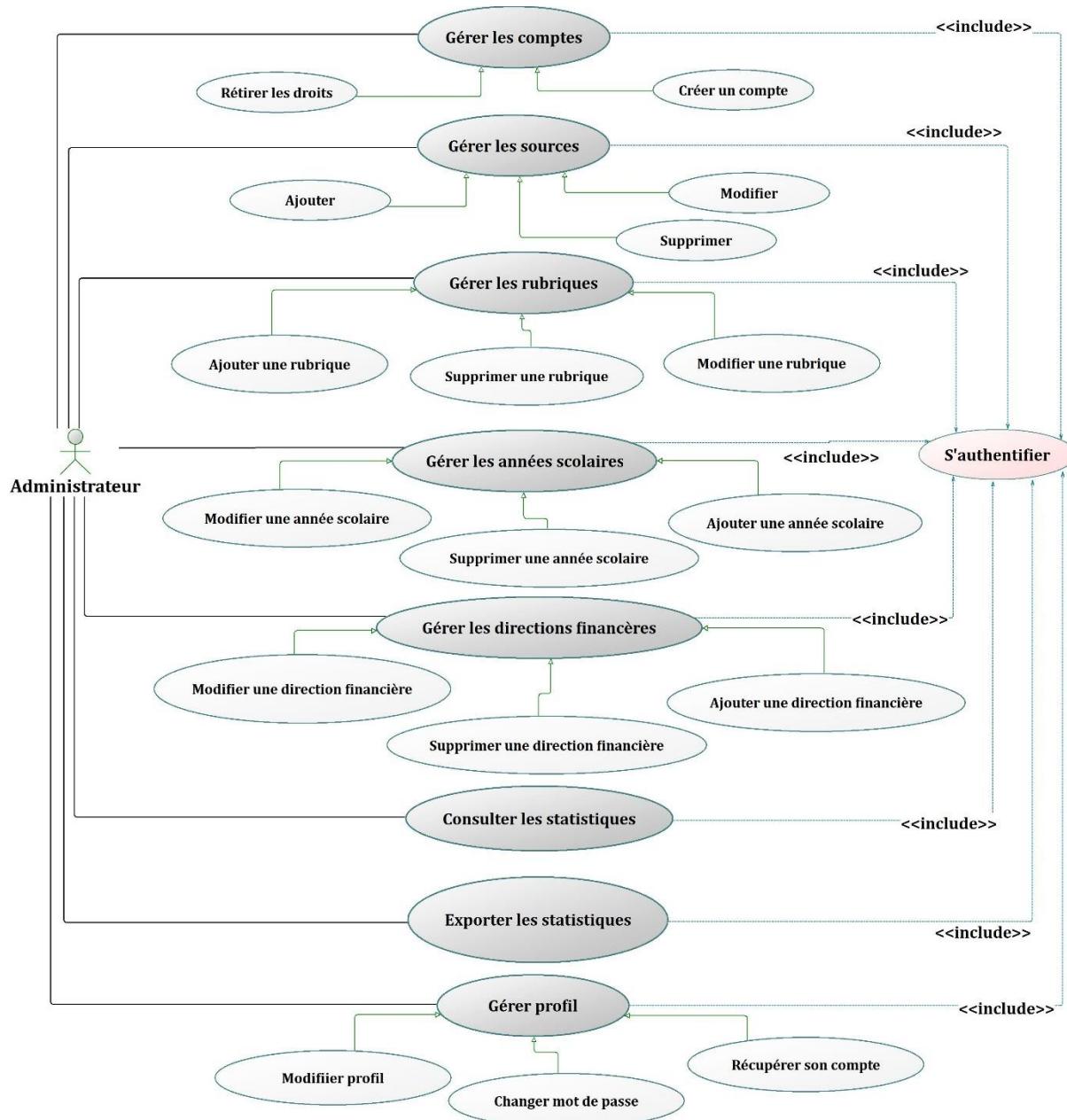


Figure 10 : Cas d'utilisation pour l'acteur "Administrateur"

➤ Pour le directeur administratif et financier

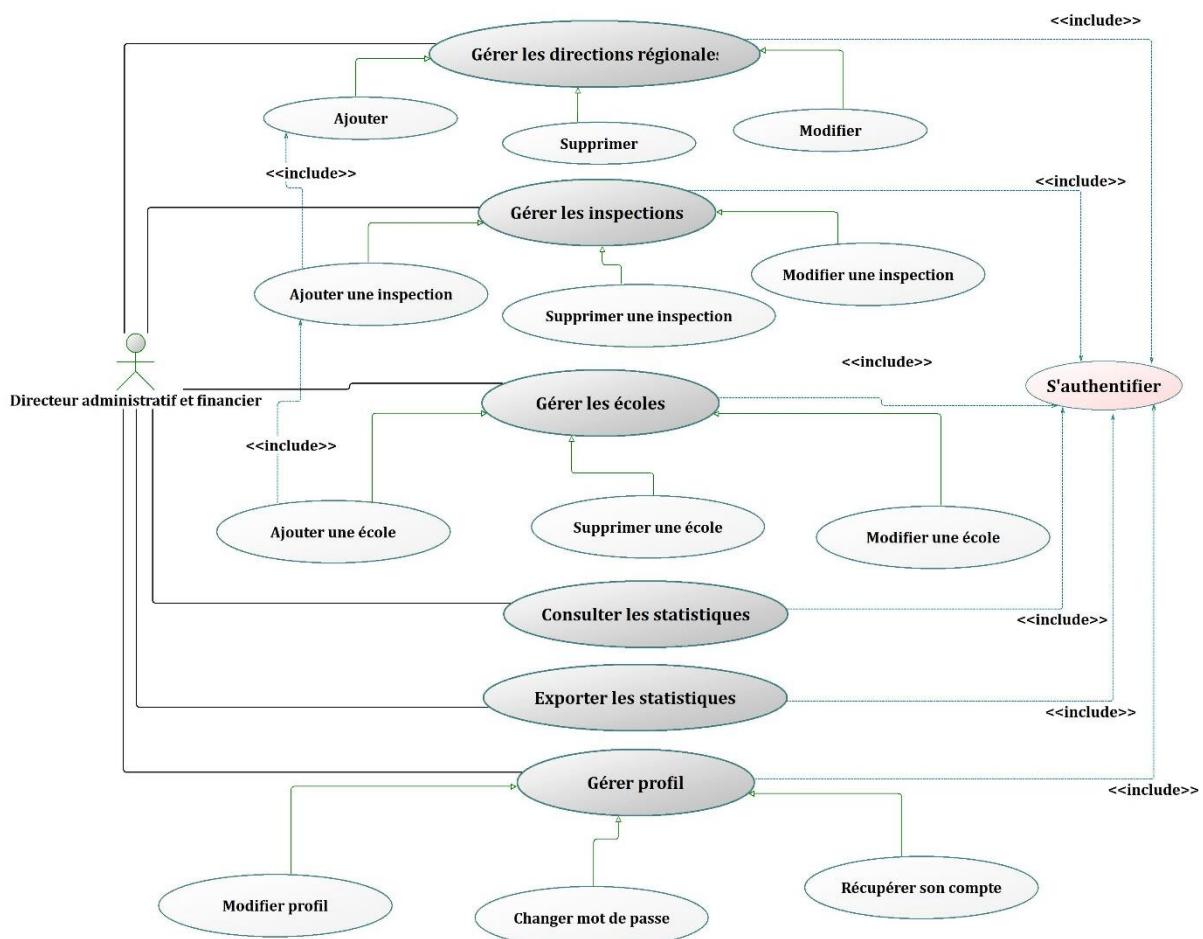


Figure 11 : Cas d'utilisation pour l'acteur "Directeur administratif et financier"

➤ Pour le directeur régional

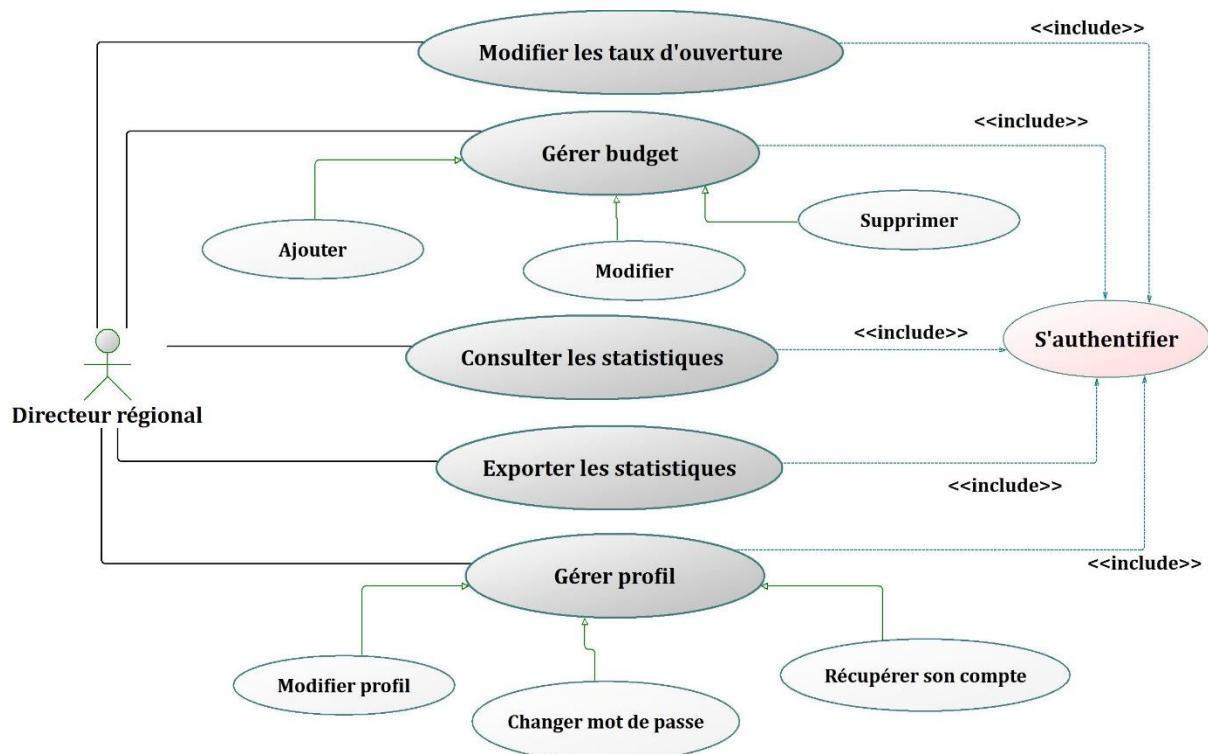


Figure 12 : Cas d'utilisation pour l'acteur "Directeur régional"

➤ Pour l'inspecteur

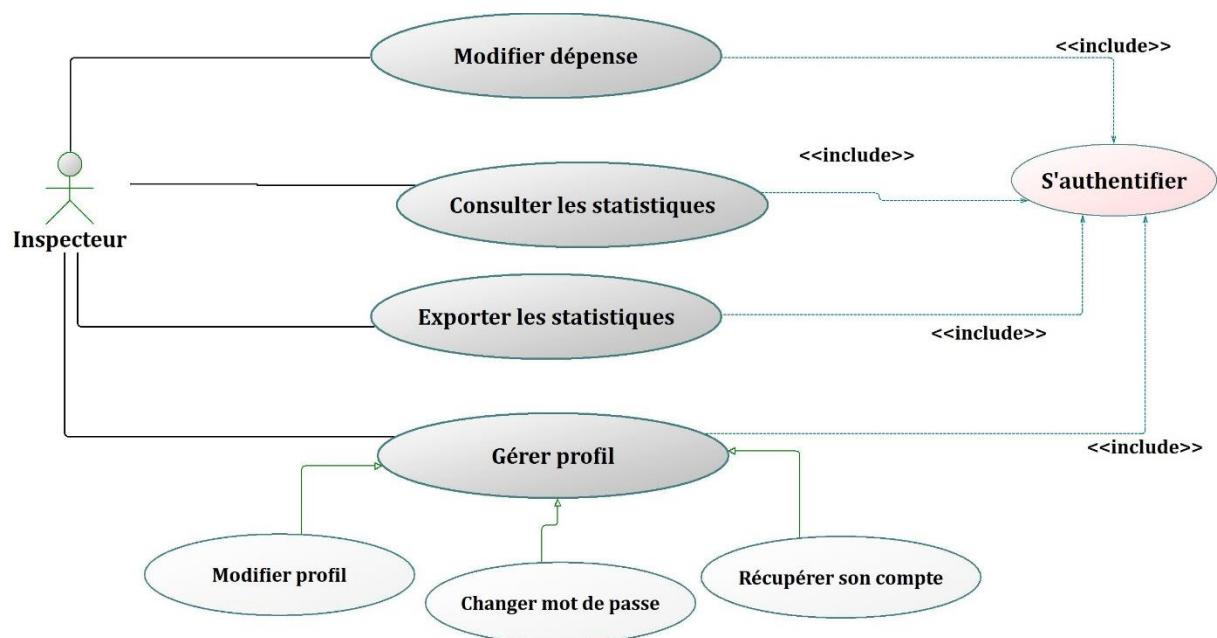


Figure 13 : Cas d'utilisation pour l'acteur "Inspecteur"

➤ Pour le directeur d'école

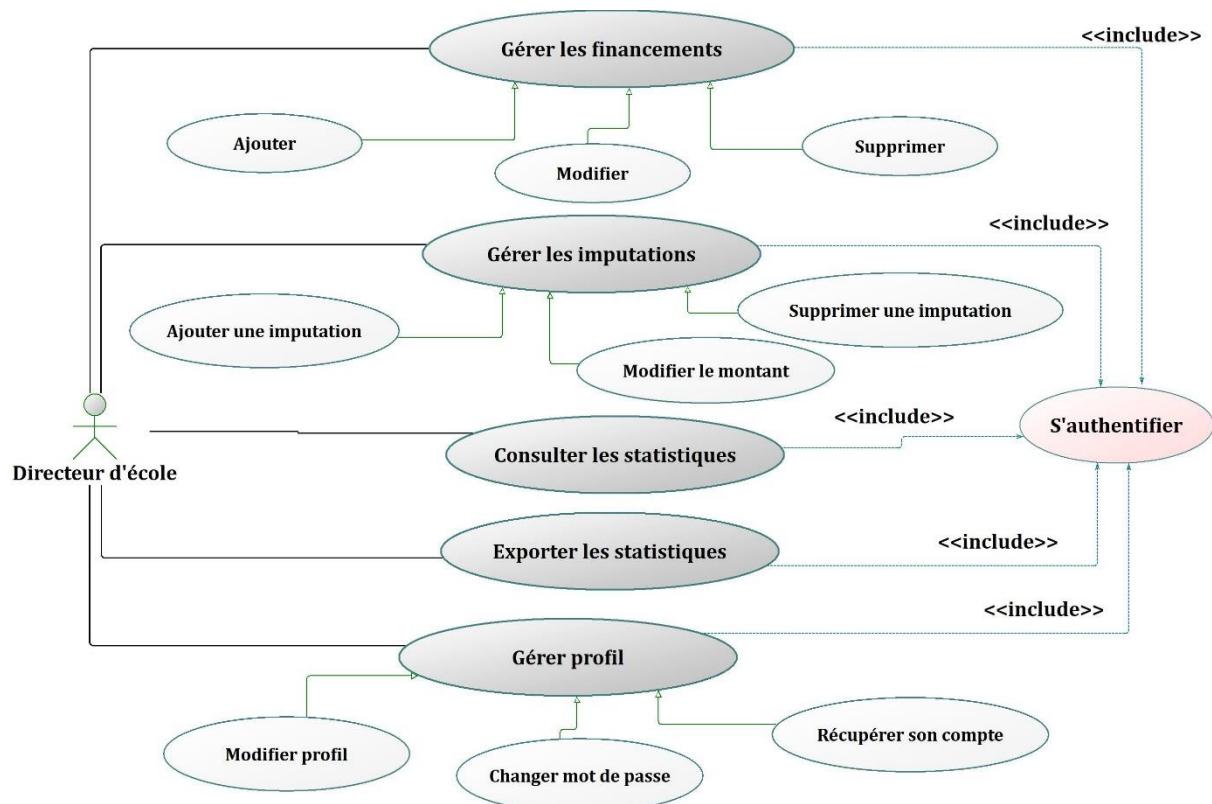


Figure 14 : Cas d'utilisation pour l'acteur "Directeur d'école"

➤ Pour le comptable d'école

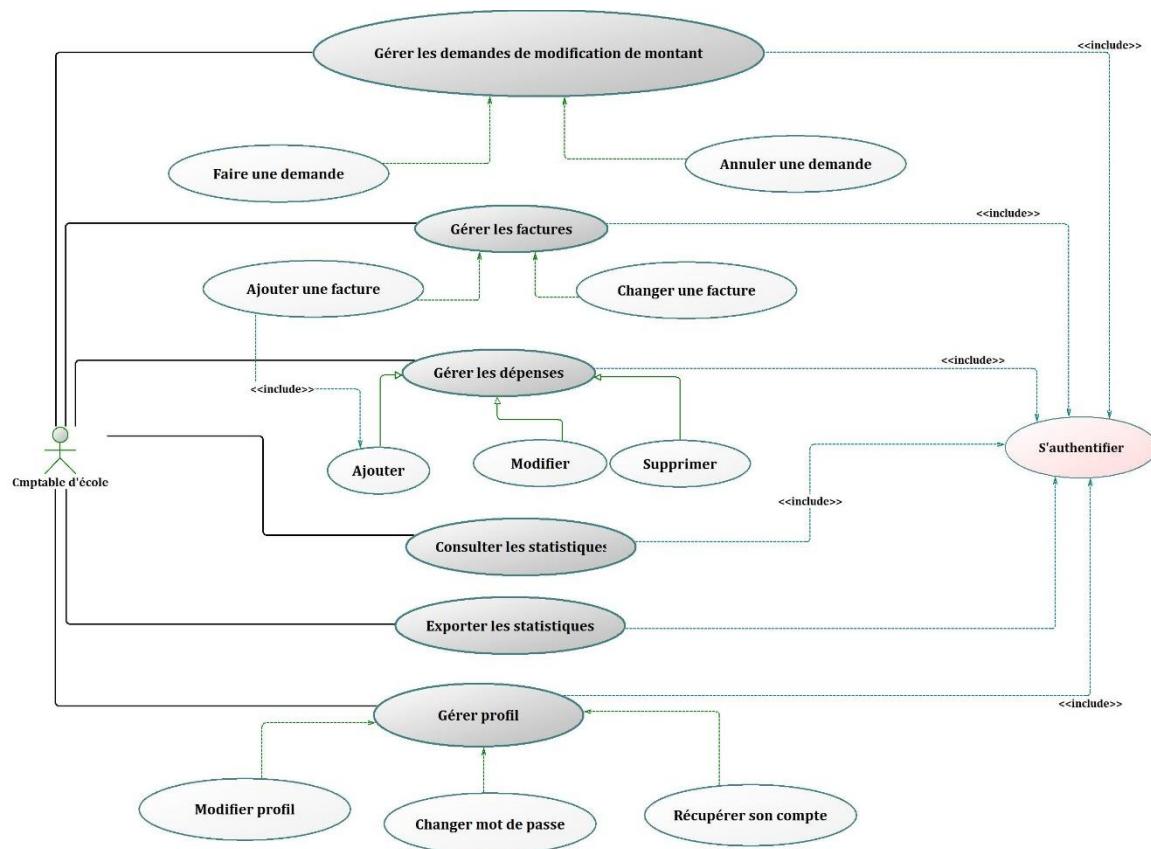


Figure 15 : Cas d'utilisation pour l'acteur "Comptable d'école"

2.3.2.2 Description textuelle de quelques cas d'utilisation

La description d'un cas d'utilisation consiste à recenser de manière textuelle toutes les interactions qui se produisent entre les acteurs et le système. Chaque cas d'utilisation doit être clairement défini avec un début et une fin, indiquant quand ces interactions ont lieu et les messages échangés. Il est également important d'inclure les différentes variantes possibles, telles que les cas nominaux, alternatifs et d'erreurs, tout en organisant les descriptions de manière à les rendre plus lisibles. Un scénario représente une séquence spécifique d'enchaînements qui s'exécute du début à la fin du cas d'utilisation.

Définitions :

- ❖ Scénario nominal : également appelé scénario standard ou scénario normal, décrit le déroulement attendu et sans erreur d'une fonctionnalité ou d'une action dans un système.
- ❖ Scénario alternatif : Un scénario alternatif représente une séquence d'actions ou de comportements différents qui peuvent se produire en fonction de certaines conditions ou décisions prises pendant l'exécution d'un cas d'utilisation.
- ❖ Scénario d'exception : Un scénario d'exception représente une situation anormale ou inattendue qui peut se produire pendant l'exécution d'un cas d'utilisation.

Pour notre projet, chaque fiche de cas d'utilisation suivra la structure suivante :

- ✓ Un sommaire d'identification comprenant le titre, le résumé, les dates de création et de mise à jour (s'il y'en a eu), ainsi que la version et les acteurs responsables.
- ✓ La description des enchaînements qui détaillera le scénario nominal, les enchaînements d'exception et les préconditions.

Description textuelle du cas d'utilisation « Ajouter un utilisateur »

Tableau 13 : Description textuelle du cas d'utilisation "Ajouter un utilisateur"

SOMMAIRE D'IDENTIFICATION	
Titre : Ajouter un utilisateur	
Acteurs : Administrateur, Directeur administratif et financier	
Résumé : Ce cas d'utilisation permet à l'administrateur de créer le compte d'un Directeur administratif et financier, et au directeur administratif et financier de créer les comptes des directeurs régionaux, inspecteurs, directeurs d'école, et comptables d'école.	
Date de création : 05/06/2023	
Version : 1.0	
Responsable : KPEKPASSI Ibtihadj	
DESCRIPTION DES ENCHAINEMENTS	
Préconditions	<ul style="list-style-type: none">✓ L'acteur doit avoir une connexion internet fonctionnelle.✓ La plateforme doit être accessible et opérationnelle.
Scénario nominal	<ul style="list-style-type: none">✓ L'acteur demande l'accès au formulaire d'ajout d'un utilisateur.✓ Le système affiche le formulaire d'ajout d'utilisateur.✓ L'acteur saisit les informations requises dans le formulaire et le valide (E1)✓ Le système effectue des vérifications pour s'assurer que tous les champs obligatoires sont remplis (A1).✓ Le système vérifie l'adresse email de l'utilisateur (E2)✓ Le système vérifie si la fonction accordée à l'utilisateur n'est pas déjà occupée (E3).✓ Le système affiche un message de confirmation indiquant que l'utilisateur a été ajouté avec succès.

