

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

EN : Informatique

Spécialité: Sciences et Technologies de l'Information et de la Communication
(STIC)

*Développement d'une solution entreprise pour la
gestion en temps réel des équipes de travail*

Préparé par : Boukahoul Rokia
Bounahas khadidja

Devant le jury :

Guetiche Mourad
Boukhechem Nadir
Boubakir Mohamed

C.U.Abd Elhafid Boussouf Président
C.U.Abd Elhafid Boussouf Rapporteur
C.U.Abd Elhafid Boussouf Examineur

Année Universitaire : 2017/2018

Remerciements

Nous remercions tout d'abord Allah le tout Puissant de nous avoir aidé et donné la patience et le Courage durant ces longues et dures années d'études,

En second lieu, nous tenons à remercier notre Encadreur

*Mr : « **BoukhechemNadhir** »*

Pour son aide et ses encouragements et

Surtout pour ses précieux conseils

Qui nous ont assistés pour réaliser ce travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont Porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs Propositions.

Nous tenons à remercier L'ensemble de nos professeurs de Master 2

*STIC, particulièrement Monsieur « **BencheikhelhocineMadjed** »*

Toute personne ayant de près ou de loin contribué à

L'élaboration de ce travail un grand merci.

Rokia&Khadidja

Dédicace

Je remercie dieu qui a toujours été à mes côtés

Je dédie ce travail à

Mon père et ma mère qui ont toujours cru en moi et ont mis à ma disposition tous les moyens nécessaires et me encourager pour que je réussisse dans mes études.

A mes frères et mes sœurs.

A tout ma famille.

A mon fiancé Rochdi et toute sa famille

En reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d'atteindre cette étape de ma vie.

Spécial dédicace à ma binôme ou je dis ma sœur Khadidja

A tous les membres de ma promotion.

A tous mes chères amies.

Rokia

Dédicace

*Merci Allah (mon dieu) de m'avoir donné la capacité
d'écrire et de réfléchir, la force d'y croire, la patience*

D'aller jusqu'au bout du rêve et le bonheur.

*Je dédie ce modeste travail à celle qui m'a donné la vie,
Le symbole de tendresse, qui s'est sacrifiée pour mon bonheur
et ma réussite, à ma mère ...*

*A mon père qui a été mon ombre durant toutes les années des études,
et qui a veillé tout au long de ma vie à m'encourager,*

A mes frère et mes sœurs

A toute Ma famille

A mes amis

Spécial dédicac a ma belle Rokia

*Enfin, je tiens à remercier tous ceux qui, de près ou de loin, ont
contribué à la réalisation de ce travail.*

Khadidja

Résumé

Le projet réalisé au cours de ce mémoire de fin d'études, consiste à développer un système de la gestion des équipes de travail en temps réel. L'objectif majeur de notre projet est de proposer une solution générique pour gérer en temps réel les équipes de travail des entreprises, ce qui permettra d'améliorer leurs performances. Notre solution est composée de deux parties. La première est une application web destinée à l'administrateur de l'entreprise qui lui permet de gérer les équipes. La deuxième est une application Android qui sera utilisée par les chefs des équipes sur terrain.

Nous avons utilisé UML comme langage de modélisation, Laravel et Android studio pour la programmation, et MySQL comme système de gestion de bases de données.

Les mots clés : Solution entreprise, Gestion des équipes, temps réel, UML, Laravel, Android Studio, MySQL.

Summary

This project consists of developing a real-time work team management system. The main objective of our project is to propose a generic solution to manage the work-teams of the companies, which will improve their performances.

This solution has two parts. The first part is a web application for the administrator of the company that is allowed to manage teams. The second part is an Android application that will be used by the team-leader.

We used UML as modeling language, Laravel and Android studio for programming, and MySQL as database management system.

Key words: Enterprise solution, Team management, real time, UML, Laravel, Android Studio, MySQL.

Sommaire

Introduction Générale	
Chapitre 1 : Généralité	
Introduction	1
I. L'internet et le web	1
1. L'internet	1
2. Le Web	1
2.1. Une Page Web	1
2.2. Un site web	2
2.3 .Application Web	2
2.4. L'adresse d'une page web	2
II. Architecture client /serveur	3
1. Un serveur :	3
2. Fonctionnement de l'architecture client/serveur	3
3. Les types d'architecture client-serveur	4
3.1. Architecture à 2 niveaux (2-tiers)	4
3.2. Architecture à 3 niveaux (3-tiers)	5
3.3. Architecture multi niveaux :	5
3.4. Avantages de l'architecture client/serveur :	6
3.5. Inconvénients de l'architecture client/serveur	7
4. Les Technologies web :	7
4.1. Développement coté client :	7
4.2. Développement coté serveur :	9
4.2.1. Fonctionnement	10
III. Informatique mobile	11
1. définition	11
2..les smartphones	11
2.1. Les différentes plateformes pour Smartphones	12
2.2. Système d'exploitation	12
2.3. Système d'exploitation pour les mobiles	12
3. Les Applications mobiles	15
3.1. Définition	15
3.2. Avantages d'une application mobile par rapport à application web	16
4. découverte de la plateforme Android	16
4.1. Présentation d'Android	16
4.2. Les versions d'Android	17

4.3. Architecture d'Android	19
4.4. Pourquoi nous avons choisi Android ?	20
conclusion	21
Chapitre 2 : analyse du système	
I. Etude préliminaire	22
Introduction	22
1. L'élaboration du cahier de charges	22
1.1.présentation du projet	22
1.2. Recueil des besoins fonctionnels	23
1.3. Les besoin fonctionnels de l'administrateur	23
1.4. Les besoins fonctionnels des utilisateurs (au niveau des Smartphones)	23
1.5. Recueil des besoins non fonctionnels	23
2. Description du contexte	24
2.1.Identifications des acteurs	24
2.1.1. Au niveau de l'administrateur	24
2.1.2. Au niveau du Smartphones	25
2.2. Identifications des messages	25
2.2.1. Les messages émis par le système	25
2.2.2. Les messages reçus	25
2.3. Le diagramme de contexte du système	26
3. Le diagramme de cas d'utilisation pour les besoins fonctionnels	27
3.1.Au niveau de l'administrateur	28
3.2. Au niveau de l'utilisateur Smartphone	29
4. Description détaillée des différents cas d'utilisation	29
4.1. Au niveau de l'administrateur	29
Conclusion	52
Chapitre 3 : conception	
Introduction	53
1. Dictionnaire des données	53
1.1. Les classes et les attributs :	53
1.2. Les opérations	55
1.3.Diagramme de classe détaillée	57
2. Passage vers le model relationnel	58
2.1. Les règles de passages	58
2.2. Les tables de la base de données	58
3. Diagramme de séquence détaillé	59
Conclusion	83
Chapitre 4 : réalisation	
Introduction	84
1. Présentation des outilles et technologies utilisée	84
1.1. Développement du coté web de l'application	84
1.1.1. Le Framework de développement Laravel	84
1.1.2 .MySQL	85

1.1.3. Le Framework Bootstrap	85
1.1.4. JQuery	86
1.1.5. La technologie Ajax :	86
1.2.Développement de la partie Smartphone (Application pour Android)	86
1.2.1. Langage de programmation java	86
1.2.2. Le SDK Android	87
1.2.3. L'environnement de développement Android Studio	87
2. Les différentes interfaces de l'application	87
2.1. Au niveau du serveur	87
2.2. Au niveau du Smartphone	100
Conclusion	104
Conclusion générale	
Références	

Liste des figures

Chapitre1 : Généralité	
Figure 1: Fonctionnement de l'architecture client/serveur	4
Figure 2: architecture 2 tiers	4
Figure 3: architecture 3 tiers	5
Figure 4: Architecture multi niveaux	6
Figure 5: les requêtes http.	8
Figure 6 : Les deux modes de fonctionnement d'un site Web: client-serveur et AJAX	9
Figure 7: Fonctionnement du PHP.	10
Figure 8: exemple de smartphone	11
Figure 9: logo d'Android	13
Figure 10: logo d'Ios	13
Figure 11: logo Windows Phone	14
Figure 12: logo de BlackBerry	15
Figure 13: exemple de logos d'application mobil.	15
Figure 14: Architecture Android	19
Chapitre 02 : analyse	
Figure 1: Diagramme du cas d'utilisation Au niveau de l'administrateur.	28
Figure 2: Diagramme du cas d'utilisation au niveau de l'utilisateur Smartphone (chef d'équipe).	29
Figure 3: Diagramme de séquence du cas d'utilisation « S'authentifier ».	30
Figure 4 : Diagramme de séquence du cas d'utilisation « Gestion des équipes ».	32
Figure 5 : Diagramme de séquence du cas d'utilisation « Ajouter équipe ».	34
Figure 6: Diagramme de séquence du cas d'utilisation « Modifier équipe ».	35
Figure 7: Diagramme de séquence du cas d'utilisation « Supprimer équipe ».	36
Figure 8: Diagramme de séquence du cas d'utilisation « Afficher équipe ».	37
Figure 9: Diagramme de séquence du cas d'utilisation « Gestion des tâches ».	39
Figure 10: Diagramme de séquence du cas d'utilisation « Ajouter tâche ».	41
Figure 11: Diagramme de séquence du cas d'utilisation « Modifier tâche ».	42
Figure 12: Diagramme de séquence du cas d'utilisation «supprimer tâche».	43
Figure 13: Diagramme de séquence du cas d'utilisation «afficher tâche ».	44
Figure 14: Diagramme de séquence du cas d'utilisation «communication ».	45
Figure 15: Diagramme de séquence du cas d'utilisation «envoyer message ».	46
Figure 16: Diagramme de séquence du cas d'utilisation «consulter messages ».	47
Figure 17: Diagramme de séquence du cas d'utilisation «Suivi d'avancement»	49
Figure 18: Diagramme de séquence du cas d'utilisation «affecter tâche»	50
Figure 19: Diagramme de séquence du cas d'utilisation «Annuler affectation»	51
Figure 20: Diagramme de séquence du cas d'utilisation «afficher l'état d'avancement»	52

Chapitre 03 : Conception	
Figure 1: Diagramme de classe détaillé.	57
Figure 2: Diagramme de séquence détaillé du cas « S’authentifier».	60
Figure 3: Diagramme de séquence détaillé du cas « gestion des équipes ».	61
Figure 4: Diagramme de séquence détaillé du cas « ajouter équipe ».	62
Figure 5: Diagramme de séquence détaillé du cas « modifier équipe ».	63
Figure 6: Diagramme de séquence détaillé du cas « supprimer équipe »	64
Figure 7: Diagramme de séquence détaillé du cas « afficher équipe ».	65
Figure 8: Diagramme de séquence détaillé du cas « gestion des tâches ».	66
Figure 9: Diagramme de séquence détaillé du cas « ajouter tâche».	67
Figure 10: Diagramme de séquence détaillé du cas «modifier tâche ».	68
Figure 11: Diagramme de séquence détaillé du cas «supprimer tâche ».	69
Figure 12: Diagramme de séquence détaillé du cas « afficher tâche ».	70
Figure 13: Diagramme de séquence détaillé du cas « gestion des employés ».	71
Figure 14: Diagramme de séquence détaillé du cas « ajouter employé ».	72
Figure 15: Diagramme de séquence détaillé du cas « modifier employé ».	73
Figure 16: Diagramme de séquence détaillé du cas d’utilisation « afficher employé».	74
Figure 17: Diagramme de séquence détaillé du cas « supprimer employé».	75
Figure 18: Diagramme de séquence détaillé du cas « communication ».	76
Figure 19: Diagramme de séquence détaillé du cas «envoyer message».	77
Figure 20: Diagramme de séquence détaillé du cas « consulter messages ».	78
Figure 21: Diagramme de séquence détaillé du cas « suivi d’avancement ».	79
Figure 22: Diagramme de séquence détaillé du cas «affecter tâche»	80
Figure 23: Diagramme de séquence détaillé du cas «annuler affectation»	81
Figure 24: Diagramme de séquence détaillé du cas «afficher état d’avancement»	82
Chapitre 04 : Réalisation	
Figure1 : logo de laravel	85
Figure2 : logo Bootstrap	86
Figure 3: Logo de l’IDE Android Studio.	87
Figure 4 : Fenêtre d’authentification.	88
Figure 5 : Fenêtre d’enregistrement.	88
Figure 6 : Fenêtre page d’accueil.	89
Figure 7: Fenêtre de gestion des équipes.	89
Figure 8 : Fenêtre ajouter équipe.	90
Figure 9 : Fenêtre modifier équipe.	90
Figure 10 : Fenêtre afficher équipe.	91
Figure 11 : Fenêtre supprimer équipe.	91

Figure 12 : Fenêtre gestion des taches.	92
Figure 13 :Fenêtre ajouter tache.	92
Figure 14 : Fenêtre modifier tâche.	93
Figure 15 : Fenêtre affecter tâche.	93
Figure 16 : Fenêtre annuler l'affectation.	94
Figure 17 : Fenêtre gestion des employées.	94
Figure 18 : Fenêtre ajouter employée.	95
Figure 19 : Fenêtre afficher employé.	95
Figure 20 : Fenêtre modifier employé.	96
Figure 21 : Fenêtre supprimer employé.	96
Figure 22 : Fenêtre afficher l'état d'avancement.	97
Figure 23 : Fenêtre communication.	97
Figure 24 : Fenêtre envoyer message.	98
Figure 25 : Fenêtrereconsulter message.	98
Figure 26 : Fenêtre supprimer message.	99
Figure 27 : Fenêtre évaluer rendement des équipes.	99
Figure 28 : Fenêtre de démarrage.	100
Figure 29 : Fenêtre d'authentification.	100
Figure 30 : Fenêtre d'accueil.	101
Figure 31 : Fenêtre consulter la liste des taches.	101
Figure 32 : Fenêtre afficher tache.	102
Figure 33 : Consulter la liste des messages.	103
Figure 34 : Afficher message.	103
Figure 35 : Envoyer message.	104

Liste de tableaux

Chapitre 01 : généralité	
Tableau 1 : Les versions d'Android	19
Chapitre 02 : Analyse	
Tableau1 : Légende des messages du diagramme de contexte dynamique	27
Chapitre 03 : conception	
Tableau 1 : Dictionnaire de données avec les classes et les attributs.	55
Tableau 2 : Dictionnaire de donnée avec les opérations	56

Introduction générale

[Introduction générale]

Nul doute que l'informatique est la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité moderne. En effet, les logiciels informatiques proposent maintenant des solutions à tous les problèmes de la vie, aussi bien dans des domaines professionnels que pour des applications personnelles. Leurs méthodes de conception et de développement facilitent leur mise en place et donnent des possibilités et des fonctionnalités de plus en plus étendues.

L'évolution de l'informatique et l'apparition de l'internet, offre des possibilités considérables et accroît les possibilités de communication. De nos jours, les entreprises cherchent à améliorer leurs services grâce à des technologies de communication comme les Smartphones et les tablettes qui sont connectés à internet, ces entreprises peuvent ainsi gérer et organiser leurs travaux à distance.

En Algérie l'utilisation des nouvelles technologies est peu présente dans les entreprises. En effet plusieurs grandes entreprises tel que Sonalgaz, Algérie post, Algérie télécoms, qui possèdent des moyens considérables utilisent toujours des outils de gestion traditionnels tel que le papier et le téléphone. Un exemple concret, est leurs façons de gérer les équipes de travail. En effet ces entreprises possèdent plusieurs équipes qui interviennent dans des emplacements différents, ces équipes sont habituellement gérées par téléphone. Cette méthode de gestion est lourde et inefficace. Pour remédier à ces problèmes nous avons développé dans le cadre de ce mémoire une solution pour gérer en temps réel les équipes de travail des entreprises, ceci permettra d'améliorer considérablement leurs performances. La solution proposée permet de :

- Faciliter la communication entre le gestionnaire de l'entreprise et les équipes de travail,
- Affecter les tâches aux équipes en prenant en compte leurs emplacements,
- Envoyer des ordres de missions aux équipes,
- Suivre en temps réel l'état d'avancement des travaux,
- Evaluer le rendement des équipes de travail.

Ce mémoire est structuré comme suit :

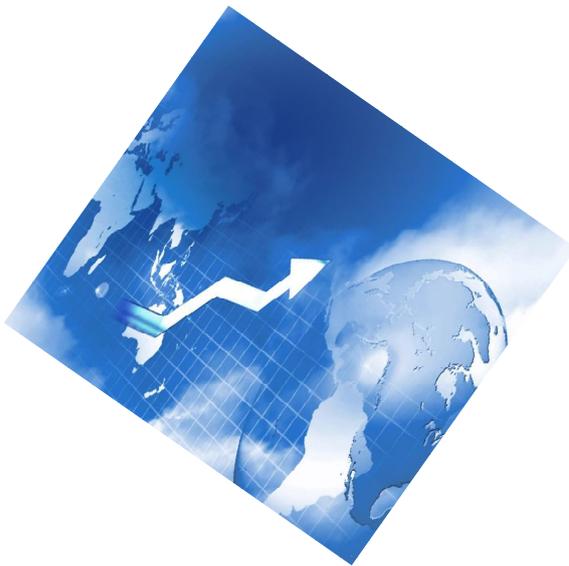
Dans le chapitre 1 nous présentons quelques technologies récentes utilisées dans le développement des applications web. Les chapitres 2 et 3 sont consacrés respectivement à l'analyse et la conception de l'application. Le chapitre 4 présente l'interface de notre

[Introduction générale]

application ainsi que les technologies utilisées pour l'implémentation. Enfin nous terminons par une conclusion générale.

CHAPITRE 01

G•n•ralit•



Introduction

Nul doute que l'informatique est la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité moderne. En effet, les logiciels informatiques proposent maintenant des solutions pour tous les problèmes de la vie, aussi bien dans des domaines professionnels que personnels.

Dans ce chapitre introductif nous allons présenter tous les technologies de l'informatique.

I. L'internet et le web

1. L'internet

C'est une infrastructure composé d'un ensemble d'ordinateur reliés sur laquelle repose des services tel que :

- ✓ Transfert de mail.
- ✓ Transfert de fichiers.
- ✓ Service DNS.
- ✓ Service de téléphonie (VOIP).
- ✓ Service de vidéo conférence.
- ✓ Le **Web** (World Wide Web).
- ✓

2. Le Web :

C'est un système qui permet de publier et consulter des pages web et des ressources (images vidéo... etc.) sur l'internet. Le web se compose d'un ensemble de pages web et ressources reliés par des liens appelés **hyperliens**. [1]

2.1. Une Page Web :

La page web désigne l'unité élémentaire d'un site web, Pour les internautes, la page web est accessible via un navigateur web (Firefox, Mozilla, Internet Explorer, etc.). Elle dispose d'une adresse web spécifique qui peut être saisie directement dans la barre d'adresse de ces navigateurs, être retrouvée via un moteur de recherche/la page d'accueil d'un site web, ou bien encore en suivant un lien hypertexte présent sur le site ou un autre site web. Sur le plan technique, la page web se résume la plupart du temps à un fichier HTML (Hyper text Markup Language) et un ensemble d'autres ressources comme des images, des vidéos, des sons, des animations, etc. [9]

2.2. Un site web :

Un site web est un ensemble de pages web visualisables dans un navigateur. Ces pages web sont reliées entre elles par des liens qui permettent de passer de l'une à l'autre. En règle générale, on reconnaît un site web à l'homogénéité du design de ses pages. L'ensemble des pages d'un site web est en général accessible sous une adresse au même nom de domaine. [2]

2.3 .Application Web :

C'est un logiciel (interaction, traitement, base de données... etc.) hébergé dans un serveur web, et qui utilise des pages web comme interface. On accède à l'application web en utilisant un navigateur Web. L'avantage des applications web est qu'on n'a pas besoin d'installer l'application sur les PC des utilisateurs, ils utilisent un simple navigateur web pour accéder à l'application. [5]

2.4. L'adresse d'une page web:

Chaque ressource sur le Web est localisée par une adresse Web :

L'adresse d'une page internet aussi nommée URL (Uniform Ressource Locator), est composée de plusieurs éléments :

- ✓ Le premier indique le protocole utilisé dans la conversation qui va s'instaurer entre le demandeur et le serveur : le plus fréquent étant http://
- ✓ Le second indique le **nom de domaine** à interroger et son éventuel préfix par exemple : www.google.fr .[6]

Pour résumer l'adresse typique d'une page internet est la suivante :

Le Protocole	Identifiant et Mot de passe (Optionnel)	Nom du serveur	Port (optionnel si le port = 80)	Chemin de la ressource

II. Architecture client /serveur :

Définition : De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc... .

Les services sont exploités par des programmes, appelés « programmes clients», s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique).[10]

1. Un serveur :

à l'intérieur des machines fonctionne un logiciel serveur. On appelle logiciel serveur un programme qui offre un service réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. Pour pouvoir offrir ces services, le serveur logiciel doit être sur un site accessible et s'exécuter en permanence. On distingue plusieurs types de serveurs :

- Serveur de fichier.
- Serveur d'application.
- Serveur de messagerie.
- Serveur web.
- Serveur base de données.

Les sites et les applications Web sont hébergés (stockés) dans des machines appelés Serveurs Web (web serveur).

On appelle serveur Web :

- ✓ Le programme qui gère les requêtes Web par exemple Apache, Jboss, IIS, nginx...etc.

- ✓ Le matériel qui stocke les pages du site ou de l'application Web.

2. Fonctionnement de l'architecture client/serveur :

L'architecture client/serveur fonctionne selon les étapes suivantes :

- ❖ le client émet une requête vers le serveur grâce à son adresse IP et le port qui désigne un service particulier du serveur.
- ❖ le serveur reçoit la demande et répond à l'aide de l'adresse IP de la machine cliente et son port.

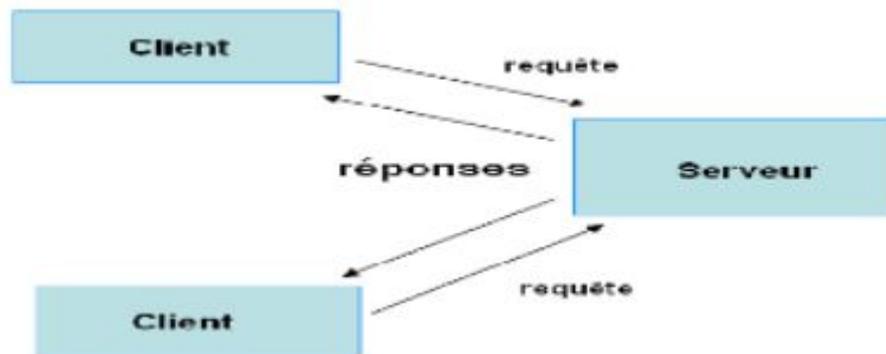


Figure 01 : Fonctionnement de l'architecture client/serveur

3. Les types d'architecture client-serveur :

3.1. Architecture à 2 niveaux (2-tiers) :

Si toutes les ressources nécessaires sont présentes sur un seul serveur, on parle d'architecture à deux niveaux ou 2 tiers (1 client + 1 serveur). [10]



Figure 2- architecture 2 tiers

3.2. Architecture à 3 niveaux (3-tiers) :

Si certaines ressources sont présentes sur un deuxième serveur (par exemple des bases de données), on parle d'architecture à trois niveaux ou 3 tiers (1 client interroge le premier serveur qui lui-même interroge le deuxième serveur). [10].

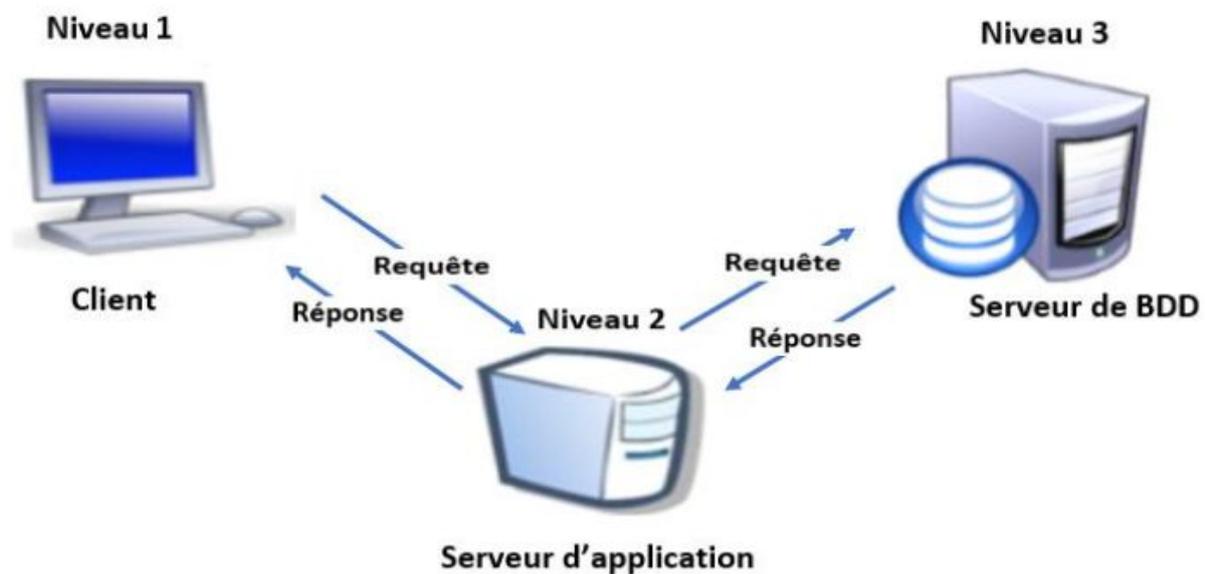


Figure 3 –architecture 3 tiers

3.3. Architecture multi niveaux :

Dans cette architecture, un serveur utilise les services de plusieurs autres serveurs afin de fournir son propre service. C'est pour ce fait que les experts l'ont appelée architecture à N-tiers ou le N correspond au nombre de services utilisés.

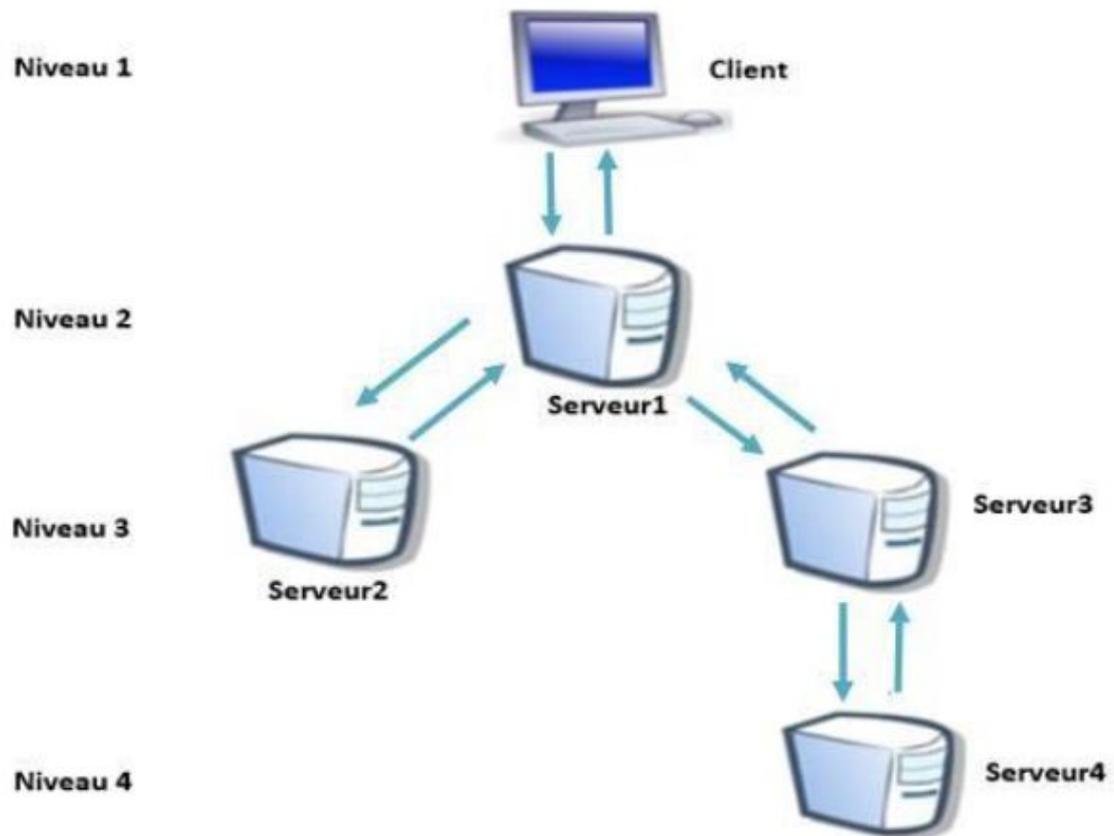


Figure 4 : Architecture multi niveaux

3.4. Avantages de l'architecture client/serveur :

- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- Les technologies supportant l'architecture client/serveur sont plus matures que les autres.
- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Recherche d'information : les serveurs étant centralisés, cette architecture est particulièrement adaptée pour retrouver et comparer de vaste quantité d'informations (moteur de recherche sur le Web).

3.5. Inconvénients de l'architecture client/serveur :

- Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair-à-pair fonctionnent mieux en ajoutant de nouveaux participants).
- Les coûts de mise en place et de maintenance peuvent être élevés.
- En aucun cas les clients ne peuvent communiquer entre eux, il faut toujours passer par le serveur.

4. Les Technologies web :

4.1. Développement coté client :

✓ **HTML** (*Hyper Text Markup Language*):

Langage de balisage hypertexte ou html est le langage utilisé pour décrire et définir le contenu d'une page web. [1]

✓ **CSS** (*Cascading Style Sheets*) :

Feuilles de style en cascade ou CSS est utilisé pour décrire l'apparence du contenu d'une page web.

✓ **HTTP** (*Hypertext Transfer Protocol*):

Pour communiquer avec un serveur Web et obtenir les ressources, un client web utilise le protocole http. Ce dernier est un protocole de la couche application, orienté client-serveur, qui permet le transfert de documents web tels que des documents HTML. [1].

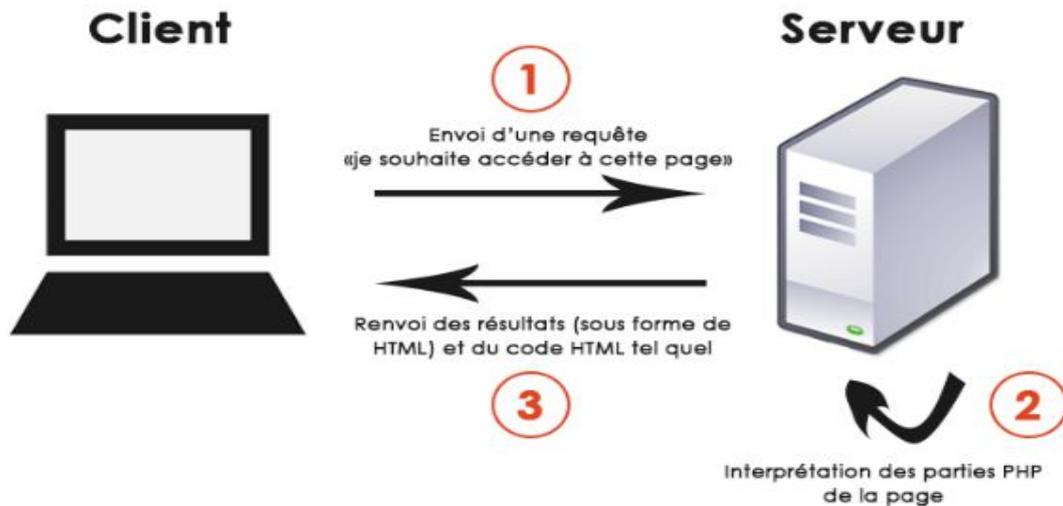


Figure 5- les requêtes http.

✓ **JavaScript:**

C'est un langage de programmation de scripts qui est exécuté principalement par le navigateur Web sur les machines clientes. JavaScript permet de donner du dynamisme au Web.[16]

✓ **XHTML** (Extensible HyperText MarkupLanguage) :

C'est un langage à base de balises servant à écrire des pages Web mais d'une manière plus rigoureuse que HTML, ce qui permet une meilleure portabilité des

Pages Web. [1]

✓ **Ajax** (AsynchroneJavaScript and XML):

Lorsque un utilisateur navigue de page en page sur un site Web traditionnel (entendez par là non-AJAX), les actions de l'internaute se traduisent par les actions suivantes:

1. Envoi d'une requête au serveur afin d'obtenir une nouvelle page
2. Calcul de la nouvelle page par le serveur et envoi des données HTML/CSS correspondantes.
3. Affichage de ces données dans le navigateur.

Cette technique fonctionne très bien dans la plupart des cas, mais parfois seule une partie de la page nécessite d'être mise à jour. C'est là qu'intervient **AJAX** :

1. Dans un premier temps, envoi d'une requête au serveur afin d'obtenir les données qui seront affichées dans une partie bien précise de la page actuelle.
2. Calcul des données demandées par le serveur et envoi de ces données au navigateur au format XML.
3. Réception des données envoyées par le programme (on dit aussi moteur) AJAX qui les a demandées et affichage dans un endroit bien précis de la page actuelle sans toucher au reste de la page.[8]

La figure suivante résume ces deux modes de fonctionnement :

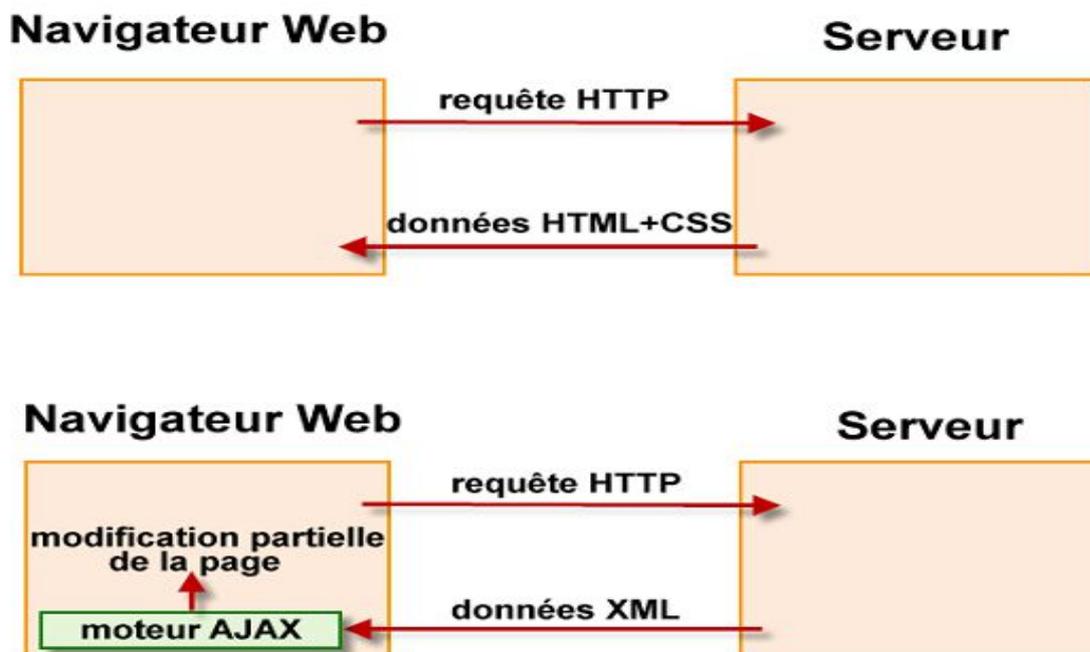


Figure 6- Les deux modes de fonctionnement d'un site Web : client-serveur et AJAX

4.2. Développement coté serveur :

✓ Le langage PHP (Hyper textPreprocessor) :

Le langage PHP est un langage script dont le code est interprété et exécuté sur le serveur. Le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques.

PHP est utilisable sur la majorité des systèmes d'exploitation, comme Linux, Microsoft Windows, Mac OS Xetc.

PHP supporte aussi la plupart des serveurs web actuels : Apache, Microsoft Internet Information Serveretc. [1]

Le schéma du fonctionnement du PHP est le suivant, toujours basé sur une requête client qui sera traitée coté serveur **[Figure 7] :**

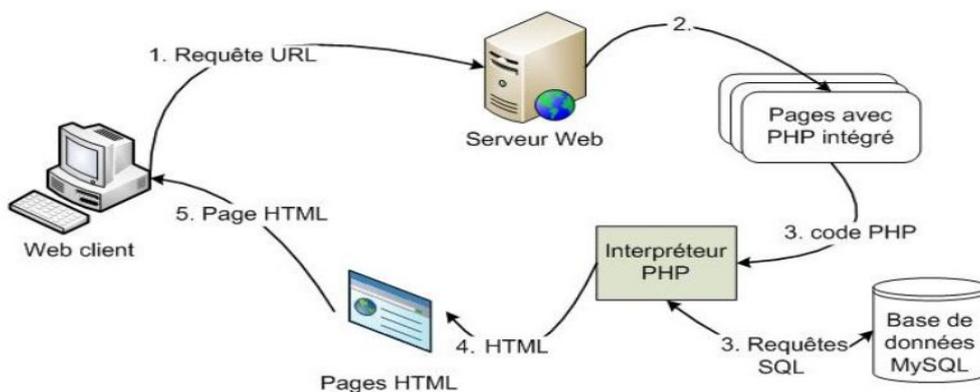


Figure 7- Fonctionnement du PHP.

4.2.1. Fonctionnement :

Lorsqu'un visiteur demande à consulter une page Web son navigateur envoie une requête à un serveur http. Si la page contient du code PHP, l'interpréteur PHP du serveur le traite et renvoie du code généré(HTML). Ce langage permet donc de générer des pages au contenu dynamique différent en fonction des données passées en paramètre, souvent par le biais de formulaires), mais peut aussi être utilisé comme langage de programmation en ligne de commande sans utiliser de serveur http ni de navigateur. [1]

✓ **JSP** (Java Server Pages) :

C'est une technologie permettant d'insérer du code java dans un fichier HTML celui-ci prend l'extension «. **jsp** ».le code java dans le fichier jsp est interprété par le serveur Web. [1]

✓ **Servlet**:

C'est une classe java qui permet d'intercepter des requêtes HTTP et retourner du code HTML. [1]

III. Informatique mobile :

1. définition :

Le concept d'informatique mobile réfère à la capacité d'accéder à des services ou à des applications indépendamment de la localisation physique des utilisateurs, de leurs comportements ou de leurs mouvements. L'informatique mobile traite de l'ensemble des solutions informatiques développées sur des plateformes comme les Netbooks, PDA, Pc portable, les Tablette, les Smartphones etc.

2. les smartphones :

Ce figure représente un exemple sur des smartphon.



Figure 8: exemple de smartphone

Un Smartphone peut être aperçu de différentes façons. On peut le voir comme un PC de poche qui possède des fonctionnalités de téléphonie ou comme un téléphone qui possède des capacités de calcul. Un téléphone doté de capacités avancées proches de celles de l'ordinateur en termes de connectivité et de capacité de calcul. Les smartphone supportent un vrai système d'exploitation et sont non seulement équipés d'un module de radiocommunication pour la

voix et l'échange de données mais également de fonctions bureautiques (agenda, carnet d'adresses, prise de note, messagerie,...), de fonctions multimédia (photo, musique, vidéo, jeux, navigation sur internet) et de toutes sorte d'applications. Les Smartphones ont souvent des caractéristiques techniques très riches. La plupart sont maintenant équipés d'appareil photo numérique et de GPS et peuvent lire des mp3 et des vidéos ainsi que des fichiers dans divers formats (Word, PDF, ...).

