

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ Mouloud MAMMERRI DE TIZI-OUZOU.  
FACULTÉ Génie électrique et informatique.  
DÉPARTEMENT Informatique.



## MEMOIRE DE FIN D'ETUDES

Pour l'obtention du  
« Master professionnel en ingénierie des systèmes d'information »  
**Thème :**

Conception et développement d'une Application Mobile pour le suivi du personnel dans un chantier.

### Réaliser par :

- Saoudi Massinissa.
- Naili Massinissa

### Encadré par :

- Mr Ramdani  
Mohamed.

### Soutenu devant le jury :

- Présidente : Mme. Cherifi
- Examineur : Mr. Sadi

## **Abstract:**

In this humble work we present to you the Procedure that we followed in our graduation project for obtaining the Professional Master diploma, which has for theme "Design and realization of a mobile application dedicated to the follow-up of personnel in a construction company", this work begins with an environment study where we discover the construction sector, its needs and the problems faced in this field, then moves on to the most used technology which has proven its worth, its efficiency and that we could use it to solve these issues, then we will talk about the type of ERP application (Enterprise Resource Planning) that we evaluated as perfect for our case study, So that we then enter into The Design phase where we will illustrate the functionalities that our system must have and the architecture of its different components, and based on this design we begin the development phase where we will demonstrate the qualities of the final product including some interfaces.

Keywords: Personnel follow-up, construction company, ERP, NodeJS, React Native, Cross-Platform

## **Résumé :**

Dans cet humble travail nous présentant la procédure suivi dans notre projet de fin d'étude pour l'obtention du diplôme Master Professionnel, qui a pour thème « Conception et réalisation d'une application mobile dédié au suivi de personnels dans une entreprise de secteur Bâtiments et Travaux Publiques », ce travail commence par une études de l'art ou on part à la découverte du secteur BTP, ses besoins et les problématiques posés dans ce domaine, puis en passe à la technologie la plus utilisé et qui a prouvé son efficacité et qu'on pourrais s'en servir pour résoudre ces problèmes, Ou on parlera du type d'application ERP (Entreprise Ressource Planning) qu'on a évalué parfait pour notre cas d'étude, Pour qu'ensuite, on entre dans La phase Conception ou on va illustrer les fonctionnalités que notre système doit posséder et l'architecture des différents composants, et on se basant sur cette conception on commence la phase réalisation ou on démontrera les qualités du produit final en incluant quelques interfaces.

Mots Clés : Suivi de personnels, Secteur BTP, ERP, NodeJS, React Native, Cross-Plateforme

## ***Remerciements***

*On adresse nos vifs remerciements :*

*A Allah le tout puissant de nous avoir donné la santé, la volonté, le courage et de nous avoir fourni, la force pour achever ce travail.*

*Nous ne saurons oublier de remercier nos parents pour leur contribution, leur soutien et leur patience, nos proches, nos amis et toutes les personnes qui nous ont aidé par leur soutien permanent dans nos études ou en dehors.*

*A notre encadreur Ramdani Mohamed pour son encadrement, son soutien sans faille et sa disponibilité. Ses conseils, ses suggestions de lecture, ses commentaires, ses corrections et ses qualités scientifiques ont été très précieux pour mener à bien ce travail.*

## *Dédicaces*

*Je dédie ce modeste travail*

*A mes chers parents, que dieu les protèges.*

*A toute ma famille.*

*A tous mes amis.*

*A toutes les personnes qui me sont chers,*

*Naili M.*

*Je dédie ce modeste travail*

*A ma mère, que dieu la protège.*

*A tous mes amis.*

*A toutes les personnes qui me sont chers, elles se  
reconnaîtront.*

*Saoudi M.*

## Sommaire

Table de figures.....	7
Introduction Générale .....	9
Chapitre 1 : Le Secteur BTP .....	10
Introduction.....	10
1.1Le secteur BTP.....	10
1.2Les spécificités métiers de secteur BTP.....	10
1.2.1Les métiers qui recrutent niveau terminal et moins .....	10
1.2.2 Les métiers qui recrutent niveau bac .....	11
1.2.3 Les métiers qui recrutent niveau bac+2/bac+3 .....	12
1.2.4 Les métiers qui recrutent niveau bac+5 .....	12
1.3 Les besoins de service ressources humaines dans une entreprise de secteur BTP .....	13
1.3.1 Gestion du recrutement .....	13
1.3.2 Gestion de Carrière .....	13
1.3.3 Gestion de paie.....	14
1.3.4 Gestion de temps .....	14
1.4 Organisme d'accueil .....	15
1.4.1 Entreprise GTS développement .....	15
1.4.2 Organigramme de l'entreprise GTS développement .....	15
1.4.3 Les activités de GTS développement.....	16
1.4.4 La problématique .....	16
Conclusion .....	17
Chapitre 2 : Les ERP.....	18
Introduction.....	18
2.1 Les progiciels ERP.....	18

2.2 Principes de fonctionnement des ERP .....	18
2.2.1 Principe Fondateur .....	18
2.2.2 Le moteur workflow .....	19
2.3 Utilisateurs des ERP .....	19
2.4 Avantages de la mise en place d'un ERP .....	19
2.5 Architecture des progiciels ERP .....	20
Conclusion .....	20
Chapitre 3 : Analyse et Conception .....	21
Introduction.....	21
3.1 Choix de cycle de développement .....	21
3.1.1 Cycle de développement en V .....	21
3.1.2 Cycle de développement en cascade.....	24
3.1.3 Déférence entre le cycle de développement en V et le cycle de développement en cascade .....	25
3.1.4 Notre Choix de cycle de développement .....	25
3.2 Outils de conception .....	26
3.2.1 Langage de modélisation UML .....	26
3.2.2 Les 13 diagrammes de l'UML .....	27
3.3 Les Besoins fonctionnels .....	27
3.4 Les Besoins non fonctionnels .....	28
3.5 Les éléments qui composent l'application.....	29
3.5.1 Définition d'un acteur: .....	29
3.5.2 Le chef de chantier .....	30
3.5.3 Le chef d'équipe .....	30
3.5.4 Le service ressources humaines .....	30

3.5.5 Directeur d'entreprise, directeur technique et le chef de projet.....	30
3.6 Les fonctionnalités que l'application doit satisfaire .....	32
3.6.1 Les fonctionnalités de directeur de l'entreprise, directeur technique et le chef de projet .....	32
3.6.2 Les fonctionnalités de services ressources humaines .....	33
3.6.3 Les fonctionnalités de chef de chantier.....	34
3.6.4 Les fonctionnalités de chef d'équipe .....	34
3.7 Les interactions de l'application avec l'utilisateur .....	35
3.7.1 Authentification .....	35
3.7.2 Ajouter un employé.....	36
3.7.3 Ajouter un poste.....	36
3.7.4 Ajouter un congé.....	37
3.7.5 Ajouter une tâche .....	37
3.7.6 Modifier un employé.....	38
3.7.7 Supprimer un employé.....	39
3.7.8 Confirmer ou annuler un congé .....	40
3.7.9 Marquer une absence .....	41
3.7.10 Evaluer un employé .....	42
3.8 Diagramme de classes.....	43
Conclusion .....	44
Chapitre 4 : Réalisation.....	45
Introduction.....	45
4.1 Outils de développement.....	45
4.1.1 Environnement NodeJS : .....	45
4.1.2 Les Frameworks JavaScript coté client.....	49

4.1.3 Outils Temps Réel :	53
4.1.4 Les systèmes de gestion de base de données	56
4.1.5 Visual Studio Code	60
4.1.6 Firebase	60
4.2 Fonctionnalités de l'application	61
4.2.1 Authentification	61
4.2.2 Consultation de la liste des employés	62
4.2.3 Consultation de la liste des postes	63
4.2.4 Consultation de la liste des tâches	64
4.2.5 Consultation de la liste des congés	65
4.2.6 Ajout d'un employé et ajout d'un poste	66
4.2.7 Modifier ou supprimer un employé	67
4.2.8 Ajouter une tâche	68
4.2.9 Ajouter une absence ou une évaluation à un employé	69
4.2.10 Ajouter un congé	70
Conclusion	70
Conclusion générale	71
Bibliographie et Webographie	72

## Table de figures

Figure 1 : Organigramme de l'entreprise GTS.....	15
Figure 2 : Architecture logiciel .....	20
Figure 3 : Cycle de Développement en V.....	22
Figure 4 : Cycle de Développement en Cascade .....	24
Figure 6 : Diagramme de contexte.....	31
Figure 7 : Diagramme de cas d'utilisation « directeur Technique », « Directeur », « chef projet » .....	32
Figure 8 : Diagramme de cas d'utilisation « SRH ».....	33
Figure 9 : Diagramme de cas d'utilisation « chef de chantier ».....	34
Figure 10 : Diagramme de cas d'utilisation « chef d'équipe ».....	34
Figure 11 : Diagramme de séquence « Authentification » .....	35
Figure 12 : Diagramme de séquence « Ajouter un Employé » .....	36
Figure 13 : Diagramme de séquence « Ajouter un Poste ».....	36
Figure 14 : Diagramme de séquence « Ajouter un Congé ».....	37
Figure 15 : Diagramme de séquence « Ajouter une Tâche ».....	37
Figure 16 : Diagramme de séquence « Modifier un Employé » .....	38
Figure 17 : Diagramme de séquence « Supprimer un Employé » .....	39
Figure 18: Diagramme de séquence « Confirmer ou Annuler un Congé » .....	40
Figure 19 : Diagramme de séquence « Marquer une Absence » .....	41
Figure 20 : Diagramme de séquence « Evaluation d'un Employé » .....	42
Figure 21 : Diagramme de Classes .....	43
Figure 22 : Interface « Authentification ».....	61
Figure 23 : Interfaces « Liste des Employés » et « Détails employé ».....	62
Figure 24 : Interfaces « Liste des Postes » et « Détails poste ».....	63
Figure 25 : Interfaces « Liste des tâches » et « Détails Tâche ».....	64
Figure 26 : Interfaces « Liste des Congés » et « Détails Congé » .....	65
Figure 27 : Interfaces « Ajouter un Employé » et « Ajouter un Poste ».....	66
Figure 28 : Interfaces « Modifier un Employé » et « Supprimer un Employé » .....	67
Figure 29 : Interface « Ajouter une Tâche ».....	68

Figure 30 : Interfaces « Marquer Absence » et « Evaluer un Employé ».....	69
Figure 31 : Interface « Ajouter un Congé » .....	70

# Introduction Générale

Dans chaque entreprise on reconnaît cinq fonctions principales, la fonction technique, la fonction humaine ou personnel, la fonction commerciale, la fonction recherche et développement et enfin la fonction financière. De toutes ces fonctions, celle humaine revêt une importance capitale dans le fonctionnement d'une entreprise.

Le problème de gestion de cette entité dans l'entreprise hôte pour les raisons de gestion des contrats, de gestion des effectifs et gestion des absences. C'est la raison pour laquelle nous avons été amenés à concevoir une application mobile dont l'objectif premier est d'optimiser la gestion du personnel. Cela peut permettre, nous énumérons :

- Meilleure planification des départs en congés annuels et suivi des congés de maladie et de maternité.
- Rapidité et efficacité de l'insertion des nouveaux employés dans le SI.
- Assurer une affectation des tâches et création de nouvelles équipes en temps réel.

Pour réaliser notre projet, nous avons utilisé un ensemble de notions relatives aux méthodes de conception des systèmes d'information et aux méthodes de conduite des projets. Ce document détaillera donc les travaux effectués tout au long de la réalisation de ce projet. Le travail réalisé a été divisé en quatre chapitres :

\_\_Dans le premier chapitre, on définit le secteur BTP et ces besoins, on présentera l'entreprise GTS développement qui est une entreprise active dans le secteur BTP.

\_\_Ensuite au deuxième chapitre, on parlera de l'ERP qui est un type de logiciels qui sert à la gestion des ressources d'une entreprise. On détaillera son principe de fonctionnement, ses utilisateurs et ses avantages.

\_\_Puis, au troisième chapitre, on s'introduit dans la phase analyse et conception, ou on définit les besoins et les fonctionnalités que notre produit final doit satisfaire, son architecture et les choix des outils de développement qu'on a pris pour la prochaine phase.

\_\_Enfin, dans le quatrième chapitre, on finira par une illustration de travail réaliser, en montrant quelques captures d'écran prise de l'application.

# Chapitre 1 : Le Secteur BTP

## Introduction

Dans ce chapitre nous partons à la découverte du secteur BTP, qui est considéré comme l'un des secteurs les plus actives en Algérie, il bénéficie d'un grand budget étatique et de gros investissements privés et c'est un secteur qui recrute en divers postes de travail en fonction des qualifications des employés et les besoins de l'entreprise.

Pour Bien Comprendre les besoins de ce secteur nous avons saisi une opportunité offerte par une entreprise fonctionnaire dans ce domaine, ou on a eu la chance d'avoir une vue sur terrain des besoins et les problématiques de cette entreprise et surtout la gestion d'un large effectifs qui est une tâche difficile mais cruciale afin d'affecter des tâches aux équipes et planifié les futurs projets en se basant sur ce qu'il y'a comme mains d'œuvres et leurs disponibilités.

## 1.1 Le secteur BTP

Le bâtiment et travaux publics, plus communément appelé par son abréviation BTP, est un secteur économique qui rassemble l'ensemble des activités et des corps de métier intervenant dans la conception et la construction des bâtiments publics et privés et des infrastructures telles que les routes ou les canalisations.

## 1.2 Les spécificités métiers de secteur BTP

### 1.2.1 Les métiers qui recrutent niveau terminal et moins

#### **Manœuvre :**

C'est un poste qui ne nécessite pas de qualifications, il doit effectuer des taches polyvalentes ou précises, et il doit porter soutien aux ouvriers qualifiés en attendant sa promotion vers un poste supérieur.

**Maçon :**

Le maçon est le premier à intervenir sur un chantier de construction. Il a pour mission de créer les fondations du futur bâtiment, en fonction des instructions du géomètre. C'est lui qui monte les murs, installe les différentes cloisons, pose les panneaux préfabriqués, les fenêtres, etc.

**Couvreur :**

Le couvreur est le spécialiste de la construction et de la réfection des toitures. Ce professionnel prépare les chantiers et pose les matériaux nécessaires à la construction des toits. Lorsque cela est nécessaire, il s'occupe également de la réparation et de l'entretien des toits.

**Peintre en bâtiment :**

Le peintre en bâtiment est le dernier professionnel qui intervient sur un chantier de construction. Son objectif : peindre les murs et les plafonds. Cet employé du bâtiment a la touche finale du chantier de construction : il décape, colmate, ponce, et pour finir, peint.

**Chef d'équipe BTP :**

Dans le BTP, le chef d'équipe est sous la responsabilité du chef de chantier. Il doit être un ouvrier expérimenté, il doit montrer une certaine sagesse, esprit de Leadership et de responsabilité.

## **1.2.2 Les métiers qui recrutent niveau bac**

**Électricien du bâtiment :**

L'électricien du bâtiment installe les interrupteurs, les prises, la vidéosurveillance, les systèmes d'alarme et tout ce qui est en lien avec le système électrique d'un bâtiment. Il travaille en lien étroit avec les maçons et les plaquistes.

### **1.2.3 Les métiers qui recrutent niveau bac+2/bac+3**

#### **Chef de chantier :**

Le chef de chantier a pour mission principale de diriger les équipes et de coordonner les travaux sur un chantier. C'est lui qui assure le bon déroulement des travaux. Il a également de nombreuses tâches administratives telles que la gestion du personnel, ou encore, le respect des coûts.

#### **Conducteur de travaux :**

Le conducteur de travaux a pour mission de planifier et d'organiser les aménagements des chantiers. Il est le responsable de la gestion des moyens techniques, humaines et financiers lors de la réalisation d'un chantier de construction. Il fait en sorte de respecter les délais et les règles de la sécurité tout au long du déroulement d'un projet.

### **1.2.4 Les métiers qui recrutent niveau bac+5**

#### **Chef de projet BIM (Building Information Modeling) :**

Le chef de projet BIM est l'ingénieur qui met en place la stratégie BIM de l'entreprise. Il récupère les maquettes de ses collaborateurs et prépare les différentes réunions de coordination. Son objectif est de coordonner les différentes interventions sur les maquettes.

#### **Ingénieur BTP :**

L'ingénieur d'études ou ingénieur BTP détermine quels procédés techniques seront utilisés lors de la construction d'un ouvrage. Il réalise des études, effectue des simulations, et prépare les plans des projets de construction dont il a la charge.

#### **Chargé d'affaires dans le BTP :**

Le chargé d'affaires a pour mission de gérer le portefeuille de clients et de prospecter afin d'élargir celui-ci. Mais il est également présent sur les chantiers où il peut être chargé d'acheter le matériel ou de gérer les différents intervenants qui participent à la réalisation d'un chantier.

## **1.3 Les besoins de service ressources humaines dans une entreprise de secteur BTP**

### **1.3.1 Gestion du recrutement**

- Définition de l'Organigramme détaillé de la société.
- Identification des postes vacants.
- Saisi/Suivi des offres d'emploi et Gestion du plan de recrutement.
- Processus détaillé de recrutement et Intégration automatique de l'employé sélectionné dans le système paie.
- Gestion des alertes et notifications.

### **1.3.2 Gestion de Carrière**

- Gestion de mobilisation administrative (Horizontale et/ou Verticale).
- Gestion de la fiche de poste.
- Gestion de décisions administratives.
- Évaluation de la performance des employés.
- Traitement des besoins et des plans de formation.
- Gestion Administrative du Personnel.
- Suivi et Édition des attestations de travail-certificat de travail-titre de congé.
- Suivi des reliquats de congé de tout le personnel.
- Gestion d'évaluations (Mensuelles, trimestrielles et périodiques).
- Gestion des Formations.
- Identification des besoins en formation.
- Détails Centres de formation et Formateurs.
- Plan et Planification de formation.
- Relevé de présence à la formation.
- Résultats et Évaluation Cycles Formations.

### **1.3.3 Gestion de paie**

- Gestion des employés (Détails employés).
- Attribution et Modification des rubriques de Paie Fixes et Variables.
- Gestion des différents types de paie et les méthodes de calculs.
- Rubrique temps (Liaison directe avec les pointeuses et la gestion du temps).
- Gestion des congés, avances, indemnités et dus employés.
- Intégration de la grille de salaire.
- Gestion détaillée des primes et mode de calculs.
- Régularisation salariale (Blocage/Déblocage & Verrouillage).
- Gestion des alertes et notifications (Rappels).
- Calcul et Édition de la masse salariale selon la période demandée.
- Interfaçage avec les logiciels de Gestion du temps.
- Comptabilisation de la paie.
- Gestion des services sociaux (CNAS, DAS, CNR).

### **1.3.4 Gestion de temps**

- Codification des périodes, jours fériés, autorisations, congés....
- Politique Heures de travail ainsi que les groupes de travail.
- Gestion des autorisations collectives et individuelles.
- Calcul Temps de présence ainsi que la mise à jour de la rubrique présence dans la paie.

## 1.4 Organisme d'accueil

### 1.4.1 Entreprise GTS développement

GTS développement est une SARL (Société A Responsabilités limitées) exerçant dans différents secteurs y'est compris le secteur BTP, elle se situe au Local n°06 ,11 rue OUTALEB Amar, Tizi-Ouzou Ville. Elle dispose d'environ 30 engins lourds et matériels stratégiques, et elle emploie environ 80 salariés.