- ✓ L'acteur est redirigé vers la liste des utilisateurs qu'il gère

Scénario alternatif

(A1) : Les champs obligatoires ne sont pas remplis

- ✓ Le système détecte que certains champs obligatoires ne sont pas remplis.
- ✓ Le système affiche un message d'erreur indiquant les champs manquants
- ✓ L'utilisateur est invité à corriger les champs manquants en remplaçant les informations requises.
- ✓ Une fois que les champs obligatoires sont correctement remplis, l'utilisateur peut reprendre le processus d'ajout d'un utilisateur.

Scénario d'exception

(E1) : L'acteur annule l'opération

- ✓ Le processus d'ajout de l'utilisateur s'arrête.
- ✓ L'envoi des informations échoue.

(E2) : L'email de l'utilisateur est incorrecte

- ✓ Le système détecte que l'adresse email saisie par l'utilisateur n'est pas valide.
- ✓ Le système affiche un message d'erreur indiquant que l'adresse email n'est pas valide.
- ✓ Le processus d'ajout de l'utilisateur s'arrête.
- ✓ L'envoi des informations échoue.

(E3) : La fonction accordée à l'utilisateur est occupée

- ✓ Le système détecte que la fonction accordée à l'utilisateur est déjà occupée par un autre utilisateur.
- ✓ Le système affiche un message d'erreur indiquant que la fonction est occupée.
- ✓ Le processus d'ajout de l'utilisateur s'arrête.
- ✓ L'envoi des informations échoue.

Post conditions

- ✓ Les informations de l'utilisateur sont enregistrées avec succès dans la base de données.
- ✓ Le nouvel utilisateur est autorisé à s'authentifier dans le système.
- ✓ Le nouvel utilisateur peut effectuer différentes actions dans le système selon ses permissions.

Description textuelle du cas d'utilisation « **S'authentifier** »

Tableau 14 : Description textuelle du cas d'utilisation « S'authentifier »

SOMMAIRE D'IDENTIFICATION	
Titre : S'authentifier	
Acteurs : Administrateur, Directeur administratif et financier, Directeur régional, Inspecteur, Directeur d'école, Comptable d'école	
Résumé : Ce cas d'utilisation permet aux différents acteurs du système de se connecter en toute sécurité en utilisant leurs identifiants d'authentification.	
Date de création : 06/06/2023	
Version : 1.0	
Responsable : KPEKPASSI Ibtihadj	
DESCRIPTION DES ENCHAINEMENTS	
Préconditions	
<ul style="list-style-type: none">✓ L'acteur doit avoir une connexion internet fonctionnelle.✓ La plateforme doit être accessible et opérationnelle.	
Scénario nominal	
<ul style="list-style-type: none">✓ L'utilisateur saisit les paramètres de connexion (E1).✓ Le système effectue des vérifications pour s'assurer que tous les champs obligatoires sont remplis (A1).✓ Le système effectue une vérification pour s'assurer de la validité des paramètres saisis (A2)	

- ✓ Le système redirige l'utilisateur connecté vers le tableau de bord correspondant à son rôle dans le système.

Scénario alternatif

(A1) : Les champs obligatoires ne sont pas remplis

- ✓ Le système détecte que certains champs obligatoires ne sont pas remplis.
- ✓ Le système affiche un message d'erreur indiquant les champs manquants
- ✓ L'utilisateur est invité à corriger les champs manquants en remplaçant les informations requises.
- ✓ Une fois que les champs obligatoires sont correctement remplis, l'utilisateur peut reprendre le processus d'authentification.

(A2) : Les paramètres de connexion ne sont pas valides

- ✓ Le système détecte que les paramètres de connexion fournis ne sont pas valides.
- ✓ Le système affiche un message d'erreur indiquant une authentification invalide.
- ✓ L'utilisateur est invité à vérifier ses paramètres de connexion et à les saisir à nouveau.

Scénario d'exception

(E1) : Le formulaire n'est pas envoyé

- ✓ Le processus d'authentification s'arrête.

Post conditions

- ✓ La date et l'heure de connexion sont enregistrées dans la base de données
- ✓ Le nouvel utilisateur peut effectuer différentes actions dans le système selon ses permissions.

Description textuelle du cas d'utilisation « Initier la planification budgétaire»

Tableau 15 : Description textuelle du cas d'utilisation « Initier la planification budgétaire»

SOMMAIRE D'IDENTIFICATION
Titre : Initier la planification budgétaire
Acteurs : Administrateur
Résumé : Ce cas d'utilisation permet à l'administrateur d'initier la planification budgétaire pour une année scolaire.
Date de création : 06/06/2023
Version : 1.0
Responsable : KPEKPASSI Ibtihadj
DESCRIPTION DES ENCHAINEMENTS
Préconditions
<ul style="list-style-type: none">✓ L'administrateur doit avoir une connexion internet fonctionnelle.✓ La plateforme doit être accessible et opérationnelle.
Scénario nominal
<ul style="list-style-type: none">✓ L'administrateur demande l'accès aux informations de l'année en cours.✓ Le système affiche les informations de l'année en cours.✓ L'administrateur demande au système d'initier la planification budgétaire en cliquant sur le bouton correspondant à cette action.✓ Le système enregistre la date d'initiation dans la base de données.✓ Le système génère les informations concernant les taux d'ouvertures budgétaires pour chaque direction régional.✓ Le système autorise les directeurs d'école à renseigner les informations sur les financements obtenus.✓ Le système autorise les directeurs d'école à créer de nouvelles lignes budgétaires.

- ✓ Le système reproduit les informations concernant les financements reçus lors de l'année précédente dans la nouvelle année. (A1)
- ✓ Le système reproduit les informations concernant les imputations budgétaires de l'année précédente dans la nouvelle année. (A1)
- ✓ Le système affiche un message de succès indiquant que la planification budgétaire pour l'année scolaire a été initiée avec succès.

Scénario alternatif

(A1) : Les informations de l'année précédente n'existent pas dans le système

- ✓ Le déroulement continu au point suivant du scénario nominal

Scénario d'exception :

-- Néant --

Post conditions

- ✓ Le bouton permettant d'initier la planification budgétaire pour l'année scolaire est désactivé.
- ✓ Les directeurs d'école sont autorisés à renseigner les informations sur les financements obtenus.
- ✓ Les directeurs d'école peuvent ajouter de nouvelles lignes budgétaires.

Description textuelle du cas « Créeer une révision budgétaire»

Tableau 16 : Description textuelle du cas « Créeer une révision budgétaire»

SOMMAIRE D'IDENTIFICATION	
Titre : Créeer une révision budgétaire	
Acteurs : Directeur régional (Enseignement général), Directeur administratif et financier (Enseignement technique)	
Résumé : Ce cas d'utilisation permet aux acteurs concernés d'initier une révision budgétaire pour les écoles affiliées.	
Date de création : 06/06/2023	
Version : 1.0	
Responsable : KPEKPASSI Ibtihadj	
DESCRIPTION DES ENCHAINEMENTS	
Préconditions	<ul style="list-style-type: none">✓ L'acteur doit avoir une connexion internet fonctionnelle.✓ La plateforme doit être accessible et opérationnelle.
Scénario nominal	<ul style="list-style-type: none">✓ L'acteur demande l'accès au formulaire de création de révision budgétaire.✓ Le système affiche le formulaire de création de révision budgétaire.✓ L'acteur saisit les informations requises dans le formulaire et le valide (E1)✓ Le système effectue des vérifications pour s'assurer que tous les champs obligatoires sont remplis (A1).✓ Le système vérifie les dates afin d'éviter toute coïncidence avec les révisions budgétaires précédemment créées pour les écoles affiliées à la direction régionale (E2).✓ Le système affiche un message de confirmation indiquant que la révision budgétaire a été créée avec succès
Scénario alternatif	

(A1) : Les champs obligatoires ne sont pas remplis

- ✓ Le système détecte que certains champs obligatoires ne sont pas remplis.
- ✓ Le système affiche un message d'erreur indiquant les champs manquants
- ✓ L'utilisateur est invité à corriger les champs manquants en remplaçant les informations requises.
- ✓ Une fois que les champs obligatoires sont correctement remplis, l'utilisateur peut reprendre le processus de création de la révision budgétaire.

Scénario d'exception

(E1) : L'acteur annule l'opération

- ✓ Le processus s'arrête et l'ajout de la commande échoue

(E2) : Dates en conflit avec une révision budgétaire existante

- ✓ Le système détecte que les dates renseignées sont en conflit avec celles d'une précédente révision.
- ✓ Le système affiche un message d'erreur indiquant les champs manquants
- ✓ Le processus s'arrête et la création de la révision budgétaire s'annule.

Post condition

- ✓ Les écoles affiliées à la direction régionale ont la possibilité de modifier les informations sur les lignes budgétaires de l'année en cours.
- ✓ Les écoles affiliées à la direction régionale ont la possibilité d'enregistrer de nouveaux financements obtenus pour l'année en cours.

Description textuelle du cas « **Enregistrer une dépense**»

Tableau 17 : Description textuelle du cas « Enregistrer une dépense»

SOMMAIRE D'IDENTIFICATION	
Titre : Enregistrer une dépense	
Acteurs : Comptable d'école	
Résumé : Ce cas d'utilisation permet aux comptables d'école d'enregistrer une dépense dans le système	
Date de création : 06/06/2023	
Version : 1.0	
Responsable : KPEKPASSI Ibtihadj	
DESCRIPTION DES ENCHAINEMENTS	
Préconditions	<ul style="list-style-type: none">✓ Le directeur régional doit avoir une connexion internet fonctionnelle.✓ La plateforme doit être accessible et opérationnelle.
Scénario nominal	<ul style="list-style-type: none">✓ Le comptable demande la page affichant la liste des imputations budgétaires de l'année en cours✓ Le système affiche la page correspondante✓ Le comptable identifie la ligne budgétaire sur laquelle il souhaite enregistrer sa dépense.✓ Le comptable clique sur le bouton permettant d'afficher le formulaire de saisie des informations sur la dépense.✓ Le comptable saisit les informations requises dans le formulaire et le valide (E1)✓ Le système effectue des vérifications pour s'assurer que tous les champs obligatoires sont remplis (A1).

- ✓ Le système effectue une vérification pour s'assurer que la dépense à enregistrer ne dépasse pas la limite d'ouverture budgétaire fixée par le directeur régional (E2).
- ✓ Le système enregistre la dépense avec succès.
- ✓ Le système affiche un message de confirmation indiquant que la dépense a été enregistrée avec succès.

Scénario alternatif

(A1) : Les champs obligatoires ne sont pas remplis

- ✓ Le système détecte que certains champs obligatoires ne sont pas remplis.
- ✓ Le système affiche un message d'erreur indiquant les champs manquants.
- ✓ L'utilisateur est invité à corriger les champs manquants en remplaçant les informations requises.
- ✓ Une fois que les champs obligatoires sont correctement remplis, l'utilisateur peut reprendre le processus de création de la dépense.

Scénario d'exception

(E1) : L'acteur annule l'opération

- ✓ Le processus s'arrête et l'ajout de la commande échoue

(E2) : Échec de l'enregistrement de la dépense

- ✓ Le système détecte que la dépense ne peut pas être effectuée.
- ✓ Le système affiche un message d'erreur indiquant l'échec de l'enregistrement de la dépense.
- ✓ Le scénario nominal est interrompu.

2.3.2.3 Diagrammes d'activité

En UML, le diagramme d'activité est utilisé pour représenter les événements déclenchés en fonction des états du système et modéliser les comportements. Il s'agit d'un organigramme qui décrit la séquence d'étapes dans le déroulement d'un cas d'utilisation ou d'un ensemble de cas d'utilisation. Dans le contexte de notre projet, voici quelques exemples de diagrammes d'activité associés.

Diagramme d'activité du cas d'utilisation « Ajouter un utilisateur »

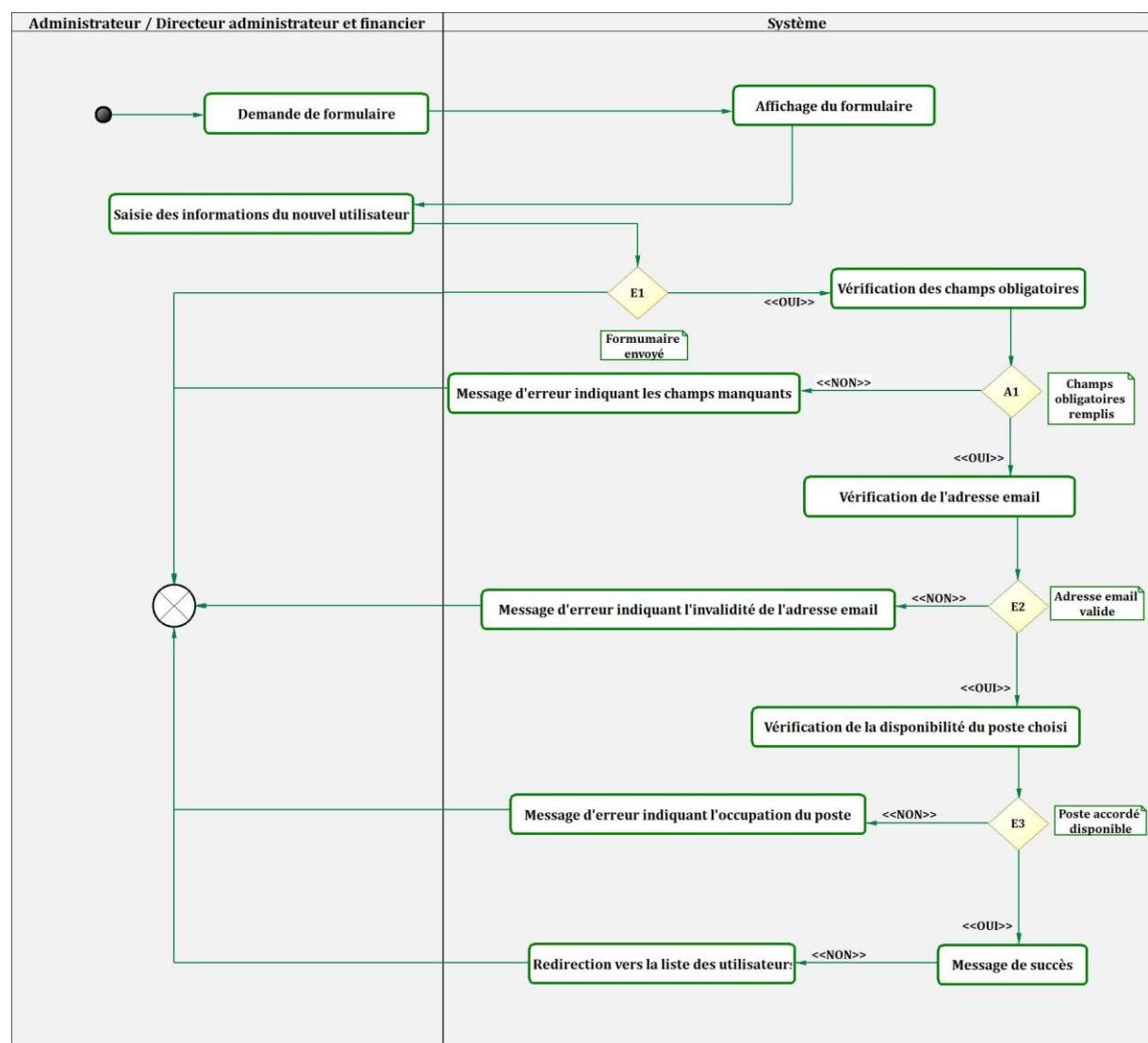


Figure 16 : Diagramme d'activité du cas d'utilisation « Ajouter un utilisateur »

Diagramme d'activité du cas d'utilisation « S'authentifier »

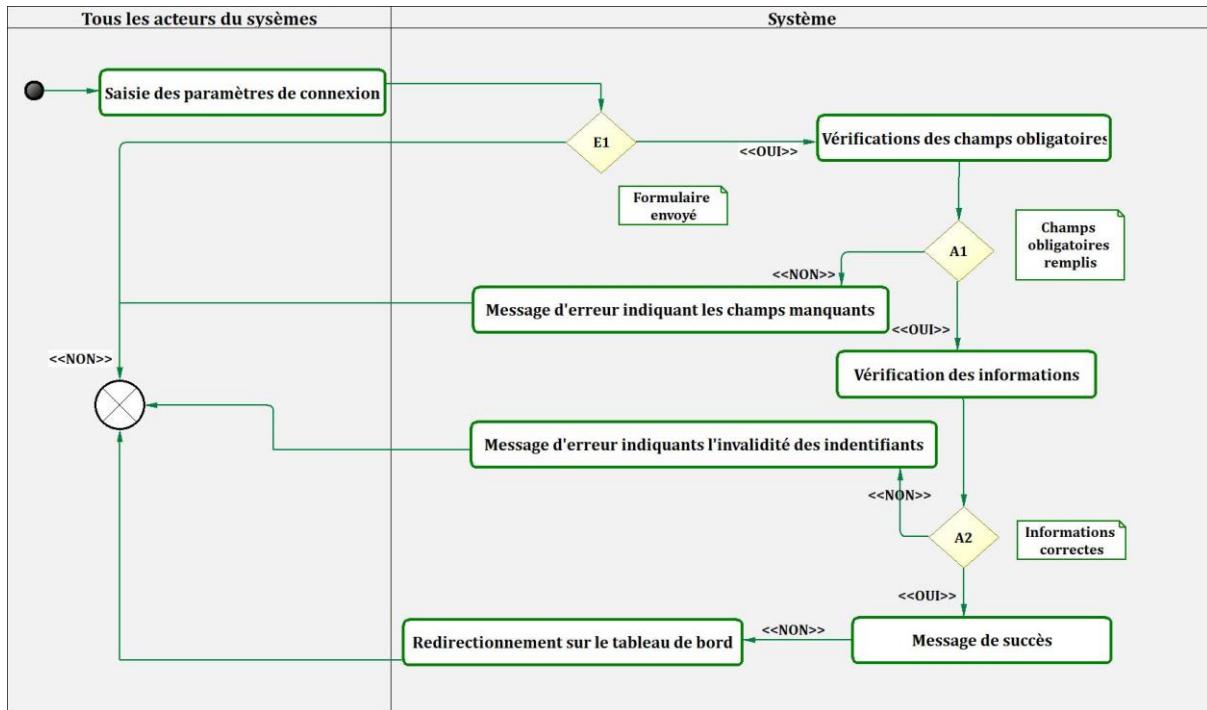


Figure 17 : Diagramme d'activité du cas d'utilisation « S'authentifier »

Diagramme d'activité du cas d'utilisation « **Initier la planification budgétaire** »

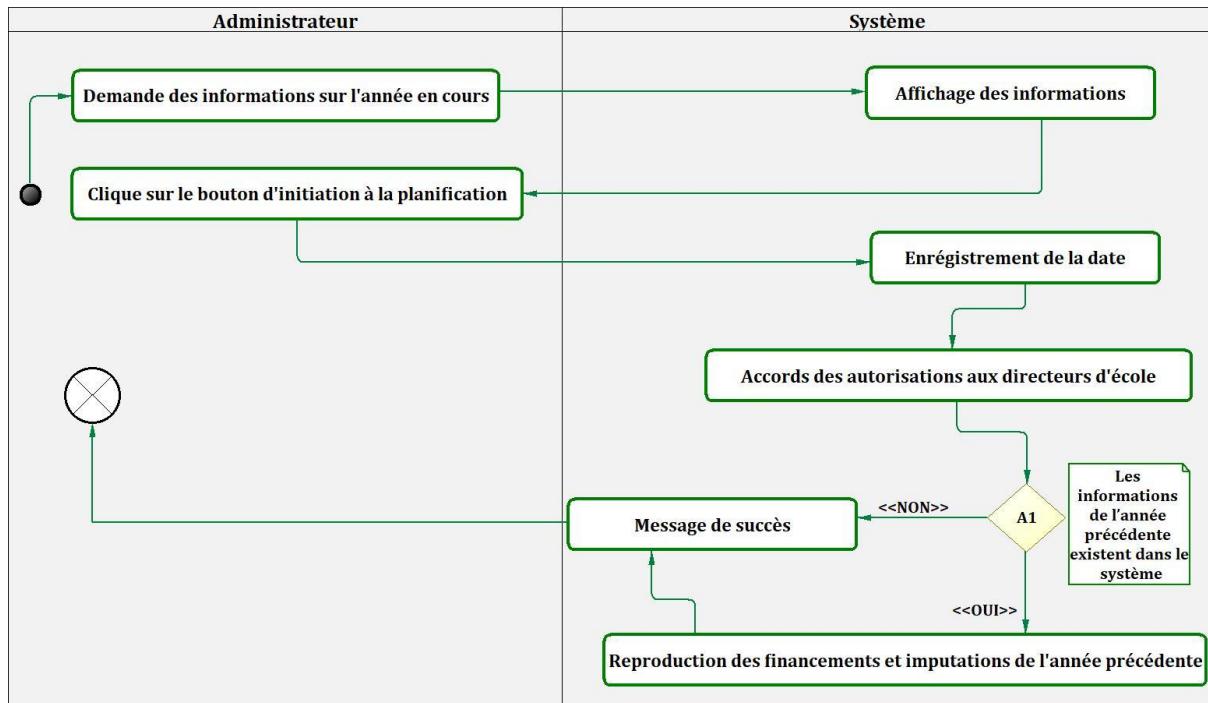


Figure 18 : Diagramme d'activité du cas d'utilisation « **Initier la planification budgétaire** »

