

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Mouloud MAMMERI de TIZI-OUZOU**  
**Faculté de Génie Electrique et Informatique**  
**Département d'Informatique**



**Domaine : Mathématiques et Informatique**

**Filière : Informatique**

**Spécialité : Conduite des projets informatiques**

# **Conception et développement d'une application web de modèles de planning en ReactJS et NodeJS pour l'anticipation des besoins des grands projets puis l'ordonnancement des taches.**

**Encadré par :**

**Mr Ramdani Mohamed**

**Réalisé par :**

**Azzouz Wissam .**

**Taieb Zineb.**

**Membres du jury :**

**Mr Filali.**

**Mr Sadi.**

**Année universitaire : 2019/2020**

## **Remerciements :**

**Nous tenons à témoigner notre reconnaissance à toute personne qui a contribué de près ou de loin à la réalisation de ce mémoire.**

**Nous remercions vivement dans un premier temps notre encadreur de mémoire Mr Ramdani pour sa patience, sa disponibilité et surtout ses judicieux conseils.**

**Nous remercions également l'ensemble des membres du jury qui ont accepté d'évaluer notre travail.**

**Aussi, nous tenons à présenter nos sincères remerciements à nos deux familles ainsi que nos amis pour leur soutien constant et leurs encouragements.**

# Résumé

Le secteur BTP fait appel régulièrement à l'intervention d'une multitude de corps de métier durant différentes phases du projet, la maîtrise des différents volets des projets nécessite une bonne méthode de gestion telle que la planification.

L'objectif de ce projet est la conception, le développement et la mise en place d'une plateforme de gestion et de suivi des projets des entreprises du secteur BTP.

Le système à réaliser comporte plusieurs fonctionnalités dont : la gestion des différents plannings de l'entreprise, leur modification ainsi que leur consultation, la gestion des tâches qui nous permet d'insérer différentes tâches, les consulter et les modifier, l'étude des besoins en ressources humaines, matérielles et matériaux ainsi que leur gestion, le suivi temporel sous forme d'un calendrier qui permet à l'utilisateur d'avoir une vision globale sur le timing des tâches.

Pour la réalisation du projet nous avons suivi le cycle de développement en cascade. En ce qui concerne le développement nous avons opté pour la pile MERN qui utilise NODE comme environnement de développement accompagné de ses différents Framework entre autres REACT utilisé pour l'interface utilisateur, Express et MongoDB comme SGBD

Notre travail consiste à automatiser le processus de gestion des projets à l'aide du diagramme de GANTT qui permet de réaliser un planning.

# Abstract

The construction sector usually needs the intervention of a multitude trades during the project, the mastering of the different aspects of the projects requires a good management method such as planning.

The objective of this project is the conception, the development and the implementation of a management project and the monitoring platform for companies of the construction sector.

The implementation of the system includes several functionalities such as: the management of the various schedules of the company, their modification and their consultation, the management of the tasks which allows us to insert different tasks, consult them and modify them, the study of the needs in human and material resources as well as their management, time monitoring in the form of a calendar which allows the user to have a global vision of the timing of the tasks.

To implement the project we followed the cascade development cycle. We also opted for the MERN stack which uses NODE as a development environment that uses various frameworks including REACT used for the user interface, Express and MongoDB as DBMS

Our job is to automate the management project process with the GANTT chart used for planning.

## Liste des figures :

Figure 1: Architecture d'un ERP .....	26
Figure 2:Modèle en Cascade .....	36
Figure 3:Modèle en V .....	38
Figure 4:Diagramme de cas d'utilisation général .....	55
Figure 5:Diagramme de cas d'utilisation pour le cas «élaborer plannings" .....	58
Figure 6:Diagramme de cas d'utilisation pour les cas «Gérer plannings" .....	59
Figure 7:Diagramme de cas d'utilisation pour le cas «gérer les ressources » :.....	61
Figure 8:Diagramme de cas d'utilisation pour le cas «gérer les taches" .....	63
Figure 9:Diagramme de séquence cas "connexion" .....	65
Figure 10:Diagramme de séquence cas "Gérer les plannings" .....	67
Figure 11:Diagramme de séquence pour le cas "gérer les plannings des ressources" : .....	69
Figure 12:Diagramme de séquence cas "Modifier une tache" .....	71
Figure 13:diagramme de séquence cas: « Modifier une ressource » .....	72
Figure 14:Diagramme de séquence pour le cas "Suppression d'une tache".....	74
Figure 15:Diagramme de séquence pour le cas "suppression d'une ressource" .....	76
Figure 16:Diagramme de classe.....	79
Figure 17:Architecture MERN.....	82
Figure 18:Mappage d'objets entre Node et MongoDB géré via Mongoose. ....	84
Figure 19:Page d'accueil.....	86
Figure 20:Tableau de bord .....	87
Figure 21:Gantt .....	87
Figure 22:Liste des taches .....	88
Figure 23:Détails taches .....	89
Figure 24:Liste des ressources.....	89
Figure 25:Espace utilisateur .....	90

## Sommaire

Remerciements :	2
Résumé	3
Abstract	4
Liste des figures :	5
Introduction Générale	9
Chapitre1 : Le secteur du BTP	10
Introduction	11
1. Secteur BTP (Bâtiments et travaux techniques)	12
1.1. Définition :	12
2. Présentation de l'entreprise GTS développement :	13
3. Les activités de GTS Développement	13
4. Organigramme de l'entreprise GTS Développement	13
5. Le planificateur	19
5.1. Les niveaux d'intervention du planificateur :	20
5.2. Compétences et actions :	22
Conclusion :	22
Chapitre 2 : Technologies pour le BTP	23
Introduction	24
1. Les ERP	24
1.1. Définition :	24
1.2. Les ERP pour qui ?	25
1.3. Architecture détaillée d'un ERP :	25
1.4. Principe de fonctionnement:	26
1.5. Qu'est un moteur Workflow ?	27

1.6.	Les principaux éditeurs d'ERP : .....	28
1.7.	Les principaux ERP du marché: .....	28
1.8.	Avantages et inconvénients des ERP : .....	28
2.	Les Technologies de NodeJS .....	29
2.1.	Comparaison entre NodeJS et JavaEE : .....	29
2.1.	Finalement, Java ou Node.js ? .....	32
	Conclusion .....	32
	CHAPITRE 3 : Architecture logicielle .....	33
	Introduction .....	34
1.	Présentation du projet .....	34
2.	Démarche de développement .....	34
2.1.	Le cycle de développement en Cascade : .....	35
2.1.1.	Le fonctionnement du cycle de développement en cascade : .....	35
2.1.2.	Avantages et inconvénient : .....	36
2.2.	Le cycle de développement en V : .....	37
2.2.2.	Aventages et inconvénients : .....	39
2.3.	Choix du cycle de développement : .....	40
2.4.	Diagramme de contexte : .....	41
3.	Outils de conception .....	42
3.1.	Merise : .....	42
3.1.1.	Les niveaux de description et modèles associé : .....	43
3.2.	UML : .....	46
3.2.1.	Les différents types de diagrammes UML : .....	47
3.3.	Modélisation avec UML .....	50
3.3.1.	Diagrammes d'UML : .....	51
3.3.2.	Cas d'utilisation : .....	51
3.3.2.1.	Diagramme de cas d'utilisation général : .....	55
3.3.2.2.	Diagramme de cas d'utilisation pour le cas « élaborer plannings » : .....	56

3.3.2.3.	Diagramme de cas d'utilisation pour le cas « gérer les plannings ».....	58
3.3.2.4.	Diagramme de cas d'utilisation pour le cas « gérer les ressources» .....	60
3.3.2.5.	Diagramme de cas d'utilisation pour le cas « gérer les taches » : .....	62
3.4.	Diagrammes de séquence .....	64
3.5.	Diagramme de classes .....	77
	Conclusion.....	79
	CHAPITRE 4: Réalisation .....	80
	<b>Introduction</b> .....	81
1.	Définition de la technologie MERN stack .....	81
2.	Fonctionnement de la pile MERN :.....	82
2.1.	Niveau interface frontale React js : .....	82
2.2.	Niveau serveur Express.js et Node.js :.....	82
2.3.	Niveau de base de données MongoDB.....	83
3.	Node Package Manager (NPM): .....	84
4.	Cas d'utilisation de la pile MERN : .....	85
5.	Pourquoi choisir la pile MERN? .....	85
6.	Captures d'écran : .....	86
6.1.	Page d'accueil :.....	86
6.2.	Tableau de bord : .....	86
6.3.	Gantt : .....	87
6.4.	Liste des taches : .....	88
6.5.	Détails tache :.....	88
6.6.	Liste des ressources :.....	89
6.7.	Espace utilisateur : .....	90
	Conclusion.....	90
	Conclusion générale .....	91
	Bibliographie .....	92



## Introduction Générale

De nos jours, l'informatique est présente dans tous les domaines d'activités et devenue de plus en plus vaste et diversifiée.

C'est à ce titre que le domaine de construction des bâtiments n'échappe pas à la règle et l'ensemble des services et départements faisant partie vivent une informatisation progressive.

Dans ce cadre, notre projet de fin d'études a été axé sur la réalisation d'un planning d'organisation des chantiers ayant pour objectif principal l'automatisation du processus de gestion et de suivi de l'avancement des projets en temps réel.

Notre mémoire est structuré comme suit :

- Le premier chapitre définit le contexte général du projet et présente le secteur du BTP.
- Le deuxième chapitre est consacré aux technologies qui accompagnent le BTP à savoir les ERP .
- Le troisième chapitre est entièrement consacré à la conception détaillée du projet. Il contient la démarche de développement ainsi l'ensemble des diagrammes nécessaires à la bonne compréhension et cohésion du système. Du diagramme d cas d'utilisation au diagramme de classe, le tout est accompagné des scénarios et descriptions nécessaires.
- Le dernier chapitre, donne une vue globale de la solution et détaille la réalisation de l'application tout en proposant des aperçus de celle-ci.
- Pour terminer, une conclusion permettant de faire une synthèse du travail réalisé.

# **Chapitre1 : Le secteur du BTP**

## **Introduction**

ce premier chapitre décrit une recherche menée dans le domaine de construction des bâtiments et travaux publics, plus fréquemment connu sous le nom du secteur BTP, que nous allons présenter en nous appuyant sur un exemple d'organigramme de l'entreprise GTS tout en spécifiant le rôle de chacun des acteurs intervenants , plus particulièrement le rôle du planificateur.

## **1. Secteur BTP (Bâtiments et travaux techniques)**

### **1.1. Définition :**

La construction est le fait d'assembler différents éléments d'un édifice en utilisant des matériaux et des techniques appropriées.[4]

Le secteur du bâtiment et des travaux publics (BTP) regroupe toutes les activités de conception et de construction des bâtiments publics et privés, industriels ou non, et des infrastructures telles que les routes ou les canalisations. [2]

Les projets sont cependant, généralement conçus et menés par un « maître d'œuvre » (architecte, bureau d'études), pour le compte du client, appelé « maître d'ouvrage ». Pour des chantiers de grande ampleur ou complexes, plusieurs intervenants peuvent se partager les tâches de maîtrise d'ouvrage ou de maîtrise d'œuvre. [2]

La construction elle-même est effectuée généralement par une ou plusieurs entreprises (on parle alors d'un « groupement d'entreprises »), responsable(s) du marché de travaux. Cette entreprise appelée souvent « entreprise générale » fait également appel à de la sous-traitance. [2]

**La sous-traitance :**Est le fait de déléguer une ou plusieurs personnes spécialistes .Dans le secteur du BTP, la sous-traitance est le fait de faire appel à une ou à un groupe de personnes externes à l'entreprise principale afin d'effectuer une tâche non couverte par l'entreprise en gardant la responsabilité à cette dernière et en respectant le cahier des charges et les exigences de celle-ci. Le sous-traitant n'a pas de contact direct avec le client. [2]

## **2. Présentation de l'entreprise GTS développement :**

GTS est une SARL (Société à responsabilité limitée) de génie civil et de bâtiments localisée à Tizi Ouzou . Elle dispose d'environ 30 engins lourds et matériels stratégiques, et elle compte environ 80 salariés.

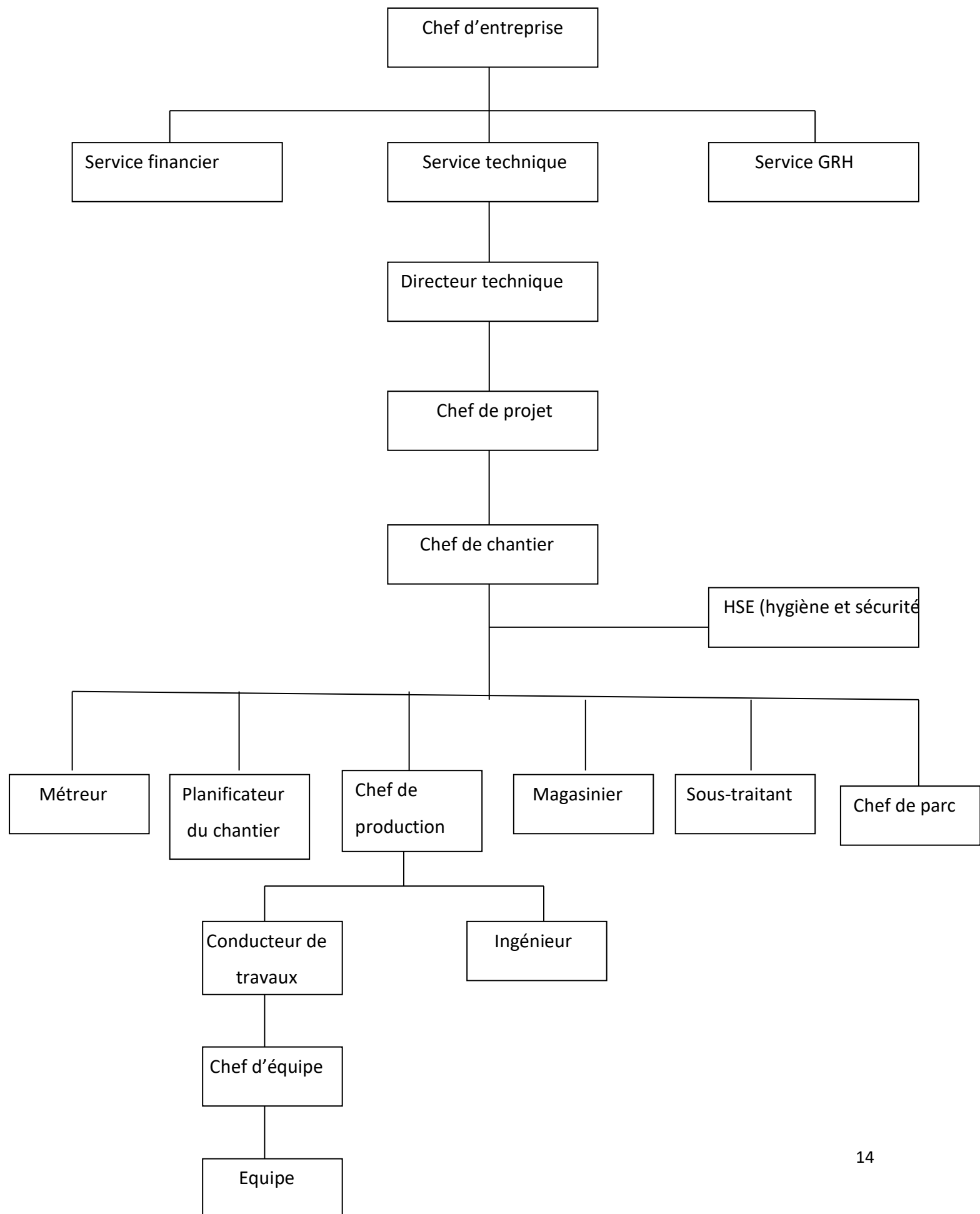
## **3. Les activités de GTS Développement**

Les activités de GTS Développement se résument en :

- Activités de terrassement.
- Activités de génie civil industriel.
- Activités de bâtiments.
- Activités de construction de charpentes métalliques légères.