### 1.4.2 Organigramme de l'entreprise GTS développement

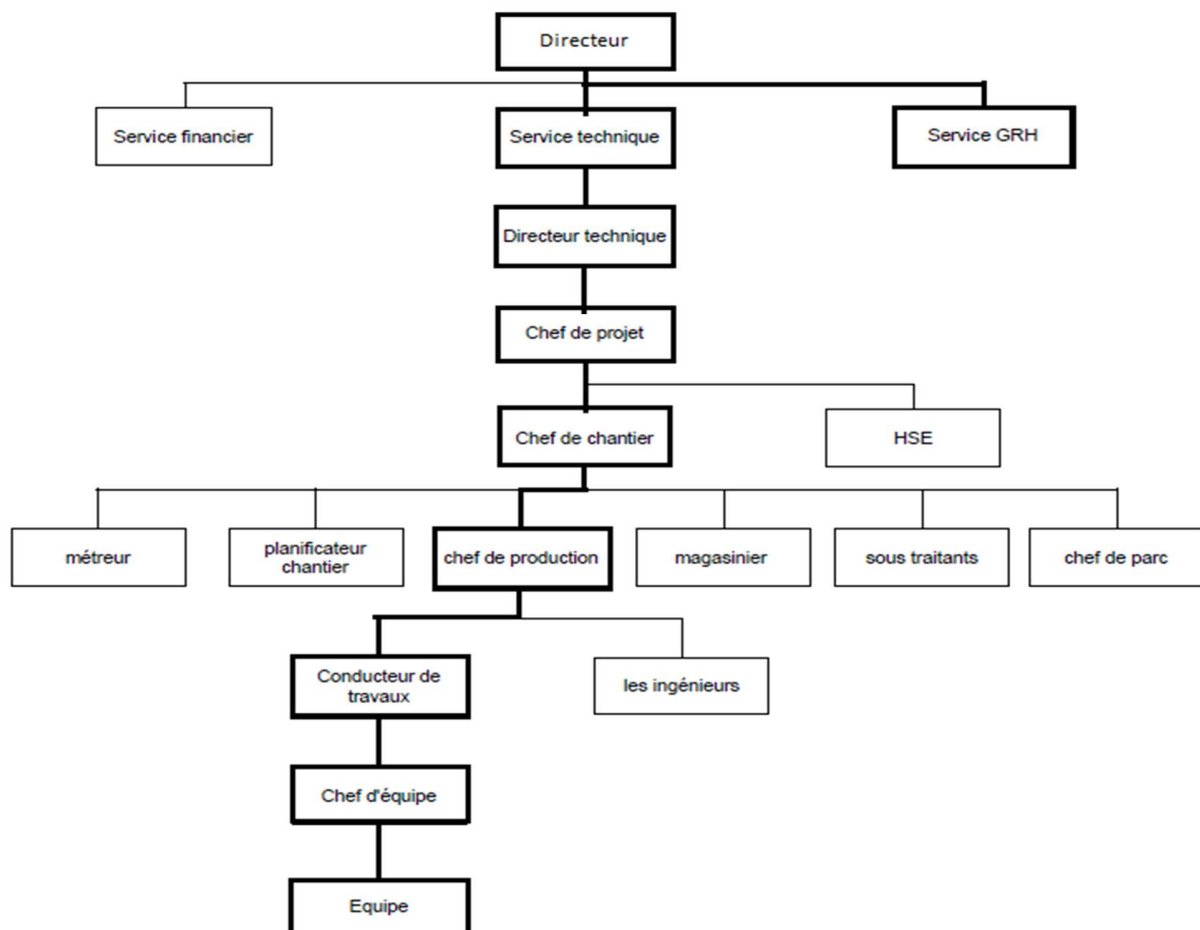


Figure 1 : Organigramme de l'entreprise GTS

### 1.4.3 Les activités de GTS développement

- Activité terrassement.
- Activité génie civil industriel.
- Activité bâtiment.
- Activité engineering.
- Activité construction en charpente métallique légère.

### 1.4.4 La problématique

Les applications mobiles sont actuellement en pleine croissance grâce aux succès des Smartphones et des Tablettes. Cette tendance offre aux entreprises des formidables opportunités pour créer des nouveaux services ou élargir ses supports de communication. Dans cette perspective, **l'entreprise GTS Développement** souhaite exploiter cette opportunité pour rendre la tâche de suivi de personnel plus simple et plus efficace.

Une application mobile pour le suivi de personnel permet de faire évoluer la communication entre le service de ressources humaines et les fonctionnaires sur chantier, Cette application transmettra les données en temps réel à ses utilisateurs ce qui permettra d'augmenter la performance de l'entreprise.

Ainsi, l'objectif de notre projet est de développer une application mobile qui permettra à l'entreprise d'évoluer et mieux gérer leurs personnels.

## **Conclusion**

Les grandes entreprises BTP doivent trouver des solutions plus modernes afin de résoudre leurs problèmes de gestion des ressources.

La gestion des ressources humaines a connu beaucoup de changement et afin de suivre son évolution, les entreprises doivent s'adapter à ce changement à l'aide d'un système d'information pour assurer une bonne gestion de leurs effectifs et planifier pour leurs futurs projets.

Le principal problème que face les entreprises du secteur BTP dans la gestion des employés, est la coordination entre service ressources humaines qui est un service administratif, et le chantier qui change d'emplacement à chaque nouveau projet et qui se trouve, la plupart du temps, loin du siège de l'entreprise.

Donc, l'idéal pour ces entreprises c'est d'avoir un système d'information qui sera partagé par tous leurs services, y'est compris le service ressources humaines et le chantier, afin d'avoir accès aux mêmes données et en temps réel, pour ajout ou récupération des informations des employés nécessaires à leurs fonctionnements.

# **Chapitre 2 : Les ERP**

## **Introduction**

La gestion des ressources est une tâche cruciale dans une entreprise dans n'importe quel secteur, si elle est faite d'une manière optimale, elle permet d'assurer un meilleur rendement dans ses activités et l'aboutissement aux résultats voulus, la délivrance du produit final dans les délais prévus, assurer sa bonne qualité et avec le cout minimal.

C'est dans ce contexte que les entreprises cherchent des outils informatiques afin de gérer leurs ressources d'une manière efficace et productive, et c'est là qu'intervient l'intérêt d'un ERP.

## **2.1 Les progiciels ERP**

Un ERP, "Enterprise Ressource Planning" traduit en français par Progiciel de Gestion Intégré ou PGI, est un progiciel qui permet de gérer l'ensemble des processus d'une entreprise intégrant l'ensemble de ses fonctions comme la gestion des ressources humaines, la gestion financière et comptable, l'aide à la décision, la vente, la distribution, l'approvisionnement, la production ou encore du e-commerce.

## **2.2 Principes de fonctionnement des ERP**

### **2.2.1 Principe Fondateur**

Le principe fondateur d'un ERP est de construire des applications informatiques correspondant aux diverses fonctions citées précédemment d'une manière modulaire sachant que ces modules sont indépendants entre eux, tout en partageant une base de données unique et commune au sens logique.

## **2.2.2 Le moteur workflow**

L'autre principe qui caractérise un ERP est l'usage de ce qu'on appelle un moteur de workflow et qui permet, lorsqu'une donnée est enregistrée dans le Système Informatique, de la propager dans les modules qui ont l'utilité, selon une programmation prédéfinie.

Un moteur de workflow est un dispositif logiciel permettant d'exécuter des instances de workflow (l'enchaînement des activités décrit par la définition de processus de workflow).

Il s'agit en effet de formaliser les traitements à réaliser, le cheminement à suivre et les acteurs concernés pour accomplir un travail précis).

## **2.3 Utilisateurs des ERP**

Les ERP sont principalement destinés aux grandes entreprises ou multinationales du fait d'un coût important. Cependant, le marché des ERP tend à se démocratiser vers les PME (petites et moyennes entreprises) /PMI (petites et moyennes industries).

## **2.4 Avantages de la mise en place d'un ERP**

Concrètement, les avantages de la mise en place d'un ERP sont les suivants:

- L'intégrité et l'unicité du SI, un ERP permet d'éviter la redondance d'informations entre différents SI de l'entreprise, Ceci se traduit par le fait qu'il peut exister plusieurs bases de données "physiques" mais celles-ci respectent la même structure.
- Mises à jour en temps réel, Un avantage important car L'utilisateur a la possibilité de récupérer des données de manière immédiate, ou encore de les enregistrer.
- Un ERP peut être un outil multilingue et multidevise, il est donc adapté au marché mondial, en particulier aux multinationales.
- Un ERP permet de maîtriser la gestion de la variété des ressources de l'entreprise.

## 2.5 Architecture des progiciels ERP

L'architecture logicielle décrit d'une manière symbolique et schématique les différents éléments d'un système informatique comme le montre la figure suivante :

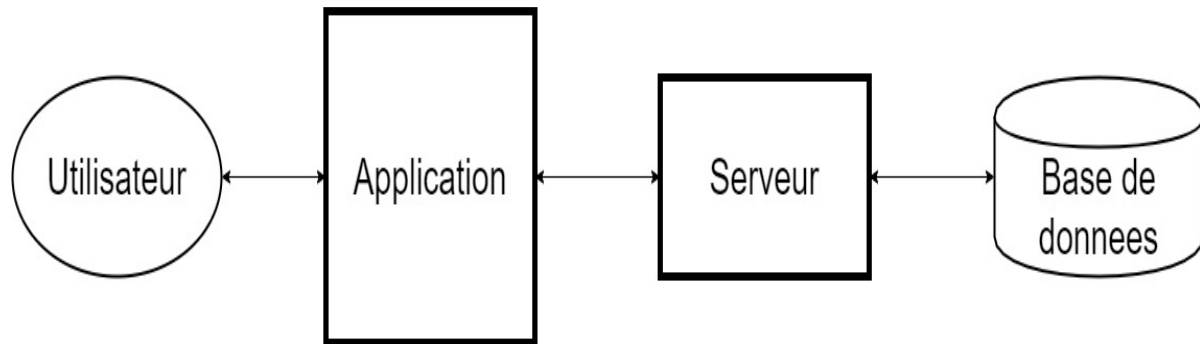


Figure 2 : Architecture logiciel

## Conclusion

Les ERP sont un très bon moyen pour gérer les ressources d'une entreprise qui exerce dans le secteur BTP, elles permettent la coordination entre les différents services de l'entreprise, du début de projet jusqu'à la fin, et avec l'introduction des applications Mobiles dans l'ERP, qui sert d'intermédiaire entre chantier et les services administratives, la tâche du suivi de personnel n'a jamais été aussi facile et coordonnée.

# Chapitre 3 : Analyse et Conception

## Introduction

Nous entamons la phase analyse et conception, ou on doit faire des choix concernant la méthode de travail et les outils qu'on va utiliser, puis nous allons élaborer l'architecture de notre application mobile, et cela en se basant sur l'analyse des besoins qu'on doit satisfaire dans notre produit final.

On commence par le choix d'un cycle de développement, entre le cycle de développement en cascade et le cycle développement en V, Ou en doit choisir le cycle le plus adéquat a notre réalité.

Puis, on passe aux choix des outils de développement, des choix basés sur les avantages offerts par ces outils et leurs adaptabilités à notre cas.

Ensuite, on entame l'analyse des besoins qu'on doit satisfaire, des besoins répartis en besoins fonctionnels et besoins non-fonctionnels, et après on identifie les acteurs intervenants dans l'application, on définit leurs noms et leurs rôles, pour qu'on puisse illustrer cette réalité par un diagramme de contexte.

En Finira par la conception qui est la phase la plus importante dans notre travail, elle permet l'illustration du produit final en utilisant des diagrammes UML, tels que les diagrammes de cas d'utilisation, de séquences et de classes.

## 3.1 Choix de cycle de développement

### 3.1.1 Cycle de développement en V

Le cycle de développement en V est une évolution du cycle en cascade qui manquait de réactivité, Il évite les retours en cas d'anomalie rencontrée. Il est composé d'une phase descendante puis montante, la phase montante envoie des informations vis-à-vis de la phase descendante.

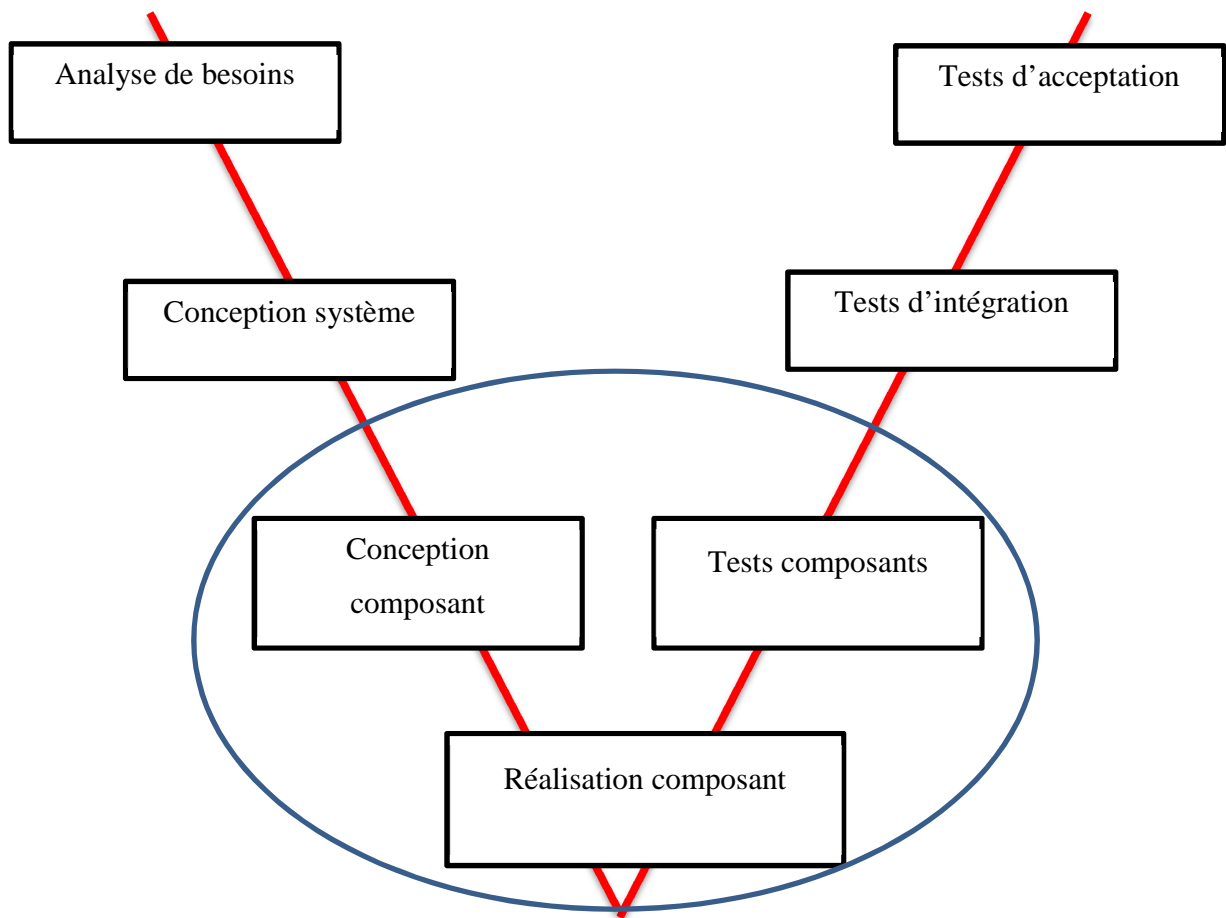


Figure 3 : Cycle de Développement en V

### 3.1.1.1 Les étapes du cycle de développement en V

Le cycle en V est un cycle composé de 3 grandes phases contenant 8 étapes de conception d'un produit :

#### La phase de conception

- Analyse des besoins et faisabilité.
- Spécifications.
- Conception Architecturale.
- Conception détaillée.

### **La phase de réalisation**

- Codage.
- Testes unitaires.

### **La phase de validation**

- Testes d'intégration
- Testes de validation et Recette

#### **3.1.1.2 Les avantages du cycle en V**

La méthode de cycle en V possède de nombreux avantages dans le développement d'une application. La mise en œuvre du projet est sécurisée dans la mesure où chaque phase ne commence qu'à partir du moment où la précédente est terminée. Elle permet de bien préparer en amont les phases de spécification. En face de chaque phase de spécification, est mis en place un système de vérification, ce qui assure un meilleur produit au final.

#### **3.1.1.3 Les inconvénients du cycle en V :**

Il faut bien planifier un projet qui suit la méthode du cycle en V. Si le projet prend du retard, le temps accordé pour les tests risque d'être insuffisant. D'autant plus que la phase de validation est une phase indispensable pour le projet, elle permet de garantir la qualité du produit livré.

### 3.1.2 Cycle de développement en cascade

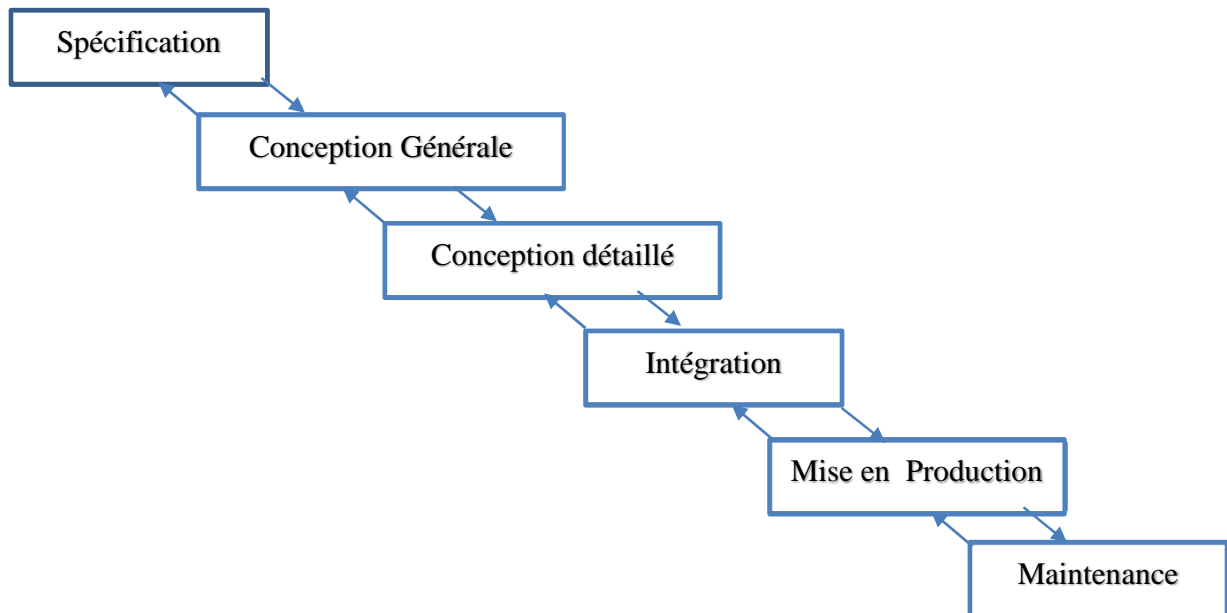


Figure 4 : Cycle de Développement en Cascade

#### 3.1.2.1 Les étapes de de cycle de vie en cascade

Cette méthode constitue six ou sept étapes :

##### **L'analyse des besoins :**

On étudie la faisabilité du projet pour définir les besoins et les exigences. Seuls les processus majeurs sont concernés. En estimant les différents coûts du projet et les matériaux nécessaires. Il faut chiffrer les charges et envisager des délais.

##### **L'analyse du système :**

Cette étape permet de peaufiner les points vus dans l'étape précédente. On peut commencer à traiter les points mineurs.

##### **Conception :**

Ici on effectue les différents choix techniques.

### **Implémentation et tests unitaires :**

Dans cette étape on écrit des programmes en fonction du langage choisi, les développeurs vont programmer. Une fois terminé, on effectue des tests unitaires. Ces tests vont permettre de vérifier si tout fonctionne correctement.

### **Validation et tests d'intégration :**

Le client à son tour effectue des tests pour valider le système. Les tests d'intégration se font souvent avec les utilisateurs.

### **Exploitation et maintenance :**

Dans cette dernière étape, il ne reste plus qu'à déployer le système. Et la maintenance permet de corriger les problèmes et les anomalies.

## **3.1.3 Déférence entre le cycle de développement en V et le cycle de développement en cascade**

Le modèle de cascade est un processus continu, tandis que le modèle V est un processus simultané. Par rapport à un logiciel de rendu en utilisant le modèle cascade, le nombre d'erreurs dans le logiciel fait en réalisé en modèle V sont moins. Cela est dû au fait qu'il y'a des activités d'essais, qui sont effectuées simultanément dans le modèle V. Par conséquent, le modèle de cascade est utilisé, lorsque les besoins de l'utilisateur sont fixes. Si les besoins de l'utilisateur sont incertains et ne cessent de changer, le modèle V est la meilleure alternative. Aussi des changements dans le logiciel de modèle de cascade est une tâche difficile.