Diagramme d'activité du cas d'utilisation « **Créer une révision budgétaire** »

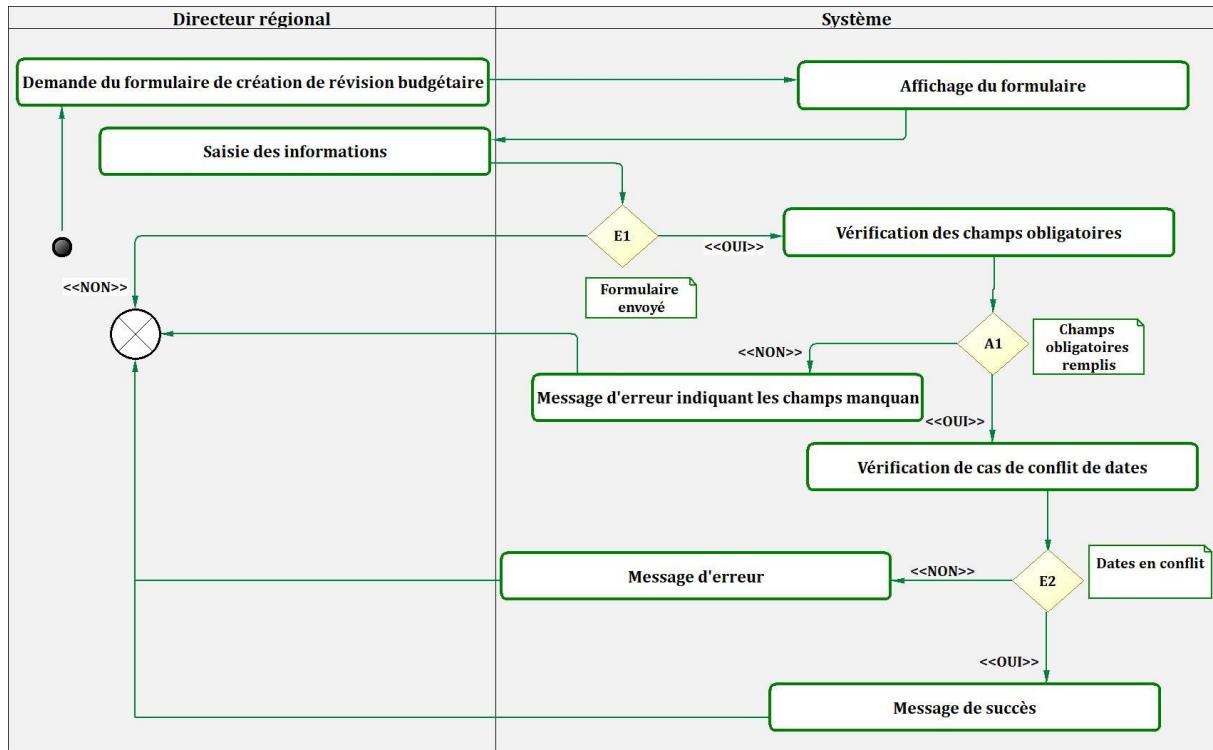


Figure 19 : Diagramme d'activité du cas d'utilisation « *Créer une révision budgétaire* »

Diagramme d'activité du cas d'utilisation « Créeer une dépense »

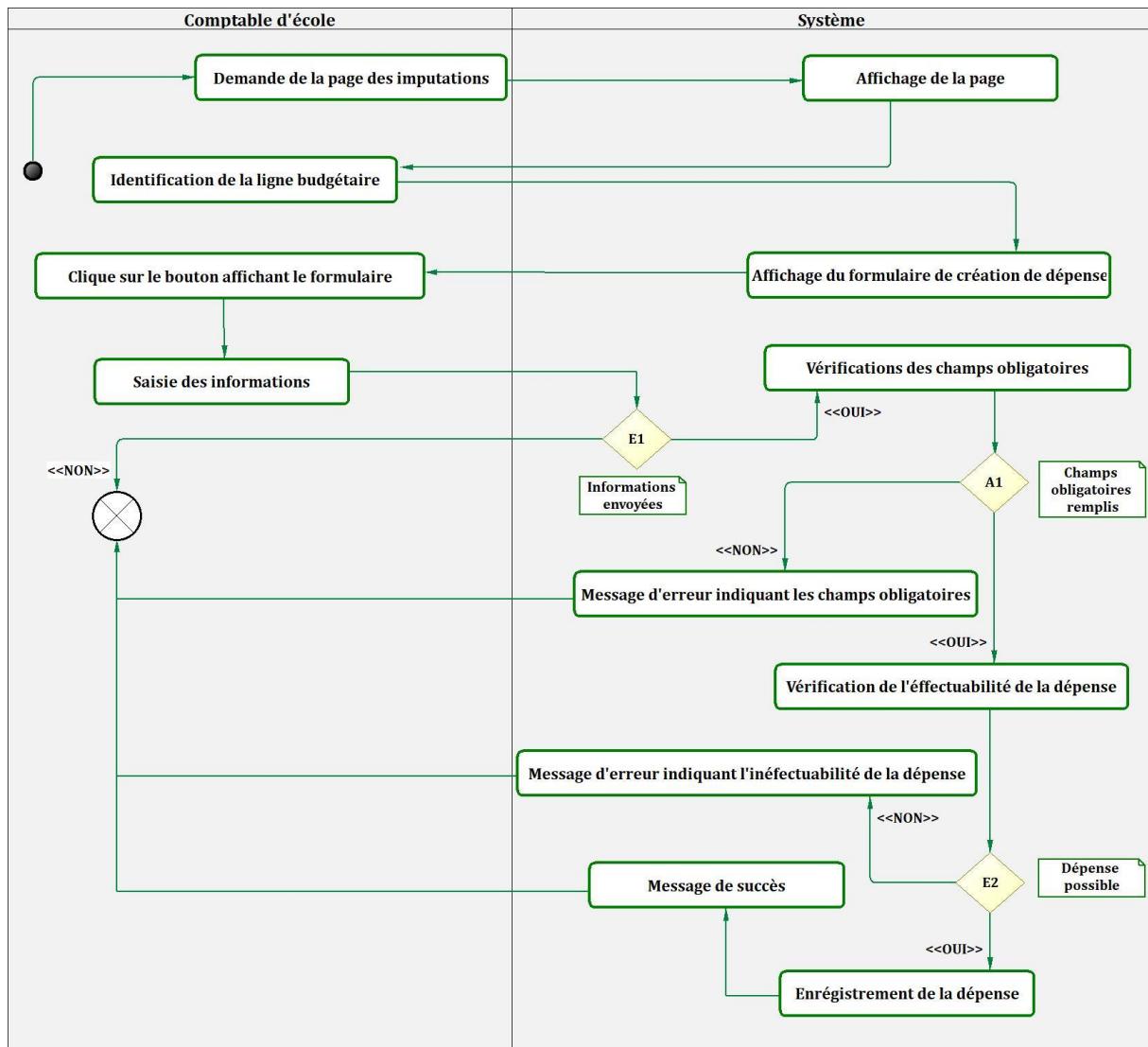


Figure 20 : Diagramme d'activité du cas d'utilisation « Créeer une dépense »

2.3.2.4 Diagrammes de séquence

Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »

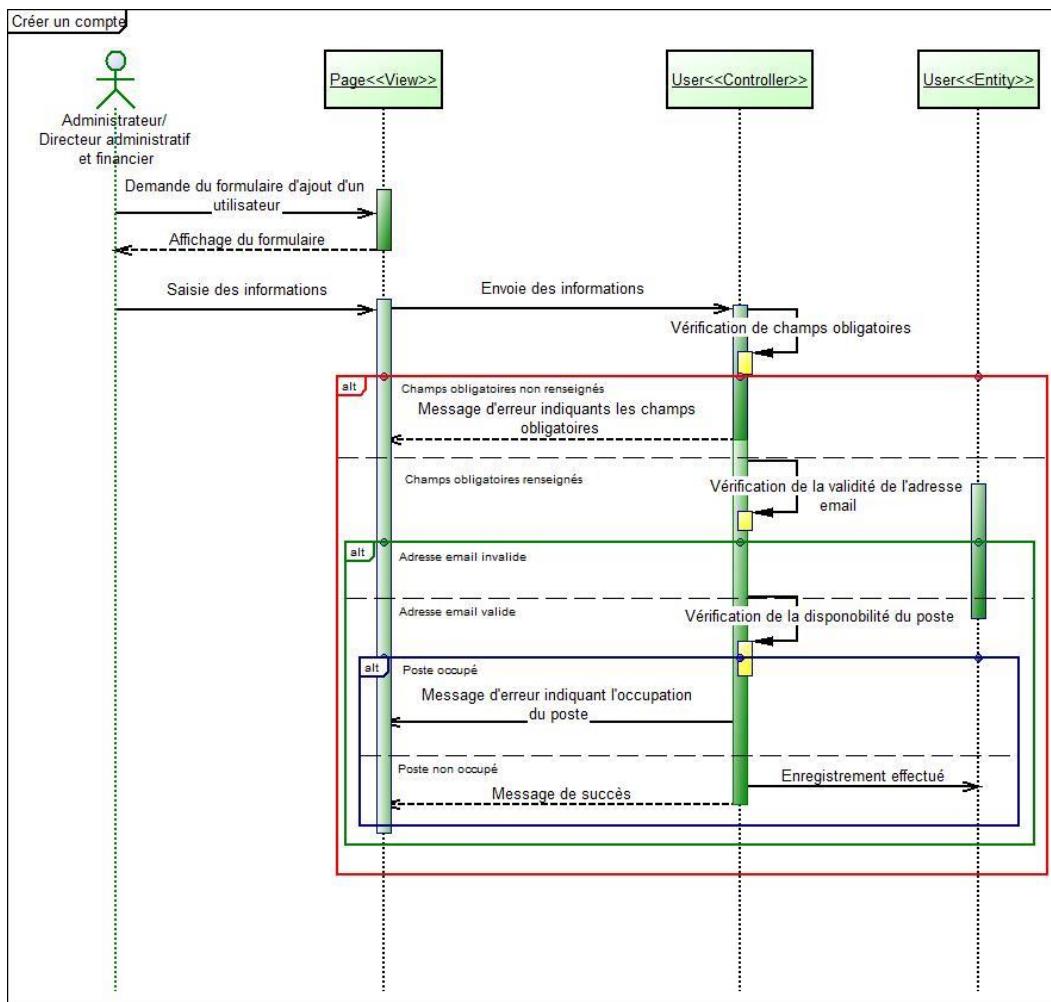


Figure 21 : Diagramme de séquence du cas d'utilisation « Ajouter un utilisateur »

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

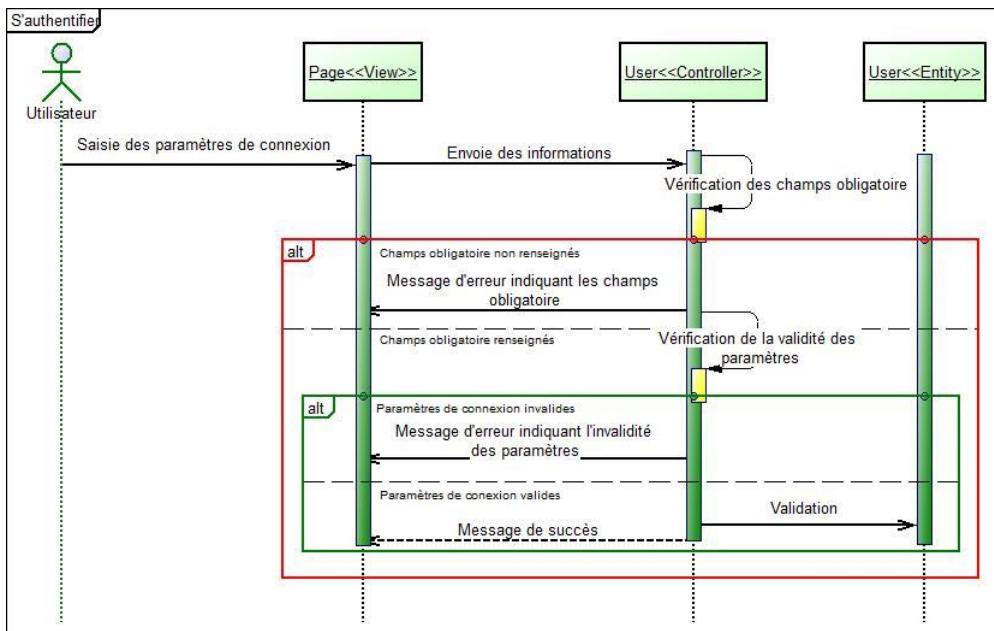


Figure 22 : Diagramme de séquence du cas d'utilisation « S'authentifier »

Diagramme de séquence du cas d'utilisation « **Initier la planification budgétaire** »

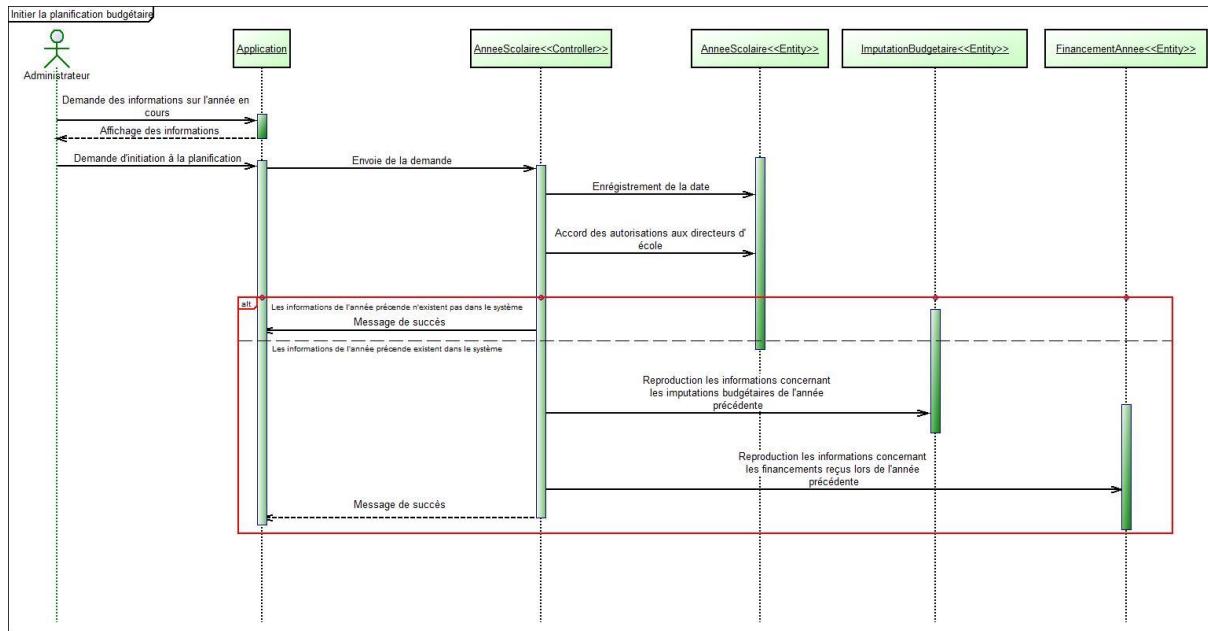


Figure 23 : Diagramme de séquence du cas d'utilisation « *Initier la planification budgétaire* »

Diagramme de séquence du cas d'utilisation « **Créer une révision budgétaire** »

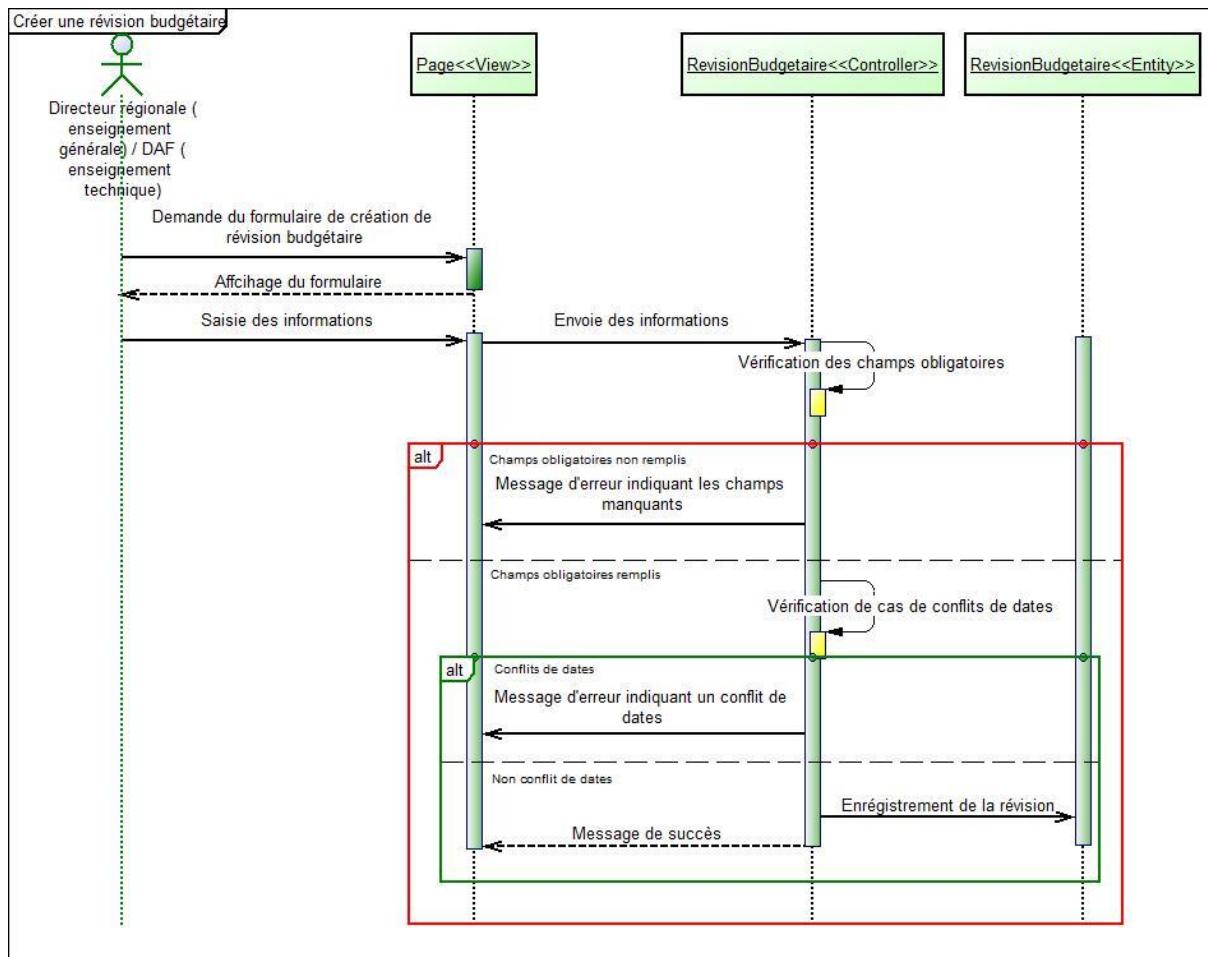


Figure 24 : Diagramme de séquence du cas d'utilisation « *Créer une révision budgétaire* »

Diagramme de séquence du cas d'utilisation « **Créer une dépense** »

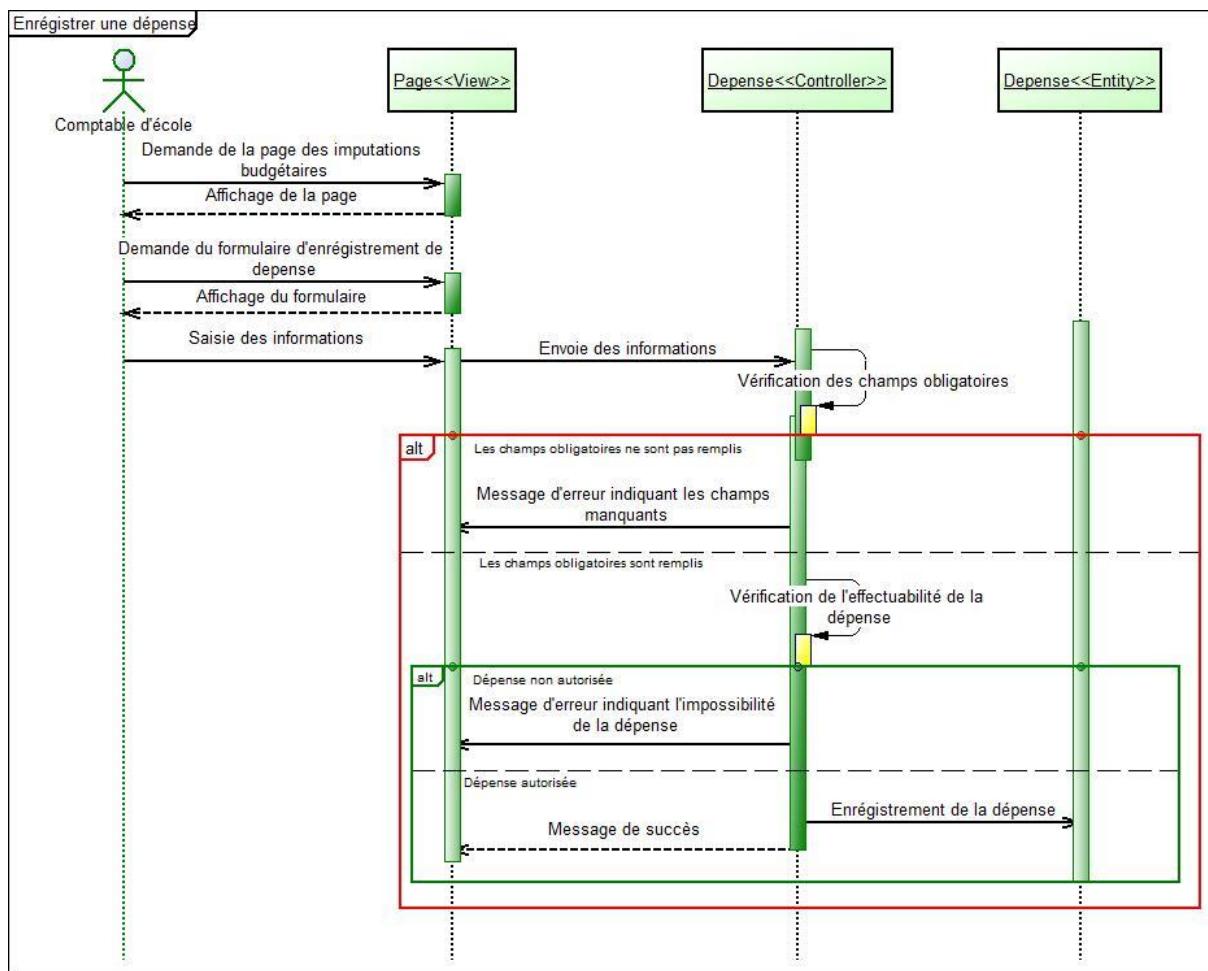


Figure 25 : Diagramme de séquence du cas d'utilisation « *Créer une dépense* »

2.3.3 Analyse

2.3.3.1 Diagrammes de classes

Le diagramme de classes, pierre angulaire de l'ingénierie logicielle, offre une représentation visuelle des classes et interfaces d'un système, ainsi que de leurs relations intrinsèques. Cette modélisation orientée objet revêt une importance primordiale et demeure incontournable dans ce contexte. Contrairement au diagramme de cas d'utilisation, qui scrute un système du prisme des acteurs, le diagramme de classes s'attarde sur sa structure interne.