## **4. Organigramme de l'entreprise GTS Développement**

**Organigramme :**



## **Les différents rôles des acteurs intervenants :**

- 1. Chef d'entreprise :** Il chapote la hiérarchie de l'entreprise. Il est chargé de fournir les matériaux, le matériel ainsi que la main d'œuvre nécessaire à l'exécution des travaux. Il est censé mener à terme les travaux tout en respectant la réglementation du code du travail voire les conditions du contrat qu'il a signé, à savoir le délai.
- 2. La direction technique :** Composée du directeur technique et ses collaborateurs. Ces derniers sont des techniciens ou des ingénieurs en génie civil.

Le directeur technique : En tant qu'expert technique, le directeur technique BTP contribue au positionnement commercial de l'entreprise. Il intervient en amont dans les phases de prospection et de réponse aux appels d'offre. Puis, dans la phase d'exécution des travaux, il encadre les équipes importantes (ingénieurs d'étude, économistes de construction).

Il accomplit les tâches suivantes :

- Réaliser et superviser les études techniques et coordonner les travaux.
- Responsable de la qualité des réalisations, du suivi budgétaire et des résultats financiers des chantiers.
- Former les personnes sous sa responsabilité aux meilleures pratiques et techniques d'étude et d'exécution des chantiers.
- Assurer le montage des dossiers de soumissions des appels d'offres.
- Participer aux réunions techniques et à la sélection des sous-traitants.
- Recruter, encadrer et animer les équipes.
- Assister les équipes dans l'organisation de leur travail et des priorités.
- Organiser les études et remettre régulièrement des rapports.

Dans le cas où l'entreprise n'emploie pas un chef de projet, un chef de chantier ou un conducteur de travaux, c'est au directeur technique ou ses collaborateurs d'assumer leurs tâches.

- 3. DRH :** (Direction des ressources humaines), le directeur ou le gestionnaire des ressources humaines agit sur le recrutement, le licenciement ainsi que la formation du personnel de l'entreprise. Il est chargé également du management social et de la communication interne.

Les missions principales du GRH sont :

- Recrutement et dotation en personnel.
- Evaluation et amélioration du rendement.
- Administration de la rémunération et des avantages sociaux.
- Gestion des carrières dans l'entreprise en valorisant la mobilité et la formation.
- Sensibilisation des employés aux impératifs d'hygiène et de sécurité.

- 4. Le chef de projet :** C'est un technicien (ou un groupe de techniciens) qui appartient au service technique de l'entreprise. Il est chargé de la supervision et de la réalisation des études liées à la construction d'un ouvrage réalisé par l'entreprise. Il se porte garant de la faisabilité technique du projet, du respect des délais et coûts de construction, ainsi que des normes environnementales et de sécurité.

Missions principales\_:

- Coordonner les actions des différents intervenants (les différents services, les fournisseurs et les sous-traitants).
- Participer aux études de conception et optimiser les solutions techniques.
- Etablir le cahier des charges.
- Organiser et manager l'équipe du projet.
- Structurer, mobiliser et piloter les ressources (humaines, budgétaires...).

- 5. Conducteur des travaux :** C'est un salarié de l'entreprise en charge de l'exécution des travaux. Il assume la responsabilité d'un ou de plusieurs chantiers sur les plans



financiers, commerciaux, achats, juridiques, techniques et humains. C'est le pivot de l'organisation et de l'exploitation dirigée par les chefs de chantiers.

Il accomplit les missions suivantes :

- Il élabore le devis matière prévisionnel faisant ressortir les besoins en personnel, matériel, matériaux avec leurs coûts nécessaires à l'exécution des travaux.
- Il élabore le planning d'exécution des travaux.
- Il dresse le planning d'utilisation du matériel et du personnel (ordre d'acheminement du personnel et du matériel) conformément au planning d'exécution.
- Il gère toutes les démarches administratives pour le démarrage du chantier.
- Il prépare et organise le chantier.
- Il identifie les problèmes techniques de terrains et les résout ou les transmet à son chef pour résolution.
- Il établit les attachements en vue de l'établissement des décomptes.
- Il assure le suivi budgétaire du chantier (bonne exécution du devis).
- Il supervise les activités des autres intervenants sur le chantier (chef de chantier).

**6. Le chef de chantier :** C'est l'homme des travaux. Il joue le rôle de charnière entre la main d'œuvre ouvrière et les intellectuels du siège. Il est chargé de :

- répartir les tâches aux chefs d'équipes.
- établir les rapports hebdomadaires et journaliers avec l'aide du commis pointeur.
- contrôler et vérifier la bonne exécution des tâches confiées aux chefs d'équipes.

- gérer le personnel et le matériel sous sa responsabilité.
- mettre en application les règlements d'hygiène et de sécurité sur le chantier.
- assurer l'organisation, le commandement et l'exécution des travaux en respectant les règlements en vigueur et le programme établi.
- Prévoir les besoins du chantier en main d'œuvre, matériaux, matériel et outillage.
- assurer l'implantation des ouvrages et les tracer.

**7. Le chef d'équipe** : Ouvrier qualifié possédant la maîtrise de son métier. Il est chargé de conduire une équipe suivant les directives données par les agents de maîtrise.

Dans le cas où une équipe ne possède pas un chef, c'est le supérieur direct qui la prend en charge.

**8. Le manœuvre** : personnel de simple exécution, sans responsabilité, effectuant des tâches auxiliaires, ne nécessitant pas de spécialisation ou d'adaptation préalable: rangements, manutention manuelle de terre et chargement sur camions, nettoyage ordinaire, gardiennage...etc.

**9. L'ouvrier spécialisé** : personnel effectuant des travaux simples nécessitant une spécialisation ou une adaptation préalable, sans initiative particulière ni formation professionnelle. Exemple : dosage selon les instructions reçues, fabrication à la bétonnière usuelle des bétons et mortiers couramment utilisés...etc.

**10. L'ouvrier qualifié** : personnel effectuant des travaux pouvant appeler des initiatives et nécessitant des connaissances acquises par formation professionnelle ou pratique équivalente. Exemple : maçonnerie, coffrage, carrelage, assemblage et façonnage d'armatures...etc.

**11. Le sous-traitant** : Désigne toute personne morale chargée par l'entrepreneur de réaliser une partie des travaux. L'entrepreneur doit obtenir l'accord préalable du Maître d'ouvrage avant de formaliser un contrat de sous-traitance. Il ne peut sous-

traiter la totalité de son marché. Les documents administratifs du marché fixent parfois le pourcentage maximum du contrat qu'il est possible de sous-traiter.

Le personnel de l'entreprise est classé en :

- Personnel permanent : qui a un contrat de durée indéterminée (CDI).
- Personnel contractuel : qui a un contrat de durée déterminée (CDD).

## **5. Le planificateur**

### **La planification :**

On peut définir la planification comme l'organisation du déroulement des différentes étapes d'un projet.

Parmi les différentes phases d'un projet, on trouve le découpage, qui consiste, comme son nom l'indique, à le découper en tâches. Une fois la liste exhaustive des tâches établie, elles vont devoir être ordonnancées de façon à déterminer les interdépendances et l'ordre dans lequel elles devront être réalisées. Elles vont être priorisées en fonction de leur criticité, des charges nécessaires à leur réalisation et de leurs éventuelles dépendances. [5]

Une organisation des tâches va ressortir de ce travail, qui va permettre de définir des jalons de réalisation, qui une fois intégrés au tableau de bord de suivi de projet, permettront de surveiller son avancement. On obtient ainsi une vision claire et synthétique du projet, permettant d'anticiper les éventuels dépassements de délais et les ressources nécessaires à la réalisation des différentes tâches. [4]

Le planificateur effectue un travail essentiel dans la préparation d'un chantier. Son intervention se situe à différents niveaux :

- la préparation administrative et juridique.
- la construction de différents plannings et calendriers pour la réalisation du chantier.

L'objectif est de garantir une organisation sans failles permettant de contrôler et d'optimiser l'utilisation du temps et des ressources disponibles afin de respecter les délais impartis.

L'essentiel du travail du planificateur s'effectue au bureau. Il est en relation avec toute une série d'interlocuteurs en interne (chef de projet, métreur, chef de chantier, acheteur, coordinateur sécurité, diviseur) et en externe (administrations, assureur, sous-traitants, pompiers, etc.).

En termes d'hierarchie, le planificateur travaille sous les ordres du chef de projet. Néanmoins, cette fonction n'existe telle quelle que dans des grosses entreprises de construction et généralement, plusieurs employés effectuent les tâches de préparation et planification sous la conduite d'un responsable de service. [4]

### **5.1. Les niveaux d'intervention du planificateur :**

#### **Phase Avant-projet:**

Un organisme, privé ou publique, propose des chantiers en appel d'offres pour la réalisation d'une prestation dans le but de mettre en concurrence plusieurs entreprises du BTP et ainsi de trouver la meilleure pour réaliser le projet. Pour sélectionner celle qui correspond le mieux à sa demande, le commanditaire (maitre d'ouvrage) juge les entreprises sur plusieurs critères comme le délai de réalisation prévu ou le budget prévisionnel.

L'entreprise intéressée par l'offre prépare et rédige un dossier de réponse contenant deux parties : la candidature de l'entreprise et la réponse.

- La candidature : contient toutes les informations concernant l'entreprise, ses activités, ses points forts, ses domaines d'expertises, qui composent l'équipe, Quelles prestations est-elle habilitée à réaliser ? possède-t-elle les compétences nécessaires pour répondre efficacement à cet appel d'offres ? Quelles sont ses références ?
- La réponse : étant un dossier détaillé en indiquant la stratégie pour y répondre, le budget prévisionnel ainsi que le délai estimé. C'est dans la rédaction de cette réponse que le planificateur intervient pour la 1ère fois. [1]

### **Phase de préparation du projet :**

Lorsque le projet est accepté par le maître d'ouvrage, la phase de préparation débute.

Sur la base des différents documents constituant le dossier (cahier des charges, plans, fiches techniques, etc.) et après l'analyse de ces derniers, le planificateur met en route la préparation administrative et juridique. Il effectue différentes démarches auprès des différents interlocuteurs.

Tout d'abord, il introduit auprès des autorités compétentes les demandes de permis et suit le dossier jusqu'à obtention de ces dernières. Il participe à la concertation auprès des assurances (contrats, primes, etc.). Il intervient dans les concertations auprès des autorités communales (autorisations concernant l'occupation de la voie publique), auprès de services d'incendie (plans de sécurité) et enfin auprès des sociétés agréées en matière de recyclage et d'élimination des déchets de construction (plan environnemental). Durant cette phase de préparation, il collabore avec l'architecte, le bureau d'étude, le coordinateur de sécurité et bien entendu il travaille sous la conduite du chef de projet. [1]

### **Phase de Planification :**

Dans la phase de planification des tâches, le planificateur élabore tout d'abord le plan de mise en œuvre (les différentes étapes) sur la base des plans, du cahier des charges, du métré, du contrat et du budget. Pour réaliser ceci, le planificateur doit maîtriser les processus et les techniques de la construction. Ce travail effectué va lui permettre de dégager le planning général. En collaboration avec le métreur, il affine le planning pour constituer les plannings du personnel, des besoins matériels et en matériaux. Ces derniers seront soumis aux sous-traitants pour acceptation ou modification. En collaboration avec l'acheteur, il élabore le planning des achats (quand et comment commander afin d'être livré à temps). Avec le service opérationnel (chef et conducteurs de chantier) et le coordinateur de sécurité, il réalise le planning d'aménagement du chantier.

Durant la réalisation, le planificateur suit et contrôle le respect des délais et des échéances. Il est susceptible d'apporter des modifications à son travail en réaction à des événements inattendus (problèmes techniques, retards de livraisons, etc.).

Cette fonction spécifique n'existe qu'au sein des grosses entreprises de construction. Dans ce cas, le service ou le département de planification s'occupe uniquement de cette tâche. Dans des structures plus modestes, la fonction de planificateur est menée de front avec d'autres tâches et elle est prise en charge par le diviseur, le chef de projet ou de chantier ou la direction de l'entreprise. [1]

### **5.2. Compétences et actions :**

- Connaître le processus technique de la construction.
- Maîtriser les méthodes de planification et les logiciels spécifiques en la matière.
- Pouvoir analyser un cahier des charges, des plans...
- Connaître les normes et procédures auprès des autorités compétentes en matière de construction.
- Connaître le marché des assurances.
- Maîtriser les normes de sécurité.
- Posséder des compétences commerciales et de négociation.
- Maîtriser des logiciels de bureautique. [5]

### **Conclusion :**

Vu la complexité du secteur BTP, nous sommes amenés à apporter une solution informatique plus optimale qui permet à la fois une gestion rigoureuse des plannings de travail ainsi que les ressources dont l'entreprise dispose.

## **Chapitre 2 : Technologies pour le BTP**

## Introduction

Connu pour être un secteur particulièrement dynamique, la gestion d'une entreprise de BTP est parfois difficile au vu du nombre important de ses intervenants. Aussi, ce secteur est particulièrement exposé à une multiplicité de tâches souvent répétitives et chronophages.

Un logiciel ERP Également connu sous l'appellation **PGI** (Progiciel de gestions intégrées), est un outil indispensable pour automatiser certaines opérations inhérentes au bon fonctionnement d'une entreprise. Sur ce, nous allons présenter les logiciels ERP, ensuite nous enchaînons avec une petite introduction des technologies Node.

## 1. Les ERP

### 1.1. Définition :

L'**ERP** (Entreprise Resource Planning) qui signifie le progiciel de Gestion intégré : est un outil informatisé qui permet le pilotage de l'entreprise, sa particularité est d'embarquer en un même logiciel et une seule base de données les fonctionnalités nécessaires à la gestion des activités de l'entreprise : gestion comptable, gestion commerciale, gestion des stocks, la GRH.[6]

Pour être qualifié d'ERP, une solution logicielle doit couvrir au moins deux principes fondamentaux qui sont les suivants :

- Construire des applications informatiques sous forme de modules indépendants mais parfaitement compatibles sur une base de données unique et commune.
- L'usage d'un moteur de Workflow permet de définir l'ensemble des tâches d'un processus et de gérer leur réalisation dans tous les modules du système qui en ont besoin.