## **3.1.4 Notre Choix de cycle de développement**

Pour notre projet nous avons opté pour **le cycle de développement en V**, pour ce qu'il offre d'avantages et c'est le cycle le mieux adapter à notre cas de travail car On est confronté à une situation où on doit développer des composants(packages) séparément, puis les intégrer dans un seul projet après avoir effectué des tests de validation sur ces composants, dans notre vision

du projet, on a prévu d'avoir 3 packages principales : Gestion de carrière, gestion de temps, gestion de paie, donc on aura 3 composants à développer séparément, comme on a déjà une conception dédié pour chaque package, on a qu'a commencé le développement puis effectuer des tests de validation sur ce package si les tests sont biens déroulé alors on l'intègre au projet principal puis on passe au prochain composant.

Le but de cette méthode de travail est d'assurer qu'on a une fonctionnalité à présenter au client à un moment donné pour avoir son avis et ses suggestions pour nous.

## **3.2 Outils de conception**

Pour bien détailler les fonctionnalités de notre application et les interactions entre l'utilisateur et l'application on a choisi le langage de modélisation UML.

### **3.2.1 Langage de modélisation UML**

Abréviation de Unified Modeling Language, est un langage de modélisation normalisé composé d'un ensemble intégré de diagrammes développé pour aider les développeurs de systèmes et de logiciels à spécifier, visualiser, construire et documenter les artefacts des systèmes logiciels, ainsi que pour la modélisation d'entreprise et d'autres systèmes non logiciels.

L'UML est une partie très importante du développement de logiciels orientés objet et du processus de développement logiciel. L'UML utilise principalement des notations graphiques pour exprimer la conception de projets logiciels. L'utilisation de l'UML aide les équipes de projet à communiquer, à explorer les conceptions potentielles et à valider la conception architecturale du logiciel.

### 3.2.2 Les 13 diagrammes de l'UML

- Le diagramme de package.
- Le diagramme de cas d'utilisation.
- Le diagramme d'activité.
- Le diagramme de classes.
- Le diagramme d'objets.
- Le diagramme de séquence.
- Le diagramme d'activité.
- Le diagramme de collaboration.
- Le diagramme global d'interaction.
- Le diagramme de temps.
- Le diagramme de structure composite.
- Le diagramme de composants.
- Le diagramme de déploiement.

En rajoutant aussi **Le diagramme de contexte** :

Ce diagramme n'est pas officiellement désigné comme diagramme UML. Il ne fait donc pas parti des 13 diagrammes « officiels », mais il est utile pour la définition des acteurs, avant de commencer à s'intéresser à d'autres aspects, tels que les packages et les cas d'utilisation.

### 3.3 Les Besoins fonctionnels

Dans cette partie nous détaillons les fonctionnalités, que le système doit fournir aux différents acteurs, qui se présentent comme suit :

- **Gérer les utilisateurs :**

Mettre à jour les utilisateurs ainsi que l'attribution des droits d'accès aux données.

- **Création d'un compte administrateur :**

«Le compte qui sera entre les mains du responsable RH dans l'entreprise et l'attribution des droits d'accès aux données.

- **Gérer les nouvelles données en temps réels :**

Les évaluations continues, les absences, les remarques en gardant trace des gens qui les partagent.

- **Gérer les notifications en temps réels :**

Les trois acteurs doivent être prévenus en temps réel de tout changement ou message.

- **Dans le cadre de ce travail, l'application devra être extensible :**

C'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier des fonctionnalités.

### 3.4 Les Besoins non fonctionnels

- **Authentification :**

Consistant à assurer que seules les personnes autorisées aient accès aux ressources, ces personnes doivent accéder à leurs espaces de travail après avoir saisi leur nom et leur mot de passe, cet espace de travail doit être créé par l'administrateur de l'application, dans notre cas, c'est le service de ressources humaines qui s'occupe d'attribution des comptes aux utilisateurs concernés.

- **Intégrité et confidentialité :**

C'est-à-dire garantir que les données sont bien celles que l'on croit être et qu'elle arrive à la bonne personne, et ça en donnant les droits d'accès aux ressources pour les personnes autorisées, afin d'éviter tout action non-voulu qui touche l'intégrité des données et aussi en instaurant un pare-feu qui sert de bouclier a tout intrusion.

- **Disponibilité :**

Permettant de maintenir le bon fonctionnement de l'application, et ça en assurant une maintenance périodique de l'application et de tout le système (code, BDD, Serveur, Terminaux...).

- **Non répudiation :**

Permettant de garantir qu'une transaction ne peut être niée, en utilisant des ID de session reliant un utilisateur à une action donnée, avec archivage de toute transaction avec toutes les données qui la concerne.

- **Fraicheur des données :**

Toute action (modification, insertion, suppression, communication...) doit s'effectuer d'une manière synchrone sur tous les terminaux, pour garantir une mise à jour des données et l'arrivée de l'info en moindre temps possible.

- **Ergonomie:**

L'application doit offrir une interface conviviale, explicite et simple à utiliser.

- **Le contrôle des champs :**

L'application doit avoir un contrôleur des champs de saisis, pour éviter l'introduction des informations qui ne correspondent pas aux types des champs.

## **3.5 Les éléments qui composent l'application**

Nous allons maintenant énumérer les acteurs susceptibles d'interagir avec le système. Tout d'abord, nous commençons par définir ce qui est un acteur.

### **3.5.1 Définition d'un acteur:**

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec l'application.

Donc dans notre cas, Les acteurs de notre application sont :

### **3.5.2 Le chef de chantier**

- Noter le travail d'un employé sur chantier, et ça en utilisant une échelle prédéfinie.
- Noter les absents sur chantier.
- Affecter des tâches pour les employés et effectuer des remarques sur le déroulement de travail.

### **3.5.3 Le chef d'équipe**

- Consulter la liste des employés de son équipe.
- Affecter des congés pour les employés de son équipe.

### **3.5.4 Le service ressources humaines**

- L'insertion des données de nouveaux employés.
- Récupérer les notes et les remarques faites par les chefs de chantiers, pour les utiliser dans leur travail

### **3.5.5 Directeur d'entreprise, directeur technique et le chef de projet**

- Consulter la liste des employés et la liste des postes
- Envoyer et Recevoir des notifications.

Et pour mieux présenter ces acteurs et leurs interactions avec l'application, nous avons mis en œuvre le diagramme de contexte suivant :

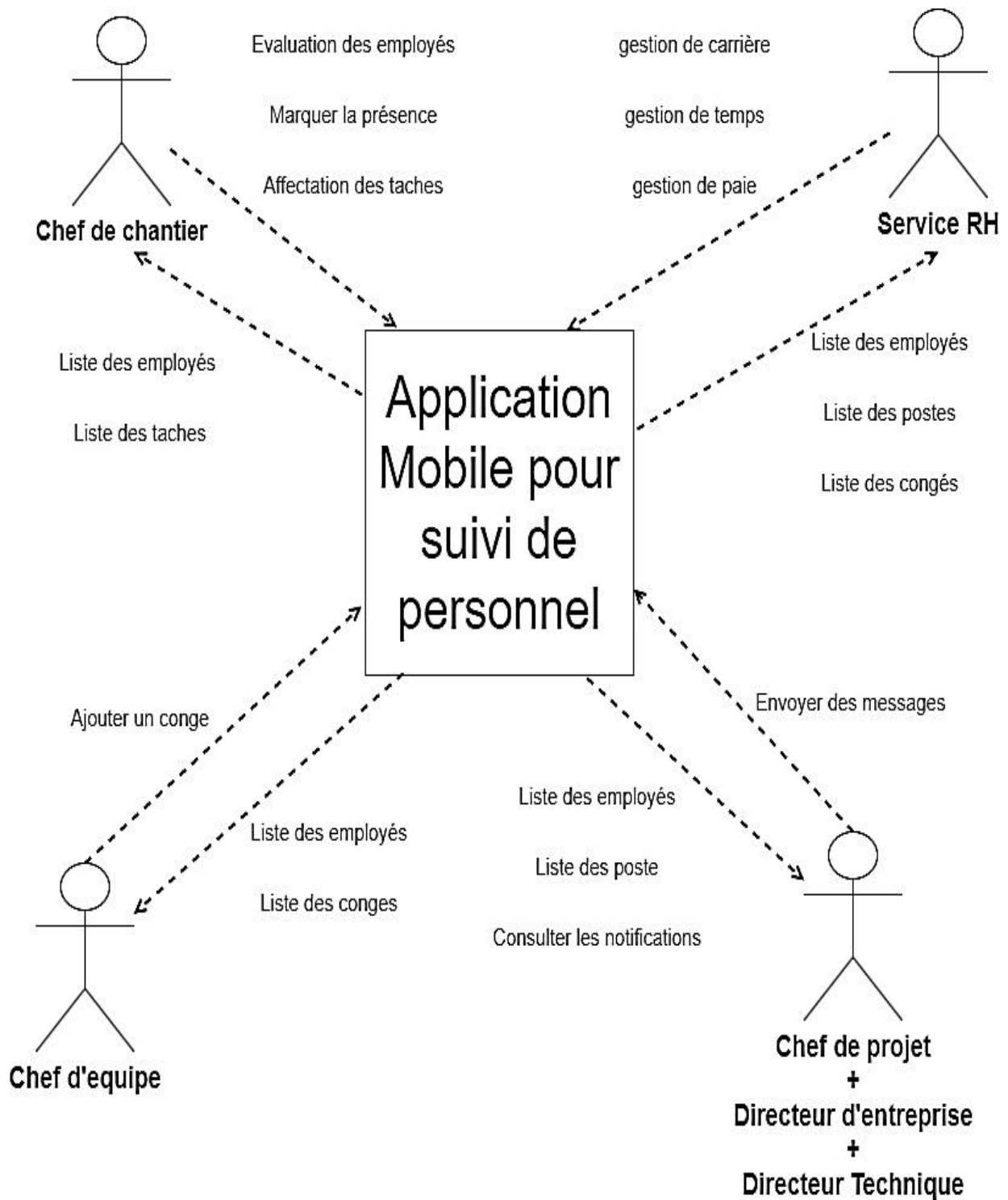


Figure 5 : Diagramme de contexte

## 3.6 Les fonctionnalités que l'application doit satisfaire

Notre application doit assurer certaines fonctionnalités pour les éléments mentionnés précédemment qu'on va illustrer avec des diagrammes de cas d'utilisation.

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les interfaces de future application. C'est le premier diagramme UML constitué d'un ensemble d'acteurs qui agit sur des cas d'utilisation et qui décrit sous la forme d'actions et des réactions, le comportement d'un système du point de vue utilisateur.

### 3.6.1 Les fonctionnalités de directeur de l'entreprise, directeur technique et le chef de projet

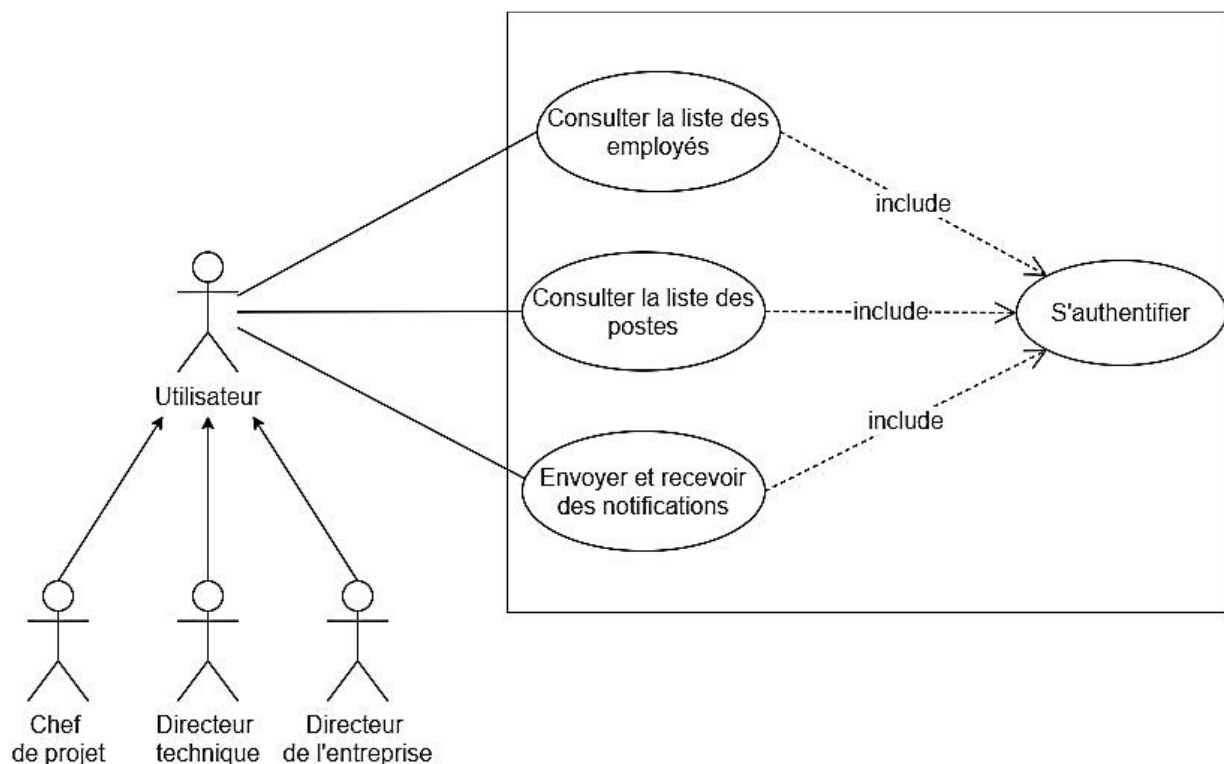


Figure 6 : Diagramme de cas d'utilisation « directeur Technique », « Directeur », « chef projet »

### 3.6.2 Les fonctionnalités de services ressources humaines

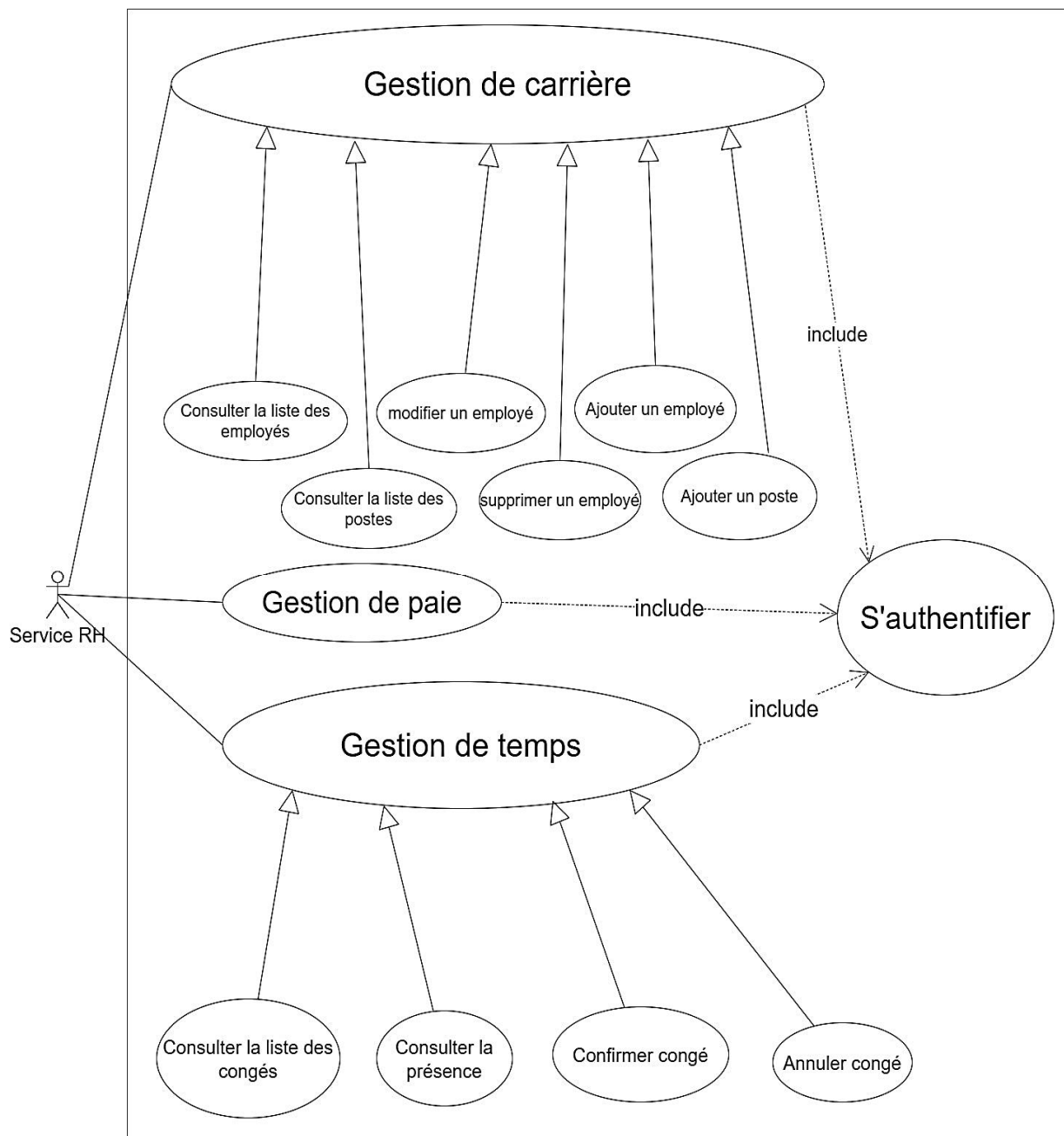


Figure 7 : Diagramme de cas d'utilisation « SRH »

### 3.6.3 Les fonctionnalités de chef de chantier

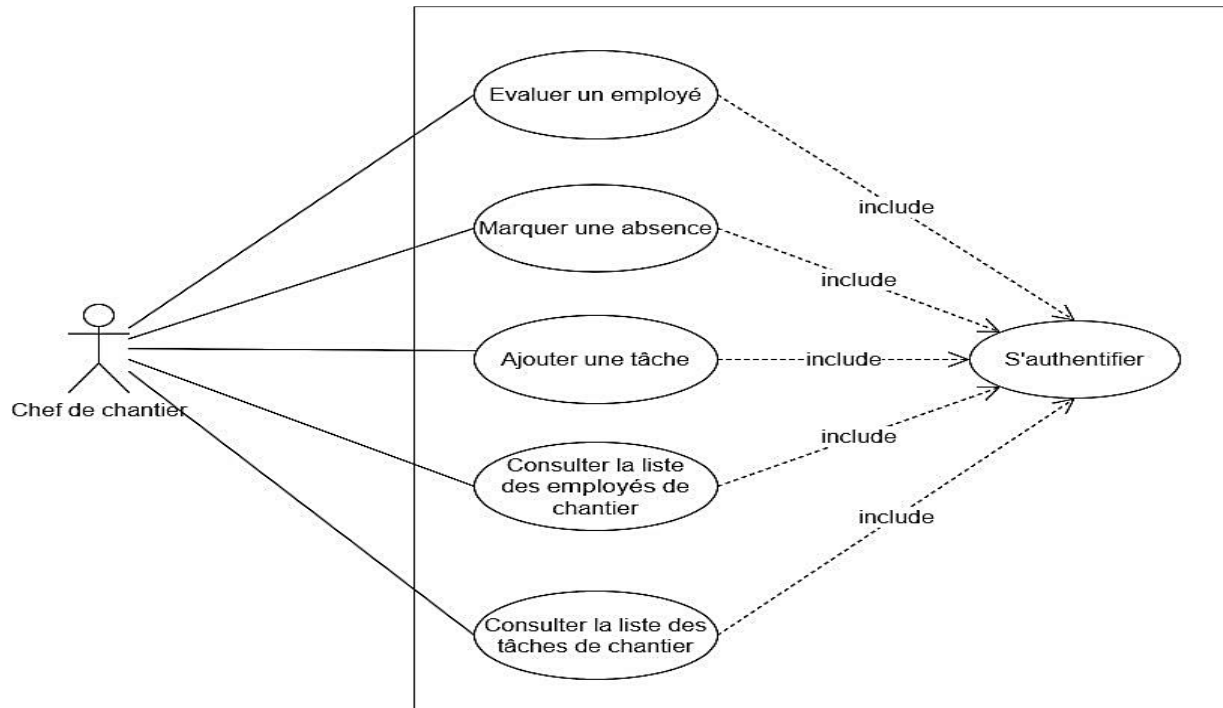


Figure 8 : Diagramme de cas d'utilisation « chef de chantier »

### 3.6.4 Les fonctionnalités de chef d'équipe

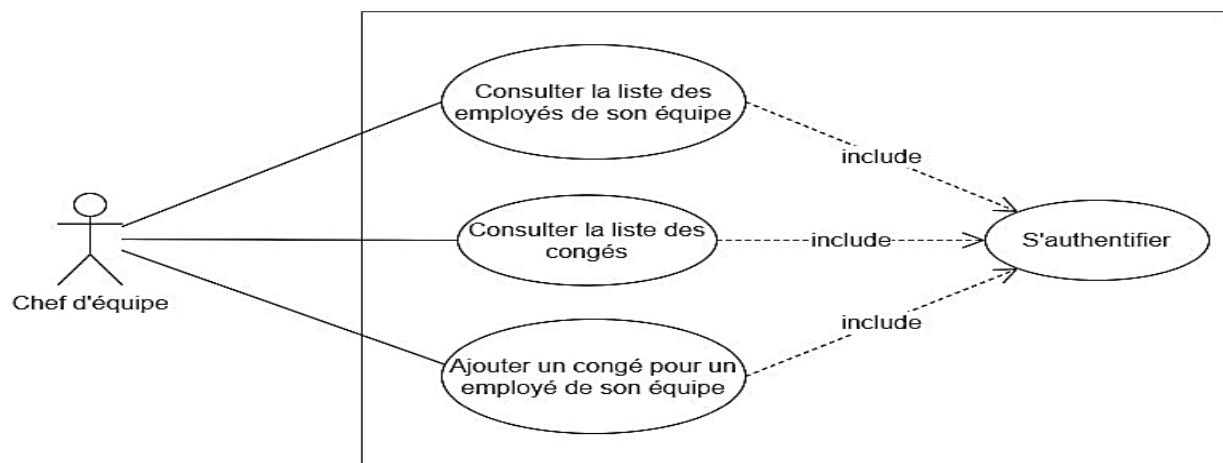


Figure 9 : Diagramme de cas d'utilisation « chef d'équipe »

## 3.7 Les interactions de l'application avec l'utilisateur

Ici nous allons mentionner les différentes interactions de l'application avec l'utilisateur qu'on va illustrer avec des diagrammes de séquences.

