

Formation BUT Informatique

RAPPORT DE STAGE

3^{ème} année – 2024/2025

Stage réalisé du 24 Mars au 29 Juin 2025

Thème :

*Conception et réalisation d'une application mobile
pour la gestion des Projets BTPH.*

Cas d'étude : Entreprise Construb-Est – Direction de Bejaia.

Réalisé par : Yanis KRIM

Maitre du Stage

Jean-Christophe DUBACQ

Tuteur pédagogique

Redouane OUZEGGANE

Formation suivie à



99, avenue Jean-Baptiste Clément, 93430,
Villetaneuse.

Stage réalisé à



Boulevard BOUMDAOUI Nacer, Bloc H1 Z.I
- Bejaia

Année Universitaire : 2024-2025

REMERCIEMENTS

Mes remerciements vont en premier lieu à tous les enseignants et enseignantes de mon établissement, à savoir *IUT Villetaneuse*, pour leur accompagnement constant tout au long de ma formation de BUT. Leurs enseignements (Cours, TD et TP), leurs conseils avisés et leur soutien constant ont été essentiels pour acquérir les compétences nécessaires à la réalisation de ce stage.

J'adresse, également, mes plus sincères remerciements à l'ensemble du personnel de Construb-EST (Direction de Bejaïa), en particulier Monsieur le Direction, BENAAKCHA, Mlle BENLAKHAL Melaaz et ZEGHLI Dounia, pour leur accueil chaleureux et leur professionnalisme, qui m'ont permis une intégration rapide dans le monde de travail de cette entreprise et d'assurer les missions qui m'ont été confiées.

Je tiens particulièrement à remercier Monsieur DUBACQ et Mme LESAUNIER d'avoir accepté de juger mon travail, de corriger mon rapport de stage et de constituer les membres de jury de ma soutenance.

Je remercie également mon tuteur pédagogique, M. OUZEGGANE, pour son suivi régulier, sa disponibilité et ses conseils pertinents, qui ont grandement contribué à la réussite de ce stage.

En fin, je tiens à exprimer ma profonde gratitude envers toute personne qui a contribué, du près ou du loin, au bon déroulement de mon stage au sein de l'entreprise Construb-EST Bejaïa.

Yanis KRIM

TABLE DES MATIERE

REMERCIEMENTS	i
TABLE DES MATIERE	ii
LISTE DES FIGURES	vi
LISTE DES TABLEAUX.....	viii
LISTE DES ABREVIATIONS	ix
INTRODUCTION GENERALE.....	1
CHAPITRE I : ORGANISME D'ACCUEIL ET RECUEIL DES BESOINS	4
I.1. Introduction.....	4
I.2. Présentation du Construb-Est - Bejaia.....	4
I.2.1. Historique.....	4
I.2.2. Organigramme.....	4
I.2.3. Service Informatique.....	6
I.2.4. Situation Informatique	6
I.3. Problématique & proposition de solution	7
I.3.1. Problématique	7
I.3.2. Solution Proposée.....	7
I.4. Recueil des besoins (Cahier des charges)	8
I.4.1. Présentation générale	8
I.4.2. Besoins fonctionnels.....	8
I.4.3. Besoins non fonctionnels.....	9
I.5. Processus d'analyse, de conception et de développement	9
I.5.1. UML, XP et UP	9
I.5.2. Description de la démarche	10
I.6. Conclusion	11
CHAPITRE II : ETUDE ET DESCRIPTION DES BESOINS	13

II.1. Introduction	13
II.2. Analyse du contexte.....	13
II.2.1. Identification des acteurs	13
II.2.2. Diagramme de contexte dynamique	14
II.2.3. Messages.....	14
II.3. Cas d'utilisation	16
II.3.1. Identification des cas d'utilisation.....	16
II.3.2. Description textuelle	18
Cas d'utilisation « <i>S'authentifier</i> ».....	18
Cas d'utilisation « <i>Gérer les informations des utilisateurs</i> ».....	19
Cas d'utilisation « <i>Gérer les projets BTPH</i> » (Chef de département).....	21
II.3.2. Diagrammes de cas d'utilisation.....	23
II.4. Diagramme de séquence système.....	27
II.4.1. D.S.S. pour le cas d'utilisation « <i>S'authentifier</i> »	27
II.4.2. D.S.S. pour le cas d'utilisation « <i>Gérer les utilisateurs</i> »	28
II.4.3. D.S.S. pour le cas d'utilisation « <i>Gérer les projets BTPH</i> »	28
II.5. Maquettes de l'IHM	28
II.5.1. L'authentification.....	32
II.5.2. Espace administrateur.....	32
II.5.3. Espace directeur	33
II.5.4. Espace chef-département.....	33
II.6. Conclusion.....	34
CHAPITRE III : ANALYSE ET CONCEPTION.....	36
III.1. Introduction.....	36
III.2. Classes candidates.....	36
III.2.1. Cas d'utilisation « <i>S'authentifier</i> »	37
III.2.2. Cas d'utilisation « <i>Gérer les informations des utilisateurs</i> »	38
III.2.3. Cas d'utilisation « <i>Gérer les droits d'accès</i> »	38
III.2.4. Cas d'utilisation « <i>Visualiser le tableau de bord</i> »	38

III.3. Diagramme de séquence détaillé (d'interaction).....	39
III.3.1. D.S.D. du cas d'utilisation « S'authentifier ».....	40
III.3.2. D.S.D. du cas d'utilisation « Gérer les utilisateurs ».....	41
III.4. Diagramme de classe de domaine.....	43
II.4.1. Diagramme de classe	43
III.4.2. Dictionnaire de données : Classes, attributs et responsabilité	43
III.6. Schéma relationnel.....	49
III.7. Conclusion	50
CHAPITRE IV : REALISATION ET TESTS.....	52
IV.1. Introduction.....	52
IV.2. Langages et outils de programmation	52
IV.2.1. Visual Studio Code.....	52
IV.2.2. PostgreSQL.....	52
IV.2.3. PgAdmin.....	52
IV.2.4. Javascript	53
IV.2.5. Node JS	53
IV.2.6. Express JS	53
IV.2.7. React JS	53
IV.2.8. Cordova	53
IV.3. Schéma physique de la base de données.....	53
IV.4. Architecture de la solution.....	55
IV.4.1. Diagramme de déploiement.....	55
IV.4.2. Eléments du diagramme de déploiement.....	56
a- Base de données	56
b- Serveur Backend	56
c- Serveur Frontend	56
d- Application web.....	56
e- Application mobile	56
IV.4.3. Structure de code source.....	57

a- Backend	57
b- Frontend.....	57
IV.5. Tests	59
IV.6. Conclusion	59
CONCLUSION GENERALE	61
BIBLIOGRAPHIE.....	64
ANNEXES	66
Annexe A – Fiche technique de l’entreprise Construb-Est.....	66
Annexe B – Organigramme de Construb-Est – Bejaia	70
Annexe C – SQL de la base de données ce_btph_db	71

LISTE DES FIGURES

Figure I. 1 – Organigramme de Construb-EST – Bejaia.....	5
Figure I. 2 – Phases de capture, analyse et conception (Démarche suivie) [01].....	11
Figure II. 1 – Relation entre les acteurs du système.....	14
Figure II. 2 – Diagramme de Contexte.....	15
Figure II. 3 – Diagramme de cas d'utilisation pour l'acteur « Utilisateur »	23
Figure II. 4 – Diagramme de cas d'utilisation pour l'acteur « Administrateur ».....	23
Figure II. 5 – Diagramme de cas d'utilisation pour l'acteur « Directeur ».....	24
Figure II. 6 – Diagramme de cas d'utilisation pour l'acteur « Chef de département »	25
Figure II. 7 – Diagramme de cas d'utilisation pour l'acteur « Chef de projet »	26
Figure II. 8 – Diagramme de séquence système du cas d'utilisation « S'authentifier ».....	27
Figure II. 10 – Diagramme de séquence système du cas d'utilisation « Gérer les utilisateurs » (Partie 1).....	29
Figure II. 10 – Diagramme de séquence système du cas d'utilisation « Gérer les utilisateurs » (Partie 2).....	29
Figure II. 11 – Diagramme de séquence système du cas d'utilisation « Gérer les projets BTPH ».....	30
Figure II. 12 – Diagramme de séquence système du cas d'utilisation « Planifier un projet BTPH ».....	31
Figure II. 13 – Maquette : authentification.....	32
Figure II. 14 – Maquettes : espace administrateur.....	32
Figure II. 15 – Maquettes : espace administrateur.....	33
Figure II. 16 – Maquette : espace chef de département.....	33
Figure III. 1 – Diagramme de classe candidates « S'authentifier ».....	37
Figure III. 2 – Classes candidates « Gérer les informations des utilisateurs »	38
Figure III. 3 – Classes candidates « Gérer les droits d'accès ».....	38
Figure III. 4 – Classes candidates « Visualiser le tableau de bord »	38
Figure III. 5 – D.S.D. du cas d'utilisation « S'authentifier ».....	40
Figure III. 6 – D.S.D. du cas d'utilisation « Gérer les utilisateur »	42

Figure III. 7 – <i>Diagramme de classe du domaine</i>	44
Figure IV. 1 – <i>Schéma physique de la base de données (ce_btph_db)</i>	54
Figure IV. 2 – <i>Diagramme de déploiement</i>	55
Figure IV. 3 – <i>Structure du code source - Backend</i>	57
Figure IV. 4 – <i>Structure du code source - Frontend</i>	57
Figure IV. 5 – <i>Exemples d'interfaces de l'Application WEB</i>	58
Figure IV. 6 – <i>Interface d'accueil pour l'acteur Administrateur</i>	58

LISTE DES TABLEAUX

Tableau II. 1 – Acteurs de l’application « Suivie-Chantiers » et leur rôle.....	14
Tableau II. 2 – Messages entre acteurs et système.....	16
Tableau II. 3 – Différents cas d’utilisation par acteur.	18
Tableau II. 4 – Description textuelle du cas d’utilisation « S’authentifier ».....	19
Tableau II. 5 – Description textuelle du cas d’utilisation « Gérer les utilisateurs »....	21
Tableau II. 6 – Description textuelle du cas d’utilisation « Gérer les projets BTPH ».	22
Tableau III. 1 – Les stéréotypes du Diagramme de séquence d’interaction « S’authentifier ».....	41
Tableau III. 2 – Les stéréotypes du Diagramme de séquence d’interaction « Gérer les utilisateur ».....	41
Tableau III. 3 – Dictionnaire des données du schéma conceptuel (Diagramme du classe du domaine).....	48

LISTE DES

ABREVIATIONS

BTPH	Bâtiment et Travaux Publics et Hydrauliques
CSS	Cascade Style Sheet
D.S.D.	Diagramme de Séquence Détailée
D.S.S.	Diagramme de Séquence Système
DOM	Document Object Model
HTML	HyperText Markup Language
JS	JavaScript
JSON	JavaScript Object Notation
JSX	JavaScript XML
SD	Sequence Diagram
SD	Sequence Diagram
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
T.I.C.	Technologies d'Information et de Communication
UI	User Interface
UML	Unified Modeling Language
UP	Unified Process
UX	User eXperience
XP	eXtrem Programming

INTRODUCTION

GENERALE

INTRODUCTION GÉNÉRALE

Le développement rapide des technologies de l'information et de la communication (TIC) a profondément modifié les méthodes de travail des entreprises et des institutions, notamment dans le secteur du BTPH (Bâtiment, Travaux Publics et Hydraulique). Dans un contexte marqué par la concurrence, la pression sur les délais, et la nécessité d'optimiser les ressources, la digitalisation des processus métier devient un levier incontournable pour garantir l'efficacité, la traçabilité et le bon pilotage des projets.

C'est dans ce cadre que s'inscrit mon stage de fin de cycle, réalisé au sein de Construb-EST Bejaïa, une entreprise active dans le domaine du BTPH. Ce stage, d'une durée de trois mois (du 18 mars au 29 juin 2025), a représenté une opportunité concrète d'appliquer mes compétences en développement informatique tout en répondant à une problématique réelle et actuelle rencontrée par l'entreprise.

L'entreprise fait face à une difficulté majeure dans le suivi quotidien et hebdomadaire des projets et chantiers, en particulier en ce qui concerne la gestion des tâches, l'évaluation de l'avancement, la justification des retards, et la centralisation des données de suivi. Ces processus sont souvent réalisés de manière manuelle ou fragmentaire, ce qui engendre des pertes de temps, un manque de visibilité globale, et un risque accru d'erreurs ou d'omissions.

Face à cette problématique, l'objectif principal de ce travail est de concevoir et développer une solution logicielle permettant :

- La gestion centralisée des projets, chantiers et tâches ;
- Le suivi quotidien et hebdomadaire des activités ;
- La remontée d'informations terrain (problèmes rencontrés, avancement, retards justifiés) ;
- Une hiérarchisation claire des rôles (administrateur, directeur, chef de département, chef de projet) avec des accès différenciés ;

- Un tableau de bord dynamique pour chaque acteur selon ses responsabilités.

Ainsi, la solution envisagée repose sur une application mobile, dotée d'une interface moderne en thème clair/sombre, développée avec les technologies Node JS et React, et intégrant plusieurs acteurs. Cette application permettra d'améliorer significativement la gestion des projets et le suivi des chantiers, tout en offrant une meilleure traçabilité, une visualisation synthétique de l'état d'avancement, et une communication plus fluide entre les différents intervenants.

Pour mieux présenter notre travail, ce rapport est structuré en quatre chapitres principaux, à savoir :

- Le chapitre I est consacré à la présentation de l'entreprise Construb-EST Bejaïa, à la formulation de la problématique, ainsi qu'à l'expression des besoins fonctionnels et non fonctionnels de la solution à concevoir.
- Le chapitre II présente l'étude détaillée des besoins, en identifiant les acteurs du système et en modélisant les interactions par des diagrammes dynamiques.
- Le chapitre III traite de l'analyse fonctionnelle et de la conception de la solution à l'aide des outils de modélisation UML et des méthodes de développement adaptées.
- Le chapitre IV décrit la phase de réalisation du projet ainsi que les tests de validation, en mettant en évidence les choix techniques, les outils utilisés et les résultats obtenus.

Enfin, le rapport se conclut par une conclusion générale, mettant en lumière les enseignements tirés de cette expérience, ainsi que des perspectives d'amélioration pour la suite du projet.

CHAPITRE I

**ORGANISME D'ACCUEIL ET
RECUEIL DES BESOINS**

CHAPITRE I : ORGANISME D'ACCUEIL ET RECUEIL DES BESOINS

I.1. Introduction

Dans ce premier chapitre nous allons présenter l'organisme d'accueil, à savoir : Construb-Est Bejaia, son historique et son domaine d'activité. L'accent sera mis, par la suite, sur la gestion des différents chantiers de construction et leurs importance pour la survie de l'entreprise. A base de ça, nous dégageons une problématique lié à la gestion et au suivi des projets BTP, et nous proposons par la suite une solution informatique qui répond à un certain nombre de besoins fonctionnels et techniques (non-fonctionnels).

I.2. Présentation du Construb-Est - Bejaia

Dans cette section, nous allons présenter l'entreprise Construb-Est – Bejaia, son historique et organigramme.

I.2.1. Historique

La société de Construction et du Bâtiment de l'EST (Construb-Est) Ex EPBTP/ Bejaia a été créée par décret numéro 51.236 du 29 Aout 1981.

CONSTRUB- EST Bejaia est une Enterprise de construction générale. Elle opère dans le domaine du BTPH (Bâtiment, Travaux Publics et Hydraulique). Elle a structuré ses activités à la réalisation de projet de construction, assistance à la maîtrise d'ouvrage et maîtrise d'œuvre, la réalisation de travaux de bâtiments et de génie civil (TCE), la réalisation de charpente métallique, tuyauterie, chaudronnerie, installations industrielles.

Pour plus d'informations, voir [l'annexe A](#) (Fiche technique de l'entreprise Construb-Est).

I.2.2. Organigramme

La figure suivante illustre l'organigramme de la direction régionale de Construb-Est – Bejaia. (Pour le document original, voir [l'annexe B](#)).

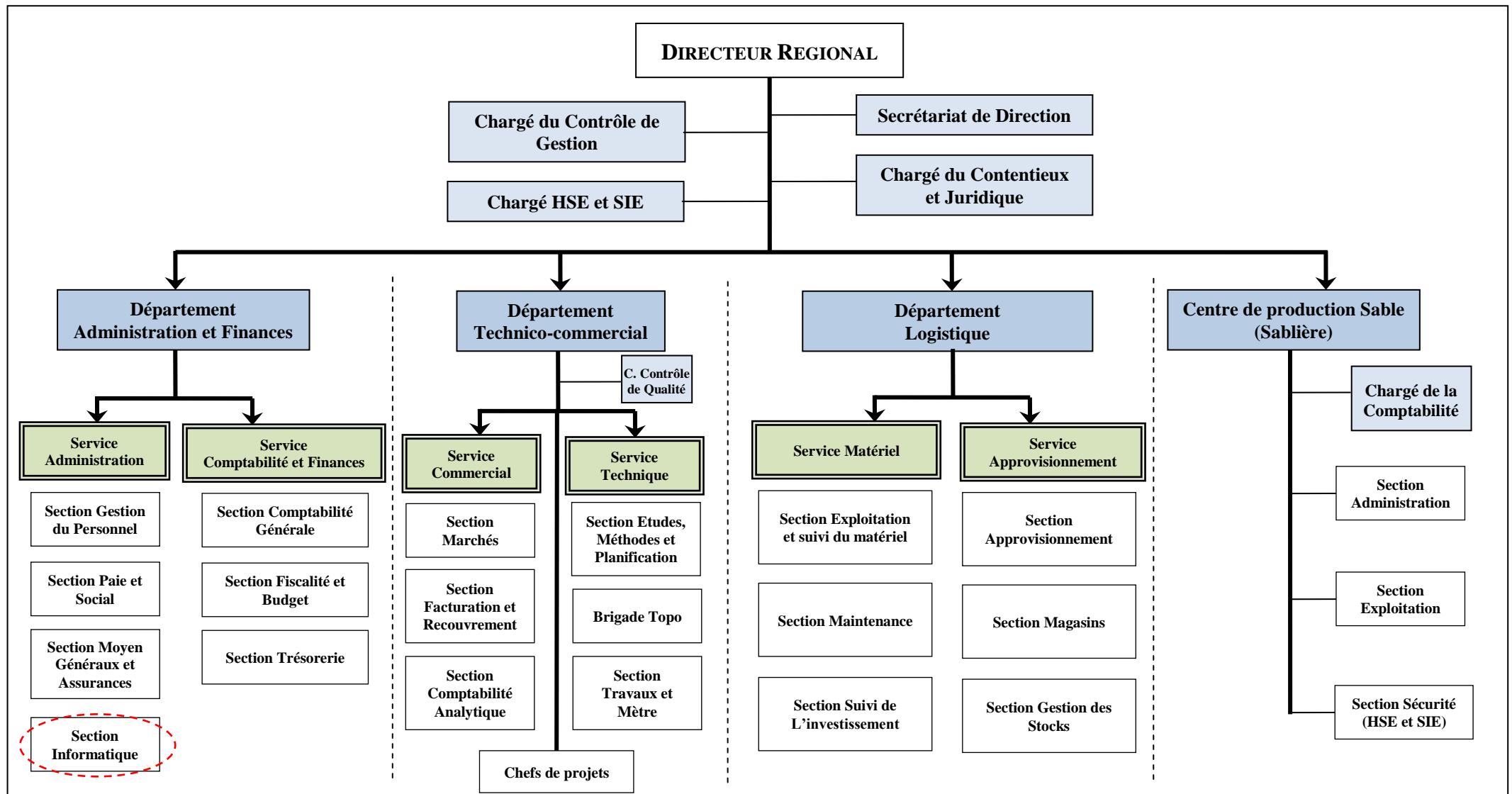


Figure I. 1 – Organigramme de Construb-EST – Bejaia.

I.2.3. Service Informatique

Comme indiqué dans la [*figure I.1*](#), le service d'accueil est la **section informatique**, qui est une petite cellule constituée de trois ingénieurs, leurs rôles respectifs :

- Deux ingénieurs s'occupent du suivi de la mise en œuvre de l'ERP OODOO, du pointage des travailleurs, du support aux autres services (régler les problèmes techniques : imprimantes, réseaux, ...)
- Un ingénieur s'occupe de l'analyse de données récolté des différents projets BTPH.

Notre stage a été réalisé au niveau de ce service, et plus précisément, il a été conduit par M.OUZEGGANE et les deux ingénieurs travaillant sur le système d'information de l'entreprise.

I.2.4. Situation Informatique

Pour l'entreprise Construb-EST, nous présentons la situation informatique de quelques services/sections qui sont au niveau de la Direction Régionale de Bejaia.

- Direction : Il y a deux postes de travail, le premier pour le Directeur et le second pour la secrétariat.
- Une machine serveur avec un système d'exploitation Windows Server 2010.
- **Service administration**
 - Chaque session possède au moins deux ordinateurs.
- **Service comptabilité et finances**
 - Section « Comptabilité Générale » : Trois postes de travail équipés d'ordinateurs
 - Section « Fiscalité et Budget » : trois ordinateurs et un PC portable.
- **Service commercial**
 - Chaque section possède un poste de travail équipé d'un ordinateur.
- **Service technique**
 - Chaque section possède deux à trois ordinateurs.

Tous ces ordinateurs sont reliés à un réseau local câblé, avec possibilité d'accès à l'internet. En plus de ça, il y a un réseau sans-fil (Wifi) pour quelques postes de travail.

I.3. Problématique & proposition de solution

Dans cette section nous présenterons la problématique dégagée lors de notre stage, ainsi qu'une description de la solution proposée.

I.3.1. Problématique

L'importante, voire l'unique, source d'argent de l'entreprise Construb-Est Bejaia est la réalisation des projets de construction et des travaux publics. D'où, une gestion et un suivi efficace de ces projets sont importants, cependant, et après l'étude de l'existant que nous avons réalisé, nous avons constaté qu'il n'y a pas d'outils informatiques permettant, au moins, un suivi des travaux réalisés dans les différents projets BTPH (Bâtiment, Travaux Publics et Hydraulique). La gestion actuelle consiste à communiquer des informations via :

- courriers papiers
- appels téléphoniques
- rarement par mail

Ceci engendre beaucoup de problèmes comme :

- erreurs d'information
- information papier difficile à exploiter
- difficulté à suivre efficacement les différents chantiers
- les décideurs n'ont d'informations exactes sur le travail réalisé par les chefs de projets

I.3.2. Solution Proposée

cités dans la section précédente, nous avons opté à l'élaboration d'une architecture logicielle extensible, dans laquelle nous allons concevoir une application mobile "*Suivi-Projets-BTPH*" qui permet aux différentes parties prenantes (Directeur, chef de département et chefs de projet) de communiquer efficacement sur tout ce qui concerne l'avancement des travaux de tout chantier de construction. Le choix de l'architecture doit permettre aux futurs concepteurs/développeurs d'ajouter d'autres types d'applications (Web et/ou Desktop) ou bien d'ajouter d'autres aussi des applications mobiles pour prendre en charge différents aspects : stock, comptabilités, ...

I.4. Recueil des besoins (Cahier des charges)

I.4.1. Présentation générale

Nom du projet : Suivi-Projets-BTPH

Client : Entreprise Construb-Est Bejaïa

Objectif : Permettre un suivi numérique, centralisé et efficace de l'état d'avancement des différents chantiers de construction.