Optimiser l'organisation et la planification sont les piliers de la coordination entre les ressources humaines et les exigences des services opérationnels des entreprises. Dans l'univers des systèmes d'information et de partage d'informations à vocation transversale, le logiciel ERP démontre ses capacités éprouvées en tant qu'outil décisionnel. [7]



## 1.2. Les ERP pour qui ?

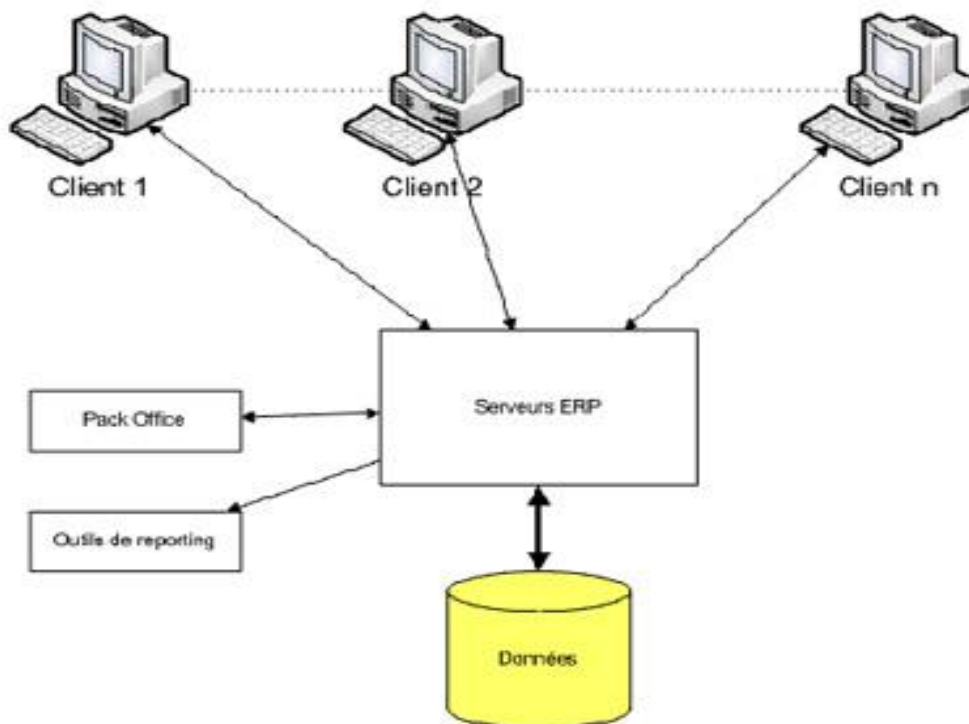
Les ERP sont principalement destinés aux grandes entreprises ou multinationales du fait d'un coût important. Cependant, le marché des ERP tend à se démocratiser vers les PME/PMI. Certains éditeurs conçoivent un ERP uniquement pour ce type de structure. Enfin, il existe des ERP open source ce qui revient moins cher, puisqu'il n'y a pas de coût de licence. En revanche, il faut inclure dans le calcul du coût d'acquisition total, les frais de maintenance et l'assistance technique [8].

### Objectifs des ERP :

- La **gestion du personnel** : le progiciel permet d'attribuer à chacun les tâches qu'il aura le temps d'effectuer. La surcharge sera ainsi évitée. Les capacités, les qualifications et les qualités de chaque employé seront mises en valeur et serviront à la productivité de l'entreprise.
- La **gestion des ressources** : l'ERP centralisera l'ensemble des informations matérielles, humaines et comptables de la société. Il permettra donc un gain en productivité non-négligeable.
- La **planification et le développement des projets** : une fois l'ERP installé. Cet outil établira un diagramme de Gantt, permettra l'échelonnement des tâches et des étapes, mais aussi la meilleure répartition des missions. Tous ces indicateurs ne pourront qu'aider à la réalisation des expansions et améliorations futures.[9]

## 1.3. Architecture détaillée d'un ERP :

Concernant le déploiement d'un ERP, celui-ci est la plupart du temps client/serveur comme le décrit le schéma ci-dessous



**Figure 1: Architecture d'un ERP**

Les ERP présentent une structure informatique de type Client/serveur à trois niveaux :

1. Un serveur de base de données PostgreSQL QL (qui peut contenir plusieurs bases de données).
2. Un serveur d'application (contenant les objets de gestion, le moteur de Workflow, le générateur d'édition ...etc.).
3. Un serveur de présentation (Open ERP web) qui permet à l'utilisateur de se connecter à Open ERP avec n'importe quel navigateur internet.
4. De plus, les ERP sont compatibles Pack Office, en particulier pour Power Point et Excel.[7]

#### **1.4. Principe de fonctionnement:**

Le principe fondateur d'un PGI est de construire des applications informatiques (gestion des commandes, des stocks, de la paie, de la comptabilité...) :

- de manière modulaire et intégrée au niveau des traitements offerts (les différents modules qui le composent sont indépendants mais parfaitement compatibles entre eux) ;

- de manière rigoureuse et cohérente au niveau des données gérées (partage d'une base de données unique et commune).

Cela comble une lacune importante :

- Dans la situation préexistante aux PGI, des applications *sur mesure*, d'origine diverse, coexistent tant bien que mal, ne partagent pas ou peu leurs données, et ne sont pas forcément toujours prévues pour travailler simplement et correctement ensemble. Les équipes informatiques ont alors fort à faire pour construire et mettre en place des interfaces *ad hoc*, dont le fonctionnement pratique peut réserver des surprises.
- La compatibilité entre les modules qui composent les PGI est garantie par leur concepteur : la compatibilité du module achat avec le module stock est garantie, qui est lui-même compatible avec le module gestion de commandes. Les données sont désormais supposées standardisées et partagées, ce qui élimine les saisies multiples et évite (en théorie) l'ambiguïté des données multiples de même nature
- L'autre principe qui caractérise un PGI est l'usage systématique de ce qu'on appelle un moteur de **workflow** (qui n'est pas toujours visible de l'utilisateur), et qui permet, lorsqu'une donnée est entrée dans le système d'information, de la propager et d'offrir des vues logiques pertinentes dans tous les modules du système qui en ont besoin, selon une programmation prédéfinie.

Ainsi, on peut parler de PGI lorsqu'on est en présence d'un système d'information composé de plusieurs applications partageant une seule et même base de données, par l'intermédiaire d'un système automatisé prédéfini éventuellement paramétrable (un moteur de *workflow*).[7]

### 1.5. Qu'est un moteur Workflow ?

On appelle workflow ou « gestion de flux de travail » : la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier ou « processus d'affaires » ou encore « processus opérationnel ».

Il s'agit en effet de formaliser les traitements à réaliser, le cheminement à suivre et les acteurs concernés pour accomplir un travail précis).[7]

### **1.6. Les principaux éditeurs d'ERP :**

On distingue deux types d'ERP:

- les ERP propriétaires : édités par des sociétés, ce qui implique l'achat d'une licence.
- les ERP open source : qui sont "gratuits". Nous ne nous intéresserons qu'aux ERP propriétaires.[8]

### **1.7. Les principaux ERP du marché:**

- SAP (leader mondial).
- Oracle/People Soft.
- SAGE ADONIX.
- Microsoft.
- SSA Global.
- GEAC.
- Intentionia/Lawson.
- Infor Global Solutions.

### **1.8. Avantages et inconvénients des ERP :**

**Avantages d'un ERP :**

- L'intégrité et l'unicité du SI : c'est à dire qu'un ERP permet une logique et une ergonomie unique à travers sa base de données, elle aussi unique au sens "logique". Ceci se traduit par le fait qu'il peut exister plusieurs bases de données "physiques" mais celles-ci respectent la même structure. En bref, un ERP permet d'éviter la redondance d'information entre différents SI de l'entreprise.
- L'utilisateur a la possibilité de récupérer des données de manière immédiate ou encore de les enregistrer : Un avantage important, les mises à jour dans la base de données sont effectuées en temps réel et propagées aux modules concernés.
- Un ERP est un outil multilingue et multidevise : il est donc adapté au marché mondial, en particulier aux multinationales.

- Pas d'interface entre les modules : il y a synchronisation des traitements et optimisation des processus de gestion. De même, la maintenance corrective est simplifiée car celle-ci est assurée directement par l'éditeur et non plus par le service informatique de l'entreprise. (Celui-ci garde néanmoins sous sa responsabilité la maintenance évolutive: amélioration des fonctionnalités, évolution des règles de gestion, etc.).
- Un ERP permet de maîtriser les stocks : élément important pour la plupart des entreprises car les stocks coûtent chers.

#### **Inconvénients d'un ERP :**

- Demande de l'organisation et de la rigueur.
- La mise en œuvre est complexe.
- La bonne connaissance et la remise en cause des processus de l'entreprise.
- Des difficultés d'appropriation par le personnel de l'entreprise liées aux changements.

## **2. Les Technologies de NodeJS**

NodeJS est un environnement d'exécution JavaScript open source utilisé pour exécuter du code JavaScript. 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 ». Il a été créé par Ryan Dahl en 2009. 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.[13]

### **2.1. Comparaison entre NodeJS et JavaEE :**

- Java est un langage de programmation orienté objet qui existe depuis plus de 20 ans. Malgré son grand âge, il s'impose, encore aujourd'hui, comme une référence pour le développement côté serveur.
- Node.js n'est quant à lui pas un langage mais plutôt un environnement d'exécution permettant d'utiliser du JavaScript côté serveur. Il utilise pour cela le moteur JS V8

(Google Chrome). Si JavaScript est un langage presque aussi ancien que Java, Node.js est beaucoup plus récent : il fut créé en 2009 seulement.

- Nous avons donc deux solutions pouvant s'utiliser côté serveur, qui possèdent chacune ses avantages et ses inconvénients. Différents facteurs vont permettre de mieux choisir la technologie à utiliser, au-delà, bien sûr, des contraintes liées à l'entreprise par exemple.[13]

#### **Performance et rapidité :**

- Avec son serveur multithread, Java est un langage qui s'adapte très bien aux applications avec une forte consommation de ressources CPU (streaming vidéo, édition d'image, etc.). Si une requête demande beaucoup de calculs, par exemple, les autres threads prennent le relais, permettant, en général, un impact réduit sur la vitesse et la performance.
- Avec son single-thread, Node.js dévoile plutôt son potentiel pour les applications effectuant de nombreuses requêtes IO (entrées/sorties). En effet, les opérations IO sont non-bloquantes, contrairement à Java, retirant ainsi la nécessité d'attente pour chaque opération. C'est bien simple, plus il y a de requêtes IO, plus Node.js s'affirme face à Java. En termes d'applications, on peut inclure toutes celles qui incluent des opérations massives de lecture et d'écriture comme la connexion, l'acquisition de données, les sauvegardes (backups), le processus de transaction et des applications en temps réel (RTA).[13]

#### **Indépendance et compatibilité :**

- L'un des principaux atouts de Java par rapport à Node.js, c'est son indépendance. Il peut s'exécuter sur n'importe quelle plateforme peu importe sa plateforme d'origine grâce à la JVM. Node.js, quant à lui, possède des modules qui sont liés à l'OS, ce qui nécessite une adaptation pour être exécuté sur un OS différent.
- Un des avantages de Node.js serait plutôt le langage qu'il utilise : le JavaScript. Celui-ci est également utilisé en frontend ce qui favorise les migrations de code, la logique utilisée sur le navigateur pouvant fonctionner sur le serveur avec une adaptation minime.[13]

#### **Rapidité d'écriture :**

- Le développement d'une application avec Node.js est plus rapide qu'avec Java. Il demande généralement moins de lignes de code et moins de fichiers. L'application en Node.js fut réalisée presque 2 fois plus vite avec moins de développeurs, contenant 33% de lignes de code en moins et 40% de fichiers en moins.
- Néanmoins, la rapidité d'écriture ne peut être le seul argument, ce qui permet tout à fait à Java de rester dans la course.[13]

#### **Ecosystème :**

- Java, notamment par son ancienneté, dispose de très nombreuses bibliothèques. Celles-ci sont majoritairement open-sources et gérées par des développeurs Java confirmés, appartenant à une large communauté.
- Node.js s'articule plutôt autour de modules, disponibles en grande quantité malgré son jeune âge. Et pour cause, sa communauté est composée de passionnés très dynamiques, en faisant l'une des plus actives sur Git Hub. Ces modules sont le plus souvent consacrés à des utilisations très spécifiques et facilement installables via le gestionnaire NPM (Node Package Manager). Ceci participe finalement à sa particularité d'être très extensible.[13]

**Côté Framework :** c'est Node.js qui en propose le plus avec express.js ou encore meteor.js, pour citer les plus connus. Java de son côté est plutôt orienté sur les bibliothèques.[13]

#### **Tableau comparatif :**

JAVA	NODE.JS
Applications utilisant beaucoup de CPU	Applications web back-end
Internet des Objets (IoT)	Applications desktop
Sites e-commerce	Des applications en temps réel (beaucoup de requêtes IO)
Technologie pour le Big Data	

JAVA	NODE.JS
API	Outils de développement

## 2.1. Finalement, Java ou Node.js ?

Avec la popularité croissante de Node.js, on pourrait être tenté de l'utiliser pour suivre la tendance. Pourtant, ce sont bien les caractéristiques finales de l'application qui doivent déterminer la technologie backend à utiliser. On ne le dira jamais assez mais une réflexion en amont orientée besoins/solutions reste la meilleure manière d'aborder un projet.[13]

## Conclusion

Outre les nombreux volets développés durant ce chapitre, une bonne conception s'impose pour arriver à formaliser toutes les étapes préliminaires et atteindre notre objectif qui constitue la réalisation d'un planning d'organisation des chantiers.



## **CHAPITRE 3 : Architecture logicielle**

## **Introduction**

La création d'un logiciel met en œuvre un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus qui démarre d'une problématique posée jusqu'à arriver à la conception.

Dans ce chapitre, nous allons décrire l'architecture globale de notre logiciel en effectuant simultanément l'étude des données et des traitements et en gérant les interactions de ces dernières avec l'utilisateur .de là en découle la description des bases de données à créer.

### **1. Présentation du projet**

Le projet consiste à proposer une solution informatique à la problématique de planification de projets dans le domaine du BTP.

L'ampleur de certains projets des entreprises, les facteurs qui rentrent dans la réalisation d'un projet, les ressources, les contraintes imposées, les prises de risque rendent la planification d'un projet impérative et liée directement à son succès.