2.1. Les différentes plateformes pour Smartphones :

On veut par une plateforme la base logicielle et matérielle sur laquelle repose les applications du Smartphone. La partie logicielle consiste principalement en un système d'exploitation et ses services.

2.2. Système d'exploitation :

Le système d'exploitation (SE, en anglais Operating System ou OS) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications informatiques de l'utilisateur (traitement de texte, jeu vidéo, etc.). Il fournit aux programmes applicatifs des points d'entrée génériques pour les périphériques.

2.3. Système d'exploitation pour les mobiles :

Les systèmes d'exploitation mobiles (OS) peuvent être définis comme les logiciels permettant à un smartphone ou un téléphone mobile basique de fonctionner. Ils permettent de ce fait aux utilisateurs de pouvoir passer un appel téléphonique, naviguer sur leurs téléphones parmi toutes les rubriques, télécharger des applications ou encore paramétrer et personnaliser leurs smartphones.

On distingue une vingtaine de système d'exploitation sur le marché nous présenterons ici uniquement les plus dominants :

- **Android de Google** : depuis maintenant de nombreuses années, Android, développé par les ingénieurs de Google, s'impose comme l'OS de référence du marché, loin devant ses concurrents. L'avantage de ce système d'exploitation se résume dans le fait qu'il est mis à disposition de tous les fabricants de téléphones, à condition qu'ils respectent certaines conditions liés à la mise en avant de services Google. Pour le reste, l'OS est personnalisable à souhait, par les consommateurs comme par les fabricants. Comme le démontre le graphique ci-dessous, de nos jours, la majorité des téléphones vendus dans le monde tournent sous Android. On peut ainsi citer de très nombreuses marques de smartphones, parmi lesquelles Samsung, Acer, HTC, LG, Sony ou encore Huawei.



Figure 9 : logo d'Android

- **iOS d'Apple** : avec Android, iOS de l'entreprise américaine Apple concentre une très large majorité des smartphones, les autres que nous vous présentons plus bas étant loin de les égaler. L'iOS d'Apple est connu et présenté par comme étant le système d'exploitation proposant la meilleure expérience utilisateur. Avec son flat design apparu à partir de l'iOS 7, Apple propose son logiciel uniquement sur ses propres produits. Il faut de ce fait passer par l'Apple Store pour installer sur son iPhone ou iPad les applications qui vous plaisent, parmi les centaines de milliers existantes. L'écosystème d'Apple rendant tous les produits de la marque compatibles entre eux, permet une ergonomie inégalée à ce jour, positionnant Apple comme fabricant éminent, et comme second plus gros système d'exploitation au monde en volume.



Figure 10 : logo d'iOS

- **Windows Phone de Microsoft** : Les smartphones tournant sous Windows Phone restent à ce jour limités face à Android, bien que toutefois présents en nombre. Microsoft, qui a pris le virage des smartphones en retard, souffre de son manque d'adoption par les constructeurs et consommateurs. Basé sur le design de Windows 8 pour PC, il n'est pas du goût de tout le monde. Il reste cependant aimé par les utilisateurs pour la possibilité de personnaliser de manière poussée la page d'accueil et sa simplicité de gestion des applis. Face aux géants que sont iOS et Android, il a toutefois de la peine à intéresser suffisamment les développeurs d'applications, certains d'entre eux préférant le délaissé au profit des OS beaucoup plus fréquentés, au grand déplaisir des utilisateurs. Mais Microsoft a déclaré que le projet Windows Phone a été abandonné.



Figure 11 : logo Windows Phone

- **BlackBerry OS de RIM** : ancienne star de la téléphonie, prisée par le monde professionnel et particulièrement des cadres d'entreprise, l'OS BlackBerry de RIM a perdu de ses parts de marché depuis la dernière décennie. Il n'empêche que les smartphones de la marque, avec leur design complètement revus, intéressent maintenant une partie ciblée des consommateurs. A l'attention des personnes avides de claviers physiques, BlackBerry OS est tourné avant tout vers la lecture des e-mails et SMS. Fait identifiable, les applications Android fonctionnent maintenant sous BlackBerry OS, rendant plus attrayant le système d'exploitation de RIM, en proposant un très large catalogue d'applications compatibles innovantes.



Figure 12 : logo de BlackBerry

3. Les Applications mobiles :

3.1. Définition :

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un Smartphone ou d'une tablette, etc. Les applications mobiles sont adaptées aux différents environnements techniques des Smartphones et à leurs contraintes et possibilités ergonomiques (écran tactile notamment). Elles permettent généralement un accès plus confortable et plus efficace à des sites ou services accessibles par ailleurs en versions mobile ou web. L'essentiel du temps passé sur les Smartphones est consacré à des applications mobiles.



Figure13- exemple de logos d'application mobil.

3.2. Avantages d'une application mobile par rapport à application web :

- Le principal avantage d'une application mobile comparé à une web-app ou à un site mobile est son ergonomie.
- Elle est de bien meilleure qualité et de ce fait, l'expérience utilisateur est supérieure comparée à celle obtenue avec une web-app ou un site mobile.
- Ceci vient du fait qu'en développant une application mobile, le format, la navigation et le contenu sont adaptés au smartphone.
- L'application permet également d'utiliser et d'intégrer les fonctionnalités du téléphone (accéléromètre, gyroscope, GPS, caméra...), ce qui peut rendre l'utilisation très riche.
- Autre avantage, pour fonctionner une application ne nécessite pas de connexion à internet.
- Une application mobile est également plus appréciée du grand public car il est plus facile de trouver une application qu'un site mobile avec les « stores » proposés par Apple, Google, BlackBerry, etc...
- Les applications permettent aussi d'alerter les mobinautes sur des événements via des notifications push.

4. Découverte de la plateforme Android :

4.1. Présentation d'Android :

Android est une plate-forme logicielle open source destinée aux appareils mobiles qui intègre différents composants : un système d'exploitation, Un middleware, une interface graphique et Un éventail d'applications. Android n'est donc pas lié à un appareil donné, un constructeur de téléphone mobile, ou un opérateur. Android laisse la possibilité aux développeurs de créer leurs propres applications à l'aide du langage de programmation Java, et Google fournit aux développeurs de nombreux outils pour écrire et mettre au point leurs programmes : émulateur pour les tests, SDK, et plugins pour le développement, ainsi qu'une documentation fournie.

Android repose sur un noyau Linux qui gère les services système comme la sécurité, la gestion mémoire, la gestion des processus, la pile réseau.

4.2. Les versions d'Android :

Le tableau suivant représente les différentes versions d'Android :

La version	Nom de version	Date de sortie	API level
Android 8.1	Oreo	05/12/2017	27
Android 8.0		21/08/2017	26
Android 7.1.2	Nougat	04/04/2017	25
Android 7.1.1		05/12/2016	25
Android 7.1		04/10/2016	25
Android 7.0	Nougat	22/08/2016	24
Android 6.0.1	Marshmallow	07/10/2015	23
Android 6		05/10/2015	23
Android 5.1.1	Lollipop	21/04/2015	22
Android 5.1		09/03/2015	22
Android 5.0.2		19/04/2014	21
Android 5.0.1		02/12/2014	21
Android 5.0		17/10/2014	21
Android 4.4.4		KitKat	23/06/2014
Android 4.4.3	14/04/2014		19
Android 4.4.2	09/12/2013		19
Android 4.4.1	05/12/2013		19
Android 4.4	21/10/2013		19
Android 4.3	Jelly Bean	24/07/2013	18
Android 4.2.2		11/02/2013	17

Android 4.2.1		27/11/2012	17
Android 4.1.2		09/10/2012	16
Android 4.1.1		23/07/2012	16
Android 4.1		09/07/2012	16
Android 4.0.4	Ice Cream Sandwich	28/03/2012	15
Android 4.0.3		16/12/2011	15
Android 4.0.2		28/10/2011	14
Android 4.0.1		19/10/2011	14
Android 3.2.6	Honeycomb	15/02/2012	13
Android 3.2.4	Honeycomb	15/12/2011	13
Android 3.2.2		30/09/2011	13
Android 3.2.1		20/09/2011	13
Android 3.2		15/07/2011	13
Android 3.1		10/05/2011	12
Android 3.0		22/02/2011	11
Android 2.3.7	Gingerbread	21/09/2011	10
Android 2.3.6		02/09/2011	10
Android 2.3.5		25/07/2011	10
Android 2.3.4		10/05/2011	10
Android 2.3.3		09/02/2011	10
Android 2.3		06/12/2010	9
Android 2.2	Froyo	20/05/2010	8
Android 2.1	Eclair	12/01/2010	7

Android 2.0.1		03/12/2009	6
Android 2.0		26/10/2009	5
Android 1.6	Donut	15/09/2009	4
Android 1.5	Cupcake	30/04/2009	3
Android 1.1	Banana bread	09/02/2009	2
Android 1.0	Apple pie	23/09/2008	1

Tableau 1 : les versions d'Android

4.3 . Architecture d'Android :

Ce schéma donne l'ensemble des fonctionnalités fournies par Android :

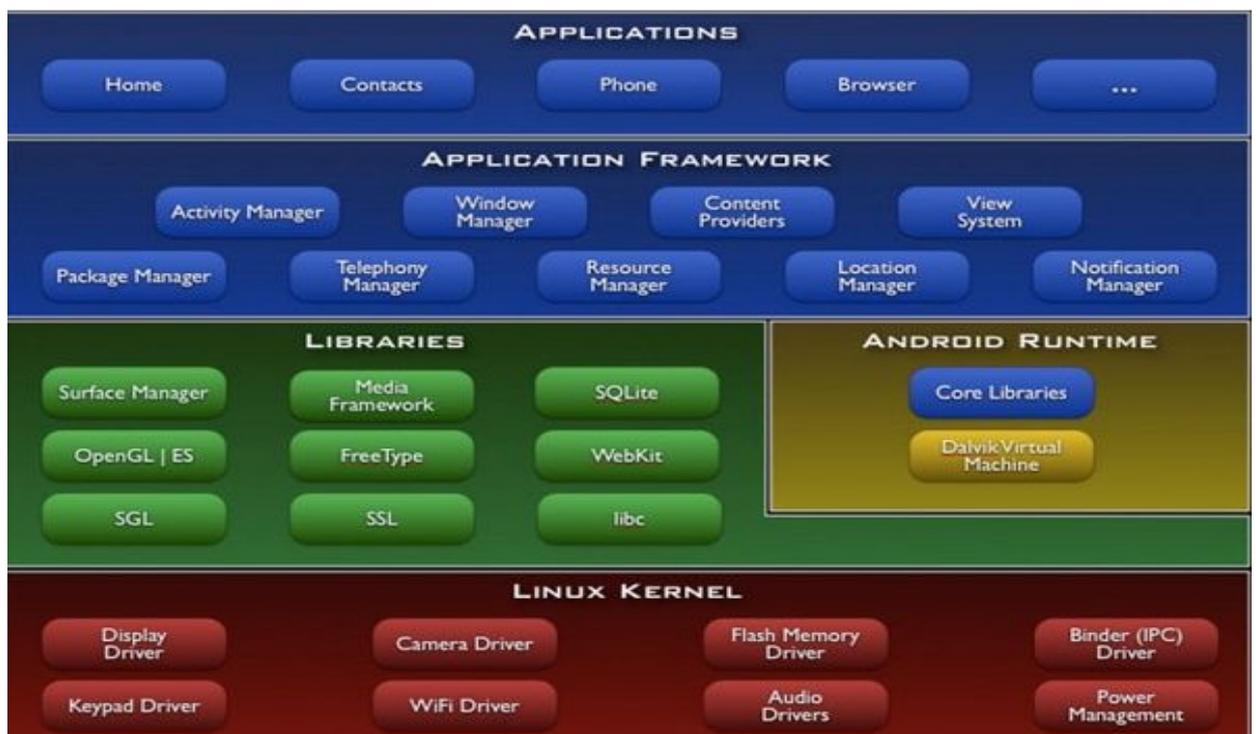


Figure 14 : Architecture Android

- ✓ **Application :** Android est fourni avec un ensemble de programmes de base (dites aussi applications natives) permettant d'accéder à des fonctionnalités telles que les emails, les SMS, le calendrier, les photos, les cartes, le web, etc. Ces applications sont développées à l'aide du langage de programmation Java et kotlin. Il convient également de noter que ces applications sont développées par Google et donc pour Google. Le calendrier va, par

exemple se synchroniser avec Google Calendar alors que les cartes seront issues de Google Maps. Malgré cette position quelque peu dominante, Google précise tout de même que ces applications peuvent tout à fait être supprimées par l'utilisateur (qui maîtrise un minimum l'informatique) qui le souhaite (même l'application « Home » équivalent de l'explorer sous Windows peut être modifié).

- ✓ **Le Framework** : Android laisse les développeurs créer leurs propres applications. Pour ce faire, les programmeurs peuvent accéder aux mêmes APIs que celles utilisées par les applications natives d'Android.
- ✓ **Les bibliothèques** : En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android. Ces bibliothèques sont en réalité accessibles au développeur par l'intermédiaire du Framework Android. En effet, le Framework Android effectue, de façon interne, des appels à des fonctions C/C++ beaucoup plus rapides à exécuter que des méthodes Java standard. La technologie Java Native Interface (JNI) permet d'effectuer des échanges entre le code Java et le code C et C++.
- ✓ **Android Runtime** : Chaque application Android tourne dans son propre processus et avec sa propre instance de machine virtuelle Dalvik (Dalvik VM). Dalvik VM a été conçue pour optimiser l'exécution multiple de machines virtuelles. La Dalvik VM repose sur le noyau Linux qui s'occupe des fonctions telles que les threads ou la gestion bas-niveau de la mémoire.
- ✓ **Linux Kernel** : Android est basé sur le noyau Linux (2.6). Il profite donc de ces caractéristique de stabilité et efficacité en terme de gestion mémoire, gestion de processus, modèle de sécurité, soutien de bibliothèque partagé, etc. en plus de ça le noyau d'Android est adapté au mobile par des module spécifique au contexte comme la gestion d'énergie.

4.4. Pourquoi nous avons choisi Android ?

Nous avons choisie l'Android à cause des points suivants :

- ✚ **Le système** : Android est une plateforme en code source ouverte et basée sur linux. elle puissante, moderne et sûre. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de remplacer les composants existants. Les utilisateurs peuvent adapter les applications à leur besoins.

- ✚ **La gestion des données :** Avec un téléphone Android on copie les fichiers où l'on veut sur la carte SD (ou la mémoire interne) et tout est détecté par le système. Chez Apple la logique est différente. Le seul moyen de mettre des données sur le mobile est de passer par iTunes.
- ✚ **Optimisation pour les services Google :** Android a été développé depuis le début avec l'intention d'intégration avec Google (ce qui n'est juste mais bon). Les services Google fonctionnent parfaitement sur Android et grâce à ça Android bénéficiera bien du succès de Google. Les développeurs aussi peuvent intégrer plus facilement des services Google dans leurs applications.
- ✚ **Le coût :** Développer une application pour Android ne vous coûtera absolument rien. Tous les outils nécessaires sont fournis par Google et gratuitement ainsi que la documentation.
- ✚ **La position du système et le taux de progression :** Android domine le marché mondial des Smartphones et leurs utilisateurs augmentent rapidement.

Conclusion :

Dans ce chapitre nous avons présenté tous les technologies de l'informatique, le web, les applications web et les applications mobile .dans le prochain chapitre nous allons faire la phase d'analyse de notre système.

CHAPITRE 02

Analyse du système



Introduction :

Dans ce chapitre nous présentons la partie d'analyse de notre system d'une façon détaillée.

I. Etude préliminaire

L'étude préliminaire ou (pré étude) est la première étape de notre processus dedéveloppement. Elle consiste à effectuer un premier repérage des besoins fonctionnels etopérationnels. Dans un premier temps, nous allons introduire l'étude de cas qui servira de fil conducteur tout au long du développement de notre application en donnant une version textuellepréliminaire du cahier des charges. Dans un second temps, nous commencerons par ladétermination des besoins fonctionnels en considérant le système comme une boîte noire. Après avoir identifié les acteurs qui interagissent avec le système, nous développerons un premier modèle UML de niveau contexte, pour pouvoir établir précisément les fonctionnalitésdu système.

1. L'élaboration du cahier de charges :

Le cahier de charges se définit comme un acte, un document de référence qui permet à un dirigeant d'entreprise de préciser les conditions, les règles et les exigences d'un travail à accomplir, en vue de résoudre un problème spécifique ou d'améliorer une situation donnée tout en déterminant les résultats attendus.

1.1. Présentation du projet :

Plusieurs entreprise actuellement tel que Sonal gaz, Algérie post, Algérie télécoms possèdent des équipes de travail sur terrain qui interviennent sur des emplacements différents, habituellement ces équipes sont gérés par téléphone, cette méthode de gestion est lourde et inefficace. L'objectif de ce projet est de proposé une solution générique pour gérer en temps réel les équipes de travail des entreprises, ce qui permettra d'améliorer leurs performances.

Cette solution est décomposée en deux parties, la première c'est une application web pour l'administrateur de l'entreprise qui lui permet de gérer les équipes. Et la deuxième est une application Android qui sera utilisé par le chef des équipes.

1.2. Recueil des besoins fonctionnels :

Nous avons effectué une étude détaillée du sujet pour identifier les besoins de l'application, et ceci afin de répondre à la fois aux besoins de l'administrateur et des utilisateurs d'application mobile (chef d'équipe).

1.3. Les besoins fonctionnels de l'administrateur :

Il fait les actions suivantes :

- Gérer les équipes de travail.
- Communiquer avec les équipes.
- Gestion des tâches.
- Gestion des employés.
- Suivre l'avancement du travail (temps réel).

1.4. Les besoins fonctionnels des utilisateurs (au niveau des Smartphones) :

Il fait les actions suivantes :

- Contacter l'administrateur.
- Consulter les tâches.
- Evaluer l'avancement des tâches.

1.5. Recueil des besoins non fonctionnels:

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

- ❖ La disponibilité : l'application doit être disponible pour être utilisée par n'importe quel utilisateur.
- ❖ La sécurité de l'accès aux informations critiques : nous devons prendre en considération la confidentialité des données de clients surtout au niveau de l'authentification.
- ❖ La fiabilité : les données fournies par l'application doivent être fiables.
- ❖ La convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités.

- ❖ Une solution ouverte et évoluée : l'application peut être améliorée par l'ajout d'autres modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.

2. Description du contexte :

2.1. Identifications des acteurs

Un acteur est un utilisateur type du système. Il représente une responsabilité par rapport au système ou un rôle plutôt qu'une personne physique.

Dans la suite nous présenterons les différents acteurs qui interagissent avec notre système.

2.1.1. Au niveau de l'administrateur :

A ce niveau nous avons une application web avec un acteur administrateur. Cet acteur accède de manière sécurisée au système. Il peut :

- Gérer les équipes :
 - ✓ Ajouter équipe.
 - ✓ Modifier équipe.
 - ✓ Supprimer équipe.
 - ✓ Afficher les informations de l'équipe.
 - ✓ Evaluer le rendement de l'équipe
- Gérer les tâches :
 - ✓ Ajouter tâche.
 - ✓ Modifier tâche.
 - ✓ Supprimer tâche.
 - ✓ Affecter une tâche à une équipe.
 - ✓ Afficher la liste des tâches.
 - ✓ Annuler l'affectation d'une tâche.
- Gérer les employés :
 - ✓ Ajouter employé.
 - ✓ Modifier employé.
 - ✓ Supprimer employé.
 - ✓ Afficher employé.
- Communiquer équipe :
 - ✓ Envoyer message.
 - ✓ Consulter la liste des messages.

- ✓ Supprimer message.
- Suivre l'avancement du travail en temps réel.

2.1.2. Au niveau du Smartphones :

A ce niveau nous avons une application avec un acteur utilisateur (chef d'équipe). Cet acteur accède de manière sécurisée au système. Il peut :

- Contacter l'administrateur.
- ✓ Envoyer message.
- ✓ Consulter la boîte de réception.
- ✓ Consulter les notifications.
- Consulter les tâches.
- Evaluer l'avancement des tâches (pourcentage de travail).

2.2. Identifications des messages :

Le message est le seul moyen de communication entre les objets pour déclencher une activité chez le récepteur. Cette notion est utilisée également pour décrire les interfaces de plus haut niveau entre les acteurs et le système. Notre système est une collection de deux systèmes : le premier au niveau de l'administrateur et le deuxième au niveau des Smartphones. Le système interagit avec les différents acteurs à l'aide des messages échangés entre eux.

2.2.1. Les messages émis par le système :

- Information des équipes.
- Information des tâches.
- Information de l'avancement du travail.

2.2.2. Les messages reçus :

- Demande de consulter la liste des équipes.
- Demande de consulter la liste des tâches affectées aux équipes.
- Information de travail demandé à chaque équipe.
- Information d'avancement du travail.
- Boîte de réception.

2.3. Le diagramme de contexte du système :

La modélisation du contexte consiste à représenter de façon synthétique tous les messages échangés entre système et acteurs sur un diagramme, que l'on peut qualifier de diagramme de contexte dynamique. Ce diagramme dynamique est présenté grâce à un diagramme de collaboration UML comme suit :

- Le système étudié est représenté par un objet central.
- Cet objet central entouré par un autre objet symbolisant les différents acteurs.
- Des liens relient le système à chaque acteur.
- Sur chaque lien sont montrés les messages en entrée et en sortie du système.

Légende des messages du diagramme de contexte dynamique	
<p>1 : Administrateur —————→ système</p> <ul style="list-style-type: none"> • Information des équipes ; • Information des tâches ; • Information des employés ; • Information d'avancement du travail ; • Messages ; 	
<p>2 : système —————→ Administrateur</p> <ul style="list-style-type: none"> • Information des équipes ; • Information des tâches ; • Information des employés ; • Information d'avancement du travail ; • Messages reçus ; 	
<p>3 : système —————→ utilisateur</p>	

<ul style="list-style-type: none">• Information des taches ;• Messages reçus ;
<p>4 : utilisateur \longrightarrow système</p> <ul style="list-style-type: none">• Demande de consulter les taches ;• Envoyer Messages ;

Tableau1 : Légende des messages du diagramme de contexte dynamique

3. Le diagramme de cas d'utilisation pour les besoins fonctionnels :

Après identification des cas d'utilisation et leurs acteurs, nous allons les représenter graphiquement sur un diagramme de cas d'utilisation qui représente le fonctionnement du système vis-à-vis de l'utilisateur :

3.1 Au niveau de l'administrateur :

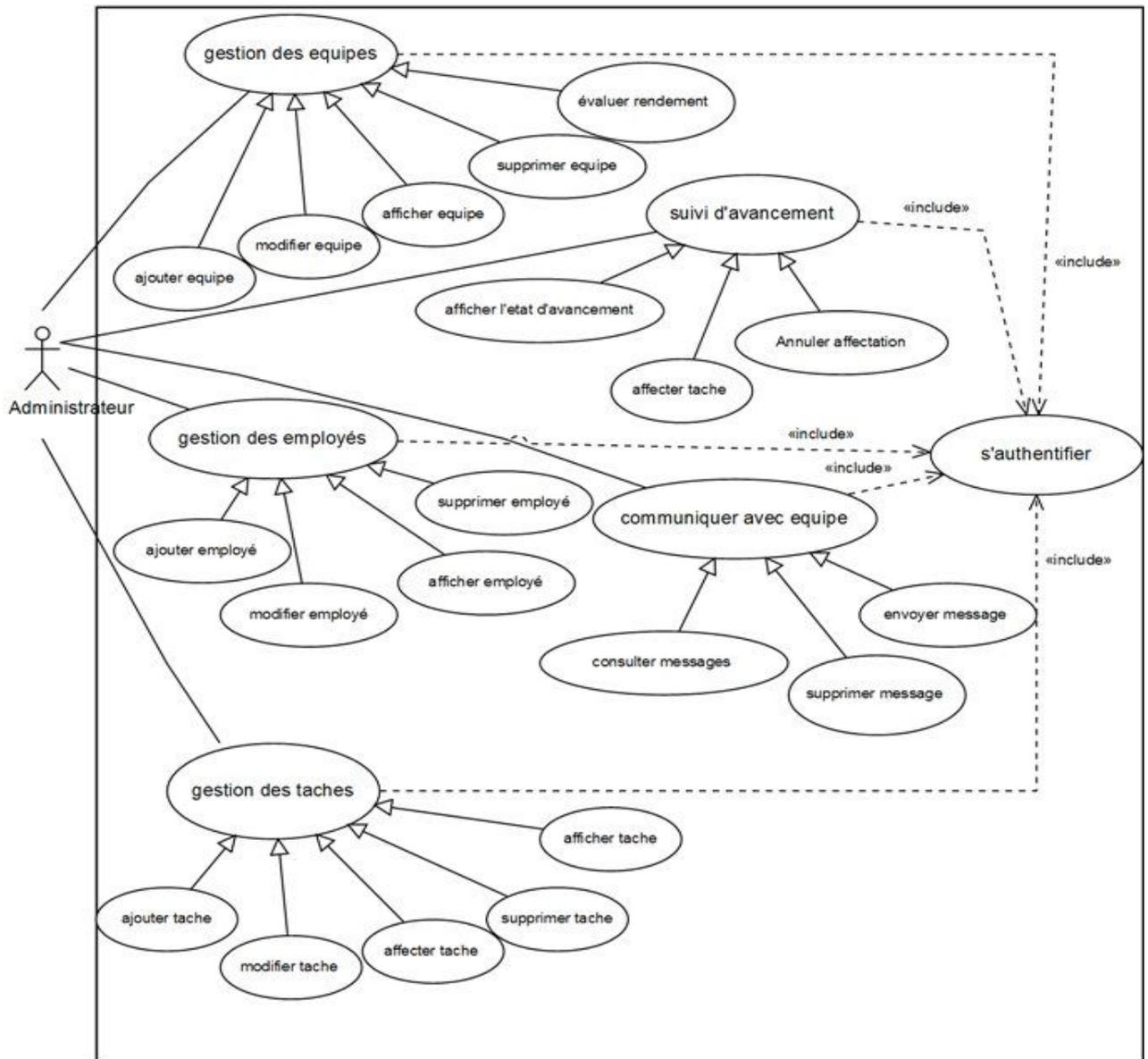


Figure 1: Diagramme du cas d'utilisation Au niveau de l'administrateur.

3.2. Au niveau de l'utilisateur Smartphone :

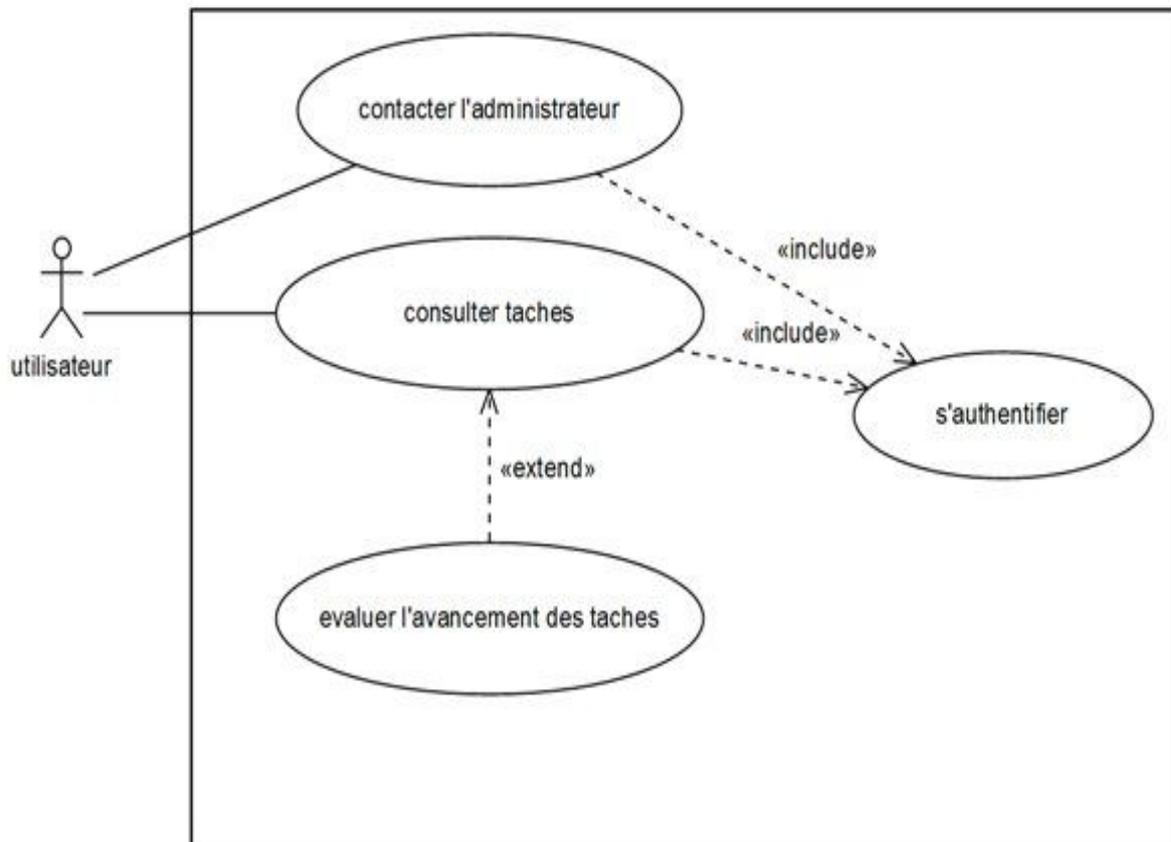


Figure 2: Diagramme du cas d'utilisation au niveau de l'utilisateur Smartphone (chefd'équipe).

4. Description détaillée des différents cas d'utilisation

4.1. Au niveau de l'administrateur :

❖ Cas d'utilisation « S'authentifier » :

Acteur	L'administrateur.
Objectif	Vérifier l'autorisation d'accès à l'espace de travail.
Pré condition	Compte existant.
Post Condition	L'administrateur s'est authentifié.

<p>Scenario nominal</p>	<ol style="list-style-type: none"> 1. le système affiche le formulaire de saisie du nom d'utilisateur et du mot de passe. 2. l'administrateur saisit le nom d'utilisateur et le mot de passe. 3. le système vérifie la validité du nom d'utilisateur et du mot de passe. 4. le système affiche l'espace de travail.
<p>Scenario alternatif</p>	<p>-le système détecte que le nom d'utilisateur ou le mot de passe incorrecte :</p> <ol style="list-style-type: none"> 1. Le système propose à l'utilisateur une nouvelle fois de saisir le nom d'utilisateur et le mot de passe, reprendre le scenario nominal au point 2.

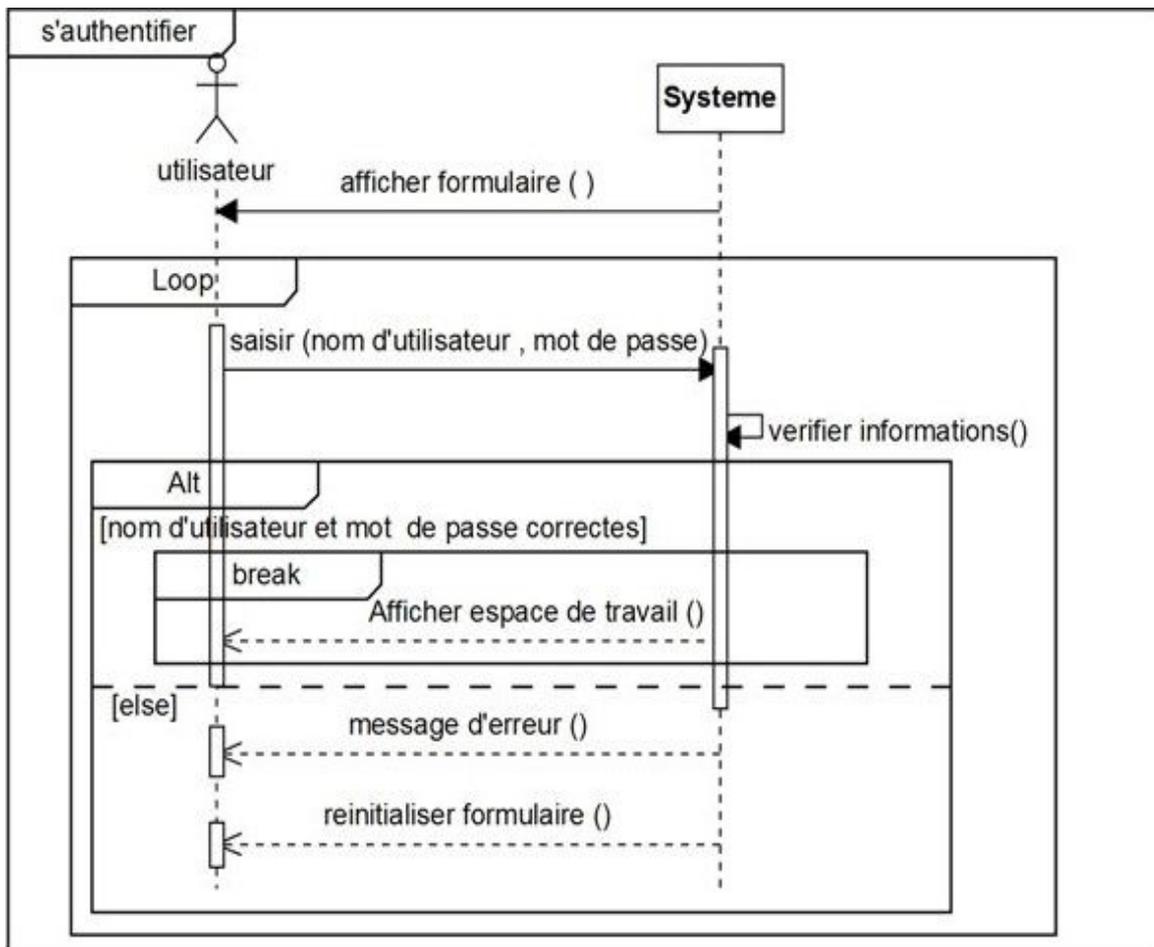


Figure 3: Diagramme de séquence du cas d'utilisation « S'authentifier ».

❖ Cas d'utilisation « Gestion des équipes ».

Acteur	L'administrateur
Objectif	Ajouter/modifier/supprimer/afficher : équipe
Pré condition	-L'administrateur doit être authentifié. -L'administrateur possède des informations concernant les équipes.
Post condition	Les informations ont été enregistrées.
Scenario nominal	Commence lorsque l'administrateur demande au système la gestion des équipes. 1. l'administrateur accède au système. 2. le système présente l'état actuel. 3. -l'administrateur choisit l'opération ajouter équipe sous cas : ajouter équipe. -l'administrateur choisit l'opération modifier équipe sous cas : modifier équipe. -l'administrateur choisit l'opération supprimer équipe sous cas: supprimer équipe. -l'administrateur choisit l'opération afficher équipe sous cas : afficher équipe. 4. le système enregistre les modifications.
Scenario alternatif	-lorsque l'administrateur annule l'accès, il retourne à l'étape 2. -lorsque l'administrateur saisi des informations incomplètes ou erronées .le système affiche un message d'erreur et retourne à l'étape 2.

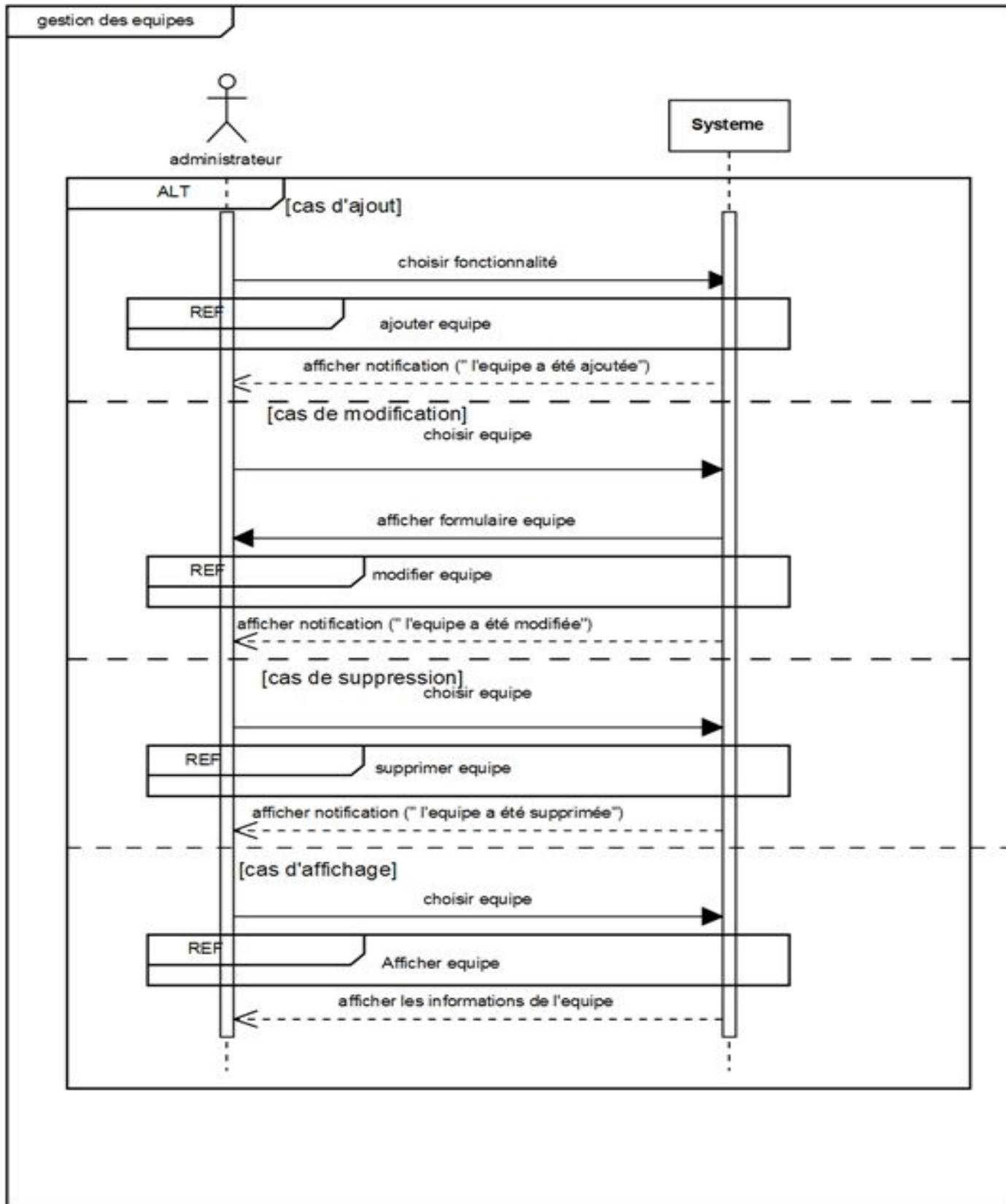


Figure 4 : Diagramme de séquence du cas d'utilisation « Gestion des équipes ».