Les diagrammes de séquence peuvent servir à illustrer les cas d'utilisations décrits précédemment. Ils permettent de représenter la succession chronologique des opérations réalisées par un acteur et qui font passer d'un objet à un autre pour représenter un scénario.

### 3.7.1 Authentification

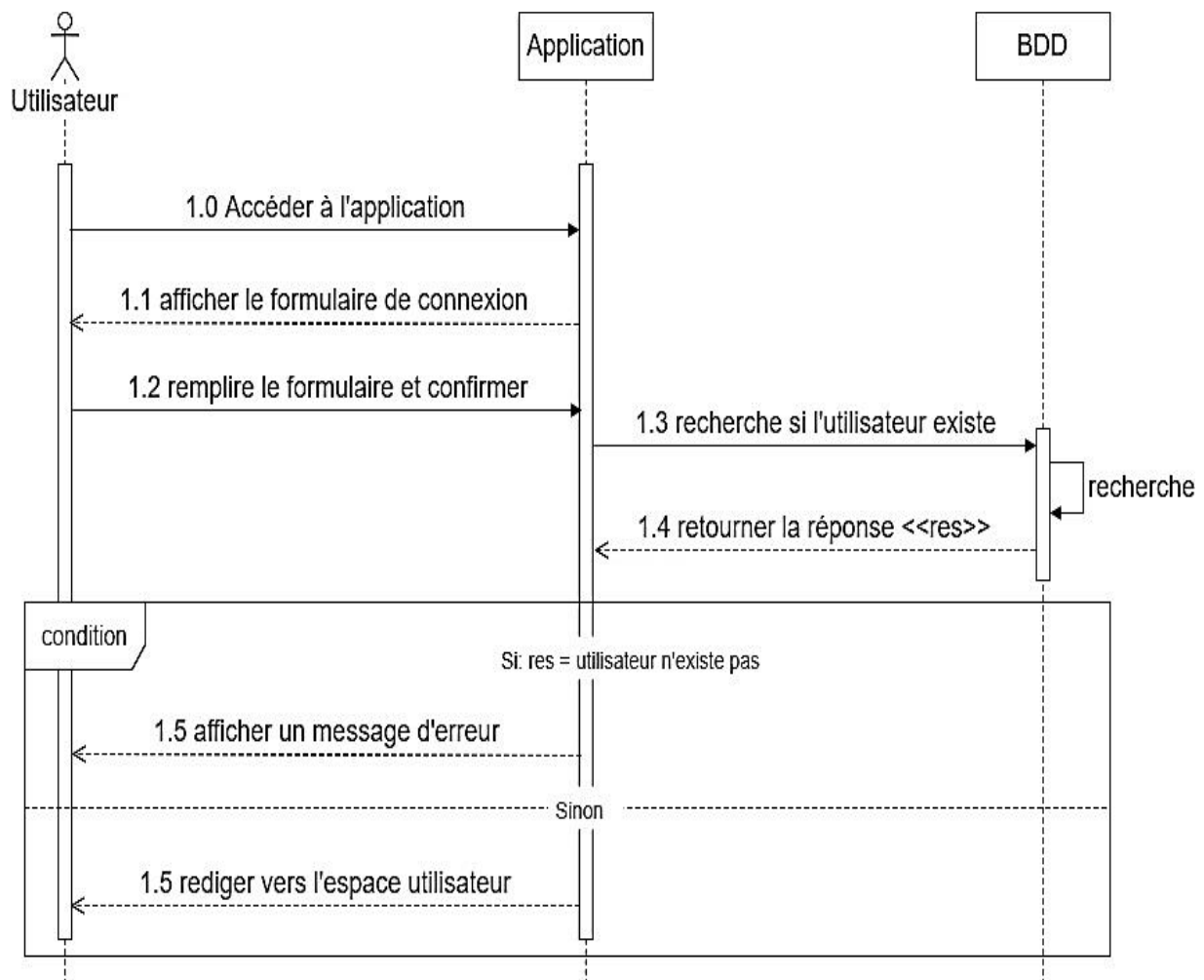


Figure 10 : Diagramme de séquence « Authentification »

### 3.7.2 Ajouter un employé

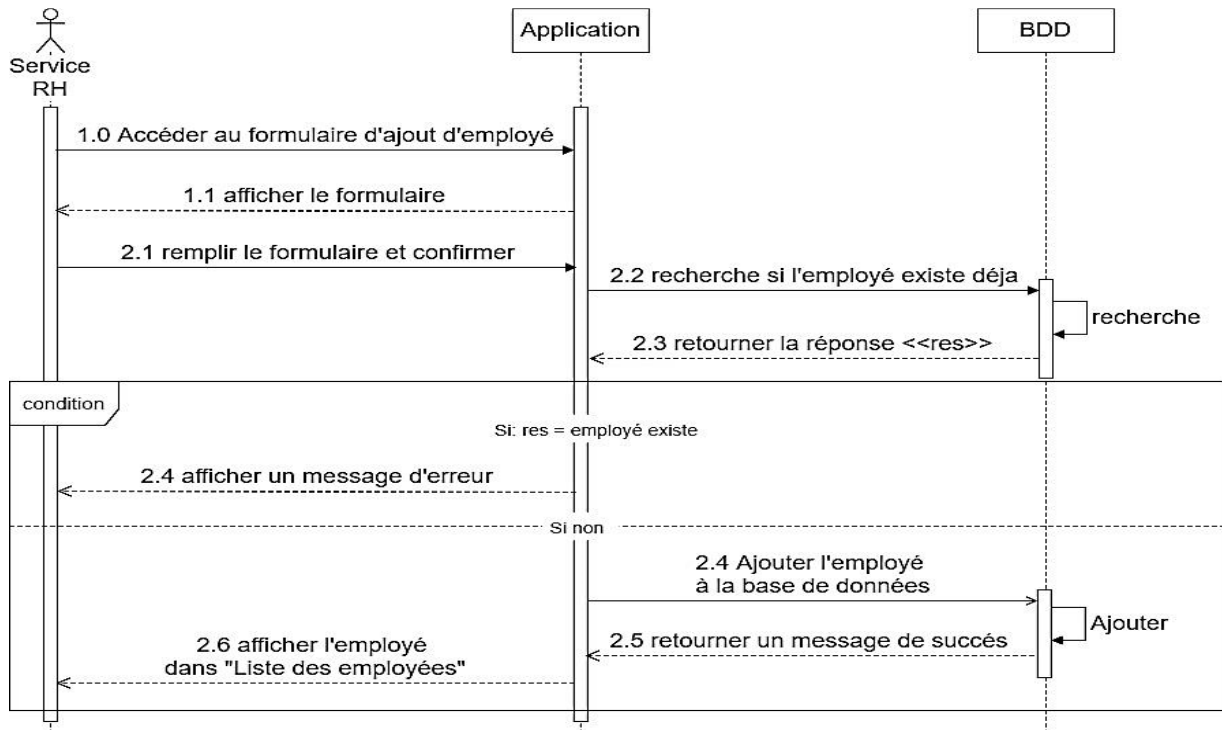


Figure 11 : Diagramme de séquence « Ajouter un Employé »

### 3.7.3 Ajouter un poste

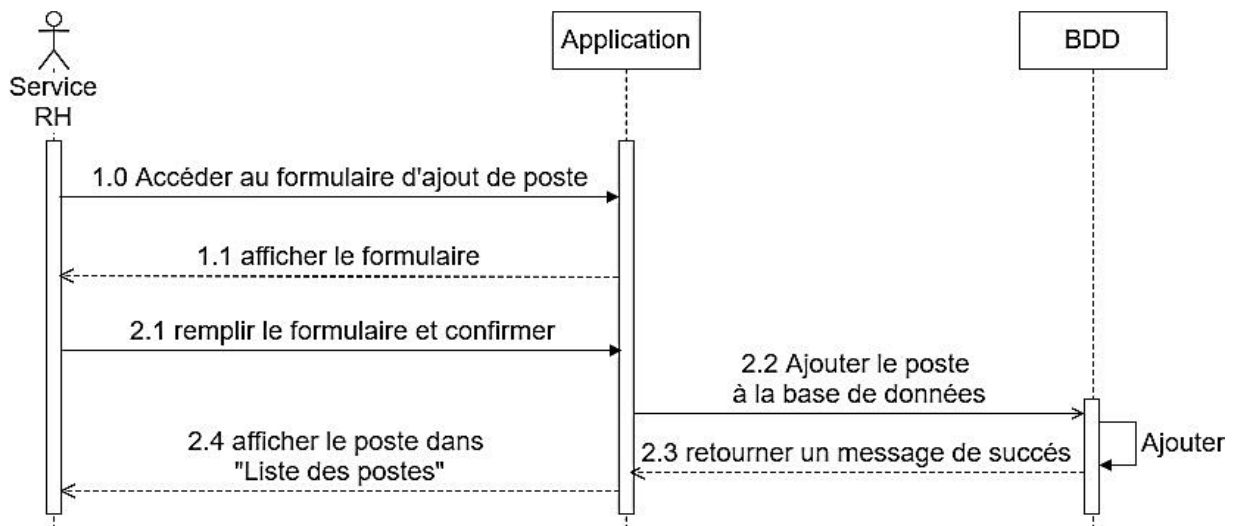


Figure 12 : Diagramme de séquence « Ajouter un Poste »

### 3.7.4 Ajouter un congé

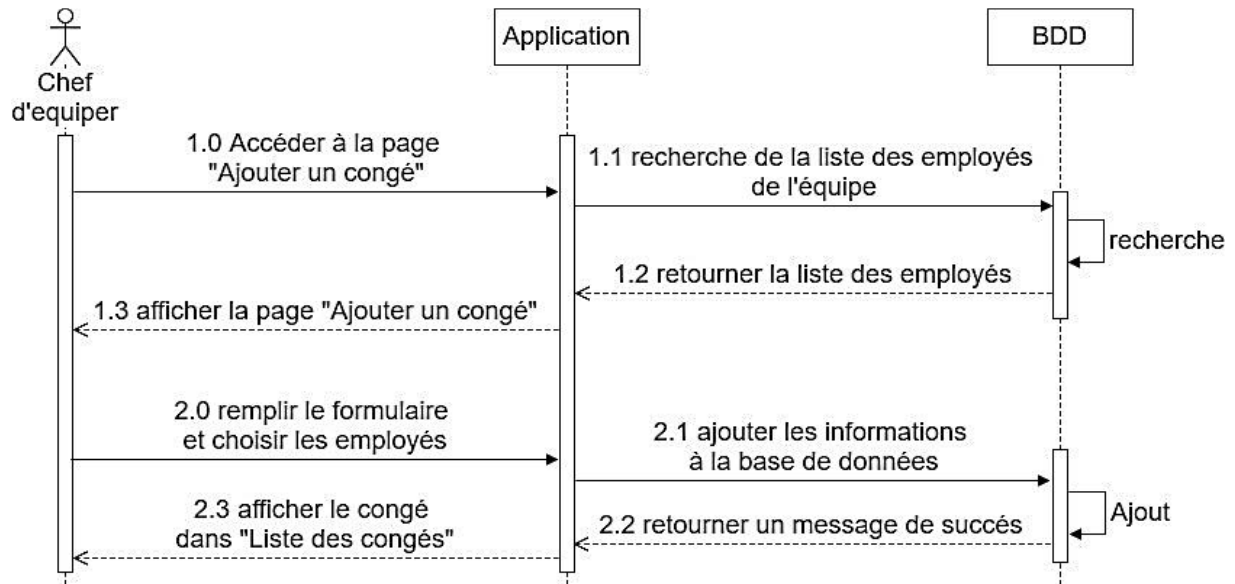


Figure 13 : Diagramme de séquence « Ajouter un Congé »

### 3.7.5 Ajouter une tâche

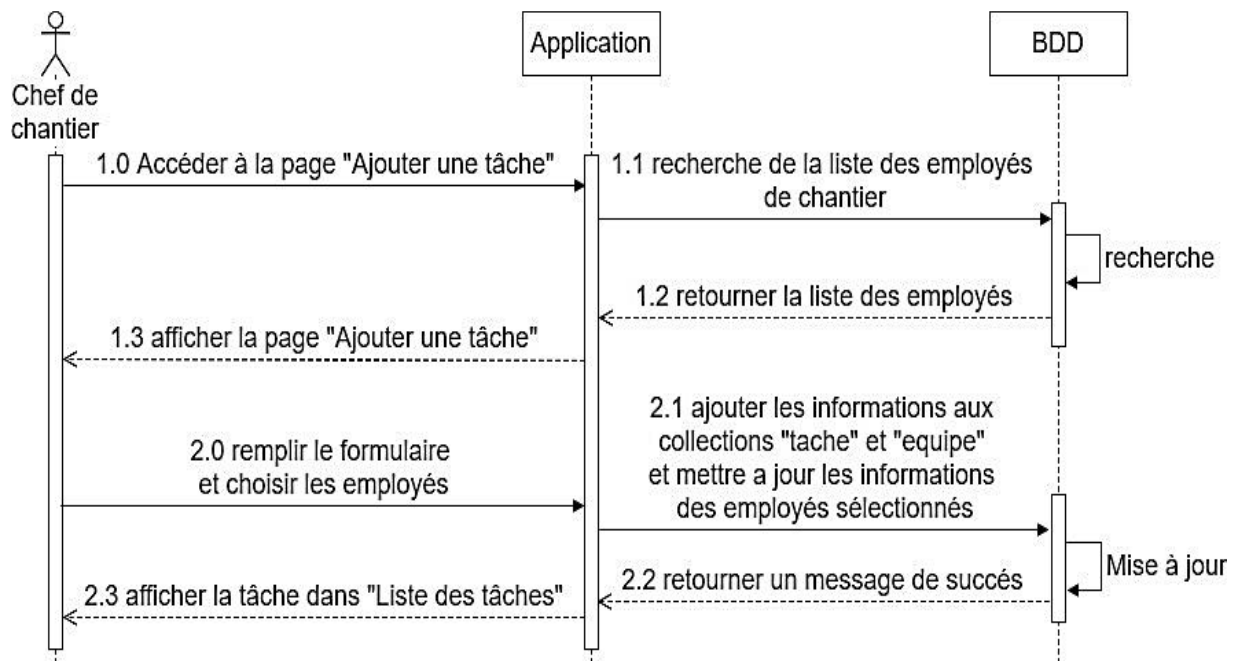


Figure 14 : Diagramme de séquence « Ajouter une Tâche »

### 3.7.6 Modifier un employé

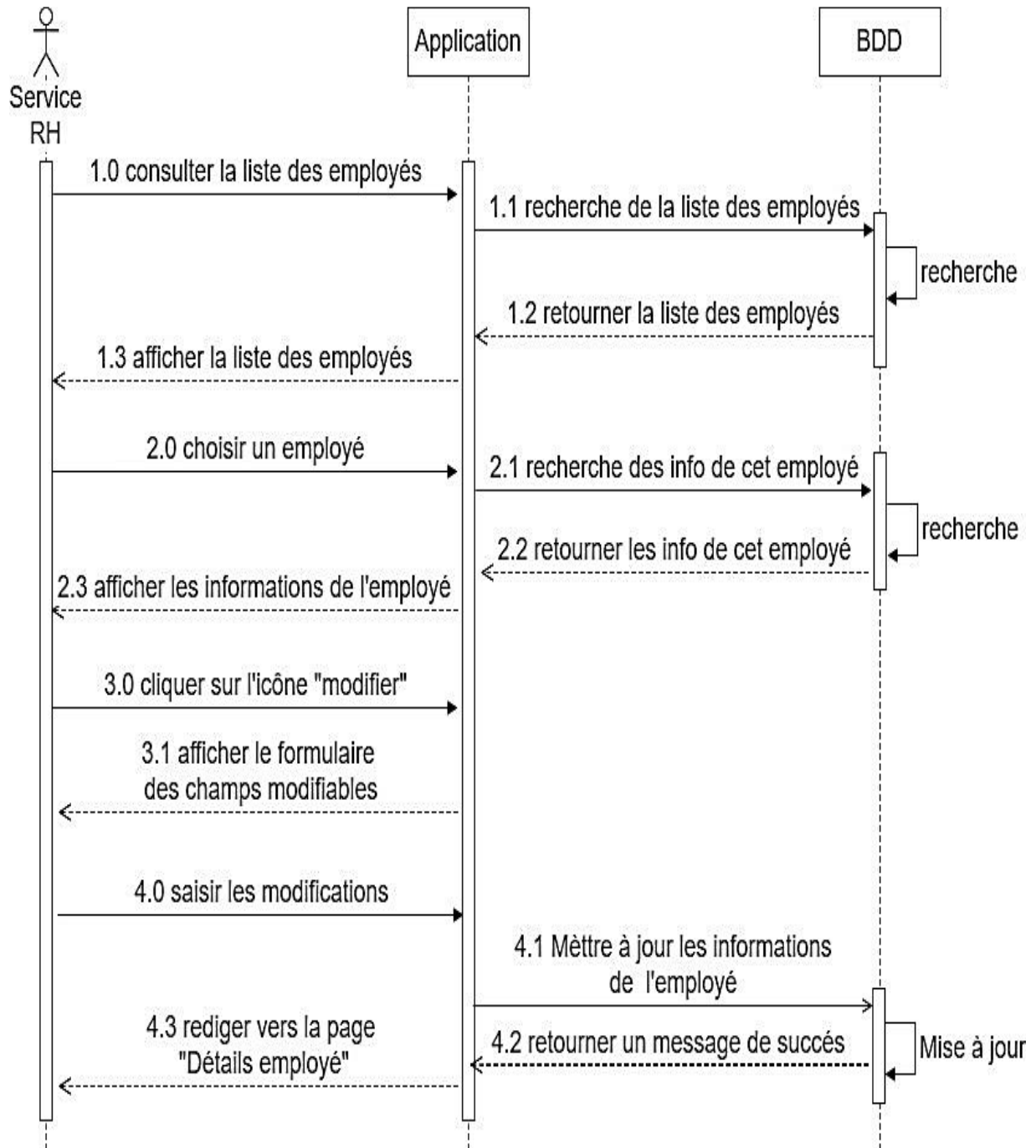


Figure 15 : Diagramme de séquence « Modifier un Employé »