### **2. Démarche de développement**

Le processus de développement constitue un facteur déterminant dans la réussite d'un projet, du fait qu'il cadre ses différentes phases et caractérise les principaux traits de sa conduite. Pour cela, le choix d'une méthode de développement, qui soit adéquate aux particularités et exigences d'un projet, doit être élaborée au préalable afin d'obtenir un produit de qualité qui répond aux besoins et aux attentes des utilisateurs. Parmi les processus de développement les plus répandus on trouve les modèles en Cascade et en V.

### **2.1. Le cycle de développement en Cascade :**

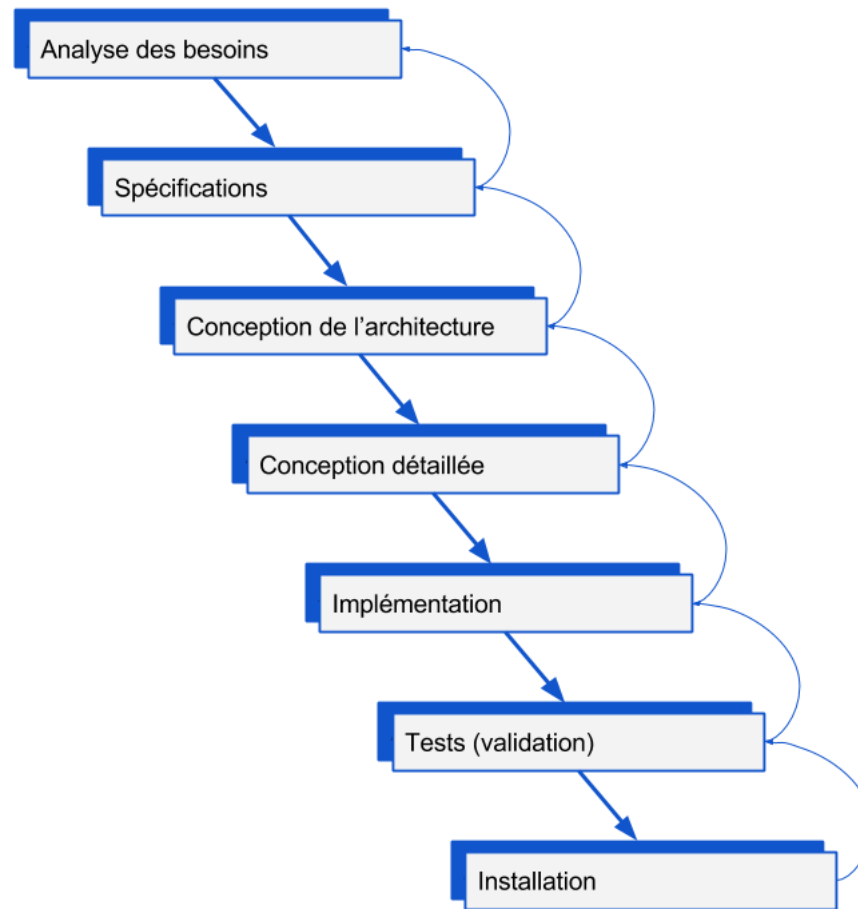
Le modèle en cascade (en anglais : *waterfall model*) est un modèle de gestion linéaire qui divise les processus de développement en phases de projet successives.

Présenté par Winston W. Royce en 1970, le modèle en cascade originel est hérité du BTP. Il se base sur 2 idées fondamentales :

- Une étape ne peut pas être débutée avant que la précédente ne soit achevée : inutile de monter les murs tant que les fondations ne sont pas coulées.
- La modification d'une étape du projet a un impact important sur les étapes suivantes. [14]

#### **2.1.1. Le fonctionnement du cycle de développement en cascade :**

Ce modèle comporte 7 phases : analyse des besoins, spécifications, conception de l'architecture, conception détaillée, implémentation, tests (validation) et enfin installation. Chacune de ces phases doit produire un ou plusieurs livrables définis à l'avance et à une date d'échéance fixée. On ne peut passer d'une étape à l'autre que lorsque les livrables de l'étape en cours sont jugés satisfaisants. Si tout se passe bien on passe à la phase suivante, sinon on remonte à la phase précédente, voire même en début de cycle si une anomalie critique est détectée.



**Figure 2:Modèle en Cascade**

### **2.1.2. Avantages et inconvénient :**

Avantages	Inconvénients
Une structure simple grâce à des phases de projet clairement délimitées.	Les projets complexes ou à plusieurs niveaux ne peuvent que rarement être divisés en phases de projet clairement définies.
Une bonne documentation du processus de développement par des étapes clairement définies.	Une faible marge pour les ajustements du déroulement du projet en raison d'exigences modifiées.
Les coûts et la charge de travail peuvent être estimés dès le début du projet.	L'utilisateur final est uniquement intégré dans le processus de production après la programmation.
Les projets structurés d'après le modèle en cascade peuvent être représentés facilement sur un axe temporel.	Les erreurs sont parfois détectées uniquement à la fin du processus de développement.

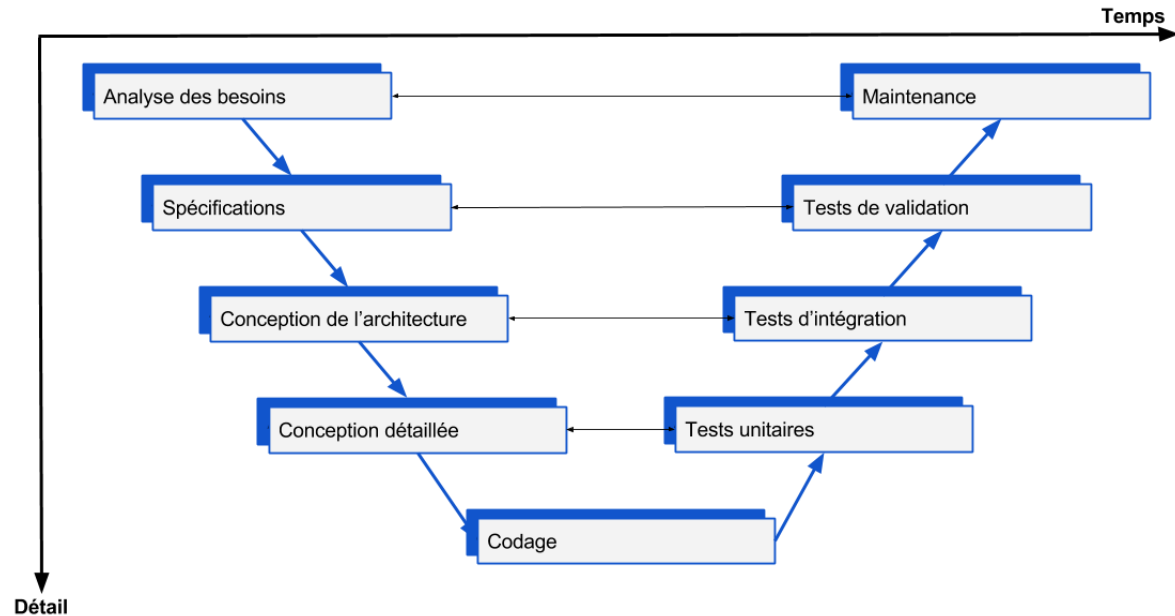
## 2.2. Le cycle de développement en V :

Le cycle en V ou V model en anglais est un modèle utilisé dans différents processus de développement, notamment dans le développement de logiciels. Élaboré dans les années 90 sous sa forme originale, il est perfectionné au fil des ans et adapté aux méthodes de

développement

L'idée de départ  
aux années 70 et a été  
sorte de prolongement

Outre les  
développement d'un  
défini parallèlement  
afférentes à mettre en  
d'assurance qualité et  
les différentes phases  
elles. Le cycle en V doit  
qui rappelle la lettre V.[14]



contemporaines.

remonte cependant  
imaginée comme une  
du modèle en cascade.

différentes phases de  
projet, le cycle en V  
les démarches  
place en termes  
détaille la façon dont  
doivent interagir entre  
son nom à sa forme

### 2.2.1. Fonctionnement du cycle de développement en V :

Les nouvelles phases du cycle en V arrivent donc comme ceci :

**Figure 3:Modèle en V**

spécifications, la conception architecturale, la conception détaillée

- **Codage** : développement de l'application

- **Etude et analyse** : l'analyse ou la définition des besoins, la rédaction des

- **Tests et validations** : tests unitaires, test d'intégration, tests de validation et maintenance corrective.

Sur le schéma ci-dessus on constate la présence de deux axes : le temps et le détail, c'est à cela qu'est dûe l'appellation en V du cycle : plus on avance dans l'étude plus le niveau de détail est précis, ensuite on code, et plus on avance dans les tests moins le niveau de détails est précis.

### 2.2.2. Avantages et inconvénients :

Avantages	Inconvénients
Optimisation de la communication entre les parties prenantes grâce à des modalités et des responsabilités clairement définies.	Du point de vue des développeurs, cette démarche s'avère souvent trop simpliste, car elle ne reflète pas intégralement le processus de développement.
Risques maîtrisés et meilleure planification grâce à des fonctions, des structures et des résultats bien définis en amont.	La gestion de projet occupe une place de plus en plus importante.

Avantages	Inconvénients
Amélioration de la qualité du produit grâce à l'intégration de mesures liées à l'assurance qualité	La rigidité relative de la structure ne permet guère de réagir avec souplesse aux modifications en cours de développement et favorise donc un déroulement relativement linéaire du projet.
Réduction des coûts grâce à un processus transparent de l'ensemble du cycle de vie du produit.	

### 2.3.Choix du cycle de développement :

Il existe différents types d'approches propres au secteur du développement de logiciels qui peuvent convenir en fonction des projets et de la structure des équipes. Le choix des approches est relativement large.

Dans le cadre de développement de notre projet, le cycle en cascade convient le plus, pour ce que ce dernier offre en termes de respect des délais et budgets grâce à sa rigidité et ses étapes bien définies.



## 2.4. Diagramme de contexte :

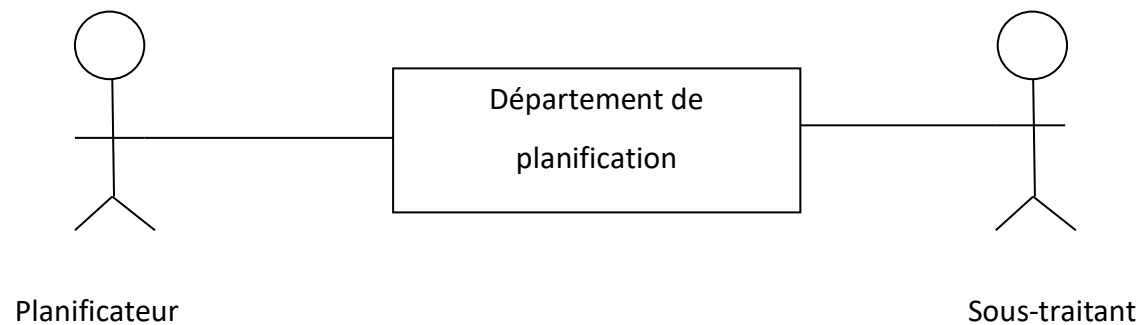
Le diagramme de contexte délimite le domaine d'étude en précisant ce qui est à la charge du système et en identifiant l'environnement extérieur au système étudié avec lequel ce dernier communique. Ses composants sont :

- Les acteurs externes. Un acteur externe est une entité externe au système étudié qui interagit avec le système.
- Un processus unique symbolisant le Système Information étudié.
- Echange entre le système étudié et son environnement

Dans notre analyse du travail du planificateur nous avons recensé le contexte ou le processus et les acteurs externes à savoir :

Le contexte : Planification des taches d'un projet.

Les acteurs : planificateur, sous-traitant.



Après étude détaillée du domaine du BTP et du métier du planificateur, nous pouvons résumer le processus de planification comme suit :

- Le travail de planification se fait par un planificateur en collaboration avec d'autres acteurs internes et externes à l'entreprise pour produire différents plannings qui vont préparer et accompagner le processus de construction.
- On peut distinguer 3 plannings :
  - Un planning général qui est produit à partir du cahier des charges.
  - Les plannings du personnel, matériels et matériaux élaboré en collaboration avec la sous-traitance.

### **3. Outils de conception**

#### **3.1. Merise :**

C'est une méthode d'analyse et de conception des SI basée sur le principe de la séparation des données et des traitements.

La vocation de MERISE est double, d'une part elle représente une méthode de conception de systèmes d'information (SI) et d'autre part elle propose une démarche méthodologique de développement de SI.

Les atouts majeurs de MERISE en tant que méthode de conception :

- Une approche globale sur les SI menées parallèlement sur les données et les traitements.
- Une description du SI par niveau : niveau conceptuel, niveau logique ou organisationnel, niveau physique ou opérationnel.
- Une description du SI en utilisant un formalisme précis, simple et rigoureux, pour la description des données.

- La présentation visuelle, notamment des modèles conceptuels, contribue dans une large mesure à l'établissement d'un dialogue constructif entre les intervenants qui collaborent ensemble pour concevoir le nouveau SI.[10]

Les points forts de la méthode en tant que méthode de développement :

- Un découpage du processus de développement en 4 étapes
  - Etude préalable.
  - Etude détaillée.
  - Réalisation.
  - Mise en œuvre.
- Une description détaillée de la structure de travail à mettre en place pour mener à bien le développement du SI.

MERISE distingue trois niveaux dans la description d'un SI, à chaque niveau correspondent une préoccupation et un ensemble de modèles pour la présentation des données et des traitements. Les trois niveaux représentent le cycle d'abstraction d'un SI.

### **3.1.1. Les niveaux de description et modèles associé :**

#### **Niveau conceptuel :**

Ce niveau de préoccupation correspond aux finalités de l'entreprise, il s'agit de décrire le « QUOI » en faisant abstraction des contraintes organisationnelles et techniques.

Les modèles utilisés pour la description du SI sont :

- **Le modèle conceptuel de données MCD** : la description des données et relation est faite à l'aide des trois concepts du formalisme individuel :
  - Objet (individu) : Un objet ou individu est une entité pourvue d'une existence propre, il confirme au choix de gestion de l'entreprise.
  - Relation entre objets ou individus : est une association entre deux ou plusieurs entités, une relation est dépourvue d'existence propre.
  - Propriété ou attribut : est une donnée élémentaire sur un objet ou sur une relation entre objets. Le nom de la propriété est inscrit à l'intérieure de l'objet.
  - Cardinalités : la cardinalité d'un objet par rapport à une relation s'exprime par deux nombres appelé cardinalité minimale (0 ou 1) et cardinalité maximale (1 ou n).
- **Le modèle conceptuel des traitements MCT** : la description dynamique du SI est faite à l'aide des concepts suivants :
  - Processus : constitue un sous ensemble de l'activité de l'entreprise dont les points d'entrées et de sorties sont stables et indépendants des choix d'organisations.
  - Opération : est constituée d'un ensemble d'actions qui sont exécutables sans interruptions. Elle est déclenchée pour répondre à la déclaration d'un événement qui produit un résultat.