I.4.2. Besoins fonctionnels

F1. Authentification et gestion des utilisateurs

F1.1. Chaque utilisateur (Directeur, Chef de département, Chef de projet) doit pouvoir se connecter via un identifiant et un mot de passe.

F1.2. Le système doit permettre une gestion des rôles et des permissions selon les profils.

F2. Gestion des projets

F2.1. Affichage de la liste des projets en cours.

F2.2. Possibilité d'ajouter, modifier ou clôturer un projet (selon les permissions).

F2.3. Consultation des détails d'un projet (localisation, budget, échéances, intervenants, etc.).

F3. Suivi de l'avancement des travaux

F3.1. Les chefs de projet doivent pouvoir renseigner l'état d'avancement quotidien ou hebdomadaire des travaux.

F3.2. Ajout de commentaires, photos, documents liés à l'avancement.

F3.3. Suivi des jalons et des tâches prévues vs réalisées.

F4. Communication et notifications

F4.1. Système de messagerie ou de commentaires entre les différents utilisateurs liés à un projet.

F4.2. Notifications automatiques en cas de nouvelles mises à jour, retards, ou anomalies.

F5. Rapports et tableaux de bord

F5.1. Génération de rapports d'avancement des projets.

F5.2. Tableau de bord synthétique pour le directeur (vue globale sur tous les projets).

F5.3. Vue filtrable par département, par statut ou par période.

I.4.3. Besoins non fonctionnels

NF1. Extensibilité

- Le système doit être conçu de manière modulaire pour faciliter l'ajout de nouvelles fonctionnalités (par exemple : gestion du stock, comptabilité, etc.).
- L'architecture doit permettre l'intégration future d'applications Web et Desktop.

NF2. Portabilité

- L'application mobile doit être compatible avec Android (iOS en option ou future évolution).

NF3. Sécurité

- Les données doivent être chiffrées.
- Authentification sécurisée (avec possibilité d'intégrer plus tard une authentification à deux facteurs).
- Gestion des accès et des permissions.

NF4. Performance

- L'application doit être réactive même en cas de mauvaise connexion (possibilité de synchronisation différée).
- Temps de réponse < 2 secondes pour les actions critiques (consultation, enregistrement).

NF5. Facilité d'utilisation (UX)

- Interface simple, ergonomique et adaptée aux utilisateurs non techniques.
- Navigation intuitive pour minimiser la formation nécessaire.

NF6. Sauvegarde et résilience

- Les données doivent être automatiquement sauvegardées sur un serveur central.
- Système de synchronisation locale/serveur en cas de coupure réseau.

I.5. Processus d'analyse, de conception et de développement

I.5.1. UML, XP et UP

En ce qui concerne le formalisme et l'enchaînement des étapes d'analyse et de conception que nous allons adopter à partir du prochain chapitre, nous nous sommes basé

sur le langage de modélisation UML (Unified Modeling Language), en se focalisant sur les diagrammes suivants :

- Diagramme de communication
- Diagramme de cas d'utilisation
- Diagramme de séquence
- Diagramme d'activité
- Diagramme de classe

Et concernant la démarche suivie, elle se situe à mi-chemin entre XP (eXtrem Programming) et UP (Unified Process) décrite dans le *chapitre 09* du [\[01\]](#). Cette démarche possède les caractéristiques suivantes : [\[01\]](#)

- conduite par les cas d'utilisation, comme UP, mais bien plus simple ;
- relativement légère et restreinte, comme XP, mais sans négliger les activités de modélisation en analyse et conception ;
- fondée sur l'utilisation d'un sous ensemble nécessaire et suffisant du langage UML : modéliser 80% des problèmes en utilisant 20% d'UML

I.5.2. Description de la démarche

La démarche que nous allons suivre est basée sur les phases suivantes :

Phases 01 - Identification des besoins : Dans cette phase, nous identifions les futurs acteurs de notre solution, le diagramme de contexte (communication entre les acteurs et le système), diagramme de cas d'utilisation (modéliser les fonctionnalités), diagrammes de séquence système (décrire les C.U. avec ordre chronologique des messages), Maquettes IHM (Interface Homme-Machine).

Phases 02 – Analyse du domaine : Nous élaborons le diagramme de domaine (classes d'entités candidates), diagramme de classes participantes et diagramme d'activité de navigation

Phases 03 - Conception : Le diagramme d'interaction (pour attribuer les responsabilités aux classes d'analyse, diagramme de classe de conception).

La [figure I.2](#) illustre les trois phases (Identification des besoins, Analyse et Conception)

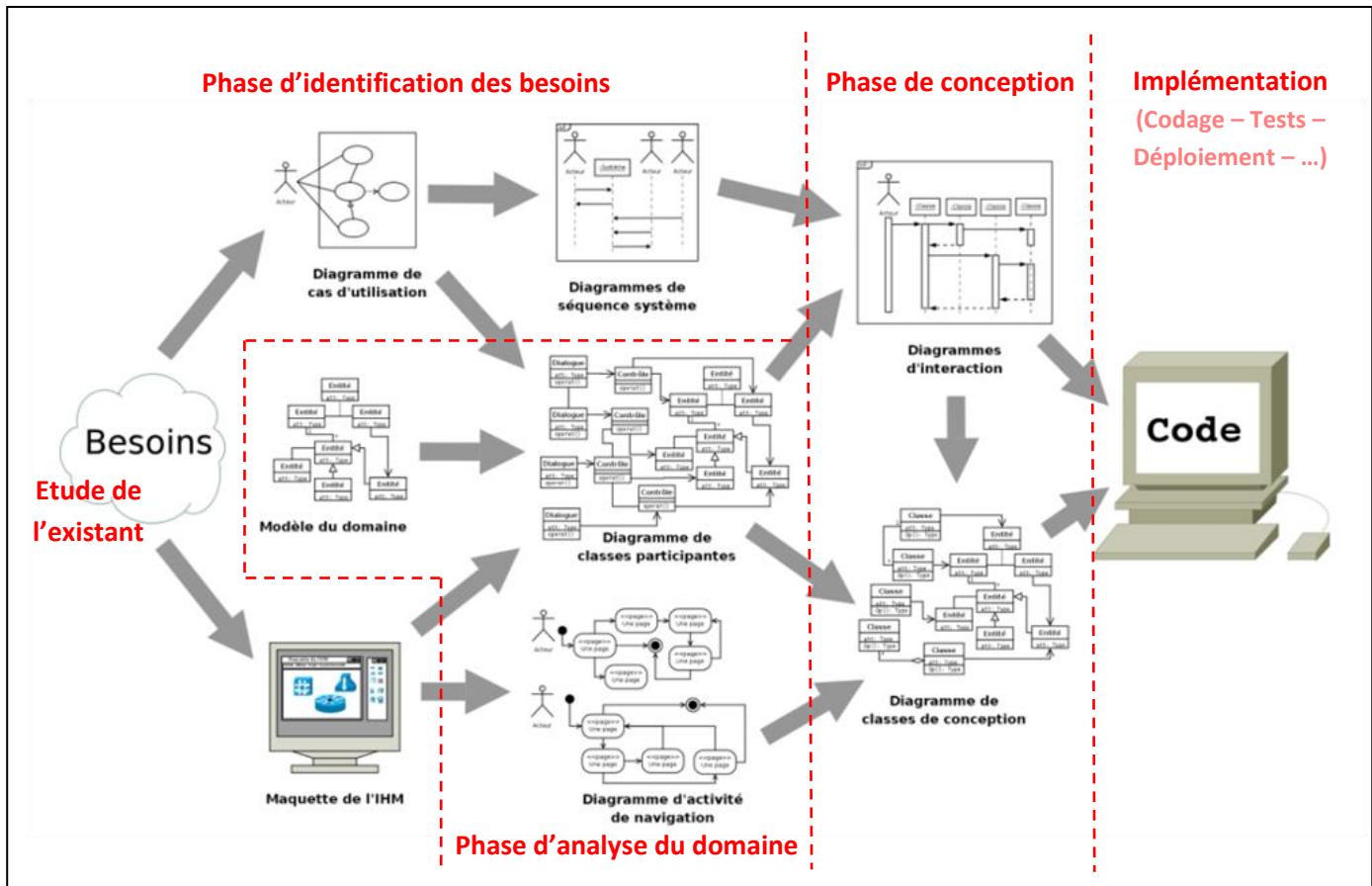


Figure I. 2 – Phases de capture, analyse et conception (Démarche suivie) [01].

I.6. Conclusion

Dans ce premier chapitre nous avons présenté l'historique de la Direction Régionale de Construb-Est Bejaia, son organigramme, la problématique visée ainsi que la solution proposée qui consiste en une application mobile afin de gérer les différents projets de travaux de bâtiments et hydraulique (BTPH). Nous avons conclu ce chapitre par un cahier des charges contenant les besoins fonctionnels et non-fonctionnels (besoins techniques) ainsi qu'une description rapide de la démarche d'analyse et de conception que nous allons adopter. Le chapitre suivant continue avec une identification des acteurs et des cases d'utilisation, leurs descriptions ainsi quelques maquettes de notre solution.

CHAPITRE II

ETUDE ET DESCRIPTION DES BESOINS

CHAPITRE II : ETUDE ET DESCRIPTION DES BESOINS

II.1. Introduction

Dans cette phase de notre stage, nous effectuons l'analyse du contexte dans laquelle nous identifions les différents acteurs de notre application ainsi que leurs interactions avec cette dernière, et ceci afin de réaliser les fonctionnalités attendues qui seront modélisé sous forme de cas d'utilisation. Pour mener à bien cette phase, nous commençons par le diagramme de contexte qui modélise la communication entre les différents acteurs et l'application, les différents cas d'utilisation pour chaque acteur, la description de ces cas d'utilisation en utilisant un formalisme textuel. Pour bien clarifier le résultat attendu par notre client (à savoir l'entreprise Construb-Est), nous avons aussi élaboré des maquettes pour chaque cas d'utilisation.

II.2. Analyse du contexte

Dans cette section, nous étudions le contexte de l'application, en déterminons les différents acteurs de notre solution, et leur interaction avec le système, en détaillant les différents messages entrants et sortants.

II.2.1. Identification des acteurs

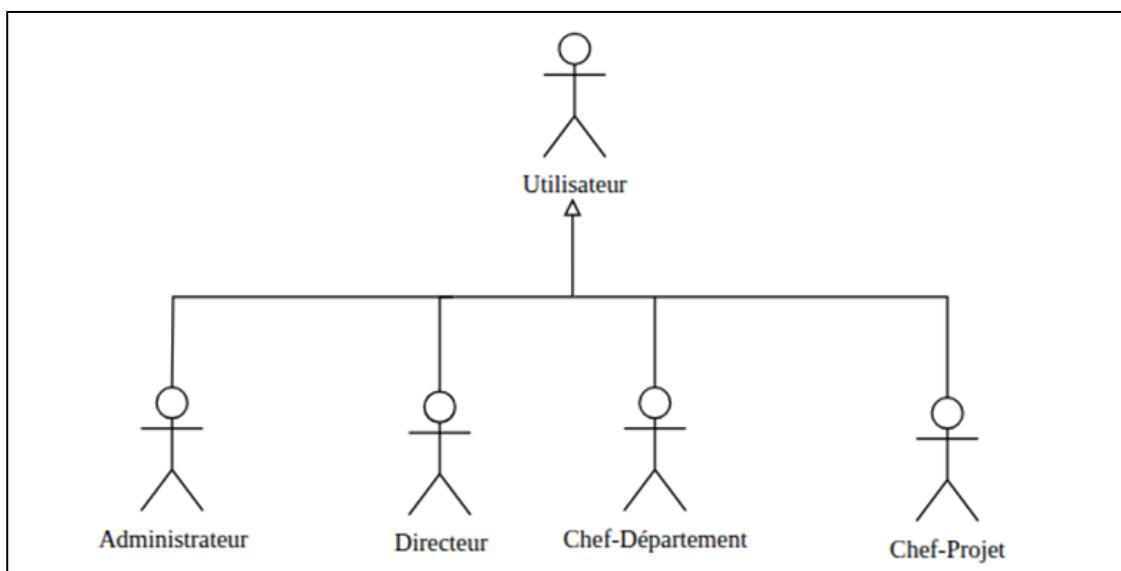
Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et /ou modifier directement l'état du système, en émettant et /ou en recevant des messages susceptibles d'être porteur de données. En ce qui concerne notre système, nous avons pu identifier les acteurs suivants :

Acteur	Rôle
Administrateur	<ul style="list-style-type: none">- Créer les comptes pour les utilisateurs de l'application- Gérer les accès
Directeur	<ul style="list-style-type: none">- Suivre des différents chantiers- Consulter les différents rapports et tableau de bord
Chef de Département	<ul style="list-style-type: none">- Piloter plusieurs chantiers- Coordonner les chefs de projets

Chef de Projet	<ul style="list-style-type: none"> - Suivre l'exécution quotidienne du chantier (tâches, livraisons, respect du planning, ...) - Indiquer les points bloquant (météo, incident sécurité, ...) - Remplir l'état d'avancement journalier et hebdomadaire.
----------------	--

Tableau II. 1 – Acteurs de l'application « Suivie-Chantiers » et leur rôle.

En plus de ces acteurs, nous ajoutons un acteur abstrait (et générique) « *Utilisateur* » (voir page 81 de [02]) qui représente toute entité non encore identifié par le système. Ainsi pour cet acteur, la seule fonctionnalité accessible est l'authentification (identifier et déterminer les droits d'accès). Le diagramme suivant illustre la relation entre les trois acteurs :

**Figure II. 1 – Relation entre les acteurs du système**

II.2.2. Diagramme de contexte dynamique

Dans la [figure II.2](#), l'interaction du futur système avec son environnement extérieur a été illustrée. Plus précisément, cette interaction s'effectuera entre le système, qui est considéré comme une boîte noir, et les différents acteurs identifiés précédemment, en identifiant les différents messages échangés entre chaque acteur et le système.

II.2.3. Messages

Pour expliquer les messages de la [figure II.2](#), le [tableau II.2](#) est présenté à la page suivante.

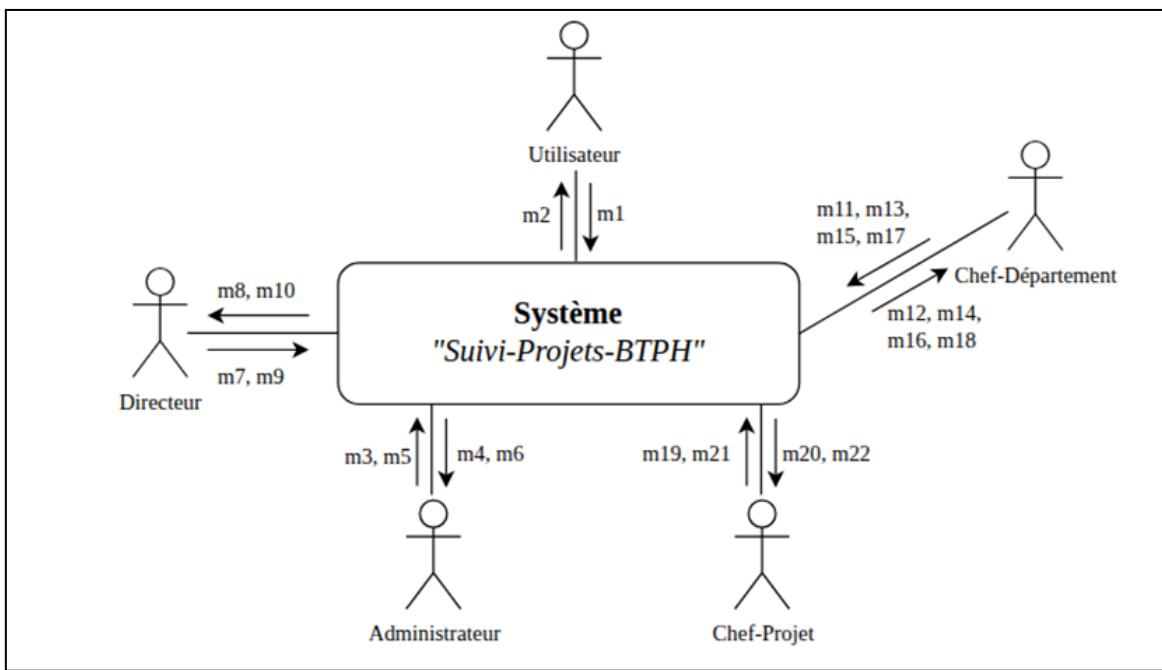


Figure II. 2 – Diagramme de Contexte

Acteur	Message Acteur -> Système		Message Système -> Acteur	
Utilisateur	m1	L'authentification (identifiant et mot de passe)	m2	Réponse du système (soit erreur soit l'affichage de l'interface adéquate selon le rôle de l'utilisateur).
Administrateur	m3	Gérer les utilisateurs : Ajouter, Modifier, Supprimer des utilisateurs	m4	Afficher les interfaces adéquates et exécuter les opérations sélectionnées.
	m5	Gérer les permissions des utilisateurs	m6	Afficher l'interface de gestion de permission
Directeur	m7	Gérer les chefs de département : ajouter, modifier, supprimer	m8	Afficher l'interface pour la gestion des chefs de départements
	m9	Consulter le tableau de bord et l'état d'avancement des projets	m10	Afficher les informations sur un projet : Avancement et budget

Chef-Département	m11	Gérer les chefs de projet : ajouter, modifier, supprimer, ...	M12	Afficher l'interface de gestion des chefs de projets.
	m13	Gérer les projets : ajouter, supprimer, modifier ...	m14	Afficher l'espace de gestion de projet et réaliser les opérations sélectionnées.
	m15	Suivi des projets : état d'avancement, budget, ...	m16	Afficher l'état d'avancement des projets ainsi que la consommation des budgets
	m17	Demander des rapports / statistiques par projet	m18	Générer des dashboards et des rapports par projet
Chef-Projet	m19	Accéder à mes projets réalisés	m20	Afficher la liste et le détail des projets réalisés
	m21	Gérer le projet en cours : indiquer les tâches réalisées, justifier les tâches non-réalisées, remplir l'état d'avancement : par jour, semaine ou mois.	m22	Afficher l'espace de gestion des tâches d'un projet non achevé.

Tableau II. 2 – Messages entre acteurs et système.

II.3. Cas d'utilisation

Cette phase est très importante car ça représente le départ de la phase d'analyse qui se centre sur les cas d'utilisation. Selon [02], Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné.

II.3.1. Identification des cas d'utilisation

Dans ce qui suit, nous allons énumérer les différents cas d'utilisation pour chaque acteur du système. Pour mieux présenter ces cas d'utilisation, nous avons opté pour une structure tabulaire (voir [02] page 65). Le tableau suivant présente les différents cas d'utilisation identifié pour notre système :

Cas d'utilisation	Acteur Principal / Acteurs Secondaires	Messages Emis / Reçus par les acteurs
<i>S'authentifier</i>	Utilisateur	Emet : Information d'identification et le mot de passe. Reçoit : L'interface adéquate à l'utilisateur.

<i>Gérer les informations des utilisateurs</i>	Ajouter	Administrateur	Emet : Ajout, Modification, Suppression et consultation des utilisateurs. Reçoit : Confirmation d'ajout, de modification et de suppression.
	Modifier		
	Supprimer		
	Désactiver		
<i>Gérer les droits d'accès</i>		Administrateur	Emet : Ajout, Modification, Suppression des droits d'accès et associer ces droits aux utilisateurs. Reçoit : Confirmation des actions réalisées.
<i>Visualiser le tableau de bord (Dashboard)</i>	Filtrer par date	Directeur	Emet : Ajout, Modification, Suppression et consultation des propriétaires. Reçoit : Confirmation d'ajout, de modification et de suppression.
	Filtrer par projet		
<i>Gérer les chefs de département</i>	Ajouter	Directeur	Emet : Ajout, Modification, Suppression et consultation des chefs de département. Reçoit : Confirmation d'ajout, de modification et de suppression.
	Modifier		
	Supprimer		
	Consulter		
<i>Exporter des données</i>	PDF	Directeur	Emet : Choisir le type d'exportation. Reçoit : Des données sous format PDF ou Excel.
	Excel		
<i>Gérer les projets BTPH</i>	Ajouter	Chef de Département	Emet : Ajouter, modifier, supprimer et consulter les détails d'un projet. Reçoit : Confirmation des actions choisies.
	Modifier		
	Supprimer		
	Consulter		
<i>Gérer les chefs de projet</i>	Ajouter	Chef de Département	Emet : Ajouter, modifier, supprimer et consulter les informations de chef de département Reçoit : Confirmation des actions choisies.
	Modifier		
	Supprimer		
	Consulter		
<i>Suivi des projets</i>		Chef de Département	Emet : Consulter les tâches réalisées. Reçoit : Etat de réalisation.
<i>Visualiser le tableau de bord</i>		Chef de Département	Emet : Sélectionner les dates, les projets, Reçoit : Etat d'avancement et le budget consommé.
<i>Exporter les données</i>	PDF	Chef de Département	Emet : Choisir le type d'exportation. Reçoit : Des données sous format PDF ou Excel.
	Excel		

<i>Gérer les projets BTPH</i>	Indiquer les tâches réalisées	Chef de projet	Emet : Sélectionner le projet en cours, indiquer les tâches réalisées, celles en retard, Reçoit : Confirmation des actions sélectionnées.
	Indiquer les tâches en retard		
	Indiquer l'état d'avancement		
<i>Consulter l'historique des projets réalisés</i>		Chef de projet	Emet : sélectionner un projet déjà Reçoit : Afficher les informations du projet sélectionné.

Tableau II. 3 – Différents cas d'utilisation par acteur.

II.3.2. Description textuelle

Chaque cas d'utilisation énuméré dans le [tableau II.2](#) sera décrit, textuellement, en suivant d'un formalisme inspiré de ce celui décrit dans [02]. (p. 71). Dans ce rapport, nous allons choisir quelques cas d'utilisation à décrire, comme « S'authentifier », « Gérer les informations des utilisateurs », « Gérer des projets BTPH » pour l'acteur chef de département.