### 3.7.7 Supprimer un employé

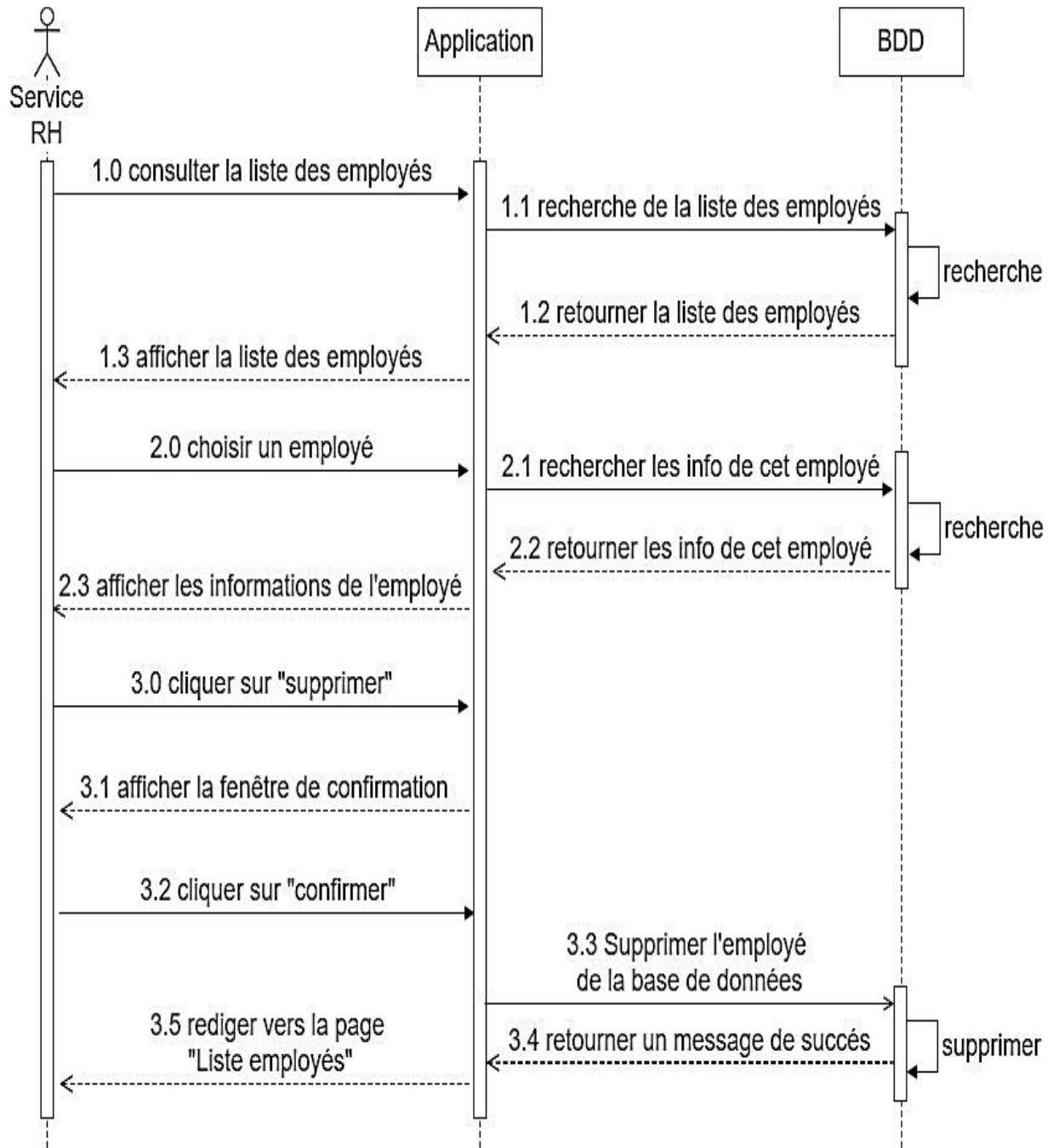


Figure 16 : Diagramme de séquence « Supprimer un Employé »

### 3.7.8 Confirmer ou annuler un congé

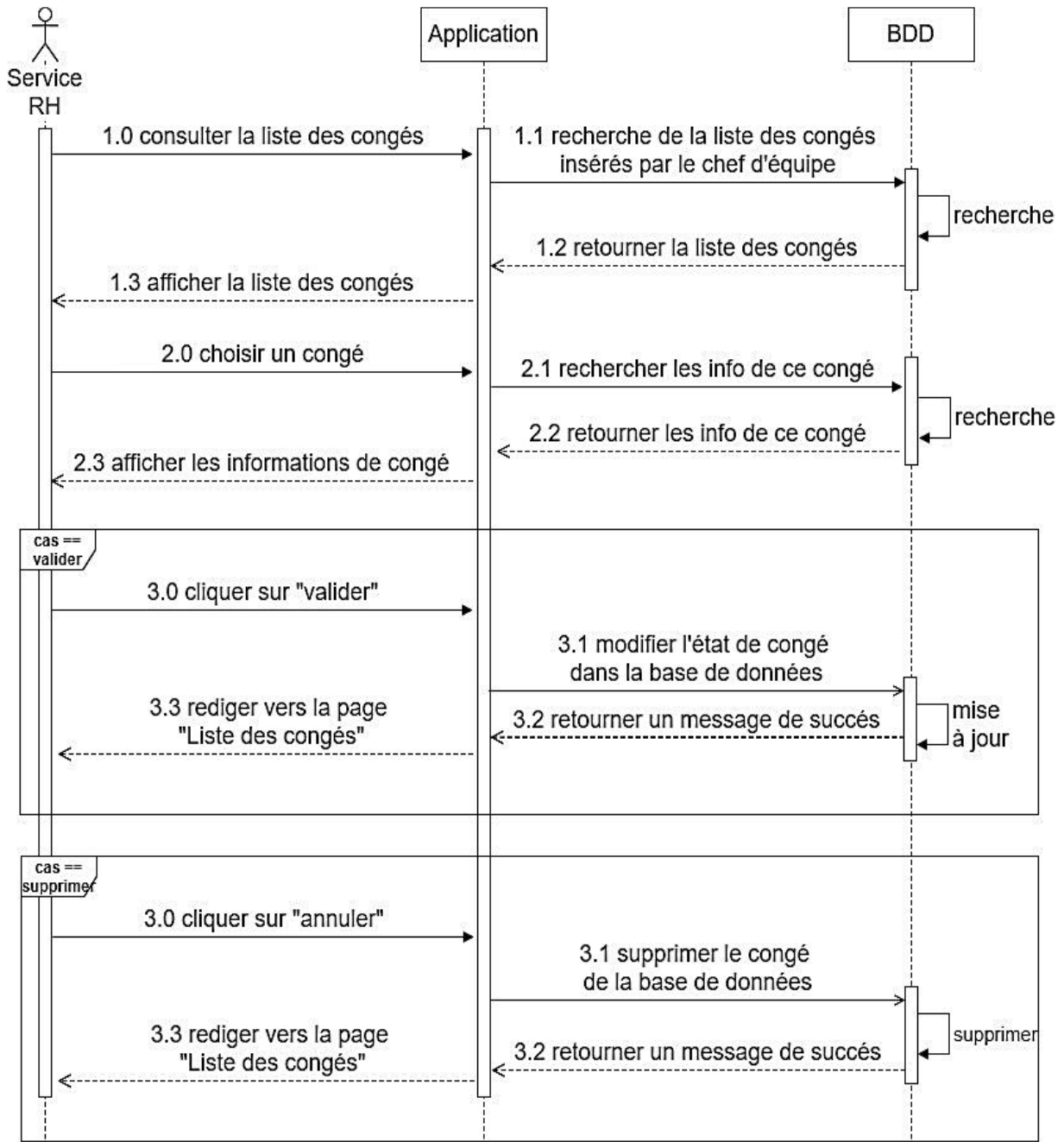


Figure 17: Diagramme de séquence « Confirmer ou Annuler un Congé »

### 3.7.9 Marquer une absence

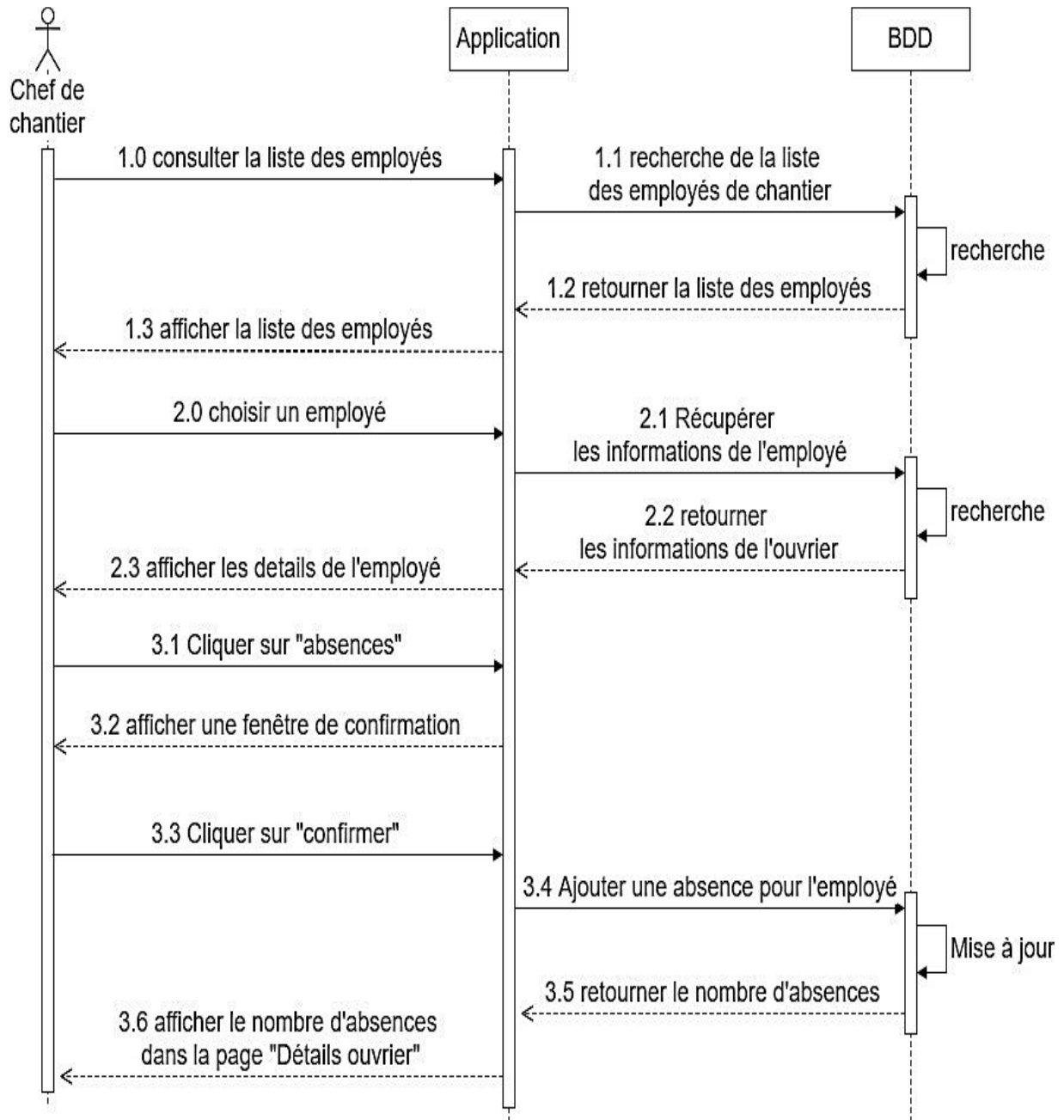


Figure 18 : Diagramme de séquence « Marquer une Absence »

### 3.7.10 Evaluer un employé

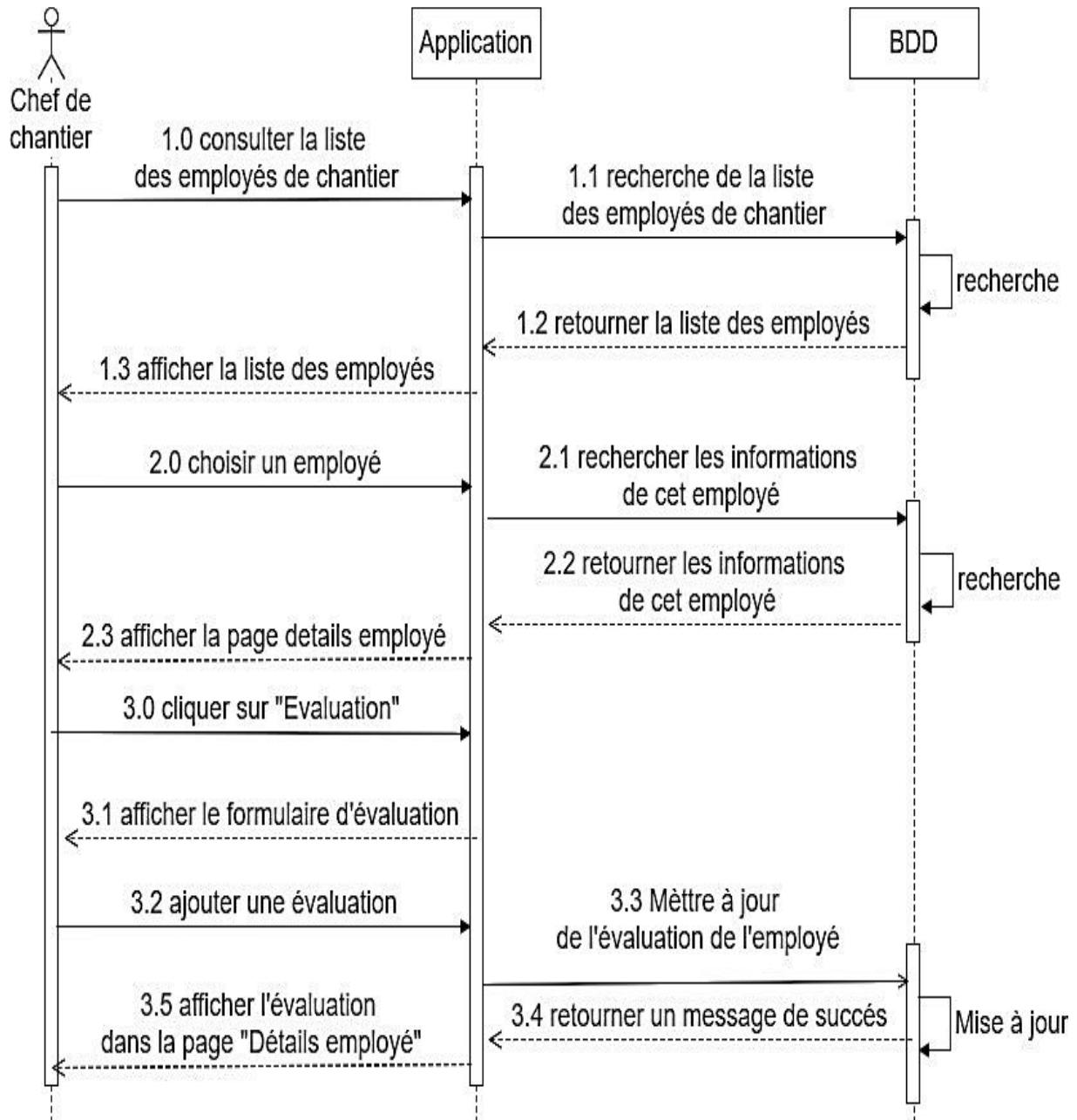


Figure 19 : Diagramme de séquence « Evaluation d'un Employé »

### 3.8 Diagramme de classes

Le diagramme de classes exprime la structure statique du système en termes des classes et des relations entre ces classes. Il fait abstraction des aspects dynamiques et temporels. La figure suivante montre le diagramme de classes qui collaborent au fonctionnement de notre application.

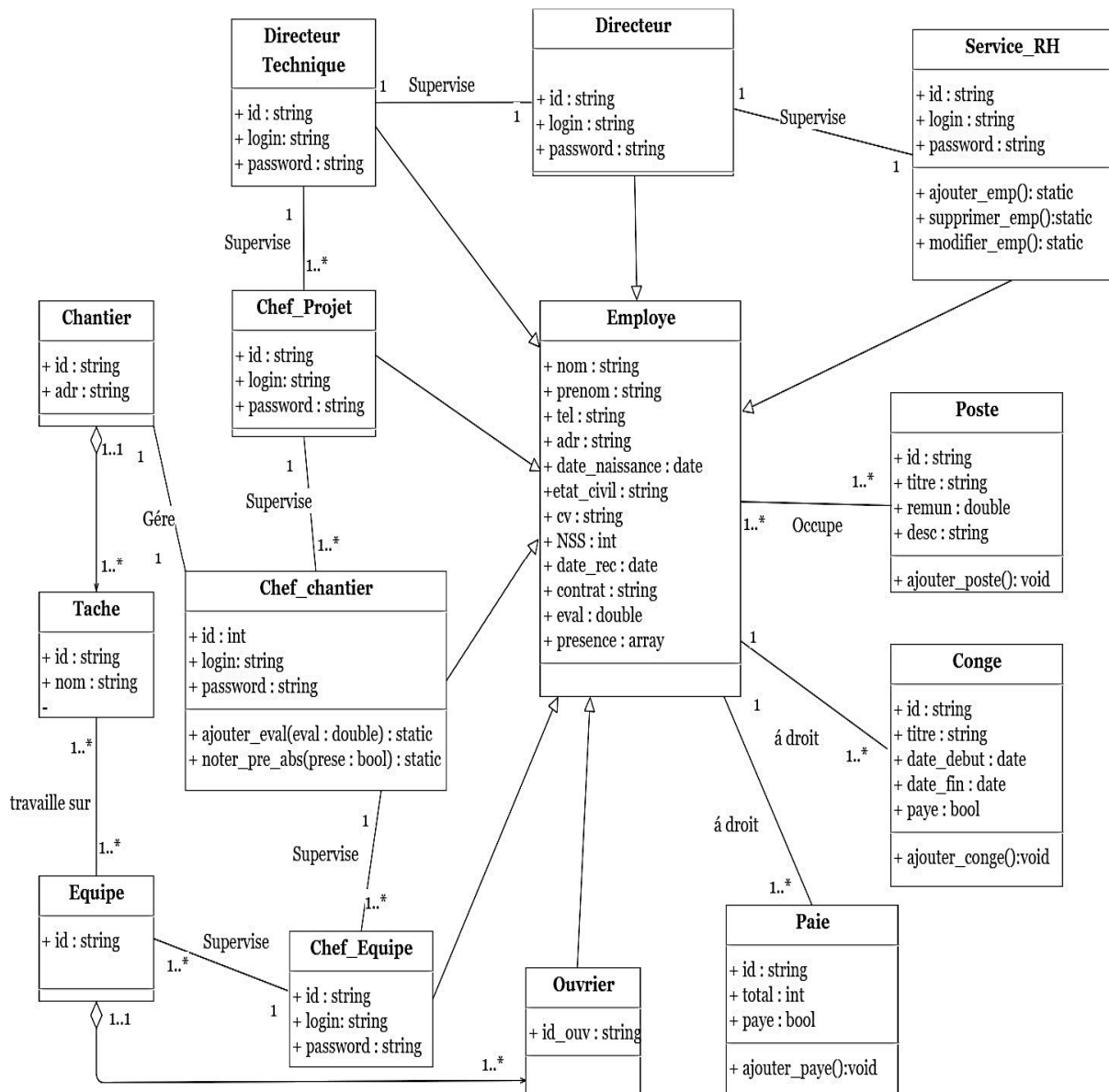


Figure 20 : Diagramme de Classes

## **Conclusion**

Pour abstraire la communication temps réel avec NodeJS On a opté pour l'utilisation de Socket.io qui utilise le protocole WebSocket. Et pour la gestion de base de données on a choisi MongoDB. Et pour le front end on a choisi React Native.