**Niveau logique ou organisationnel :**

Les choix d'organisation sont pris en compte à ce niveau. La répartition homme/machine, le mode de fonctionnement : temps réel ou temps différé, la répartition géographique des données et des traitements.

Les modèles associés à ce niveau de description sont :

- **Le modèle logique des données MLD** : est construit à partir du MCD en tenant compte des éléments suivants :
  - Le niveau et type d'automatisation : il s'agit de ne retenir dans le MLD que la partie du MCD qui sera automatisée.
  - L'orientation des choix techniques concernant le système de gestion de base de données. Actuellement, on assiste à une mutation des logiciels de gestion de bases de données dites hiérarchiques vers des systèmes relationnels.
- **Le modèle organisationnel des traitements MOT**: permet de représenter par procédure les phases et tâches exécutées par chaque poste de travail. En d'autres termes « QUI FAIT QUOI ET OU ? »

#### **Niveau physique ou opérationnel :**

A ce niveau le « COMMENT FAIRE ? » est décrit, tous les choix techniques sont définis et les organisations physiques de données sont spécialisées au travers du modèle physique des données MPD et la description des traitements est réalisée pour chaque transaction (temps réel) ou chaque unité de traitement (temps différé) au travers du Modèle Physique des Traitements MPT.

- **Modèle physique des données MPD** : sa description est étroitement liée aux choix automatiques et informatiques concernant le système de gestion de base de données. On constate aujourd'hui que les entreprises optent généralement à l'utilisation d'un SGDB de type relationnel de plus l'environnement technique de développement influencera aussi largement la description du niveau physique des données.

- **Modèle physique des traitements MPT** : le but est de décrire l'architecture logicielle qui devra être réalisée à partir du Modèle Organisationnel des Traitements.[11]

### 3.2. UML :

Pour faire face à la complexité croissante des systèmes d'information, de nouvelles méthodes et outils ont été créés à savoir la programmation orientée objet (P.O.O.). Face à ce nouveau mode de programmation, les méthodes de modélisation classiques (telle MERISE) ont rapidement montré certaines limites et ont dû s'adapter.

UML permet donc de modéliser une application selon une vision objet. L'appréhension d'UML est complexe car UML est à la fois :

- une norme : UML est devenu une norme OMG (Object Management Group).
- un langage de modélisation objet.
- un support de communication : Il permet d'utiliser un langage commun grâce à sa notation graphique.
- un cadre méthodologique : l'élaboration des modèles.

**L'encapsulation** : consiste à masquer les détails d'implémentation d'un objet, en définissant une interface.

**L'interface** : est la vue externe d'un objet, elle définit les services accessibles (offerts) aux utilisateurs de l'objet.

L'encapsulation facilite l'évolution d'une application car elle stabilise l'utilisation des objets : on peut modifier l'implémentation des attributs d'un objet sans modifier son interface.

L'encapsulation garantit l'intégrité des données, car elle permet d'interdire l'accès direct aux attributs des objets.

**L'héritage** : est un mécanisme de transmission des propriétés d'une classe (ses attributs et méthodes) vers une sous-classe. Une classe peut être spécialisée en d'autres classes, afin d'y ajouter des caractéristiques spécifiques ou d'en adapter certaines.

Plusieurs classes peuvent être généralisées en une classe qui les factorise, afin de regrouper les caractéristiques communes d'un ensemble de classes. La spécialisation et la généralisation permettent de construire des hiérarchies de classes. L'héritage peut être simple ou multiple. Il évite la duplication et encourage la réutilisation.

**Le polymorphisme** : représente la faculté d'une même opération de s'exécuter différemment suivant le contexte de la classe où elle se trouve. Ainsi, une opération définie dans une superclasse peut s'exécuter de façon différente selon la sous-classe où elle est héritée. Le polymorphisme augmente la généricité du code. [12]

### **Diagrammes :**

UML propose plusieurs types de diagrammes. Chaque type de diagramme UML possède une structure et véhicule une sémantique. Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système.

#### **3.2.1. Les différents types de diagrammes UML :**

Il existe 2 types de vues du système qui comportent chacune leurs propres diagrammes :

- Les vues statiques :
  - Diagrammes de cas d'utilisation

- Diagrammes d'objets
- Diagrammes de classes
- Diagrammes de composants
- Diagrammes de déploiement
- Les vues dynamiques :
  - Diagrammes de collaboration
  - Diagrammes de séquence
  - Diagrammes d'états-transitions
  - Diagrammes d'activités

**Vu statique :**

Le but de la conceptualisation est de comprendre et structurer les besoins du client, filtrer et organiser les besoins.

**Diagrammes de cas d'utilisation :**

Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Les cas d'utilisation identifient les utilisateurs du système (acteurs) et leurs interactions.



**Diagramme de classe :**

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. Il permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine.

**Diagramme de composants :**

Les diagrammes de composants décrivent les composants et leurs dépendances dans l'environnement de réalisation. Un composant est une vue physique qui représente une partie implémentable d'un système. Un composant peut être du code, un script, un fichier de commandes, un fichier de données, une table ... Il peut réaliser un ensemble d'interfaces qui définissent alors le comportement offert à d'autres composants.

**Diagramme de collaboration :**

Le diagramme de collaboration permet de mettre en évidence les interactions entre les différents objets du système et les messages qu'ils échangent.

**Diagramme de séquence :**

Le diagramme de séquence est une variante du diagramme de collaboration. Par opposition aux diagrammes de collaboration, ils possèdent intrinsèquement une dimension temporelle. Ils privilégient ainsi la représentation temporelle à la représentation spatiale.

**Vue Dynamique :**

La vue des cas d'utilisation est une description fonctionnelle des besoins, structurée par rapport à des acteurs. Le passage à l'approche objet s'effectue en associant une collaboration à chaque cas d'utilisation.

### **Le diagramme de séquence :**

Permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente le temps, l'axe horizontal les objets qui collaborent.

Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

Une période d'activité correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet qui lui sert de sous-traitant.

Les périodes d'activité se représentent par des bandes rectangulaires placées sur la ligne de vie des objets.

### **Diagrammes d'états-transitions :**

Ils ont pour rôle de représenter les traitements (opérations), ils définissent l'enchaînement des états de classe et font donc apparaître l'ordonnancement des travaux. Le diagramme d'états-transition est associé à une classe pour laquelle on gère différents états : il permet de représenter tous les états possibles ainsi que les événements qui provoquent les changements d'état.[12]

## **3.3. Modélisation avec UML**

Pour passer à la conception, nous nous fondons sur les principes de l'approche orientée objet. À cet effet, nous passons d'une structuration fonctionnelle via les cas d'utilisation, à une structuration objet via les classes.

### 3.3.1. Diagrammes d'UML :

### 3.3.2. Cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision statique et globale du comportement fonctionnel d'un système logiciel.

#### Identification des acteurs :

Un acteur représente un élément externe qui interagit avec le système. Cet élément peut être un utilisateur ou un système tiers (autre ordinateur, autre programme, base de données). En observant les utilisateurs directs qui interagissent avec le système que nous tentons de définir, nous trouvons ces acteurs qui opèrent avec le système : Planificateur.

#### Cas d'utilisation « connexion »

Titre	Connexion
Acteur	Planificateur.
Pré-conditions	L'utilisateur doit avoir le droit d'accès.
Scénario	L'utilisateur saisit ses droits d'accès. Le système vérifie si l'utilisateur existe sinon, un message d'erreur s'affiche. Le système redirige l'utilisateur vers la page. d'accueil.

#### Cas d'utilisation « Gérer les plannings »

Titre	Gérer les plannings
Acteurs	Planificateur.

Pré-condition	L'utilisateur doit s'authentifier.
Scénario	<p>Le planificateur choisit un planning. Le système cherche le planning demandé et l'affiche.</p> <p>Le planificateur demande à effectuer des modifications.</p> <p>Le système envoie le formulaire de modification.</p> <p>Le planificateur remplit le formulaire. Le système enregistre les modifications et met à jour la base de données.</p>

#### Cas d'utilisation « Modifier une tache »

Titre	Modifier une tache
Acteurs	Planificateur.
Pré-condition	L'utilisateur doit s'authentifier.
Scénario	<p>Le planificateur demande de consulter la liste des taches.</p> <p>Le système cherche la liste et la renvoie</p> <p>Le planificateur choisit une tache et demande à la modifier.</p> <p>Le système envoie le formulaire de modification.</p> <p>Le planificateur remplit le formulaire. Le système enregistre les modifications et met à jour la base de données.</p>

Cas d'utilisation « Modifier une ressource »

Titre	Modifier une ressource
Acteurs	Planificateur.
Pré-condition	L'utilisateur doit s'authentifier.
Scénario	<p>Le planificateur demande de consulter la liste des ressources.</p> <p>Le système cherche la liste et la renvoie</p> <p>Le planificateur choisit une ressource et demande à la modifier.</p> <p>Le système envoie le formulaire de modification.</p> <p>Le planificateur remplit le formulaire.</p> <p>Le système enregistre les modifications et met à jour la base de données.</p>

Cas d'utilisation « Supprimer une ressource »

Titre	Supprimer une ressource
Acteurs	Planificateur.
Pré-condition	L'utilisateur doit s'authentifier.
Scénario	<p>Le planificateur demande de consulter la liste des ressources.</p> <p>Le système cherche la liste et la renvoie</p> <p>Le planificateur choisit une ressource et demande à la Supprimer.</p> <p>Le système demande la confirmation de la suppression.</p>

	<p>Le planificateur confirme la suppression.</p> <p>Le système enregistre les modifications et met à jour la base de données.</p>
--	---

#### Cas d'utilisation « Supprimer une tâche »

Titre	Supprimer une tâche
Acteurs	Planificateur.
Pré-condition	L'utilisateur doit s'authentifier.
Scénario	<p>Le planificateur demande de consulter la liste des tâches.</p> <p>Le système cherche la liste et la renvoie</p> <p>Le planificateur choisit une tâche et demande à la Supprimer.</p> <p>Le système demande la confirmation de la suppression.</p> <p>Le planificateur confirme la suppression.</p> <p>Le système enregistre les modifications et met à jour la base de données.</p>

### 3.3.2.1. Diagramme de cas d'utilisation général :

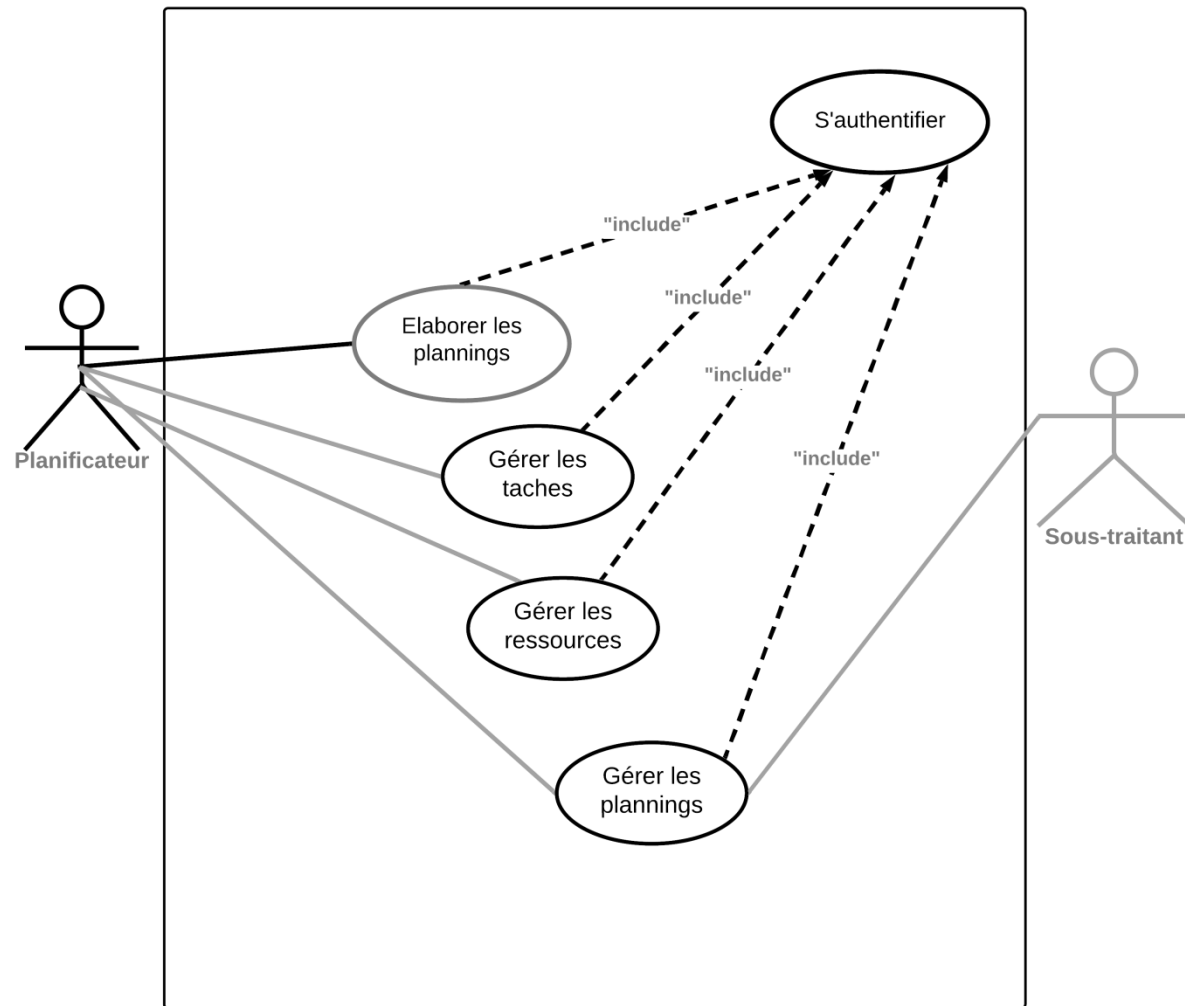
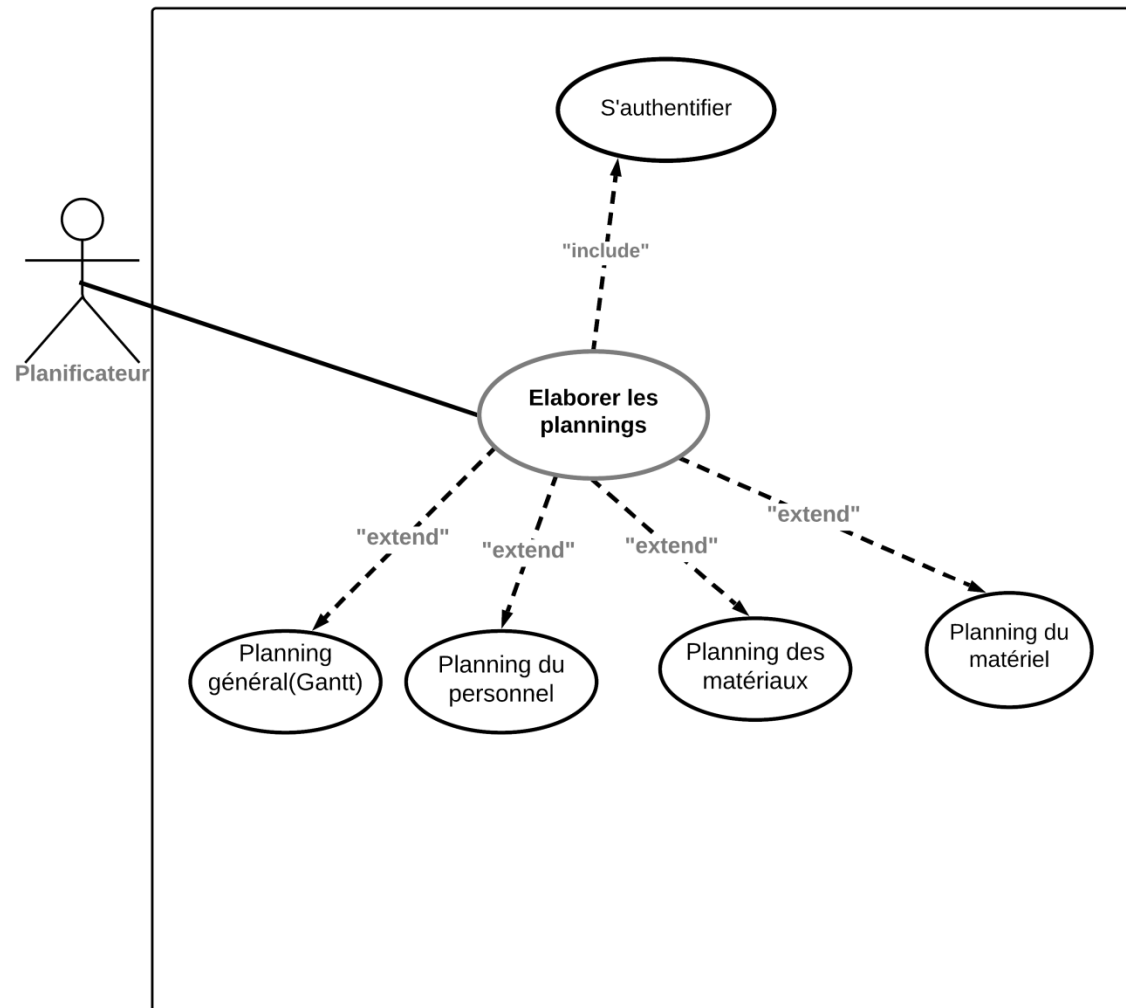