- **Cas d'utilisation « S'authentifier »**

Sommaire d'Identification			
Titre	S'authentifier		
But	Permet d'identifier l'utilisateur et de savoir son rôle dans l'application (Administrateur, Chef de département ou de projet ou Directeur).		
Résumé	Lors de l'accès au système, l'utilisateur doit fournir un identifiant et un mot de passe puis valide le formulaire. Le système utilise l'identifiant et le mot de passe pour vérifier si ces informations sont correctes ou bien non. Dans le cas d'erreur, le système affiche un message et invite l'utilisateur et refaire l'opération, sinon (cas de succès), le système affiche l'interface adéquate pour l'utilisateur.		
Acteur(s)	Utilisateur		
Date de Création	10/04/2025	Date de Mise à Jour	12/04/2025
Version	1.0.0	Responsable	KRIM Yanis
Pré-conditions	/		
Scénario nominal	Ce cas d'utilisation est déclenché lorsqu'un utilisateur veut accéder à l'application. Enchainement (a) – S'identifier par un identifiant et mot de passe		

	<p>L'utilisateur fournit un identifiant (<i>nom d'utilisateur</i>) et un mot de passe secret, et envoie les informations au système afin d'accéder à son espace de travail. Le système effectue des vérifications.</p> <ul style="list-style-type: none"> - Si l'utilisateur a omis l'identification et/ou mot de passe alors déclencher [Exception 01 : Champ(s) vide(s)]. - Si l'identifiant et/ou mot de passe ne sont pas correcte alors il faut exécuter [Exception 02 : Champ(s) incorrect(s)]. - Si les informations fournies sont correcte on affiche l'espace de travail de l'utilisateur identifié.
Enchaînements Alternatifs	<p>Enchainement (b) – Lors du premier accès réussi Lorsqu'un utilisateur est authentifié correctement pour la première fois, l'application lui affiche un formulaire obligeant l'utilisateur à modifier son mot de passe.</p>
Exceptions	
<p>[Exception 01 : Champ(s) vide(s)] : Le système affiche une erreur à l'utilisateur lui indiquant qu'il a oublié son identifiant et/ou son mot de passe, et l'invite à les compléter.</p>	
<p>[Exception 02 : Champ(s) incorrect(s)] : Le système indique à l'utilisateur qu'une erreur est détectée liée à son identifiant et/ou à son mot de passe, il l'invite à resaisir son identifiant et/ou son mot de passe.</p>	

Tableau II. 4 – Description textuelle du cas d'utilisation « S'authentifier ».

- Cas d'utilisation « *Gérer les informations des utilisateurs* »

Sommaire d'Identification			
Titre	Gérer les informations des utilisateurs		
But	Permet à l' <i>administrateur</i> de créer des utilisateurs en leurs affectant un rôle : Directeur, Chef de département et Chef de projet.		
Résumé	L'administrateur peut gérer les utilisateurs soit en : <ul style="list-style-type: none"> - ajoutant un nouvel utilisateur ; - modifiant un utilisateur ; - désactivant un utilisateur ; - supprimant l'utilisateur ; 		
Acteurs	Administrateur		
Date de Création	10/04/2025	Date de Mise à Jour	13/04/2025
Version	1.0.2	Responsable	KRIM Yanis
Pré-conditions	L'administrateur doit être déjà authentifié		

Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsqu'un administrateur veut manipuler les informations liées aux utilisateurs de l'application.</p>
	<p>Enchainement (a) – Ajouter un utilisateur</p> <p>Lorsqu'un nouvel utilisateur de l'application doit être ajouté, l'administrateur réalise les actions suivantes :</p>
	<ul style="list-style-type: none"> - accéder au formulaire d'ajout ; - remplir le formulaire (nom, prénom, identifiant, mot de passe, ...) et valider ces informations - Si une ou plusieurs informations obligatoires sont manquantes, alors déclencher <u>Exception 03 : Champ(s) vide(s)</u>. - Sinon, le système vérifier la validité des informations, s'il y au moins une information invalide déclencher <u>Exception 04 : Champ(s) invalide(s)</u>. - Sinon, l'application crée l'utilisateur et affiche un message de succès pour l'administrateur
	<p>Enchainement (b) – Modifier les informations d'un utilisateur</p> <p>Dans des situations, quelques informations sont à corriger ou à mettre à jour. Dans ce cas l'administrateur sélectionne un utilisateur et :</p>
	<ul style="list-style-type: none"> - accède au formulaire de modification - modifier les informations ciblées (voulues) et soumettre le formulaire - le système effectue une vérification : <ul style="list-style-type: none"> - Si une ou plusieurs informations obligatoires sont manquantes, alors déclencher <u>Exception 03 : Champ(s) vide(s)</u>. - Sinon, le système vérifier la validité des informations, s'il y au moins une information invalide déclencher <u>Exception 04 : Champ(s) invalide(s)</u>. - Sinon, l'application confirme que les informations ont été modifiées avec succès.
	<p>Enchainement (c) – Désactiver un utilisateur</p>
	<p>Pour des raisons de sécurité, si l'utilisateur prend sa retraite, ou il est en maladie ou bien en congé, l'administrateur procède à la désactivation de son compte :</p>
	<ul style="list-style-type: none"> - L'administrateur sélectionne l'utilisateur concerné - Exécuter l'opération « <i>désactiver</i> » - L'application confirme l'opération.
	<p>Enchainement (d) – Supprimer un utilisateur</p>
	<p>L'administrateur peut supprimer un utilisateur (par exemple dans le cas où ce dernier a quitté l'entreprise) :</p>
	<ul style="list-style-type: none"> - L'administrateur sélectionne l'utilisateur concerné par la suppression - Exécuter l'opération « <i>supprimer</i> » - Le système affiche un message de confirmation (Oui / Non) - Si l'utilisateur choisit « <i>Non</i> », l'opération est annulée (pas de

	<ul style="list-style-type: none"> - suppression) - Si l'utilisateur choisit « <i>Oui</i> », l'application réalise l'opération de suppression et affiche une confirmation pour l'administrateur.
Enchaînements Alternatifs	/
Exceptions	
[Exception 03 : Champ(s) vide(s)] :	
Le système affiche une erreur à l'administrateur lui indiquant qu'il a oublié une ou plusieurs informations obligatoires du formulaire d'ajout/modification, et l'invite à les compléter.	
[Exception 04 : Champ(s) invalide(s)] :	
Le système indique à l'administrateur qu'il y a une ou plusieurs informations invalides dans le formulaire d'ajout/modification et invite l'administrateur à les ressaisir.	

Tableau II. 5 – Description textuelle du cas d'utilisation « Gérer les utilisateurs ».

- Cas d'utilisation « *Gérer les projets BTPH* » (Chef de département)

Sommaire d'Identification			
Titre	Gérer les projets BTPH		
But	Permet au <i>chef de département</i> de créer des projets BTPH, de les modifier, de les supprimer ou bien des les consulter.		
Résumé	<p>Le chef de département peut gérer les projets soit en :</p> <ul style="list-style-type: none"> - créant un nouvel utilisateur ; - modifiant des informations d'un projet ; - supprimant un projet ; - consultant le détail de projet ; 		
Acteurs	Chef de département (principal), Chef de projet (secondaire)		
Date de Création	20/04/2025	Date de Mise à Jour	10/05/2025
Version	1.0.0	Responsable	KRIM Yanis
Pré-conditions	Le chef de département doit être déjà authentifié		
Scénario nominal	<p>Ce cas d'utilisation est déclenché lorsqu'un chef de département veut gérer des informations liées aux projets BTPH.</p> <p>Enchainement (a) – Ajouter un projet</p> <p>Lorsqu'un nouveau projet BTPH est planifié par le service technique, le chef de département réalise les actions suivantes, afin de le créer :</p> <ul style="list-style-type: none"> - accéder au formulaire d'ajout du projet; - remplir le formulaire (désignation du projet, type, son adresse, chef de projet, date début, date fin, ...) et valider toutes ses informations - Si une ou plusieurs informations obligatoires sont manquantes, alors déclencher [Exception 05 : Champ(s) vide(s)]. - Sinon, le système vérifier la validité des informations, s'il y au moins une information invalide déclencher [Exception 05 : Champ(s) invalide(s)]. - Sinon, l'application crée le projet et affiche un message de confirmation. 		

	<p>Enchainement (b) – Modifier les informations d'un projet</p> <p>Pour mettre à jour les informations d'un projet, et éventuellement, corriger des erreurs lors de l'enchainement (a), le chef de département réalise les actions suivantes :</p> <ul style="list-style-type: none"> - sélectionner le projet et accède à son formulaire de modification - modifier les informations ciblées et soumettre le formulaire - le système effectue une vérification : <ul style="list-style-type: none"> - Si une ou plusieurs informations obligatoires sont manquantes, alors déclencher [Exception 05 : Champ(s) vide(s)]. - Sinon, le système vérifier la validité des informations, s'il y au moins une information invalide déclencher [Exception 06 : Champ(s) invalide(s)]. - Sinon, l'application confirme la modification du projet BTPH.
	<p>Enchainement (c) – Planifier un projet</p> <p>Pour planifier un projet BTPH, le chef département :</p> <ul style="list-style-type: none"> - sélectionne le projet en question - Créer la liste des tâches du projet - Estimer les dates de réalisation et leurs durée.
	<p>Enchainement (d) – Supprimer un projet BTPH</p> <p>Le chef de département peut supprimer un projet s'il est annulé :</p> <ul style="list-style-type: none"> - sélectionner le projet concerné par la suppression - exécuter l'opération « <i>supprimer</i> » - Le système affiche un message de confirmation (Oui / Non) - Si le chef de département choisit « <i>Non</i> », l'opération de suppression est annulée (pas de suppression) - Si l'utilisateur choisit « <i>Oui</i> », l'application réalise l'opération de suppression et affiche une confirmation pour l'administrateur.
Enchaînements Alternatifs	/
	Exceptions
	<p>[Exception 05 : Champ(s) vide(s)] :</p> <p>Le système affiche une erreur au chef de département lui indiquant qu'il a oublié une ou plusieurs informations obligatoires du formulaire d'ajout/modification de projet BTPH.</p>
	<p>[Exception 06 : Champ(s) invalide(s)] :</p> <p>Le système indique au chef de département qu'il y une ou plusieurs informations invalides dans le formulaire d'ajout/modification et l'invite à les ressaisir.</p>

Tableau II. 6 – Description textuelle du cas d'utilisation « Gérer les projets BTPH ».

II.3.2. Diagrammes de cas d'utilisation

Concernant les diagrammes de cas d'utilisation, ils sont illustrés par les figures suivantes, par acteur :

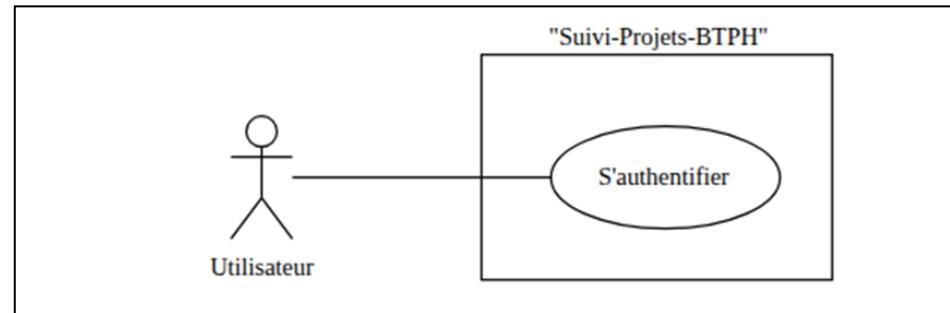


Figure II. 3 – Diagramme de cas d'utilisation pour l'acteur « Utilisateur »

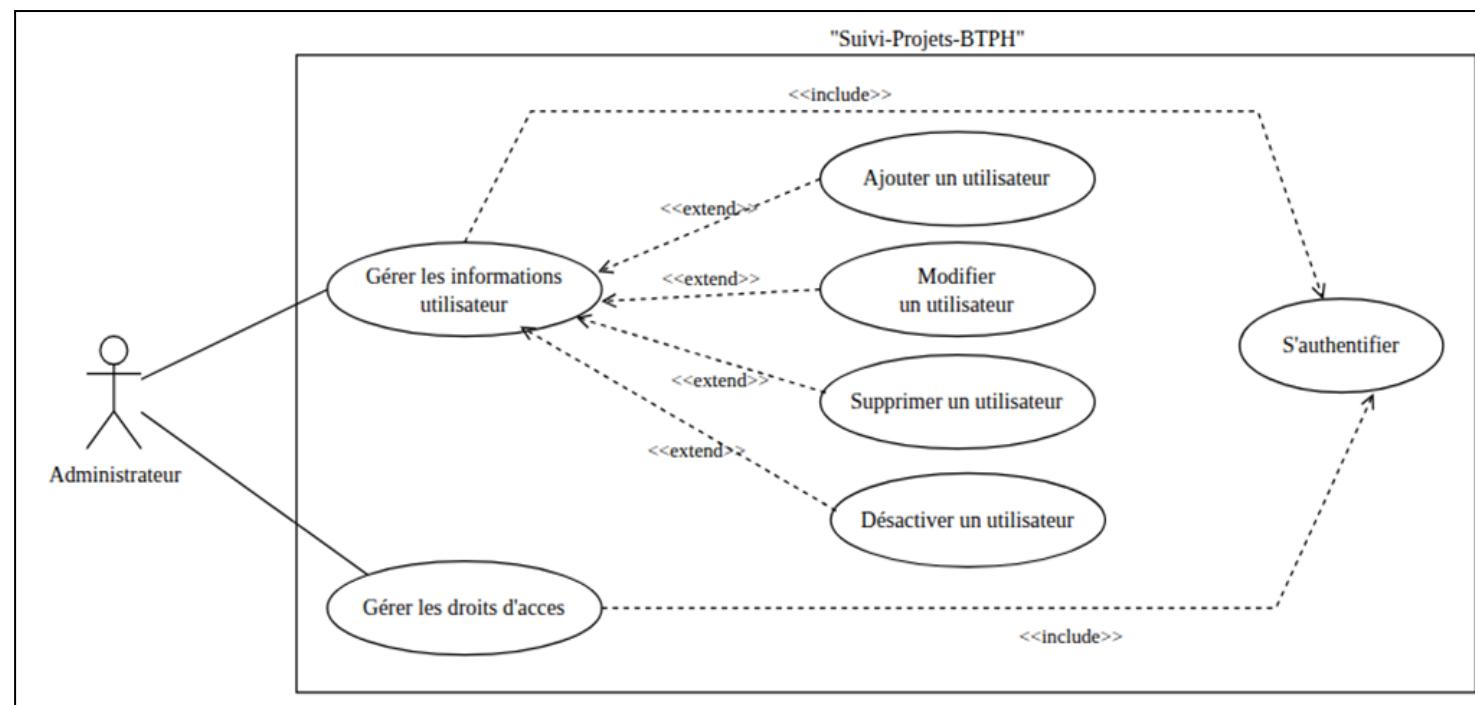


Figure II. 4 – Diagramme de cas d'utilisation pour l'acteur « Administrateur »

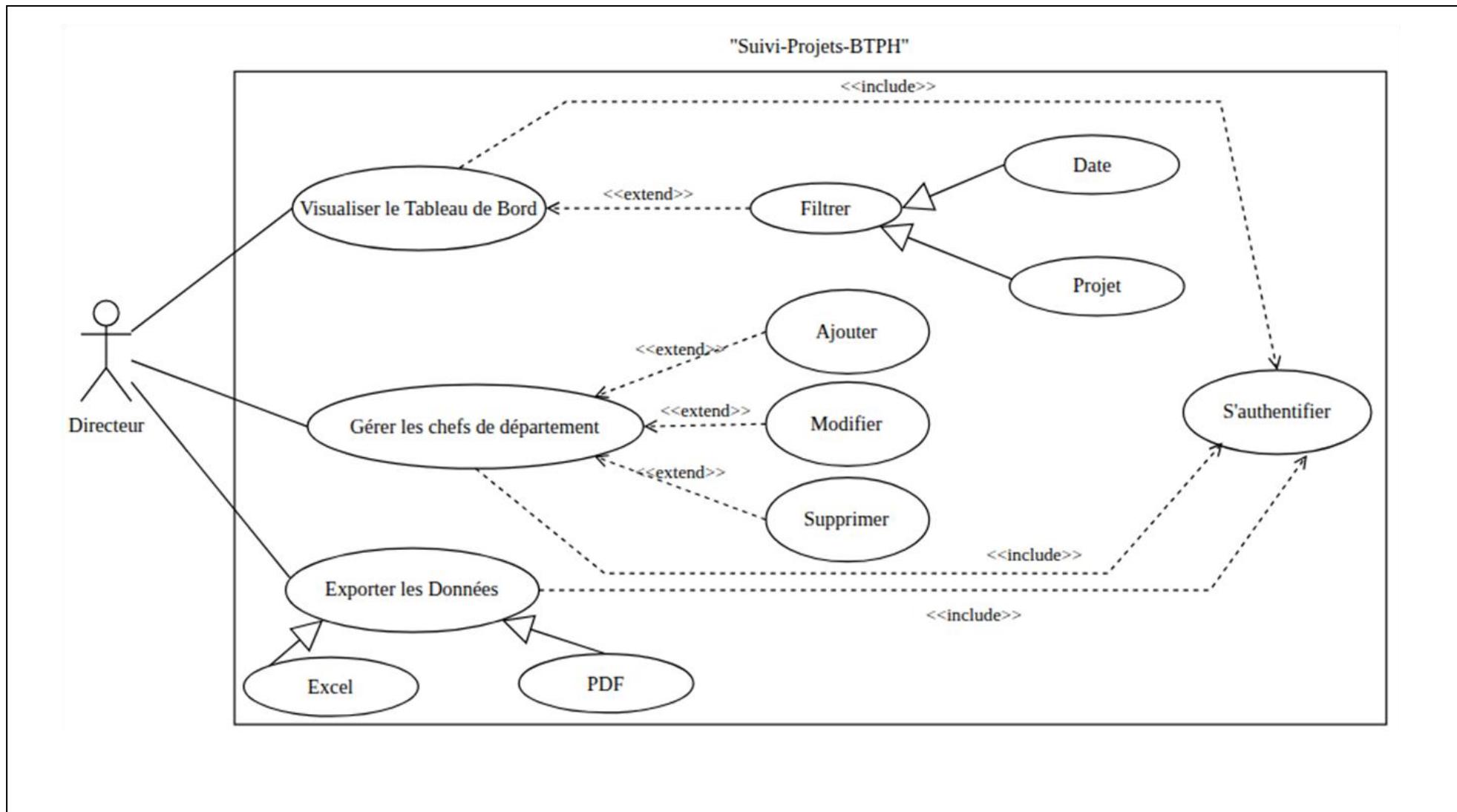


Figure II. 5 – Diagramme de cas d'utilisation pour l'acteur « Directeur »

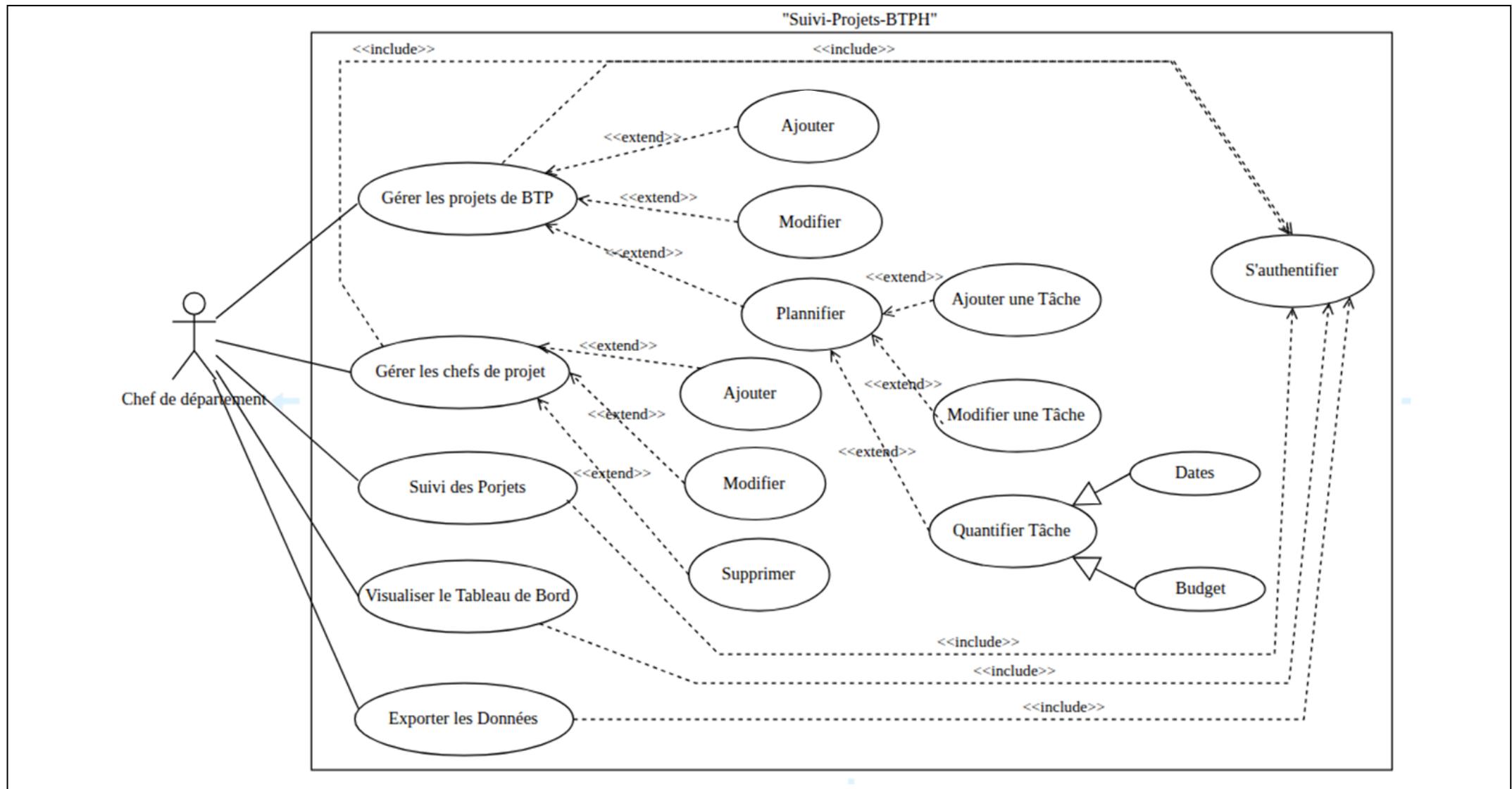


Figure II. 6 – Diagramme de cas d'utilisation pour l'acteur « Chef de département »

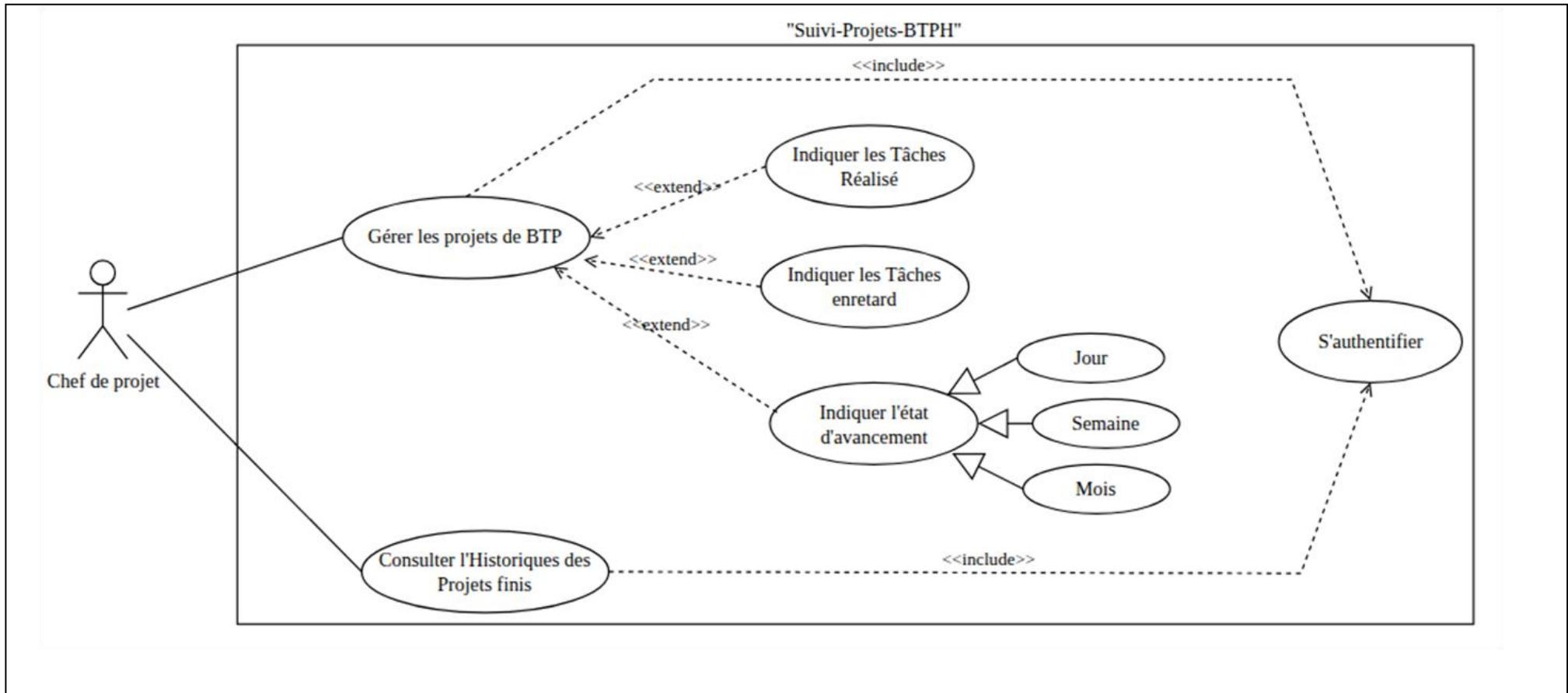


Figure II. 7 – Diagramme de cas d'utilisation pour l'acteur « Chef de projet »

II.4. Diagramme de séquence système

En plus de la description textuelle des cas d'utilisation, nous utilisons les diagrammes de séquence système (D.S.S.) pour illustrer l'ordre chronologique des messages échangés entre les acteurs et le système. Dans cette phase d'identification et d'étude des besoins (analyse des besoins) le système est vu comme une boîte noire.