❖ Cas d'utilisation « Ajouter équipe »

Acteur	L'administrateur
Objectif	Ajouter une équipe.
Pré condition	L'administrateur doit être authentifié.
Post condition	L'équipe est ajoutée.
Scenario nominal	<ol style="list-style-type: none">1. le système affiche le formulaire d'ajout2. l'administrateur remplit les informations3. le système vérifie les informations.4. le système affiche une notification « l'équipe a été ajouté ».
Scenario alternatif	Lorsque l'administrateur saisi des informations incomplètes ou erronées : -le système affiche un message d'erreur.

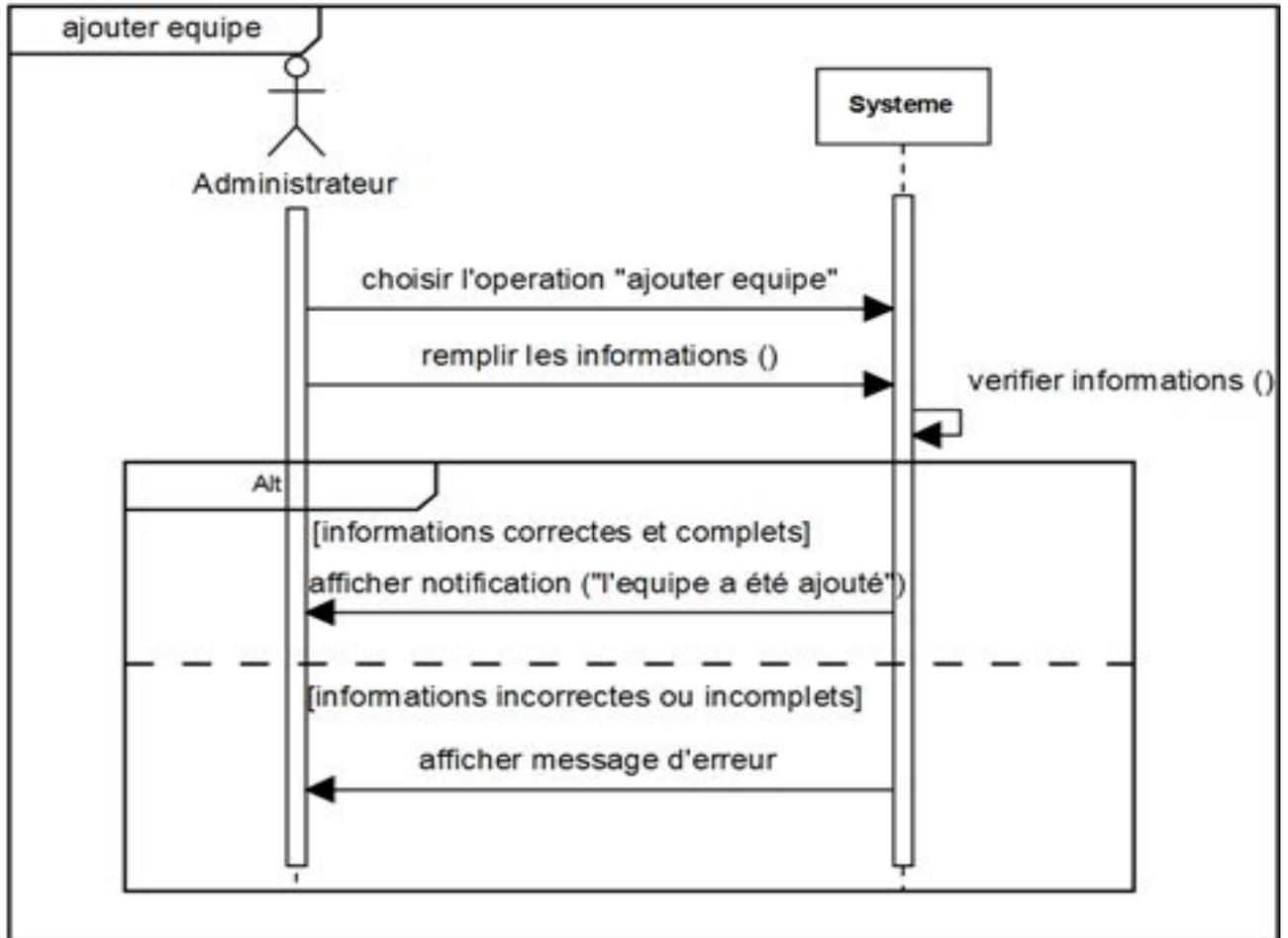


Figure 5 : Diagramme de séquence du cas d'utilisation « Ajouter équipe ».

❖ Cas d'utilisation « Modifier équipe » :

Acteur	L'administrateur
Objectif	Modifier une équipe.
Pré condition	L'administrateur doit être authentifié. L'équipe existe.
Post condition	L'équipe est modifiée.

<p>Scenarionominal</p>	<ol style="list-style-type: none"> 1. le système affiche la liste des équipes 2. l'administrateur choisi une équipe. 3. le système affiche formulaire d'équipe. 4. l'administrateur remplit les informations. 5. le système vérifie les informations. 6. le système affiche une notification « l'équipe a été modifié ».
<p>Scenarionalternatif</p>	<p>Lorsque l'administrateur saisi des informations incomplètes ou erronées :</p> <ul style="list-style-type: none"> -le système affiche un message d'erreur. -le système reprend le scenario à l'étape 3.

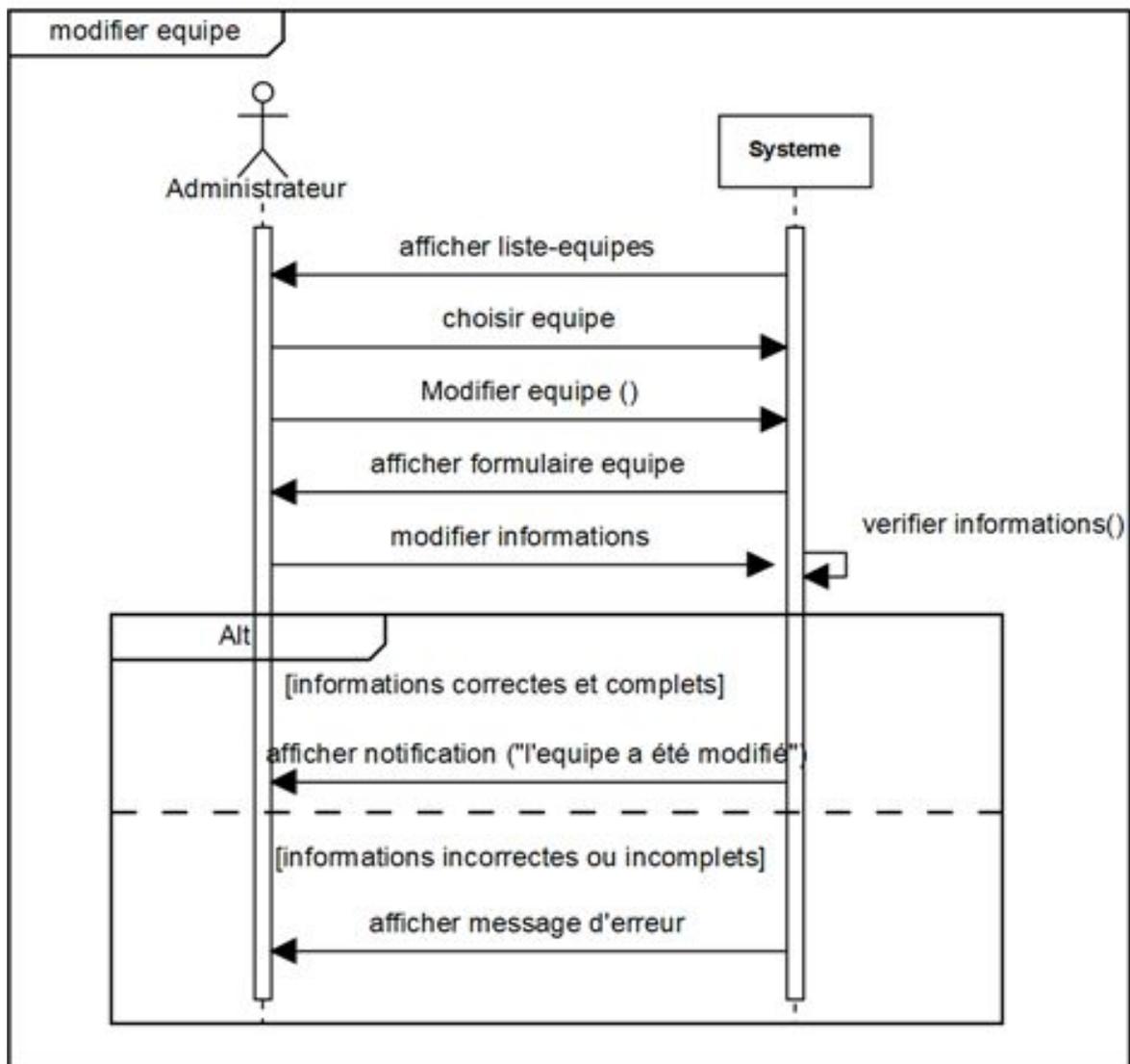


Figure 6: Diagramme de séquence du cas d'utilisation « Modifier équipe ».

❖ Cas d'utilisation « Supprimer équipe »

Acteur	L'administrateur
Objectif	supprimer une équipe.
Pre condition	L'administrateur doit être authentifié. L'équipe existe.
Post condition	L'équipe est supprimée.
Scenario nominal	<ol style="list-style-type: none"> 1. le système affiche la liste des équipes 2. l'administrateur choisi une équipe. 3. l'administrateur supprime l'équipe. 4. le système affiche une notification « l'équipe a été supprimée ».

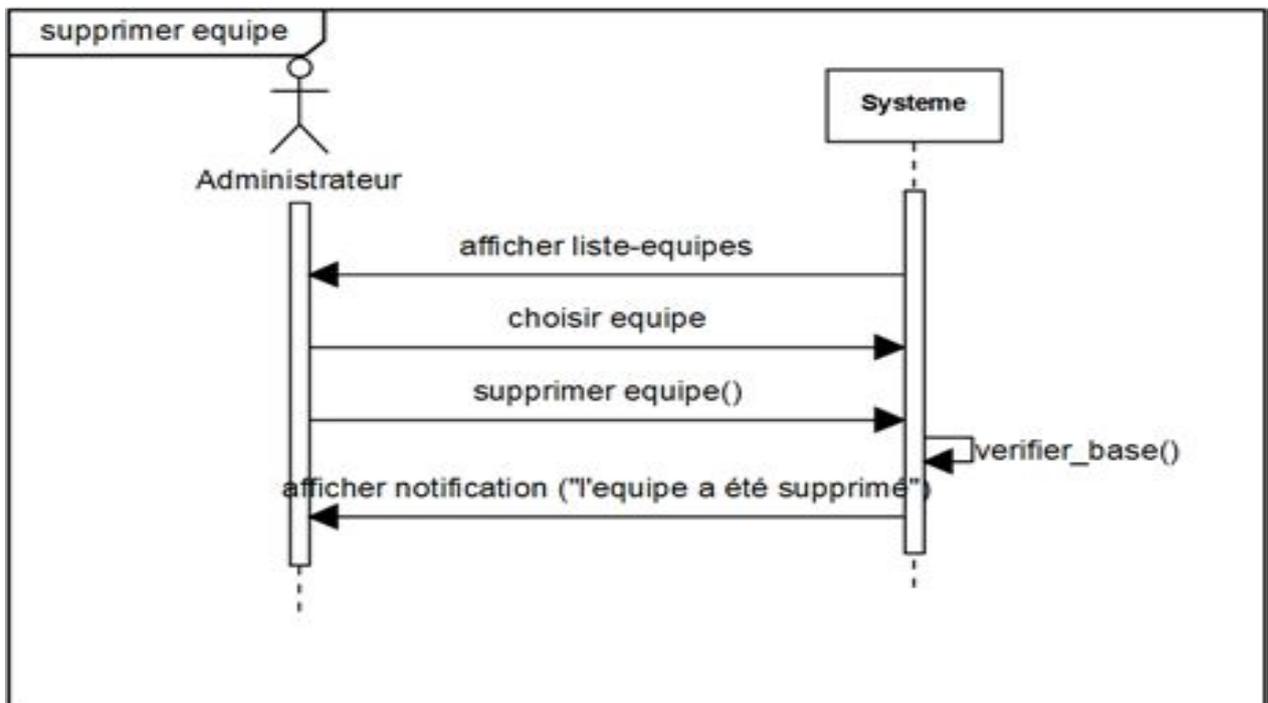


Figure 7: Diagramme de séquence du cas d'utilisation « Supprimer équipe ».

❖ Cas d'utilisation «Afficher équipe » :

Acteur	L'administrateur
Objectif	Afficher tous les informations d'une équipe.
Pre condition	L'administrateur doit être authentifié. L'équipe existe.
Post condition	toutes les informations de l'équipe sont affichées.
Scenario nominal	1. le système affiche la liste des équipes 2. l'administrateur choisi une équipe. 3. le système affiche les informations de l'équipe.

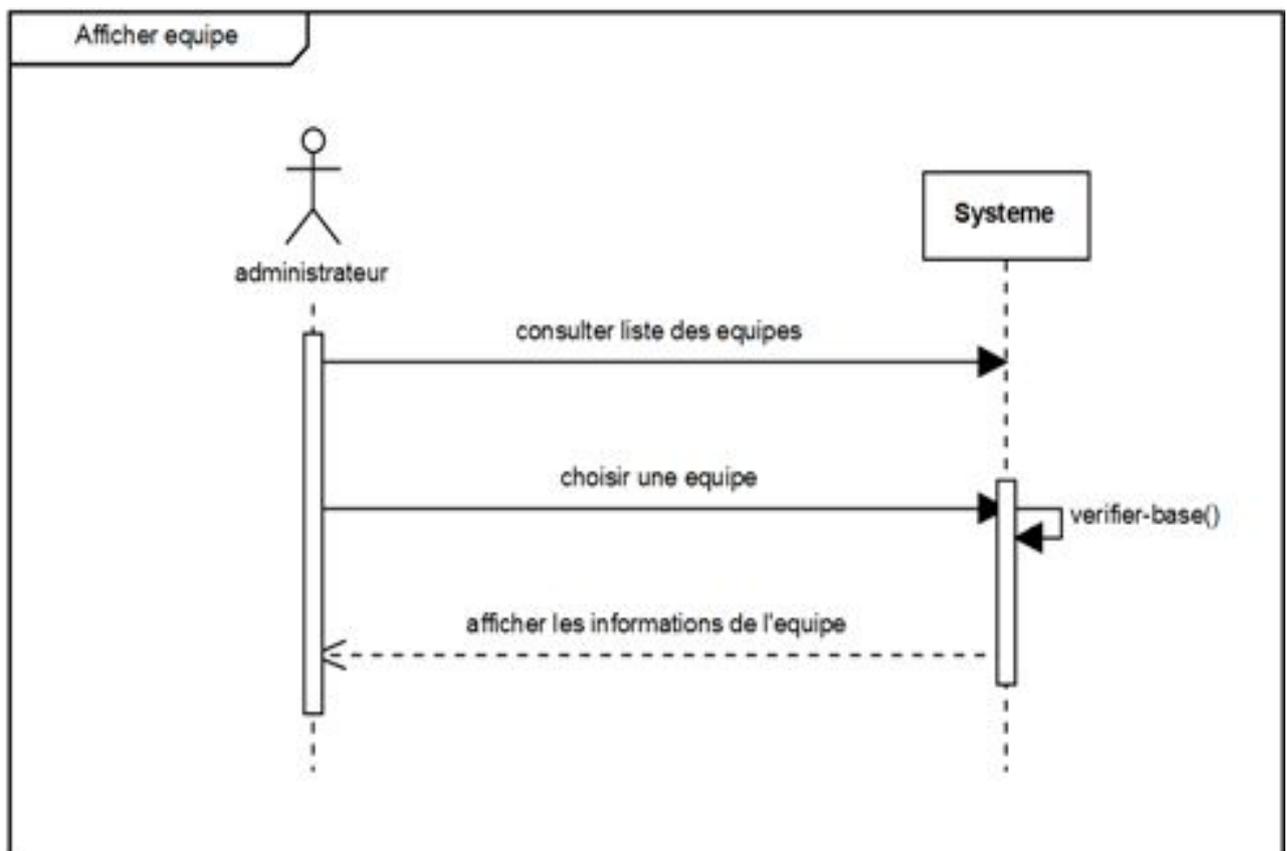


Figure 8: Diagramme de séquence du cas d'utilisation « Afficher équipe ».

❖ Cas d'utilisation « Gestion des tâches » :

Acteur	L'administrateur
Objectif	Ajouter/modifier/supprimer/afficher : tache
Pré condition	L'administrateur doit être authentifié. L'administrateur possède des informations concernant les taches.
Post condition	Les informations ont été enregistrées.
Scenario nominal	Commence lorsque l'administrateur demande au système la gestion des taches. 1. l'administrateur accède au système. 2. le système présente l'état actuel. 3. -l'administrateur choisit l'opération ajouter tache sous cas : ajouter tache. -l'administrateur choisit l'opération modifier tache sous cas : modifier tache. -l'administrateur choisit l'opération supprimer équipe sous cas : supprimer tache. -l'administrateur choisit l'opération afficher tache sous cas : afficher tache. 4. le système enregistre les modifications.
Scenario alternatif	-lorsque l'administrateur annule l'accès, il retourne à l'étape 2. -lorsque l'administrateur saisi des informations incomplètes ou erronées. le système affiche un message d'erreur et reprend le scénario à l'étape 2.

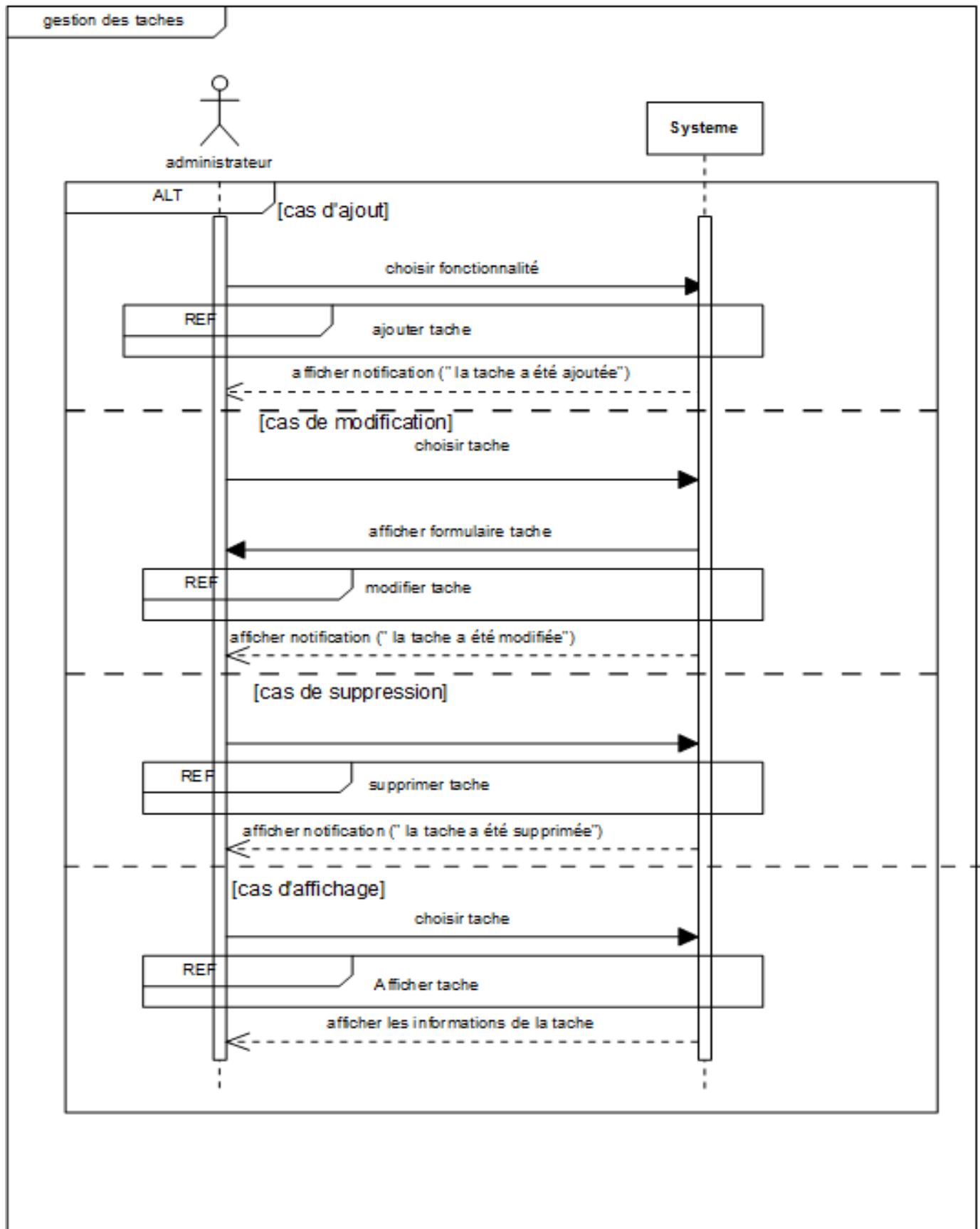


Figure 9: Diagramme de séquence du cas d'utilisation « Gestion des tâches ».

❖ Cas d'utilisation « Ajouter tâche » :

Acteur	L'administrateur
Objectif	Ajouter une tâche.
Pré condition	L'administrateur doit être authentifié.
Post condition	La tâche est ajoutée.
Scenario nominal	<ol style="list-style-type: none">1. le système affiche le formulaire d'ajout2. l'administrateur remplit les informations3. le système vérifie les informations.4. le système affiche une notification « la tâche a été ajouté ».
Scenario alternatif	Lorsque l'administrateur saisi des informations incomplètes ou erronées : -le système affiche un message d'erreur.

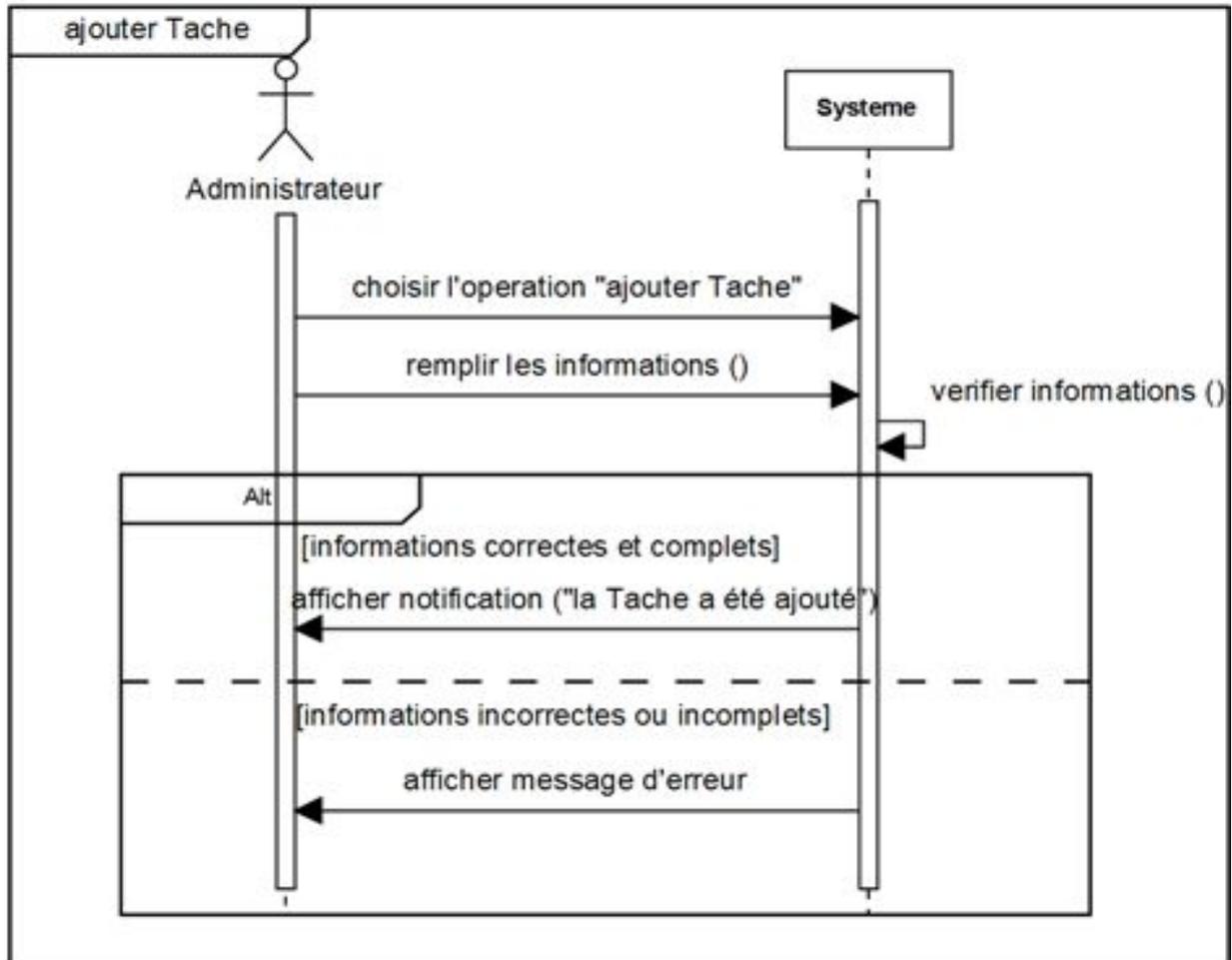


Figure 10: Diagramme de séquence du cas d'utilisation « Ajouter tâche ».

❖ Cas d'utilisation « Modifier tâche » :

Acteur	L'administrateur
Objectif	Modifier une tâche.
pré condition	L'administrateur doit être authentifié. La tâche existe.
Post condition	La tâche est modifiée.
Scenario nominal	1. le système affiche la liste des tâches 2. l'administrateur choisi une tâche. 3. le système affiche formulaire de tâche. 4. l'administrateur remplit les informations.

	<p>5. le système vérifie les informations.</p> <p>6. le système affiche une notification « la tâche a été modifié ».</p>
Scenario alternatif	<p>Lorsque l'administrateur saisi des informations incomplètes ou erronées :</p> <ul style="list-style-type: none"> -le système affiche un message d'erreur. -le système reprend le scenario à l'étape 3.

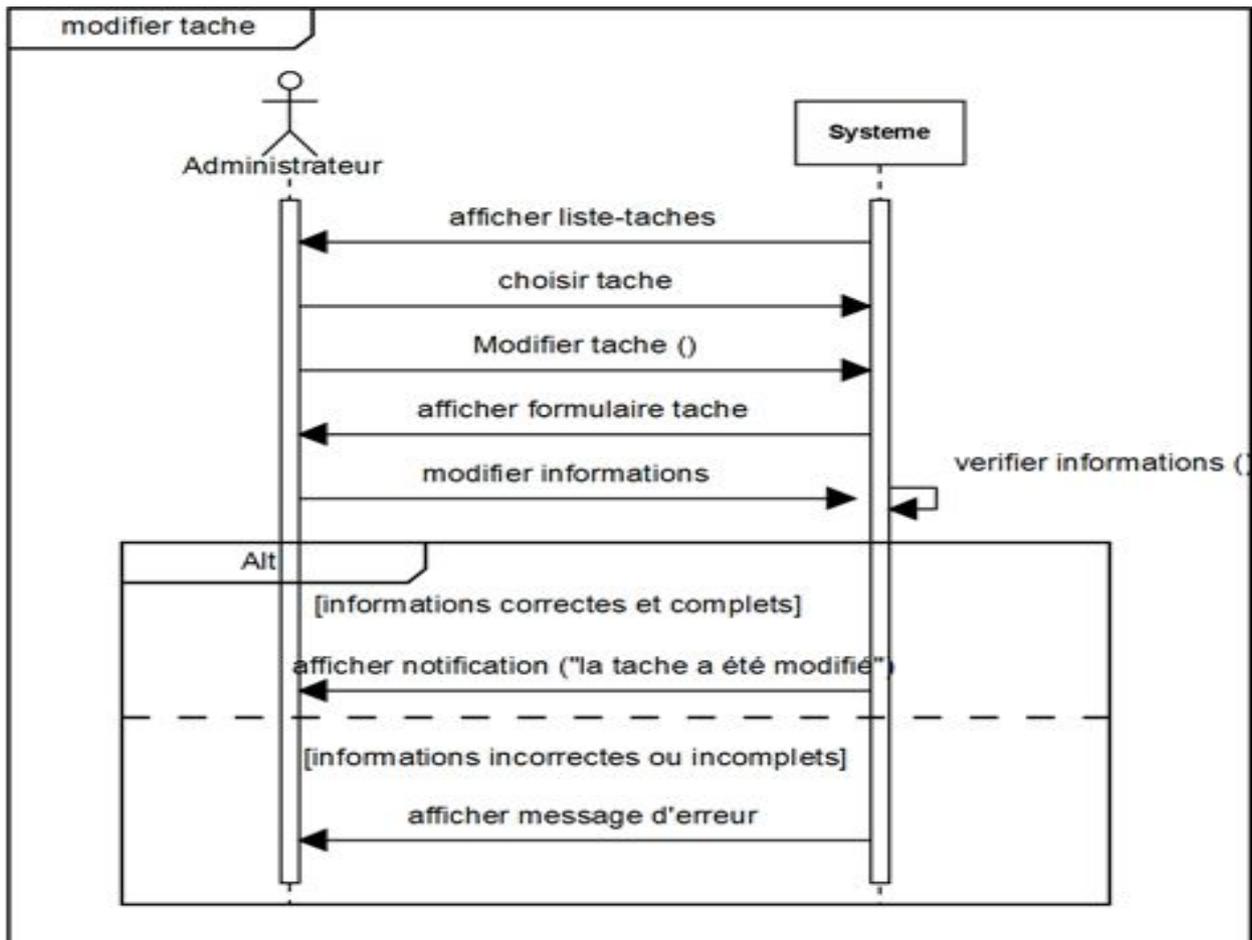


Figure 11: Diagramme de séquence du cas d'utilisation « Modifier tache ».

❖ Cas d'utilisation « Supprimer tâche » :

Acteur	L'administrateur
Objectif	supprimer une tâche.
Pré condition	L'administrateur doit être authentifié. La tâche existe.
Post condition	La tâche est supprimée.
Scenario nominal	<ol style="list-style-type: none"> 1. le système affiche la liste des tâches 2. l'administrateur choisi une tâche. 3. l'administrateur supprime la tâche. 4. le système affiche une notification « la tâche a été supprimée ».

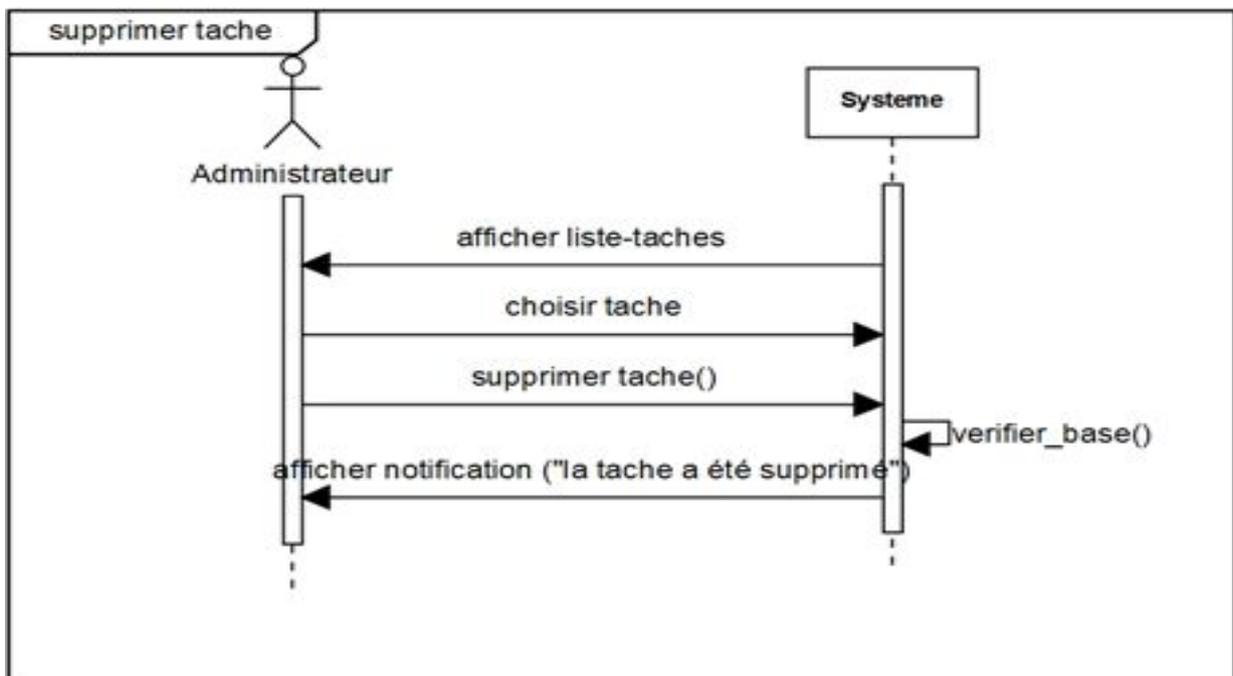


Figure 12: Diagramme de séquence du cas d'utilisation «supprimer tâche».

❖ Cas d'utilisation «Afficher tâche» :

acteur	L'administrateur
Objectif	Afficher tous les informations d'une tâche.
Pré condition	L'administrateur doit être authentifié. La tâche existe.
Post condition	Toutes les informations de la tâche sont affichées.
Scenario nominal	1. le système affiche la liste des tâches 2. l'administrateur choisi une tâche. 3. le système affiche les informations de la tâche.

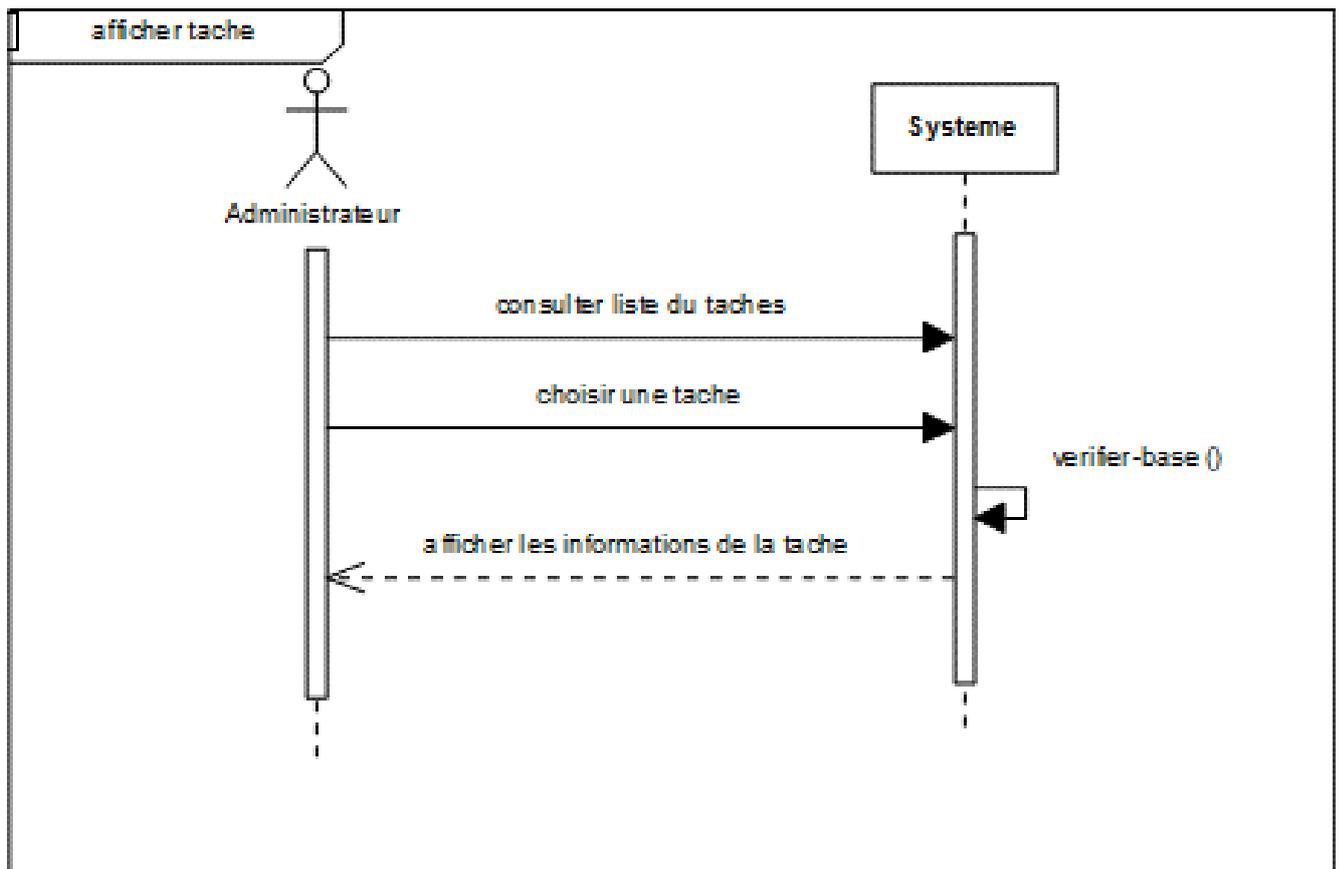


Figure 13: Diagramme de séquence du cas d'utilisation «afficher tâche».

❖ Cas d'utilisation « communication » :

Acteur	Administrateur
Objectif	-envoyer message a une équipe. -consulter boite de réception -lire les messages reçus.
Pre condition	L'administrateur doit être authentifié
Post condition	le message doit être envoyé les messages reçus doivent être affiché
Scenario nominal	1. l'administrateur accède au system 2. -l'administrateur choisi l'opération envoyer message - l'administrateur choisi l'opération consulter message.