S'inscrivant dans une perspective statique, il se dispense de considérations temporelles. Chaque classe énonce les responsabilités, le comportement et la nature d'un ensemble d'objets, dont les instances sont les membres de ladite classe. En dévoilant la structure des objets manipulés, le diagramme de classes façonne l'architecture de notre système.

Une classe, incarnation abstraite d'entités douées de caractéristiques communes, préside à cette modélisation. Ce schéma illustre les entités en jeu ainsi que les connexions qui les tissent. En ce qui concerne notre projet, le diagramme de classes se dessine de la manière suivante :

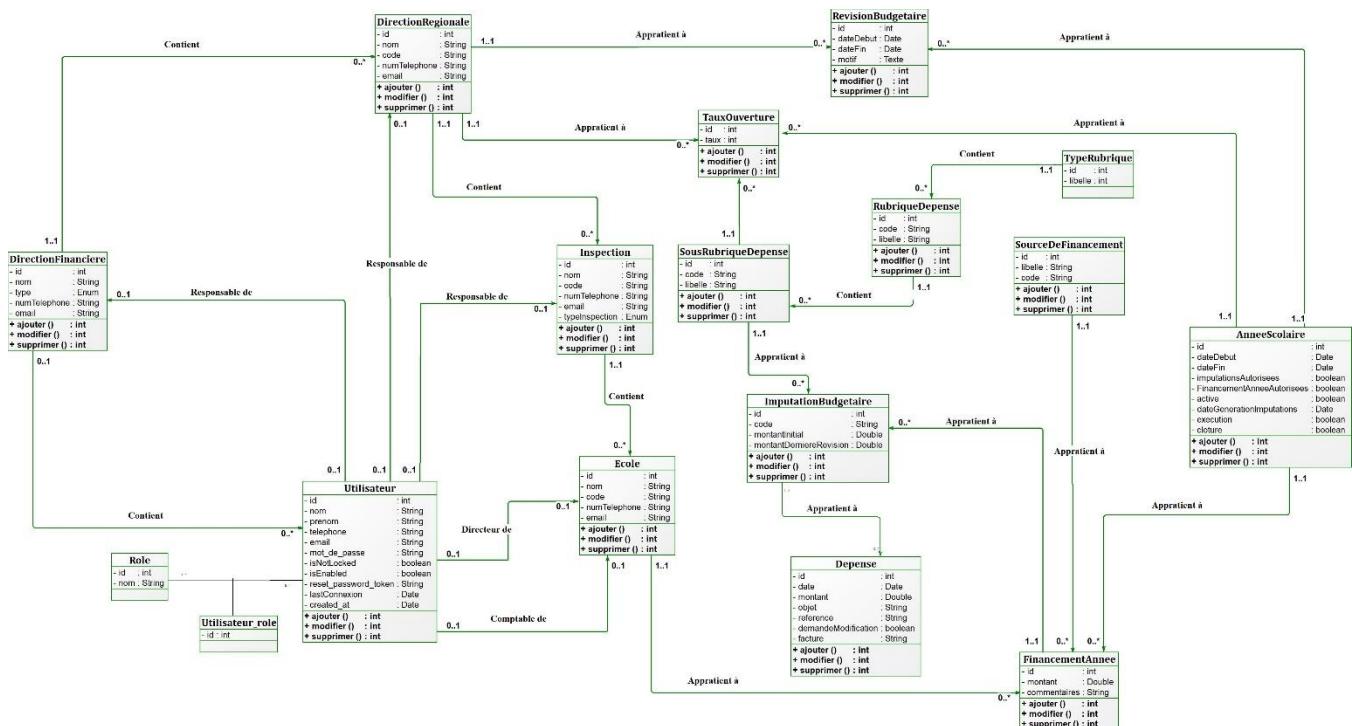


Figure 26 : Diagramme de classe

2.3.4 Conception

2.3.4.1 Diagramme de déploiement

Le diagramme de déploiement a pour rôle de dépeindre l'interaction du système avec l'infrastructure physique, et comment les composants du système s'y répartissent tout en tissant leurs liens respectifs. Aligné sur l'architecture matérielle adoptée dans le cas de notre projet, voici le schéma de déploiement associé :

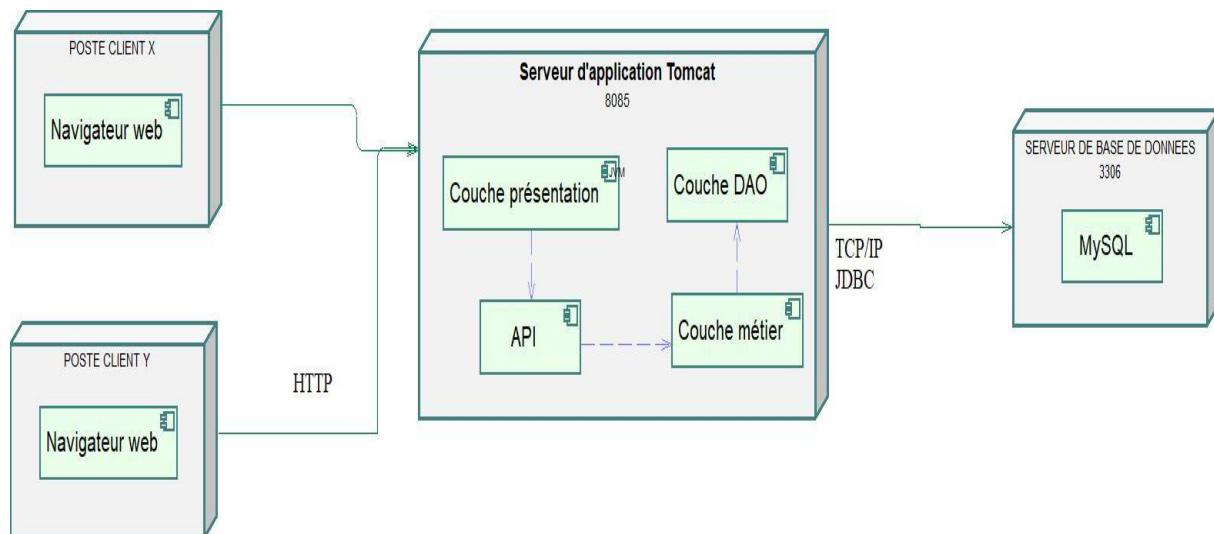


Figure 27: Diagramme de déploiement

PARTIE 3 : REALISATION ET MISE EN ŒUVRE

La troisième partie de ce document se concentre sur l'implémentation du système en vue de sa mise en service. Après avoir déterminé les données et les traitements nécessaires dans le cahier des charges et le dossier d'analyse et conception, cette étape nous permettra de concrétiser le projet. Cette partie est divisée en quatre sous-sections : la mise en œuvre du système, la présentation de l'application, la sécurité de l'application et la configuration de la base de données.

3.1 MATERIEL ET LOGICIELS UTILISES

3.1.1 Matériel utilisé

Le matériel utilisé pour notre stage de développement est un ordinateur portable qui présente les caractéristiques suivantes :

Tableau 18 : Matériel utilisé

PROPRIETE	CARACTERISTIQUES
Marque et modèle	DELL Latitude E5570
Processeur et Fréquence	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
Système d'exploitation	Windows
Type de système	Système d'exploitation 64 bits, processeur x64
Mémoire RAM	8 Go
Disque dur	256 Go

3.1.2 Logiciels utilisés

Nous avons utilisé différents outils logiciels dans le cadre de notre projet, que nous pouvons classer en deux catégories : les outils d'administration de la base de données et les outils de programmation et de développement.

3.1.2.1 Outil d'administration de base de données

Un SGBD (Système de Gestion de Base de Données) est un outil logiciel essentiel pour implémenter et administrer une base de données. Il permet de stocker, lire, écrire, modifier et supprimer les données de manière efficace. Avec un SGBD, il devient plus facile

d'interagir avec la base de données, d'effectuer des opérations telles que l'insertion, la mise à jour, la suppression et la récupération des données. C'est un composant clé pour assurer la fiabilité, la sécurité et les performances d'une base de données.

✓ **PhpMyAdmin 5.2.0**



Figure 28 : Logo de PhpMyAdmin

PhpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL et Maria DB, réalisée principalement en PHP et distribuée sous licence GNU GPL.

3.1.2.2 Outils de programmation et de développement

❖ **Spring boot 3.0.6**



Figure 29 : Logo de Spring boot

Spring Boot est un framework open-source basé sur Spring Framework qui simplifie et accélère le développement d'applications Java. Il offre une approche convention-over-configuration, ce qui signifie qu'il propose des configurations par défaut pour de nombreuses tâches courantes, permettant ainsi aux développeurs de se concentrer sur le code métier plutôt que sur la configuration. Spring Boot facilite également le

déploiement et la gestion des applications, en intégrant des serveurs d'applications embarqués et en offrant des outils pour la surveillance et la gestion des dépendances.

Pour notre projet, nous avons choisi d'utiliser le framework Spring Boot pour la mise en place de notre API.

❖ **Angular 15**



Figure 30 : Logo d'Angular

Angular est un framework de développement d'applications web open-source basé sur TypeScript. Il est largement utilisé pour créer des applications web riches et interactives. Angular permet de développer des applications côté client en utilisant des composants réutilisables. Il offre une architecture modulaire et utilise le principe de la programmation orientée composant (POC) pour structurer et organiser le code. Angular propose également des fonctionnalités avancées telles que la gestion des états, la liaison de données bidirectionnelle, le routage, la gestion des formulaires et la communication avec les API backend. Avec Angular, les développeurs peuvent créer des applications web robustes, évolutives et performantes.

Nous avons utilisé le framework Angular pour développer notre interface web tout en assurant une communication sécurisée avec notre API développée en Spring Boot.

❖ **Bootstrap 4**



Figure 31 : Logo de Bootstrap

Bootstrap est un framework front-end populaire pour le développement web. Il fournit une collection de classes CSS et de composants préconçus qui permettent de créer rapidement et facilement des interfaces utilisateur attrayantes et réactives. Avec Bootstrap, vous pouvez créer des mises en page adaptatives, des grilles de colonnes, des formulaires stylisés, des boutons, des modales, des barres de navigation, des carrousels, des onglets et bien plus encore, sans avoir à écrire beaucoup de code CSS personnalisé. Il est également compatible avec la plupart des navigateurs modernes et s'adapte automatiquement aux différentes tailles d'écran, ce qui le rend idéal pour le développement d'applications web mobiles et réactives. En utilisant Bootstrap, vous gagnez du temps et de l'effort dans la création d'interfaces utilisateur professionnelles et attrayantes.

Dans notre projet, l'utilisation de Bootstrap s'est avérée très bénéfique. Les composants prêts à l'emploi ainsi que les classes CSS pratiques nous ont permis de développer rapidement une interface utilisateur attrayante et réactive.

❖ **PrimeNG**

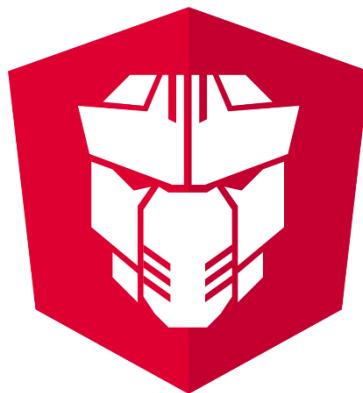


Figure 32 : Logo de PrimeNg

PrimeNG est une bibliothèque de composants UI pour Angular. Elle fournit une collection complète de composants prêts à l'emploi, tels que des boutons, des tableaux, des formulaires, des menus, des graphiques, des calendriers, etc. PrimeNG facilite le développement d'interfaces utilisateur attrayantes et fonctionnelles dans les applications Angular. Elle offre une grande variété de fonctionnalités, des options de personnalisation et une compatibilité avec les navigateurs modernes. Grâce à PrimeNG, les développeurs peuvent gagner du temps en utilisant des composants prédefinis et se concentrer sur la logique métier de leur application.

Dans notre projet, l'utilisation des composants prêts à l'emploi de la bibliothèque PrimeNG s'est avérée très utile, car elle nous a permis de développer rapidement une interface web moderne et interactive avec des fonctionnalités avancées, tout en bénéficiant de la personnalisation facile grâce à des thèmes personnalisables. Cela a considérablement accéléré le développement et amélioré l'expérience utilisateur de notre application.

❖ **Git**



Figure 33 : Logo de Git

Git est un système de contrôle de version décentralisé largement utilisé dans le développement de logiciels. Il permet de suivre les modifications apportées aux fichiers source et de coordonner le travail collaboratif entre les membres d'une équipe de développement. Git enregistre chaque modification de fichier dans un référentiel, ce qui permet de conserver un historique complet des changements. Il offre des fonctionnalités telles que la création de branches pour travailler sur des fonctionnalités ou des correctifs indépendamment, la fusion des branches pour intégrer les modifications, le suivi des conflits de fusion, la gestion des versions et la collaboration entre les développeurs. Git est apprécié pour sa rapidité, sa flexibilité et sa capacité à gérer efficacement des projets de toutes tailles.

Dans notre projet, l'utilisation de GIT s'est avérée extrêmement utile. Ce système de contrôle de version nous a permis de gérer efficacement les modifications du code source, et de revenir à des versions antérieures du projet en cas de besoin. Grâce à GIT, nous avons pu garder une trace complète de l'historique des modifications et assurer la cohérence du code. Cela a grandement facilité notre processus de développement .

❖ **Github**

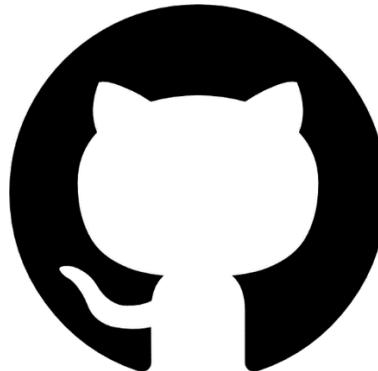


Figure 34 : Logo de Github

GitHub est une plateforme en ligne basée sur Git qui facilite la collaboration, le partage et la gestion de projets de développement. Les développeurs peuvent héberger leurs dépôts de code source, travailler en équipe et proposer des modifications via des demandes de fusion. C'est un outil populaire pour l'hébergement de projets open source et offre une interface conviviale ainsi que des fonctionnalités avancées de gestion de projet.

Dans notre projet, l'utilisation de GitHub s'est révélée extrêmement précieuse. Cette plateforme d'hébergement basée sur Git nous a permis de stocker notre code source de manière sécurisée.

❖ **MySQL**



Figure 35 : Logo de MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR) largement utilisé. Il est apprécié pour sa simplicité, sa performance et sa fiabilité. MySQL permet de stocker, gérer et récupérer efficacement des données structurées. Il est compatible avec de nombreux langages de programmation et offre des fonctionnalités avancées telles que

la gestion des transactions, les indexations optimisées et la réplication des données. MySQL est utilisé dans de nombreux environnements, des petites applications web aux grandes entreprises.

Dans notre projet, l'utilisation de MySQL s'est avérée extrêmement bénéfique. Ce système de gestion de base de données relationnelle nous a permis de stocker et de gérer efficacement les données de manière sécurisée. MySQL nous a offert une solution robuste pour organiser les informations de mon application et faciliter les opérations de recherche, de filtrage et de mise à jour des données.

❖ **IntelliJ IDEA**

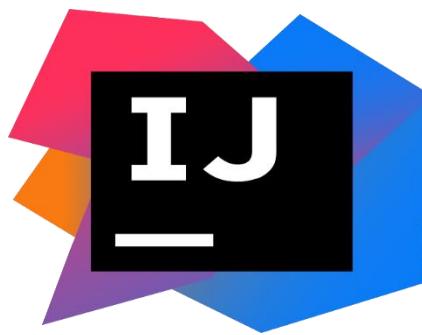


Figure 36 : Logo de IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré (IDE) populaire utilisé pour la création d'applications logicielles. Il est développé par JetBrains et offre des fonctionnalités avancées pour les développeurs. IntelliJ IDEA prend en charge plusieurs langages de programmation, notamment Java, Kotlin, Groovy, Scala et bien d'autres. Il fournit des fonctionnalités telles que la complétion automatique de code, le débogage, la refactorisation, la gestion de version, les tests unitaires et bien plus encore. L'IDE est réputé pour sa convivialité, sa performance et ses outils de productivité qui facilitent le développement d'applications de qualité.

Dans notre projet, l'utilisation d'IntelliJ IDEA s'est révélée extrêmement bénéfique. Ce puissant environnement de développement intégré (IDE) nous a permis de coder de manière plus efficace et de gérer notre projet de manière organisée. Grâce à ses fonctionnalités avancées telles que l'autocomplétion, le débogage, etc..., IntelliJ IDEA a considérablement amélioré notre productivité.

3.2 ARCHITECTURES MATERIELLE ET LOGICIELLE DE L'APPLICATION

3.2.1 Architecture matérielle

Dans notre projet, nous avons opté pour une architecture distribuée de type trois tiers. Cette approche architecturale joue un rôle essentiel dans la structuration et l'organisation de notre système, en définissant la manière dont les différents éléments (logiciels, matériels, humains et informations) interagissent entre eux.

L'architecture trois tiers, également connue sous les appellations architecture à trois niveaux ou à trois couches, repose sur le concept de client-serveur, où des modules indépendants sont mis en place pour gérer l'interface utilisateur (GUI), les processus logiques et fonctionnels, ainsi que l'accès aux données. Cette architecture se distingue par la présence de trois parties distinctes, chacune ayant un rôle spécifique :

- 1. La présentation des données** : Cette partie concerne l'affichage des données et l'interaction avec l'utilisateur. Elle est réalisée à travers un navigateur web qui permet d'afficher notre interface utilisateur conviviale et réactive. Cette interface offre une expérience utilisateur intuitive et attrayante, favorisant ainsi une interaction fluide avec le système.
- 2. Le traitement métier des données** : Cette partie met en œuvre les règles de gestion et la logique applicative de notre solution. Elle est gérée par un serveur applicatif, spécifiquement le serveur Tomcat, qui héberge notre backend développé en Spring Boot. Le serveur applicatif assure l'exécution des opérations métier, le traitement des requêtes et la gestion des processus fonctionnels de notre application.
- 3. L'accès aux données persistantes** : Cette partie est dédiée à la gestion des données qui nécessitent une conservation à long terme, voire permanente. L'accès à ces données se fait à travers un serveur de base de données MySQL. Ce serveur permet le stockage, la récupération et la manipulation des données de manière sécurisée et efficace, assurant ainsi la persistance des informations essentielles à notre application.

En adoptant cette architecture trois tiers, nous visons à optimiser la performance, la scalabilité et la maintenabilité de notre système. Chaque partie joue un rôle clé dans le fonctionnement global de l'application, permettant ainsi de séparer les préoccupations et de favoriser la modularité et la réutilisabilité du code. De plus, cette architecture facilite la gestion des mises à jour et des évolutions de notre solution, en offrant une structure bien organisée et adaptée à nos besoins.

L'architecture physique de notre projet est alors la suivante :

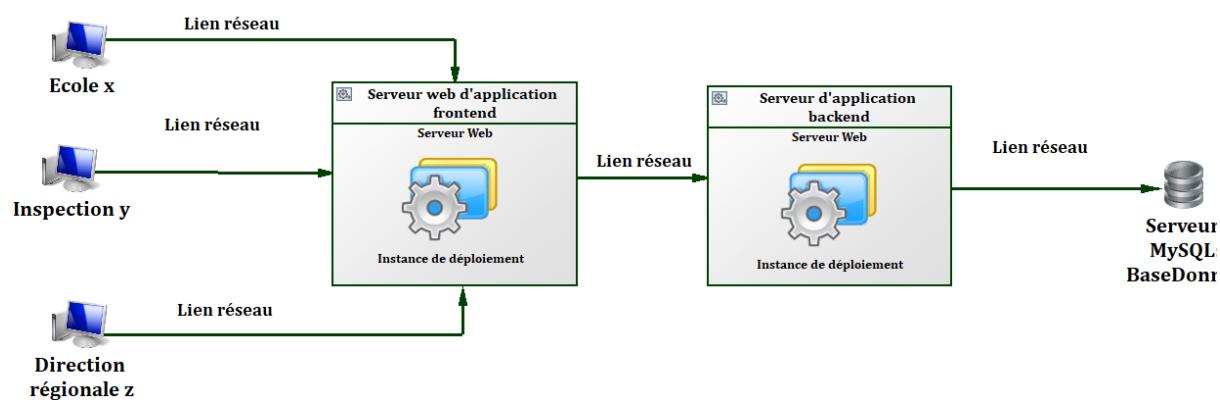


Figure 37 : Architecteure physique du projet

3.2.2 Architecture logicielle

L'architecture logicielle constitue une représentation symbolique et schématique des divers composants d'un ou plusieurs systèmes informatiques, ainsi que de leurs interrelations et interactions. Dans le cadre de notre projet, nous avons opté pour une architecture MVC.