Ainsi, pour chaque cas d'utilisation, nous élaborons un D.S.S. dans lequel, l'acteur ou les acteurs ainsi que le système sont représentés par une ligne de vie, et l'échange de messages entre acteur(s) et système est représenté par des flèches de communication (messages synchrones et messages asynchrones). Dans ce qui suit, nous présentons des diagrammes de séquence de quelques cas d'utilisation : « S'authentifier », « Gérer les informations des utilisateurs » et « Gérer les projets BTPH » (Chef de département).

II.4.1. D.S.S. pour le cas d'utilisation « S'authentifier »

La figure suivante, nous illustrons le diagramme de séquence du cas d'utilisation « S'authentifier » (sd pour *sequence diagram*) :

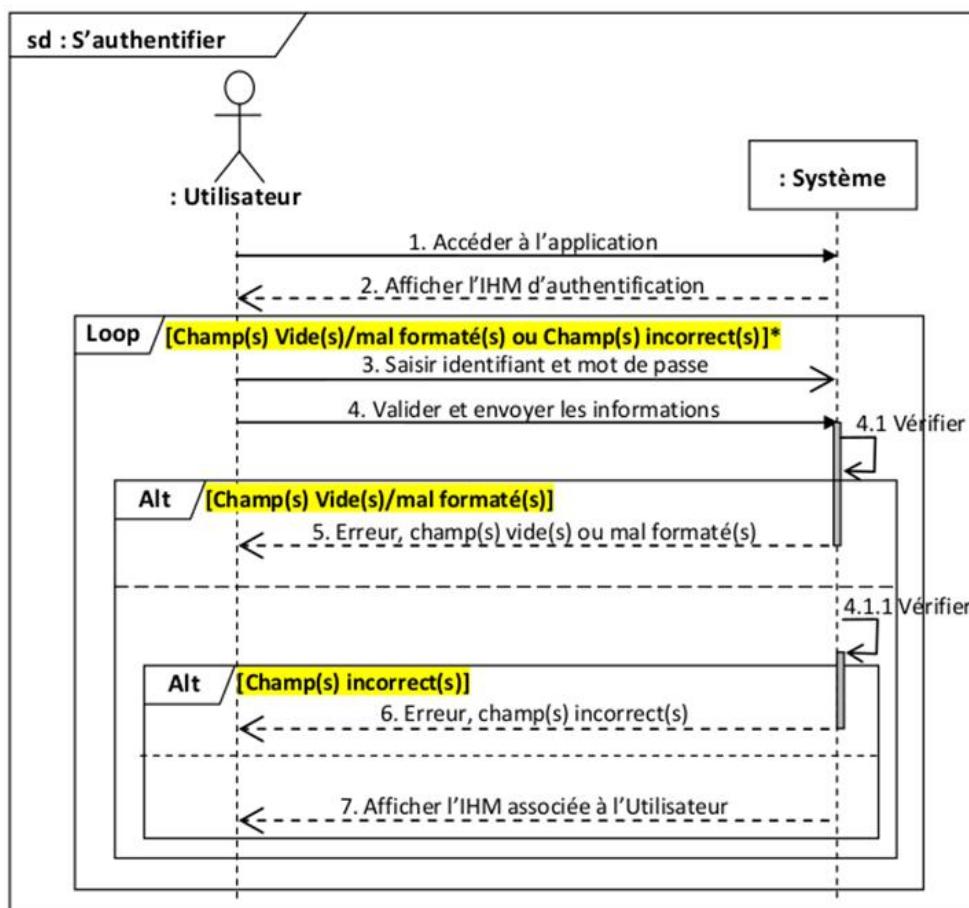


Figure II. 8 – Diagramme de séquence système du cas d'utilisation « S'authentifier »

Les stéréotypes « Loop » et « Alt », respectivement pour les boucles et les alternatifs, permettent de décrire plusieurs scénarios d'un cas d'utilisation. Pour la [figure II.8](#), Pour la boucle « Loop », on refait le (re)saisie du formulaire authentification à chaque fois que les champs sont vides, mal-écrit ou bien invalides (incorrectes). Pour une mesure de sécurité, on peut limiter ce Loop à un nombre réduit de tentatives (4 tentative par heure).

II.4.2. D.S.S. pour le cas d'utilisation « Gérer les utilisateurs »

Les [figure II.9](#) et [figure II.10](#) illustrent le diagramme de séquence du cas d'utilisation « Gérer les informations des utilisateurs ».

II.4.3. D.S.S. pour le cas d'utilisation « Gérer les projets BTPH »

Le dernier D.S.S. élaboré dans ce rapport est illustré dans la [figure II.11](#) (Diagramme de séquence du cas d'utilisation « Gérer les projets BTPH »).

II.5. Maquettes de l'IHM

Afin de mieux visualiser l'interface utilisateur de l'application et de faciliter la communication avec les parties prenantes, des maquettes d'IHM (Interface Homme-Machine) ont été réalisées à ce stade du projet. Une maquette d'IHM est un produit temporaire qui permet d'avoir une vue concrète mais non définitive de l'apparence future de l'application. Ces maquettes permettent à l'utilisateur de se projeter dans l'environnement applicatif, même si les fonctionnalités réelles ne sont pas encore implémentées.

Dans notre cas, nous avons distingué deux types de maquettes complémentaires :

- Les maquettes légères (ou wireframes), qui représentent de manière simple la structure et l'organisation des écrans, sans mise en forme graphique détaillée (*ce que nous avons réalisé*).
- Les maquettes lourdes, qui illustrent l'interface finale avec les couleurs, typographies, icônes et comportements attendus (comme les animations ou transitions). Ce travail de prototypage permet de valider l'ergonomie, l'expérience utilisateur (UX : User eXperience) et la navigation dès les premières phases du développement, tout en récoltant des retours constructifs de la part des utilisateurs finaux.

A partir de la section II.5.1 quelques maquettes sont présenté.

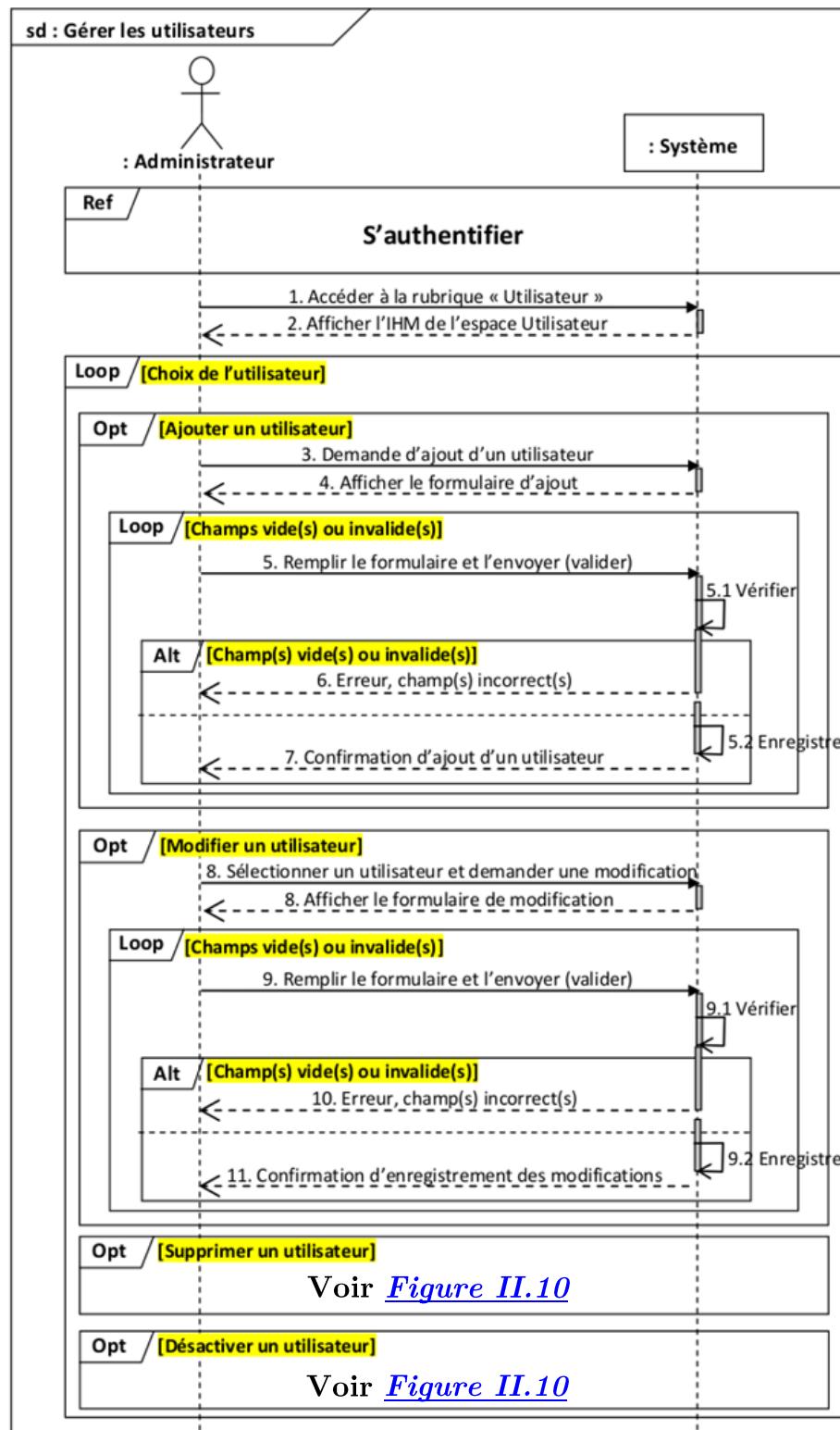


Figure II. 10 – Diagramme de séquence système du cas d'utilisation « Gérer les utilisateurs » (Partie 1)

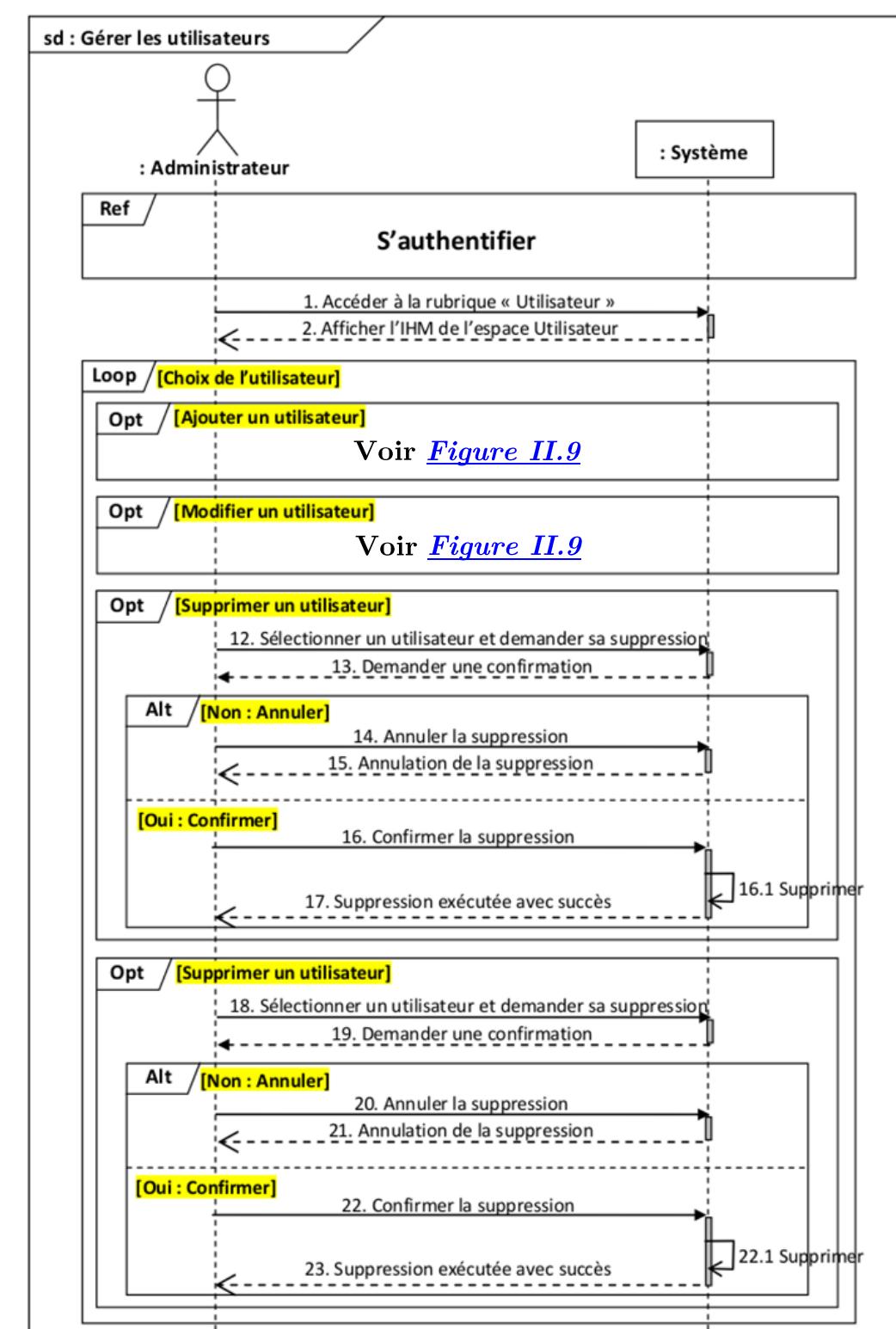


Figure II. 10 – Diagramme de séquence système du cas d'utilisation « Gérer les utilisateurs » (Partie 2)

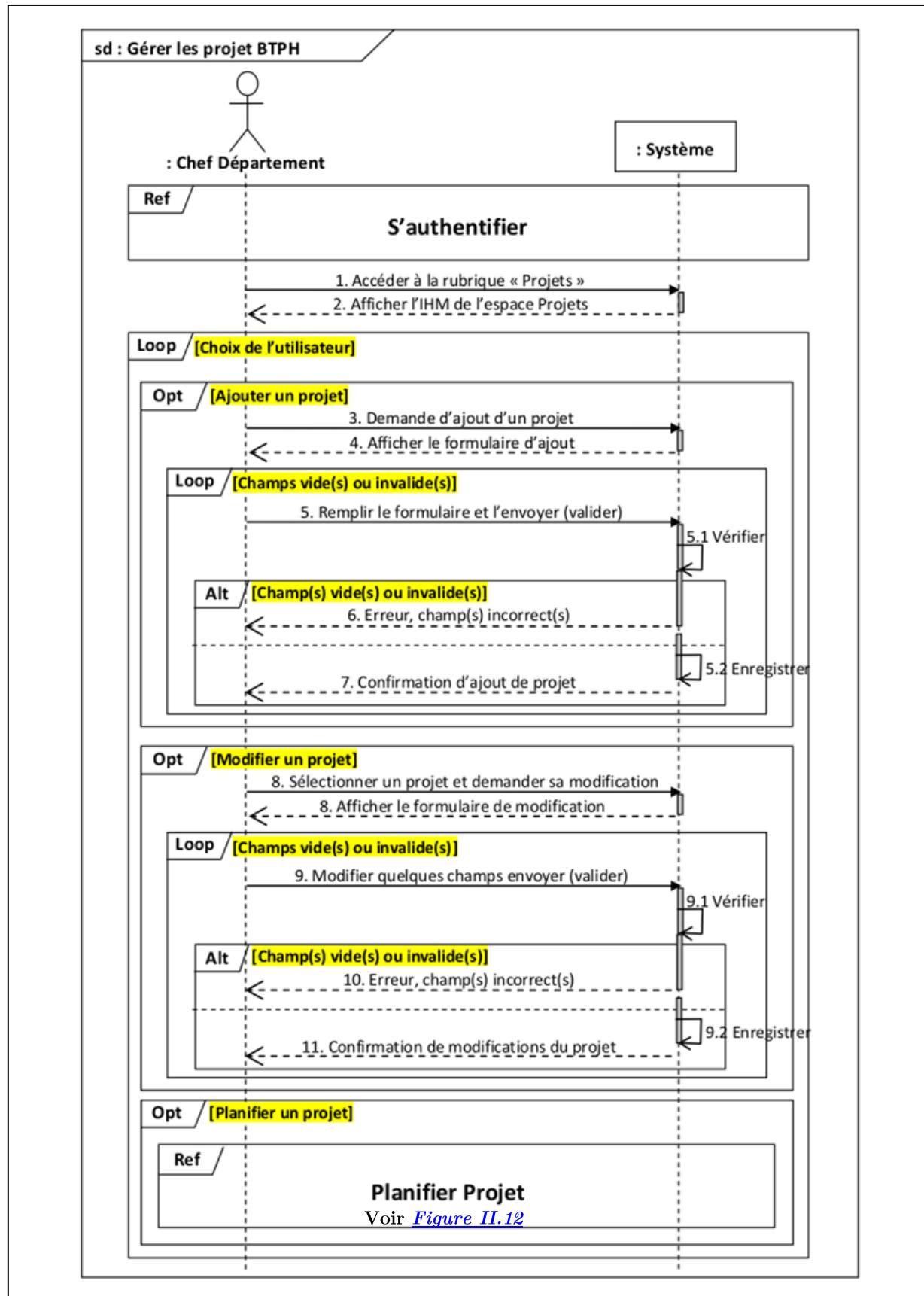


Figure II. 11 – Diagramme de séquence système du cas d'utilisation « Gérer les projets BTPH »

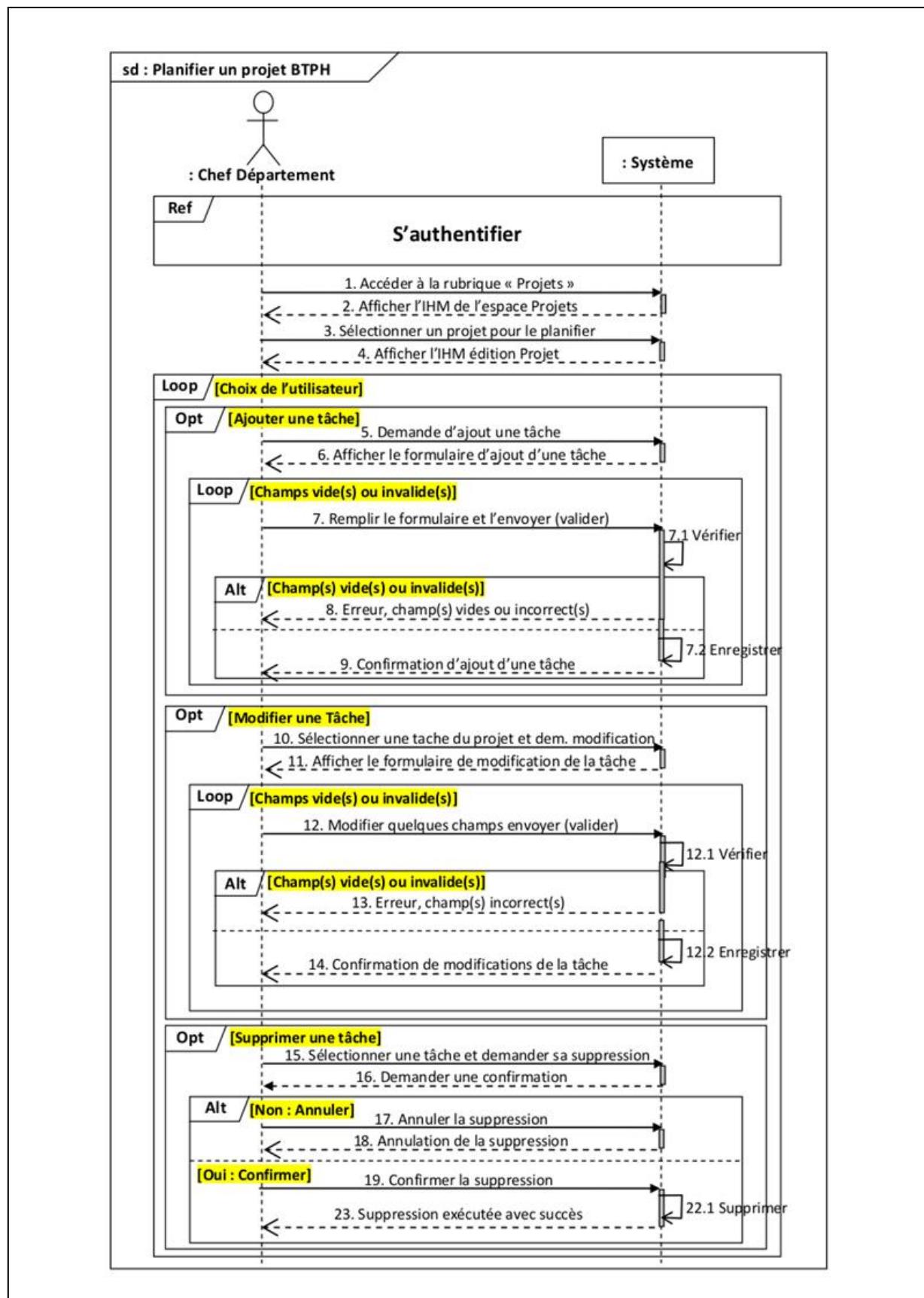


Figure II. 12 – Diagramme de séquence système du cas d'utilisation « Planifier un projet BTPH »

II.5.1. L'authentification

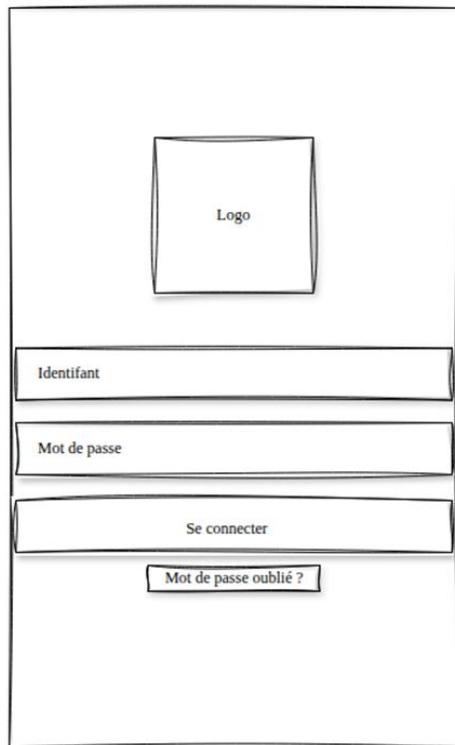


Figure II. 13 – Maquette : authentification.