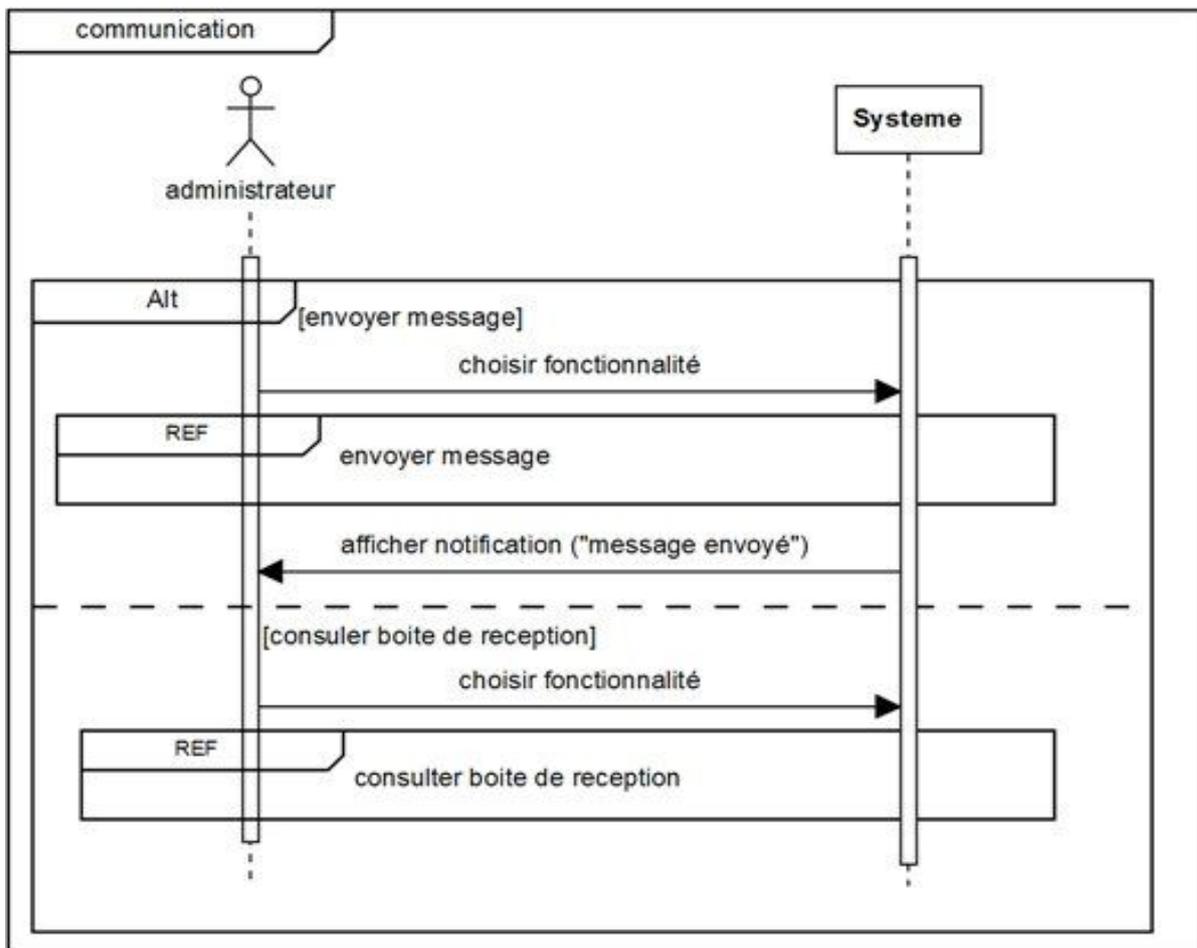


Figure 14: Diagramme de séquence du cas d'utilisation «communication ».

❖ Cas d'utilisation « envoyer message » :

Acteur	Administrateur
Objectif	Envoyer message
pré condition	L'administrateur doit être authentifié. Une équipe doit être sélectionnée
Post condition	Le message est envoyé
Scénario nominal	L'administrateur sélectionne une équipe L'administrateur écrit et envoie le message Le système affiche une notification « le message a été envoyé ».

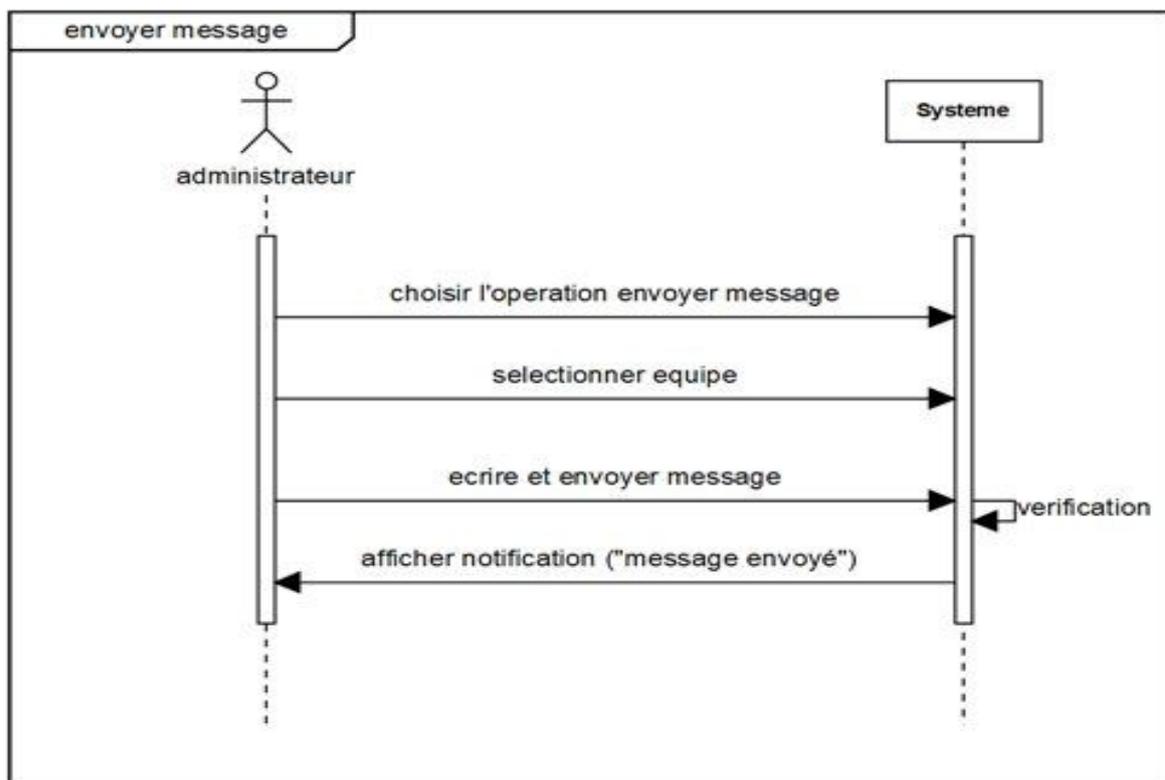


Figure 15: Diagramme de séquence du cas d'utilisation «envoyer message ».

❖ Cas d'utilisation « consulter messages » :

Acteur	Administrateur
Objectif	Afficher les messages reçus.
Pre condition	L'administrateur doit être authentifié. Message existe
Post condition	Le message est affiché.
Scenario nominal	1. Le système affiche la liste des messages 2. L'administrateur choisie un message 3. Le system affiche le message complet.

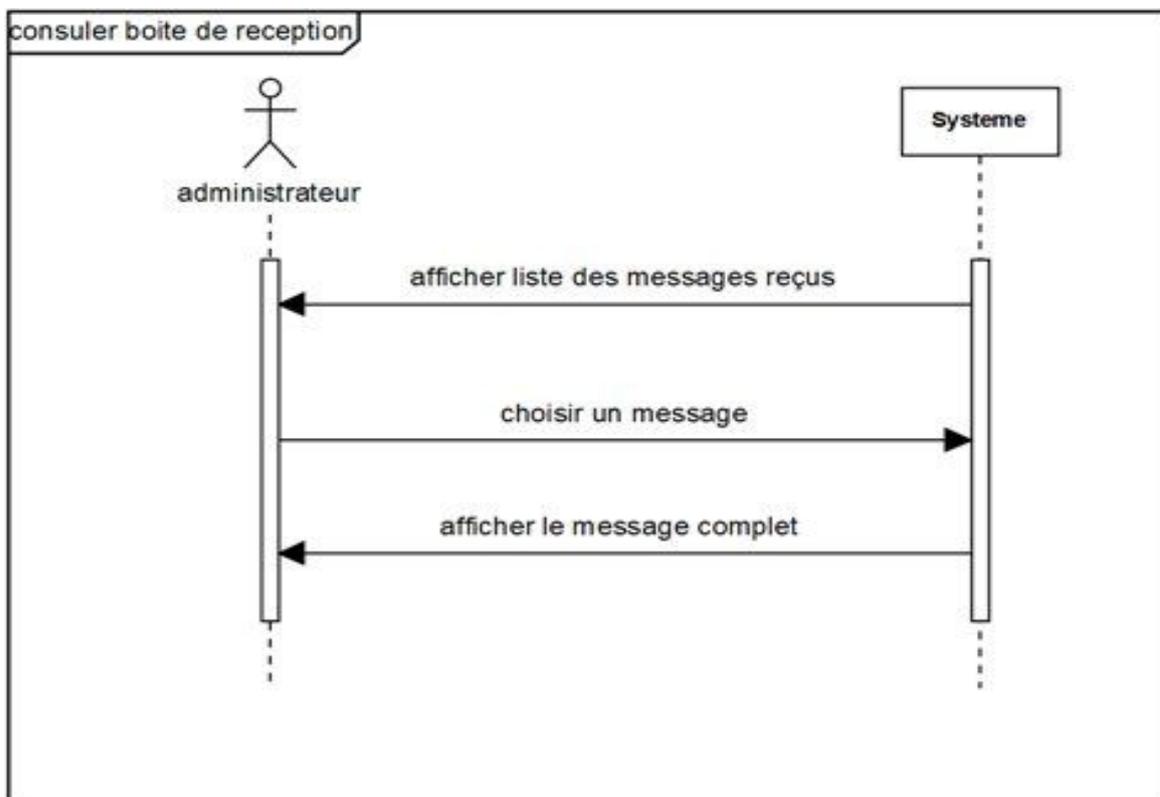


Figure 16: Diagramme de séquence du cas d'utilisation «consulter messages ».

❖ Cas d'utilisation « suivi d'avancement » :

Acteur	Administrateur
Objectif	-afficher l'état d'avancement des tâches. -affecter les tâches. -annuler l'affectation des tâches.
pré condition	L'administrateur doit être authentifié. équipe existe. tâche existe.
Post condition	La tâche est affectée. L'affectation est annulée. L'état d'avancement des taches est affiché.
Scenario nominal	1. l'administrateur accède au system 2. -l'administrateur choisi l'opération afficher l'état d'avancement des tâches. - l'administrateur choisi l'opération affecter tache. - l'administrateur choisi l'opération supprimer affectation.

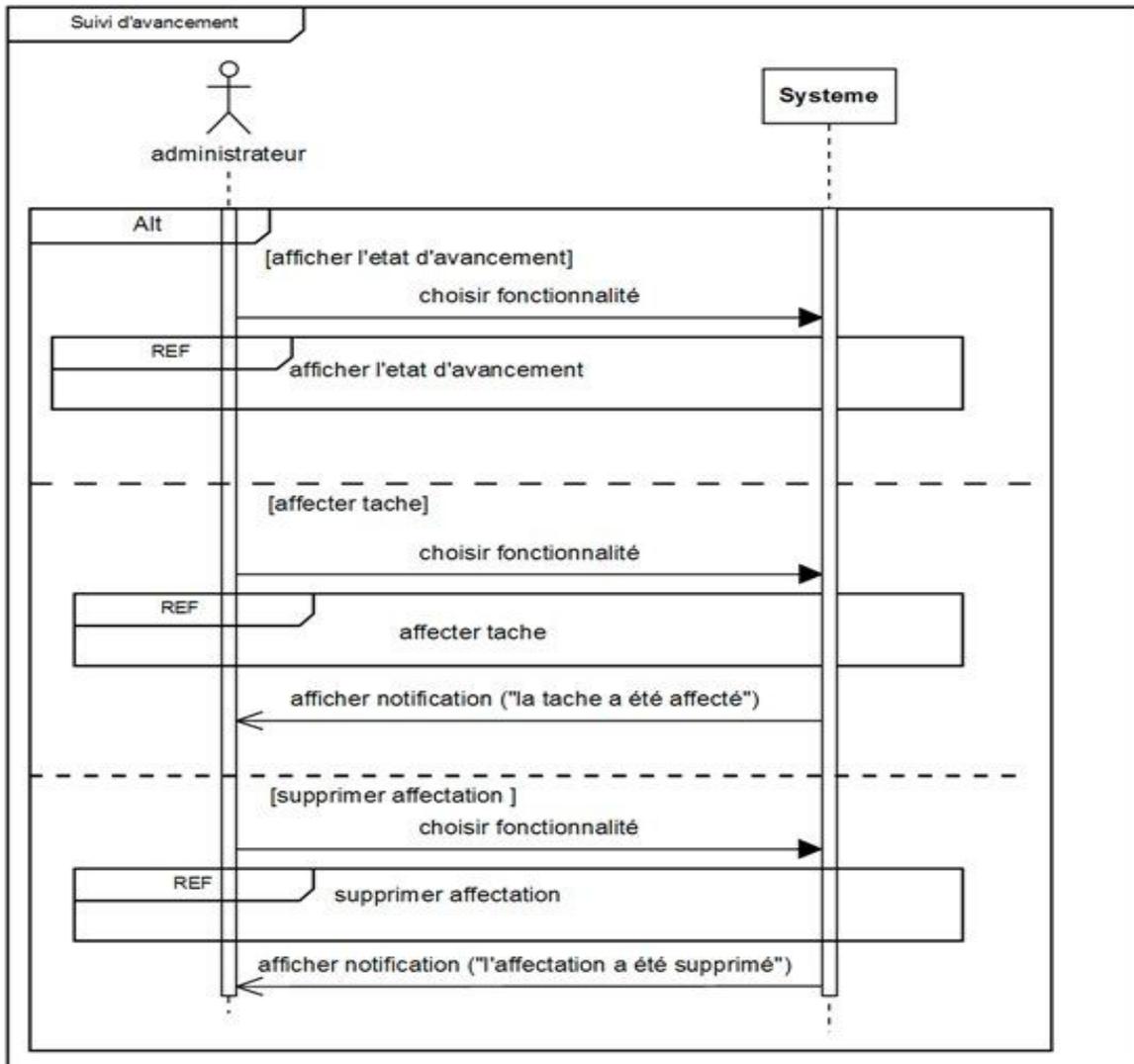


Figure 17: Diagramme de séquence du cas d'utilisation «Suivi d'avancement»

❖ Cas d'utilisation « affecter tâche » :

Acteur	Administrateur
Objectif	Affecter une tache a une équipe.
pré condition	L'administrateur doit être authentifié. tache existe. équipe existe.
Post condition	La tâche est affectée.
Scenario nominal	1. Le système affiche les taches non affectées. 2. L'administrateur choisie une tache 3. l'administrateur sélectionne une équipe.

	<p>4. l'administrateur affecter la tâche.</p> <p>5. le système affiche notification « la tâche a été affecté ».</p>
--	---

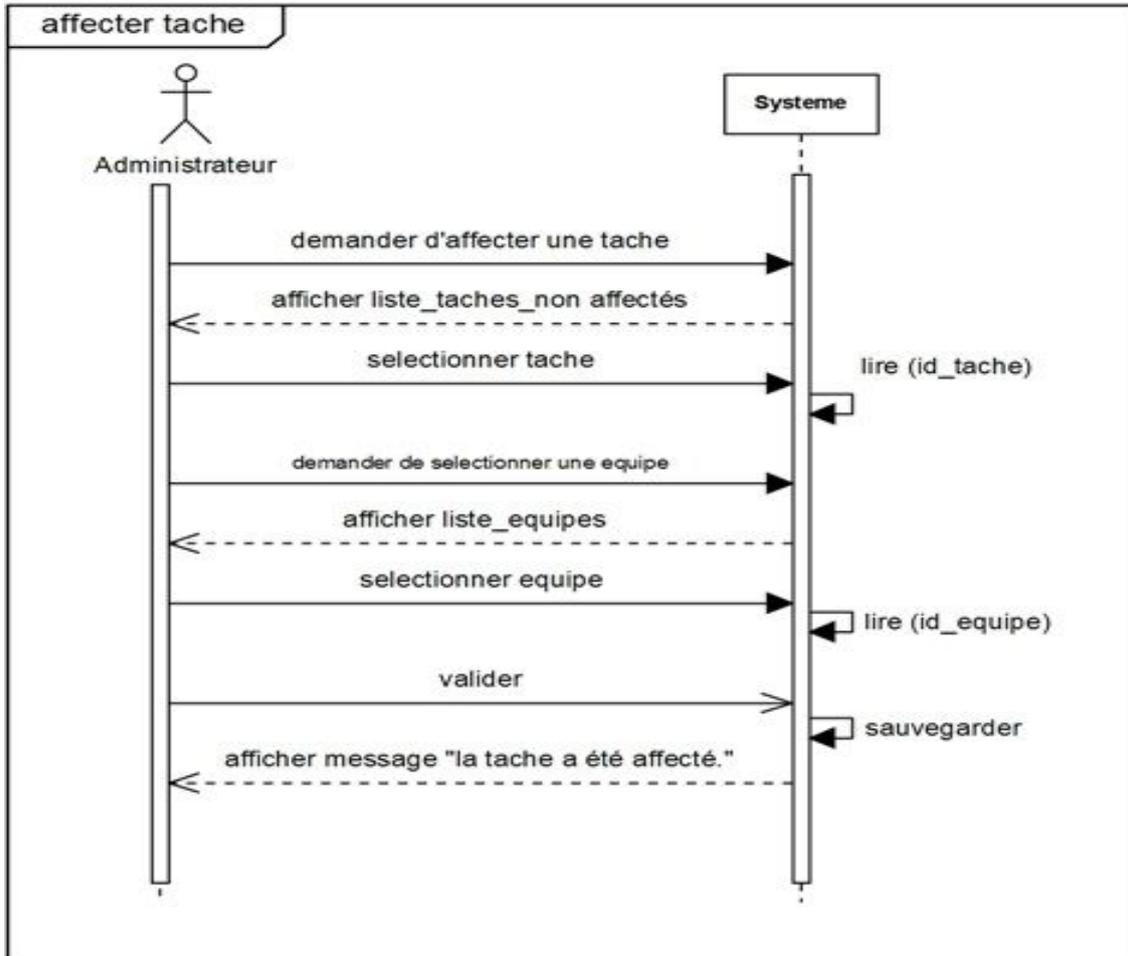


Figure 18: Diagramme de séquence du cas d'utilisation «affecter tache»

❖ Cas d'utilisation «Annuler affectation» :

Acteur	Administrateur
Objectif	Supprime l'affectation d'une tache.
pré condition	L'administrateur doit être authentifié. tâche existe tâche affectée.

Post condition	L'affectation est supprimée.
Scenario nominal	<ol style="list-style-type: none"> 1. Le système affiche les taches affectées. 2. L'administrateur choisie une tache 3. l'administrateur supprime l'affectation de la tâche. 4. le système affiche notification « l'affectation a été annulée ».

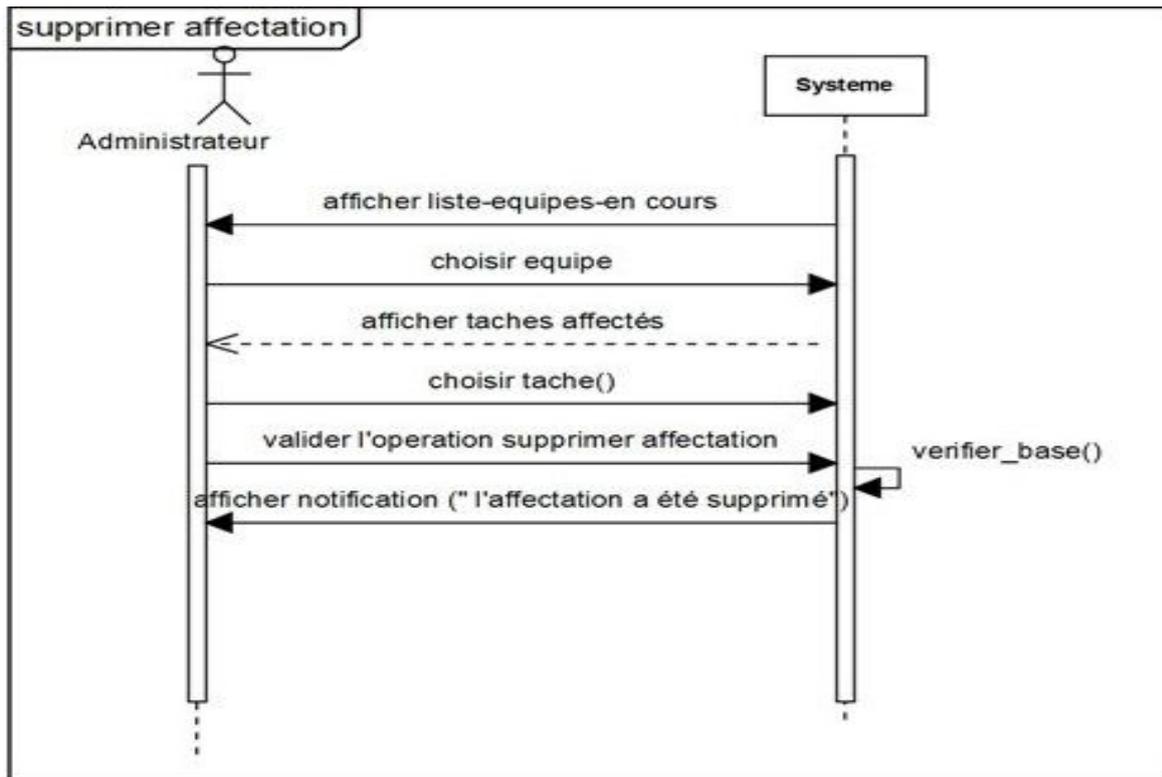


Figure 19: Diagramme de séquence du cas d'utilisation «Annuler affectation»

❖ Cas d'utilisation « afficher l'état d'avancement » :

Acteur	Administrateur
Objectif	Afficher l'état d'avancement des taches.
pré condition	<p>L'administrateur doit être authentifié.</p> <p>équipe existe.</p> <p>tache affectée.</p>

Post condition	L'état d'avancement est affiché.
Scenario nominal	<ol style="list-style-type: none"> 1. Le système affiche les équipes. 2. L'administrateur choisie une équipe 3. le système affiche l'état d'avancement des taches affectées a l'équipechoisi.

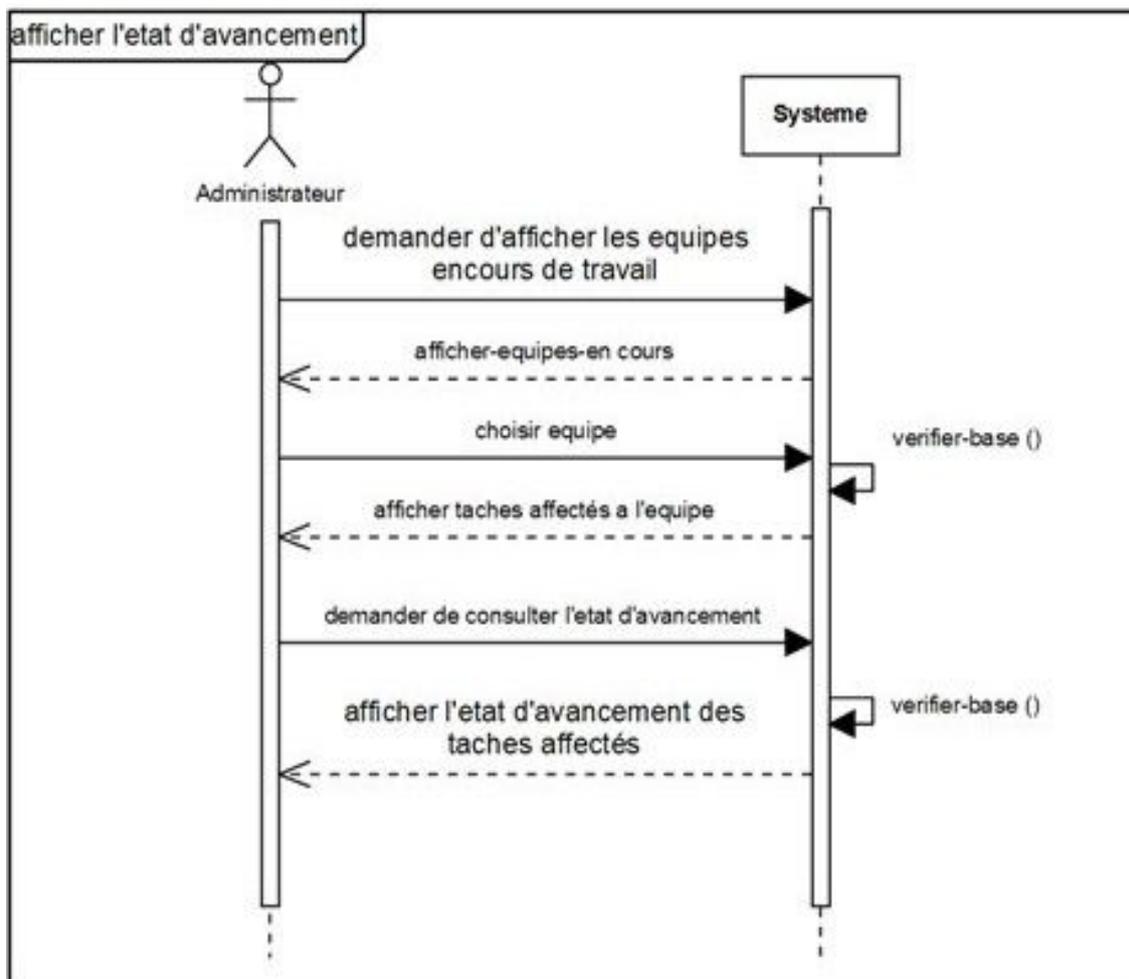


Figure 20: Diagramme de séquence du cas d'utilisation «afficher l'état d'avancement»

Conclusion :

Durant ce chapitre nous avons recueilli toutes les informations d'aspect fonctionnel qui font différent action (gestion des équipes, communication , gestion des taches, suivi d'avancement des taches ...) afin de pouvoir fixer les principales fonctionnalités que doit disposer notre future Système, Dans le prochain chapitre nous allons commencer la partie de conception.

CHAPITRE 03

Conception



Introduction

La conception est la phase ultime de la modélisation avec UML. Après la modélisation des besoins puis l'organisation de la structure de la solution, la conception vient construire et documenter précisément les classes et les méthodes qui constituent le codage de la solution à partir de :

- ❖ diagramme de classe.
- ❖ diagrammes de séquence détaillée.

1. Dictionnaire des données :

1.1. Les classes et les attributs :

Classes	Attributs	Description	Types
Administrateur	ID_administrateur	Identifiant de table administrateur	Int
	Nom	le nom du compte	String
	Email	l'email	String
	Password	le mot de passe	String
Équipe	ID_équipe	Identifiant de table équipe	Int
	Nom	le nom du compte	String
	Password	Le mot de passe	string
	Chef	Le chef d'équipe	Int
	Description	Une description sur l'équipe	String
	longitude_GPS	Longitude GPS de l'adresse de l'équipe	double
	latitude_GPS	Latitude GPS de l'adresse de l'équipe	Double
Tâche	ID_tâche	Identifiant de la table tache	Int
	Titre	Le titre de la tache	String
	Priorité	La priorité de la tache	String

	Durée	Durée de la tache	Int
	Pourcentage d'avancement	Pourcentage d'avancement de la tâche.	Int
	Adresse	L'adresse.	String
	longitude_GPS	Longitude GPS de l'adresse de la tache	double
	latitude_GPS	Latitude GPS de l'adresse de la tache	Double
Message	ID_message	Identifiant de table message	Int
	Objet	L'objet du message	String
	Contenu	Le contenu du message	String
	Direction	La direction du message	booléen
Employé	ID_employé	Identifiant de table employé	Int
	Nom	le nom de l'employé	String
	Fonction	La fonction de l'employé	String
	Spécialité	La spécialité de l'employé	String

Tableau 1 : Dictionnaire de données avec les classes et les attributs.

1.2. Les opérations :

Classes	Opération	Description
Administrateur	Créer_compte ()	Créer un compte administrateur
	Consulter_compte ()	Vérifier le login de l'administrateur
Équipe	Ajouter_équipe ()	Ajouter une équipe
	Afficher_équipe ()	Afficher les informations d'une équipe .

	Modifier_équipe ()	Modifier les informations d' une équipe.
	Supprimer_équipe ()	Supprimer une équipe.
	Evaluer_rendement_équipe ()	Evaluer le rendement d'une équipe
Tâche	Ajouter_tâche ()	Ajouter une tâche.
	Modifier_tâche ()	Modifier les informations d'une tâche
	Supprimer_tâche ()	Supprimer une tâche
	Afficher_tâche ()	Afficher les informations d'une tâche
	Affecter_tâche ()	Affecter une tâche a une équipe
	Annuler_affectation_tâche ()	Annuler l'affectation d'une tâche
	Evaluer_avancement_tâche ()	Evaluer l'avancement d'une tâche (pourcentage d'avancement)
Message	Envoyer_message ()	Envoyer un message
	Consulter_messages ()	Consulter la liste des messages reçus.
	Supprimer_message ()	Supprimer un message

Tableau 2:Dictionnaire de donnée avec les opérations

1.3. Diagramme de classe détaillée :

Nous représentons dans cette partie notre système à l'aide des diagrammes de classe détaillée.

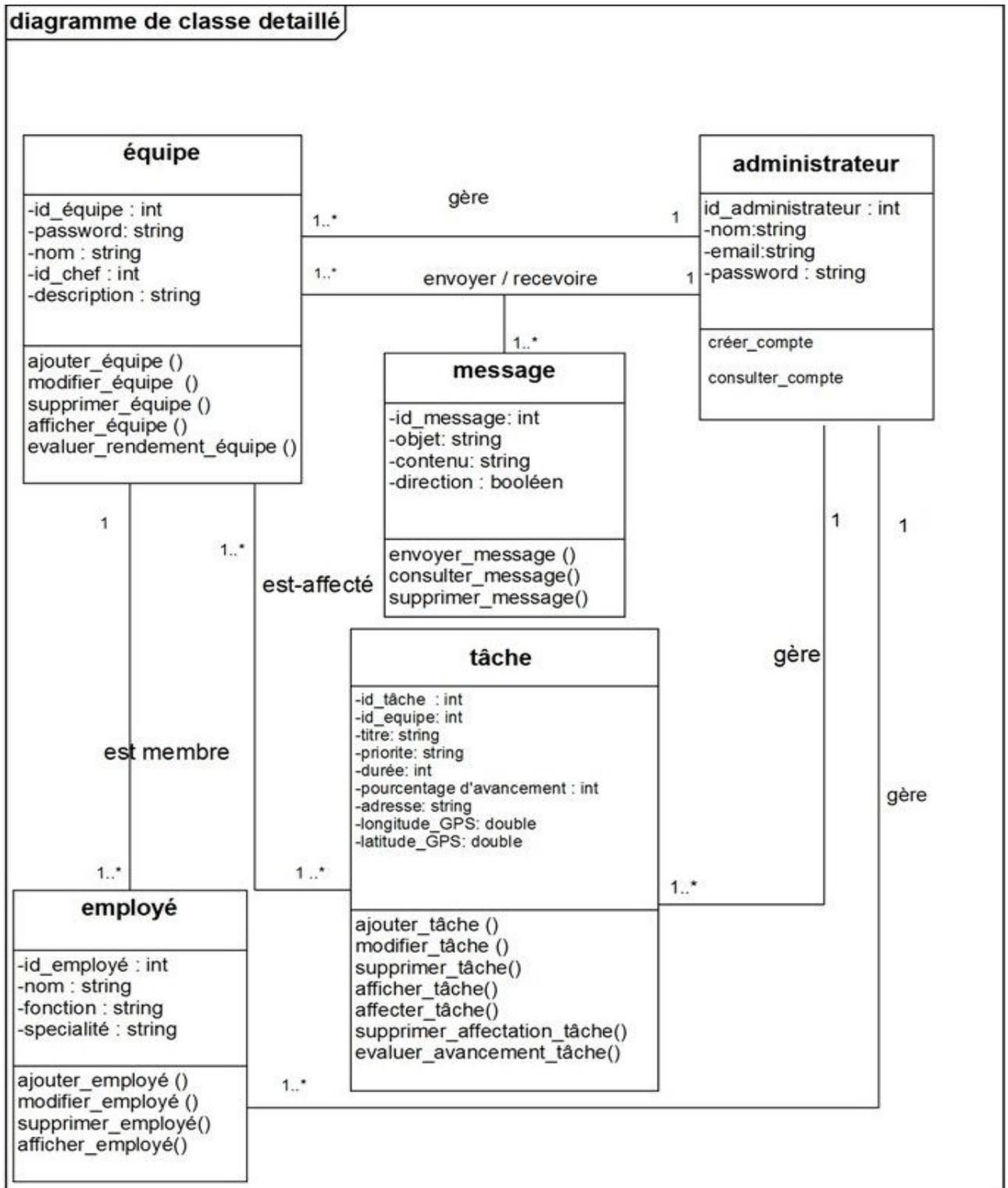


Figure 1:Diagramme de classe détaillé.

2. Passage vers le model relationnel

2.1. Les règles de passages

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règle, nous avons utilisé les règles suivantes :

- ❖ **Règle1** : toute classe devient une relation, les attributs de la classe deviennent des attributs de la relation .Si la classe possède un identifiant il devient la clé primaire de la relation, sinon, il faut ajouter une clé primaire arbitraire.
- ❖ **Règle2** : pour représenter une association 1 vers 1 (1..1) entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.
- ❖ **Règle3** : pour représenter une association 1 vers plusieurs (1..*), on procède comme une association 1 vers 1, excepté que c'est forcément la relation du coté plusieurs qui reçoit comme clé étrangère la clé primaire de la relation du coté 1.
- ❖ **Règle4** : pour représenter une association plusieurs vers plusieurs (*..*), il faut introduire une nouvelle relation dont les attributs sont les clés primaires des relations en association, et dont la clé primaire est la concaténation de ces deux attributs. Si l'association possède des attributs, ils deviennent des attributs de la relation correspondante.
- ❖ **Règle5** : cas de héritage, transformer chaque sous classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.
- ❖ **Règle6** : cas de composition, la clé primaire de la classe composée devient clé étrangère de la classe composant.
- ❖ **Règle7** : cas d'agrégation, le même principe que la Règle 3.

2.2. Les tables de la base de données

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs associations, à des tables dans la base données. Les tables générées sont :

- **Administrateur** (+Id administrateur, nom, email, password).

- équipe** (+**Id équipe**, #**id_chef_equipe**, password, nom, description, longitude_GPS, latitude_GPS).
- Employé** (+**Id employé**, #**Id équipe**, nom, fonction, spécialité).
- Tâche** (+**Id tâche**, #**Id équipe**, **titre**, priorité, durée, pourcentage d'avancement, adresse, longitude_GPS, latitude_GPS).
- Message** (+**Id message**, #**id_administrateur**, #**id équipe**, objet, contenu, direction).

3. Diagramme de séquence détaillé :

Le diagramme de séquence est un diagramme d'interaction entre les objets, qui met l'accent sur le classement des messages par ordre chronologique durant l'exécution du système. Un diagramme de séquence est un tableau dans lequel les objets sont rangés sur l'axe des abscisses et des messages par ordre d'apparition sur l'axe des ordonnées. Dans cette phase nous représentons à l'aide des diagrammes de séquences détaillés les différents cas d'utilisation de notre système.

❖ Diagramme de séquence détaillé du cas «S’authentifier » :

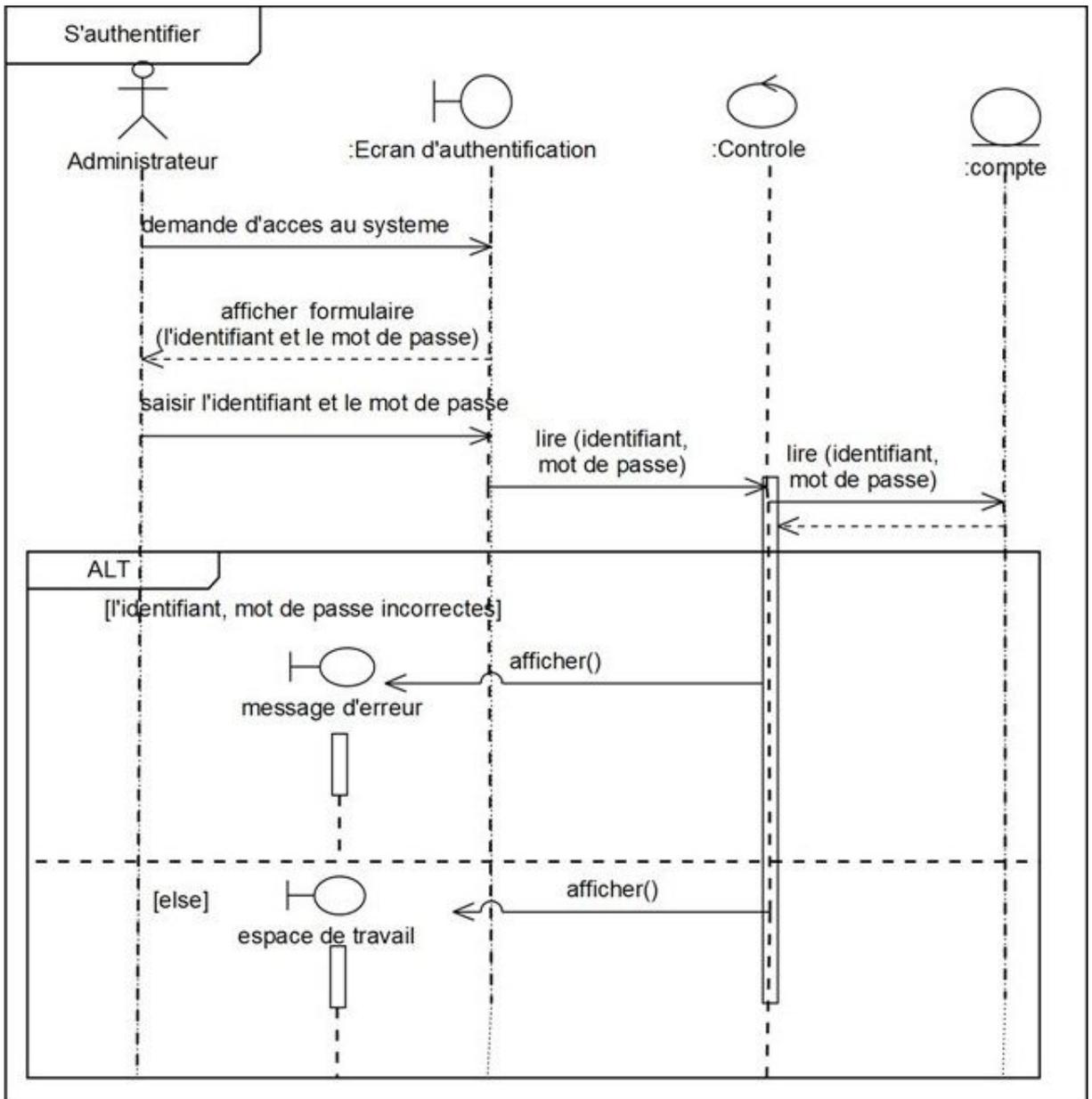


Figure 2: Diagramme de séquence détaillé du cas « S’authentifier».