Avec l'achèvement de ce chapitre, nous avons eu la base de quoi commence la phase de développement, ou ce chapitre, nous servira de repère pour suivre le bon acheminement de la phase de réalisation, et savoir où mettre les pieds et garder trace de l'avancement de notre projet.

# Chapitre 4 : Réalisation

## Introduction

Dans cette partie, dédiée à la réalisation et à l'implémentation, nous allons commencer à définir les outils de développement utilisés pour la réalisation de notre application. On va ensuite passer à la présentation du travail réalisé dans notre application à travers des imprimés écran des interfaces développées.

## 4.1 Outils de développement

Nous énumérons les différents outils utilisés tout au long de ce projet pour la modélisation et la mise en place de notre application.

### 4.1.1 Environnement NodeJS :

NodeJS est un environnement d'exécution JavaScript open source utilisé pour exécuter du code JavaScript coté serveur. Le processus de développement de NodeJS a changé le paradigme selon lequel JavaScript est utilisé principalement côté client. C'est pourquoi NodeJS est devenu l'un des éléments fondamentaux du paradigme « JavaScript partout ». NodeJS a été créé par Ryan Dahl en 2009. Il est propulsé par le moteur V8 de Google, qui est le même moteur que celui utilisé dans le navigateur web Google Chrome. Cet environnement d'exécution n'a pas besoin d'un serveur web séparé ou d'autres dépendances, il existe des bibliothèques intégrées qui s'en occupent.

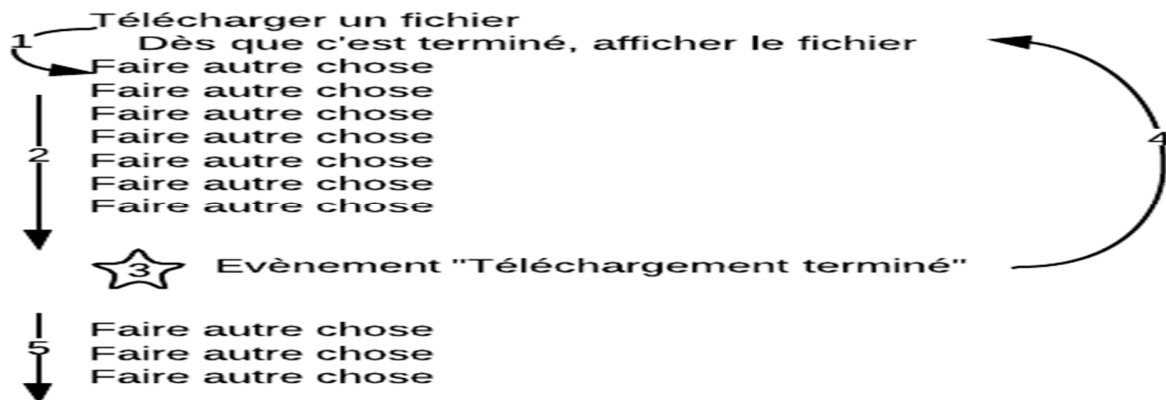
#### 4.1.1.1 Spécificités de NodeJS :

- Prend en charge le multithreading et le serveur intégré.
- Mise en cache des modules - les modules sont téléchargés et initialisés lorsqu'ils sont appelés pour la première fois, puis ils sont constamment disponibles.
- Le module Stream facilite le travail avec des fichiers volumineux.
- NodeJS prend en charge le développement Web en temps réel.
- Propose de différents Framework ce qu'il le rend riche

### 4.1.1.2 Les fonctions callback

NodeJS est un environnement de développement JavaScript basé sur les évènements. Ce que signifie qu'il y a un seul *thread* mais aucune opération n'est bloquante. Ainsi, les opérations un peu longues (chargement d'un fichier, téléchargement d'une page web, démarrage d'un serveur web...) sont lancées en tâche de fond et une fonction de *callback* est appelée quand l'opération est terminée.

Une **fonction callback** est une fonction de rappel passée en paramètre d'une autre fonction et qui sera appelée à la fin de cette dernière pour effectuer une autre opération. Plus clairement, utiliser un callback se résume à dire "j'exécute la fonction B une fois que la fonction A est terminée".



### 4.1.1.3 Gestionnaires de paquets

npm (Node Package Manager) et yarn (Yet Another Resource Negotiator) sont des gestionnaires de paquets pour la programmation en JavaScript, ils offrent la possibilité d'utiliser, de mettre à jour et de publier du code entre la communauté des développeurs dans le monde entier.

#### Spécificités de Yarn et Npm :

- La Stabilité : ces gestionnaires de paquets garantissent une stabilité pour l'application sur le long terme, ceci en gardant trace de toutes les dépendances de notre application (dans le fichier package.json).

- Une large documentation : npm et yarn offrent une large documentation qui propose des tutoriaux (sous forme audio-visuel ou manuscrit), et ils offrent des espaces (sites et blogs) pour les développeurs afin de partager leurs expériences et problèmes.
- Ouvertures au monde entier : ils offrent la possibilité à n'importe qui de partager sa création de n'importe où et à n'importe quel moment, ce qui assure l'innovation, la maintenance, et la mise à jour des paquets.

#### **4.1.1.4 Les Frameworks de NodeJS**

##### **Express.js**

Express.js, est un Framework qui s'occupe des protocoles, processus, etc. de bas niveau. Il est flexible car il existe de nombreux modules disponibles sur npm, qui peuvent être directement connectés à Express.

Contrairement à ses concurrents comme Rails et Django, qui ont une façon avisée de créer des applications, Express n'a pas de meilleure façon de faire quelque chose. Il est très flexible et enfichable.

##### **Sails.js**

Sails.js est un Framework NodeJS MVC (modèle – vue – contrôleur) qui suit le principe de la convention sur la configuration. Il s'inspire du Framework Web Ruby on Rails populaire qui permet de créer rapidement des API REST, des applications d'une seule page et des applications en temps réel (basées sur WebSockets). Il utilise largement les générateurs de code qui permettent de créer une application avec moins d'écriture de code - en particulier de code commun qui peut être autrement échafaudé.

Le cadre est construit sur Express.js, l'une des bibliothèques NodeJS les plus populaires, et Socket.io, une bibliothèque / moteur JavaScript pour ajouter des communications en temps réel, bidirectionnelles et basées sur des événements.

##### **Sails.js possède de nombreuses fonctionnalités intéressantes:**

- Il est construit sur Express.js.
- Il a un support en temps réel avec WebSockets.

- Il a une bonne documentation.

**Il existe actuellement quelques inconvénients importants, tels que:**

- Pas de prise en charge de la requête JOIN dans Waterline.
- Le développement est très lent.

#### **4.1.1.5 NodeJS comparé à d'autres langages ou environnements**

### **PHP**

PHP (préprocesseur hypertexte) est un langage de script côté serveur à usage général qui a été conçu à des fins de développement Web. PHP est alimenté par le moteur Zend, qui est installé sur des serveurs Web tels qu'Apache et Nginx.

#### **Spécificités de PHP :**

- Le code PHP peut également être intégré dans le balisage HTML ou HTML5. Cette flexibilité a contribué à la popularité du langage, PHP est désormais utilisé sur plus de 80% des serveurs web.
- Il est facile de configurer l'environnement de développement pour PHP.
- La majorité des services d'hébergement Internet prennent en charge PHP.
- PHP a une plus grande bibliothèque d'instructions et de questions fréquemment posées (il y en avait beaucoup en 20 ans).

### **Python**

Python est un langage de programmation, ce qui n'est pas le cas pour NodeJS, qui est un environnement d'exécution pour JavaScript.

Python est plus universel et un langage polyvalent, car les applications python vont bien au-delà de web.

## **Après la comparaison on conclut**

Comme pour le python, NodeJS ou PHP, le choix dépend des cas d'utilisation, car tout dépendra du type d'application qu'on veut concevoir et les critères qu'on privilégie (performance, utilisabilité, rapidité ...etc.) :

- Le python est le meilleur choix pour les développeurs non-expérimentés car il est plus universel.
- NodeJS est le meilleur choix pour développer des applications temps réel, comme les chats, des applications de paris, des applications comme Twitter ou des interfaces de messagerie instantanée.
- En ce qui concerne le PHP, il est plus pratique lorsque le projet contient les technologies CMS telles que WordPress, Drupal, Joomla, etc. Ou en cas d'une base de données relationnelle, optez pour PHP.

## **4.1.2 Les Frameworks JavaScript coté client**

### **4.1.2.1 React.js**

React.js est la bibliothèque JavaScript la plus populaire pour la création d'interfaces utilisateur, créée par Facebook. Nous pouvons créer des applications ou des sites Web modernes et rapides à page unique avec React.

### **Les composants de react.js**

Un composant est un bloc de code indépendant et réutilisable, qui divise l'interface utilisateur en plus petits morceaux. On peut créer une page Web ou une interface utilisateur à partir de nombreux petits blocs de code (composants). React a deux types de composants: fonctionnel et classe.

Composant fonctionnel : Un composant fonctionnel est essentiellement une fonction JavaScript (ou ES6) qui renvoie un élément React.

**Exemple :**

```
function Bonjour (props) {return <h1> Bonjour, {props.name} </h1> ;}
```

Composant classe : Les composants de classe sont des classes ES6. Ils sont plus complexes que les composants fonctionnels, notamment les constructeurs, les méthodes de cycle de vie, la fonction render() et la gestion des états (données).

**Exemple:**     Import React, {component} from 'react';

```
export default class ExampleComposant extends component {  
  
render(){ return (<div> exemple de composant classe </div> ); }  
  
}
```

#### 4.1.2.2 Vue.js

Vue.js (aussi appelé plus simplement Vue), est un Framework JavaScript open-source utilisé pour construire des interfaces utilisateur et des applications web monophages. Vue a été créé par Evan You et est maintenu par lui et le reste des membres actifs de l'équipe principale travaillant sur le projet.

### Fonctionnalités de vue.js

**Composants :**

Les composants Vue étendent les éléments HTML de base pour encapsuler du code réutilisable. Il s'agit donc d'éléments personnalisés auxquels le compilateur de Vue attache un comportement. Dans Vue, un composant est essentiellement une instance de Vue avec des options prédéfinies

**Réactivité :**

Vue est doté d'un système de réactivité qui utilise des objets JavaScript simples et un rendu optimisé. Chaque composant garde une trace de ses dépendances réactives pendant son rendu, de sorte que le système sait précisément quand et quels composants il faut ré-effectuer le rendu.

### **Transitions :**

Vue offre diverses façons d'appliquer les effets de transition lorsque des éléments sont insérés, mis à jour ou retirés du DOM.

### **Routage :**

Vue fournit une interface permettant de modifier ce qui est affiché sur la page en fonction du chemin d'accès de l'URL actuelle, quelle que soit la manière dont elle a été modifiée (par un lien envoyé par courrier électronique, un rafraîchissement ou des liens dans la page). De plus, l'utilisation d'un routeur frontal permet de faire une transition intentionnelle du chemin du navigateur lorsque certains événements du navigateur (c'est-à-dire les clics) se produisent sur des boutons ou des liens.

### **4.1.2.3 React Native**

React Native est un Framework JavaScript permettant d'écrire de vraies applications mobiles de rendu natif pour iOS et Android. Il est basé sur React, la bibliothèque JavaScript de Facebook pour la création d'interfaces utilisateur, mais au lieu de cibler le navigateur, il cible les plates-formes mobiles. En d'autres termes: les développeurs Web peuvent désormais écrire des applications mobiles qui semblent vraiment « natives », le tout dans le confort d'une bibliothèque JavaScript que nous connaissons déjà. De plus, React Native facilite le développement simultané pour Android et iOS.

Semblable à React pour le Web, les applications React Native sont écrites à l'aide d'un mélange de balisage JavaScript et XML, connu sous le nom de JSX. Puis, sous le capot, le « pont » React Native invoque les API de rendu natif en Objective-C (pour iOS) ou Java (pour Android). Ainsi, votre application sera rendue à l'aide de véritables composants d'interface utilisateur mobile, et *non de vues Web*, et ressemblera à n'importe quelle autre application mobile. React Native expose également des interfaces JavaScript pour les API de plate-forme, afin que vos applications React Native puissent accéder aux fonctionnalités de la plate-forme telles que la caméra du téléphone ou l'emplacement de l'utilisateur.

React Native prend actuellement en charge iOS et Android, et a également le potentiel de s'étendre aux futures plates-formes.

## Expo

Expo est un Framework et une plate-forme pour les applications React universelles. Il s'agit d'un ensemble d'outils et de services construits autour de React-Native et des plates-formes natives qui aident à développer, créer, déployer et itérer rapidement sur des applications iOS, Android et Web à partir de la même base de code JavaScript / TypeScript.

**Les spécificités d'Expo sont :**

- **Facile à prendre en main:** Il ne nécessite pas d'avoir des connaissances en développement natif pour démarrer avec expo. Il faut juste savoir coder en JS/TS et React.
- **Accessible:** Tout le monde peut démarrer avec Expo, il suffit juste d'avoir un ordinateur et une connexion internet.
- **Pas besoin d'Android Studio ou Xcode:** Les configurations de code natif sont déjà gérées.
- **L'application mobile EXPO Client:** C'est une application qui offre un rendu en temps réel ce qui permet de tester l'application sur un Smartphone en temps réel.
- **L'Expo cli (command line tool):** un ensemble d'instructions en ligne de commande qui aide à développer facilement et rapidement des projets, par exemple:
  - Démarrer un projet : **expo init**
  - Exécuter l'application : **expo start**
  - Publier l'application : **expo publish**
  - Création de l'application: expo **build:android**, **expo build:ios**, **expo build:web**

### 4.1.2.4 Ionic (Un alternative pour React Native)

Ionic est un SDK frontal open source pour le développement d'applications mobiles hybrides à l'aide de technologies Web telles que HTML, CSS et JavaScript. Il fournit des composants basés sur la technologie Web optimisée pour les mobiles ainsi que des API natives utilisant Cordova et Ionic Native.

Ionic avec sa dernière version, est performant en utilisant une manipulation DOM minimale. Angular joue également un rôle majeur dans l'augmentation des performances d'une application ionique.

Il a son propre outil d'interface de ligne de commande qui est vraiment utile pour échafauder et développer une application et surtout pour éviter d'écrire du code standard, ce qui permet de gagner un temps précieux.

## **4.1.3 Outils Temps Réel :**

### **4.1.3.1 Socket.io**

Socket.io est un module de NodeJS qui permet de créer des WebSockets, c'est-à-dire des connections bidirectionnelles entre clients et serveur qui permettent une communication en temps réel sur un autre protocole que le protocole http normalement utilisé dans les pages web. Il fonctionne sur toutes les plates-formes, navigateur ou appareil, en se concentrant également sur la fiabilité et la vitesse. Ce type de technique est utilisé pour créer des applications telles que des systèmes de communication en temps réel (i.e. des Chats), des jeux multi-utilisateurs, des applications de collaboration, etc.

Plus en détail, Socket.io permet de montrer le fonctionnement d'un protocole plus complexe par rapport au protocole http. Grâce à ce type de communication, il est possible de créer des applications pédagogiques, notamment en relation avec l'Apprentissage collaboratif.

De plus, la communication WebSocket renforce ultérieurement les concepts de programmation événementielle et d'interactivité, car pour mettre en place des applications qui se mettent à jour en temps réel il faut entremêler des événements côté-client et côté-serveur.

### **4.1.3.2 Alternatives pour Socket.io**

#### **SockJS**

SockJS est une émulation WebSocket. Il offre une API JavaScript cohérente et inter-navigateurs qui crée un canal de communication inter-domaines en duplex à faible latence entre le navigateur et le serveur Web, avec ou sans WebSockets.

Il suit un chemin de déclassement, cela signifie qu'il essaiera d'abord le meilleur protocole, puis retombera, si nécessaire, jusqu'à ce qu'il trouve un transport fonctionnel. Il est

utilisé conjointement avec le module SockJS-client côté client. Il comprend un certain nombre de transports de streaming plus rapides ainsi que des sondages.

## **SignalR**

SignalR est une nouvelle bibliothèque pour les développeurs ASP.NET qui facilite le développement de fonctionnalités Web en temps réel". SignalR permet une communication bidirectionnelle entre le serveur et le client. Les serveurs peuvent désormais transmettre instantanément du contenu aux clients connectés dès qu'il est disponible. Il prend en charge les Web Sockets et revient à d'autres techniques compatibles pour les navigateurs plus anciens. Il inclut des API pour la gestion des connexions (par exemple, les événements de connexion et de déconnexion), le regroupement des connexions et l'autorisation.

## **SocketCluster**

SocketCluster est un moteur temps réel ouvert et évolutif pour NodeJS. C'est un moteur de serveur HTTP + en temps réel rapide et hautement évolutif qui permet de créer des serveurs multiprocessus en temps réel qui utilisent tous les cœurs de processeur sur une machine / instance. Il supprime les limitations liées à l'exécution de votre serveur NodeJS en tant que thread unique et rend votre back end résilient en récupérant automatiquement des plantages des travailleurs et en agrégeant les erreurs dans un journal central.

### **Certaines des fonctionnalités offertes par SocketCluster sont :**

- Résilient à la fois sur le client et sur le back end - Les pannes de processus, les connexions perdues et autres défaillances sont gérées de manière transparente.
- S'adapte linéairement à mesure d'ajouter plus de cœurs de processeur et de travailleurs (en supposant un système d'exploitation décent) - Testé avec jusqu'à 16 cœurs.
- Analyses en temps réel - Envoyez des données aux clients qui sont représentées sous forme de compteurs, de graphiques ou de journaux en temps réel.

### 4.1.3.3 Protocole Pour Temps Réel

#### WebSocket

WebSocket est le nouveau protocole réseau créé par le W3C qui révolutionne la communication en temps réel entre clients et serveur. Il vise à développer un canal de communication bidirectionnelle et full-duplex sur un Socket TCP entre un client et un serveur.

#### Spécificités de WebSocket

- Un protocole efficace et API simple et pratique avec une petite bande passante (2 octets).
- Communication simultanée bidirectionnelle et Full duplex.
- Le temps de latence pour l'envoi et la réception d'un message est d'environ 50-100 ms. En fonction du temps de réaction des êtres humaines (perception, traitement, etc.), ce délai peut être considéré de facto du « real time » ce qui permet une vraie synchronisation entre les clients.
- L'échange entre client et serveur (ou entre serveur et client) se fait seulement s'il y a un événement qui propose un changement. Ceci réduit le trafic des requêtes/réponses à l'essentiel, tandis que dans le cas des applications qui utilisent le protocole http, souvent la plupart des requêtes asynchrones est faite juste pour contrôler s'il y a quelque chose de nouveau. Toutes ces requêtes ne sont pas nécessaires avec les WebSockets.

#### Différence par rapport au protocole http

Le protocole http est basé sur l'architecture requête/réponse qui peut être considérée comme la somme de deux mouvements unidirectionnels : la requête du client dans un sens et la réponse du serveur dans l'autre sens. Une fois que cet échange a eu lieu, il n'y a plus de contact entre le client et le serveur, si ce n'est à travers l'établissement d'un nouvel échange requête/réponse. Le protocole WebSocket, au contraire, crée une véritable connexion entre le client et le serveur, ce qui permet une communication bidirectionnelle en temps réel : le client peut envoyer un message au serveur, et le serveur peut envoyer un message à un client, sans la nécessité d'une requête/réponse http. À travers ce nouveau type de connexion, il est également

possible de créer un « groupe » de communication entre tous les clients qui ont établi une connexion WebSocket avec le même serveur.

## **Alternatives pour WebSockets :**

### **Polling**

Polling est une technique par laquelle le client demande régulièrement au serveur de nouvelles données. Le sondage peut être effectué en deux manières : short Polling et long Polling. En termes simples, short Polling est un temporisateur basé sur AJAX qui appelle à des délais fixes tandis que long Polling est basé sur Comet (c'est-à-dire que le serveur enverra des données au client lorsque l'événement du serveur se produit sans délai). Les deux ont des avantages et des inconvénients et sont adaptés en fonction du cas d'utilisation.

### **SSE**

SSE est un mécanisme qui permet au serveur de pousser de manière asynchrone les données vers le client une fois la connexion client-serveur établie. Le serveur peut alors décider d'envoyer des données chaque fois qu'un nouveau bloc de données est disponible. Il peut être considéré comme un modèle de publication-abonnement à sens unique.