II.5.2. Espace administrateur

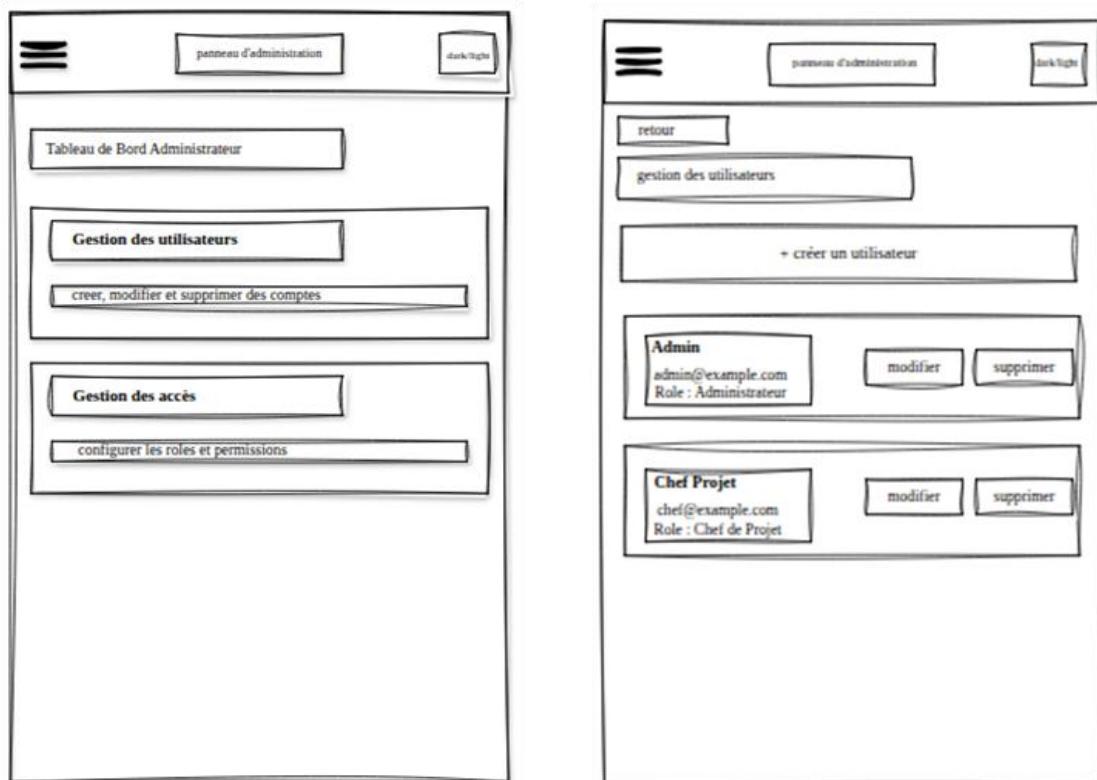


Figure II. 14 – Maquettes : espace administrateur.

II.5.3. Espace directeur

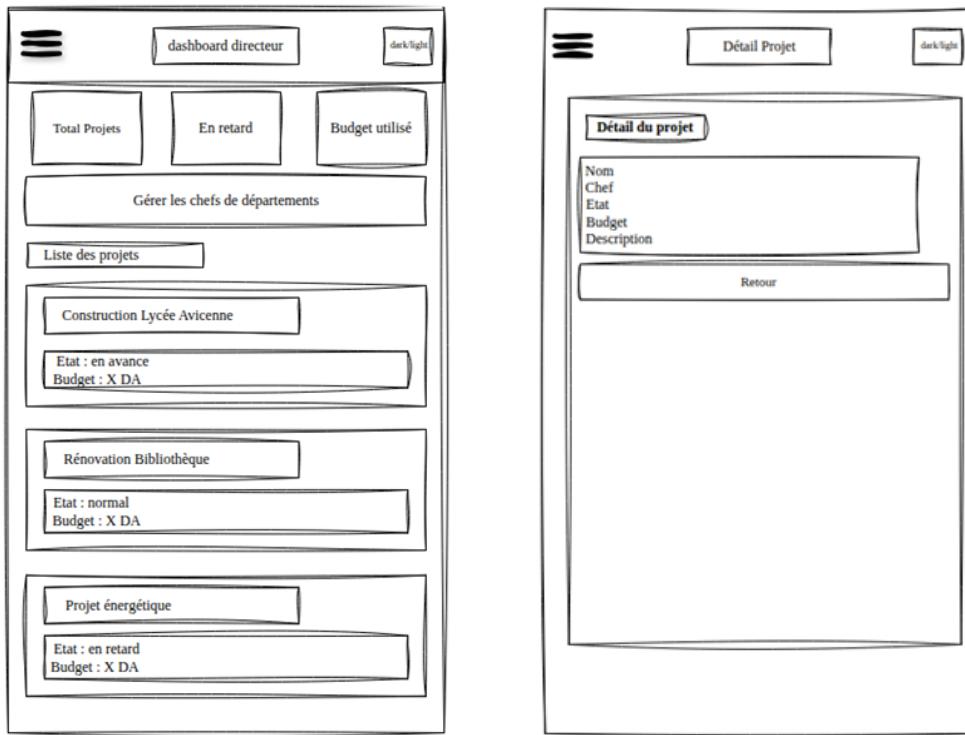


Figure II. 15 – Maquettes : espace administrateur.

II.5.4. Espace chef-département

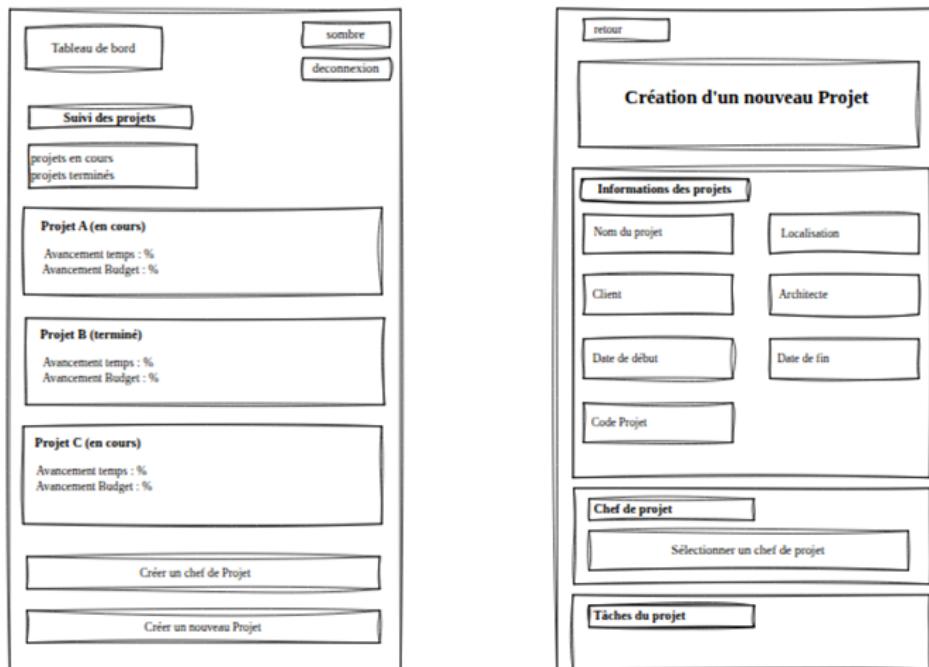


Figure II. 16 – Maquette : espace chef de département.

II.6. Conclusion

Dans ce deuxième chapitre, nous avons procédé à une étude détaillée des besoins fonctionnels du système à développer au sein de l'entreprise Construb-EST. Après une analyse du contexte et l'identification des principaux acteurs, nous avons établi les différents cas d'utilisation décrivant les interactions attendues avec le système. Ces cas ont été illustrés par des descriptions textuelles et des diagrammes UML, permettant de mieux cerner les fonctionnalités à implémenter.

Une deuxième description des cas d'utilisation a été élaborée via les diagrammes de séquence système, afin de mettre l'accès sur l'ordre chronologique des messages entre les acteurs et le système.

Dans le chapitre suivant, nous approfondirons cette phase par une analyse de domaine, à travers des diagrammes de classes candidates, et une conception via des diagrammes de séquence d'interaction (détailé) et le diagramme de classes, et nous terminons par le passage vers le schéma logique de données (schéma relationnel).

CHAPITRE III

ANALYSE & CONCEPTION

CHAPITRE III : ANALYSE ET CONCEPTION

III.1. Introduction

Après avoir défini et étudié les besoins fonctionnels et non fonctionnels du système à développer au sein de l'entreprise Construb-EST Bejaïa, ce chapitre se concentre sur l'étape essentielle de l'analyse et de la conception. Cette phase vise à passer de l'étude des besoins vers une analyse et conception orienté objet, cohérentes et évolutives, constituant ainsi le socle de la réalisation de l'application.

L'analyse débute par l'identification et la structuration des classes candidates, dérivées des cas d'utilisation étudiés précédemment. Ces classes permettent d'établir une base conceptuelle du système, qui sera affinée et validée à travers des diagrammes de séquence détaillés (d'interaction), illustrant les échanges entre objets lors de l'exécution des fonctionnalités. La phase de conception se poursuit par l'élaboration du diagramme de classes de domaine, qui modélise les entités principales, leurs relations et les cardinalités associées. Cette représentation statique du système permet ensuite de générer un schéma relationnel, en vue de la mise en place de la base de données.

III.2. Classes candidates

Le premier objectif des cas d'utilisation, et leur descriptions textuels (ce que nous avons réalisé dans le chapitre I), est de communiquer avec les parties prenantes et de comprendre leur expression préliminaire des besoins. Un deuxième objectif, aussi important que le premier, est de préparer la modélisation orienté objet. Comment ceci est réalisé ?

Ceci est réalisé en établissant, pour chaque cas d'utilisation, les premières classes candidates identifiées dans la phase de capture des besoins (principalement dans la description textuelle des cas d'utilisation). Ces classes représentent des concepts connus du *domaine métier* des utilisateurs du future système, ce qui est appelé couramment (mais abusivement) : *objets métier*, par la suite nous pouvons ajouter des concepts applicatifs : sessions, logs, ...

Dans ce qui suit, on réalise des petits diagrammes *sommaires* de classes candidates pour chaque cas d'utilisation. Ces diagrammes sont sommaires dans le sens où les classes seront établies sans propriétés (attributs) et sans comportements (méthodes), et leurs objectifs est de préparer la phase de conception : *Modèle de classe de conception* (voir [chapitre III](#)). Dans ce qui suit, nous présentant des diagrammes de classes candidates de cas d'utilisation.

III.2.1. Cas d'utilisation « *S'authentifier* »

Les classes candidates qui peuvent être utilisé dans réaliser le cas d'utilisation « *S'authentifier* » sont :

- Utilisateur : représente les comptes utilisateurs qui contiennent des données sur les utilisateurs du système
- Administrateur : qui représente l'acteur utilisateur
- Directeur : modélise l'acteur Directeur
- Chef_Département : représente l'acteur Chef de projet
- Chef_Projet : modélise l'acteur chef de projet
- Session : cette classe représente les sessions de connexions de chaque utilisateur.
- Session_Log : qui modélise les différents actions / évènements produits dans une session.

Voir la figure III.1 pour le diagramme de classes candidates du cas d'utilisation : « *S'authentifier* »

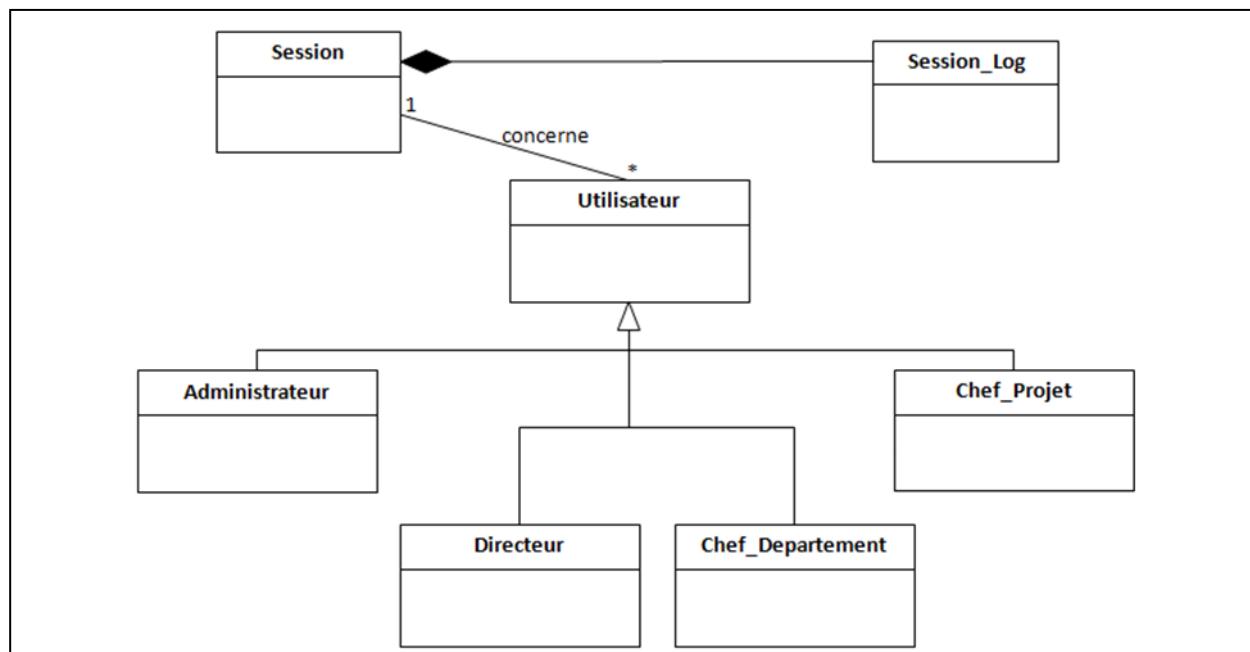


Figure III. 1 – Diagramme de classe candidates « *S'authentifier* »

III.2.2. Cas d'utilisation « Gérer les informations des utilisateurs »

Dans ce cas d'utilisation nous avons une seule classe à candidate, à savoir Utilisateur.

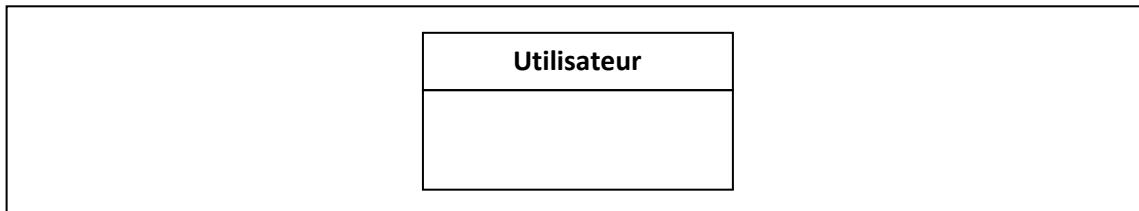


Figure III. 2 – Classes candidates « Gérer les informations des utilisateurs »

III.2.3. Cas d'utilisation « Gérer les droits d'accès »

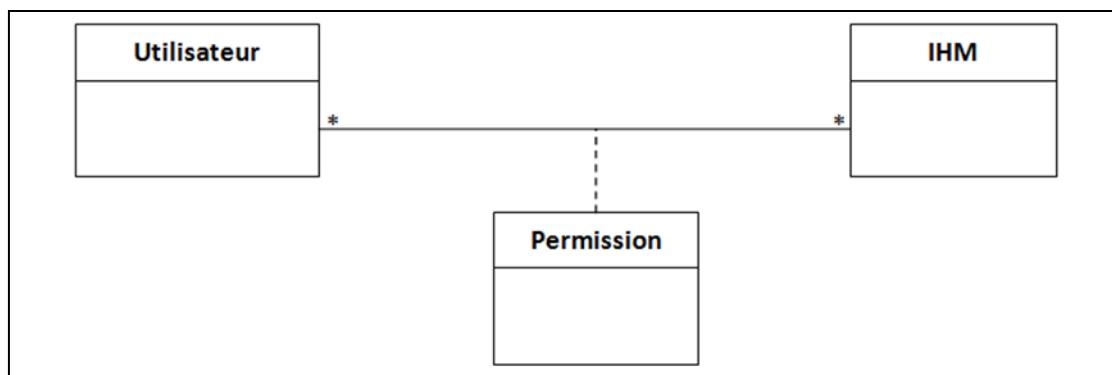


Figure III. 3 – Classes candidates « Gérer les droits d'accès »

III.2.4. Cas d'utilisation « Visualiser le tableau de bord »

Ci-dessous, le diagramme de classes candidate du cas d'utilisation « Visualiser le tableau de bord » :

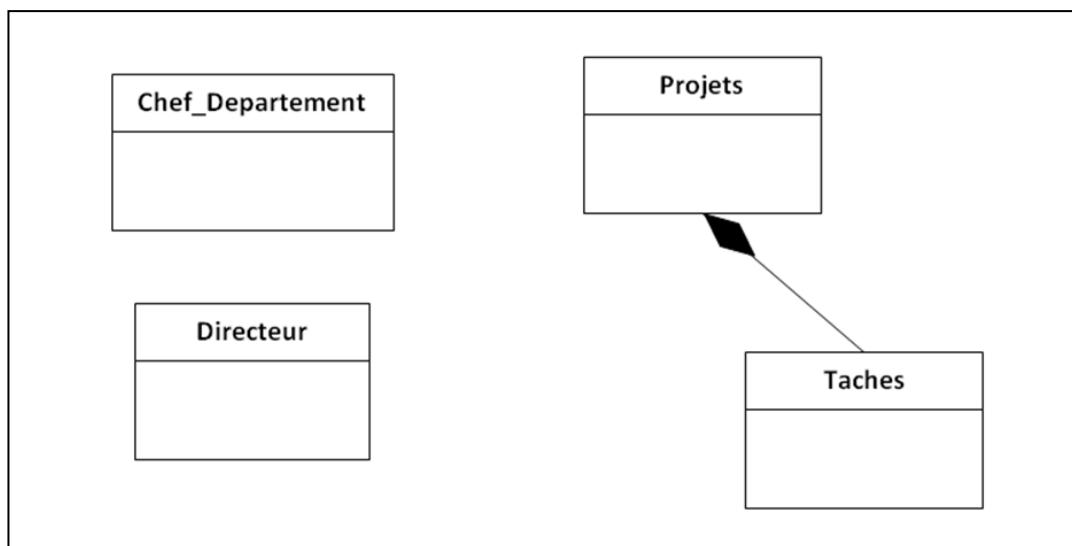


Figure III. 4 – Classes candidates « Visualiser le tableau de bord »

III.3. Diagramme de séquence détaillé (d'interaction)

Pour chaque diagramme de séquence système, nous établirons le diagramme de séquence détaillé (ou d'interaction) (D.S.D.), dans lequel le système est remplacé par les objets qui interviennent pour réaliser le cas d'utilisation concerné. Pour ce type de diagramme, nous avons trois types de classes (voir [02] page 176) :

- Classes d'interface (boundary) : des classes qui permettent l'interaction entre l'application et ses utilisateurs. Pour chaque cas d'utilisation, il y a au moins une classe d'interface. Ce type de classe est schématisé comme suit :



- Classes de Contrôle (Control) : Ce sont des classes qui contiennent les traitements et la cinématique de l'application. Elles font la transition entre les classes d'interface et les classes entité. Elles sont schématisées comme suit :



- Classes entité (entity) : Elles représentent les objets métiers, et ce sont très souvent des entités persistantes, c'est-à-dire qui vont garder leurs informations (données) après l'exécution d'un cas d'utilisation particulier. En général, elles sont enregistrées dans une base de données. Leur schématisation se fait grâce à ce stéréotype :



III.3.1. D.S.D. du cas d'utilisation « S'authentifier »

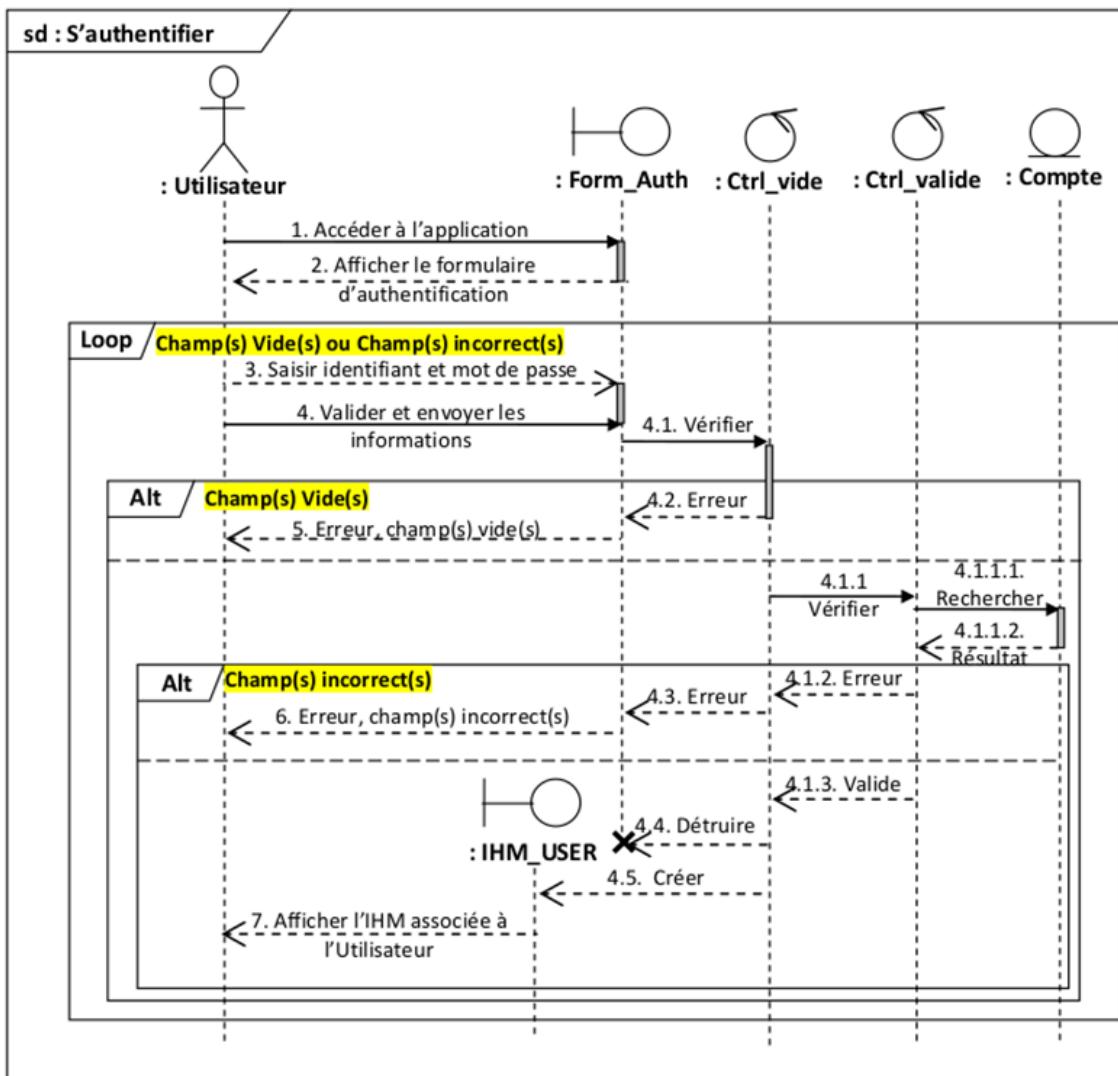


Figure III. 5 – D.S.D. du cas d'utilisation « S'authentifier »

Le tableau suivant résume les stéréotypes utilisés pour représenter les objets qui interviennent pour réaliser le cas d'utilisation « S'authentifier » :

<i>Stéréotype</i>	<i>Signification</i>
: Form_Auth	Objet interface pour le formulaire d'authentification. Ce dernier contient deux champs de saisie, à savoir, l'identifiant (login) et le mot de passe, qui permettent à l'utilisateur de renseigner son identifiant et son mot de passe. En plus, il contient un bouton pour envoyer ces informations.
: Ctrl_vide	Objet contrôle qui permet de vérifier s'il y a un ou plusieurs champs vides. Nous pouvons aussi vérifier dans ce contrôleur la forme des champs, par exemple, le mot de passe doit avoir au moins 7 caractères avec au moins deux caractères numériques, et l'identifiant doit avoir au moins 5 caractères.