- L'application web et le concept MVC

L'architecture Modèle/Vue/Contrôleur (MVC) est l'une des méthodologies permettant de structurer efficacement une interface utilisateur dans le développement de logiciels. En adoptant cette approche, nous avons divisé notre application en trois entités clés : le modèle, la vue et le contrôleur, chacun jouant un rôle essentiel dans l'interface globale.

Le modèle représente la couche de données de notre application. Il gère l'accès aux données ainsi que les opérations de mise à jour. Grâce à l'utilisation du framework Spring Boot, nous avons pu créer un modèle robuste et performant, permettant de manipuler les données de manière sécurisée et efficace.

La vue correspond à l'interface utilisateur de notre application. C'est à travers cette entité que les utilisateurs interagissent avec le système. En utilisant Angular, nous avons créé des vues attrayantes, réactives et conviviales. Les entrées de l'utilisateur sont capturées et les sorties sont affichées de manière claire et intuitive.

Le contrôleur agit comme un médiateur entre le modèle et la vue. Il coordonne les actions de l'utilisateur, interprète les requêtes entrantes et déclenche les actions appropriées sur le modèle. En utilisant le framework tel que Spring MVC (un autre framework issu de la famille spring framework), nous avons pu développer des contrôleurs efficaces et flexibles, garantissant une gestion fluide des interactions entre le modèle et la vue.

L'architecture MVC offre une approche structurée pour organiser notre application, facilitant ainsi la maintenance, l'évolutivité et la réutilisation du code. En découplant notre système en ces trois entités distinctes, nous avons pu améliorer la lisibilité du code, faciliter les tests et favoriser une future collaboration des développeurs sur notre code.

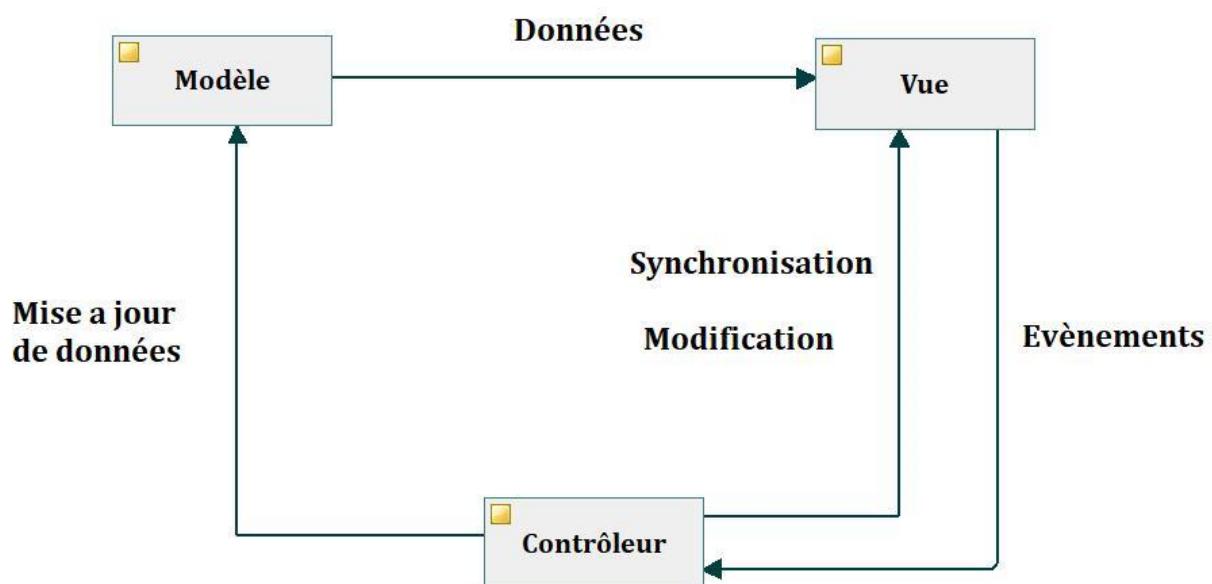


Figure 38 : Architecture MVC

3.3 SECURITE DE L'APPLICATION

- ❖ Notre plateforme sécurise les échanges de données en utilisant des jetons JWT (JSON Web Tokens) pour chaque requête, garantissant ainsi l'authentification et l'autorisation des utilisateurs. Cela permet un contrôle strict de l'accès à la plateforme, assurant ainsi la confidentialité et la sécurité des informations.
- ❖ Nous accordons une grande importance à la sécurité des données, notamment en ce qui concerne les mots de passe. Nous avons mis en place un mécanisme de cryptage des mots de passe pour garantir leur confidentialité. Ainsi, même en cas de violation de la sécurité, les mots de passe restent sécurisés.
- ❖ En ce qui concerne la disponibilité de la plateforme, nous nous engageons à assurer sa disponibilité 24h/7j. Cela signifie que la plateforme est opérationnelle en permanence, ce qui garantit la disponibilité des données pour les utilisateurs.

En résumé, notre plateforme met en place des mesures de sécurité avancées, telles que l'authentification, l'autorisation, le cryptage des mots de passe. De plus, nous nous engageons à maintenir la disponibilité de la plateforme pour répondre aux besoins de nos utilisateurs à tout moment.

3.4 MISE EN PLACE DE LA BASE DE DONNEES

Dans le cadre de notre projet, nous avons opté pour l'utilisation de l'ORM **Hibernate** de Spring Boot pour la création de notre base de données. Cette approche moderne diffère de la méthode traditionnelle d'écriture de scripts SQL, en offrant un mécanisme de gestion de la base de données via des classes Java appelées "entités".

Plutôt que de rédiger des requêtes SQL pour définir la structure de la base de données, nous exploitons des classes Java annotées avec des balises spécifiques de Spring, telles que **@Entity** et **@Table**, afin de définir nos tables et leurs attributs. Ces classes d'entités agissent comme des modèles de données et reflètent directement les tables au sein de la base de données. L'avantage central de cette approche est la flexibilité offerte pour migrer d'un SGBD (Système de Gestion de Base de Données) à un autre sans avoir à revoir et réécrire les requêtes SQL de création de tables.

Cette manière de procéder permet d'assurer la continuité du fonctionnement du système, tout en garantissant que le passage à un SGBD différent n'engendre pas de complications

majeures. Voici un échantillon de nos classes, directement liées aux tables, illustrant notre approche :

- ❖ Pour la table « rubriques_depenses »



```
@Setter
@Getter
@RequiredArgsConstructor
@AllArgsConstructor
@Builder
@Table(name = "rubriques_depenses")
@Entity
public class RubriqueDepense implements Serializable {

    /**
     * Classe créée le 07/05/2023
     * Dernière modification :
     */

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @Column(nullable = false)
    private String code;

    @Column(nullable = false)
    private String libelle;

    @ManyToOne
    @JoinColumn(name = "type_rubrique_id")
    private TypeRubrique typeRubrique;

}
```

Figure 39 : Structure de la table « rubriques_depenses »

❖ Pour la table « inspections»



```
@Setter
@Getter
@RequiredArgsConstructor
@AllArgsConstructor
@Builder
@Table(name = "inspections")
@Entity
public class Inspection implements Serializable {

    /**
     * Classe créée le 07/05/2023
     * Dernière modification : 14/07/2023
     */

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @Column(nullable = false)
    private String nom;

    @Column(nullable = false)
    private String code;

    @Column(nullable = true)
    private String numTelephone;

    @Column(nullable = true)
    private String email;

    @Column(nullable = false)
    @Enumerated(EnumType.STRING)
    private TypeInspection typeInspection;

    @ManyToOne
    @JoinColumn(name = "direction_regionale_id", nullable = false)
    private DirectionRegionale directionRegionale;

    @ManyToOne
    @JoinColumn(name = "user_inspecteur_id", nullable = true)
    private User inspecteur;
}
```

Figure 40 : Structure de la table « inspections»

Pour la table « revisions_budgetaires »

```
● ● ●

@Setter
@Getter
@RequiredArgsConstructor
@AllArgsConstructor
@Builder
@Table(name = "revisions_budgetaires")
@Entity
public class RevisionBudgetaire implements Serializable {

    /**
     * Classe créée le 07/05/2023
     * Dernière modification : 14/07/2023
     */

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @Temporal(TemporalType.TIMESTAMP)
    @JsonFormat(pattern = "yyyy-MM-dd HH:mm")
    @Column(nullable = false)
    private Date dateDebut;

    @Temporal(TemporalType.TIMESTAMP)
    @JsonFormat(pattern = "yyyy-MM-dd HH:mm")
    @Column(nullable = false)
    private Date dateFin;

    @Column(nullable = false)
    private String motif;

    @ManyToOne
    @JoinColumn(name = "annee_scolaire_id", nullable = false)
    private AnneeScolaire anneeScolaire;

    @ManyToOne
    @JoinColumn(name = "direction_regionale_id")
    private DirectionRegionale directionRegionale;

}
```

Figure 41 : Structure de la table « revisions_budgetaires »

❖ Pour la table « depenses »



```
@Setter
@Getter
@RequiredArgsConstructor
@AllArgsConstructor
@Builder
@Table(name = "depenses")
@Entity
public class Depense implements Serializable {

    /**
     * Classe créée le 07/05/2023
     * Dernière modification : 14/07/2023
     */

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @Temporal(TemporalType.TIMESTAMP)
    @JsonFormat(pattern = "yyyy-MM-dd HH:mm")
    @Column(nullable = false)
    private Date date = new Date(System.currentTimeMillis());

    @Column(nullable = false)
    private Double montant;

    @Column(nullable = false)
    private String commentaires;

    @Column(nullable = false)
    private String reference;

    @Column(nullable = true)
    private boolean demandeModification = Boolean.FALSE;

    @Column(nullable = true)
    private String facture;

    @ManyToOne
    @JoinColumn(name = "imputation_budgetaire_id", nullable = false)
    private ImputationBudgetaire imputationBudgetaire;
}
```

Figure 42 : Structure de la table « depenses »

- ❖ Pour la table « sources_de_financements »



```
@Setter
@Getter
@RequiredArgsConstructor
@AllArgsConstructor
@Builder
@Table(name = "sources_de_financements")
@Entity
public class SourceDeFinancement implements Serializable
{
    /**
     * Classe créée le 07/05/2023
     * Dernière modification :
     */

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @Column(nullable = false)
    private String code;

    @Column(nullable = false)
    private String libelle;
}
```

Figure 43 : Structure de la table « sources_de_financements »

En adoptant cette approche, nous avons créé une base de données robuste et adaptable, conforme aux besoins de notre projet et aux possibilités offertes par Spring Boot et Hibernate.

PARTIE 4 : GUIDE D'EXPLOITATION

Le chapitre dédié à la réalisation et à la mise en œuvre de notre application présente les outils et technologies utilisés pour concrétiser le projet. Toutefois, il est important de satisfaire certaines conditions préalables pour déployer l'application en environnement de production. De plus, une politique d'administration et de maintenance doit être établie par l'utilisateur afin d'assurer une utilisation optimale de l'application. Dans ce document, nous aborderons la configuration matérielle et logicielle, ainsi que les normes de déploiement et de suivi de notre application.

4.1 CONFIGURATION MATERIELLE ET LOGICIEL

4.1.1 Configuration matérielle

Afin de procéder à la mise en exploitation de notre application, il est essentiel de s'assurer que les serveurs et les postes clients répondent aux spécifications requises suivantes :

	Configuration minimale	Configuration requise
SERVEUR	Processeur : 2.2 GHz, 64Bits Disque Dur : 320 Gb RAM : 2 Gb	Processeur : Core i3 3.6 GHz, 64Bits Disque Dur : 500 Gb RAM : 4 Gb
CLIENT	Processeur : 1.8 GHz, 64Bits Disque Dur : 60 Gb RAM: 2 Gb	Processeur : 2.2 GHz, 64Bits Disque Dur : 320 Gb RAM: 4 Gb

Nous recommandons que toutes les machines serveurs soient pilotées par les systèmes d'exploitation Windows7 et postérieur de type 64 bits. Les postes clients peuvent avoir n'importe quel système d'exploitation doté d'un navigateur web.

4.1.2 Configuration logicielle

Afin de rendre notre application accessible en ligne, il est nécessaire de déployer son backend (application Spring Boot) ainsi que son frontend (application Angular) sur des serveurs d'hébergement. Les clients quant à eux, ils pourront y avoir accès à partir d'un navigateur web.

Appareil	Configuration minimale requise
Serveur hébergeant le web service «E-budget-ecole API »	Système d'exploitation : Linux Serveur d'application : Apache Tomcat
Serveur hébergeant la plateforme «E-budget-ecole»	Système d'exploitation : Au choix entre Linux et Windows server Serveur d'application : Apache HTTP Server
Serveur de hébergeant la base de données	Système de gestion de base de données : MySQL 5.5 ou plus
Post client	Système d'exploitation : N'importe quel système d'exploitation Navigateur : Mozilla Firefox 61 ou plus, Google Chrome 68 ou plus, Microsoft Edge.

4.2 DEPLOIEMENT ET SUIVI

4.2.1 Déploiement

Cette section concerne le déploiement de nos applications, qui se composent d'une partie front-end développée en Angular et une autre partie back-end développée en Spring Boot. Nous expliquerons le déploiement qui a été effectué pour chaque type d'application.

4.2.1.1 Déploiement de la partie backend

Le déploiement de la partie backend de notre application nécessite la mise en place d'un serveur approprié avec les composants logiciels requis :

- ✓ Serveur Apache Tomcat, WildFly ou IBM WebSphere ;
- ✓ JRE (Java Runtime Environment) ;

❖ Exemple de déploiement sur RENDER

Prérequis :

- ✓ Une application Spring boot testée en locale sur notre ordinateur ;
- ✓ Un compte Github¹⁰ ;
- ✓ Un compte Render¹¹.

Nous allons dans ce cas, déployer notre application sur Render à partir d'un conteneur Docker.

**Étape 1. Générer le fichier jar permettant de déployer l'api avec la commande
.\mvnw install**

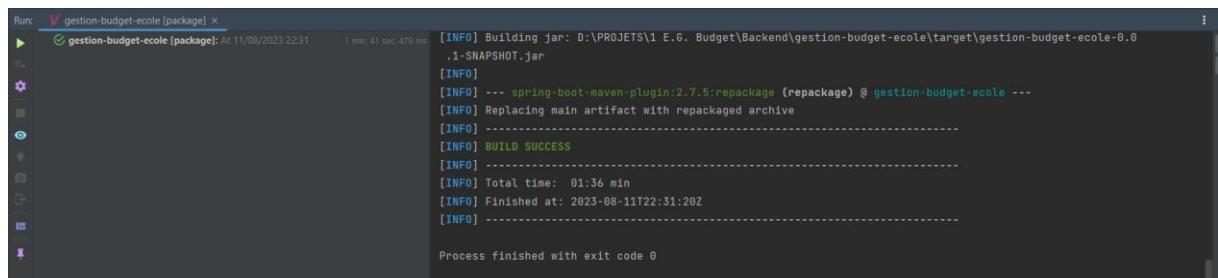


```
Terminal Local x + Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/powershell

PS D:\PROJETS\1 E.G. Budget\Backend\gestion-budget-ecole> .\mvnw install
```

Figure 44 : Commande de génération d'un fichier jar à partir d'une API Spring boot



```
Run: gestion-budget-ecole [package] x
  gestion-budget-ecole [package]: At 11/09/2023 22:31 1 min, 41 sec, 479 ms
  [INFO] Building jar: D:\PROJETS\1 E.G. Budget\Backend\gestion-budget-ecole\target\gestion-budget-ecole-0.0.1-SNAPSHOT.jar
  [INFO]
  [INFO] --- spring-boot-maven-plugin:2.7.5:repackage (repackage) @ gestion-budget-ecole ---
  [INFO] Replacing main artifact with repackaged archive
  [INFO]
  [INFO] BUILD SUCCESS
  [INFO]
  [INFO] Total time:  01:36 min
  [INFO] Finished at: 2023-09-11T22:51:20Z
  [INFO]
  [INFO] Process finished with exit code 0
```

Figure 45 : Génération du fichier jar effectuée

Après cette étape, un fichier jar sera créé dans le dossier target positionné à la racine du projet

¹⁰ Lien du site internet : <http://github.com>

¹¹ Lien du site internet : <https://render.com/>

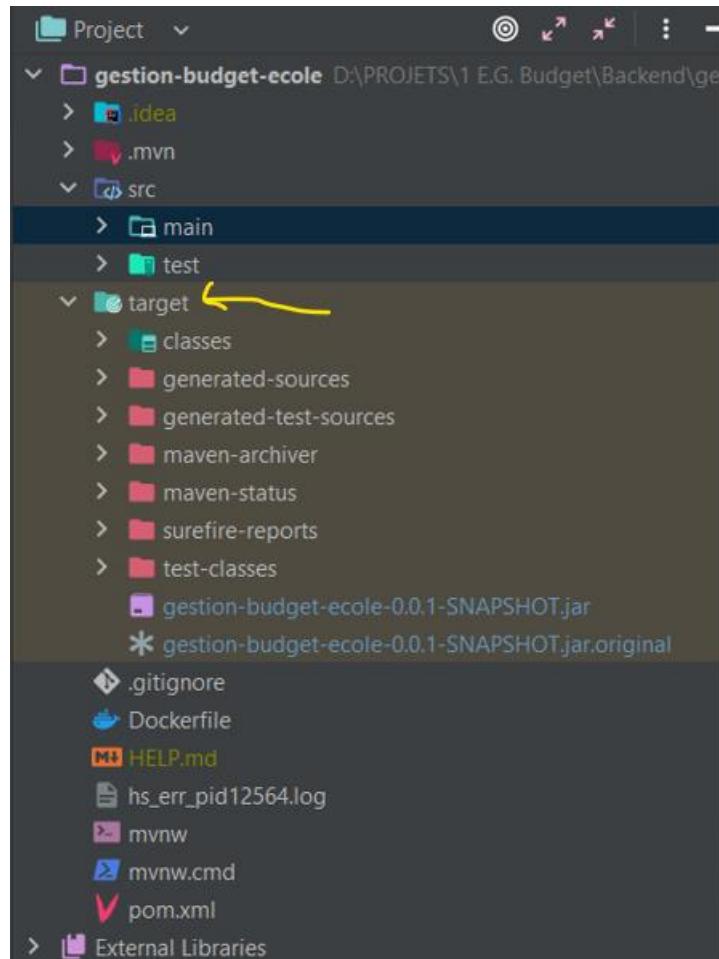


Figure 46 : Dossier target contenant le fichier jar généré

Étape 2. Créer et éditer un fichier Dockerfile à la racine du projet

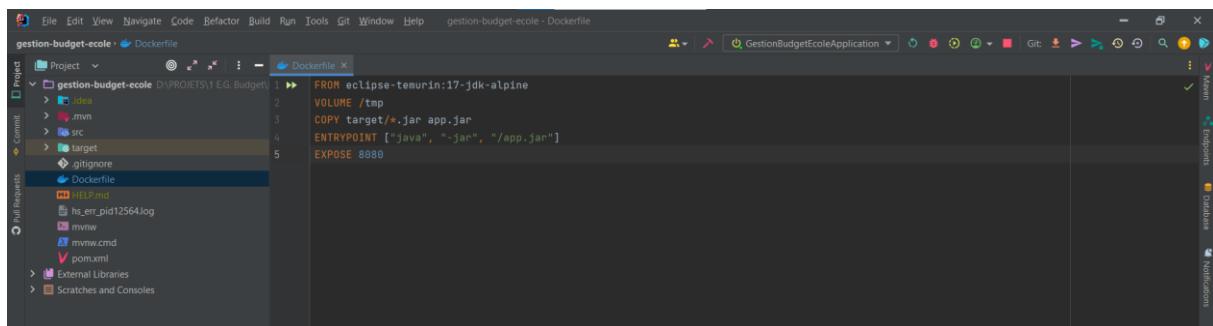


Figure 47 : Fichier Dockerfile

Explication des différentes lignes :

1. FROM eclipse-temurin:17-jdk-alpine

Cette ligne indique que votre conteneur Docker sera basé sur une image appelée "eclipse-temurin" avec la version "17-jdk-alpine". "eclipse-temurin" est une distribution d'OpenJDK, qui est une implémentation open-source de la plateforme Java.

2. VOLUME /tmp

Cette ligne définit un point de montage pour un volume dans le conteneur. Le répertoire /tmp sera utilisé pour stocker des données persistantes en dehors du conteneur, ce qui permettra à l'application de lire et d'écrire des données dans ce répertoire même après la fin de l'exécution du conteneur.

3. COPY target/*.jar app.jar

Cette ligne copie tous les fichiers avec l'extension ".jar" du répertoire "target" situé à la racine de notre projet et les renomme en "app.jar".

4. ENTRYPOINT ["java", "-jar", "/app.jar"]

Cette ligne définit la commande qui sera exécutée lorsque le conteneur sera démarré. Dans ce cas, il exécute la commande Java en lançant le fichier "app.jar" qui a été copié précédemment. Cela lance notre application Java contenue dans le JAR.

5. EXPOSE 8080

Cette ligne indique que le conteneur expose le port 8080. Nous devrions lier ce port lors de l'exécution du conteneur pour que notre application soit accessible depuis l'extérieur.