Il propose également une API client JavaScript standard nommée EventSource implémentée dans la plupart des navigateurs modernes dans le cadre de la norme HTML5 par le W3C. Les poly fils sont disponibles pour les navigateurs qui ne prennent pas en charge l'API EventSource.

## **4.1.4 Les systèmes de gestion de base de données**

### **4.1.4.1 MongoDB**

MongoDB est un système de gestion de base de données open source (SGBD) qui utilise un modèle de base de données orienté document qui prend en charge diverses formes de données. C'est l'une des nombreuses technologies de bases de données non relationnelles qui ont vu le jour au milieu des années 2000 sous la bannière Non SQL pour une utilisation dans les applications de Big Data et d'autres travaux de traitement impliquant des données qui ne cadrent

pas bien dans un modèle relationnel rigide. Au lieu d'utiliser des tables et des lignes comme dans les bases de données relationnelles, l'architecture MongoDB est constituée de collections et de documents.

## **MongoDB Compass**

MongoDB, étant l'une des bases de données Non SQL les plus utilisées, fournit une GUI (Graphical User Interface), appelée MongoDB Compass. Compass permet d'analyser et de comprendre le contenu des données sans connaissance formelle de la syntaxe de requête MongoDB. En plus elle permet d'explorer les données dans un environnement visuel.

### **Spécificités de MongoDB Compass :**

Le schéma flexible et les documents de stockage JSON, fourni par MongoDB est un très bon outil pour les développeurs pour créer des applications sur la base de données, mais peut également fonctionner comme une sorte d'obstacle afin de comprendre la structure de la base de données. Mais avec MongoDB Compass, nous avons maintenant plusieurs façons de comprendre les données et de les manipuler sans aucune connaissance des commandes Shell.

On peut aussi citer d'autres avantages comme :

- Facile à utiliser, interface graphique simple.
- Création et exécution de requêtes.
- Analyse de performance.
- Index de surveillance.

### **Avantages de MongoDB**

- Comme les autres bases de données Non SQL, MongoDB ne nécessite pas de schémas prédéfinis et stocke tout type de données. Cela donne aux utilisateurs la flexibilité de créer n'importe quel nombre de champs dans un document, ce qui facilite la mise à l'échelle des bases de données MongoDB par rapport aux bases de données relationnelles.
- L'un des avantages de l'utilisation de documents est que ces objets sont mappés à des types de données natifs dans un certain nombre de langages de programmation. De plus,

le fait d'avoir des documents intégrés réduit le besoin de jointures de base de données, ce qui peut réduire les coûts.

- Une fonction essentielle de MongoDB est son évolutivité horizontale, ce qui en fait une base de données utile pour les entreprises qui exécutent des applications de Big Data. En outre, le partage permet à la base de données de distribuer des données sur un cluster de machines. Les nouvelles versions de MongoDB prennent également en charge la création de zones de données basées sur une clé de partition.
- MongoDB prend en charge un certain nombre de moteurs de stockage et fournit des API de moteur de stockage enfichables qui permettent à des tiers de développer leurs propres moteurs de stockage pour MongoDB.

## **Inconvénients de MongoDB**

- Avec sa stratégie de basculement automatique, un utilisateur configure un seul nœud maître dans un cluster MongoDB. Si le maître échoue, un nœud esclave se convertira automatiquement en nouveau maître. Ce commutateur promet la continuité, mais il n'est pas instantané - cela peut prendre jusqu'à une minute. Par comparaison, la base de données Cassandra Non SQL prend en charge plusieurs nœuds maîtres de sorte que si un nœud maître tombe en panne, un autre se tient prêt pour une infrastructure de base de données hautement disponible.
- Le nœud maître unique de MongoDB limite également la vitesse d'écriture des données dans la base de données. Les écritures de données doivent être enregistrées sur le maître et l'écriture de nouvelles informations dans la base de données est limitée par la capacité de ce nœud maître.
- Un autre problème potentiel est que MongoDB ne fournit pas une intégrité référentielle complète grâce à l'utilisation de contraintes de clé étrangère, ce qui pourrait affecter la cohérence des données.
- De plus, l'authentification des utilisateurs n'est pas activée par défaut dans les bases de données MongoDB, un clin d'œil à la popularité de la technologie auprès des développeurs.

#### 4.1.4.2 MySQL

C'est un système de gestion de base de données SQL Open Source le plus populaire, est développé, distribué et pris en charge par Oracle Corporation. Il gère Les bases de données relationnelles, ce qui offre un environnement de programmation flexible et respect tous les règles, de sorte qu'avec une base de données bien conçue, votre application ne voit jamais de données incohérentes, en double, orphelines, obsolètes ou manquantes.

Le serveur de base de données MySQL est très rapide, fiable, évolutif et facile à utiliser. Il a été initialement développé pour gérer de grandes bases de données beaucoup plus rapidement.

MySQL Server fonctionne dans les systèmes client / serveur ou embarqués. Le logiciel de base de données MySQL est un système client / serveur qui se compose d'un serveur SQL multithread qui prend en charge différents back-ends, plusieurs programmes et bibliothèques clients différents, des outils d'administration et une large gamme d'interfaces de programmation d'applications (API).

#### 4.1.4.3 PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle objet et généraliste, le système de base de données open source le plus avancé.

##### **Fonctionnalités de PostgreSQL :**

- PostgreSQL possède de nombreuses fonctionnalités avancées qu'offrent d'autres systèmes de gestion de bases de données d'entreprise, telles que :
- Types définis par l'utilisateur.
- Héritage de table.
- Mécanisme de verrouillage sophistiqué.
- Intégrité référentielle des clés étrangères.
- Vues, règles, sous – requête.
- Transactions imbriquées (points de sauvegarde).
- Contrôle d'accès simultané multi-version (MVCC).
- Réplication asynchrone.

## 4.1.5 Visual Studio Code

Visual Studio Code est un éditeur de code source léger mais puissant disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C ++, C #, Java, Python, PHP, Go) et des environnements d'exécution (tels que .NET et Unity).

## 4.1.6 Firebase

Firebase est une plateforme développée par Google, elle est dédiée à la création des applications web ou mobiles très performantes.

### **Avantages de l'utilisation de Firebase :**

- Créez de meilleures applications : Firebase offre les fonctionnalités de stockage (grâce à Cloud Firestore), de machine Learning, d'authentification et d'hébergement.
- Nous, nous avons optez, dans le cadre de notre travail, pour l'option de CloudFirestore a fin de stocker des fichiers volumineux puis récupérer les liens cloud vers ces fichiers pour les insérer dans la base de donnée.
- Améliorez la qualité des applications : Firebase offre les possibilités de suivi des performances de l'application, la distribution des versions d'essaye aux testeurs, et la récolte des rapports d'incidents chez les utilisateurs.

## 4.2 Fonctionnalités de l'application

Nous allons présenter les fonctionnalités de l'application, sous forme de captures d'écran des interfaces développées.

### 4.2.1 Authentification

Cette fonctionnalité garantit aux utilisateurs la sécurité. L'utilisateur introduit son email et mot de passe. Puis si ces données existent dans la base de données, il aura accès à son interface dédiée.

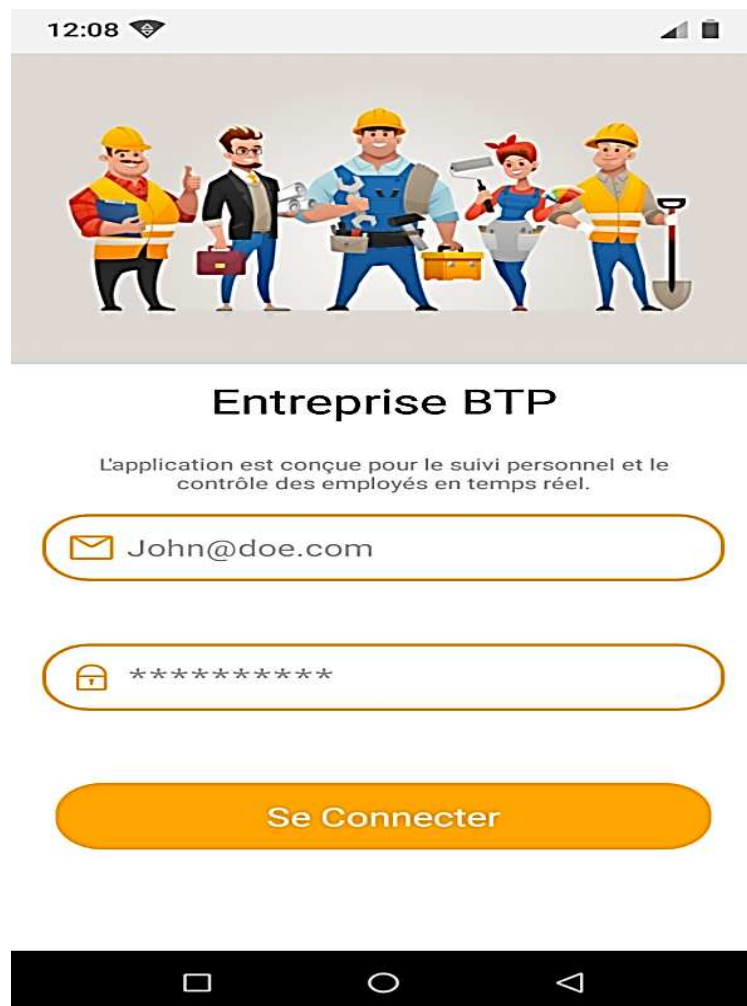


Figure 21 : Interface « Authentification »

## 4.2.2 Consultation de la liste des employés

Cette fonctionnalité offre aux utilisateurs la possibilité de consulter la liste de leurs employés ainsi que les informations de chaque employé :

- Le directeur de l'entreprise, le directeur technique, le chef de projet et le service ressources humaines ont accès à tous les employés de l'entreprise.
- Le chef de chantier aura accès aux employés de son chantier.
- Le chef d'équipe aura accès aux employés de son équipe.

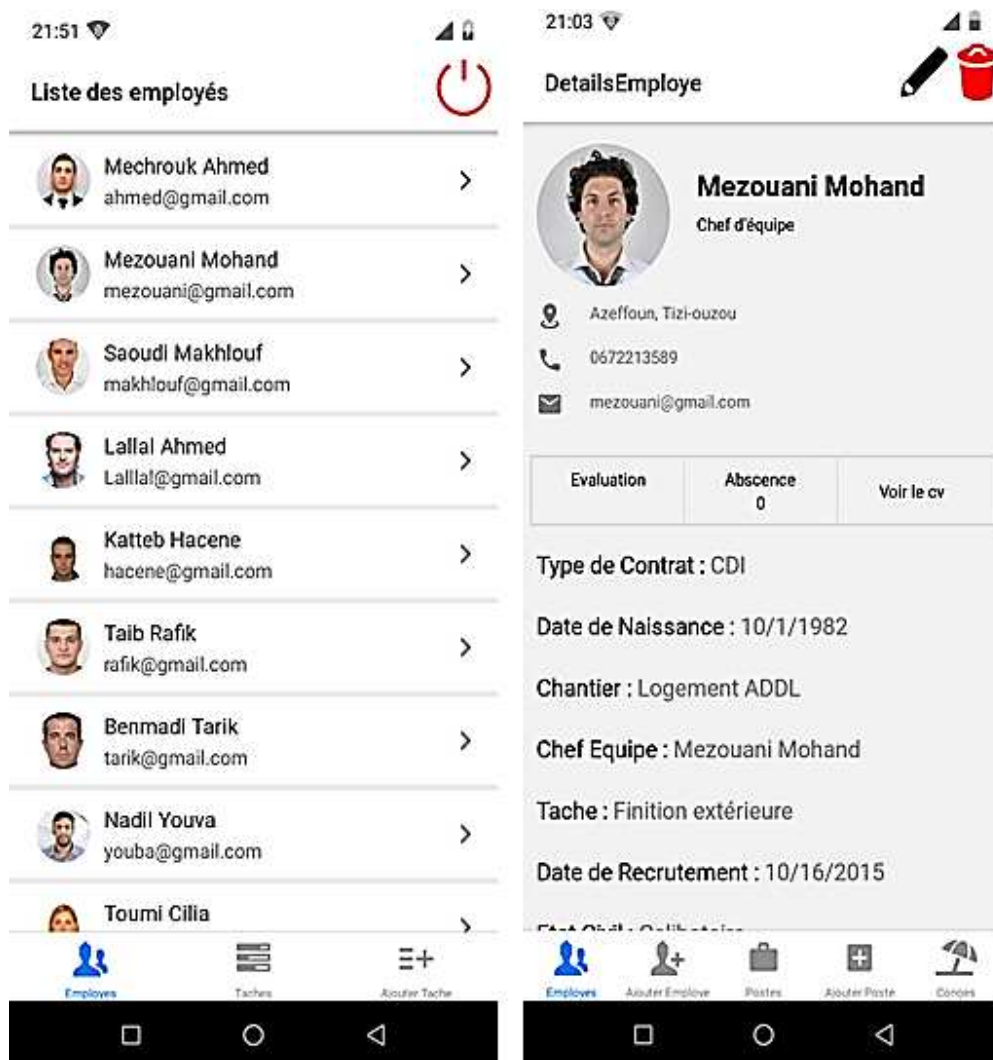


Figure 22 : Interfaces « Liste des Employés » et « Détails employé »

### 4.2.3 Consultation de la liste des postes

Cette fonctionnalité offre aux utilisateurs (Directeur de l'entreprise, directeur technique, chef de projet, service ressources humaines) la possibilité de consulter la liste des postes ainsi que les informations de chaque poste incluant la liste des employés occupant ce poste.

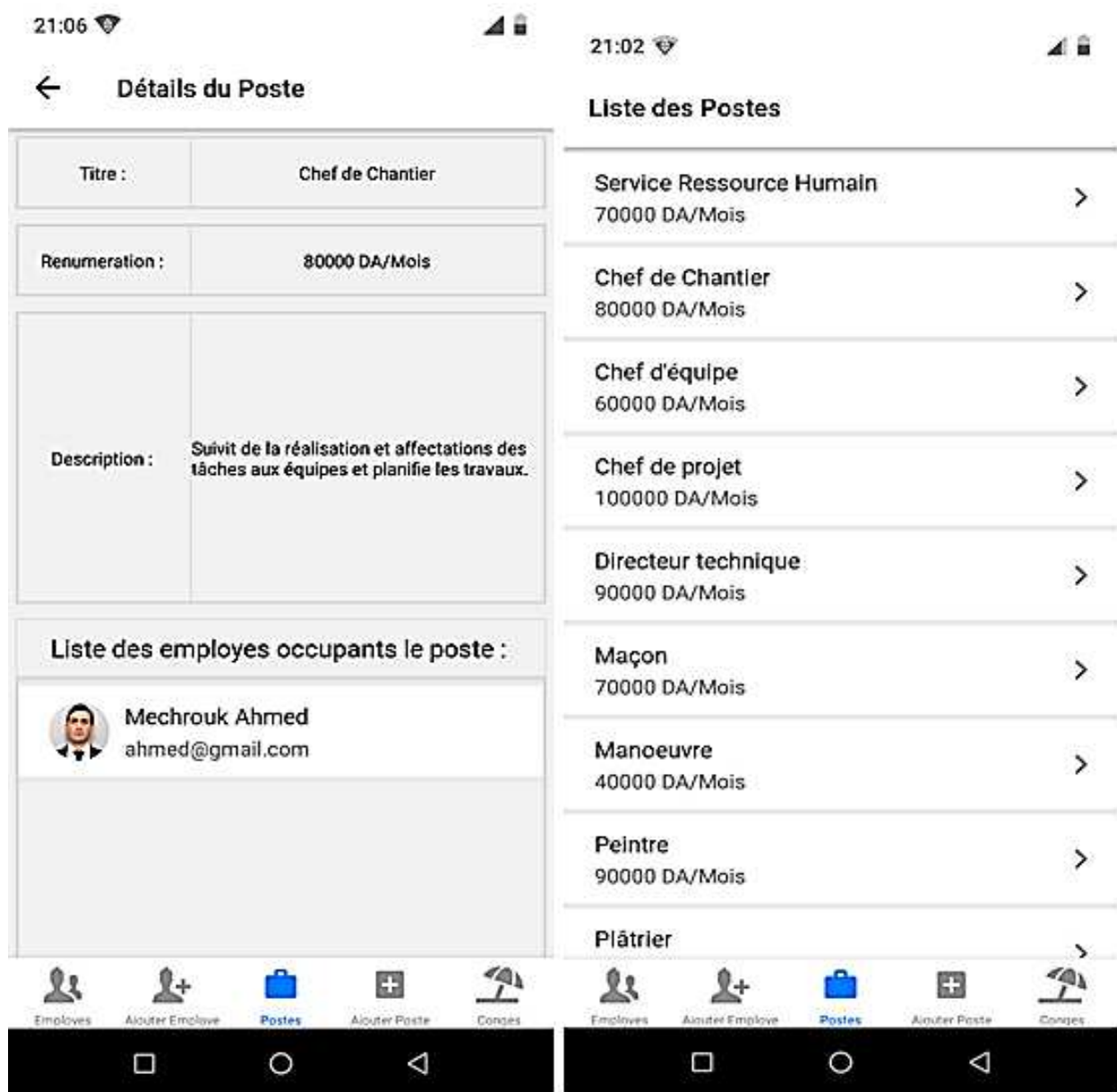


Figure 23 : Interfaces « Liste des Postes » et « Détails poste »

## 4.2.4 Consultation de la liste des tâches

Cette fonctionnalité offre au chef de chantier la possibilité de consulter la liste des tâches de chantier ainsi que les détails de chaque tâche et la liste des employés travaillons sur cette tâche.

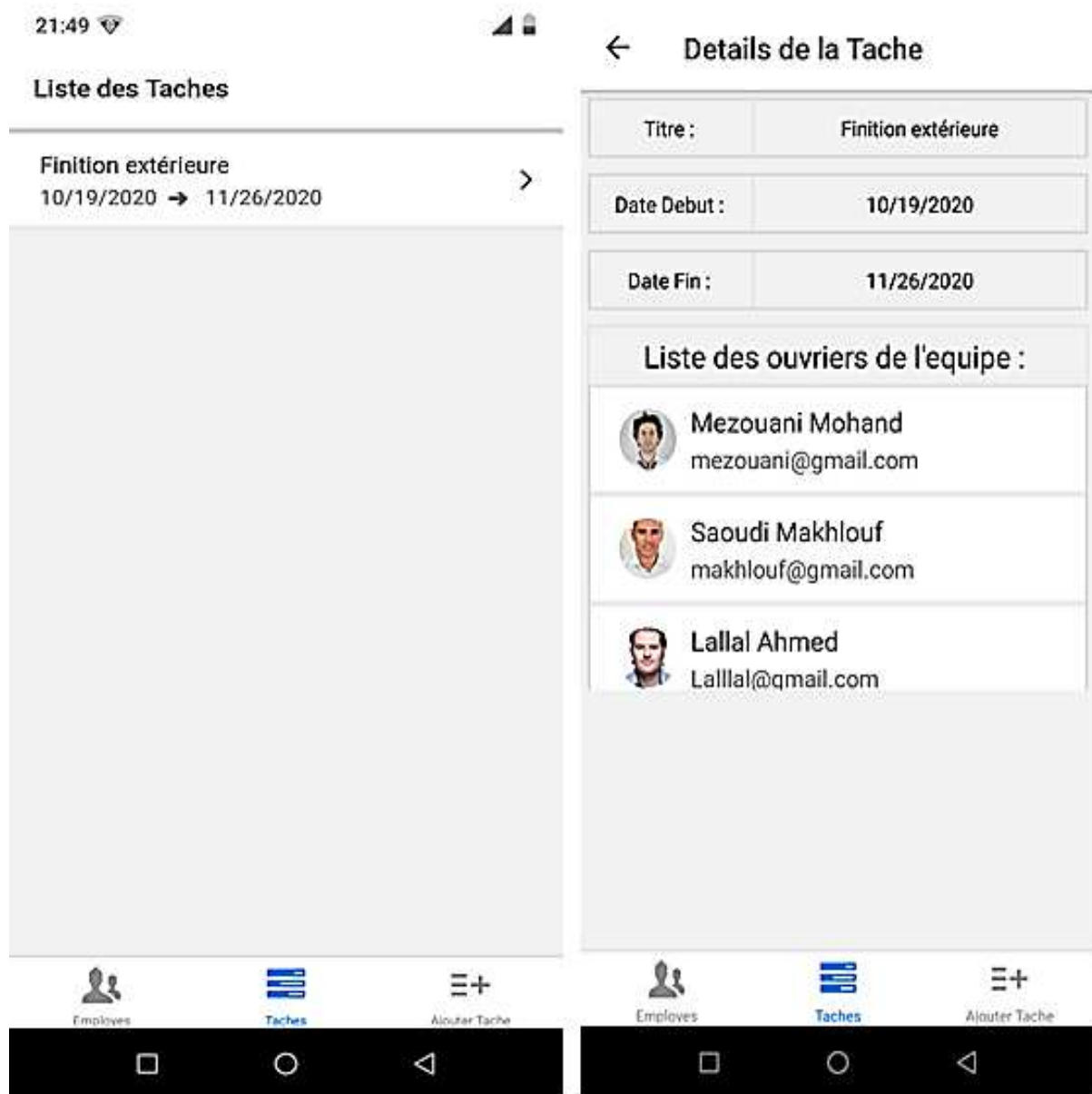


Figure 24 : Interfaces « Liste des tâches » et « Détails Tâche »

## 4.2.5 Consultation de la liste des congés

Le chef d'équipe peut consulter la liste des congés de chantier ainsi que les détails de chaque congé et la liste des employés qui ont pris ce congé.