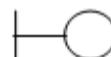
 : Ctrl_valide	Deuxième objet contrôle qui permet de vérifier si les champs renseignés par l'utilisateur sont correctes. Cette vérification s'effectue en utilisant l'objet entité de la classe « Compte » (objet persistant qui enregistre les comptes des utilisateurs du système).
 : Compte	Objet entité (objet métier) qui contient tous les comptes des utilisateurs du système.
 : IHM_USER	Objet interface qui représente la vue principale de l'utilisateur connecté. Nous pouvons la considérer comme l'espace de travail de l'utilisateur. Tous les autres cas d'utilisation seront accessibles à partir de cette interface.

Tableau III. 1 – Les stéréotypes du Diagramme de séquence d'interaction « S'authentifier ».

III.3.2. D.S.D. du cas d'utilisation « Gérer les utilisateurs »

Concernant le diagramme de séquence d'interaction de « Gérer les utilisateurs », les différents stéréotypes utilisés pour représenter les objets qui interviennent pour réaliser ce cas sont les suivant :

Stéréotype	Signification
 : Espace_Admin	Objet interface pour de l'espace administrateur, qui contient un menu pour accéder à la gestion des utilisateurs et aussi aux droits d'accès.
 : Espace_Utilisateur	Objet interface pour l'espace de gestion des utilisateurs (pour faire l'ajout, suppression, modification et désactivation)
 : F_Ajout	Objet interface qui représente un formulaire d'ajout d'un utilisateur
 : Ctrl	Objet contrôle qui permet de réaliser plusieurs d'action : <ul style="list-style-type: none"> - Contrôler les champs du formulaire d'ajout. - Rediriger vers les différentes interfaces - Communiquer avec l'objet entité « Compte »
 : Compte	Objet entité (objet métier) qui contient tous les comptes des utilisateurs du système.

Tableau III. 2 – Les stéréotypes du Diagramme de séquence d'interaction « Gérer les utilisateur ».

Pour le diagramme de séquence d'interaction, il est illustré par la [figure III.6](#).

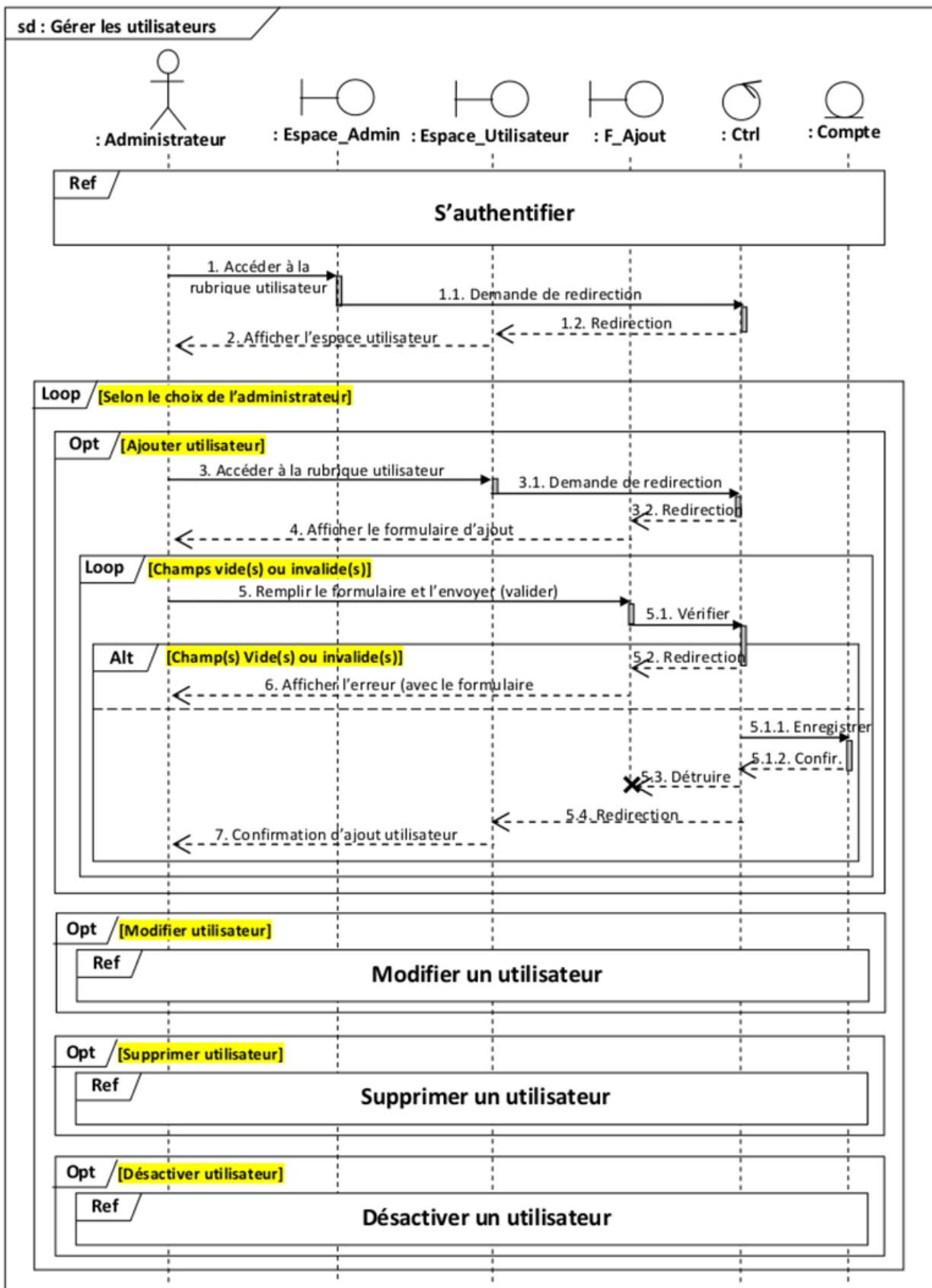


Figure III. 6 – D.S.D. du cas d'utilisation « Gérer les utilisateurs »

III.4. Diagramme de classe de domaine

Le premier objectif des cas

II.4.1. Diagramme de classe

Le diagramme de classe est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens (héritage, agrégation, composition, ...etc.)

La [figure III.7](#) illustre le diagramme de classe de notre application. Nous nous focalisons sur les classes entité (classes du domaine) qui représente les données de l'application.

III.4.2. Dictionnaire de données : Classes, attributs et responsabilité

Dans ce qui suit, nous allons décrire les différentes classes schématisées dans la [figure III.7](#). Cette description est présentée par le tableau suivant :

Classe	Responsabilité	Attributs		
		Désignation	Définition	Type
Utilisateur	Représente toute personne ayant un compte dans le système. Stocke les informations d'identification et d'authentification, ainsi que son rattachement à un service.	<u>id_utilisateur</u>	ID unique	Numérique (<i>entier positif</i>)
		nom	Nom	Alaphabétique
		prénom	Prénom	Alaphabétique
		email	Boite électronique	Alphanumérique (avec format email)
		identifiant	Nom du profil	Alphanumérique
		mot_de_passe	Mot de passe	Alphanumérique et Symboles
		role	Role de l'utilisateur	Enumérée : Admin, Directeur, Chef_Dep, Chef_Proj
Administrateur	Spécialisation de l'utilisateur avec des privilèges complets sur le système : gestion des comptes, des permissions, des services, etc.	<u>id_admin</u>	ID unique	Numérique (<i>entier positif</i>)

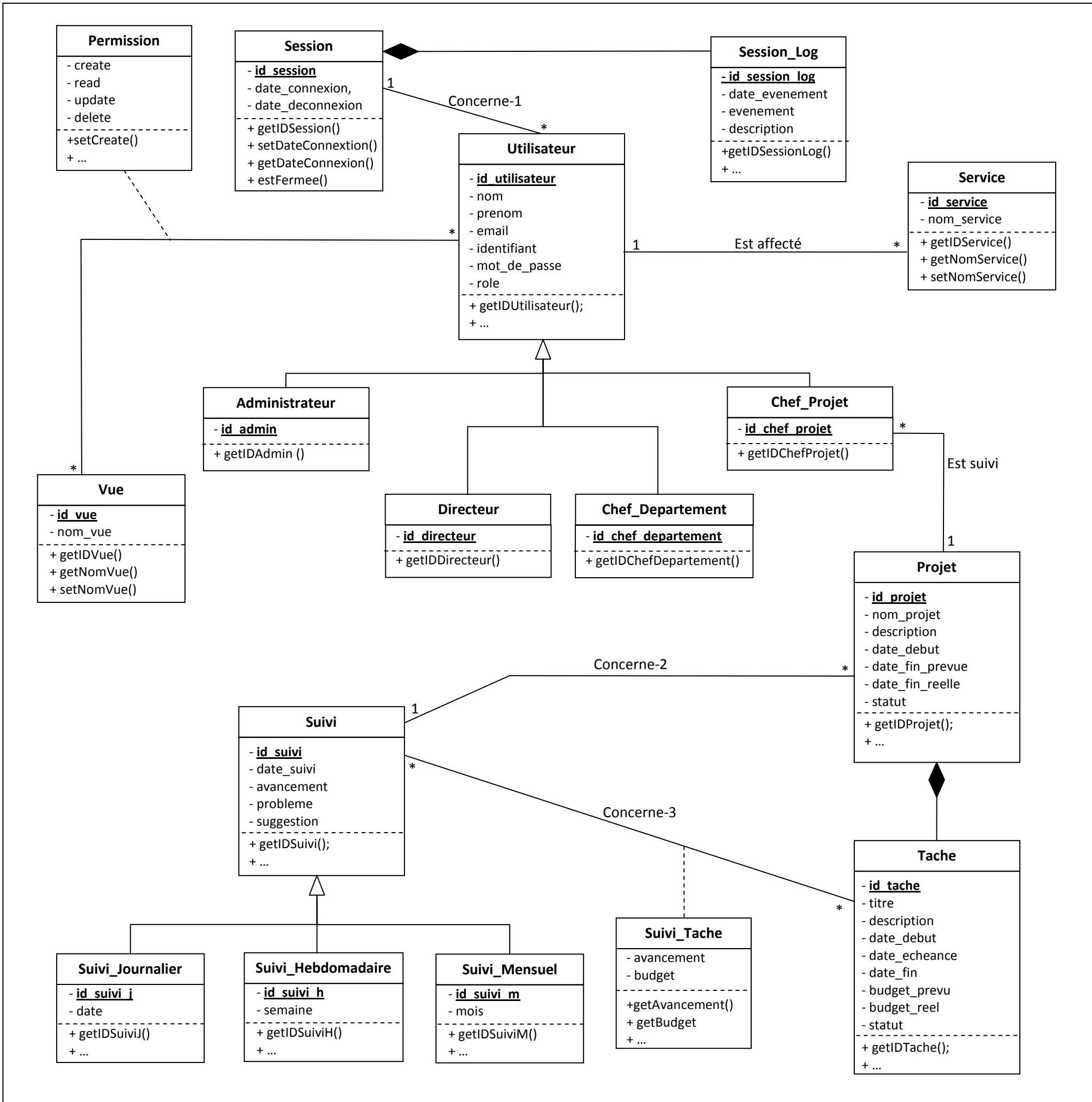


Figure III. 7 – Diagramme de classe du domaine

Classe	Responsabilité	Attributs		
		Désignation	Définition	Type
Directeur	Spécialisation de l'utilisateur qui a une vision globale des projets et des services, souvent responsable des décisions stratégiques.	<u>id_directeur</u>	ID unique	Numérique (entier positif)
Chef_Departement	Utilisateur responsable d'un département. Peut superviser plusieurs projets ou chefs de projet, et assurer la coordination transversale.	<u>id_chef_departement</u>	ID unique	Numérique (entier positif)
Chef_Projet	Utilisateur en charge de l'exécution d'un projet spécifique. Supervise les tâches, les suivis et les membres de l'équipe projet.	<u>id_chef_projet</u>	ID unique	Numérique (entier positif)
Session	Trace les connexions des utilisateurs : moment de connexion, moment de déconnexion.	<u>id_session</u>	ID unique	Numérique (entier positif)
		date_connexion	Date et heure de connexion	Date et Heure
		date_deconnexion	Date et heure de déconnexion	Date et Heure
Session_Log	Enregistre tous les événements survenus durant	<u>id_session_log</u>	ID unique	Numérique (entier positif)
		date_evenement	Date et heure de l'évènement	Date et Heure

	une session (modification, suppression, erreur, etc.), utile pour l'audit et le débogage.	evenement	Indique l'évènement journalisé	Chaîne de caractères
		description	Description de l'évènement	Chaîne de caractères
Service	Modélise les différents services de l'entreprise Construb-Est (Direction de Bejaia)	<u>id_service</u>	ID unique	Numérique (<i>entier positif</i>)
		nom_service	Nom du service	Chaîne de Caractères.
Vue	Décrit les différentes interfaces ou modules fonctionnels accessibles dans le système (ex : page projets, taches, suivis...).	<u>id_vue</u>	ID unique	Numérique (<i>entier positif</i>)
		nom_vue	Nom de la vue (IHM / screen ...)	Chaîne de Caractères.
Permission	Classe d'association qui permet d'affecter des droits d'accès à un utilisateur pour une vue (IHM) spécifique.	create	Droit d'ajout / créer	Numérique (<i>entier positif</i>)
		read	Droit de lecture (Consultation)	Logique (Booléen)
		update	Droit de mise à jour (Modification)	Logique (Booléen)
		delete	Droit de suppression	Logique (Booléen)
Projet	Représente un projet BTPH. Contient des informations de planification, de description, de statut et de gestion par un chef de projet.	<u>id_projet</u>	ID unique	Numérique (<i>entier positif</i>)
		nom_projet	Désignation du projet	Chaîne de caractère
		description	Description du projet	Chaîne de caractère
		date_debut	Date de début du projet	Date (JJ/MM/AAAA)
		date_fin_prevue	Date de fin de projet (planifiée)	Date (JJ/MM/AAAA)
		date_fin_reelle	Date de fin de	Date

			projet réelle	(JJ/MM/AAAA)
		statut	Status du projet : - En cours - En retard - Terminé	Enuméré
Tache	Représente une activité à accomplir dans le cadre d'un projet. Suivie par le chef de projet et les équipes.	<u>id_tache</u>	ID unique	Numérique <i>(entier positif)</i>
		titre	Désignation de la tâche	Chaîne de caractères
		description	Sa description	Chaîne de caractères
		date_debut	Date de début	Date et Heure
		date_echeance	Date de fin prévue	Date
		date_fin	Date de fin réel	Date
		budget_prevu	Budget prévu initialement	Numérique (Nombre réel monétaire)
		budget_reel	Budget consommé réellement	Numérique (Nombre réel monétaire)
		statut	- Non démarré - En cours - Terminé	énuméré
Suivi	Rapport global d'avancement d'un projet à une date donnée (problèmes, avancement, suggestions), rédigé par le chef de projet.	<u>id_suivi</u>	ID unique	Numérique <i>(entier positif)</i>
		date_suivi	Date de réalisation de la constatation (du suivi)	Date
		avancement	Indiquer l'avancement (en %)	Numérique (entre 0 et 100)
		probleme	Relever les problèmes qui bloquent la réalisation de la tâche	Chaîne de caractères
		suggestion	Suggérer des solutions / recommandations	Chaîne de caractères
Suivi_Journalier	Classe fille de	<u>id_suivi_j</u>	ID unique	Numérique <i>(entier positif)</i>

	« Suivi », pour rapporter l'état quotidien du projet ou des activités.			
Suivi_Hebdomadaire	Classe de Spécialisation de la classe « Suivi », pour une constatation hebdomadaire, utile pour les réunions de coordination ou de <i>reporting</i> .	<u>id_suivi_h</u>	ID unique	Numérique (<i>entier positif</i>)
		semaine	Indique la semaine concerné par le suivi, ça peut être : - N° de la semaine dans l'année - AAAA/MM/JJ la date du début de semaine (2025/02/09)	Entier (entre 1 et 52) Ou Date format : AAAA/MM/JJ
Suivi_Mensuel	Spécialisation de suivi mensuel, souvent utilisé pour les bilans de fin de mois et les rapports à la direction.	<u>id_suivi_m</u>	ID unique	Numérique (<i>entier positif</i>)
		mois	Indique la mois concerné par le suivi sous format AAAA/MM (2025/02)	Chaîne de caractère
Suivi_Tache	Classe d'association indiquant l'état d'avancement et de consommation du budget pour les tâches concerné par une opération de suivi. Sert à évaluer l'efficacité du travail et à détecter les dérives budgétaires.	avancement	(Pourcentage du suivie de la tâche)	Numérique (Réel entre 0 et 100)
		budget	Budget consommée à la date de la constatation.	Numérique (Nombre réel format monétaire)

Tableau III. 3 – Dictionnaire des données du schéma conceptuel (Diagramme du classe du domaine).

III.6. Schéma relationnel

A partir d'un diagramme de classe (*schéma logique*), un schéma relationnel (modèle logique de données) est créé, et ceci en appliquant les règles de passage suivantes :

Règle 01 – Classes d'entités : Chaque *classe d'entité* du diagramme de classe devient une relation.

Règle 02 – Association Unique-Multiple (1/0..1 - *) : dans ce cas, nous ajoutons une clé étrangère dans la relation issue de la classe qui participe une seul fois dans l'association vers la clé primaire de la relation issue de la classe qui participe plusieurs fois dans l'association.

Règle 03 – Association Multiple-Multiple (* - *) : l'association sera transformée à une relation, dont la clé primaire est la concaténation des clés primaires des associations issues des classes qui participent dans cette association.

Règle 04 – Héritage : ici, nous avons trois méthodes pour transformer l'héritage vers le schéma relationnel :

- *la méthode ascendante* : Supprimer les relations issues des sous-classes et faire migrer tous les attributs dans la relation issue de la classe mère et ajouter un nouveau attribut qui détermine le type d'une instance de cette relation ;
- *la méthode descendante* : Supprimer la relation issue de la superclasse et faire migrer ses attributs vers les toutes relations issues des sous-classes ;
- *la méthode distincte* : ne pas supprimer aucune relation, ni celle issue de la superclasse, ni celles issues des sous-classes, et définir les clés primaires des relations issues des sous classes comme clés étrangères vers la clé primaire de la relation issue de la superclasse.

Règle 05 – Composition : On met une clé étrangère dans la relation issue du composant vers la relation issue du composé.

Après l'application des règles ci-dessus citées, nous obtiendrons le schéma relationnel (*schéma logique*) suivant :

- Utilisateur (*id utilisateur*, nom, prenom, email, identifiant, mot_de_passe, role, #*id service*)
- Administrateur (#*id admin* → Utilisateur.id_utilisateur)
- Directeur (#*id directeur* → Utilisateur.id_utilisateur)
- Chef_Departement(#*id chef departement* → Utilisateur.id_utilisateur)
- Chef_Projet(#*id chef projet* → Utilisateur.id_utilisateur)

- Session(*id_session*, date_connexion, date_deconnexion, #*id_utilisateur*)
- Session_Log(*id_session_log*, date_evenement, evenement, description, #*id_session*)
- Service(*id_service*, nom_service)
- Vue(*id_vue*, nom_vue)
- Permission(#*id_utilisateur*, #*id_vue*, create, read, update, delete)
- Projet(*id_projet*, nom_projet, description, date_debut, date_fin_prevue, date_fin_reelle, statut, type, id_chef_projet → Chef_Projet.id_chef_projet)
- Tache(*id_tache*, titre, description, date_debut, date_echeance, date_fin, budget_prevu, budget_reel, statut, #*id_projet*)
- Suivi(*id_suivi*, date_suivi, avancement, probleme, suggestion, #*id_projet*, #*id_chef_projet*)
- Suivi_Journalier(#*id_suivi_j* → Suivi.id_suivi, date)
- Suivi_Hebdomadaire(#*id_suivi_h* → Suivi.id_suivi, semaine)
- Suivi_Mensuelle(#*id_suivi_m* → Suivi.id_suivi, mois)
- Suivi_Tache(#*id_suivi*, #*id_tache*, avancement, budget)

III.7. Conclusion

À l'issue de ce chapitre, l'ensemble des fondations conceptuelles de l'application a été posé à travers une démarche d'analyse et de conception orientée objet. En partant des cas d'utilisation définis précédemment, les classes candidates ont été identifiées, ce qui a permis de structurer les entités principales du système. Par la suite, nous avons élaboré les diagrammes de séquence d'interaction (détaillés) qui modélisent dynamiquement les interactions entre les objets pour chaque fonctionnalité essentielle (*cas d'utilisation*). Ces interactions ont constitué un socle indispensable pour construire le diagramme de classes de domaine (*objets entité*).

Enfin, un schéma relationnel est généré à travers le diagramme de classe de domaine, et qui est nécessaire à la création de la base de données.

Le chapitre suivant portera sur la réalisation technique de l'application ainsi que les tests menés pour valider son bon fonctionnement par rapport aux exigences initiales.