Étape 3. Mettre le projet sur github

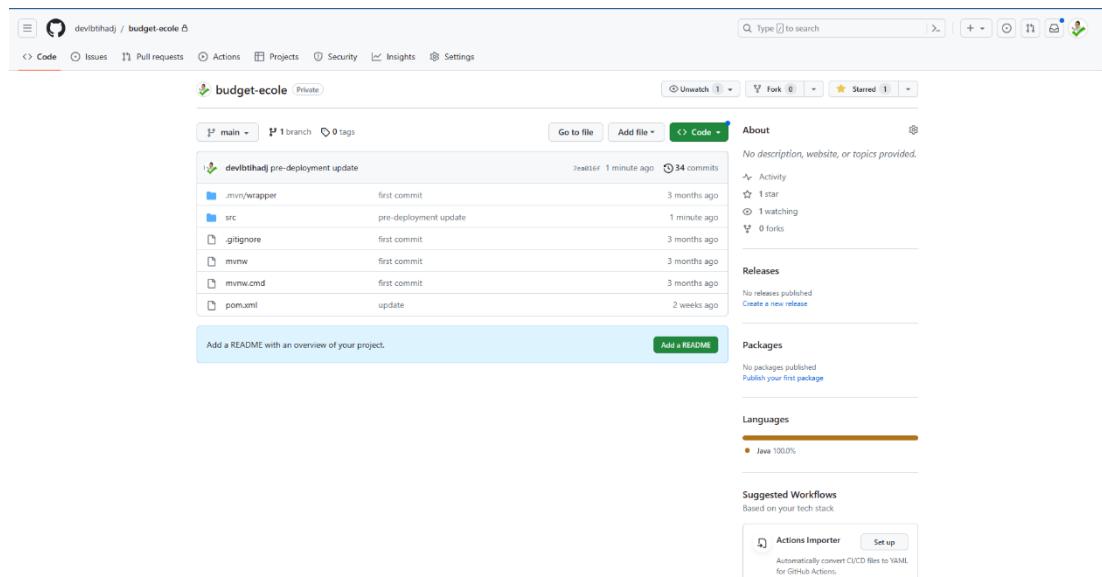


Figure 48 : Repository Github de l'API du projet

Étape 4. Se connecter Render

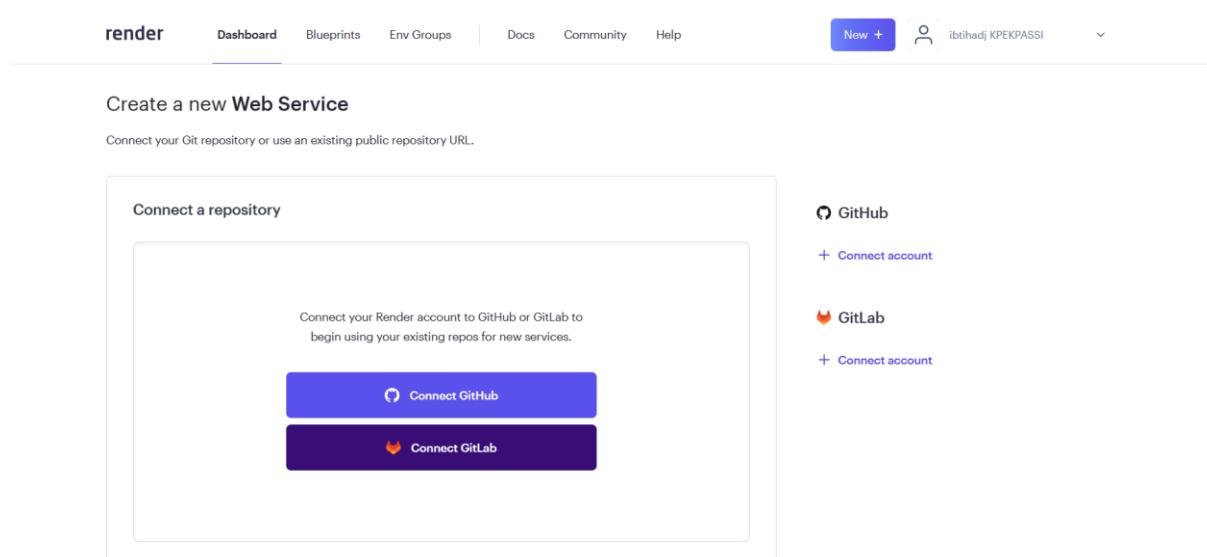


Figure 49 : Interface de Render

Étape 5. Accorder les permissions nécessaires à Render sur notre compte Github

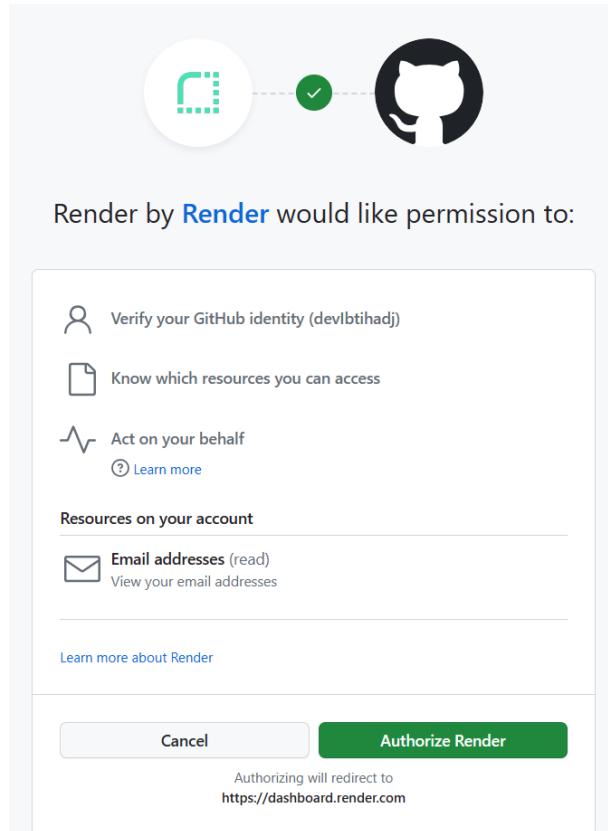


Figure 50 : Accord des autorisations à Render sur notre compte Github

Étape 6. Créer le web service et valider les paramètres

You are deploying a web service for [devlbtihadj/budget-ecole](#).

Name
A unique name for your web service.

Region
The **region** where your web service runs.

Branch
The repository branch used for your web service.

Root Directory Optional
Defaults to repository root. When you specify a **root directory** that is different from your repository root, Render runs all your commands in the **specified directory** and ignores changes outside the directory.

Runtime
The runtime for your web service.

Figure 51 : Crédit d'un web service sur github

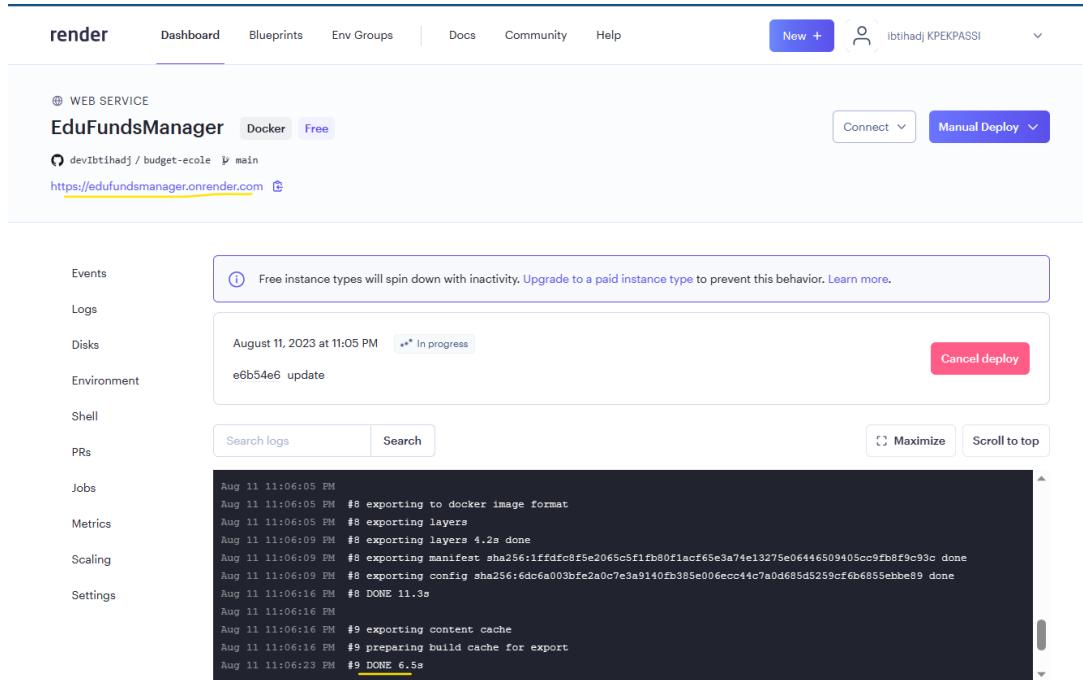


Figure 52 : Déploiement de la partie backend effectué sur Render

4.2.1.2 Déploiement de la partie Front end

Le déploiement de la partie frontend de notre application implique la configuration d'un environnement approprié pour assurer son bon fonctionnement :

- ✓ Hébergement sur un serveur HTTP, tel que Apache HTTP Server ou Nginx ;
- ✓ Assurance que les dépendances et packages requis sont correctement installés à l'aide de Node.js et du gestionnaire de paquets npm ;

Le déploiement réussi de la partie frontend permettra aux utilisateurs d'accéder à l'application à travers une interface utilisateur conviviale et réactive, offrant une expérience fluide et intuitive.

❖ Exemple de déploiement sur FIREBASE

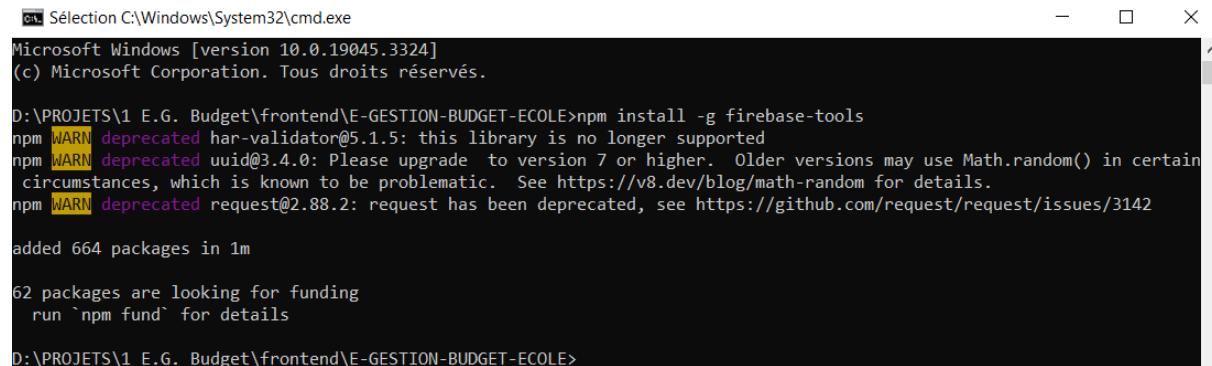
Prérequis :

- ✓ Une application Angular testée en locale sur notre ordinateur ;
- ✓ Configuration de Firebase Hosting ;
- ✓ Un compte Google.

Un déploiement sera fait sur firebase depuis notre ordinateur dans notre cas.

Pour configurer Firebase Hosting, nous passerons par 5 étapes.

Etape 1. Installation de la CLI Firebase avec la commande ‘**npm install -g firebase-tools**’



```
 Sélection C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19045.3324]
(c) Microsoft Corporation. Tous droits réservés.

D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE>npm install -g firebase-tools
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

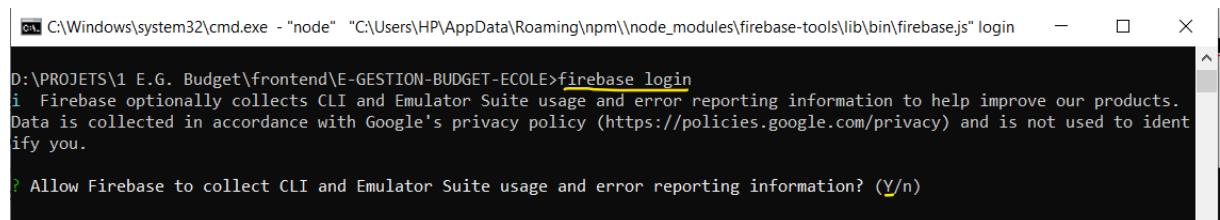
added 664 packages in 1m

62 packages are looking for funding
  run `npm fund` for details

D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE>
```

Figure 53: Installation de la CLI Firebase

Etape 2. Connexion à Google avec la commande ‘**firebase login**’



```
 C:\Windows\system32\cmd.exe - "node" "C:\Users\HP\AppData\Roaming\npm\node_modules\firebase-tools\lib\bin\firebase.js" login
D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE>firebase login
i Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products.
Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? (Y/n)
```

Figure 54 : Connexion à Google

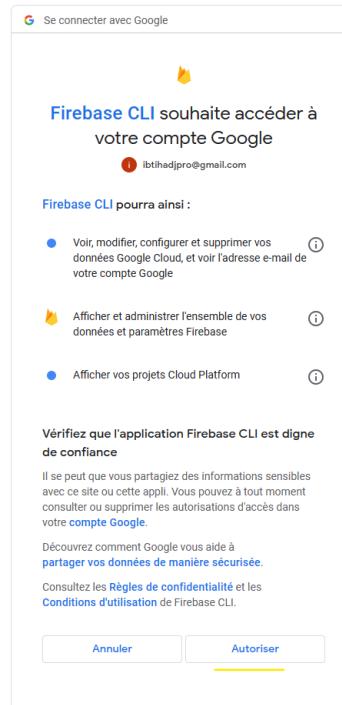


Figure 55 : Accord des autorisations à Firebase CLI

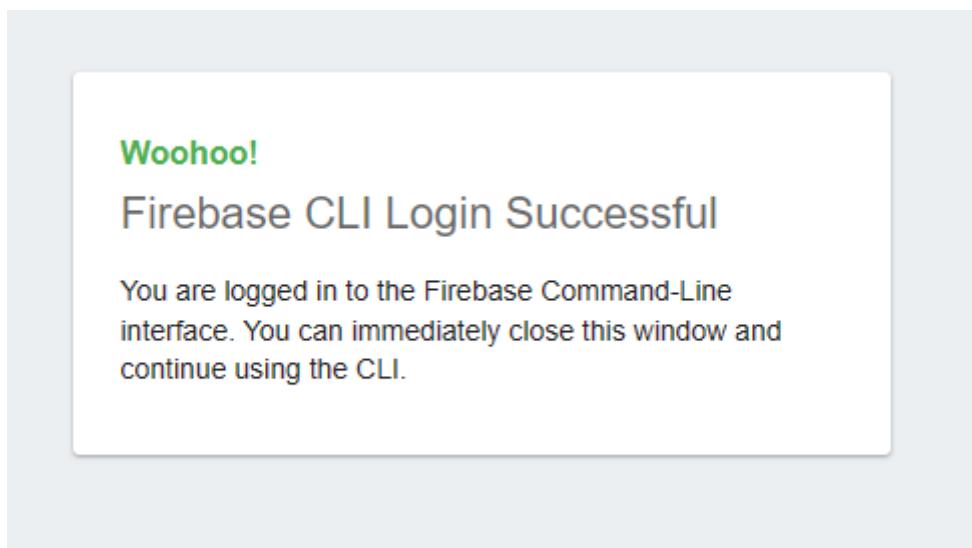


Figure 56 : Connexion à Firebase CLI réussie

Etape 3. Crée un projet Firebase

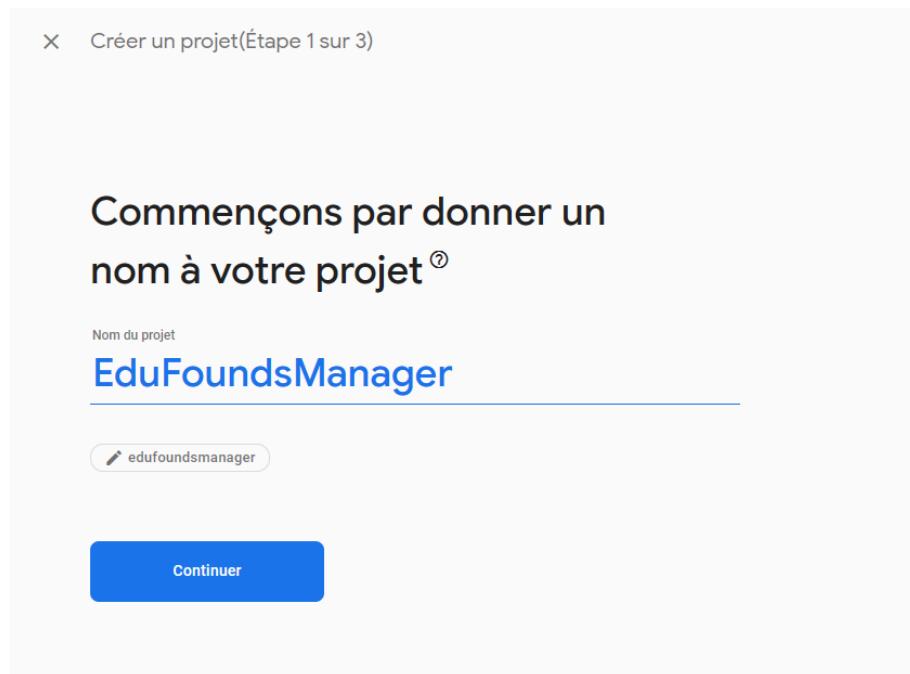


Figure 57 : Création d'un projet sur Firebase

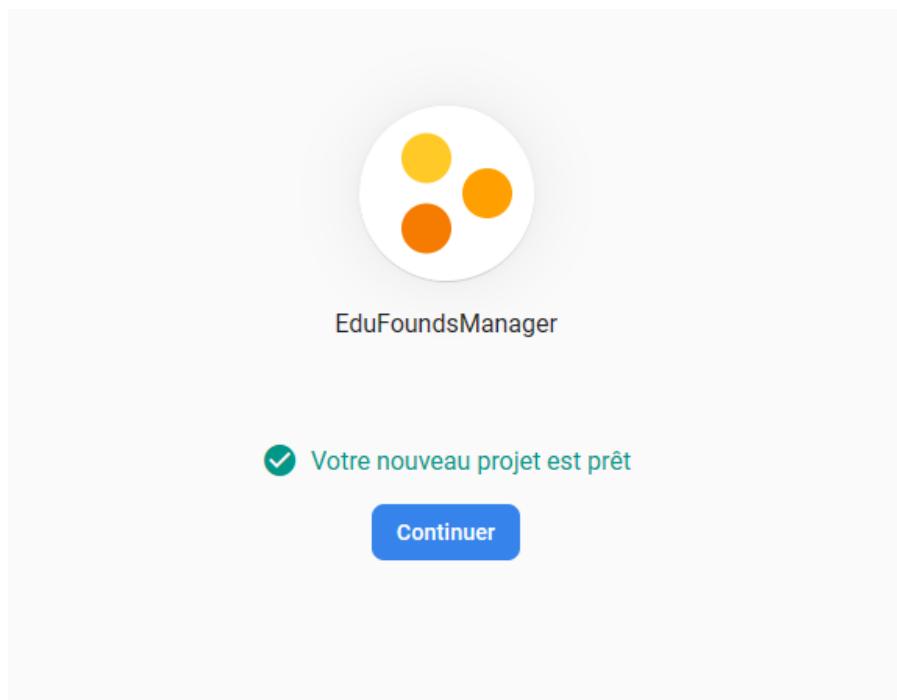


Figure 58 : Projet Firebase créé

Etape 4. Liaison entre notre application et le projet créé sur Firebase à partir de la commande **'firebase init hosting'**

Figure 59 : Liaison entre notre application front end et le projet créé sur Firebase

```
C:\Windows\system32\cmd.exe

==> Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? dist
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? Yes
? File dist/index.html already exists. Overwrite? No
i  Skipping write of dist/index.html

i  Detected a .git folder at D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE
i  Authorizing with GitHub to upload your service account to a GitHub repository's secrets store.

Visit this URL on this device to log in:
https://github.com/login/oauth/authorize?client_id=89cf50f02ac6aaed3484&state=675070367&redirect_uri=http%3A%2F%2Flocalhost%3A9005&scope=read%3Auser%20repo%20public_repo

Waiting for authentication...

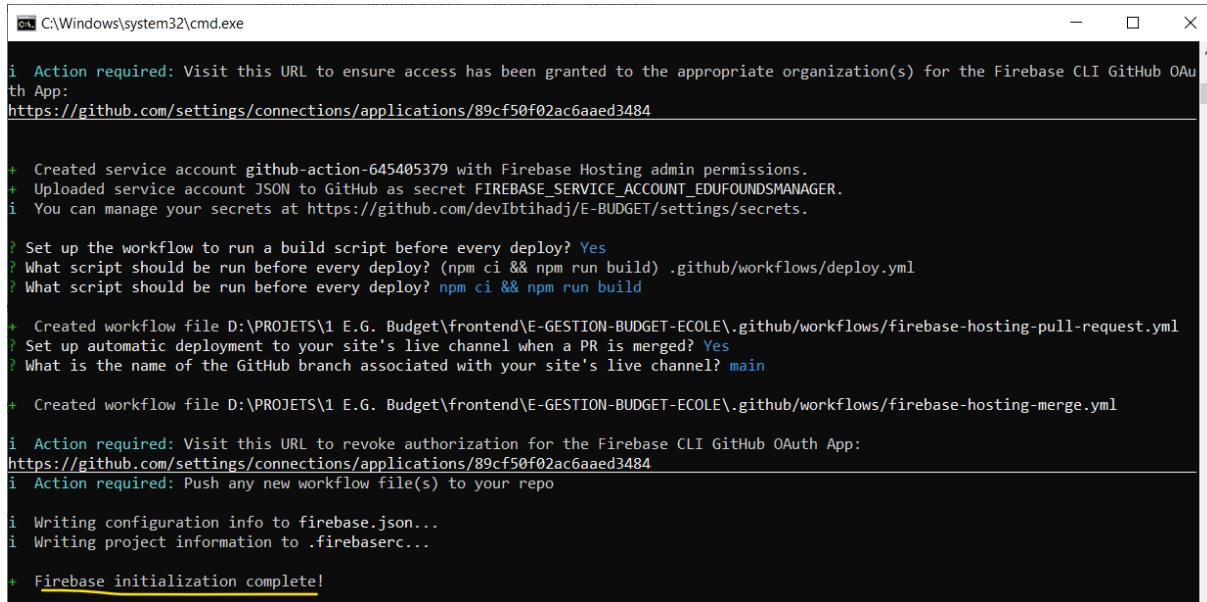
+  Success! Logged into GitHub as devIbtihadj

? For which GitHub repository would you like to set up a GitHub workflow? (format: user/repository) devIbtihadj/E-BUDGET

!  The provided authorization cannot be used with this repository. If this repository is in an organization, did you remember to grant
access?

i  Action required: Visit this URL to ensure access has been granted to the appropriate organization(s) for the Firebase CLI GitHub OAuth
App:
```