❖ Diagramme de séquence du cas «gestion des équipes » :

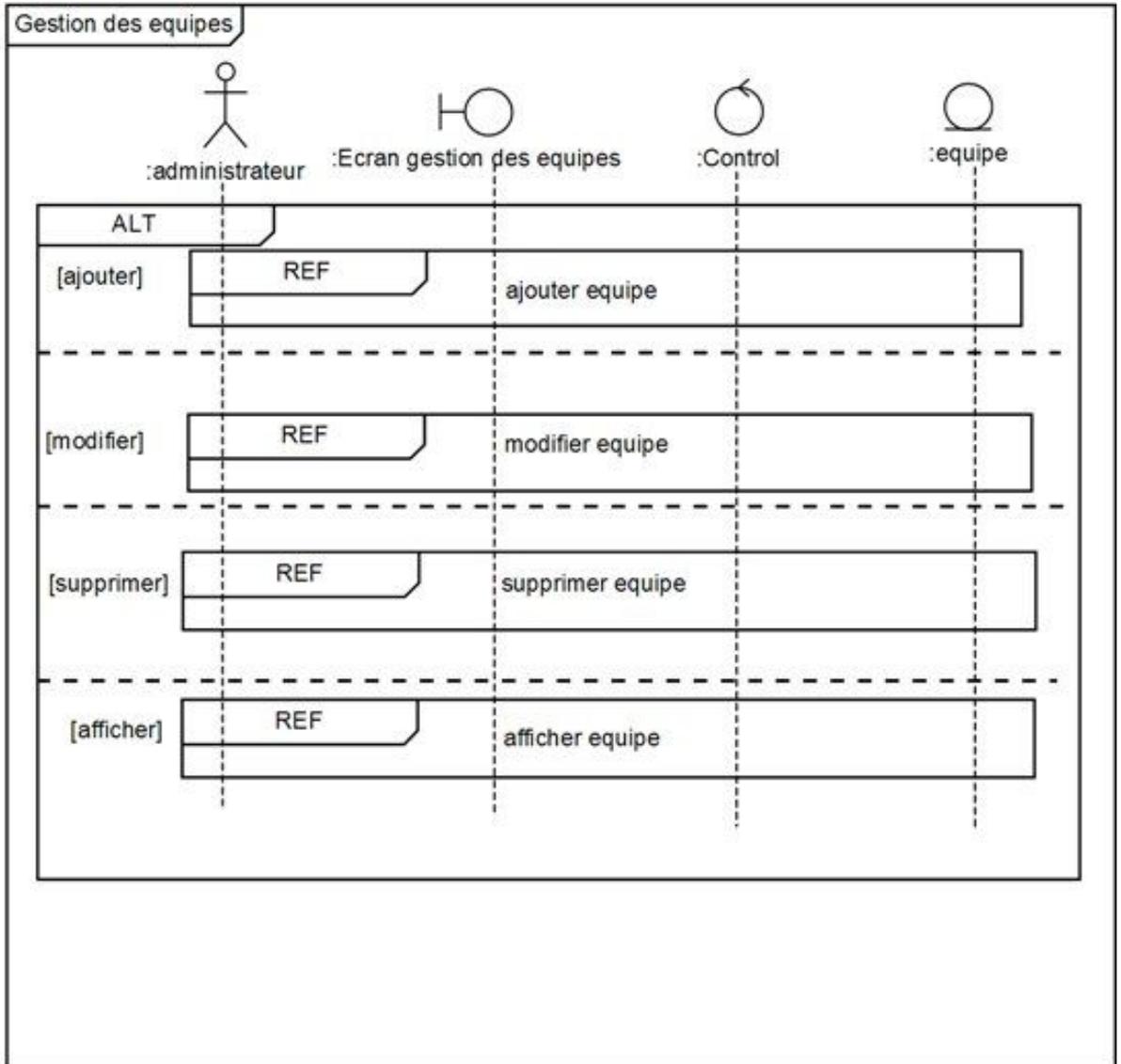


Figure 3: Diagramme de séquence détaillé du cas « gestion des équipes ».

❖ Diagramme de séquence du cas «ajouter équipe » :

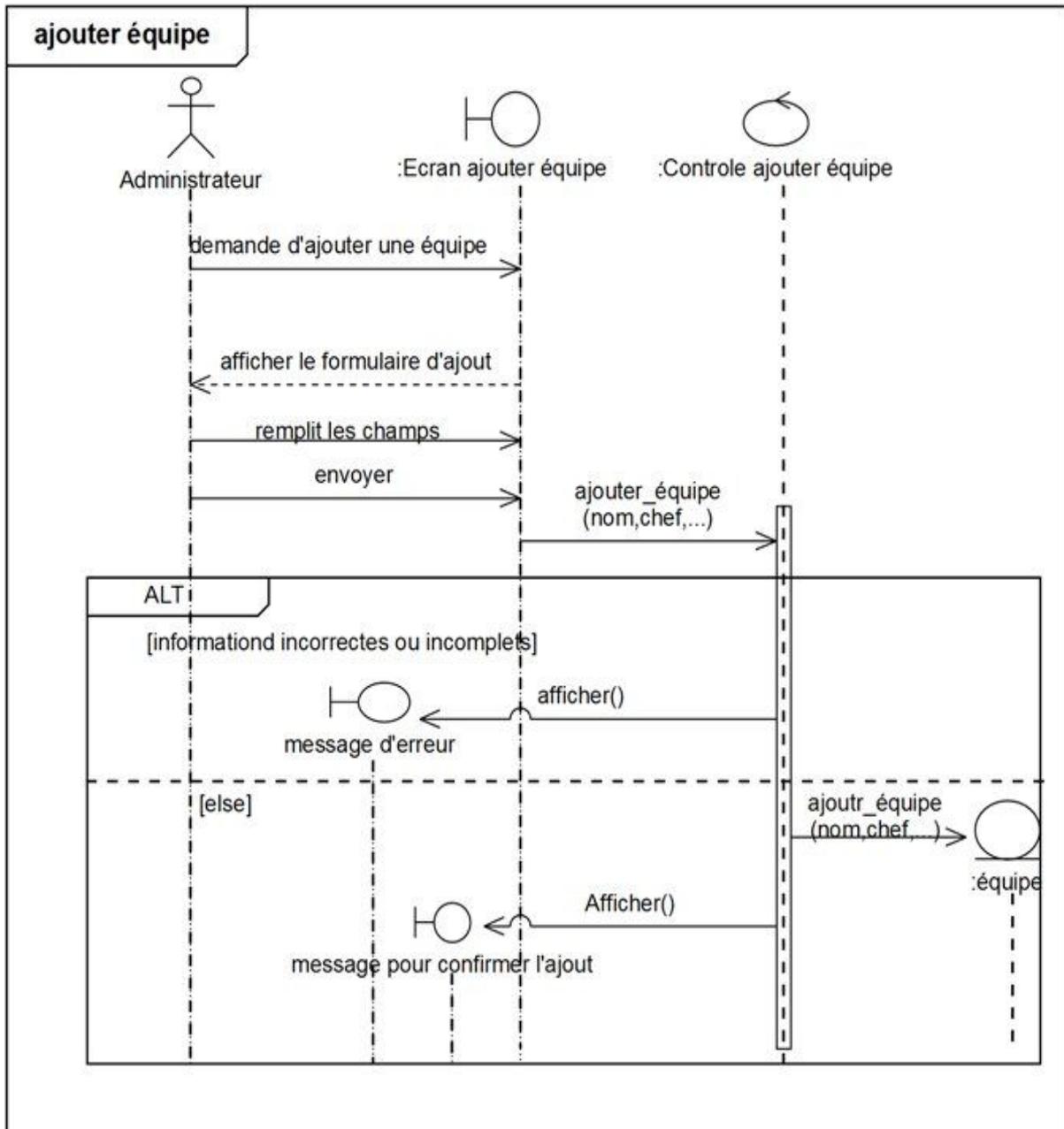


Figure 4: Diagramme de séquence détaillé du cas « ajouter équipe ».

❖ Diagramme de séquence détaillé du cas « modifier équipe » :

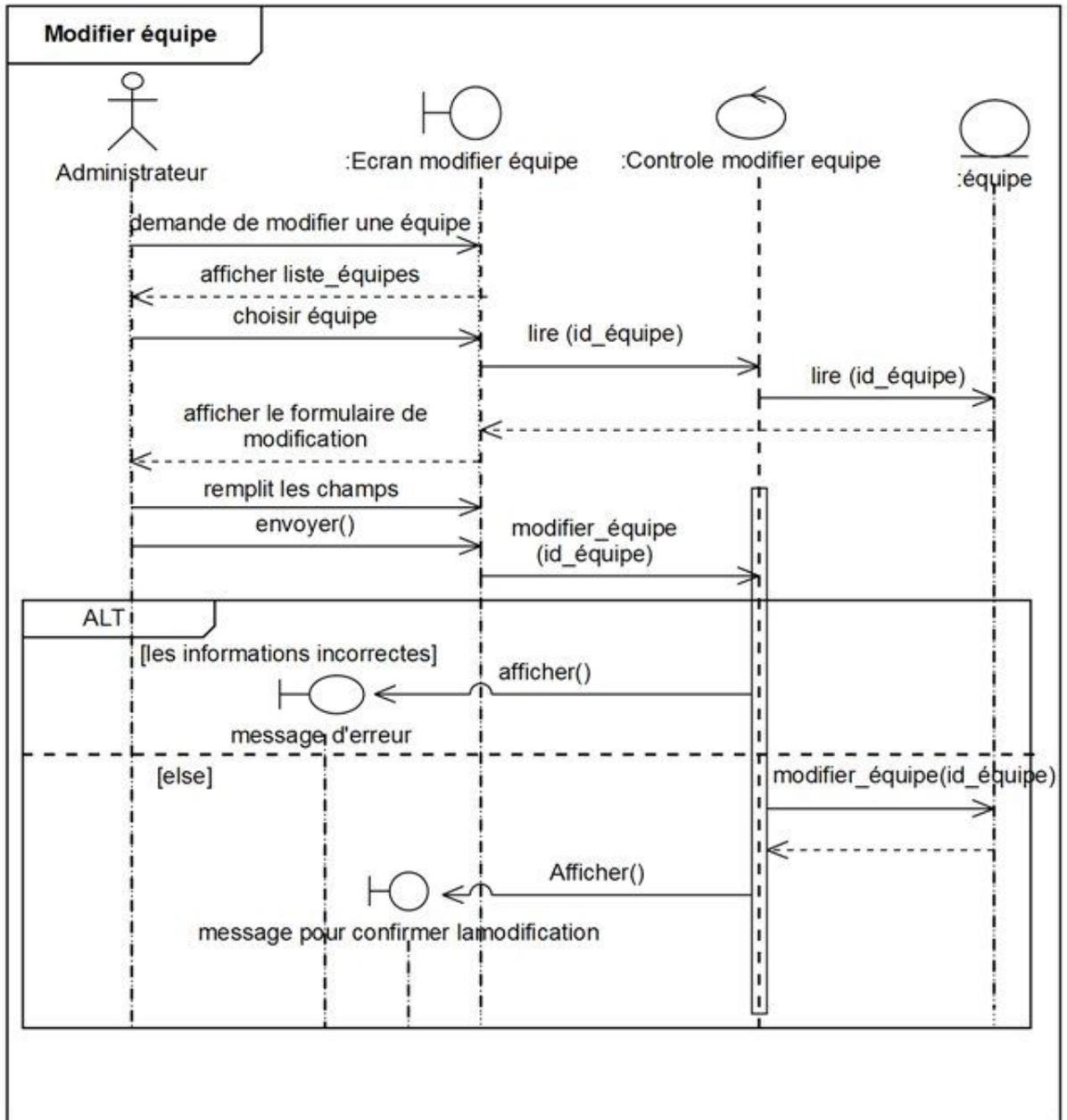


Figure 5: Diagramme de séquence détaillé du cas « modifier équipe ».

❖ Diagramme de séquence du cas «supprimer équipe » :

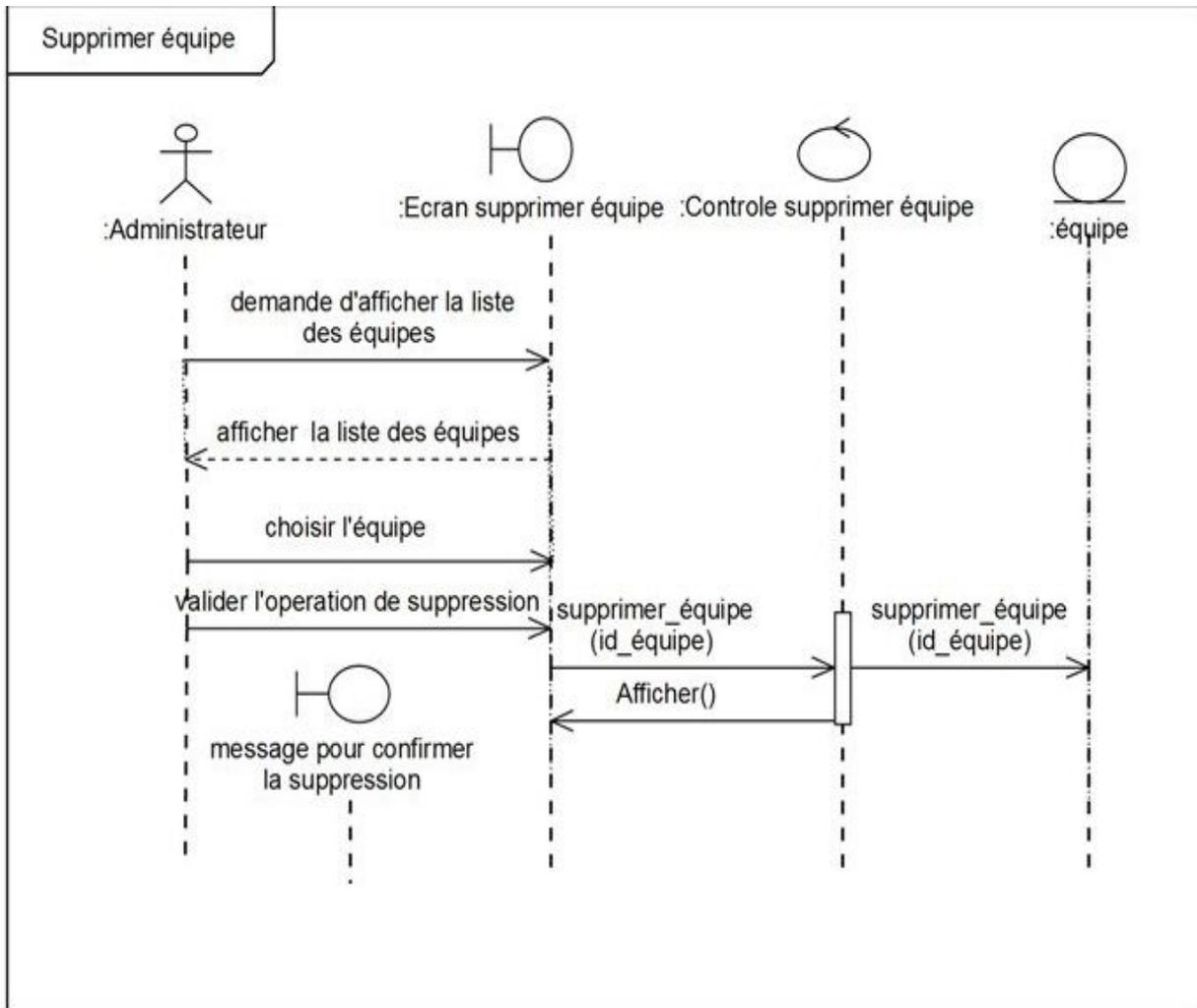


Figure 6: Diagramme de séquence détaillé du cas « supprimer équipe »

❖ Diagramme de séquence du cas «afficher équipe» :

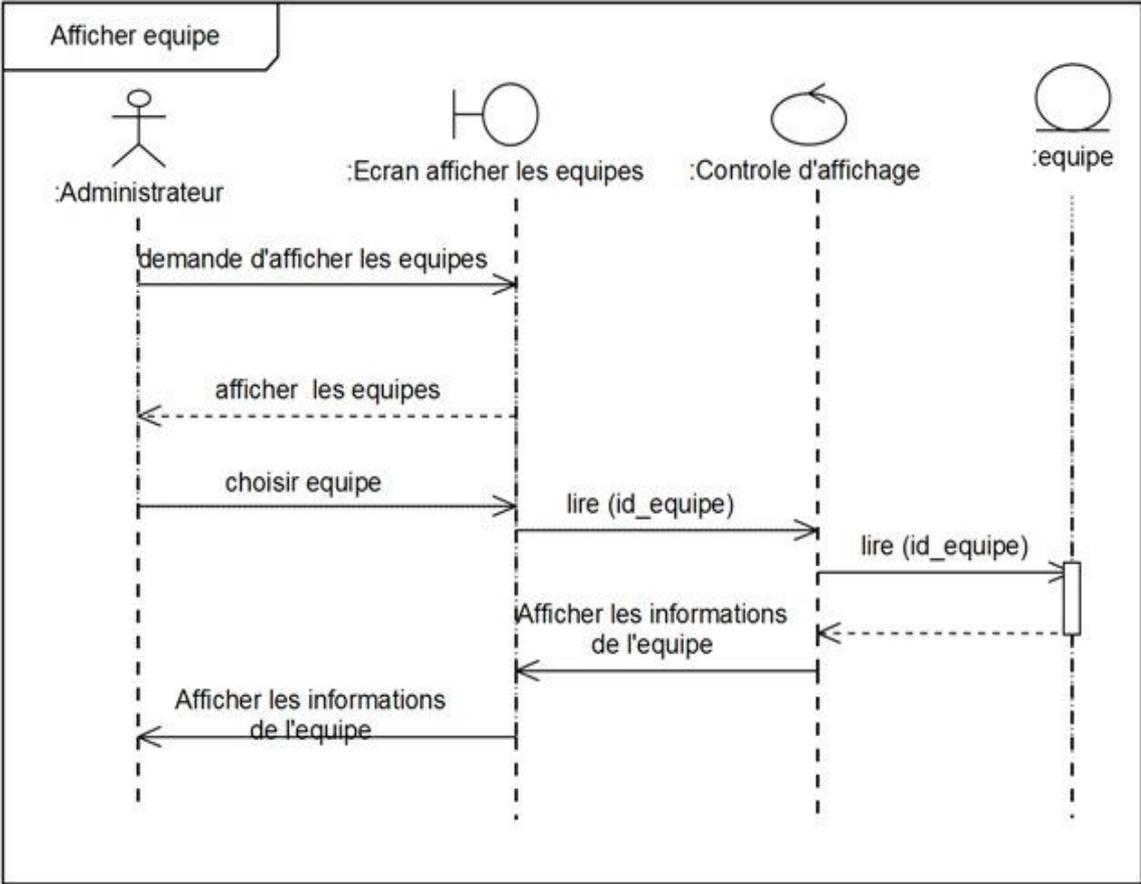


Figure 7: Diagramme de séquence détaillé du cas « afficher équipe »

❖ Diagramme de séquence du cas «gestion des tâches » :

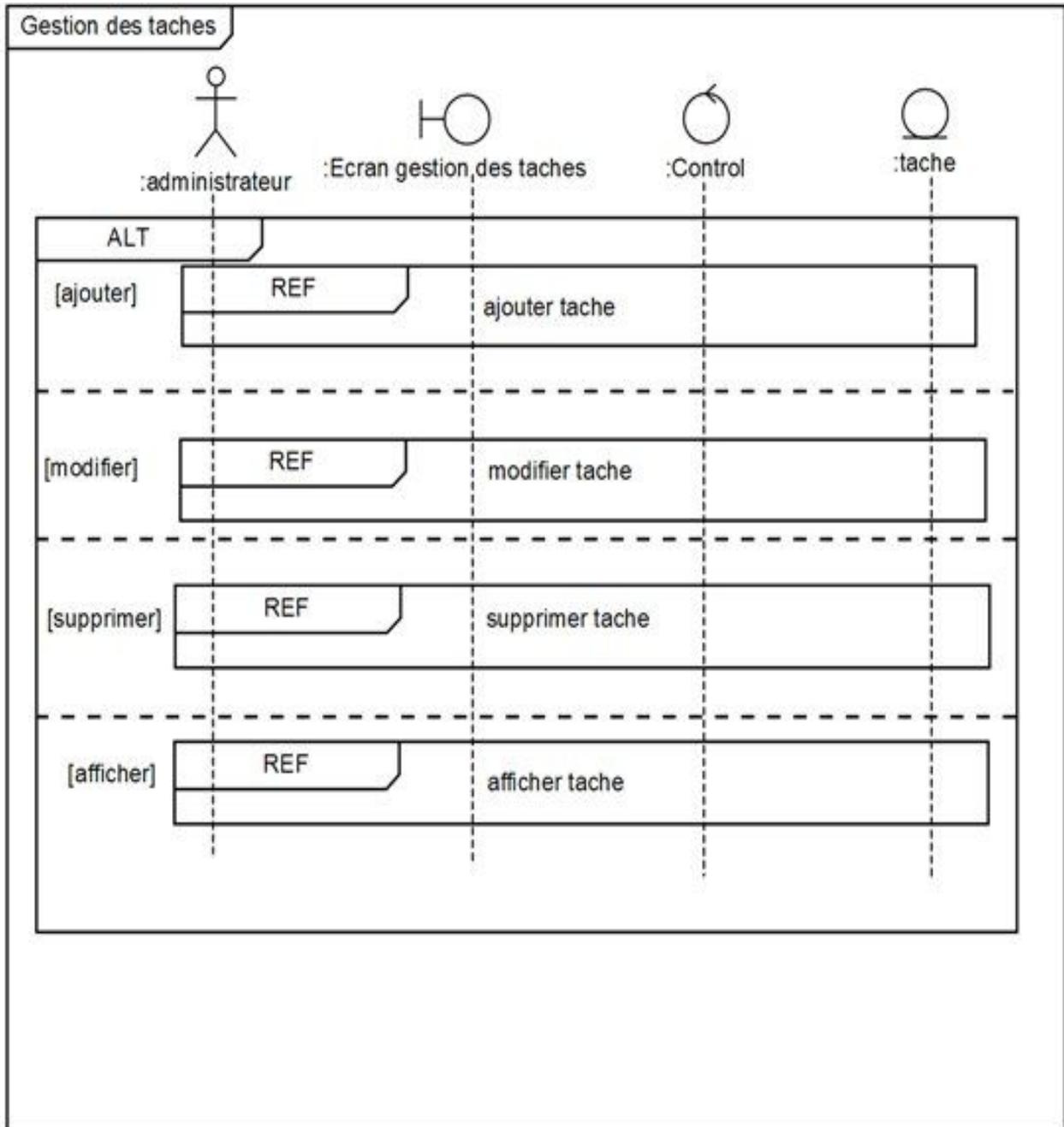


Figure 8: Diagramme de séquence détaillé du cas « gestion des tâches ».

❖ Diagramme de séquence du cas «ajouter tâche» :

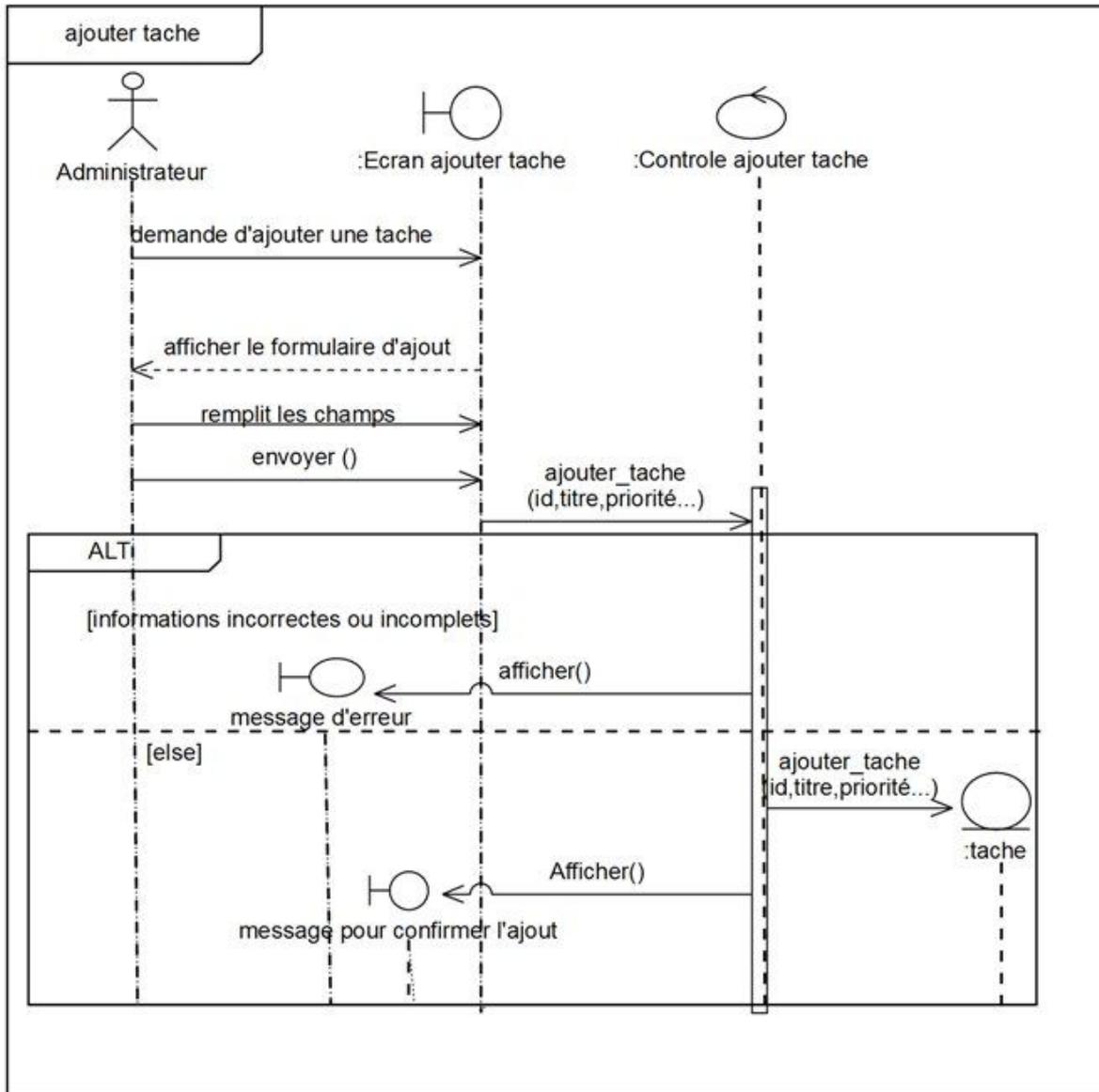


Figure 9: Diagramme de séquence détaillé du cas « ajouter tâche ».

❖ Diagramme de séquence du cas «modifier tâche» :

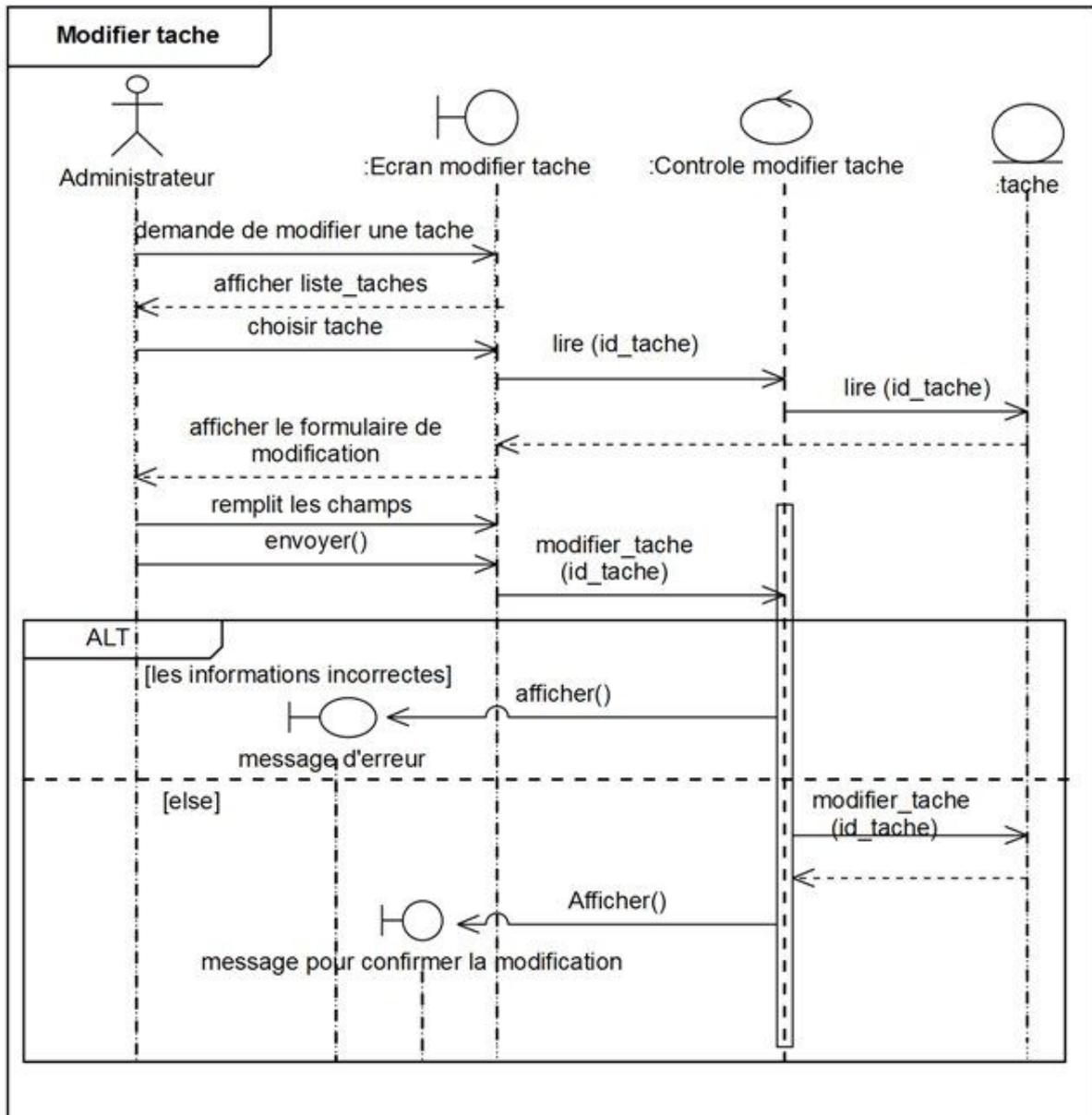


Figure 10: Diagramme de séquence détaillé du cas «modifier tâche».

❖ Diagramme de séquence du cas «supprimer tâche» :

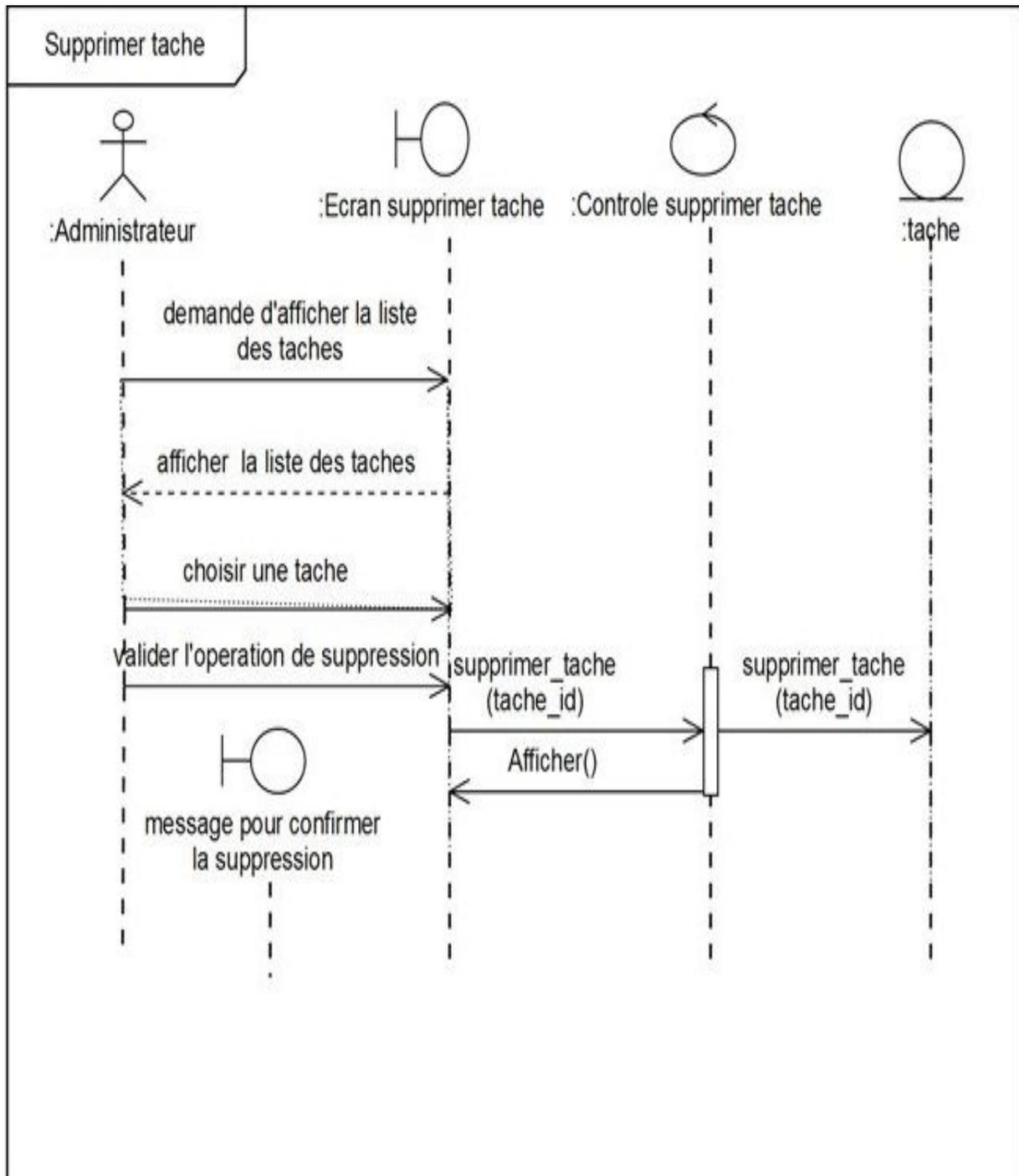


Figure 11: Diagramme de séquence détaillé du cas «supprimer tâche».

❖ Diagramme de séquence du cas «afficher tâche» :

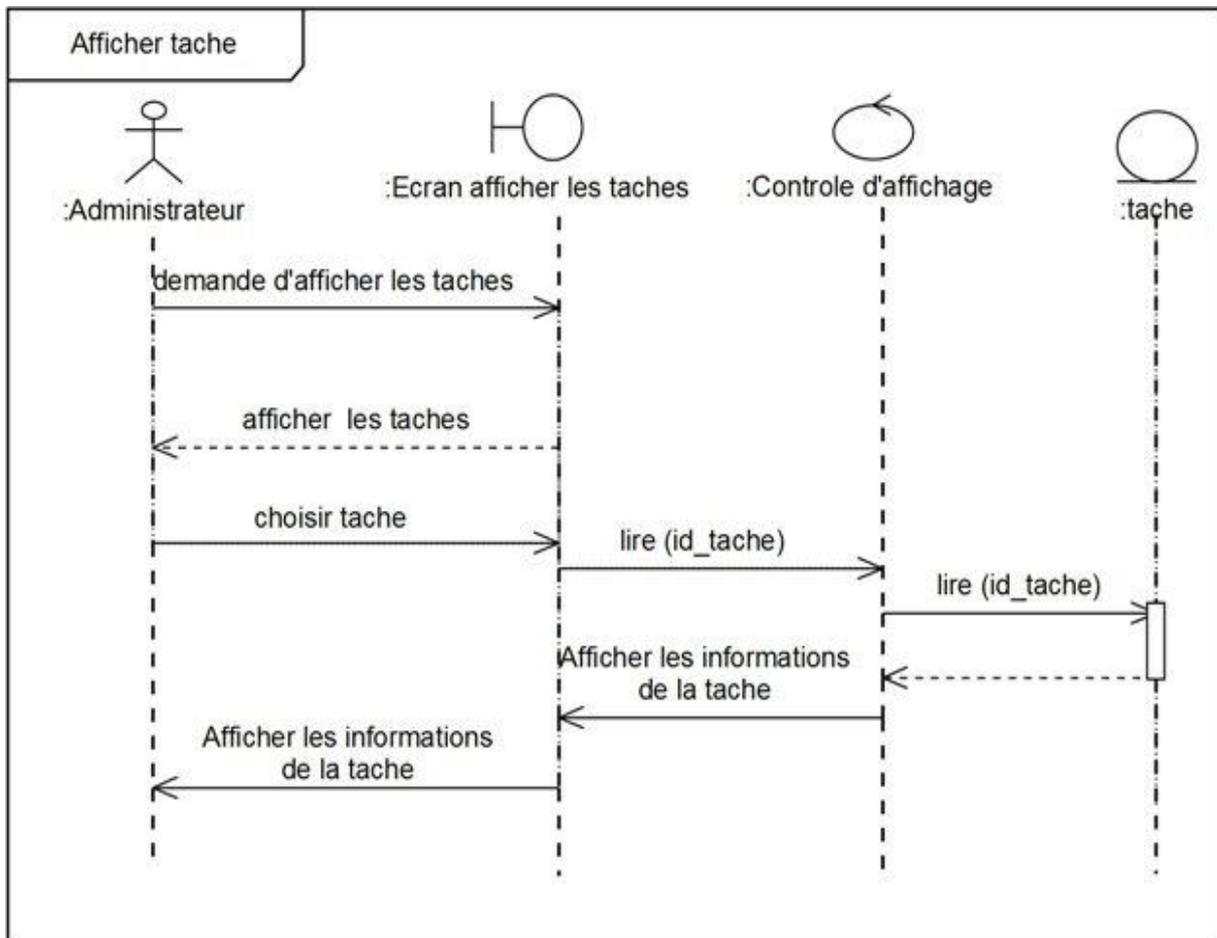


Figure 12: Diagramme de séquence détaillé du cas « afficher tâche ».