Le service ressources humaines peut consulter la liste des congés insérés par les chefs d'équipes ainsi que les détails de chaque congé et la liste des employés qui ont pris ce congé, comme il peut confirmer ou annuler un congé de cette liste.

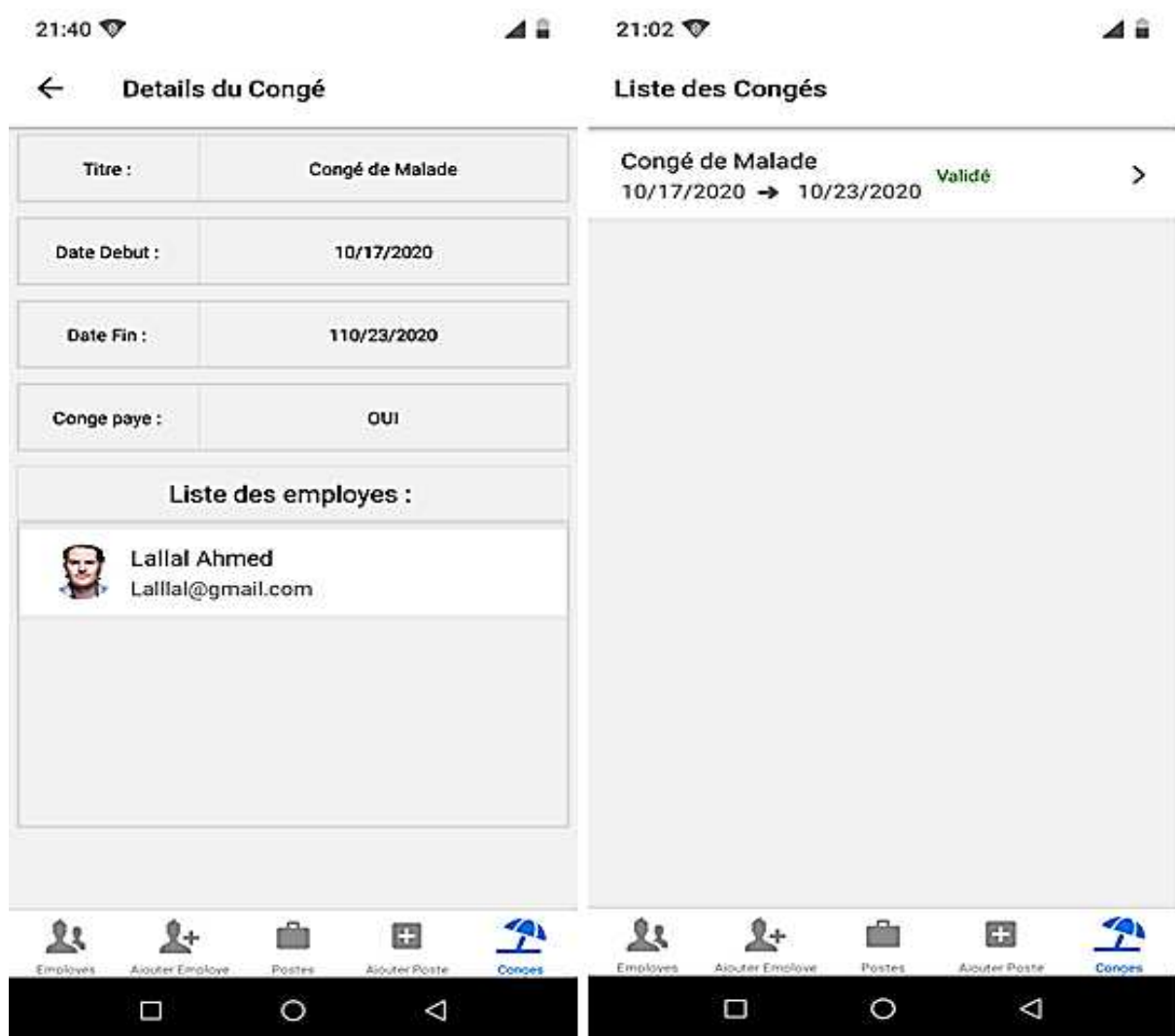


Figure 25 : Interfaces « Liste des Congés » et « Détails Congé »

## 4.2.6 Ajout d'un employé et ajout d'un poste

L'ajout d'un employé et l'ajout d'un poste sont les fonctionnalités de service ressources humaines.

The image displays two side-by-side screenshots of a mobile application interface. Both screens show the time 21:02 and signal strength icons at the top.

The left screen is titled "Ajouter un Poste" (Add a Job Post). It features three input fields: "Titre" (Title) with the value "Manoeuvre", "Renumiration" (Salary) with the value "50000", and "Description" (Description) with the placeholder text "Un poste pour ...". A prominent orange button at the bottom is labeled "Ajouter le poste".

The right screen is titled "Ajouter un Employé" (Add an Employee). It includes two buttons at the top: "Ajouter une photo" (Add a photo) and "Ajouter un CV" (Add a CV). Below these are input fields for "Nom" (Name) with the value "john", "Prenom" (First Name) with the value "doe", "Date de Naissance" (Date of Birth) with the value "10/24/2002", and "Email" with the value "johndoe@example.com".

At the bottom of both screens is a navigation bar with five icons: a person icon, a person with a plus sign, a briefcase, a plus sign, and an umbrella. The text labels below the icons are "Employes", "Ajouter Employe", "Postes", "Ajouter Poste", and "Conges". In the left screenshot, the "Ajouter Poste" icon and label are highlighted in blue.

Figure 26 : Interfaces « Ajouter un Employé » et « Ajouter un Poste »

## 4.2.7 Modifier ou supprimer un employé

Le service ressources humaines peut modifier les informations d'un employé comme il peut supprimer un employé de la base de données.

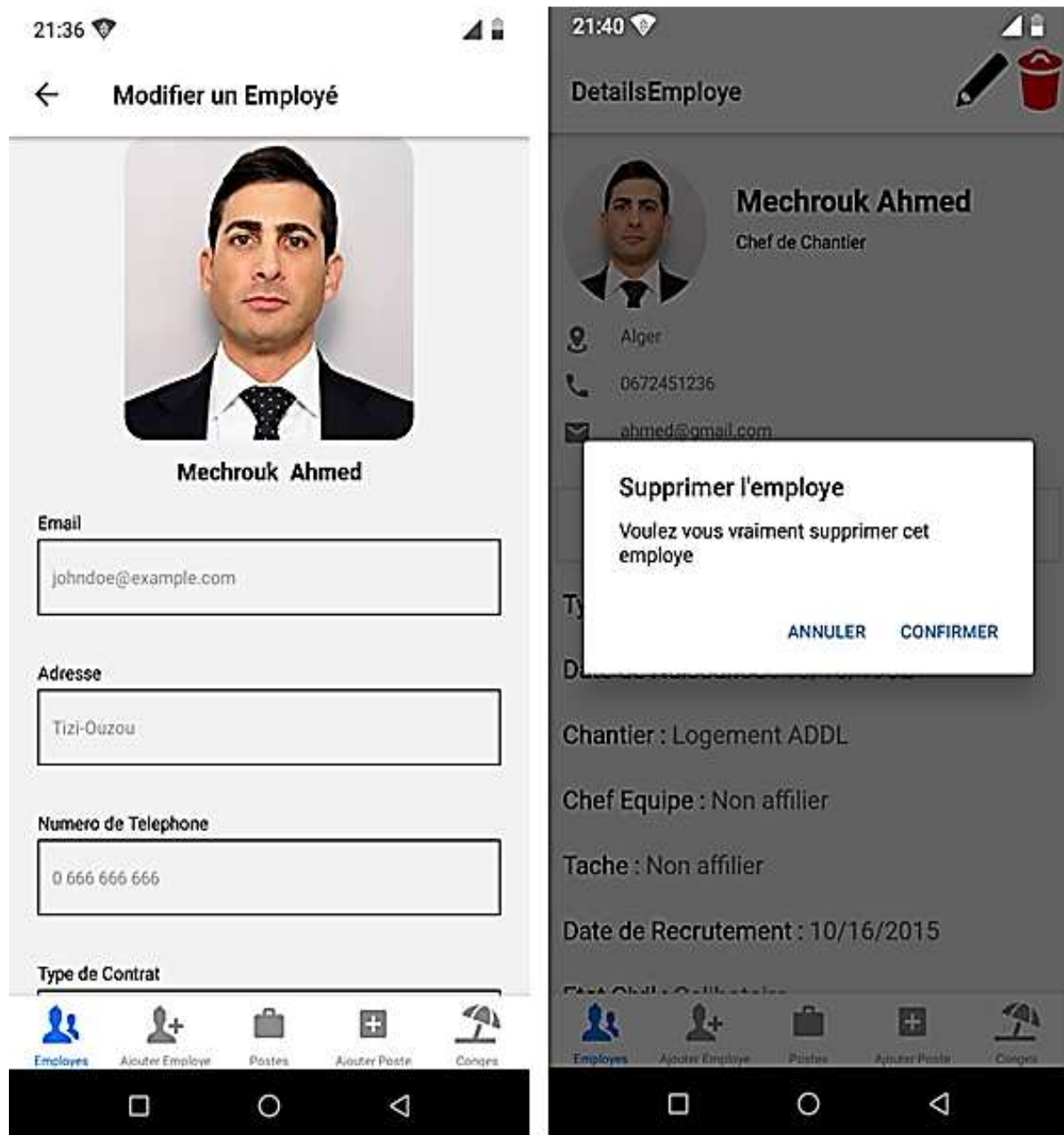


Figure 27 : Interfaces « Modifier un Employé » et « Supprimer un Employé »

## 4.2.8 Ajouter une tâche

Le chef de chantier peut un ajouter une tâche pour les employés de son chantier, les employés sélectionnés lors de l'ajout de cette tâche seront ajoutés à la base de données comme une équipe.

The screenshot shows a mobile application interface for adding a task. At the top, the status bar displays the time 21:49, a Wi-Fi signal, and battery level. The title of the screen is "Ajouter une Tache". Below the title, there are three input fields: "Titre" with the value "Tache 01s", "Date Debut" with the value "10/24/2002", and "Date Fin" with the value "10/24/2002". Below these fields is a list of three employees, each with a profile picture, name, email address, and a selection checkbox. The employees are: Mechrouk Ahmed (ahmed@gmail.com), Mezouani Mohand (mezouani@gmail.com), and Saoudi Makhoulf. At the bottom of the form is a large orange button labeled "Ajouter la tache". The bottom navigation bar contains three icons: "Employes", "Taches", and "Ajouter Tache" (highlighted in blue). The Android system bar is visible at the very bottom.

Figure 28 : Interface « Ajouter une Tâche »

## 4.2.9 Ajouter une absence ou une évaluation à un employé

Le chef de chantier peut évaluer les employés de son chantier selon le travail, la compétence, la ponctualité et la communication. Comme il peut marquer les absences des employés de son chantier.

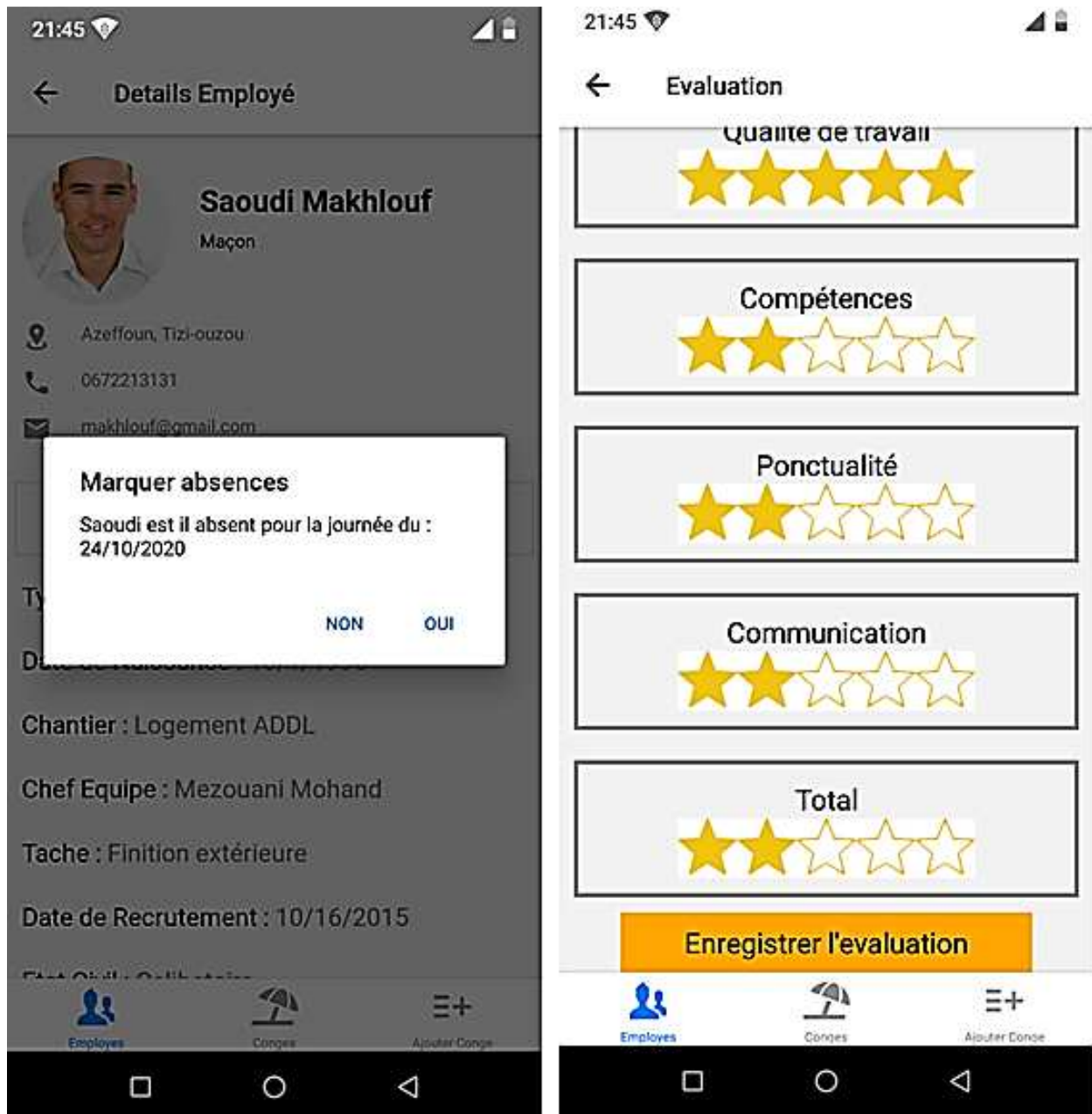


Figure 29 : Interfaces « Marquer Absence » et « Evaluer un Employé »

## 4.2.10 Ajouter un congé

Le chef d'équipe peut accorder un congé à un ou plusieurs employés de son équipe.

21:46

← Ajouter un Congé

Titre  
Conge de malade

Date Debut  
10/24/2002

Date Fin  
10/24/2002

Mezouani Mohand  
mezouani@gmail.com

Saoudi Makhoulf  
makhoulf@gmail.com

Lallal Ahmed

Paye?

Ajouter le conge

Employes Conges Ajouter Conge

Figure 30 : Interface « Ajouter un Congé »

## Conclusion

Dans ce chapitre, nous sommes passés de la théorie à la pratique, ça nous a permis d'atteindre nos objectifs décrits dans les étapes précédentes. Nous avons présenté les outils de développement qui nous ont aidés à concevoir notre application, puis on a présenté les fonctionnalités de l'application et les interfaces pour bien illustrer le travail qui a été fait.

## Conclusion générale

Afin de pouvoir mener notre projet de fin d'étude en bonne et due forme, nous avons dû établir un plan détaillé du projet en termes de durée et d'échéance. Nous avons entre autres dû bien cerner la problématique et cerner les objectifs à atteindre et ce, depuis le début de notre stage. Nous avons également défini les exigences des différents utilisateurs du système tout en étant très flexible en vue de changements futurs.

Les entreprises du secteur BTP sont en plein essor, elles ne cessent de croître et d'étendre leurs activités, et de ce fait, leurs écosystèmes ne cessent de changer entraînant avec lui un changement de leurs besoins qui attendent à chaque fois d'être comblés. C'est en partie dans cette perspective, qu'on nous a confié pour mission de concevoir et de réaliser un logiciel adéquate pour le suivi de personnels dans les chantiers.

Après avoir étudié minutieusement le processus de travail du métier de service ressources humaines et le suivi de personnels sur chantier on a pu capturer les besoins aussi bien fonctionnels que techniques, nous avons entamé l'étude conceptuelle de notre système, et enfin concrétisé tout cela par le développement effectif de la solution.

Notre système s'inscrit donc dans la future stratégie des entreprises du secteur BTP visant à informatiser et automatiser l'ensemble de leurs procédures métiers. Il s'assure notamment de la gestion du personnel depuis l'ajout d'un nouvel employé jusqu'à son affectation à une tâche dans un chantier définie passant par son évaluation et en fournissant les informations nécessaires pour le calcul de sa paie.

Cette expérience a été très enrichissante pour nous, elle nous a notamment permis de nous acclimater avec l'environnement de travail, à découvrir concrètement sur le terrain l'importance primordiale du logiciel dans les organisations. Nous avons par ailleurs pu développer nos compétences aussi bien sur le plan conceptuel et réalisation, nous avons appris à mener à terme un projet, à persévérer dans les moments difficiles mais essentiellement à toujours rebondir sur nos échecs.

## Bibliographie et Webographie

- PIECHOCKI Laurent UML en français définition d'UML du dictionnaire conception, 2005.
- **GILET. M, GILLET. P,** « SIRH: système d'information des ressources humaines », édition Dunod, 2010.
- <https://geainfolemans.github.io/SGBD/co/contexte.html#:~:text=On%20distingue%20de%20niveaux%20de,formalisant%20les%20flux%20d'informations.>

Consulter le 20 août 2020

- [https://fr.wikipedia.org/wiki/Progiciel\\_de\\_gestion\\_int%C3%A9gr%C3%A9](https://fr.wikipedia.org/wiki/Progiciel_de_gestion_int%C3%A9gr%C3%A9)

Consulter le 9 mai 2020

- <https://fr.wikipedia.org/wiki/Construction>

Consulter le 9 mai 2020

- <https://fr.wikipedia.org/wiki/Vue.js>

Consulter le 8 septembre 2020

- <https://expo.io/>

- <https://code.visualstudio.com/>

- <https://reactjs.org/>

- <https://reactnative.dev/>

- <http://nodejs.org/>

- <https://expressjs.com/>

- <https://www.mongodb.com/>

- <https://socket.io/>

- <https://firebase.google.com/>