Figure 60 : Paramétrages des préférences



```
C:\Windows\system32\cmd.exe

i Action required: Visit this URL to ensure access has been granted to the appropriate organization(s) for the Firebase CLI GitHub OAuth App:
https://github.com/settings/connections/applications/89cf50f02ac6aaed3484

+ Created service account github-action-645405379 with Firebase Hosting admin permissions.
+ Uploaded service account JSON to GitHub as secret FIREBASE_SERVICE_ACCOUNT_EDUFOUNDSMANAGER.
i You can manage your secrets at https://github.com/devIbtihadj/E-BUDGET/settings/secrets.

? Set up the workflow to run a build script before every deploy? Yes
? What script should be run before every deploy? (npm ci && npm run build) .github/workflows/firebase-hosting-pull-request.yml
? What script should be run before every deploy? npm ci && npm run build

+ Created workflow file D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE\.github/workflows/firebase-hosting-pull-request.yml
? Set up automatic deployment to your site's live channel when a PR is merged? Yes
? What is the name of the GitHub branch associated with your site's live channel? main

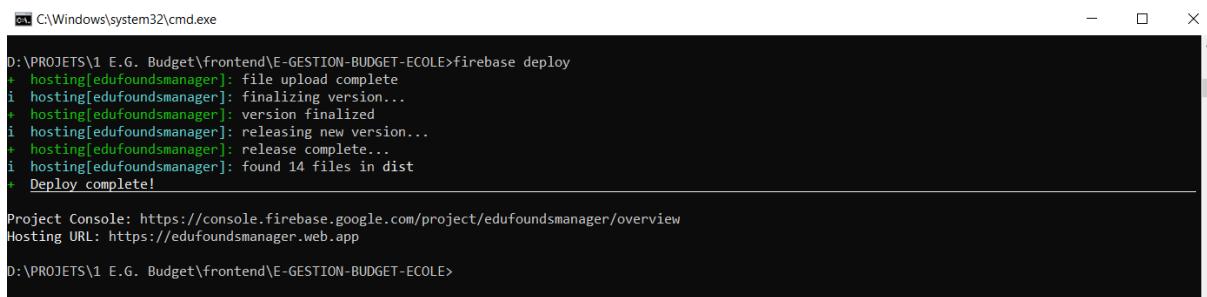
+ Created workflow file D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE\.github/workflows/firebase-hosting-merge.yml

i Action required: Visit this URL to revoke authorization for the Firebase CLI GitHub OAuth App:
https://github.com/settings/connections/applications/89cf50f02ac6aaed3484
i Action required: Push any new workflow file(s) to your repo

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...

+ Firebase initialization complete!
```

Figure 61 : Liaison effectuée avec succès



```
C:\Windows\system32\cmd.exe

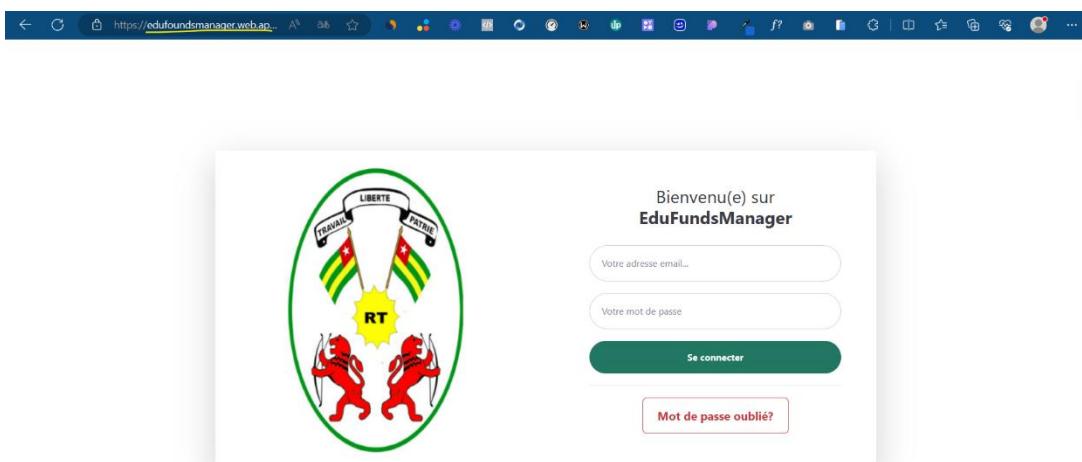
D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE>firebase deploy
+ hosting[edufoundsmanager]: file upload complete
i hosting[edufoundsmanager]: finalizing version...
+ hosting[edufoundsmanager]: version finalized
i hosting[edufoundsmanager]: releasing new version...
+ hosting[edufoundsmanager]: release complete...
i hosting[edufoundsmanager]: found 14 files in dist
+ Deploy complete!

Project Console: https://console.firebaseio.com/project/edufoundsmanager/overview
Hosting URL: https://edufoundsmanager.web.app

D:\PROJETS\1 E.G. Budget\frontend\E-GESTION-BUDGET-ECOLE>
```

Figure 62 : Déploiement effectif sur Firebase

L'application est désormais accessible via internet.¹²



¹² Lien de l'application : <https://edufoundsmanager.web.app>

4.2.2 Suivi

Dans le cadre du déploiement d'une application, il est essentiel d'assurer un suivi continu afin de prévenir tout problème pouvant entraîner une interruption de service. Le suivi de bugs et la gestion des versions sont les principales mesures à mettre en place pour assurer ce suivi efficace de l'application. Ces actions permettent de détecter et résoudre rapidement les problèmes rencontrés, garantissant ainsi une expérience utilisateur optimale.

4.2.2.1 Suivi des bugs

Odda Technology dispose en interne d'une équipe de techniciens spécialisés chargés de la résolution des bugs et des dysfonctionnements des applications. Cette équipe est en mesure d'intervenir rapidement pour identifier et corriger les problèmes rencontrés, assurant ainsi la stabilité et la fiabilité des applications.

4.2.2.2 Versionning du projet

1. Versionning de l'application

La gestion des versions d'un projet est essentielle pour suivre l'évolution des fichiers au fil du temps et pouvoir revenir à une version antérieure si nécessaire. Elle permet de garder une trace des changements effectués et assure la disponibilité d'une version stable des fichiers à tout moment. Dans le cadre de notre projet, nous avons utilisé Git, un système de gestion de versions, en combinaison avec GitHub, une plateforme de développement collaboratif basée sur Git (comme présenté dans la partie Réalisation et mise en œuvre).

Avec ce guide d'exploitation à leur disposition, les opérateurs sont en mesure de déployer notre application et de maîtriser les principaux mécanismes de suivi mis en place. Il est également important de former les utilisateurs afin qu'ils puissent utiliser l'application de manière optimale. La documentation qui suivra présentera un guide d'utilisation de notre application ainsi que les actions à prendre en cas d'erreur.

2. Versionning de la base donnée

La sauvegarde de la base de données est essentielle pour éviter de potentielles pertes de données en cas d'incidents ou de défaillance du serveur. Il existe deux types de sauvegardes : **la sauvegarde logique** et **la sauvegarde physique**.

La sauvegarde logique, également qualifiée de sauvegarde à chaud, s'avère recommandée en raison de sa capacité à assurer un accès continu aux utilisateurs de l'application. Cette méthode peut être exécutée sans interrompre le fonctionnement de la base de données, grâce à l'utilisation d'instructions SQL. Dans le cadre de notre projet, des sauvegardes automatiques sont programmées à intervalles de 24 heures. Par ailleurs, l'administrateur du système conserve la possibilité d'initier manuellement une sauvegarde, soit à titre préventif pour parer à toute défaillance du système de gestion de base de données. Les données sauvegardées demeurent exportables à tout moment vers un support physique externe, tel qu'un disque de sauvegarde.

En mettant en place une telle politique de sauvegarde adéquate, les propriétaires de l'application et du système sont protégés contre les conséquences financières de la perte de données.

4.3 MAINTENANCE : ACTION A MENER EN CAS DE CERTAINES ERREURS

La maintenance de l'application implique la documentation des erreurs qui peuvent survenir lors de son utilisation, ainsi que les actions à entreprendre pour résoudre ces erreurs.

Les erreurs les plus courantes comprennent les exceptions suivantes :

- Erreur 400 : La syntaxe de la requête utilisée pour accéder à la ressource est invalide. L'utilisateur doit corriger la requête.
- Erreur 403 : L'utilisateur n'est pas autorisé à accéder à la page à laquelle il tente d'accéder. Il peut contacter l'administrateur pour obtenir des explications supplémentaires.
- Erreur 404 : L'utilisateur tente d'accéder à une URL qui n'est pas disponible dans le contexte de l'application.

- Erreur 500 : Erreur interne du serveur. Dans ce cas, l'utilisateur doit actualiser la page. Si le problème persiste, il doit en informer le support technique.

Exemple illustrant la page d'erreur 404



Figure 63 : Exemple illustrant la page d'erreur 404

PARTIE 5 : GUIDE D'UTILISATION

Le guide d'utilisation ou mode d'emploi est un document expliquant le fonctionnement d'un logiciel ou d'un objet en général. Concrètement, il permet aux utilisateurs de prendre en main le logiciel, de se familiariser à son fonctionnement et son interface. Il sera également très utile pour les formateurs dans le cadre de la formation des utilisateurs.

5.1 DESCRIPTION TEXTUELLE DU LOGICIEL

Avant de lancer l'application, les étapes préalables suivantes doivent être accomplies :

- ❖ S'assurer que le serveur de base de données est démarré.
- ❖ Démarrer le serveur d'application back end.
- ❖ Démarrer le serveur d'application front end.

Une fois ces conditions remplies, tout utilisateur peut accéder à l'application via son navigateur (comme Mozilla Firefox, Google Chrome, ou encore Microsoft Edge) en saisissant l'URL de l'application dans la barre d'adresse d'un nouvel onglet du navigateur. Pour plus de commodité, un utilisateur pourra créer un raccourci de l'application sur le bureau de son ordinateur, ce qui lui permettra de lancer directement l'application.

Cette séquence garantit un lancement fluide de l'application, en garantissant que les serveurs nécessaires sont actifs et que la connectivité du client est établie. Une fois ces prérequis vérifiés, l'utilisateur peut profiter de l'application sans difficulté.

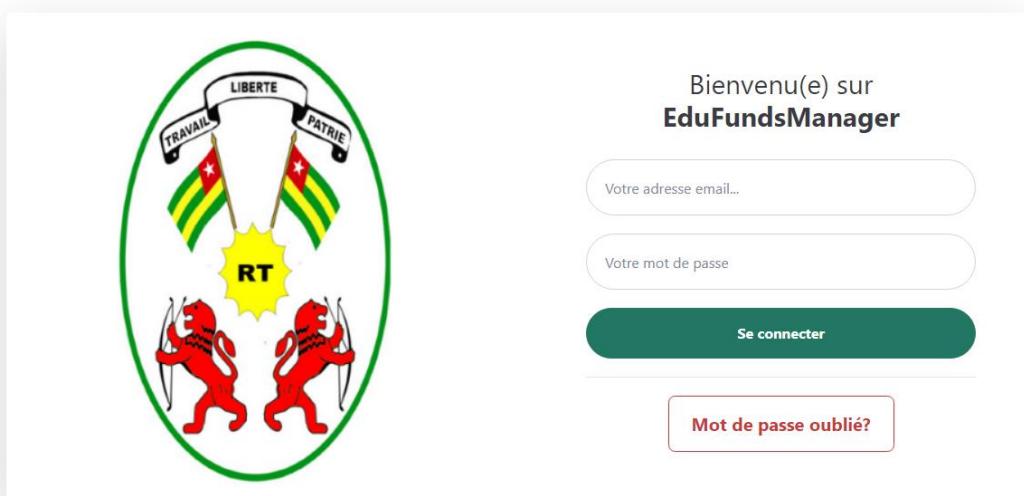


Figure 64 : Page de connexion

Pour se connecter, l'utilisateur saisit son adresse email et son mot de passe. Il appuie ensuite sur le bouton « Se connecter ». Après vérification du système, il est automatiquement redirigé vers sa page d'accueil

Une page de récupération de compte est prévue pour les utilisateurs qui ont oublié leur mot de passe, leur permettant ainsi de reprendre le contrôle de leur compte.

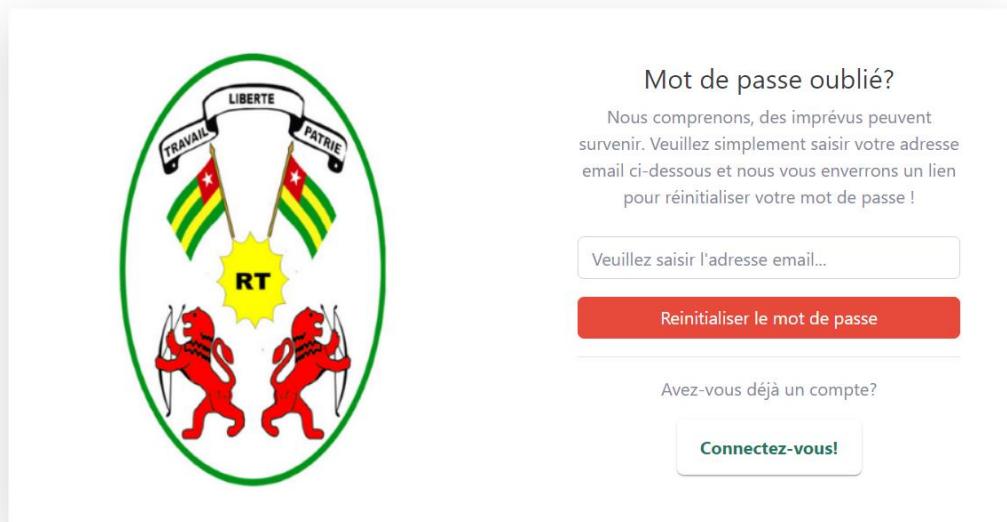


Figure 65 : Page de demande de réinitialisation de mot de passe

5.2 PLAN DE NAVIGATION

❖ Administrateur

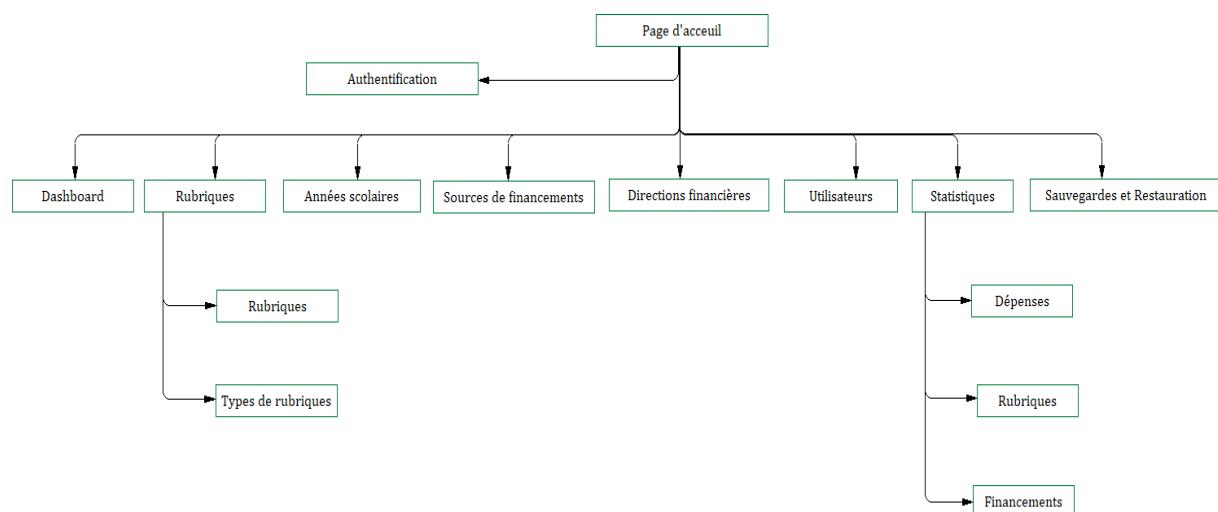


Figure 66 : Plan de navigation de l'acteur : Administrateur

❖ Directeur administrateur et financier

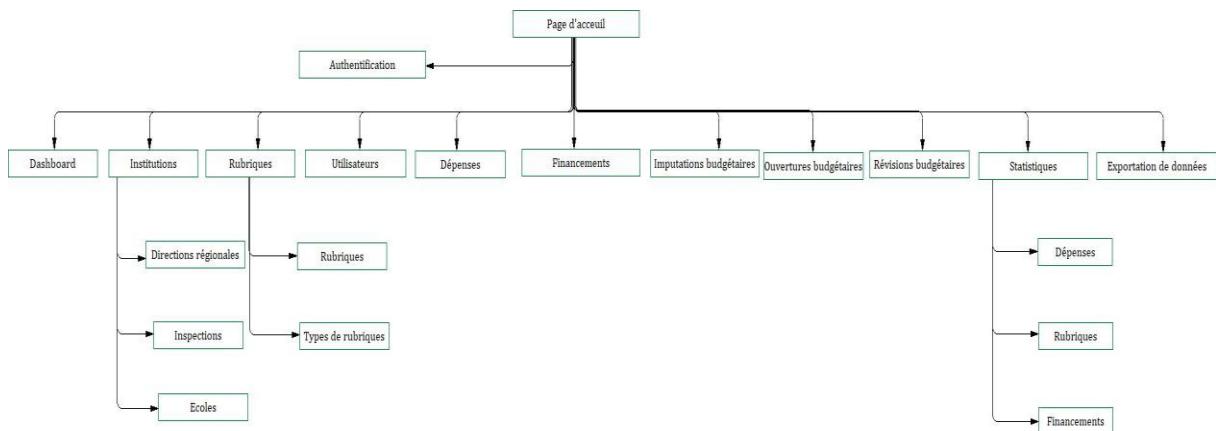


Figure 67 : Plan de navigation de l'acteur : Directeur administratif et financier

❖ Directeur régionale

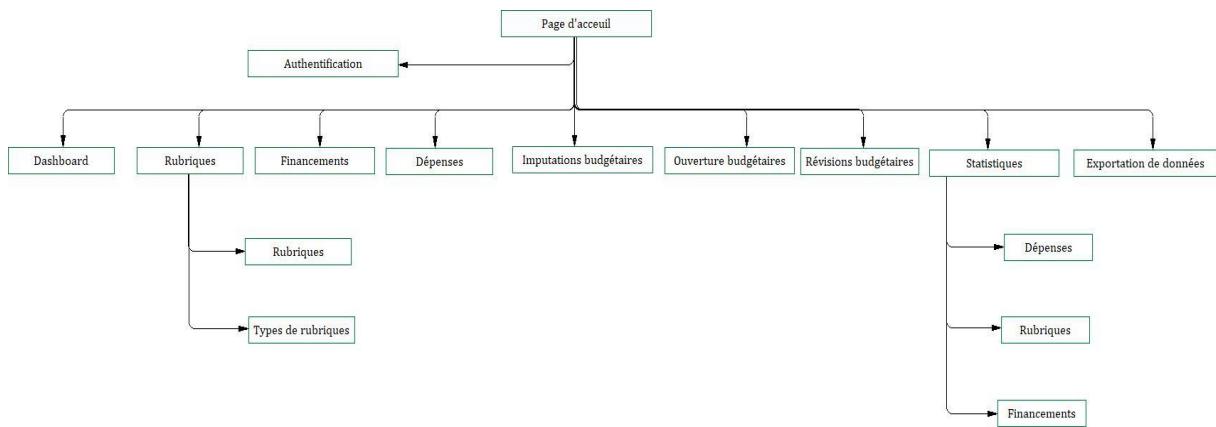


Figure 68 : Plan de navigation de l'acteur : Directeur régional

❖ Inspecteur

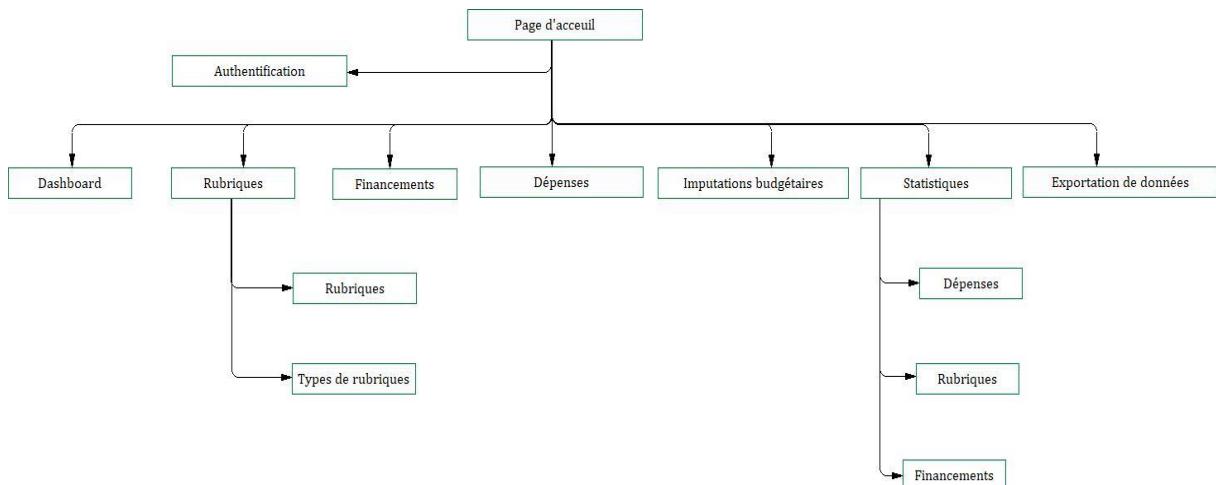


Figure 69 : Plan de navigation de l'acteur : Inspecteur

❖ Directeur d'école et Comptable d'école

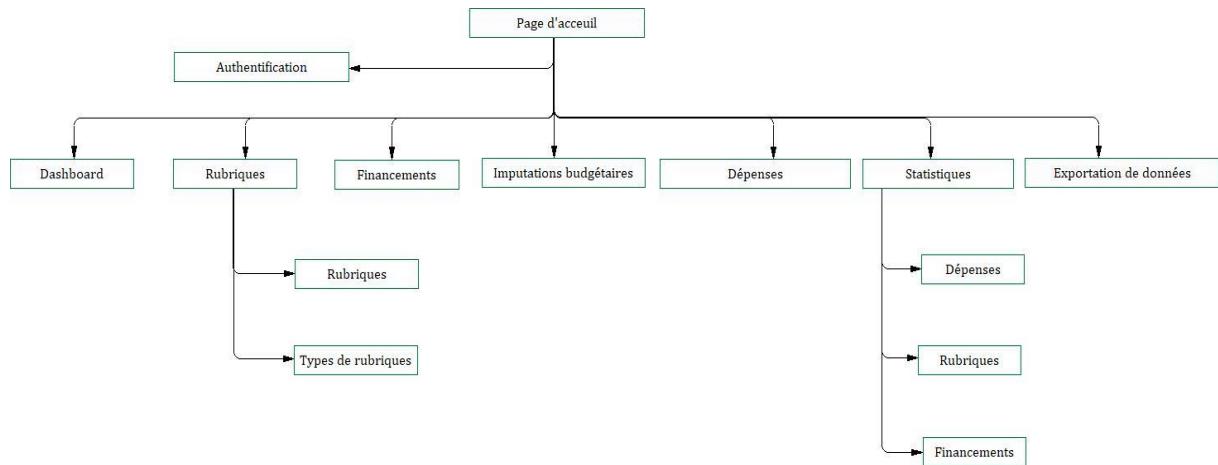


Figure 70 : Plan de navigation des acteurs : Directeur d'école et Comptable d'école

5.3 PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION

Nous considérons le scénario nominal où l'utilisateur arrive à parfaitement se connecter à son compte.