❖ Diagramme de séquence du cas « Gestion des employés » :

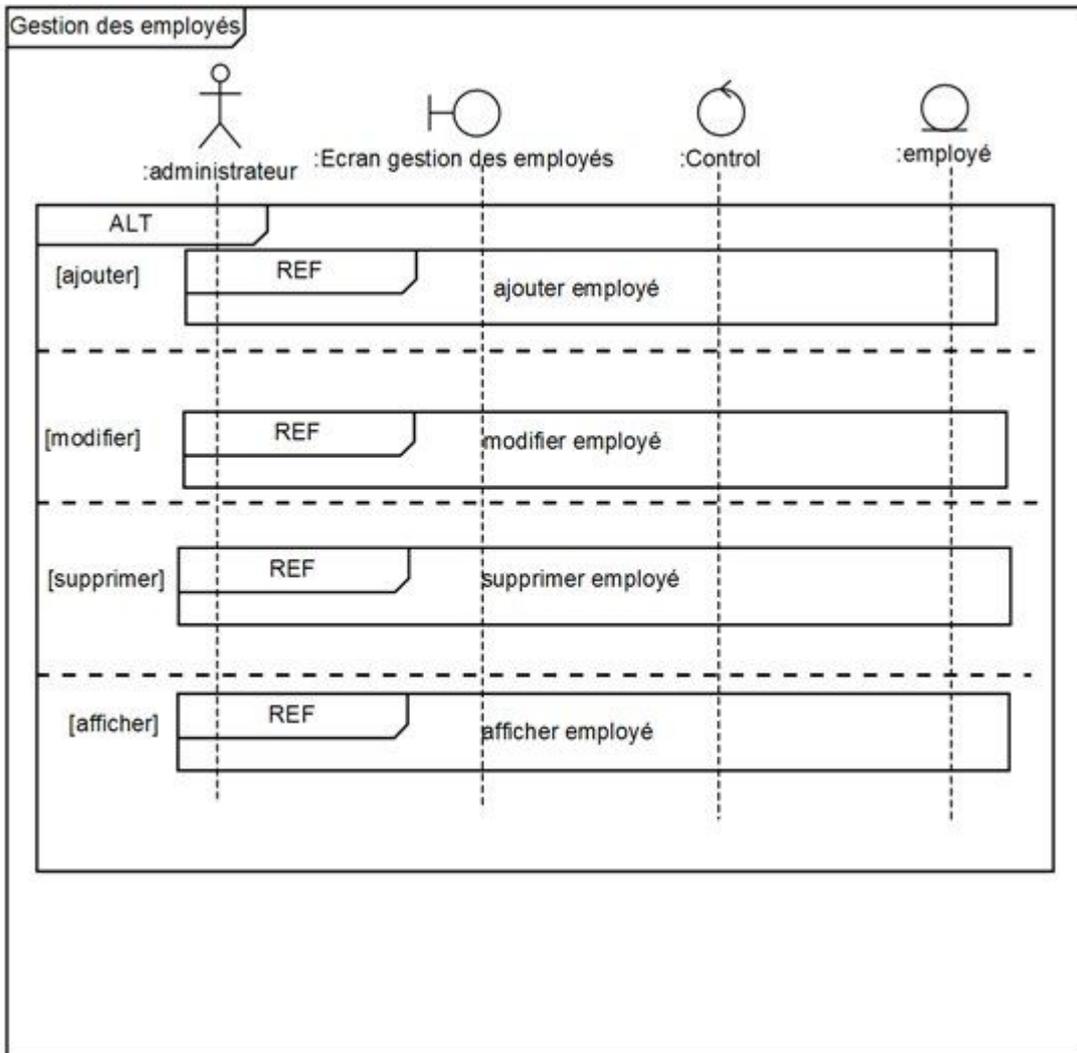


Figure 13: Diagramme de séquence détaillé du cas « gestion des employés ».

❖ Diagramme de séquence du cas « ajouter employé » :

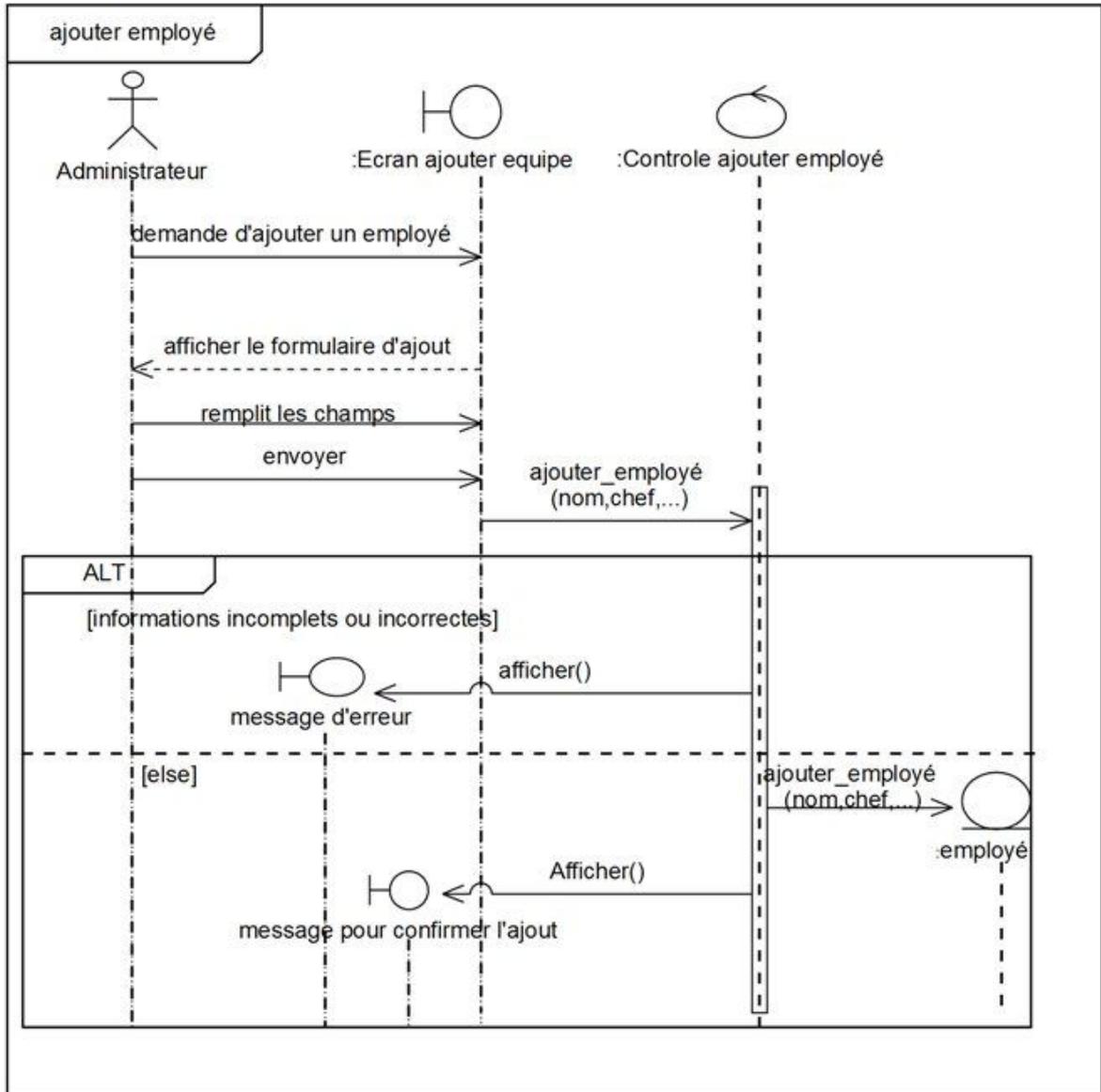


Figure 14: Diagramme de séquence détaillé du cas « ajouter employé ».

❖ Diagramme de séquence du cas «modifier employé» :

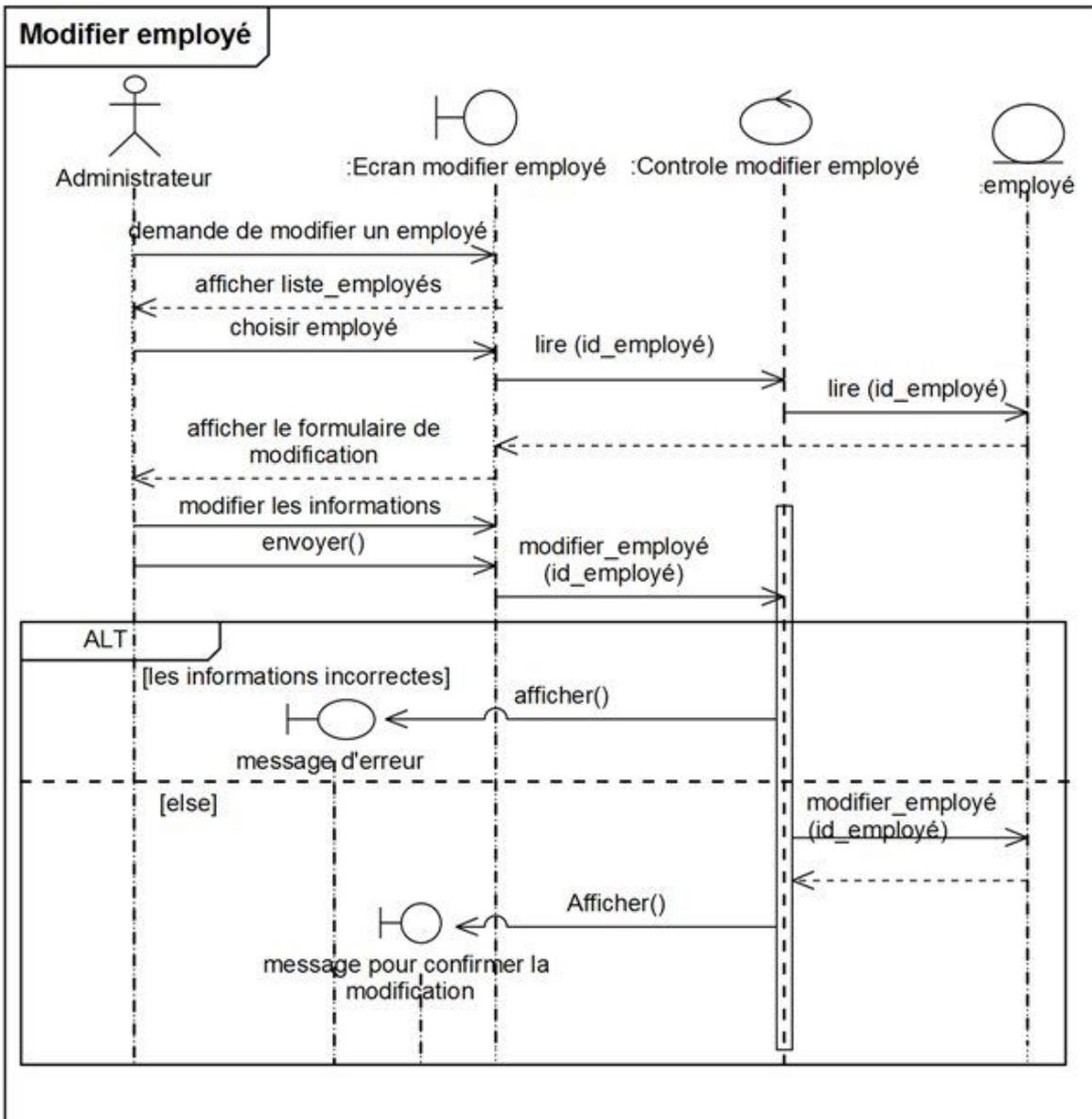


Figure 15: Diagramme de séquence détaillé du cas « modifier employé ».

❖ Diagramme de séquence du cas «afficher employé» :

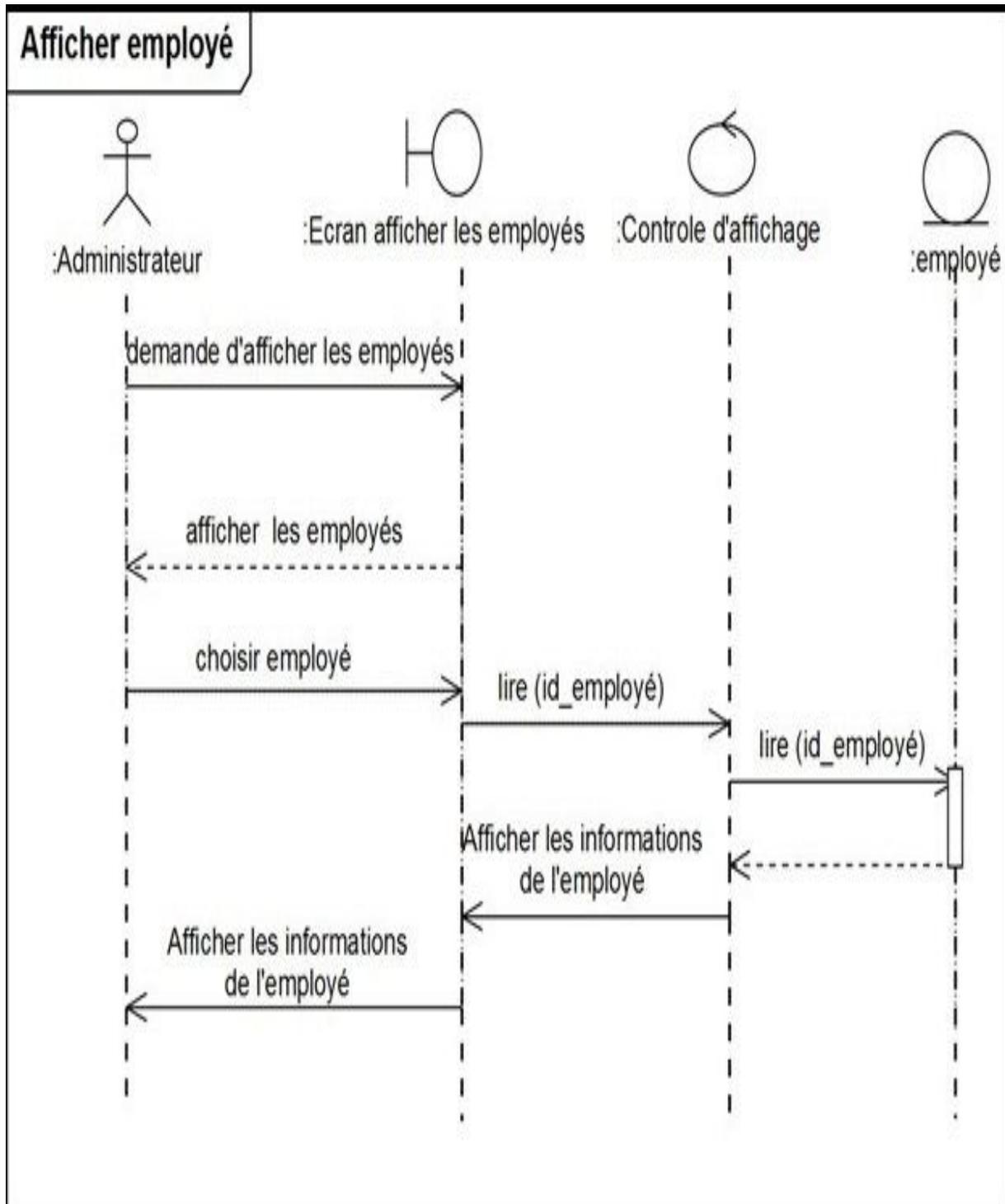


Figure 16: Diagramme de séquence détaillé du cas d'utilisation « afficher employé».

❖ Diagramme de séquence du cas «supprimer employé»

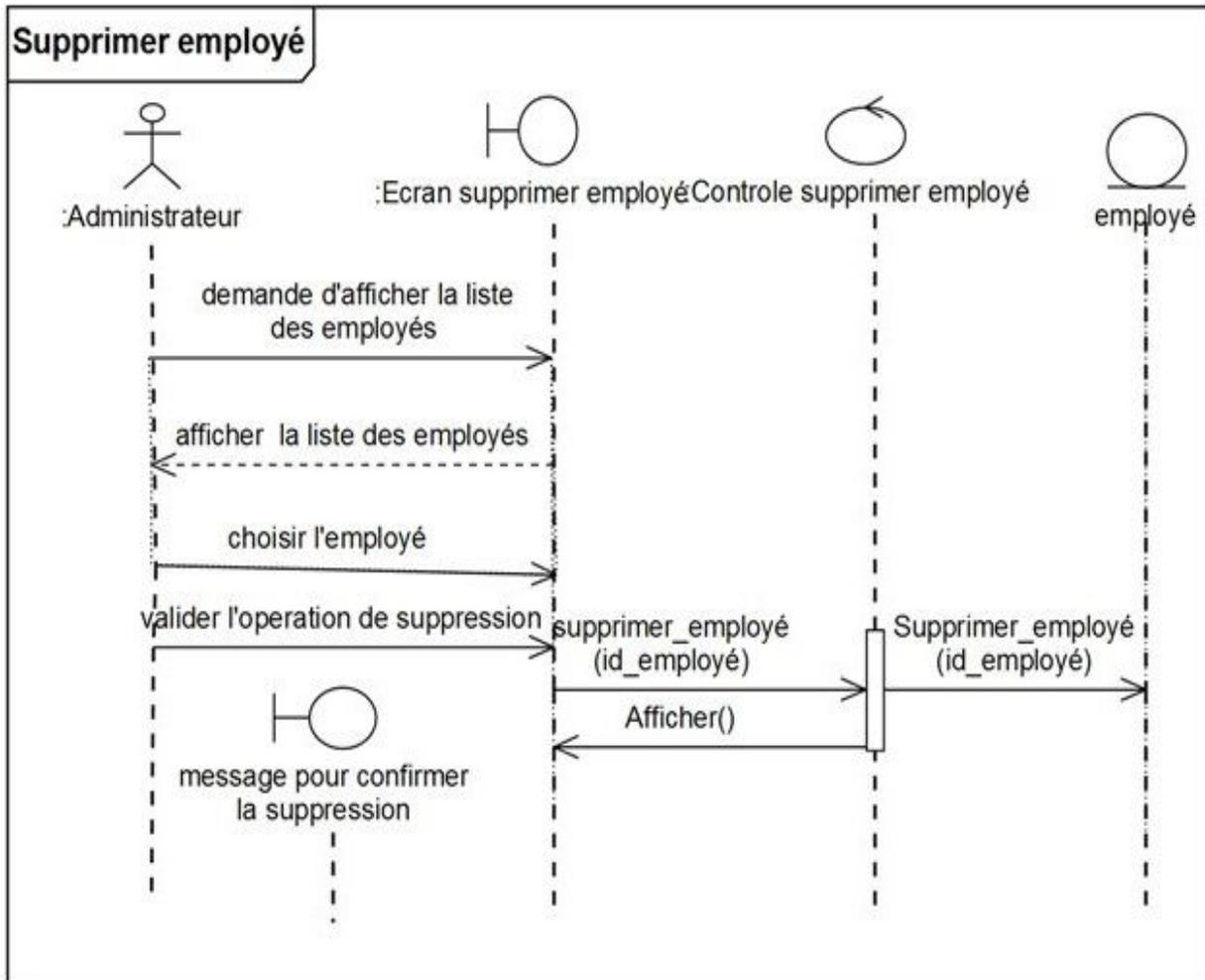


Figure 17: Diagramme de séquence détaillé du cas « supprimer employé ».

❖ Diagramme de séquence du cas «communication »

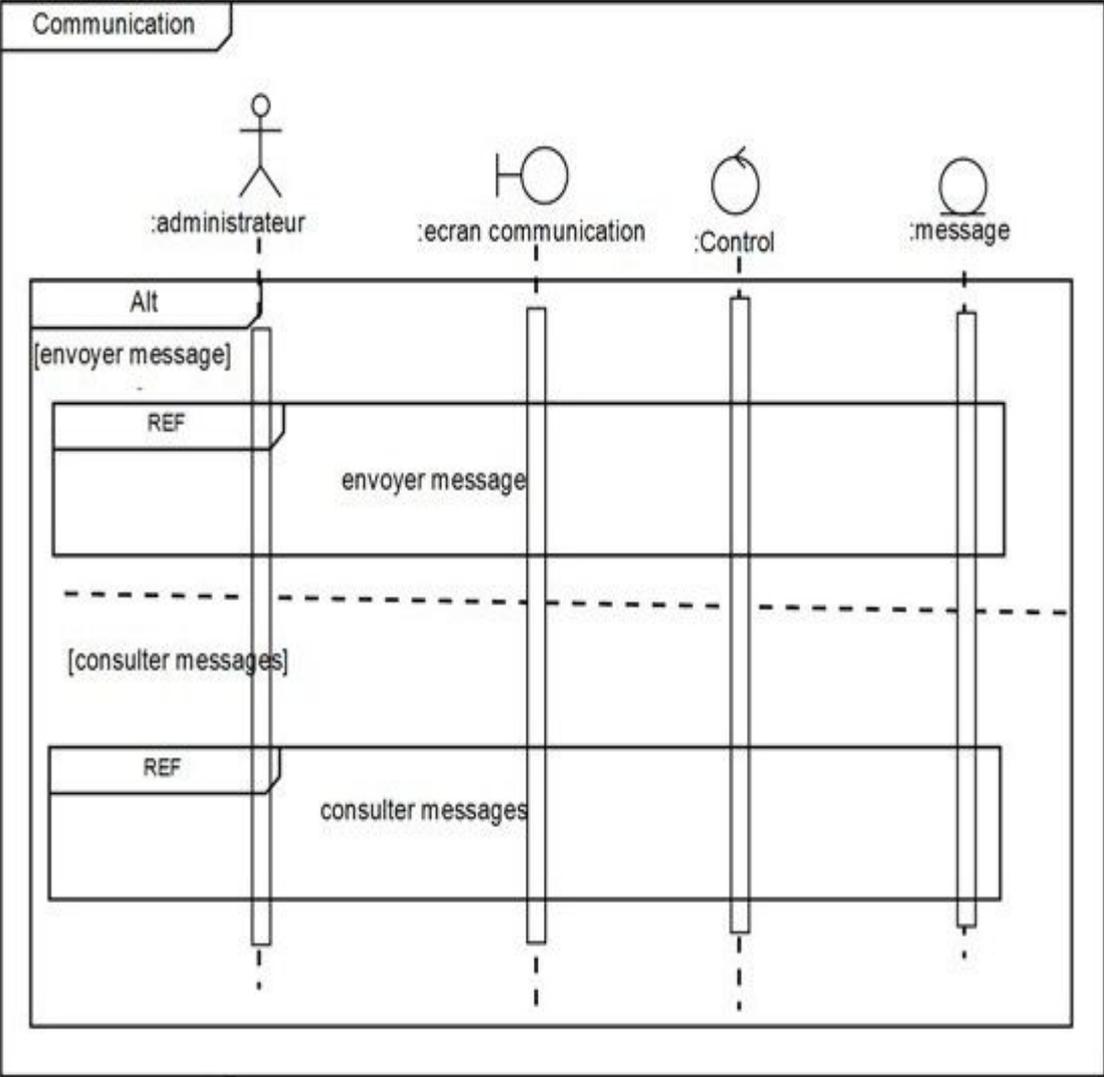


Figure 18: Diagramme de séquence détaillé du cas « communication ».

❖ Diagramme de séquence du cas «envoyer message »

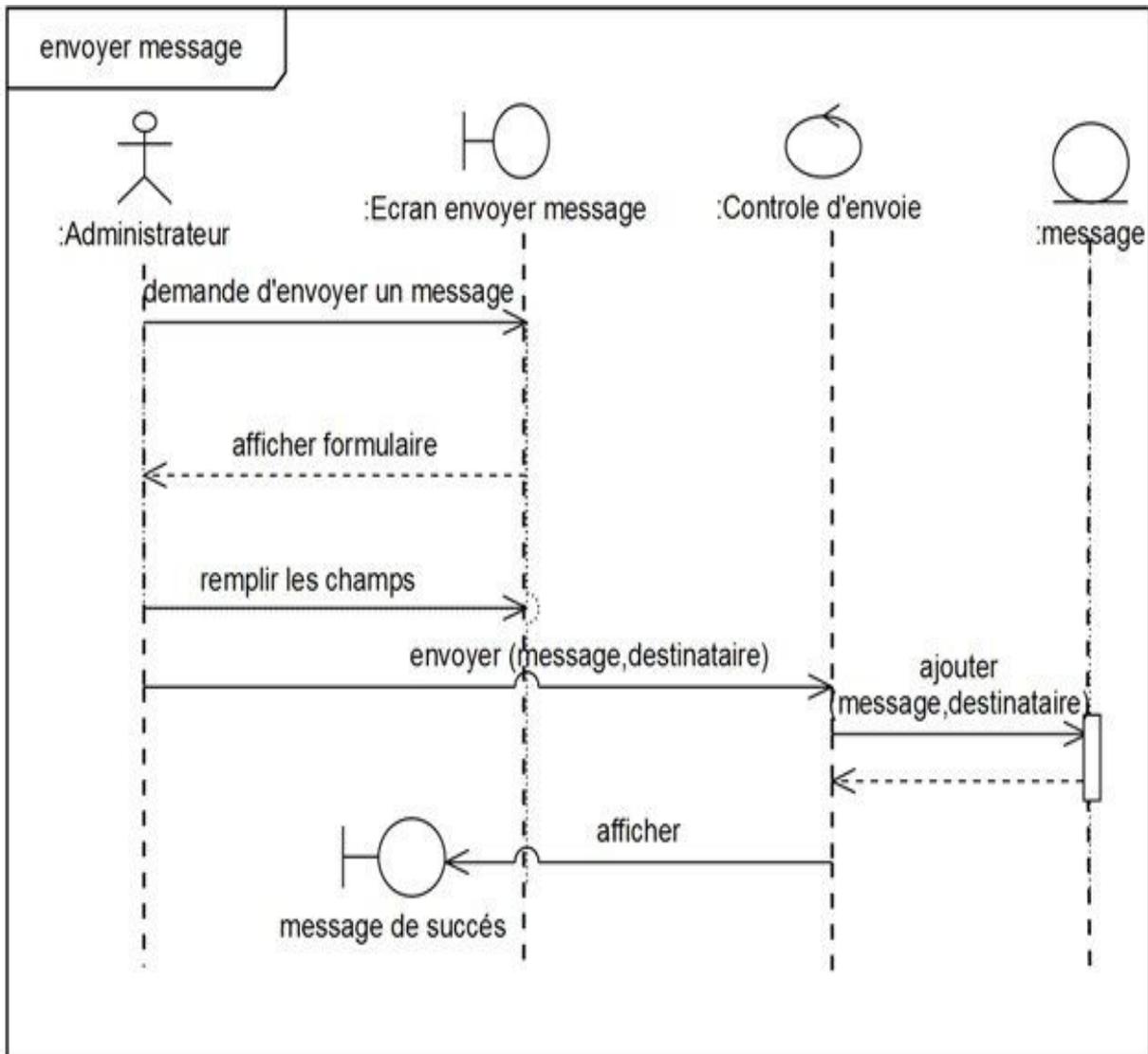


Figure 19: Diagramme de séquence détaillé du cas «envoyer message».

❖ Diagramme de séquence du cas « consulter messages »

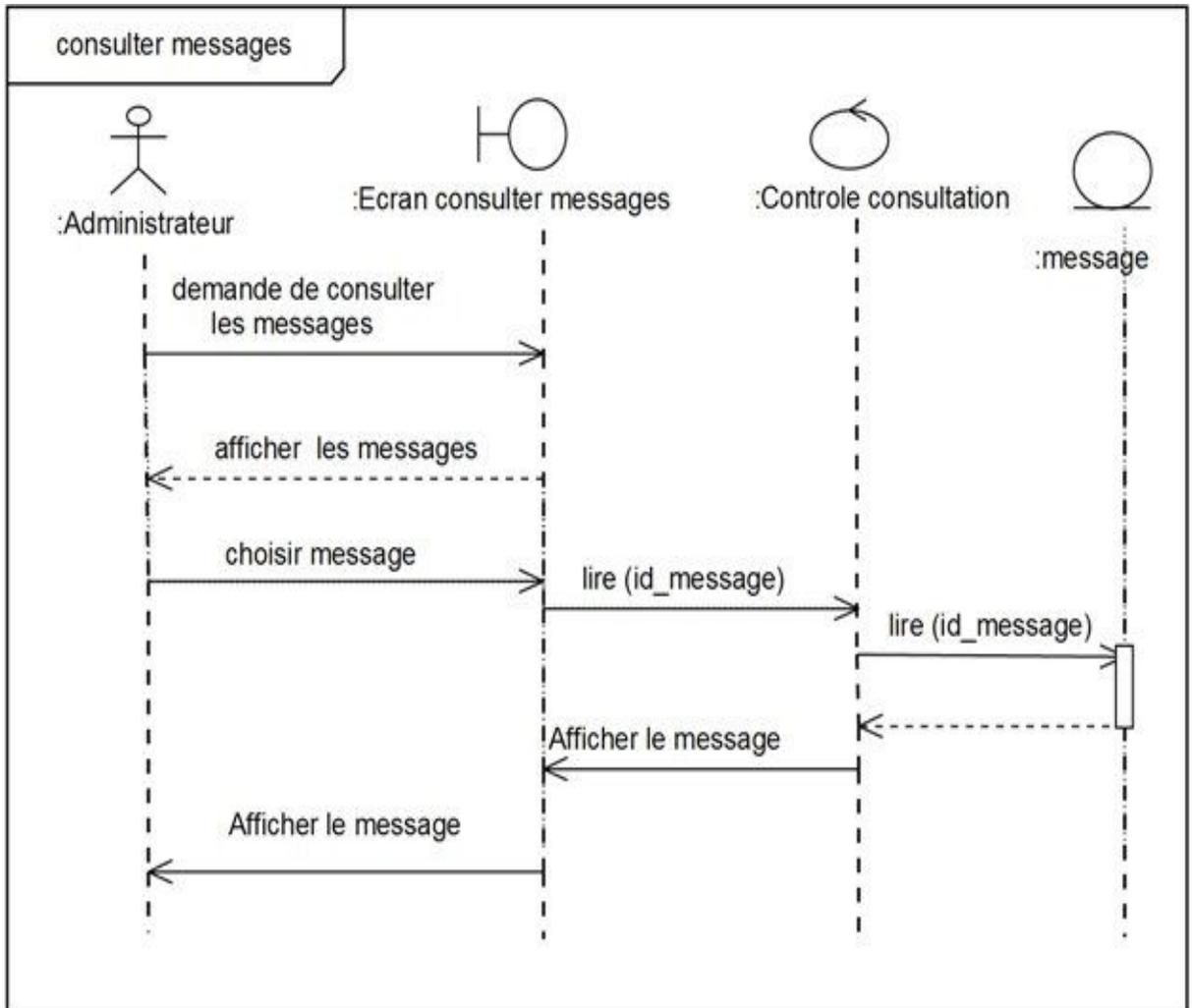


Figure 20: Diagramme de séquence détaillé du cas « consulter messages ».

❖ Diagramme de séquence du cas «suivi d'avancement» :

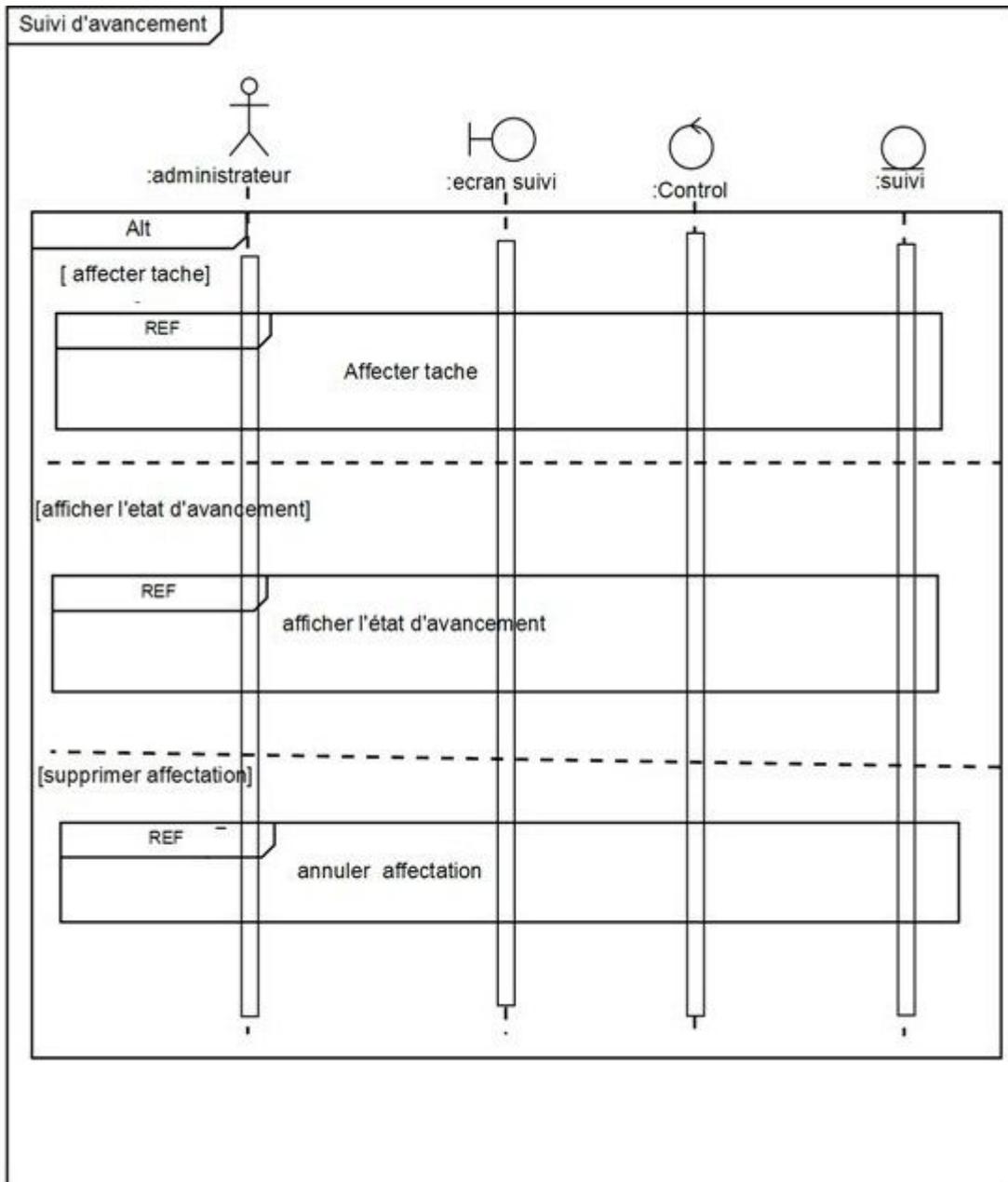


Figure 21: Diagramme de séquence détaillé du cas « suivi d'avancement ».

❖ Diagramme de séquence du cas «affecter tâche»

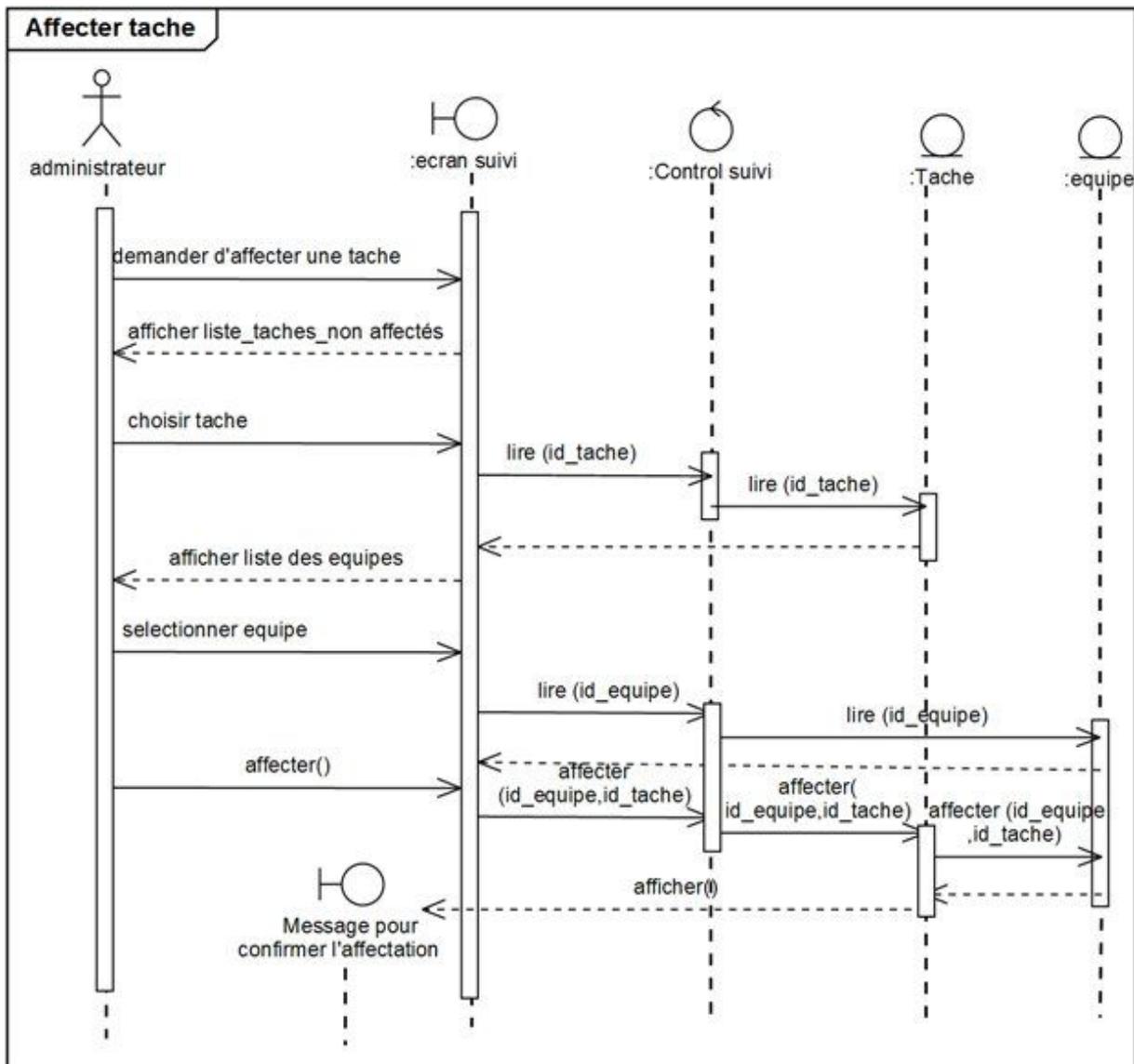


Figure 22: Diagramme de séquence détaillé du cas «affecter tâche»

❖ Diagramme de séquence du cas «annuler affectation»

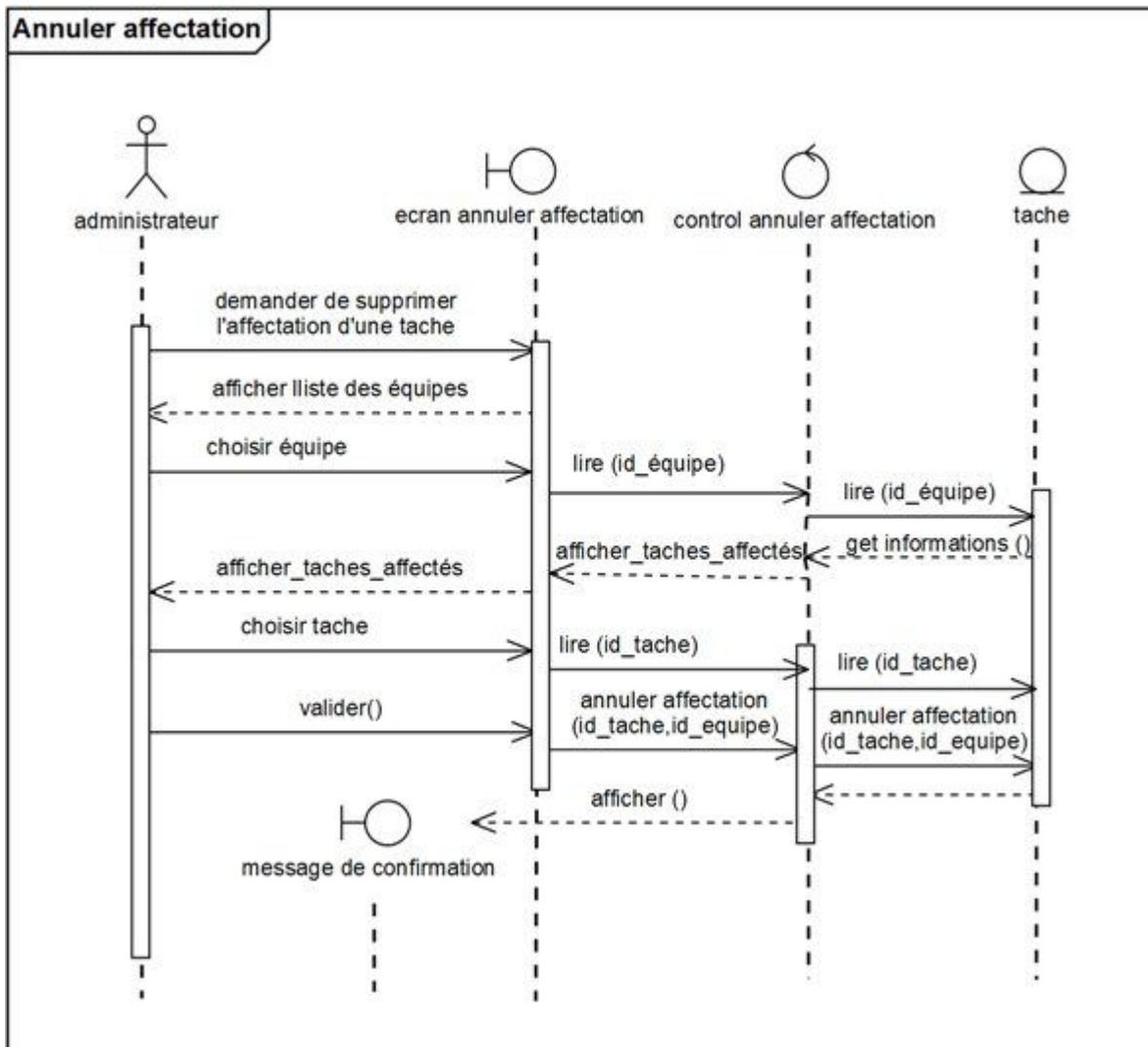


Figure 23: Diagramme de séquence détaillé du cas «annuler affectation»