CHAPITRE IV

REALISATION & TESTS

CHAPITRE IV :

REALISATION ET TESTS

IV.1. Introduction

Ce présent chapitre sera dédié à la partie de réalisation de notre application et la mise en œuvre de l'ensemble des techniques pour atteindre cet objectif. Par la suite, nous présentons son architecture en utilisant le diagramme de déploiement. Nous allons, par la suite, présenter quelques tests réalisés sur serveur *backend*, la partie *frontend* (l'application mobile / web).

IV.2. Langages et outils de programmation

Dans cette section, nous allons énumérer les différentes technologies (outils et langages de programmation) qui sont utilisées pour développer notre solution.

IV.2.1. Visual Studio Code

VS Code (ou Visual Studio Code) [\[03\]](#), téléchargeable sur : <https://code.visualstudio.com/>, est un éditeur de code source qui peut être utilisé avec une large variété de langage de programmation (C, C#, C++, Java, Javascript, ...). Il est un logiciel propriétaire et distribué sous la licence "Microsoft Software License", mais une autre version nommée VS Code Open Source a été créée aussi par Microsoft et elle est disponible via le GitHub. Il possède deux versions : une version téléchargeable construite sur la plateforme *Electron JS* () (qui est utilisé pour créer des applications desktop à base de technologies WEB), et une deuxième version web, accessible en ligne (voir vscode.dev).

IV.2.2. PostgreSQL

PostgreSQL est un puissant système de base de données relationnelle objet open source, développé activement depuis plus de 35 ans, qui lui a valu une solide réputation en termes de fiabilité, de robustesse des fonctionnalités et de performances. [\[04\]](#)

IV.2.3. PgAdmin

PgAdmin est la plateforme d'administration et de développement Open Source la plus populaire et la plus riche en fonctionnalités pour PostgreSQL, la base de données

Open Source la plus avancée au monde. pgAdmin peut être utilisé sur Linux, Unix, macOS et Windows pour gérer PostgreSQL et EDB Advanced Server 11 et plus. [\[05\]](#)

IV.2.4. Javascript

JavaScript est un langage de script incorporé dans les documents HTML et exécuté par le navigateur (Google chrome, Firefox, Microsoft Edge, ...) [\[06\]](#).

IV.2.5. Node JS

Node.js® est un environnement d'exécution JavaScript construit sur le moteur JavaScript V8 de Chrome [\[07\]](#).

IV.2.6. Express JS

Un Framework, minimaliste, pour créer des applications web et des serveurs REST API pour Node JS. [\[08\]](#)

IV.2.7. React JS

Une librairie permettant de faciliter la création des interfaces web en se basant sur des composants réutilisable, créé par Meta. [\[09\]](#)

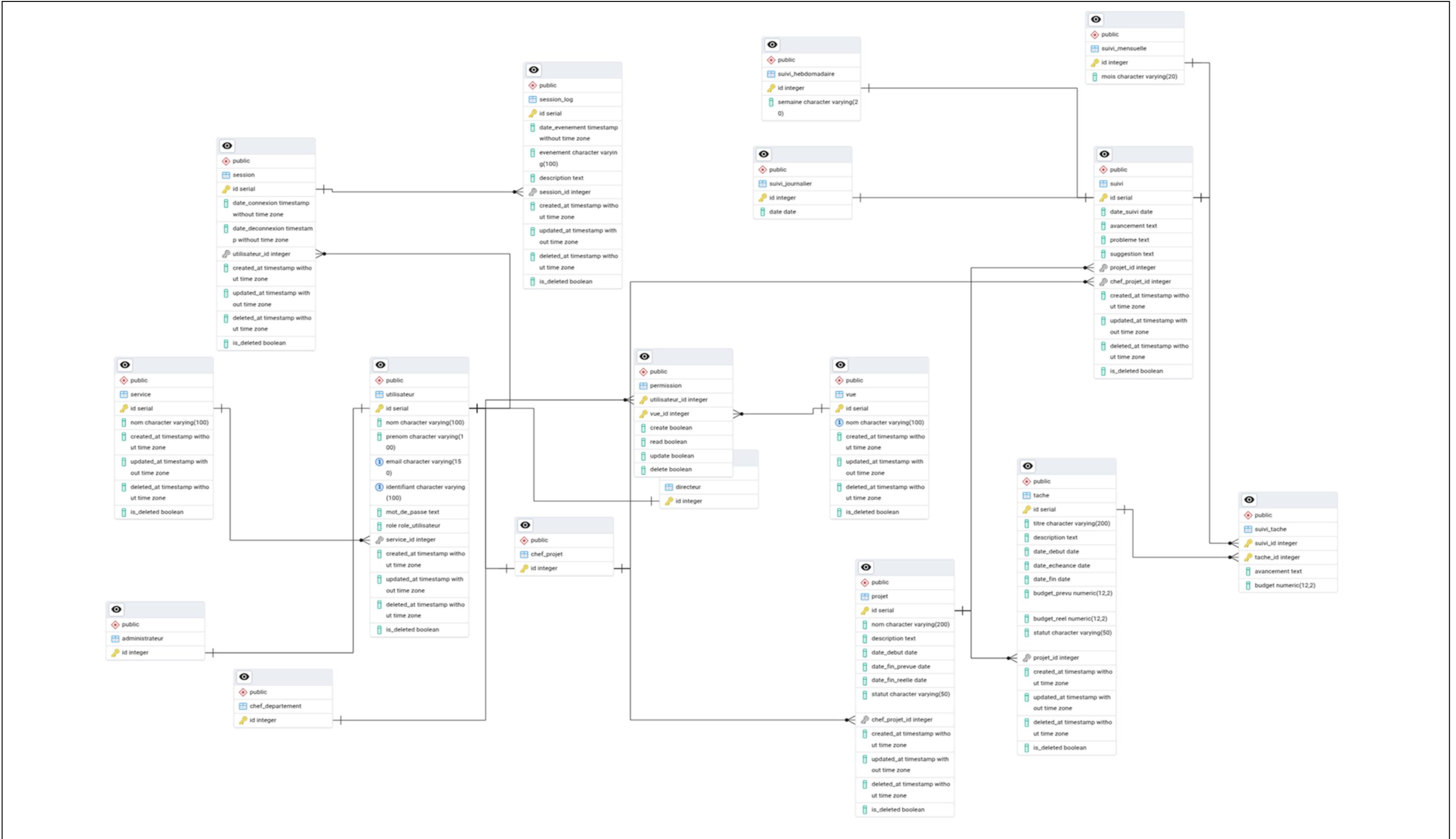
IV.2.8. Cordova

Cordova intègre votre application HTML/JavaScript dans un conteneur natif qui peut accéder aux fonctions de l'appareil de plusieurs plateformes. Ces fonctions sont exposées via une API JavaScript unifiée, ce qui vous permet d'écrire facilement un seul jeu de code pour cibler presque tous les téléphones ou tablettes sur le marché aujourd'hui et de les publier dans leurs magasins d'applications. [\[10\]](#)

IV.3. Schéma physique de la base de données

Concernant la base de données, nous avons utilisé le SGBDR (Système de Gestion de Base de Données Relationnel) PostgreSQL, ainsi que l'outil PgAdmin, qui nous a permis la création des tables, définition des champs et relations entre les tables. Nous avons implémenté (voir l'[annexe C](#) pour le code SQL) le schéma relationnel de la section [III.6](#) avec quelques modifications d'uniformisation :

- les clés primaires des relations issues des classes d'entité transformées en **id** ;
- les clés étrangères sous format xxx_id (*par exemple : service_id*)

Figure IV. 1 – Schéma physique de la base de données (`ce_btph_db`).

- ajouter les champs de traçabilité created_at, updated_at, deleted_at
- ajouter la suppression logique (*soft delete*) avec les champs is_deleted et deleted_at

Le schéma physique de la base de données créée (**ce_btph_db**) est illustré par la [figure IV.1](#). Ce dernier a été généré en utilisant l'outil *PgAdmin*.

IV.4. Architecture de la solution

Après avoir présenté la base de données (Partie des données) de notre application, nous expliquerons l'architecture de l'application en termes de déploiement des modules qui la constituent sur les différentes machines (Serveur, Ordinateur de travail et les téléphones intelligents (Smartphones)). En plus de ça, nous présentons la structuration de notre code source (Partie client et serveur).

IV.4.1. Diagramme de déploiement

La figure suivante illustre les composants de notre application leur exécution par les différentes machines :

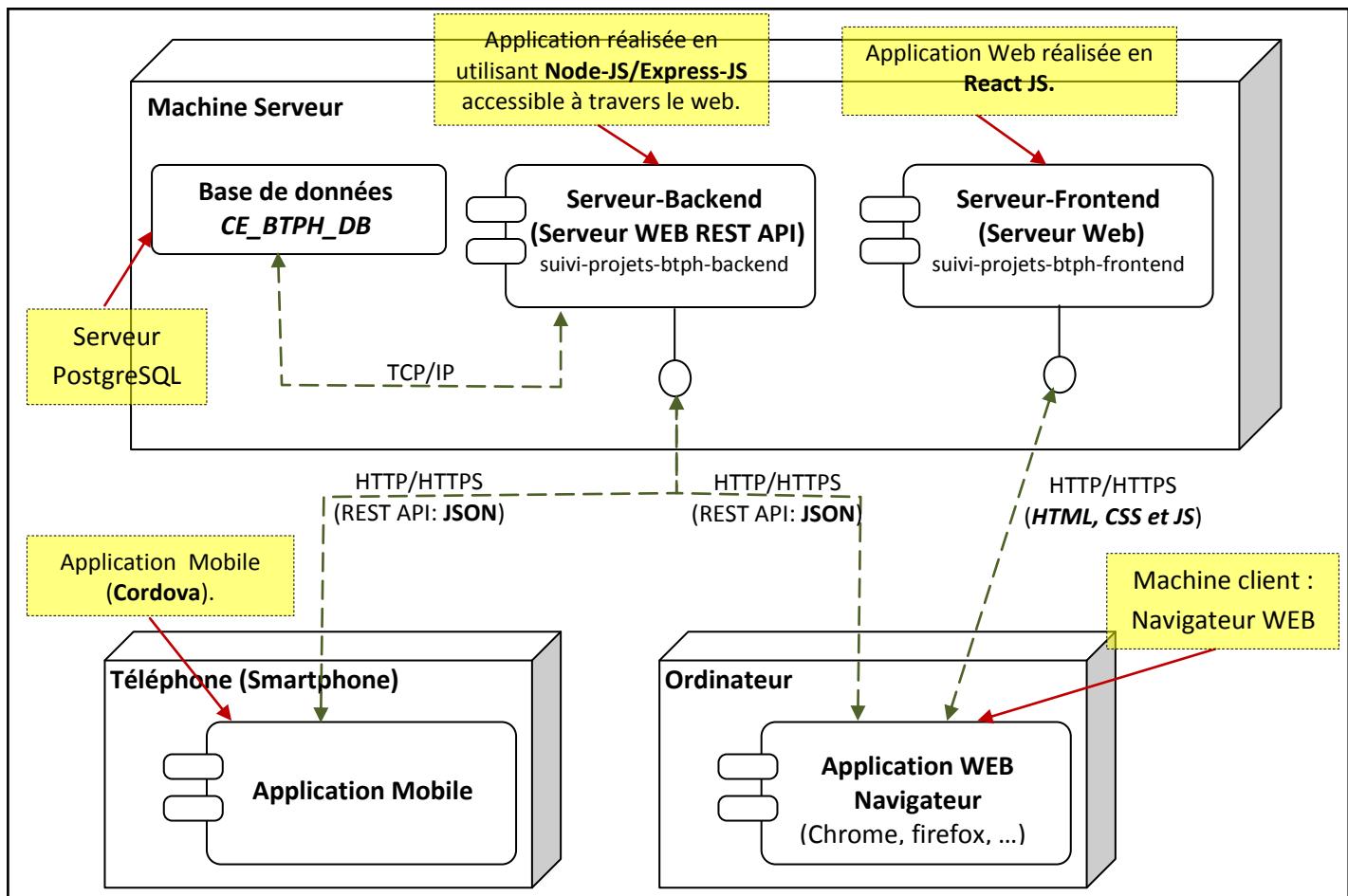


Figure IV. 2 – Diagramme de déploiement.

IV.4.2. Eléments du diagramme de déploiement

Comme indiqué par la [figure IV.2](#), nous avons les composants (modules) suivants :

a- Base de données

La base de données, nommée `ce_btph_db`, et comme précédemment, est implémentée sur un serveur PostgreSQL. Par conséquent, c'est une base de données relationnelle. Cette dernière sert à répondre aux requêtes du Serveur Backend (création, mise à jour, suppression et récupération / lecture de données). Pour le code SQL complet de notre base de données voir l'[annexe C](#).

b- Serveur Backend

Ce composant représente un serveur REST API (Representational State Transfer Application Programming Interface), réalisé avec l'environnement Node JS et le framework Express JS. Ce backend répond à des requêtes HTTP (avec méthodes : GET, POST, PUT et DELETE) avec des données sous format JSON (JavaScript Object Notation). Dans notre architecture, ce serveur répond aux requêtes émanant de l'application web ou de l'application mobile.

c- Serveur Frontend

C'est un serveur qui permet aux postes clients (Ordinateurs et smartphones), via un navigateur web (Chrome, firefox, ...), d'accéder à travers une requête HTTP (méthode GET). Ce serveur répond par une page HTML, CSS et Javascript. Ces trois derniers éléments constituent une application WEB de type Single Page Application (SPA) générée via la bibliothèque React JS.

d- Application web

Ce module permet aux utilisateurs d'accéder à l'application via une interface WEB, de type SPA. Le navigateur WEB envoie une requête HTTP vers le serveur Frontend pour obtenir le code HTML, CSS et JS. Cette requête est réalisée une seule fois par application (sauf si on rafraîchi la page web). Une fois chargé, l'application affiche une l'IHM (HTML et CSS) qui interagit avec l'utilisateur via l'exécution du JavaScript (principalement pour le traitement d'évènement ou bien communiquer avec le serveur backend).

e- Application mobile

Pour ce dernier composant, on peut l'implémenter de plusieurs façon : application natives (Java / Kotlin pour Android, Swift pour iPhone, ...) ou application hybride (Flutter ou React-Native). Dans notre cas, et pour gagner du temps, nous avons utiliser le framework Cordova [10] afin de convertir le code HTML, CSS et Javascript de l'application WEB à une application Mobile.

IV.4.3. Structure de code source

a- Backend

Pour la backend de notre solution, nommée « *suivi-projet-btph-backend* », nous illustrons à travers cette figure les grandes sections de fichiers JS du serveur Web REST API :

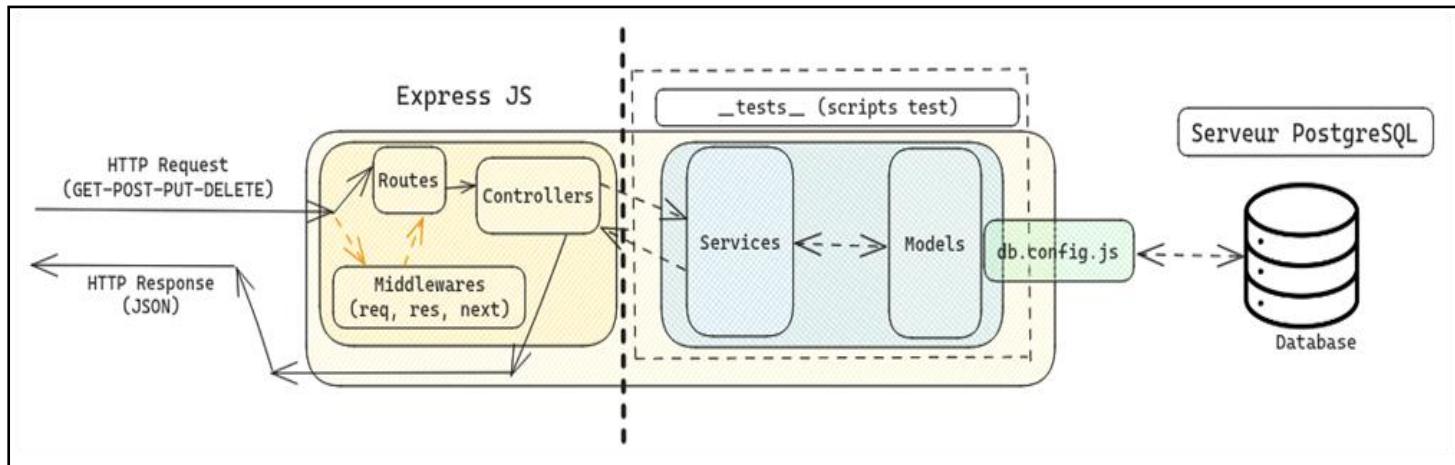


Figure IV. 3 – Structure du code source - Backend.

b- Frontend

Pour le module « *suivi-projet-btph-frontend* », nous avons utilisé du React-JS, avec un projet généré avec l'outil *Vite*. Voici la structure du projet React JS :

L'image montre une capture d'écran d'un éditeur de code (probablement VS Code) avec les détails suivants :

- Barre de menu :** File, Edit, Selection, View, Go, ...
- Barre d'outils :** Explorer, Open Editors, Find, Go To, etc.
- Explorateur (Left Bar) :**
 - Projet nommé "SUIVI-PROJETS-BTPH-FRONTEND".
 - Dossiers : dist, node_modules, public, src.
 - src sous-dossiers : components, contexts, hooks, lib, pages.
 - Fichiers : App.css, App.tsx, index.css, main.tsx, vite-env.d.ts, .gitignore, bun.lockb, components.json, eslint.config.js, index.html, package-lock.json, package.json, postcss.config.js, README.md.
- Éditeur principal (Right Side) :**
 - Moniteur de code : "App.tsx M" (en cours).
 - Code de l'application :

```

import { QueryClientProvider } from '@tanstack/react-query';
import { AuthProvider } from './contexts/AuthContext';
import { ThemeProvider } from './contexts/ThemeContext';
import { Toaster } from 'react-hot-toast';
import { RouterProvider } from 'react-router-dom';
import { Routes } from './routes';

const App = () => (
  <QueryClientProvider client={queryClient}>
    <AuthProvider>
      <ThemeProvider>
        <Toaster />
        <RouterProvider>
          <Routes>
            <Route path="/" element={<Index />} /> /*Pour Co
            <Route path="/" element={<Index />} />
            <Route path="/login" element={<Login />} />
            <Route path="/dashboard" element={<Dashboard />} />
            <Route path="/users" element={<Users />} />
            <Route path="/permissions" element={<Permissions />} />
            <Route path="/project-details" element={<ProjectDetails />} />
            <Route path="/department-heads" element={<DepartmentHeads />} />
          </Routes>
        </RouterProvider>
      </ThemeProvider>
    </AuthProvider>
  </QueryClientProvider>
)

```

Figure IV. 4 – Structure du code source - Frontend.

Voici quelques interfaces Homme-Machine de l'application WEB :

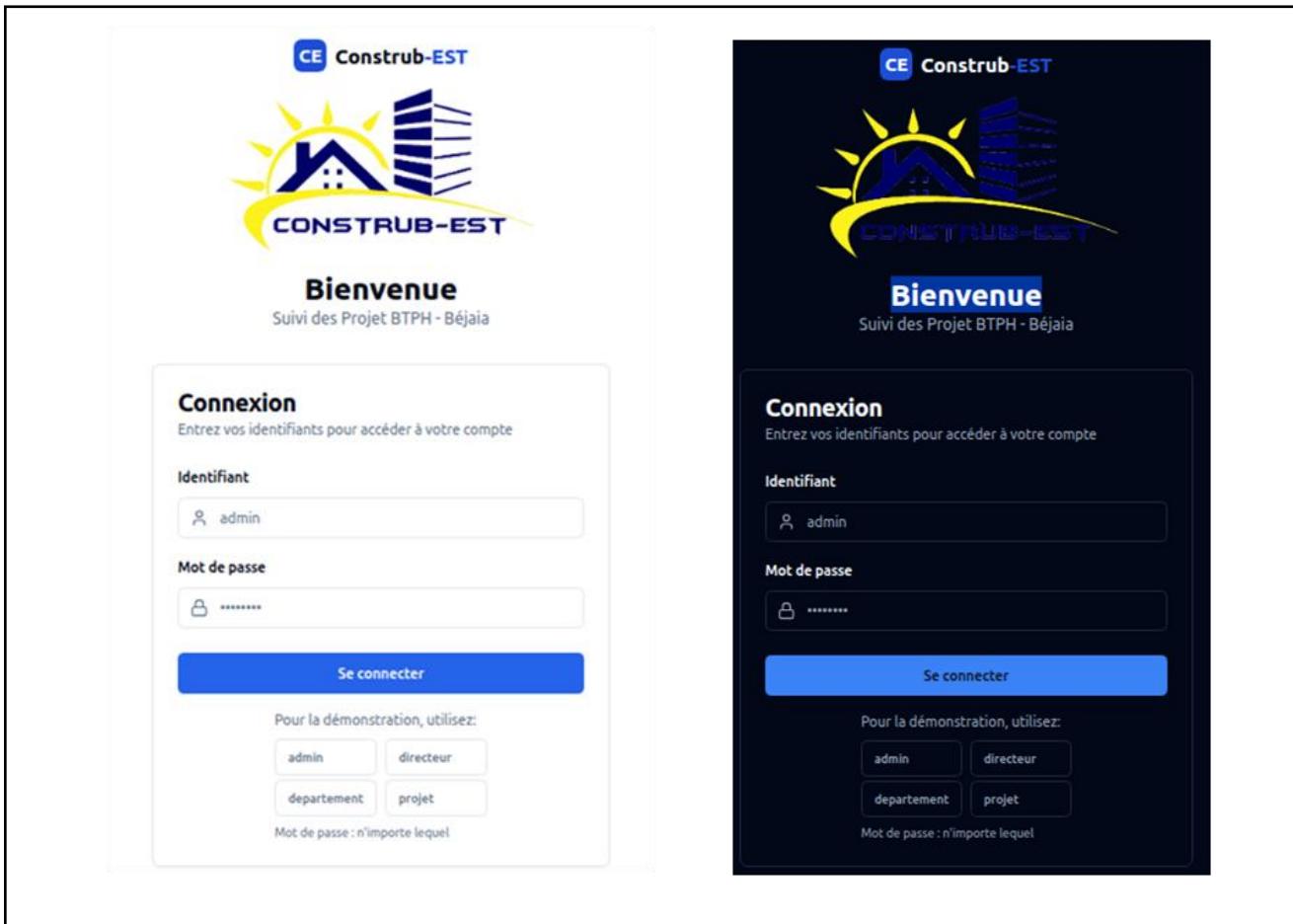


Figure IV. 5 – Exemples d'interfaces de l'Application WEB.



Figure IV. 6 – Interface d'accueil pour l'acteur Administrateur.

IV.5. Tests

Dans la phase actuel du projet, nous avons planifié de réaliser des tests pour le serveur Backend, Web et Mobile.

Pour le serveur Backend, nous allons utiliser un testeur d'API, par exemple Postman, par la suite, nous allons effectuer des tests de fonctionnalité pour la partie frontend.

IV.6. Conclusion

Dans ce dernier chapitre, nous avons vu les différentes étapes de réalisation et d'implémentation de notre solution. Vu que le stage en cours, quelques phases, comme les tests, l'implémentation de toutes les maquettes sont en cours de réalisation. Ceci est du à l'importance accordé à la phase d'étude de l'existant, d'analyse des besoins, d'analyse de domaine et de conception.