Figure 4:Diagramme de cas d'utilisation général

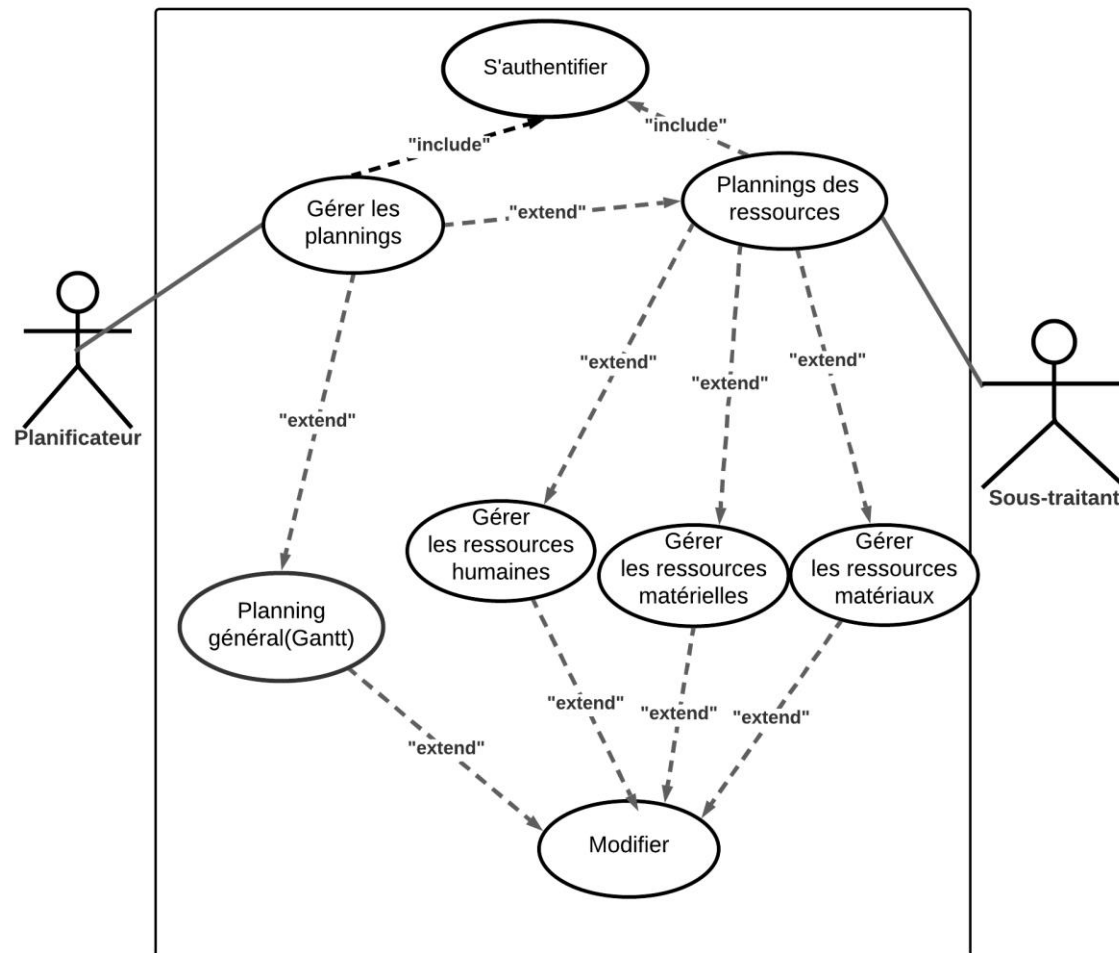
**3.3.2.2. Diagramme de cas d'utilisation pour le cas « élaborer plannings » :**





**Figure 5:Diagramme de cas d'utilisation pour le cas «élaborer plannings"**

**3.3.2.3. Diagramme de cas d'utilisation pour le cas « gérer les plannings » :**



**Figure 6:Diagramme de cas d'utilisation pour les cas «Gérer plannings»**

**3.3.2.4. Diagramme de cas d'utilisation pour le cas « gérer les ressources» :**

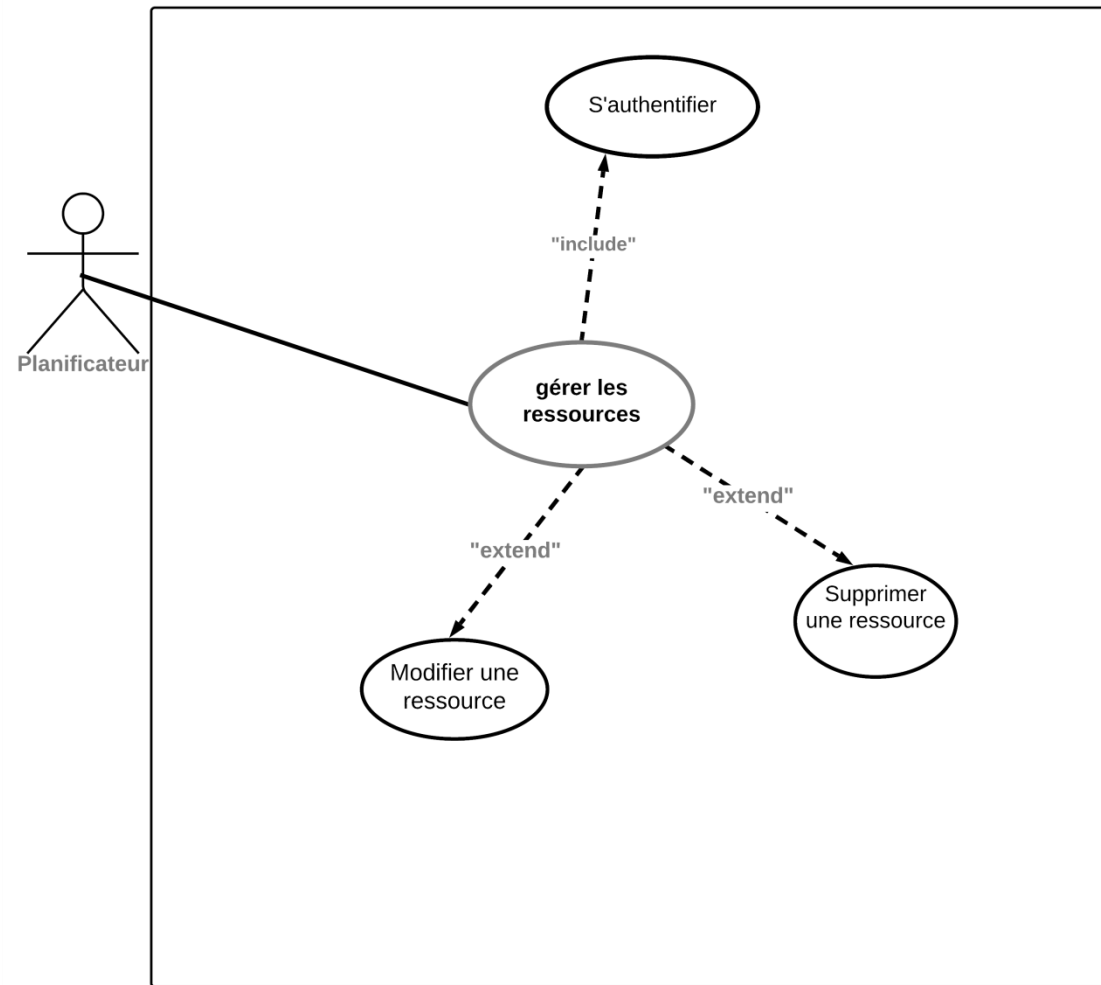


Figure 7:Diagramme de cas d'utilisation pour le cas «gérer les ressources » :

**3.3.2.5. Diagramme de cas d'utilisation pour le cas « gérer les taches » :**

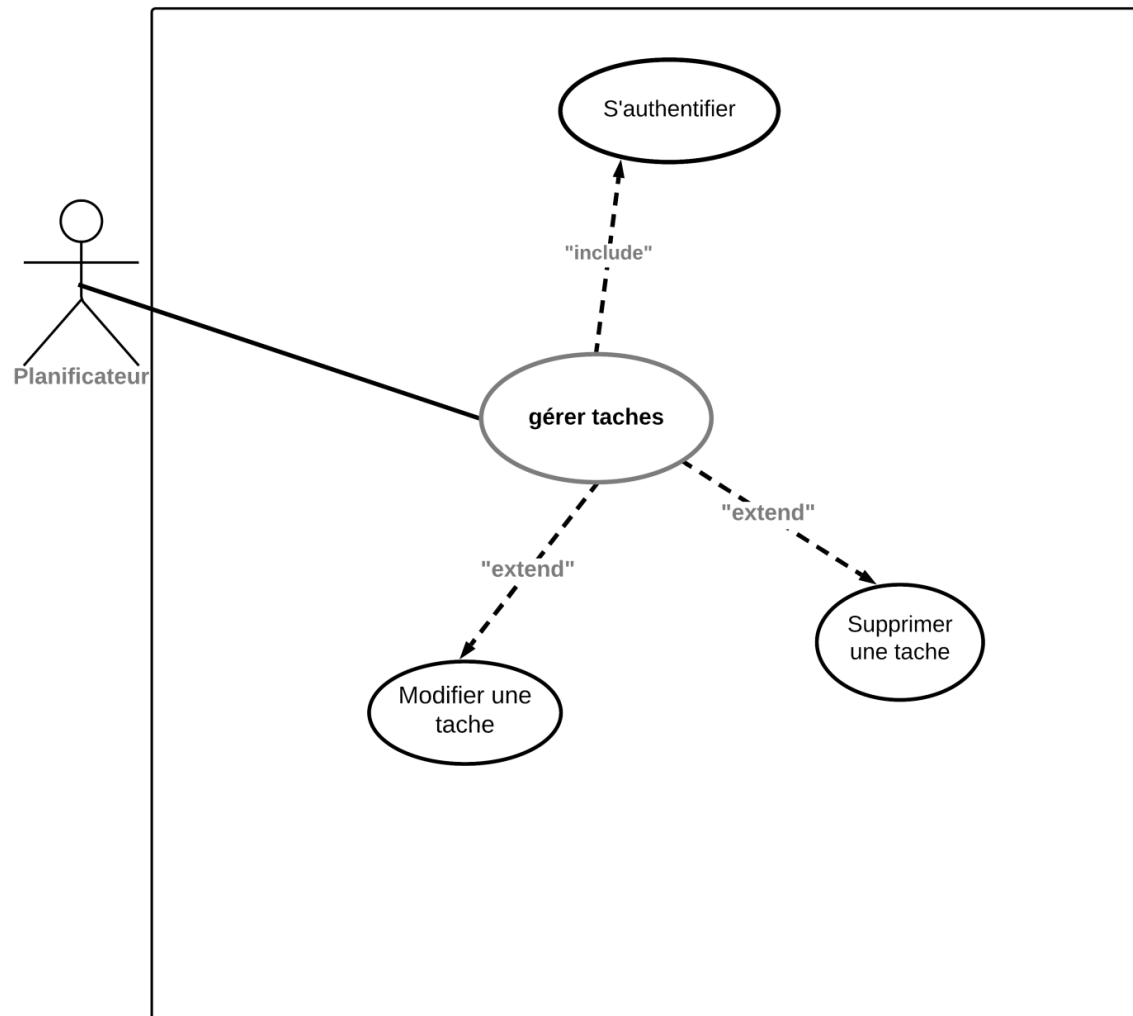


Figure 8:Diagramme de cas d'utilisation pour le cas «gérer les taches»

### **3.4. Diagrammes de séquence**

Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux. Le diagramme de séquence permet de cacher les interactions d'objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation dont le but est de décrire comment se déroulent les actions entre les acteurs ou objets. Après la description des cas d'utilisation, nous allons élaborer le modèle dynamique dans lequel nous allons décrire les scénarios de quelques cas d'utilisation, sous forme de diagrammes de séquence.