❖ Diagramme de séquence du cas «afficher l'état d'avancement»

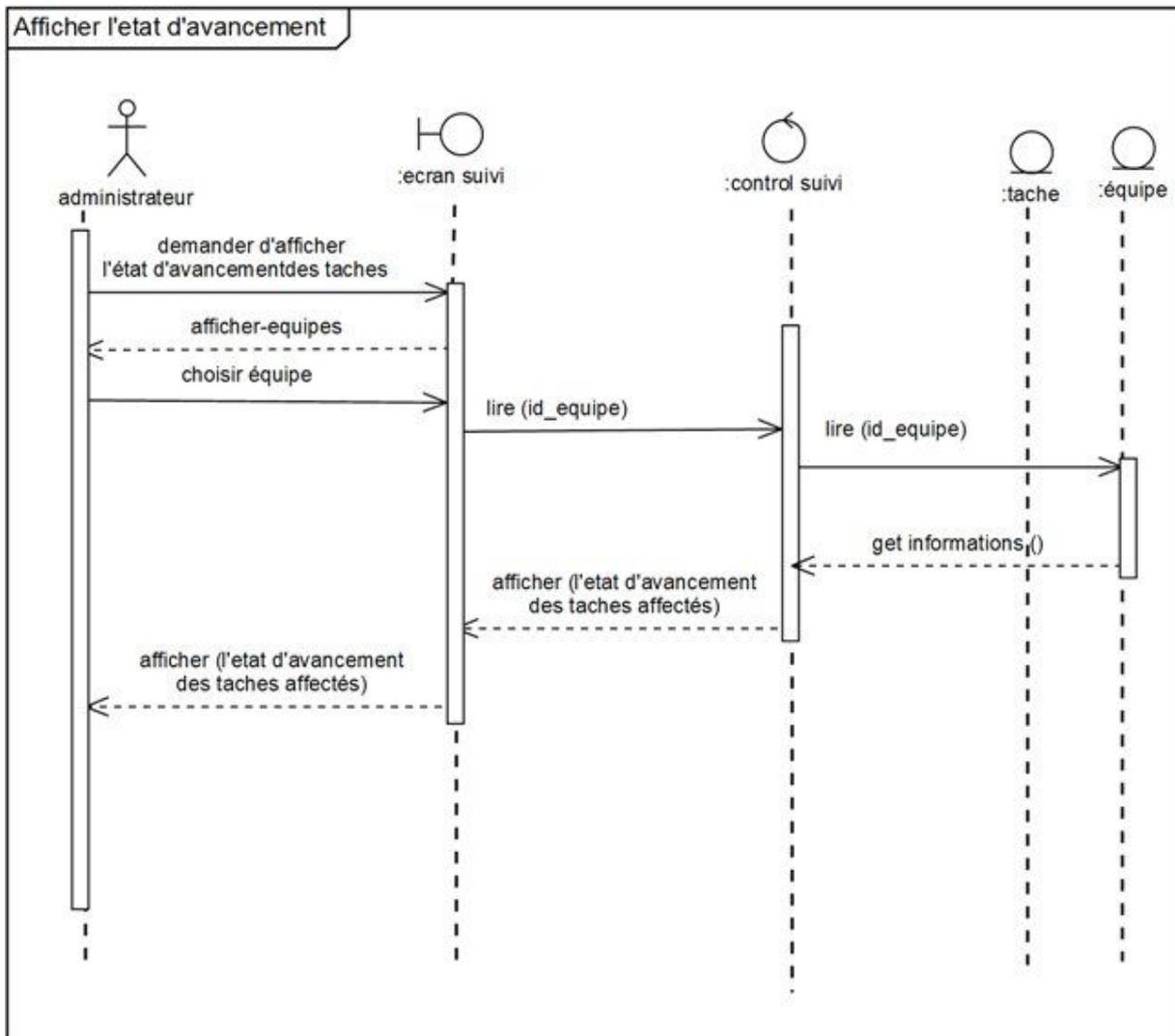


Figure 24: Diagramme de séquence détaillé du cas «afficher l'état d'avancement»

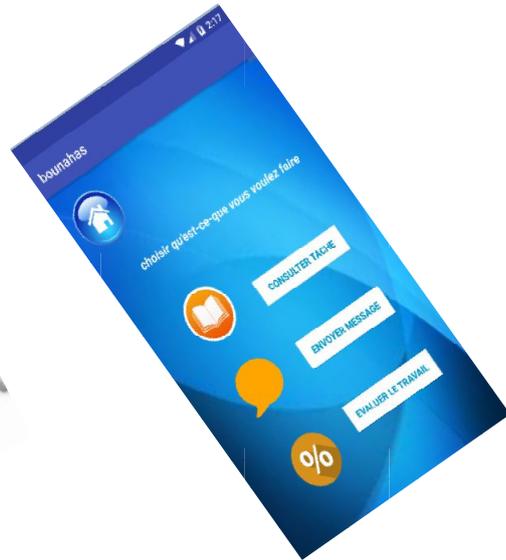
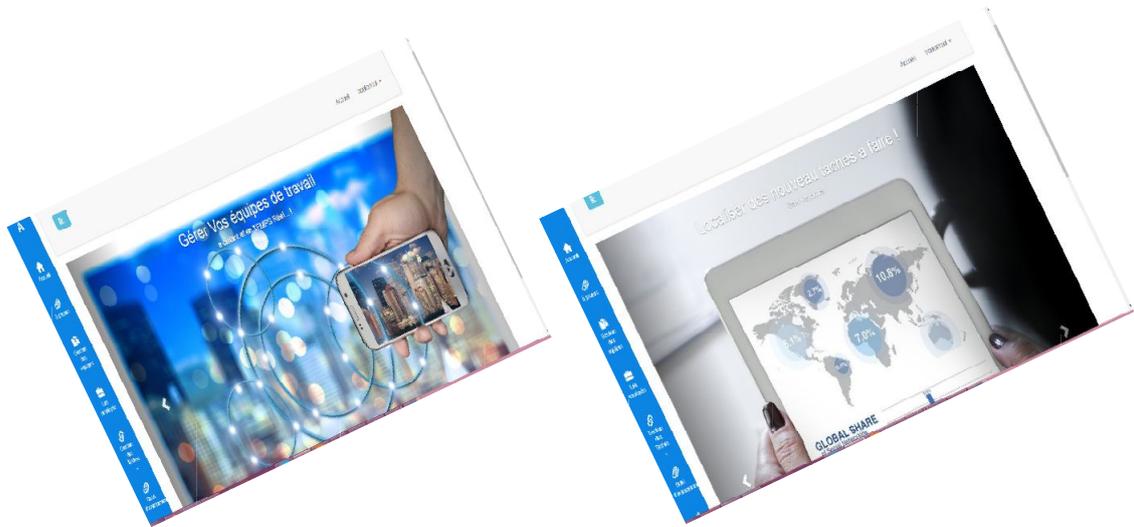
Conclusion

Dans ce chapitre nous avons détaillé les différentes classes du système et élaborer le modèle relationnelle, qui définit l'ensemble des tables de la base de données de notre système.

Le chapitre suivant, qui est le dernier, explique les détails des choix d'implémentation de notre application.

CHAPITRE 4

Réalisation



Introduction

La phase de réalisation consiste à passer du résultat de la conception détaillée à un ensemble de programmes. Dans ce chapitre, nous présentons, en premier lieu, l'environnement de travail et les outils de développement utilisés pour implémenter les différents composants de notre application, ensuite nous donnons quelques captures d'écran des différents l'interfaces.

1. Présentation des outils et technologies utilisée

Nous allons dans cette partie citer et décrire les différents outils et technologie utilisés pour implémenter le coté web (serveur Web) et le coté Smartphone (Application Android) de notre application.

1.1. Développement du coté web de l'application

1.1.1. Le Framework de développement Laravel

Un Framework est, comme son nom l'indique en anglais, un « **cadre de travail** ». L'objectif d'un Framework est généralement de simplifier le travail des développeurs informatiques (les codeurs si vous préférez), en leur offrant une architecture « **prête à l'emploi** » et qui leur permette de ne pas repartir de zéro à chaque nouveau projet. Laravel est un Framework PHP open sources basé sur l'architecture MVC (modèle, vue, contrôleur). Sortie en juin 2011, il est actuellement l'un des Framework PHP les plus populaires. Sur notre projet nous avons utilisé la version 5.6.

La figure 1 représente logo du Framework.



Figure1 : logo de laravel

1.1.2. MySQL:

MySQL est un système de gestion de base de données. Son rôle est de stocker et de gérer une grande quantité de données en les organisant sous forme de tables, et de permettre la manipulation de ces données à travers le langage de requête SQL. On ne s'occupe plus alors de la manière dont les données sont stockées sur le disque dur, de simples instructions permettent d'ajouter, de supprimer, de mettre à jour et surtout de rechercher des données dans une base de données.

Sur notre projet nous avons utilisé la version 5.7.19.

1.1.3. Le Framework Bootstrap :

Bootstrap est un Framework CSS, il consiste en un ensemble de fichiers CSS et JavaScript utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc.) de sites et des applications web. Bootstrap contient des formulaires, boutons, outils de navigation et autres éléments interactifs, il permet, grâce à son système de grilles, de développer des interfaces qui s'adaptent selon la taille du média utilisé pour l'affichage (Moniteur, Tablette, Smartphone..etc). Sur notre projet nous avons utilisé la version 3.3.7.

La figure 2 représente logo de Bootstrap.



Figure2 : logo Bootstrap

1.1.4. JQuery

JQuery est la bibliothèque JavaScript la plus utilisée actuellement, elle permet de créer des effets dynamiques sur les pages web comme des changements de couleur, des animations, et des effets de fondu. Les combinaisons sont illimitées. Sur notre projet nous avons utilisé la version 3.2.1.

1.1.5. La technologie Ajax :

L'Ajax (Asynchronous Javascript and Xml) est une technologie qui permet le rafraîchissement partiel des pages web, ce qui permet de réduire le temps de latence comparé aux applications Web classiques.

1.2. Développement de la partie Smartphone (Application pour Android)

1.2.1. Langage de programmation java

Pour programmer la partie Android de notre application nous avons choisi d'utiliser le langage Java. Ce dernier est à la fois un langage de programmation et un environnement d'exécution. Le langage java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation. Lors de la création du langage java, il avait été décidé que ce langage devait répondre aux 5 objectifs suivants : Utiliser une méthode orientée objet, permettre à un même programme d'être exécuté sur plusieurs systèmes d'exploitation différents. Pouvoir utiliser de manière les réseaux informatiques, pouvoir exécuter du code distant de manière sûre, être facile à utiliser et posséder les points forts des langages de programmation orientés objet comme C++. [11]

1.2.2. Le SDK Android

Un SDK est un ensemble d'outils permettant de créer des applications. Android SDK est un outil indispensable pour développer des applications Android. Ce dernier a été développé par Google, il nécessite cependant l'installation du Java développement Kit (JDK).[12]

1.2.3. L'environnement de développement Android Studio

Android Studio est un ensemble complet d'outils de développement Android. Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriaux. Les plateformes de développement prises en charge par ce kit sont les distributions sous Noyau Linux, Mac OS X 10.5.8 ou plus, Windows XP ou version ultérieure. L'IDE officiellement supporté était Eclipse combiné au plugin d'outils de développement d'Android (ADT), mais depuis 2015, Google officialise Android Studio qui devient alors l'IDE officiel pour le SDK Android.

La figure suivante représente logo d'Android studio.



Figure 3: Logo de l'IDE Android Studio.

2. Les différentes interfaces de l'application

Nous présentons dans ce chapitre quelques captures d'écran des interfaces principales de notre application qui illustrent les différents cas d'utilisation déjà vus dans le chapitre 2.

2.1. Au niveau du serveur

❖ L'interface d'authentification

Cette interface permet aux utilisateurs d'accéder à l'application pour l'exploitation des services offerts par celle-ci. Si le mot de passe est incorrect, le système affiche une notification pour déterminer l'erreur.

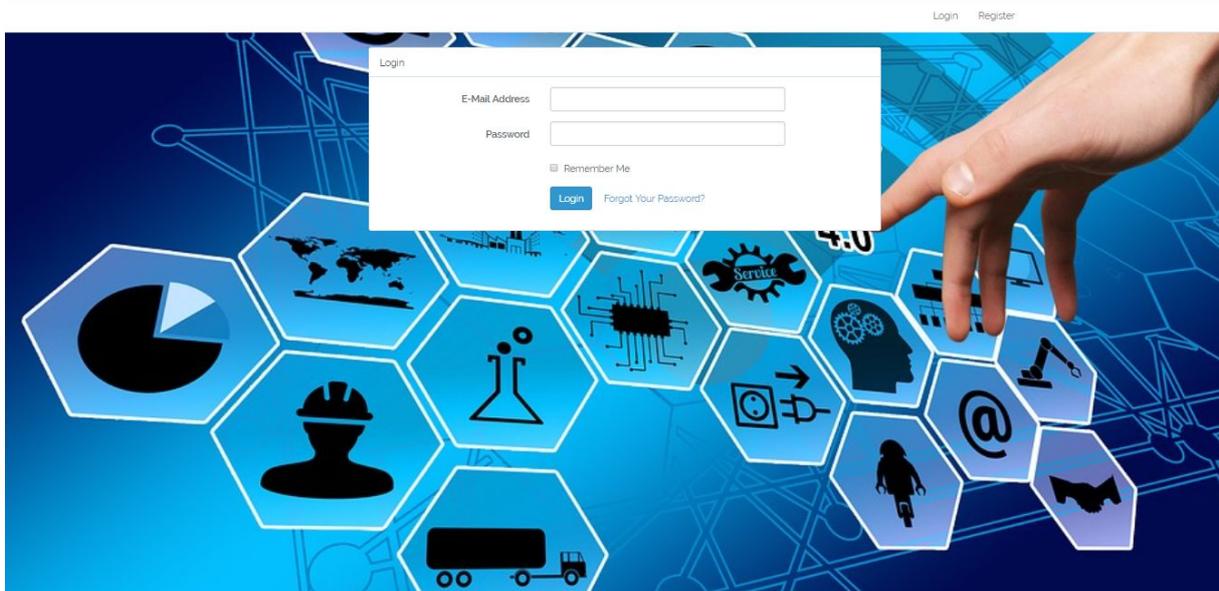


Figure 4 : Fenêtre d'authentification.

❖ L'interface d'enregistrement

Cette interface permet au utilisateur de créer un compte pour qu'être
Un administrateur de l'application.

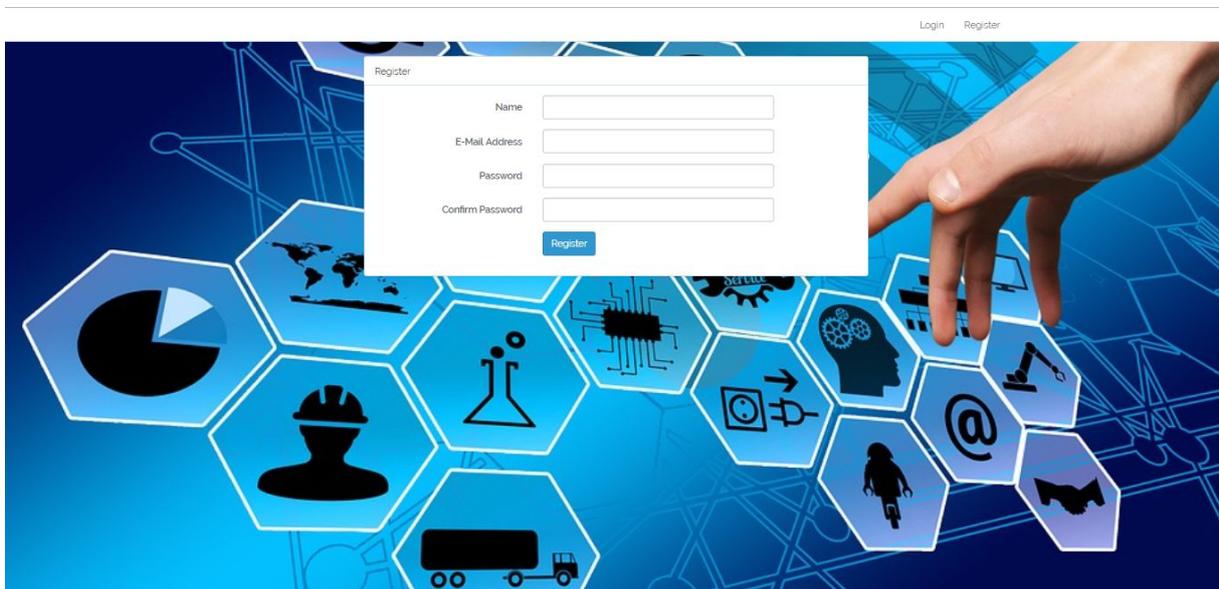


Figure 5 : Fenêtréd'enregistrement.

❖ La page d'accueil

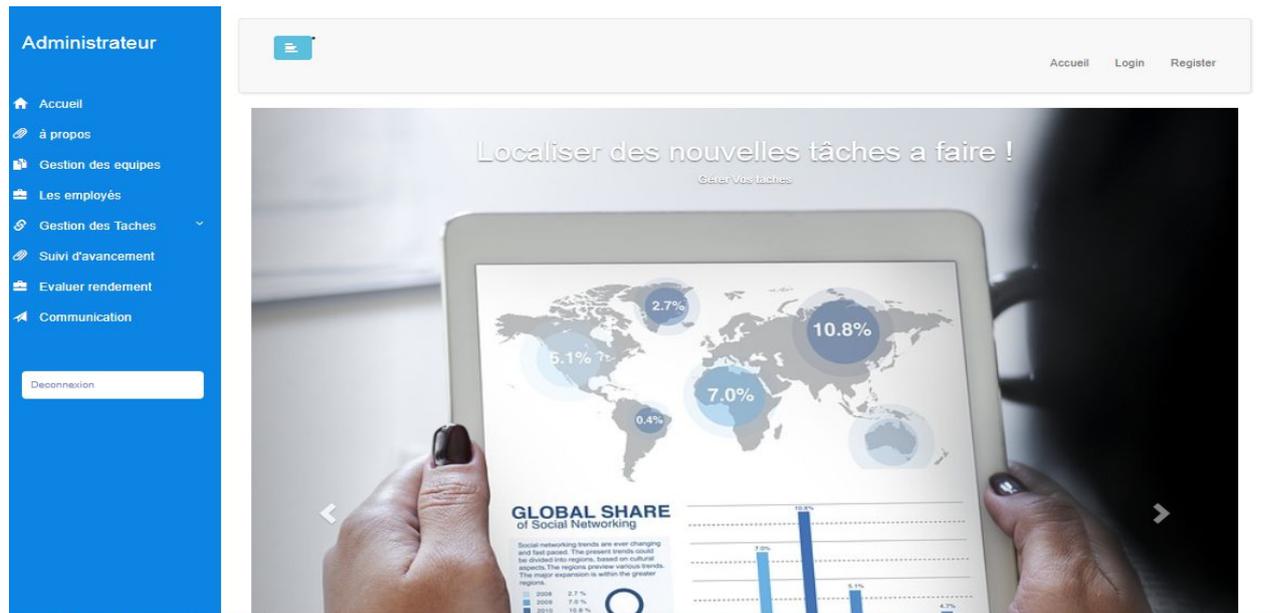


Figure 6: Fenêtrépage d'accueil.

❖ L'interface de gestion des équipes

Ces interfaces permettent au l'administrateur de gérer les déférents équipes de travail (ajouter, modifier, supprimer, afficher), pour accéder à cette interface l'administrateur doit être s'authentifé.

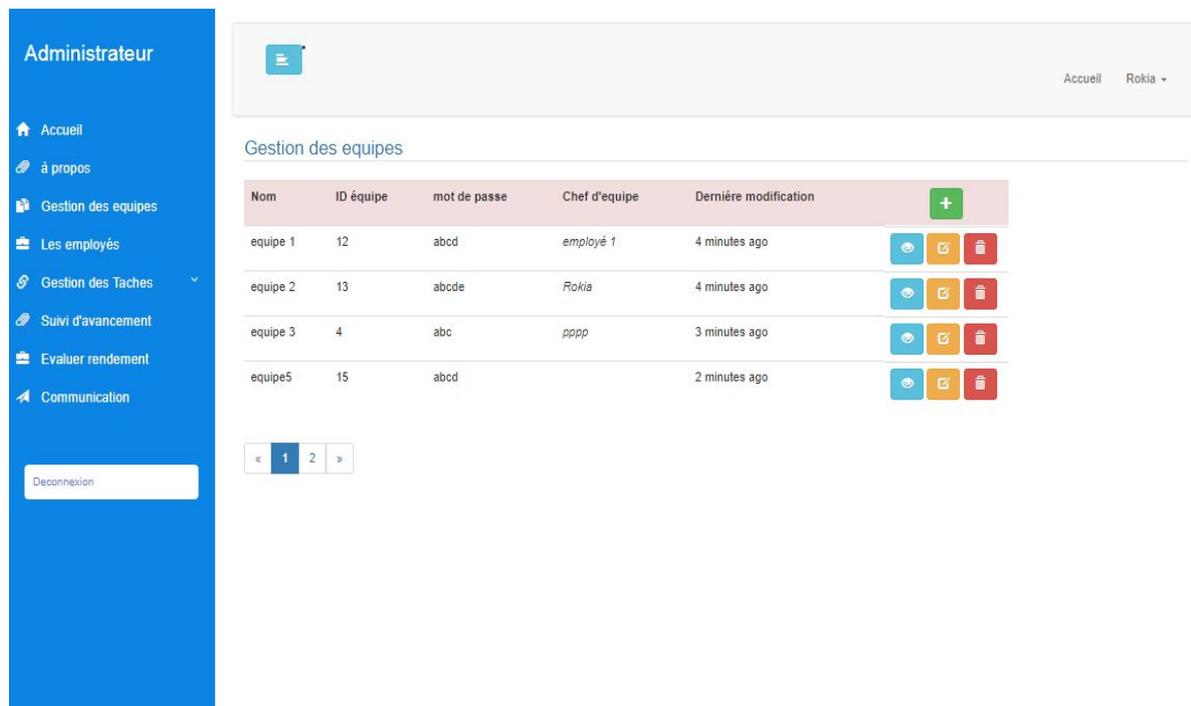


Figure 7: Fenêtré de gestion des équipes.

✓ Ajouter équipe :

The screenshot shows a web application interface for an administrator. On the left is a blue sidebar with the title 'Administrateur' and a list of menu items: Accueil, à propos, Gestion des équipes, Les employés, Gestion des Tâches (with a dropdown arrow), Suivi d'avancement, Evaluer rendement, and Communication. At the bottom of the sidebar is a 'Déconnexion' button. The main content area has a header with a hamburger menu icon and 'Accueil Rokia -'. Below the header is the title 'Ajouter une équipe'. The form contains three input fields: 'Nom de l'équipe' with the value 'nom', 'Mot de passe' with the value '...', and 'Description' with the value 'description'. At the bottom of the form are two buttons: 'Ajouter' (green) and 'Retour' (orange).

Figure 8 : Fenêtre ajouter équipe.

✓ Modifier équipe

The screenshot shows the same web application interface as Figure 8, but for editing an existing team. The main content area has the title 'Modifier les informations de l'équipe'. The form contains three input fields: 'Nom de l'équipe' with the value 'equipe 2', 'Mot de pass' with the value 'abcde', and 'Description' with the value 'my description here ____'. At the bottom of the form are two buttons: 'Modifier' (green) and 'Retour' (orange).

Figure 9 : Fenêtre modifier équipe.

✓ **Afficher équipe :**

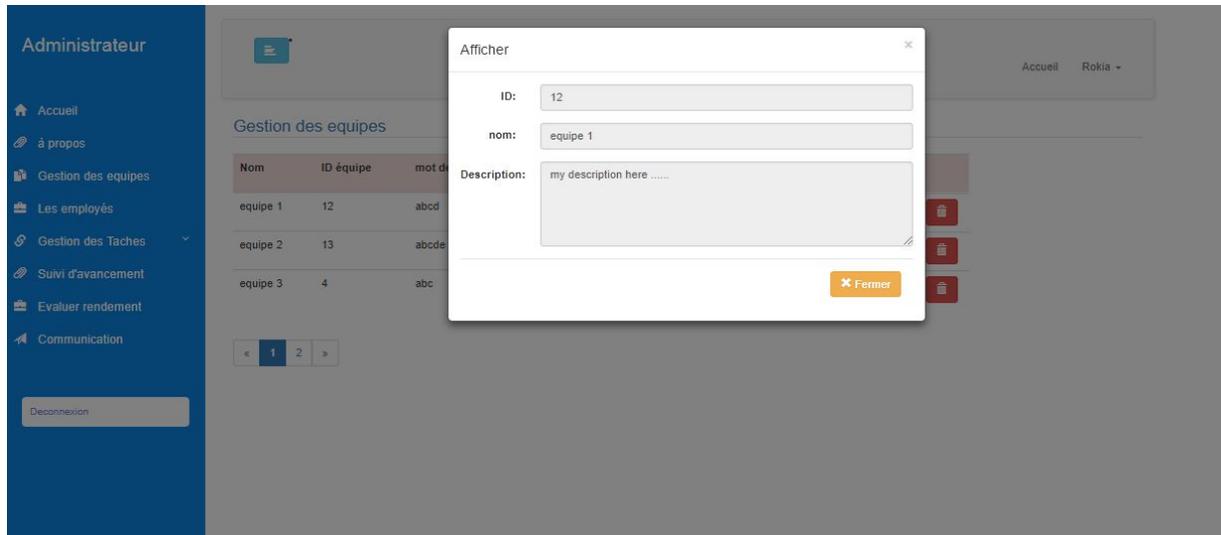


Figure 10 : Fenêtre afficher équipe.

✓ **Supprimer équipe :**

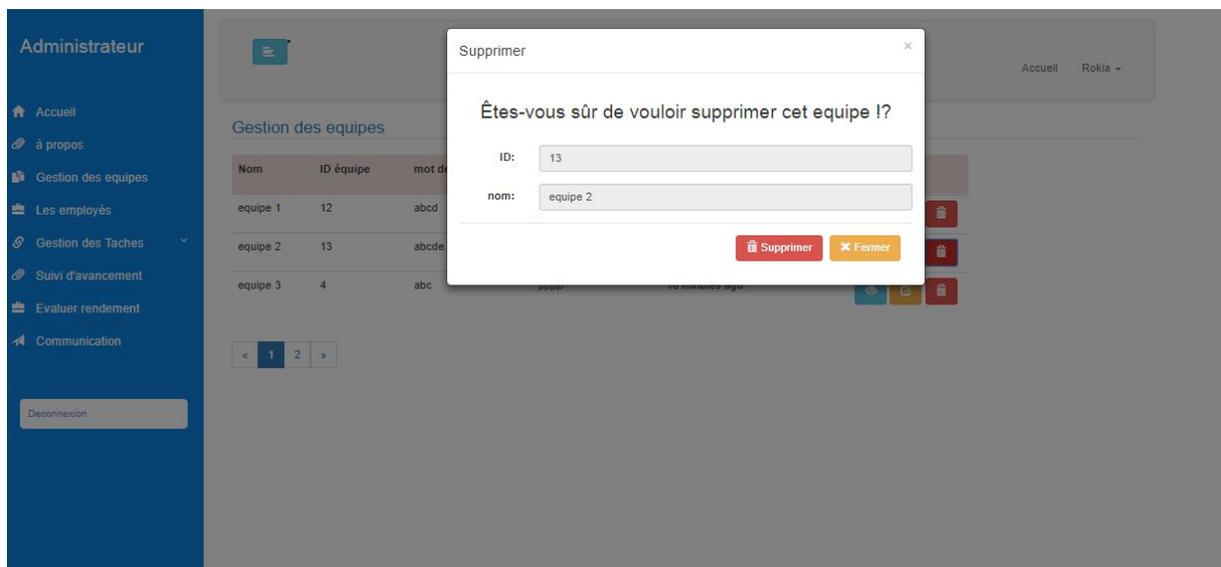


Figure 11 : Fenêtre supprimer équipe.

❖ L'interface de gestion des taches :

Cette interface permet au administrateur de gérer les déférents taches (ajouter, modifier, afficher, supprimer, affecter tache).

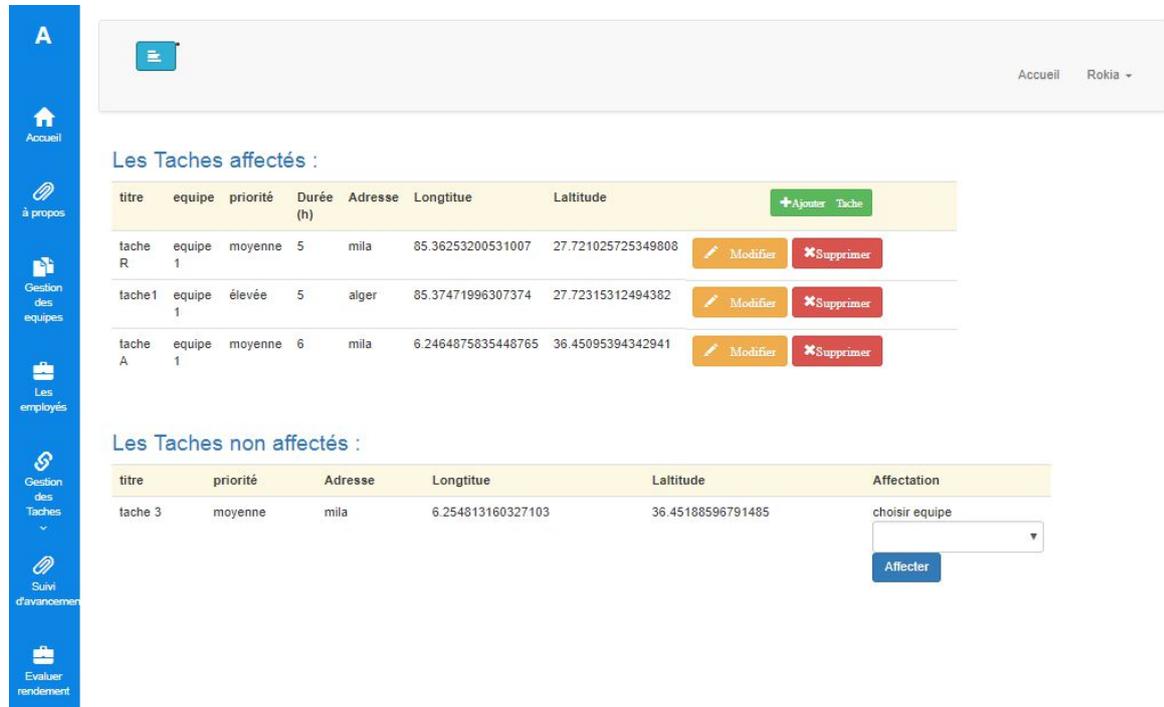


Figure 12 : Fenêtre gestion des taches.

✓ Ajouter tache

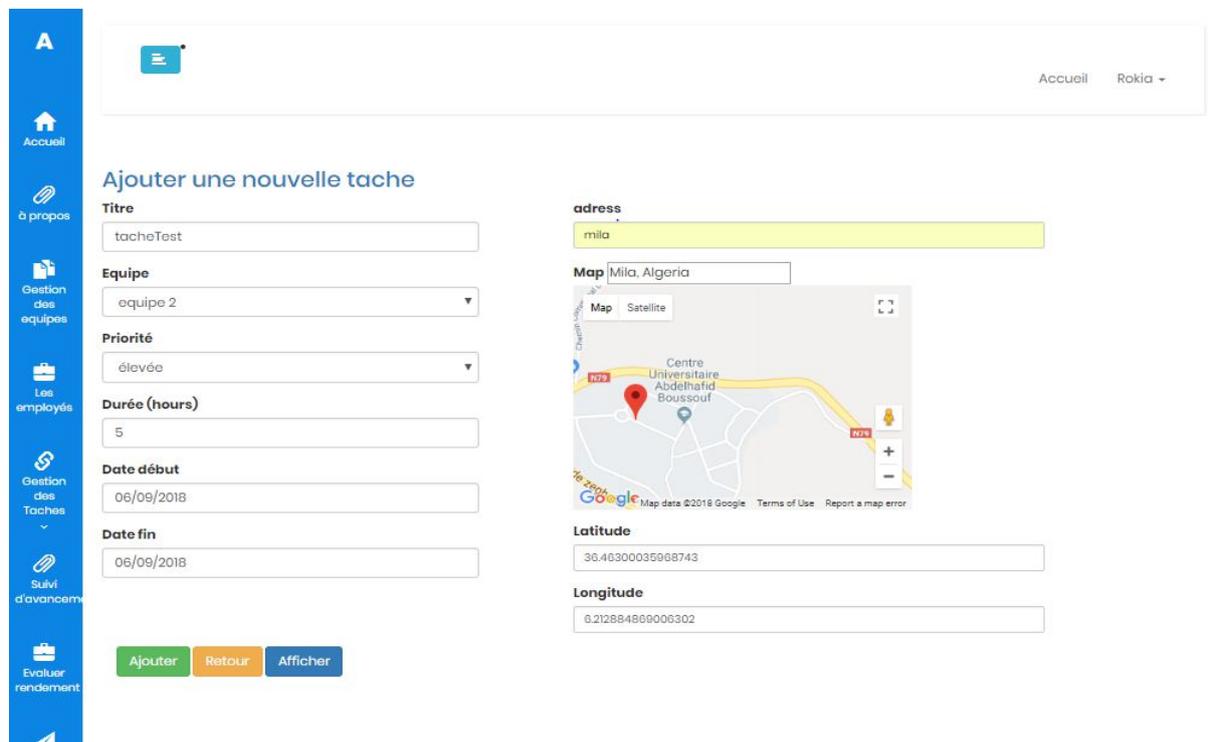


Figure 13 : Fenêtre ajouter tache.

✓ **Modifier tâche**

Accueil Rokia

Modifier les informations de la tâche

Titre

Equipe

Priorité

Durée (heures)

adresse

Map Mila, Algeria
 Map Satellite
 LOTISSEMENT BOULMERKA
 Mila
 Google Map data ©2018 Google Terms of Use Report a map error

Lat

Lng

Figure 14 : Fenêtre modifier tâche.

✓ **Affecter tâche**

Tâches non affectés :

La tâche	L'adresse	Affectation
tache R	mila	choisir equipe <input type="text" value="equipe 1"/> <input type="button" value="Affecter"/>
tache1	alger	choisir equipe <input type="text" value="equipe 1"/> <input type="button" value="Affecter"/>
tache 3	mila	choisir equipe <input type="text" value="equipe 1"/> <input type="button" value="Affecter"/>
tache A	mila	choisir equipe <input type="text" value="equipe 1"/> <input type="button" value="Affecter"/>
tache 4	mila	choisir equipe <input type="text" value="equipe 1"/> <input type="button" value="Affecter"/>

Figure 15 : Fenêtre affecter tâche.

✓ Annuler l'affectation d'une tâche

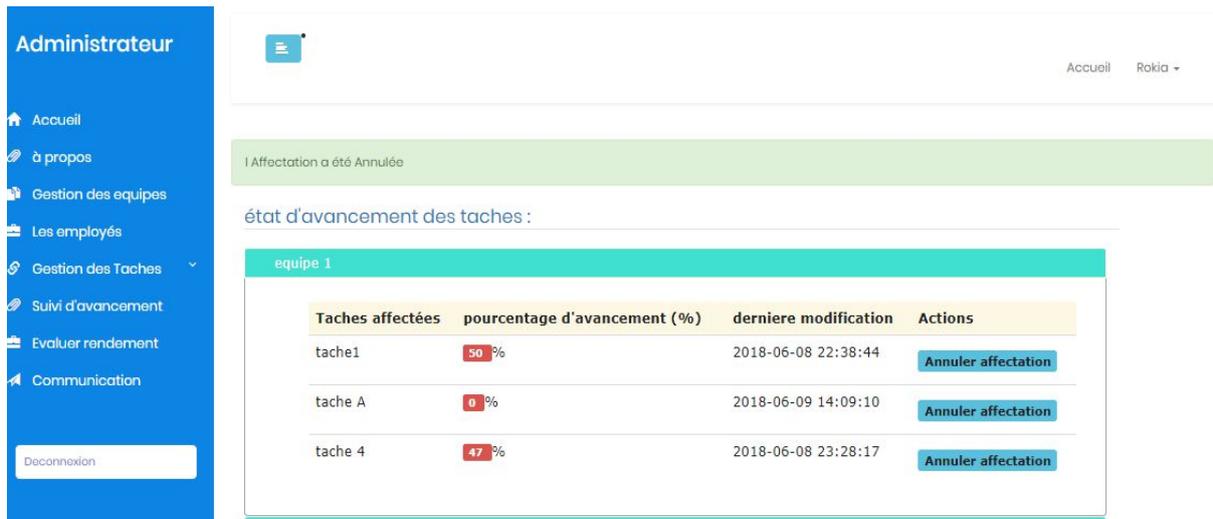


Figure 16: Fenêtre annuler l'affectation.

❖ Gestion des employés :

Cette interface permet à l'administrateur de gérer les différentes employées (ajouter, modifier, supprimer, afficher).