CONCLUSION GÉNÉRALE

CONCLUSION

GENERALE

Ce travail de mémoire a eu pour objectif de concevoir et de réaliser une application mobile destinée à améliorer la gestion des projets de bâtiment et de travaux hydrauliques au sein de la Direction Régionale de Construb-Est à Béjaïa. Face aux difficultés rencontrées dans le suivi des chantiers, la communication entre les équipes et la centralisation des informations, il est apparu nécessaire d'envisager une solution numérique adaptée aux contraintes du terrain : une application mobile s'est alors imposée comme une réponse pertinente.

Dans un premier temps, nous avons étudié le contexte organisationnel de l'entreprise, identifié les besoins réels des utilisateurs et formulé une proposition de solution. À travers cette analyse préliminaire, un cahier des charges détaillé a été établi, définissant les fonctionnalités attendues ainsi que les exigences techniques et ergonomiques.

L'étude approfondie des besoins nous a ensuite permis de mieux comprendre les attentes des différents acteurs du projet. La définition des cas d'utilisation et la modélisation des interactions ont joué un rôle clé dans la construction d'une solution centrée utilisateur.

La phase d'analyse et de conception a consisté à structurer la solution sous forme de modèles orientés objet. Les diagrammes UML (classes, séquences) et le schéma relationnel de la base de données ont servi de fondations solides à l'implémentation.

Enfin, la réalisation de l'application a débuté à l'aide d'une pile technologique moderne : Node.js et Express pour le backend, React pour l'interface utilisateur, PostgreSQL pour la gestion des données et Cordova pour le déploiement mobile. Bien que certaines fonctionnalités soient encore en cours de développement, les bases techniques sont opérationnelles et les premiers tests attestent d'une bonne adéquation avec les besoins exprimés.

Au-delà de l'aspect technique, ce projet a représenté une occasion concrète d'appliquer les compétences acquises durant mon parcours en BUT Informatique. Il a mobilisé plusieurs unités de formation, notamment la Compétence 4 (UE44) – Gérer des données d'information, la Compétence 5 (UE45) – Conduire un projet informatique, la Compétence 1 (UE61) – Réaliser un développement d'application, ainsi que la Compétence 6 (UE66) – Travailler dans une équipe informatique.

Sur un plan personnel, ce mémoire m'a permis de développer mon autonomie, ma rigueur et ma capacité à gérer un projet complexe de A à Z. Professionnellement, il m'a confronté à la réalité du terrain, m'a aidé à perfectionner mes compétences techniques et à renforcer ma capacité à collaborer avec des interlocuteurs non informaticiens.

En somme, ce projet constitue une étape essentielle de ma formation, à la fois par les connaissances techniques qu'il a permis de mettre en œuvre, mais aussi par l'expérience humaine et professionnelle qu'il a représentée. Il marque une transition importante vers le monde professionnel, en affirmant une démarche méthodique, une ouverture au travail en collaboration et une volonté constante de répondre aux besoins réels des utilisateurs.

BIBLIOGRAPHIE

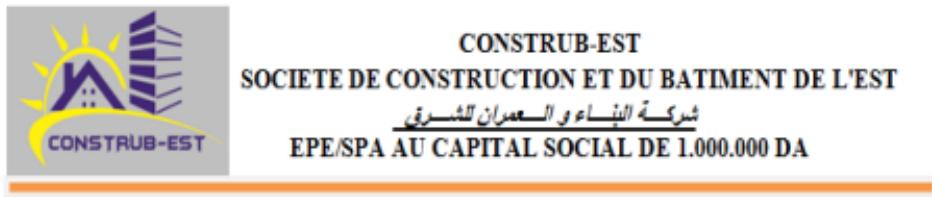
BIBLIOGRAPHIE

- [01] Laurant AUDIBERT. ‘*UML 2 - De l'apprentissage à la pratique*’. Site web. URL : <https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvre-uml>. Publié le 31 octobre 2006 - Mis à jour le 12 janvier 2009. Date de consultation : Avril 2025.
- [02] Roy GILLES. ‘*UML2 en action, de l'analyse des besoins en action*’. Presses de l'Université de Québec, 2009, première édition.
- [03] Microsoft, ‘*Your code editor. Redefined with AI*’ . VS-Code, Site web. URL: <https://code.visualstudio.com/> . Date de consultation: Avril 2025.
- [04] The PostgreSQL Global Development Group, ‘*PostgreSQL: The World's Most Advanced Open Source Relational Database*’. Page web. URL : <https://www.postgresql.org/>. Date de consultation : Mai 2025.
- [05] pgAdmin, PostgreSQL Tools, ‘*PostgreSQL: The World's Most Advanced Open Source Relational Database*’. Page web. URL : <https://www.pgadmin.org/>. Date de consultation Mai 2025.
- [06] Pluralsight team, ‘*JS JavaScript.com. Ready to try JavaScript ?*’. Page Web. URL : <https://www.javascript.com>. Date de consultation : Mai 2025.
- [07] Node.js Foundation, ‘*Node JS. A JavaScript runtime built on Chrome's V8 Javascript engine*’. Page Web. URL : <https://nodejs.org>. Date de consultation : Mai 2025.
- [08] Open JS Foundation, ‘*Express, Cadre de travail web rapide, neutre et minimaliste pour Node.js*’. Page Web. URL : <https://expressjs.com> . Date de consultation : Mai 2025.
- [09] Meta Platforms, Inc, ‘*React. La bibliothèque pour les interfaces utilisateur web et natives*’. Page Web. URL : <https://react.dev>. Date de consultation : Mai 2025.
- [10] The Apache Software Foundation, ‘*APACHE CORDOVA (TM).Mobile apps with HTML, CSS & JS*’. Page Web. URL : <https://cordova.apache.org>. Date de consultation : Mai 2025.

ANNEXES

ANNEXES

Annexe A – Fiche technique de l'entreprise Construb-Est



FICHE TECHNIQUE DE L'ENTREPRISE

Présentation de l'entreprise

Création : la société de Construction et du Bâtiment de l'EST EX EPBTP/Béjaia créée par décret numéro 51.236 du 29 Aout 1981.

CONSTRUB- EST Béjaia est une Enterprise de construction générale. Elle opère dans le domaine du BTPH. Dans le souci de mieux servir sa clientèle avec promptitude, efficacité et rigueur, elle a structuré ses activités à la réalisation de projet de construction, Assistance à la Maitrise d'ouvrage et maitrise d'œuvre , La Réalisation de Travaux de Bâtiments et de Génie Civil TCE ,la Réalisation de Charpente Métallique, tuyauterie, chaudironnerie, installations industrielles, elle se caractérise par des valeurs fortes qui sont : l'écoute et la satisfaction du client, la recherche de l'excellence, le leadership et le sens de la responsabilité. C'est en cela qu'elle se garanti un avenir prometteur et une notoriété naissante.

Qualification et classification professionnelle : Catégorie 08

Domaines d'intervention

- Travaux publics
- Bâtiment
- Hydraulique

Adresse et Situations géographique

- **Adresse :** Boulevard BOUMDAOUI Nacer, promotion immobilière EX EPBTP, Bloc H1 , Béjaia
- **Téléphone :** 034 12 00 86
- **Fax :** 034 12 00 76

MOYENS HUMAINS :

L'entreprise est dotée de 165 employés dont :

Effectif Technique

- Chef de projet : 08
- Architecte : 02
- Chefs de chantier : 04
- Métreur vérificateur : 02
- Technicien en bâtiment : 02
- Plombiers : 02
- Ferrailleurs : 12
- Coffreurs : 10
- Maçons : 14
- Ouvriers qualifiés : 20
- Grutiers : 02
- Electriciens : 02

Autres

- Chefs d'atelier maintenance : 02
- Chauffeurs : 13
- Conducteurs d'engins : 09
- Agents de sécurité : 25
- Personnel administratif : 36

MOYENS MATERIELS :

Matériel de terrassement, compactage, revêtement et forage	Matériel de transport lourd et léger	Matériel de production et mise en œuvre du béton	Matériel de manutention, levage et chargement	Compresseurs groupes électrogène et distribution d'énergie	Matériels de concassage et de broyage criblage
03 Chargeurs SIP : 04 Dumper de chantier 01 Niveleuse 03 Pelle Hydraulique 04 Rétro chargeur 01 Rouleau compacteur	01 Camion-citerne 13 Camions benne 07 Camions malaxeur 07 Tracteur routier 04 Remorque plateau 04 Remorque semi-benne 02 Remorque porte engin 02 Remorque porte citerne : 08 Véhicules utilitaire	03 Centrale à béton 03 Bétonnières 04 Mini centrale à béton 01 unité de fabrication d'agglomérés	03 Grue mobile Volvo 02 Chariot élévateur 10 Grues Tour 02 Grue montage rapide	10 Compresseurs 10 Groupes électrogènes 03 Transformateur	01 Centrale de concassage (une concession d'extraction de TVO avec station de concassage d'agrégats) 01 (un) atelier de menuiserie à bois 01 (un) atelier de façonnage des aciers

Référence professionnelle de l'entreprise

Projets réalisés les 03 Dernières années :

- Réalisation d'un centre de formation au niveau de la zone logistique extra portuaire d'ighil Ouberouak (**Bejaia Mediteranean Terminal**)
- Réhabilitation des voies de circulation (350 ml) (**Bejaia Mediteranean Terminal**)
- Réalisation de réseaux d'assainissement et des eaux pluviales (**Bejaia Mediteranean Terminal**)
- Travaux de réfection des locaux et structures du port de Béjaia (**Entreprise Portuaire de Béjaia**)
- Réhabilitation d'une parcelle d'entreposage de conteneurs de 200 m² (**Bejaia Mediteranean Terminal**)
- Aménagement d'un abri au niveau de l'appontement de la brise de mer de Bejaia (**Entreprise Portuaire de Béjaia**)
- Travaux de réfection du centre médico-social, w Béjaia (**Mutuelle Générale des Travailleurs des postes et Télécommunications**)
- Travaux de bétonnage de l'air de jonction P24/25 au niveau du port de Béjaia (**Entreprise Portuaire de Béjaia**)
- Réalisation d'une canalisation qui acheminera les Câbles fibre optique au niveau du terminal à conteneurs de BMT /SPA (**Bejaia Mediteranean Terminal**)

- Réalisation d'un réservoir interne au niveau de la bâche à eau de la ZONE EXTRA PORTUAIRE D'IGHIL OUBAROUAK (**Entreprise Portuaire de Béjaia**)
- Réhabilitation du parc visite de BMT SPA (**Bejaia Mediteranean Terminal**)
- Aménagement d'un centre de vacances d'Algérie Télécom à EL KENNAR Wilaya de JUEL (**Comité de participation d'Algérie Telecom**)
-

Photos des projets réalisés récemment :

Réalisation d'un centre de formation au niveau de la zone logistique extra portuaire d'ighil Ouberouak



Façade principale



Salle de réunion

Aménagement d'un centre de vacances d'Algérie Télécom à EL KENNAR Wilaya de JIJEL



Aménagement des espaces verts



Pavé

Réalisation d'une fontaine



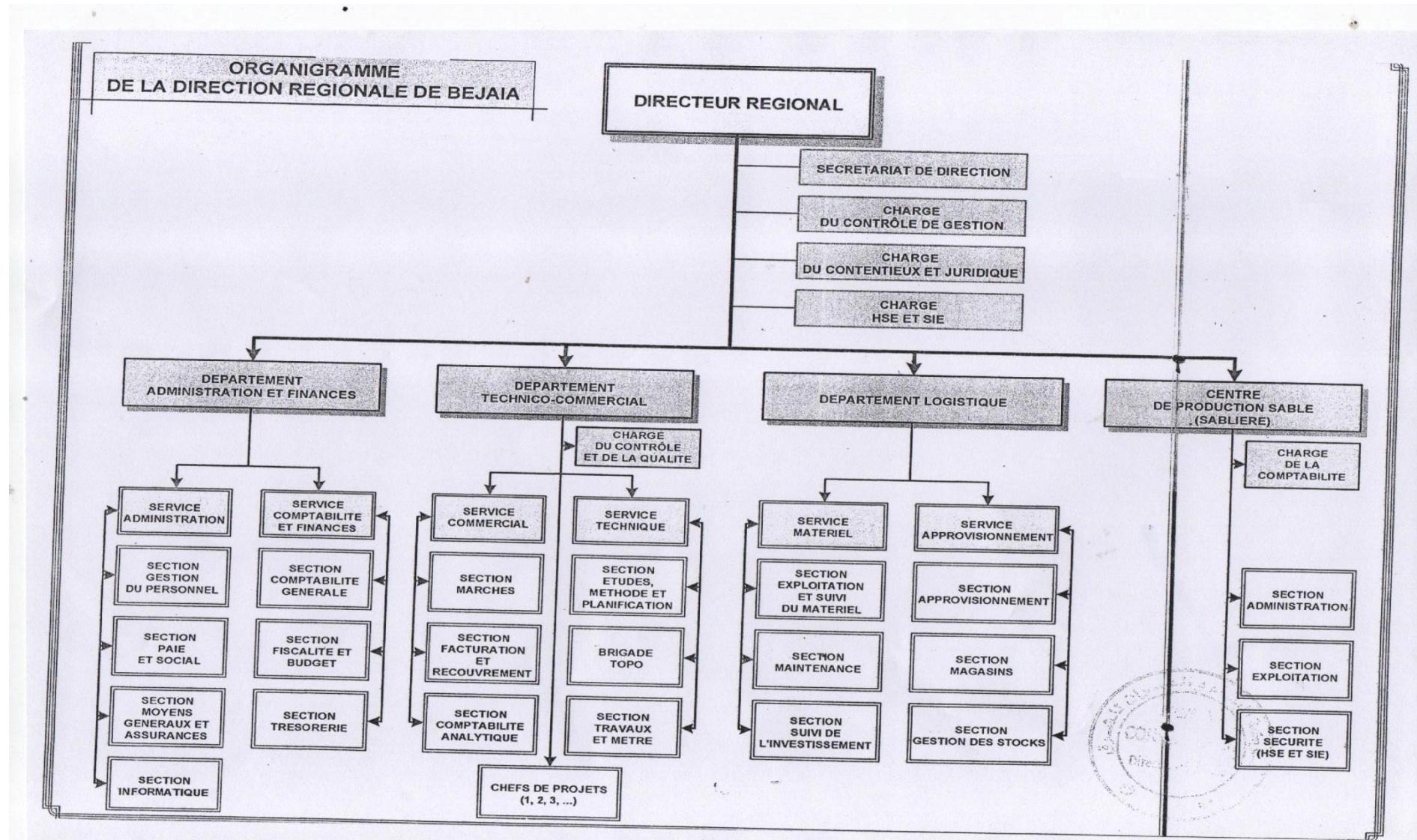
Béton Imprimé

Réhabilitation du parc visite de BMT SPA



Traitemennt de la surface de roulement en additif en FLOORPLATE à l'aide de l'hélicoptère

Annexe B – Organigramme de Construb-Est – Bejaia



Annexe C – SQL de la base de données ce_btph_db

La figure suivante montre le code SQL de notre base de données :

```

1  -- Table Service
2  CREATE TABLE Service (
3      id SERIAL PRIMARY KEY,
4      nom VARCHAR(100) NOT NULL,
5      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
6      updated_at TIMESTAMP,
7      deleted_at TIMESTAMP DEFAULT NULL,
8      is_deleted BOOLEAN DEFAULT FALSE
9  );
10
11 -- ENUM pour les rôles
12 CREATE TYPE role_utilisateur AS ENUM (
13     'Administrateur',
14     'Directeur',
15     'Chef_Departement',
16     'Chef_Projet'
17 );
18
19 -- Table Utilisateur
20 CREATE TABLE Utilisateur (
21     id SERIAL PRIMARY KEY,
22     nom VARCHAR(100) NOT NULL,
23     prenom VARCHAR(100) NOT NULL,
24     email VARCHAR(150) UNIQUE NOT NULL,
25     identifiant VARCHAR(100) UNIQUE NOT NULL,
26     mot_de_passe TEXT NOT NULL,
27     role role_utilisateur NOT NULL,
28     service_id INTEGER REFERENCES Service(id) ON DELETE SET NULL,
29     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
30     updated_at TIMESTAMP,
31     deleted_at TIMESTAMP DEFAULT NULL,
32     is_deleted BOOLEAN DEFAULT FALSE
33 );
34
35 -- Tables spécialisées des rôles (héritage par association)
36 CREATE TABLE Administrateur (
37     id INTEGER PRIMARY KEY REFERENCES Utilisateur(id) ON DELETE CASCADE
38 );
39
40 CREATE TABLE Directeur (
41     id INTEGER PRIMARY KEY REFERENCES Utilisateur(id) ON DELETE CASCADE
42 );
43
44 CREATE TABLE Chef_Departement (
45     id INTEGER PRIMARY KEY REFERENCES Utilisateur(id) ON DELETE CASCADE
46 );
47
48 CREATE TABLE Chef_Projet (
49     id INTEGER PRIMARY KEY REFERENCES Utilisateur(id) ON DELETE CASCADE
50 );
51
52 -- Table Session
53 CREATE TABLE Session (
54     id SERIAL PRIMARY KEY,
55     date_connexion TIMESTAMP NOT NULL,
56     date_deconnexion TIMESTAMP,
57     utilisateur_id INTEGER REFERENCES Utilisateur(id) ON DELETE CASCADE,
58     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
59     updated_at TIMESTAMP,
60     deleted_at TIMESTAMP DEFAULT NULL,
61     is_deleted BOOLEAN DEFAULT FALSE
62 );
63

```

Figure – Code SQL de la base de données ce_btph_db (Partie 1)

```

64  -- Table Session_Log
65  ✓ CREATE TABLE Session_Log (
66      id SERIAL PRIMARY KEY,
67      date_evenement TIMESTAMP NOT NULL,
68      evenement VARCHAR(100) NOT NULL,
69      description TEXT,
70      session_id INTEGER REFERENCES Session(id) ON DELETE CASCADE,
71      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
72      updated_at TIMESTAMP,
73      deleted_at TIMESTAMP DEFAULT NULL,
74      is_deleted BOOLEAN DEFAULT FALSE
75  );
76
77  -- Table Vue
78  ✓ CREATE TABLE Vue (
79      id SERIAL PRIMARY KEY,
80      nom VARCHAR(100) NOT NULL UNIQUE,
81      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
82      updated_at TIMESTAMP,
83      deleted_at TIMESTAMP DEFAULT NULL,
84      is_deleted BOOLEAN DEFAULT FALSE
85  );
86
87  -- Table Permission
88  ✓ CREATE TABLE Permission (
89      utilisateur_id INTEGER REFERENCES Utilisateur(id) ON DELETE CASCADE,
90      vue_id INTEGER REFERENCES Vue(id) ON DELETE CASCADE,
91      "create" BOOLEAN DEFAULT FALSE,
92      "read" BOOLEAN DEFAULT FALSE,
93      "update" BOOLEAN DEFAULT FALSE,
94      "delete" BOOLEAN DEFAULT FALSE,
95      PRIMARY KEY (utilisateur_id, vue_id)
96  );
97
98  -- Table Projet
99  ✓ CREATE TABLE Projet (
100     id SERIAL PRIMARY KEY,
101     nom VARCHAR(200) NOT NULL,
102     description TEXT,
103     date_debut DATE NOT NULL,
104     date_fin_prevue DATE,
105     date_fin_reelle DATE,
106     statut VARCHAR(50),
107     chef_projet_id INTEGER REFERENCES Chef_Projet(id) ON DELETE SET NULL,
108     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
109     updated_at TIMESTAMP,
110     deleted_at TIMESTAMP DEFAULT NULL,
111     is_deleted BOOLEAN DEFAULT FALSE
112  );
113
114  -- Table Tache
115  ✓ CREATE TABLE Tache (
116      id SERIAL PRIMARY KEY,
117      titre VARCHAR(200) NOT NULL,
118      description TEXT,
119      date_debut DATE NOT NULL,
120      date_echeance DATE,
121      date_fin DATE,
122      budget_prevu NUMERIC(12, 2),
123      budget_reel NUMERIC(12, 2),
124      statut VARCHAR(50),
125      projet_id INTEGER REFERENCES Projet(id) ON DELETE CASCADE,
126      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
127      updated_at TIMESTAMP,
128      deleted_at TIMESTAMP DEFAULT NULL,
129      is_deleted BOOLEAN DEFAULT FALSE
130  );
131
132  -- Table Suivi
133  ✓ CREATE TABLE Suivi (
134      id SERIAL PRIMARY KEY,
135      date_suivi DATE NOT NULL,
136      avancement TEXT,
137      probleme TEXT,
138      suggestion TEXT,
139      projet_id INTEGER REFERENCES Projet(id) ON DELETE CASCADE,
140      chef_projet_id INTEGER REFERENCES Chef_Projet(id) ON DELETE SET NULL,
141      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
142      updated_at TIMESTAMP,
143      deleted_at TIMESTAMP DEFAULT NULL,
144      is_deleted BOOLEAN DEFAULT FALSE
145  );
146

```

Figure – Code SQL de la base de données *ce_btph_db* (Partie 2)

```

147 -- Table Suivi_Journalier
148 ↘ CREATE TABLE Suivi_Journalier (
149     id INTEGER PRIMARY KEY REFERENCES Suivi(id) ON DELETE CASCADE,
150     date DATE NOT NULL
151 );
152
153 -- Table Suivi_Hebdomadaire
154 ↘ CREATE TABLE Suivi_Hebdomadaire (
155     id INTEGER PRIMARY KEY REFERENCES Suivi(id) ON DELETE CASCADE,
156     semaine VARCHAR(20) NOT NULL
157 );
158
159 -- Table Suivi_Mensuelle
160 ↘ CREATE TABLE Suivi_Mensuelle (
161     id INTEGER PRIMARY KEY REFERENCES Suivi(id) ON DELETE CASCADE,
162     mois VARCHAR(20) NOT NULL
163 );
164
165 -- Table Suivi_Tache
166 ↘ CREATE TABLE Suivi_Tache (
167     suivi_id INTEGER REFERENCES Suivi(id) ON DELETE CASCADE,
168     tache_id INTEGER REFERENCES Tache(id) ON DELETE CASCADE,
169     avancement TEXT,
170     budget NUMERIC(12,2),
171     PRIMARY KEY (suivi_id, tache_id)
172 );
173

```

-- Cette partie a été insérée pour créer un utilisateur administrateur

```

174 -- Insertion de l'utilisateur : identifiant=admin et mot_de_passe=admin123
175 ↘ INSERT INTO utilisateur (
176     id,
177     nom,
178     prenom,
179     email,
180     identifiant,
181     mot_de_passe,
182     role,
183     service_id
184 ) VALUES (
185     1,
186     'KRIM',
187     'Yanis',
188     'krim.yanis@construbest.dz',
189     'admin',
190     '$2b$10$6jxX4cgCg6SuPWSR.1W4i.6HB.JzN08c7G8JUYgMDzCDEKs4gSR6y',
191     'Administrateur',
192     NULL -- ou un service_id existant si souhaité
193 );
194
195 -- Insertion dans la table administrateur
196 ↘ INSERT INTO administrateur (id)
197     VALUES (1);

```

Figure – Code SQL de la base de données *ce_btph_db* (Partie 3)