#### **3.4.1. Diagramme de séquence pour le cas « Connexion » :**



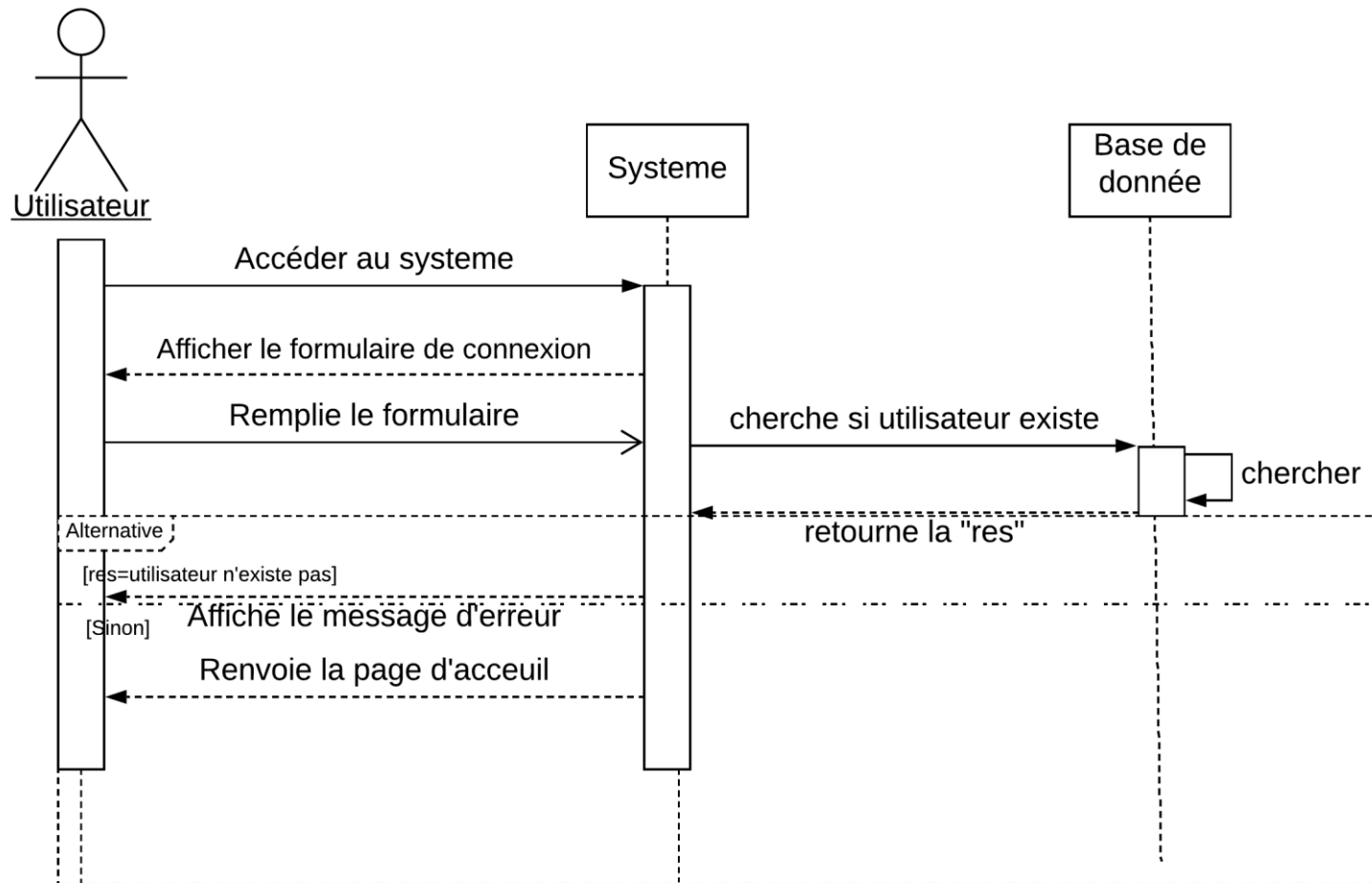


Figure 9:Diagramme de séquence cas "connexion"



### 3.4.2. Diagramme de séquence pour le cas « Gérer les plannings » :

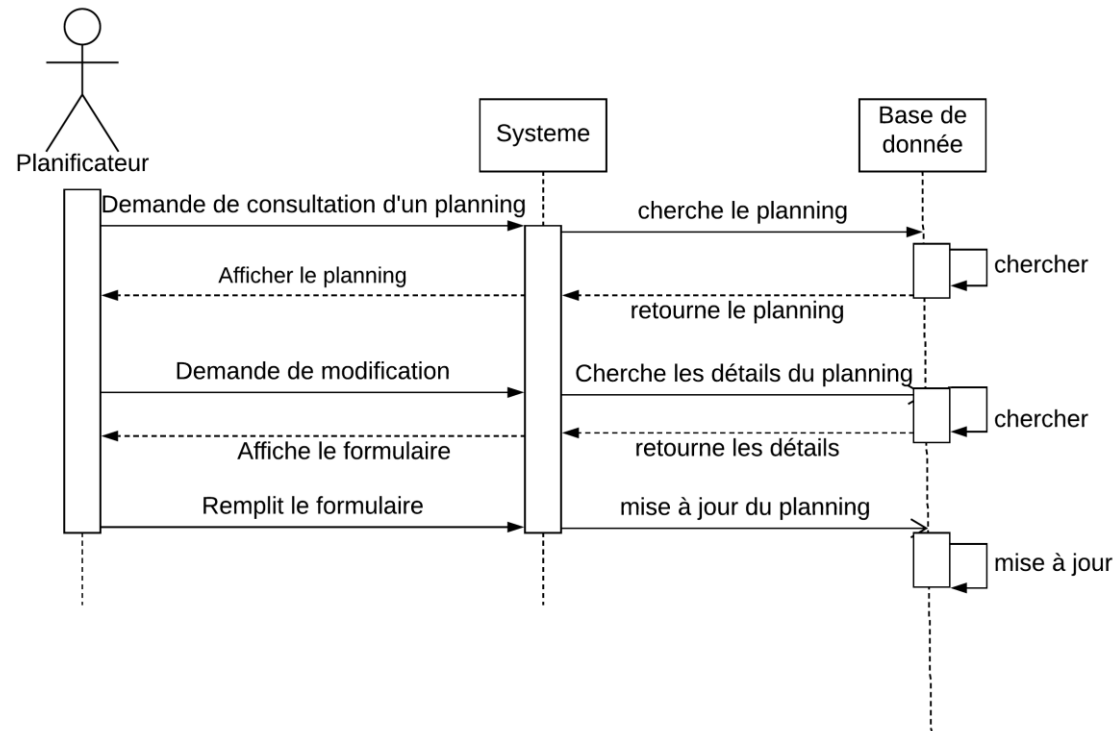


Figure 10:Diagramme de séquence cas "Gérer les plannings"

**3.4.3. Diagramme de séquence pour le cas « Gérer les plannings des ressources (matériels, matériaux et personnels) » :**

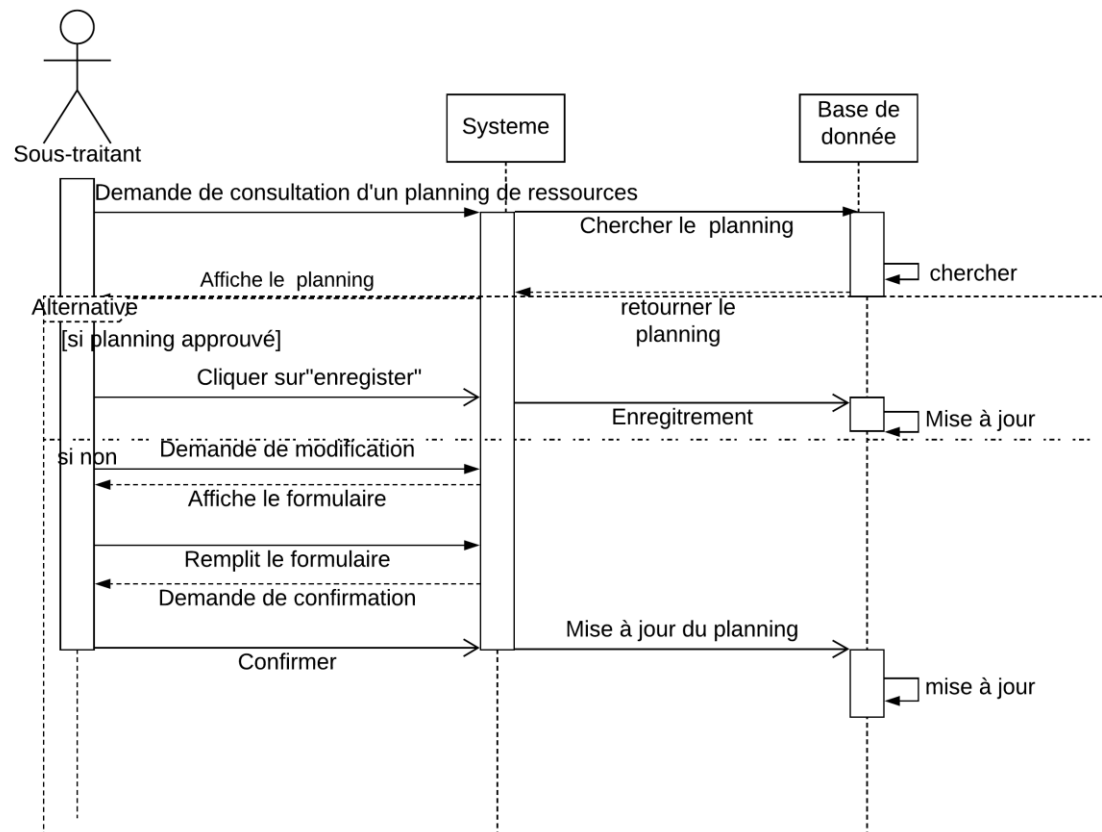


Figure 11:Diagramme de séquence pour le cas "gérer les plannings des ressources" :

#### **3.4.4. Diagramme de séquence du cas « Modifier une tâche »**

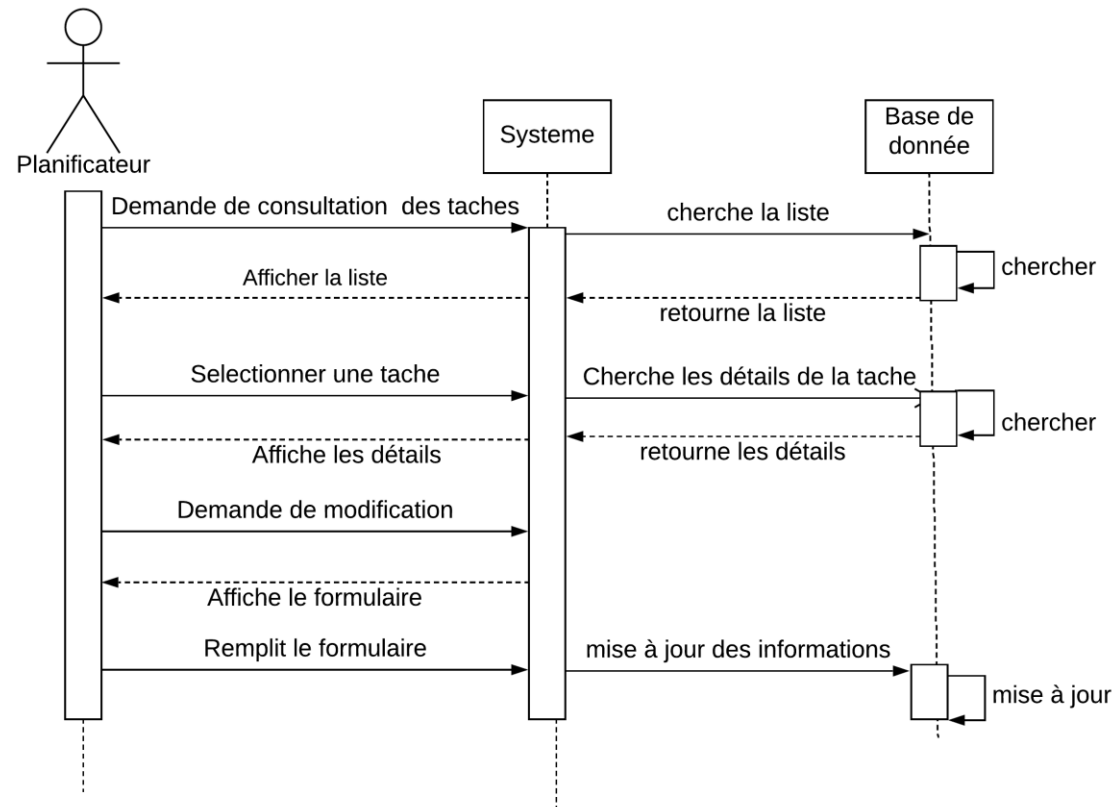


Figure 12:Diagramme de séquence cas "Modifier une tâche"