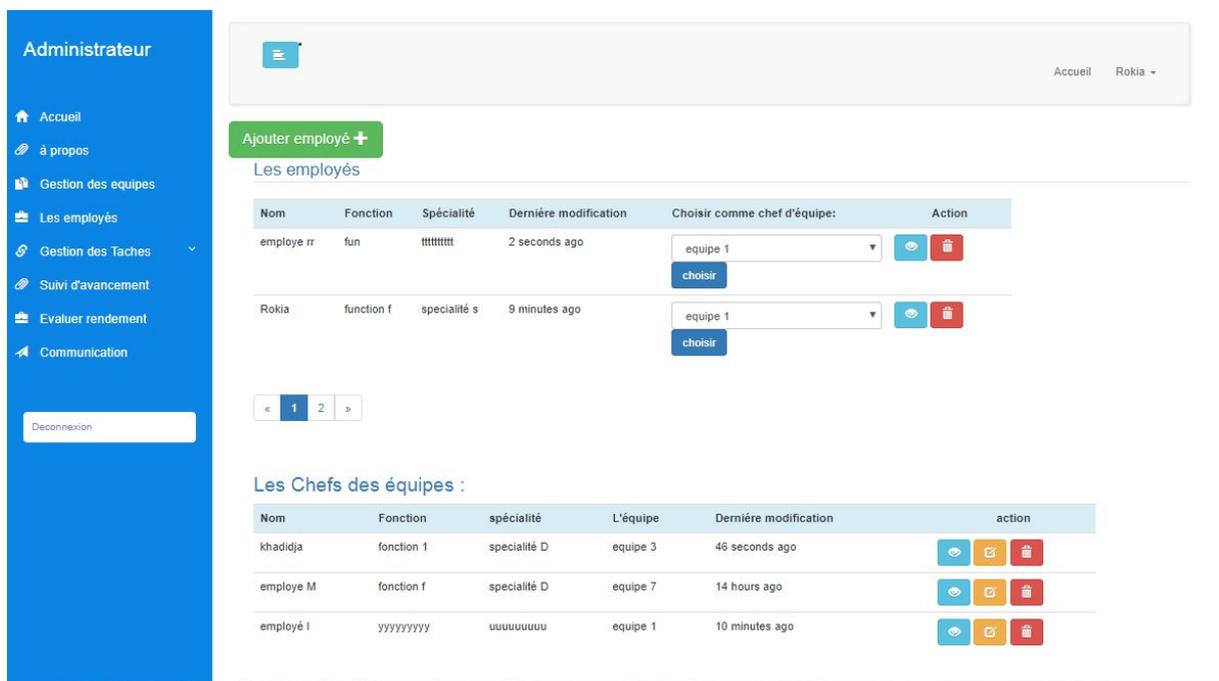


Figure 17: Fenêtre gestion des employées.

✓ Ajouter employé

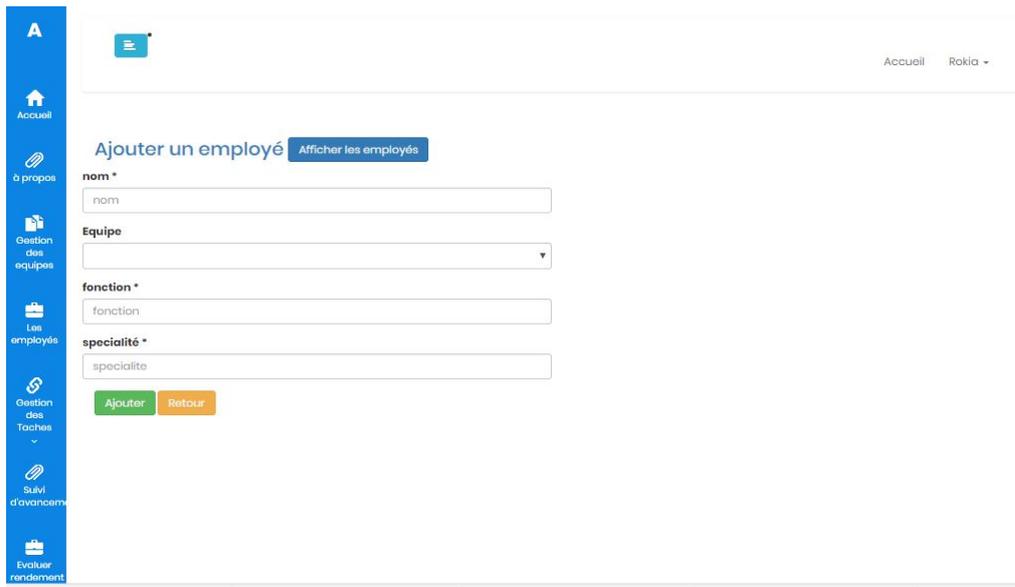


Figure 18: Fenêtre ajouter employée.

✓ Afficher employé

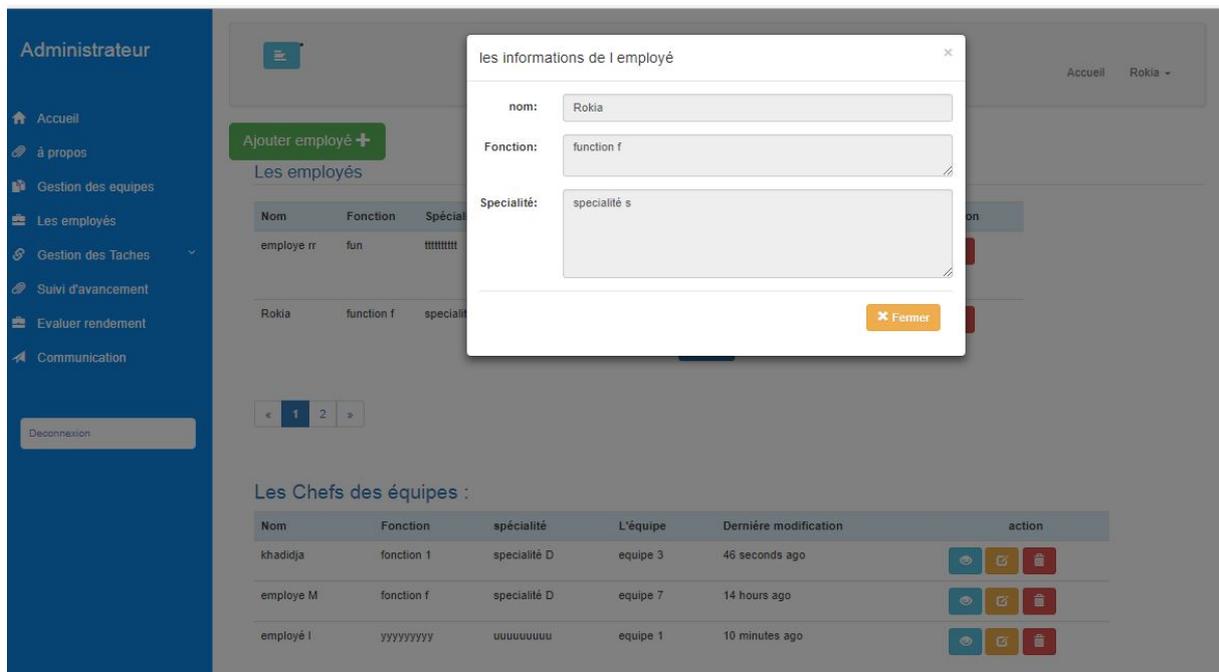


Figure 19 : Fenêtre afficher employé.

✓ **Modifier employé**

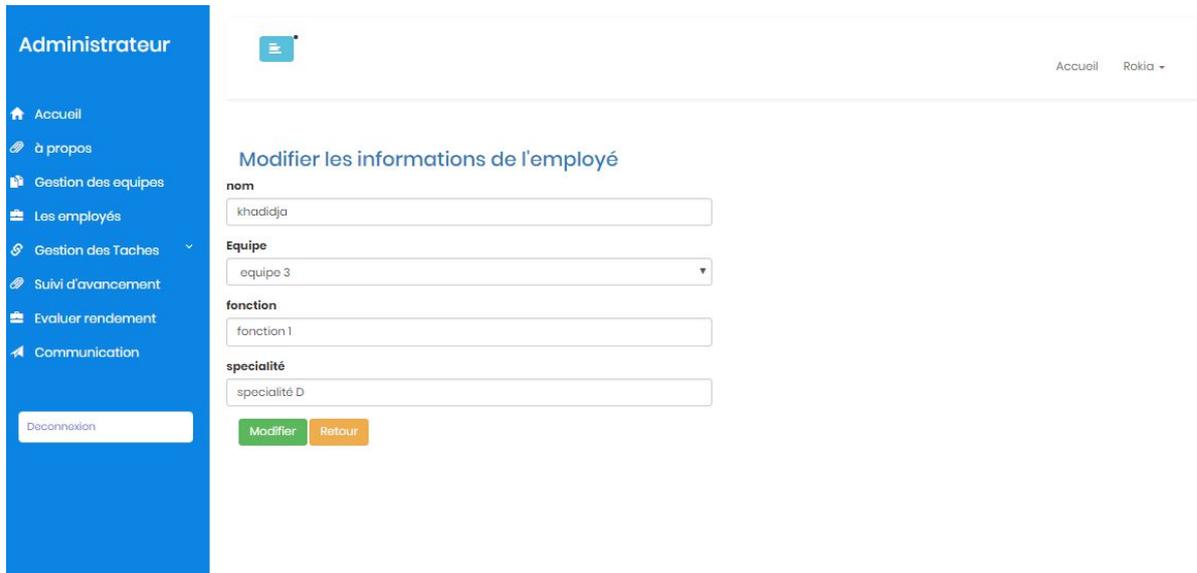


Figure 20 : Fenêtre modifier employé.

✓ **Supprimer employé**

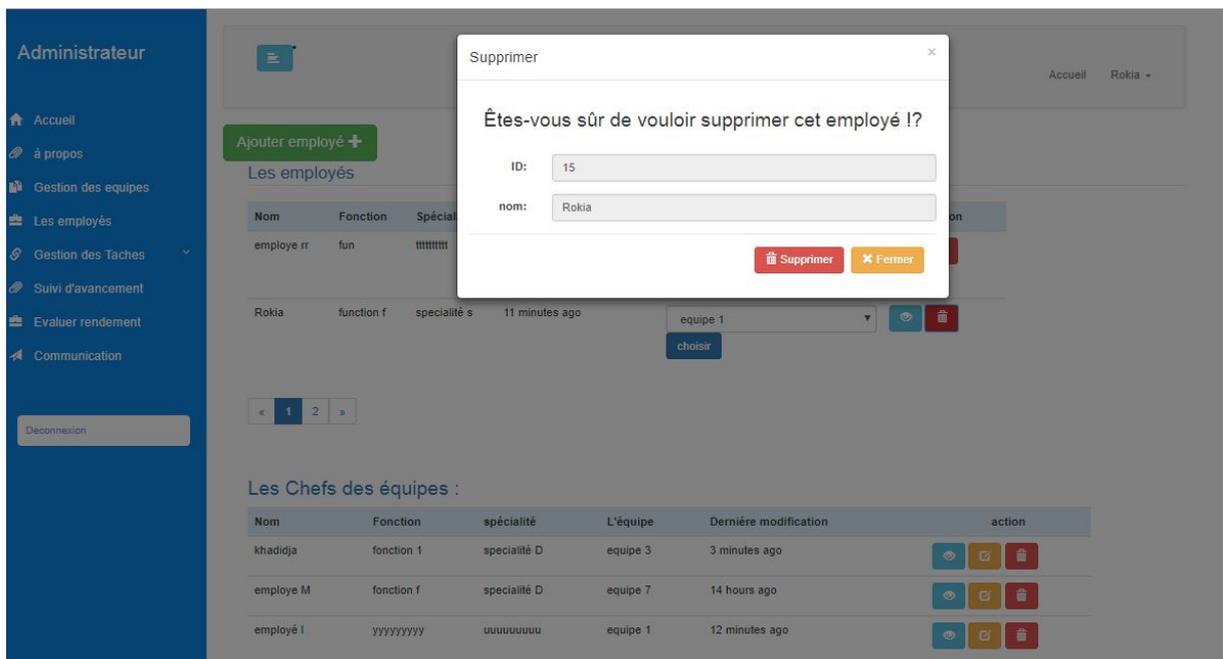


Figure 21: Fenêtre supprimer employé.

❖ Afficher l'état d'avancement des taches

Cette interface permet à l'administrateur de suivre en temps réel les états d'avancement des différentes taches réalisées par les équipes.



Figure 22: Fenêtre afficher l'état d'avancement.

❖ Communication :

Permet à l'administrateur d'envoyer des messages aux équipes et de les recevoir.

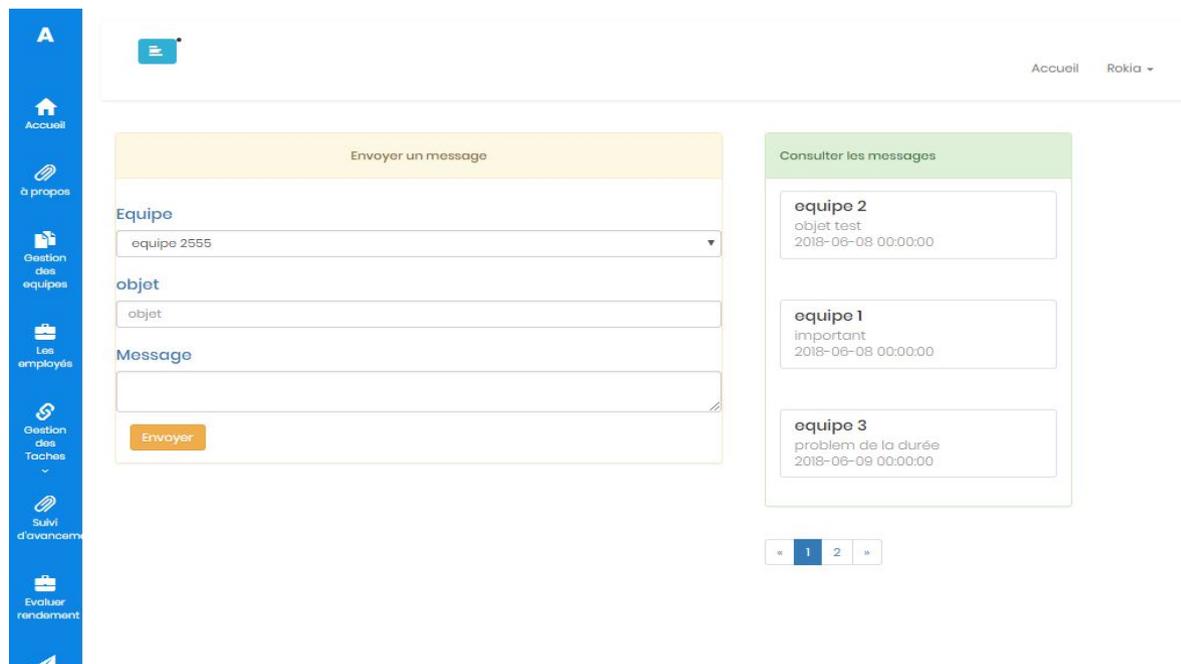


Figure 23 : Fenêtre communication.

✓ Envoyer message

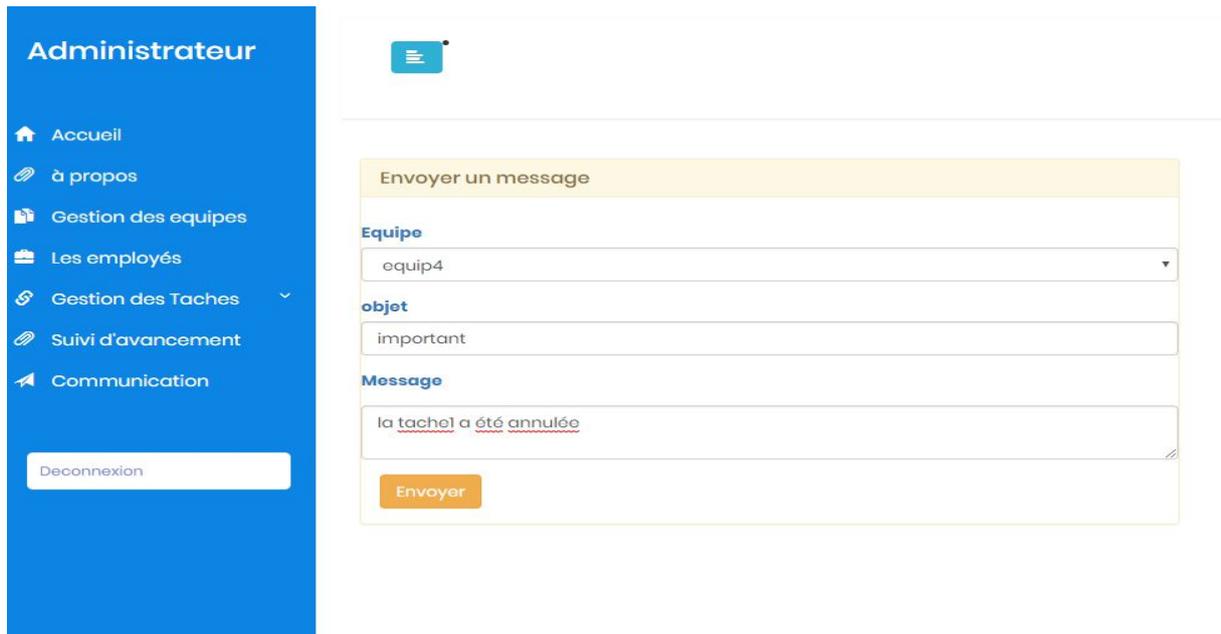


Figure 24: Fenêtre envoyer message.

✓ Consulter message :

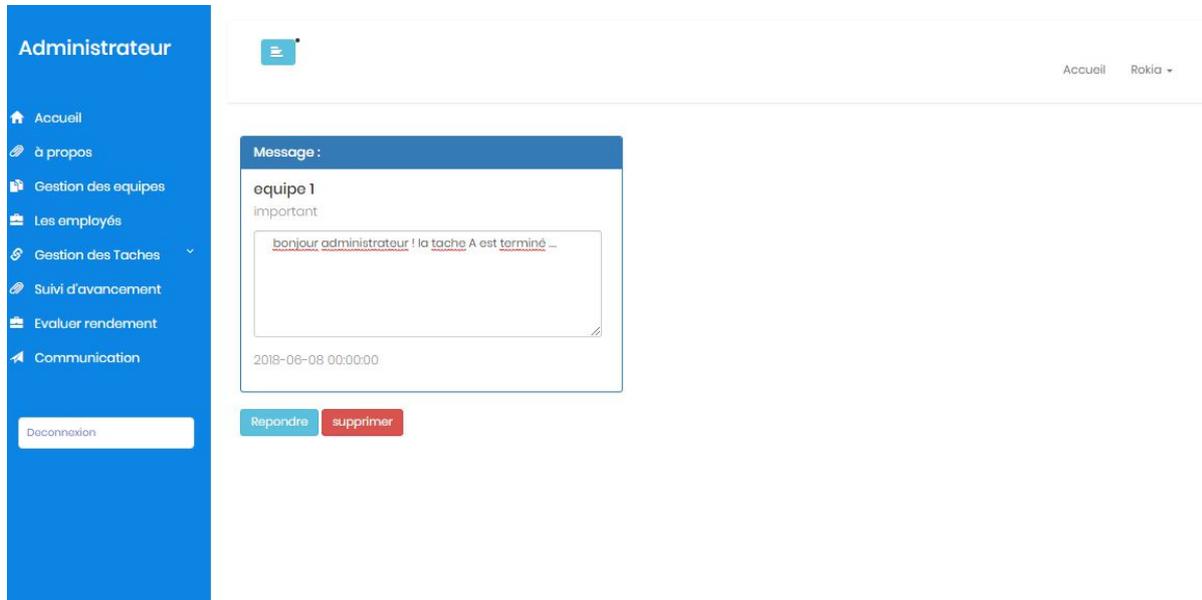


Figure 25: Fenêtre consulter message.

✓ **Supprimer message :**

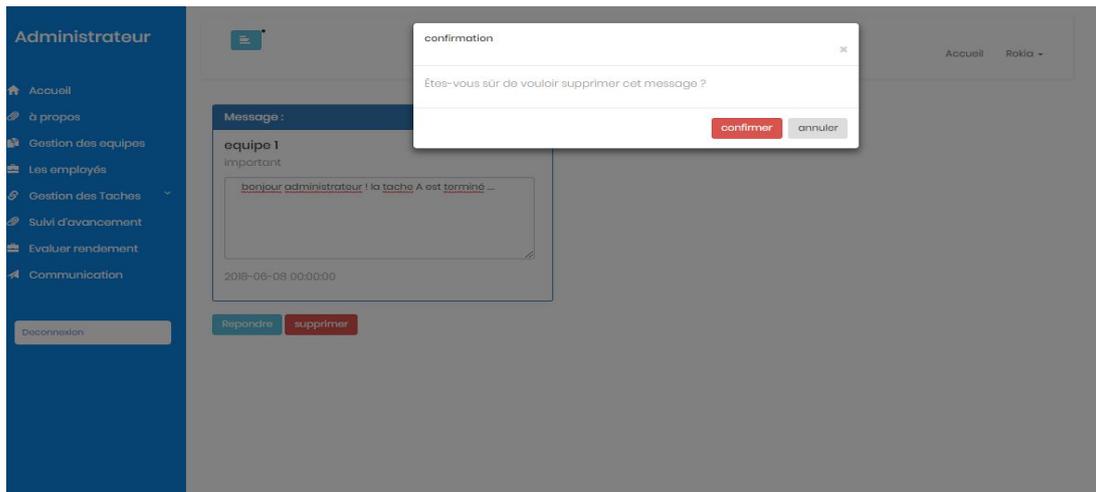


Figure 26: Fenêtre supprimer message.

✓ **Evaluer le rendement des équipes :**

L'administrateur peut évaluer le rendement de chaque équipe à partir de taches affectées ,selon les dates (temps pris et temps supposée)

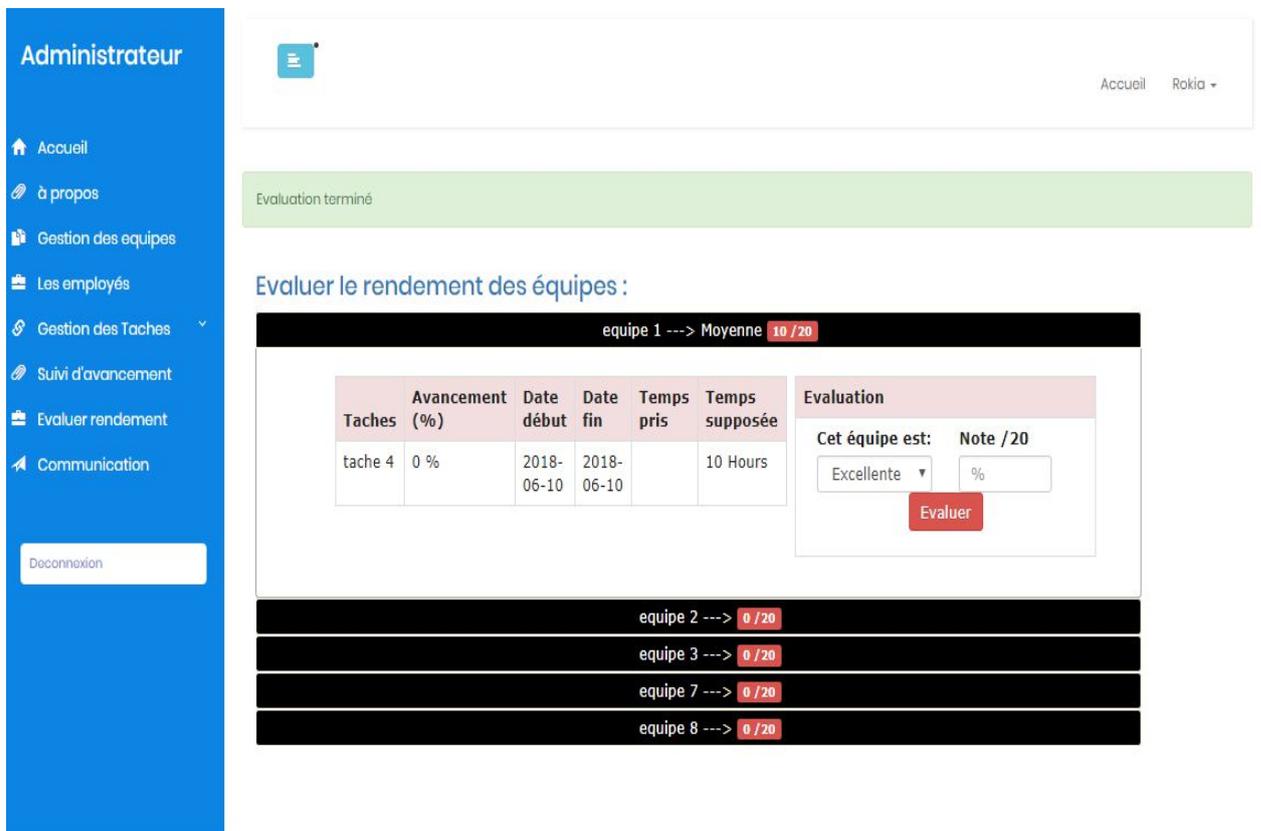


Figure 27: Fenêtre évaluer rendement des équipes.

2.2. Au niveau du Smartphone :

❖ **Interface de démarrage**



Figure 28: Fenêtre de démarrage.

❖ **Interface d'authentification**

pour que l'utilisateur de l'application peut s'authentifier il faut que l'ID et le mot de passe soit correct, l'adresse IP est utiliser pour l'accès au serveur.

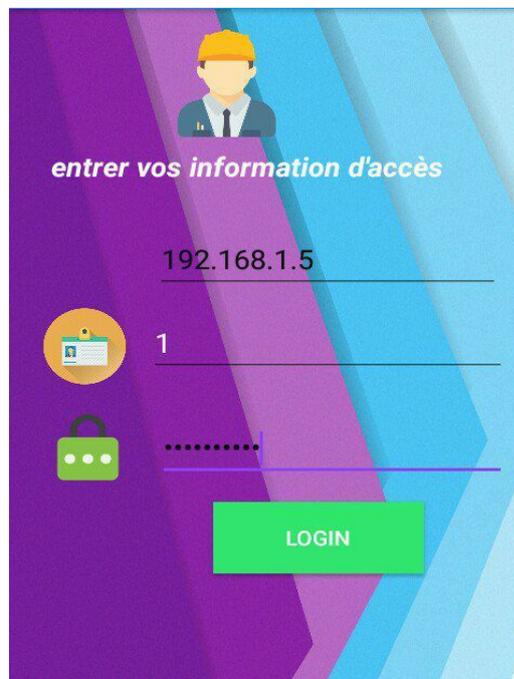


Figure 29 : Fenêtre d'authentification.

- ❖ **Page d'accueil** : Après l'authentification l'utilisateur accède à cette activité pour choisir ce qu'il veut faire soit de consulter les tâches ou les messages envoyés par l'administrateur.

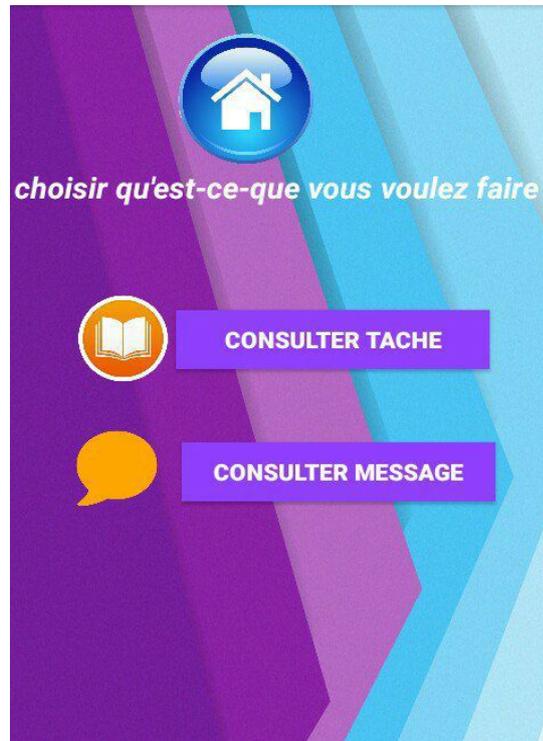


Figure 30 : Fenêtre d'accueil.

- ❖ **Consulter la liste des taches** : L'utilisateur peut avoir la liste des taches que lui affecter. Si il clique sur le bouton open il peut avoir les informations de la tâche.



Figure 31 : Fenêtre consulter la liste des taches.

❖ **afficher taches :**

l'utilisateur peut avoir les informations de la tâche, et peut évaluer l'avancement de la tâche en utilisant le bouton update pour modifier le pourcentage.

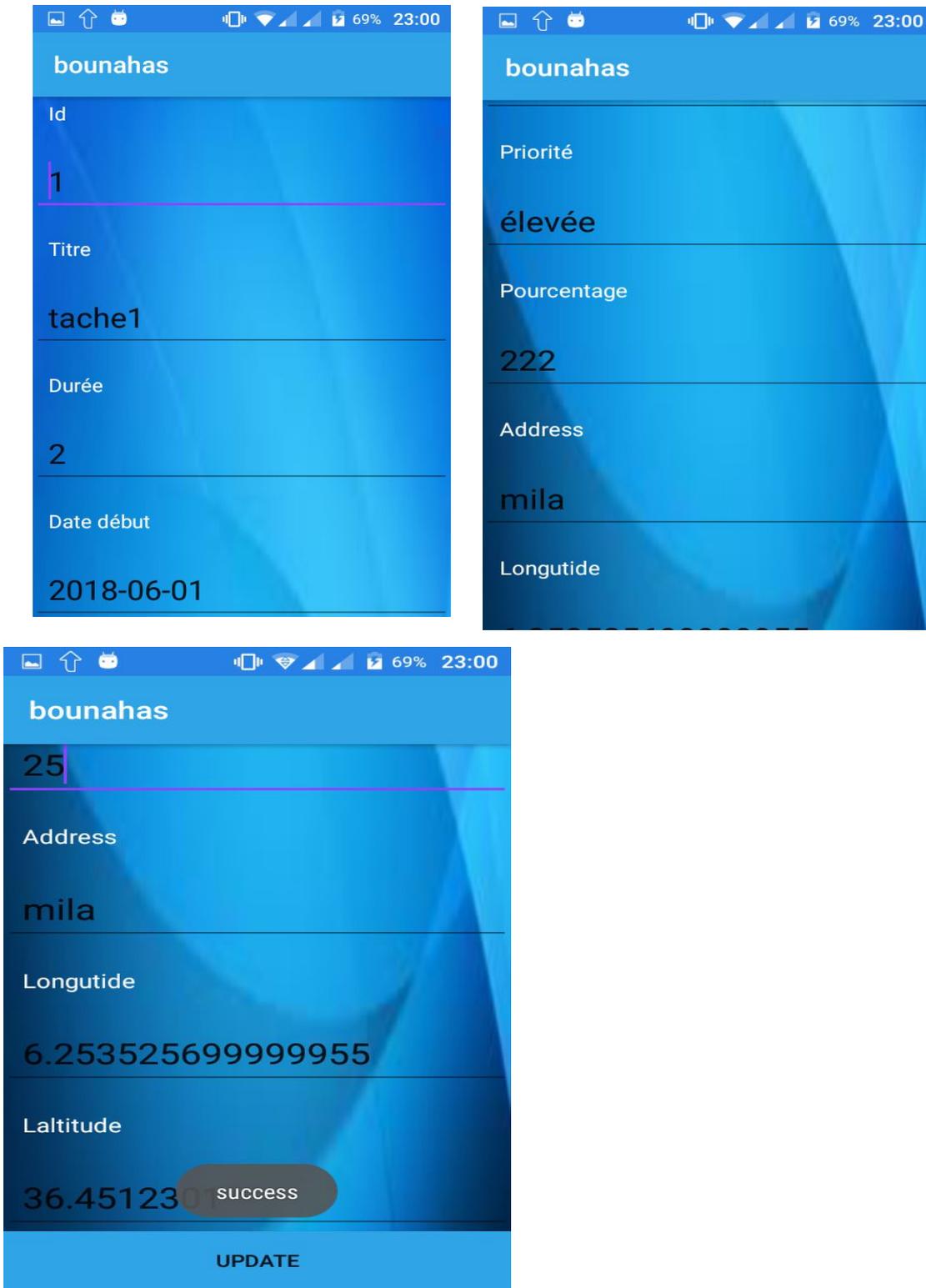


Figure 32 : Fenêtre afficher tache.

❖ **Consulter la liste des messages :**

Si l'utilisateur veut consulter la liste des messages reçu il suffit de cliquer sur le bouton consulter messages sur la figure [28]. Lorsque l'utilisateur cliquer sur le bouton consulter message il peut avoir la liste des messages.

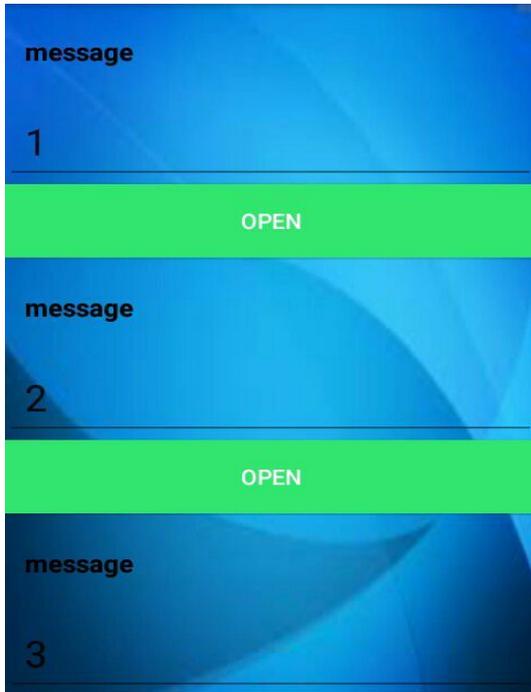


Figure 33 : consulter la liste des messages.

❖ **Afficher message:** Quand l'utilisateur clique sur le bouton Open il voit le contenu de message.

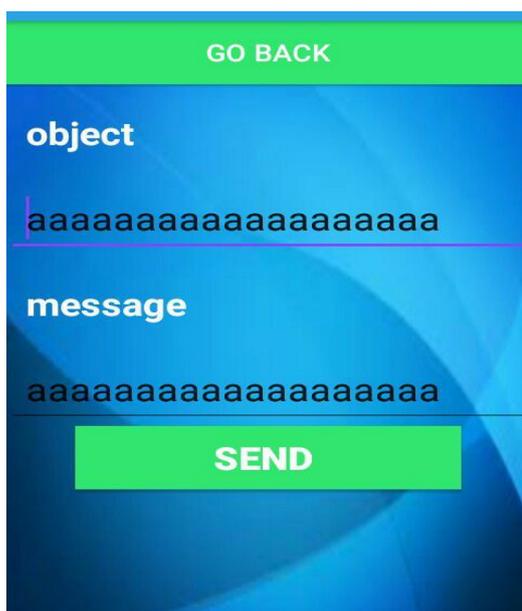


Figure 34 : afficher message.

❖ envoyer message:

il peut aussi envoyer des messages au administrateur.

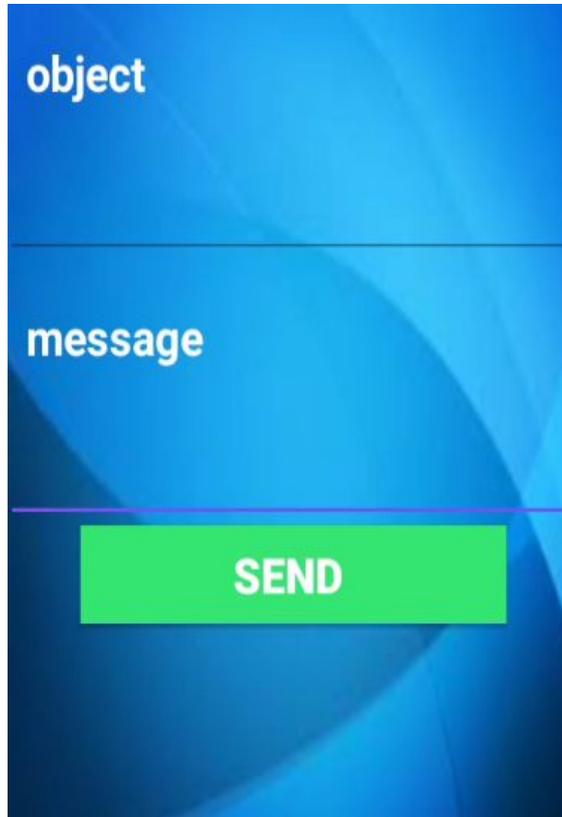


Figure 35 : envoyer message.

Conclusion

La phase d'implémentation a été concrétisée par la réalisation de l'application, en respectant la modélisation qu'on a présentée dans le chapitre précédent. Les tests effectués, durant l'étape finale, confirment que la solution proposée répond aux besoins de la gestion des équipes de travail, néanmoins notre étude couvre juste une partie des problèmes existant actuellement, avec la possibilité d'effectuer des extensions afin d'enrichir les services offerts par notre application.

Conclusion général

[CONCLUSION GENERAL]

Dans notre étude nous avons développé une application pour la gestion en temps réel des équipes de travail. Dans ce contexte, nous avons cherché à développer une application fiable, flexible et évolutive répond aux besoins des utilisateurs.

Notre application permet d'améliorer la productivité des entreprises. En effet elle est rapide, pratique et répond parfaitement aux différents besoins des entreprises dans différents domaines.

Pour la réalisation, du côté serveur, nous avons utilisé le Framework Laravel. Ce dernier permet une programmation professionnelle et sécurisée. Du côté équipement mobile, nous avons utilisé ANDROID STUDIO qui permet une compilation native très performante.

Le plus grand bénéfice tiré de cette expérience est que nous avons eu la chance d'enrichir nos connaissances dans des domaines variés comme :Laravel, Bootstrap, JQuery, Ajax, programmation Android, les RestfulApi...etc.Cependant, nous avons senti une grosse pression vu le temps qui est relativement court, donc nous n'avons pas pu réaliser tout ce qu'on a espéré faire. Par exemple l'intégration de l'intelligence artificielle pour l'affectation automatique des tâches. Nous espérons pouvoir développer plus ce travail, et convaincre les entreprises algériennes de son utilité.

[Bibliographie]

[1]	Boukhechem, Nadhir : cours développement web avancé.Mila : master stic,cour ,2017, 69p.
[2]	http://www.comprendre-internet.com/Qu-est-ce-qu-un-site-web.html
[3]	https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203265-page-web-definition/
[4]	http://www.comprendre-internet.com/Qu-est-ce-qu-un-nom-de-domaine.html
[5]	http://www.idnext.net/une-application-web-quest-ce-que-cest/
[6]	http://www.comprendre-internet.com/Qu-est-ce-que-l-adresse-d-une-page.html
[7]	http://www.besoindaide.com/ccm/initiation/peer.htm
[8]	https://openclassrooms.com/courses/simplifiez-vos-developpements-javascript-avec-jquery/premiers-pas-avec-ajax
[9]	http://www.comprendre-internet.com/Qu-est-ce-qu-une-page-web.html
[10]	http://dspace.univ-tlemcen.dz/bitstream/112/6825/1/application-client-serveur.pdf
[11]	http://ipeti.forumpro.fr/t21-definition-de-langage-java-java-script .
[12]	http://fr.wikipedia.org/wiki/Kit_de_d%C3%A9veloppement
[13]	http://fr.wikipedia.org/wiki/Java_Development_Kit
[14]	http://www.wampserver.com/
[15]	http://www.memoireonline.com/08/09/2507/m_Etude-et-realisation-du-site-web-delhotel
[16]	http://www.commentcamarche.net/contents/577-javascript-introduction-au-langagejavascript