- ❖ Serveur de base de données accessible et démarré
- ❖ Serveur d'application back end accessible et démarré ;
- ❖ Serveur d'application front end accessible et démarré.

5.3.1 Interface de gestion des sources de financements

Cette interface permet à l'administrateur de gérer les sources de financements utilisés par écoles dans l'élaboration de leurs budgets.

Code ↑↓	Libelle ↑↓	
010	Initiatives de financement participatif	
009	Fonds de dotation de l'école	
008	Programmes de parrainage d'entreprises	
007	Fonds de soutien communautaire	
006	Subventions gouvernementales	

Affichage de 1 à 5 sur un total de 10 enregistrements << < 1 2 > >> 5 <<

Figure 71 : Interface de gestion des sources de financements

5.3.2 Interface de gestion des rubriques

Cette interface permet à l'administrateur de gérer les rubriques utilisées par écoles dans l'élaboration de leurs budgets.

Code ↑↓	Libelle ↑↓	Détails	
009	Projets de recherche		
008	Formation professionnelle		
007	Matériel spécialisé		
002	Achats de matériel technique		

Affichage de 1 à 4 sur un total de 4 enregistrements << < 1 > >> 5 <<

Figure 72 : Interface de gestion des rubriques

5.3.3 Interface de suivi des dépenses d'une école

Cette interface permet aux directeurs et comptables d'écoles de suivre leurs dépenses en fonction du temps au cours d'une année scolaire

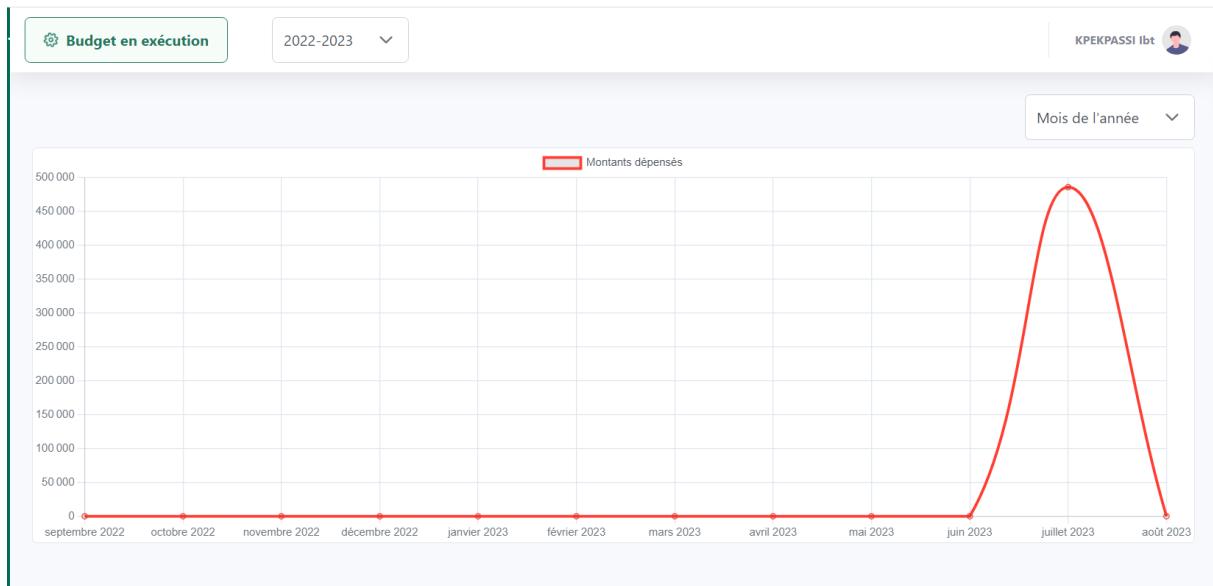


Figure 73 : Interface de suivi des dépenses d'une école

5.3.4 Interface de création d'une nouvelle révision budgétaire

Cette interface permet aux directeurs régionaux de l'enseignement général de pouvoir créer une nouvelle révision budgétaire.

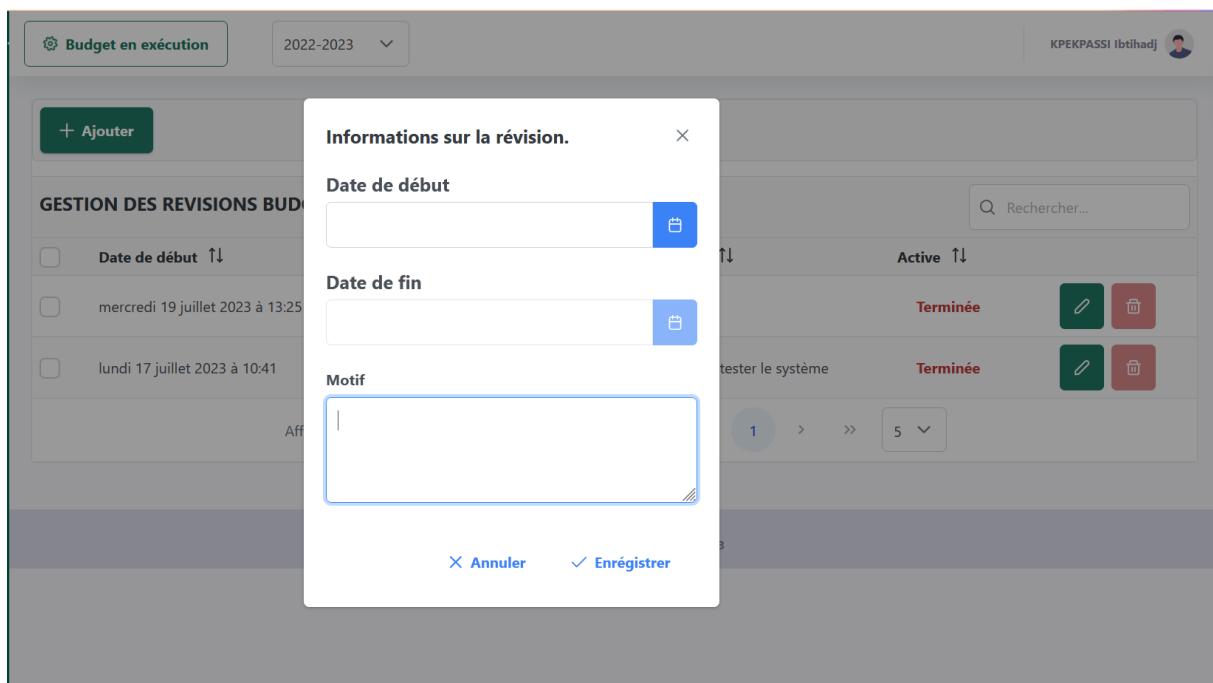


Figure 74 : Interface de création d'une nouvelle révision budgétaire

5.3.5 Tableau de bord de l'administrateur

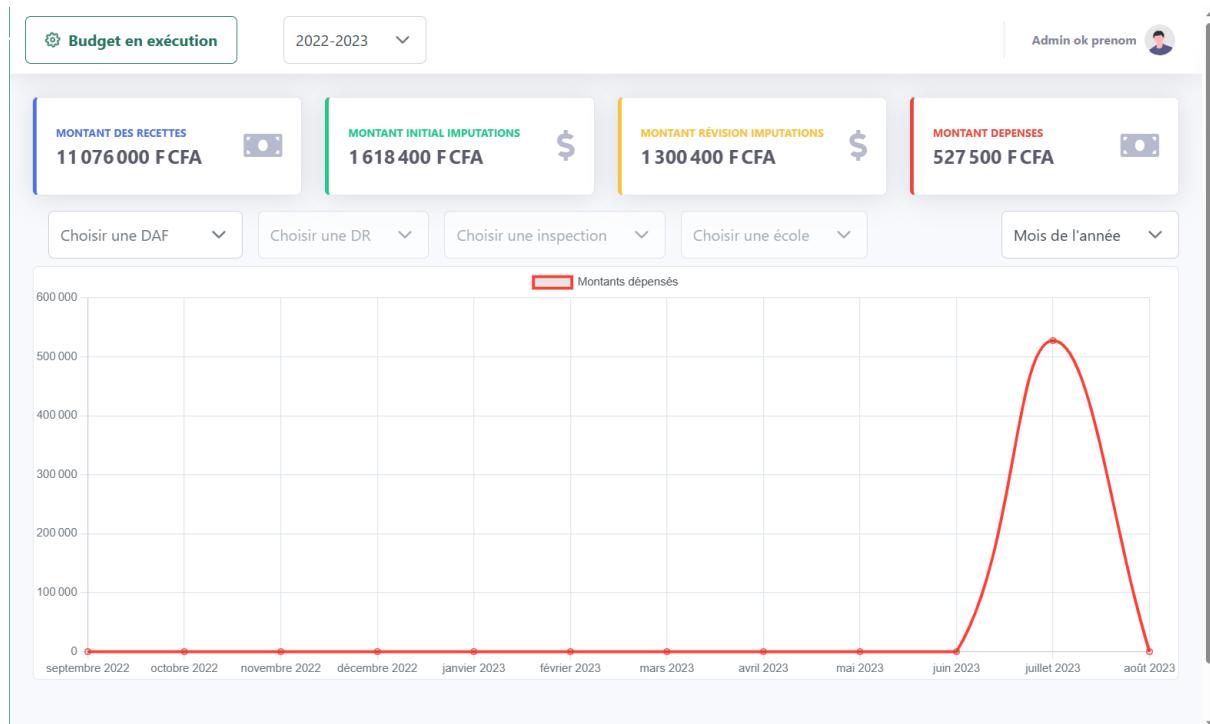


Figure 75 : Tableau de bord de l'administrateur

5.4 PRESENTATION DES ETATS

5.4.1 ETAT D'EXECUTION DE BUDGET PAR SOUS RUBRIQUE : CAS D'UNE ECOLE

MINISTERE DES ENSEIGNEMENTS PRIMAIRE,
SECONDAIRE, TECHNIQUE ET DE L'ARTISANAT
(MEPSTA)
01 BP 1393 Lomé 01 Lomé-Togo



REPUBLIQUE TOGOLAISE
Travail-Liberté-Patrie

2022-2023

ETAT D'EXECUTION PAR SOUS RUBRIQUE : EPP KOSHIGAN

DIRECTION REGIONALE GOLFE
IEPP AGOE-NYIVE EST

EPP KOSHIGAN

Libellé	Montant prévu	Montant dépensé	Reste
Équipements de bureau			
Mobilier de bureau	100000	0	100000
Total	100000	0	100000
Fournitures scolaires			
Cahiers et stylos	500000	400000	100000
Total	500000	400000	100000
Frais de communication			
Communication externe	55000	0	55000
Total	55000	0	55000
Frais de fonctionnement			
Charges locatives	120000	50000	70000
Entretien des locaux	44000	28000	16000
Total	164000	78000	86000
Frais de scolarité			
Matériel pédagogique	80000	0	80000
Total	80000	0	80000
Salaires du personnel			
Salaires enseignants	200000	0	200000
Total	200000	0	200000
Transport scolaire			
Bus scolaire	55000	8000	47000
Total	55000	8000	47000
Total de l'école EPP KOSHIGAN	1154000	486000	668000

ETAT D'EXECUTION PAR SOUS RUBRIQUE : EPP KOSHIGAN

Page 1

Figure 76 : Etat d'execution de budget par sous rubrique : cas d'une ecole

CONCLUSION

Le parcours de ce mémoire a été une exploration approfondie et stimulante à travers les méandres de la conception, du développement et de l'implémentation d'une solution innovante pour la gestion budgétaire des écoles publiques de l'enseignement primaire et secondaire. À chaque étape, nous avons privilégié les valeurs d'efficacité, de transparence et d'accessibilité dans le domaine de l'éducation.

En développant une plateforme web sur mesure, nous avons relevé les défis actuels de la gestion budgétaire et ouvert de nouvelles perspectives pour l'optimisation des ressources. Les technologies de l'information et de la communication ont modernisé les processus essentiels, favorisant des décisions éclairées.

L'intégration méthodique de méthodologies telles que les diagrammes UML et les frameworks a établi des bases solides pour une solution robuste. Chaque étape, de l'analyse aux retours des futurs utilisateurs, visait à répondre aux besoins réels du secteur éducatif.

Ce mémoire va au-delà de l'exploration technique, contribuant réellement à une gestion budgétaire transparente et efficace. Les défis surmontés et les solutions mises en œuvre ont forgé une expérience formatrice, illustrant la puissance du savoir technologique et d'une vision éducative optimale.

En conclusion, cet engagement constant envers l'innovation et l'impact positif se manifeste dans ce mémoire. Puisse la solution présentée inspirer les générations futures à repousser les frontières technologiques pour un avenir éducatif plus prometteur.

BIBLIOGRAPHIE INDICATIVE

❖ Notes de cours

- Cours d'UML de M. KETOGLIO (2021 - 2022) ;
- Cours de Gestion de projets Informatique de M. AMEVOR Kossi (2022 - 2023) ;
- Cours d'Introduction au Génie Logiciel de M. SANI Koffi (2022-2023) ;

❖ Anciens mémoires consultés

- NONON SAA Diyane David Dissoclima (2020 -2021), CONCEPTION ET IMPLEMENTATION DU MODULE DE COMMANDE ET DE SUIVI DES PIECES DETACHEES : CAS DE LA PLATEFORME MEKANO.
- YERIMA Boucharatou (2021 - 2022) MISE EN PLACE D'UN SYSTEME AUTOMATISE DE GESTION DES SALAIRES : CAS DE ODDA TECHNOLOGY

WEBOGRAPHIE INDICATIVE

Tableau 19 : Webographie indicative

Date et Heure	URL des sites consultés
Tout au long du projet	https://stackoverflow.com/
Tout au long du projet	https://angular.io/
Tout au long du projet	http://primeng.org/
Tout au long du projet	https://chat.openai.com/
Tout au long du projet	https://www.youtube.com/
Tout au long du projet	https://www.primefaces.org/diamond/icons.xhtml
02/08/2023 14h 53'	https://icons.getbootstrap.com/

TABLE DES MATIERES

DEDICACES	i
REMERCIEMENTS	ii
AVANT-PROPOS	iii
RESUME.....	iv
ABSTRACT	v
SOMMAIRE.....	vi
GLOSSAIRE.....	viii
LISTE DE FIGURES	ix
LISTE DES TABLEAUX	xii
LISTE DES PARTICIPANTS AU PROJET.....	xiii
INTRODUCTION	1
PARTIE 1 : CAHIER DE CHARGES.....	2
1.1 PRESENTATIONS	3
1.1.1 Présentation de l'IAI-TOGO	3
1.1.2 Présentation du cadre de stage	5
1.2 THEME DU STAGE	8
1.2.1 Présentation du sujet.....	8
1.2.2 Problématique.....	8
1.2.3 Intérêt du sujet.....	9
1.2.3.1 Objectifs.....	9
1.2.3.2 Résultats attendus.....	10
1.3 ETUDE DE L'EXISTANT.....	11
1.4 CRITIQUE DE L'EXISTANT.....	11
1.5 PROPOSITIONS ET CHOIX DE SOLUTION.....	12

1.5.1 PROPOSITIONS DE SOLUTION.....	12
1.5.1.1 MyScol	13
1.5.1.2 EduFundsManager	13
1.5.2 Evaluation technique des solutions	14
1.5.2.1 Evaluation technique de : MyScol.....	14
1.5.2.2 Evaluation technique de : EdufundsManager.....	14
1.5.3 Evaluation financière des solutions	15
1.5.3.1 Evaluation financière de : MyScol	15
1.5.3.2 Evaluation financière de : EduFundsManager.....	17
1.5.4 Choix de la solution	19
1.6 PLANNING PREVISIONNEL DE REALISATION	20
PARTIE 2 : ANALYSE ET CONCEPTION.....	23
2.1 CHOIX ET JUSTIFICATION DE LA METHODE D'ANALYSE	24
2.1.1 L'approche objet.....	24
2.1.2 Pourquoi l'approche objet et non l'approche fonctionnelle ?	24
2.1.3 Le langage de modélisation UML	26
2.1.4 Présentation du processus 2TUP.....	28
2.2 CHOIX ET JUSTIFICATION DE L'OUTIL DE MODELISATION	31
2.2.1 Présentation et justification du choix de PowerAMC.....	31
2.3 ETUDE DETAILLEE DE LA SOLUTION	32
2.3.1 Etude préliminaire	32
2.3.1.1 Diagramme de contexte statique.....	34
2.3.2 Capture des besoins fonctionnels.....	35
2.3.2.1 Diagrammes de cas d'utilisation.....	35
2.3.2.2 Description textuelle de quelques cas d'utilisation.....	45
2.3.2.3 Diagrammes d'activité	56

2.3.2.4 Diagrammes de séquence.....	61
2.3.3 Analyse.....	66
2.3.3.1 Diagrammes de classes.....	66
2.3.4 Conception.....	68
2.3.4.1 Diagramme de déploiement.....	68
PARTIE 3 : REALISATION ET MISE EN ŒUVRE	69
3.1 MATERIEL ET LOGICIELS UTILISES	70
3.1.1 Matériel utilisé	70
3.1.2 Logiciels utilisés	70
3.1.2.1 Outil d'administration de base de donnés	70
3.1.2.2 Outils de programmation et de développement.....	71
3.2 ARCHITECTURES MATERIELLE ET LOGICIELLE DE L'APPLICATION	78
3.2.1 Architecture matérielle.....	78
3.2.2 Architecture logicielle	79
3.3 SECURITE DE L'APPLICATION.....	81
3.4 MISE EN PLACE DE LA BASE DE DONNEES.....	81
PARTIE 4 : GUIDE D'EXPLOITATION	87
4.1 CONFIGURATION MATERIELLE ET LOGICIEL	88
4.1.1 Configuration matérielle	88
4.1.2 Configuration logicielle.....	88
4.2 DEPLOIEMENT ET SUIVI.....	89
4.2.1 Déploiement.....	89
4.2.1.1 Déploiement de la partie backend	89
4.2.1.2 Déploiement de la partie Front end	95
4.2.2 Suivi.....	101
4.2.2.1 Suivi des bugs.....	101

4.2.2.2 Versionning du projet	101
4.3 MAINTENANCE : ACTION A MENER EN CAS DE CERTAINES ERREURS.....	102
PARTIE 5 : GUIDE D'UTILISATION	104
5.1 DESCRIPTION TEXTUELLE DU LOGICIEL	105
5.2 PLAN DE NAVIGATION	106
5.3 PRESENTATION DES DIFFERENTES INTERFACES DE L'APPLICATION.....	108
5.3.1 Interface de gestion des sources de financements	108
5.3.2 Interface de gestion des rubriques.....	109
5.3.3 Interface de suivi des dépenses d'une école.....	109
5.3.4 Interface de création d'une nouvelle révision budgétaire	110
5.3.5 Tableau de bord de l'administrateur.....	111
5.4 PRESENTATION DES ETATS.....	111
5.4.1 ETAT D'EXECUTION DE BUDGET PAR SOUS RUBRIQUE : CAS D'UNE ECOLE	111
CONCLUSION	113
BIBLIOGRAPHIE INDICATIVE.....	114
WEBOGRAPHIE INDICATIVE	115
TABLE DES MATIERES	116