### 3.4.5. Diagramme de séquence du cas « Modifier une ressource » :

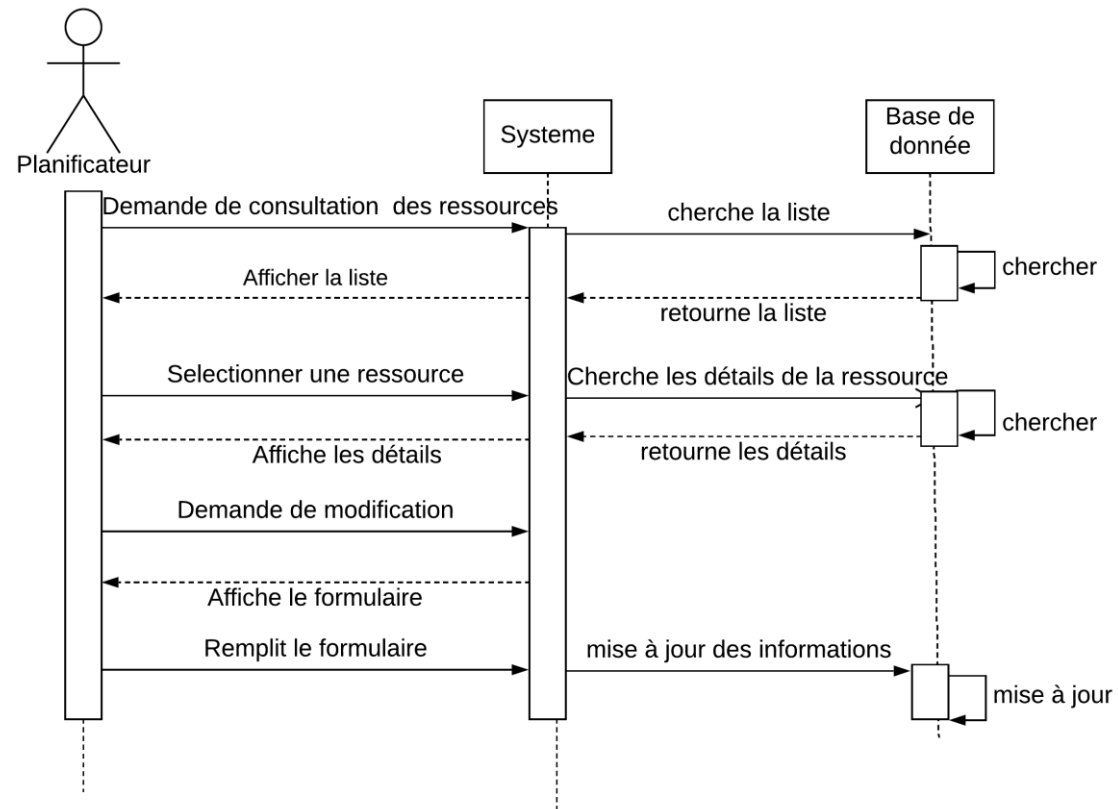
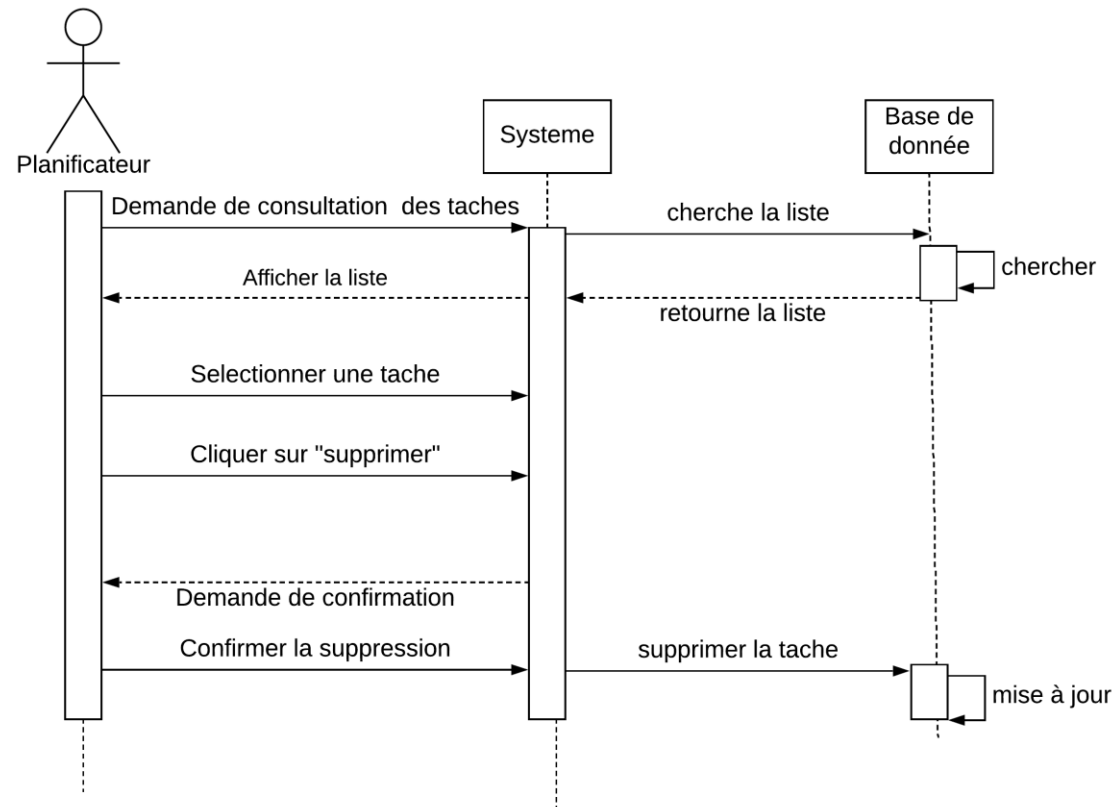


Figure 13:diagramme de séquence cas: « Modifier une ressource »



**3.4.6. Diagramme de séquence du cas « Supprimer une tâche » :**



**Figure 14:Diagramme de séquence pour le cas "Suppression d'une tâche"**

**3.4.7. Diagramme de séquence du cas « Supprimer une ressource » :**

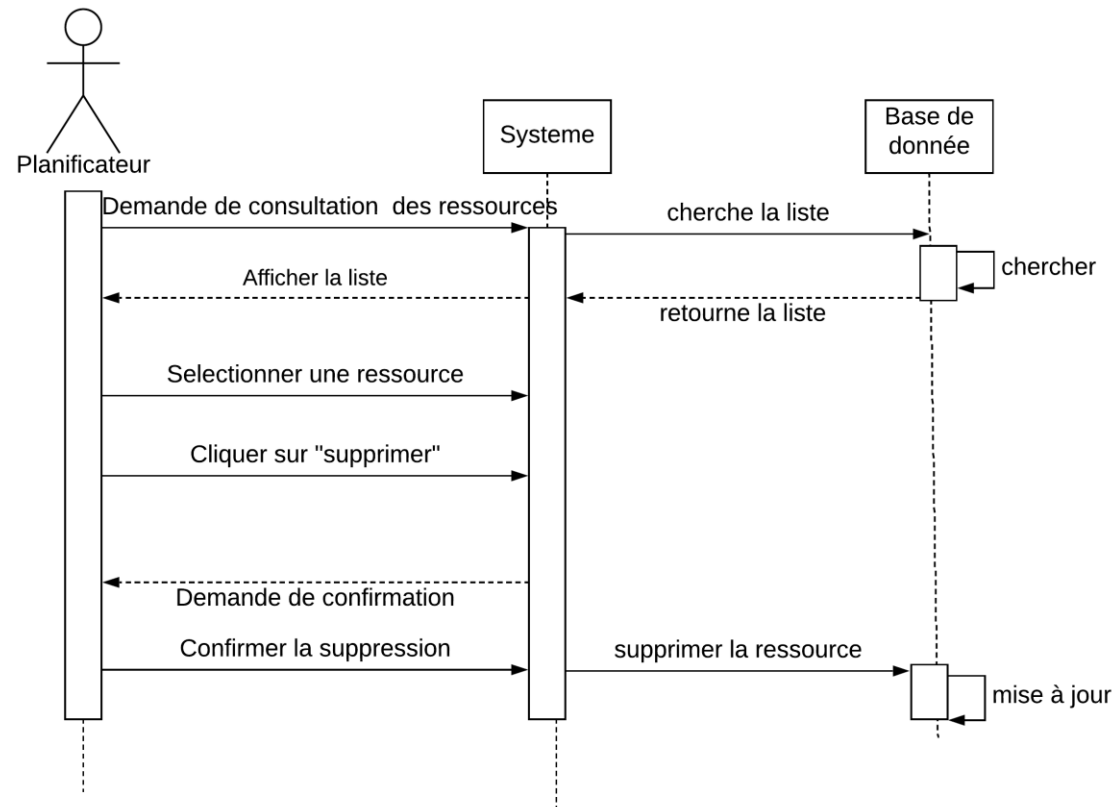
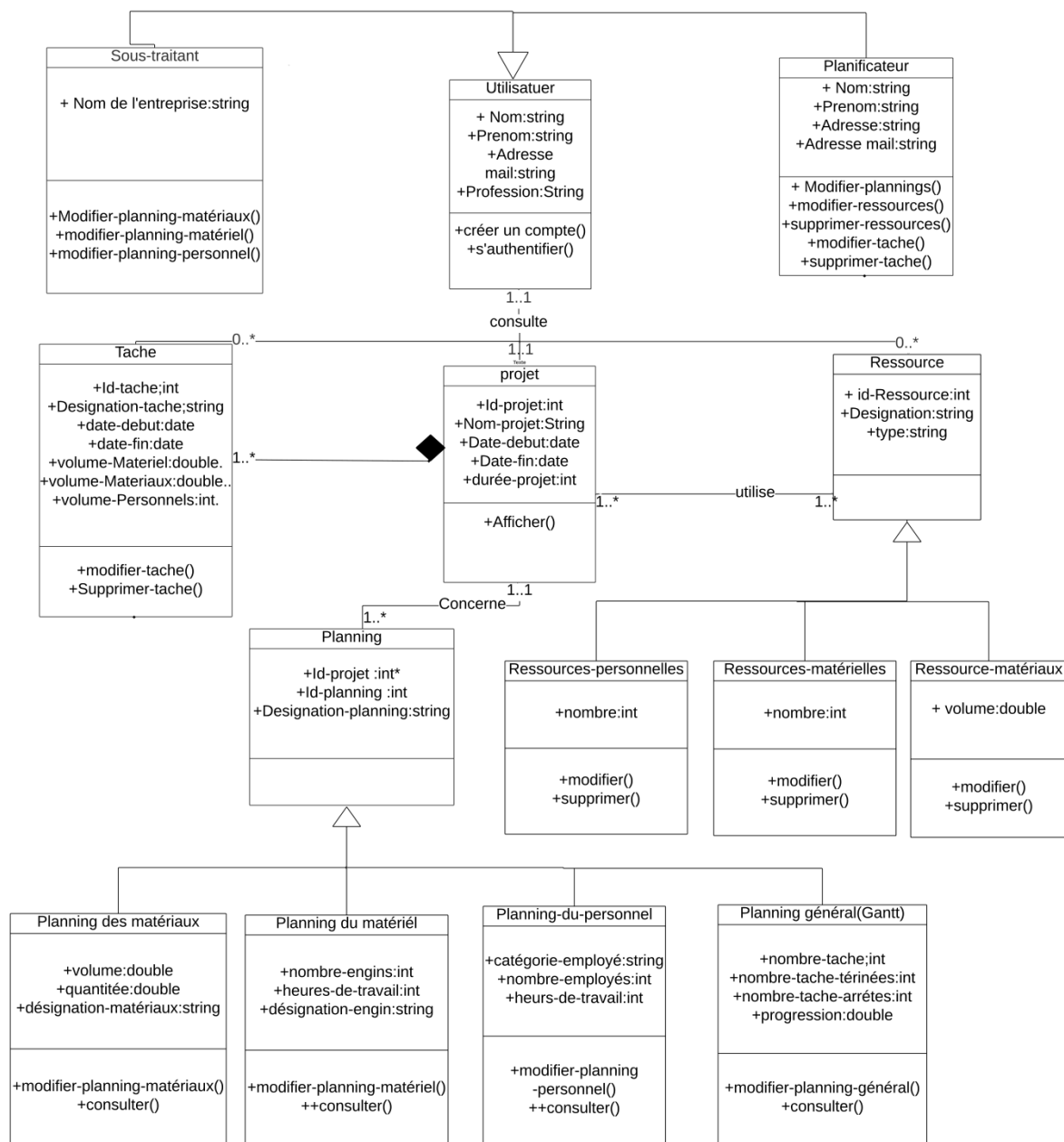


Figure 15:Diagramme de séquence pour le cas "suppression d'une ressource"

### **3.5. Diagramme de classes**

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Lors de la partie analyse du projet et des diagrammes de séquence présente, nous avons extrait un ensemble d'entités et de dépendances.



**Figure 16:Diagramme de classe**

## **Conclusion**

La phase de conception préliminaire s'avère d'une grande nécessité pour mener à bien son projet et cerner les objectifs à atteindre durant la prochaine étape du projet et de prendre en compte plusieurs dimensions dont toutes ne sont pas forcément quantifiables. De ce fait, il n'existe pas de méthode de planification miracle toute faite mais des méthodes à parfaire selon la situation et les ressources disponibles. De ce fait, nous pouvons entamer la phase de réalisation de l'application.

## **CHAPITRE 4: Réalisation**



## Introduction

Dans ce chapitre nous allons lancer la partie réalisation en indiquant les outils ainsi que l'ensemble des technologies que nous avons eues à notre disposition afin d'implémenter notre application.

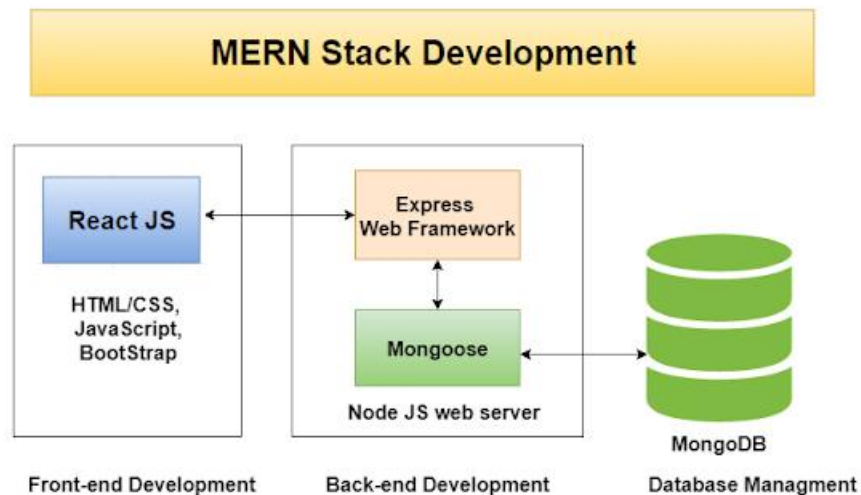
Sur ce, nous allons commencer par définir la technologie MERN. Ensuite, nous allons présenter quelques captures d'écran afin d'illustrer notre projet. Pour finir, on arrive à une conclusion.

### 1. Définition de la technologie MERN stack

La MERN représente l'alliance des technologies les plus puissantes que l'on trouve sur le marché. MERN est l'une des nombreuses variantes de la pile MEAN (MongoDB Express Angular Node) toutes basées sur JavaScript, où le Framework frontal Angular.js traditionnel est remplacé par React.js. Il s'agit d'un cadre de développement open source full stack, c'est-à-dire qu'il fournit des composants de développement front-end à back-end.

- Mongo DB.
- Express.
- Reactjs.
- Node.js.

Cette combinaison va permettre aux développeurs de créer des applications web complètes (back-end et front-end) où on utilise le JavaScript côté client et le Node.js côté serveur.[14]



**Figure 17:Architecture MERN**

## **2. Fonctionnement de la pile MERN :**

L'architecture MERN permet de construire facilement une architecture à 3 niveaux (frontend, backend, base de données) entièrement en utilisant JavaScript et JSON.

### **2.1. Niveau interface frontale React js :**

Le niveau supérieur de la pile MERN est React.js, le Framework JavaScript déclaratif pour la création d'applications dynamiques côté client en HTML.

#### **React js :**

React.js est un Framework JavaScript développé par Facebook depuis 2013. Conçu généralement pour les applications monopages à base de composants autonomes réutilisables qui maintiennent leur propre état, puis assemblés pour créer des interfaces utilisateurs complexes. React JS permet aux développeurs de modifier / éditer et d'actualiser la page pour afficher les modifications sans avoir à redémarrer ou recharger la page. [15]

### **2.2. Niveau serveur Express.js et Node.js :**

Le niveau suivant est le framework côté serveur Express.js, exécuté à l'intérieur d'un serveur Node.js.

### Express js :

Express.js se présente comme un «Framework» web rapide, minimaliste pour Node.js, il ne fournit que des fonctionnalités d'application web (et mobile) fondamentales pour le routage d'URL (faisant correspondre une URL entrante avec une fonction serveur) et la gestion des requêtes et des réponses HTTP.

### Node js :

Node JS est un environnement d'exécution JavaScript Open Source utilisé pour exécuter du code JavaScript coté client. En outre, Node.JS a la particularité d'être un langage extrêmement rapide, l'un des langages les plus rapides de tous les autres langages de la programmation informatique, car c'est également un langage asynchrone. Des entreprises de renom telles que Facebook utilisent Node.js.

La solution apportée par **Node.js** repose sur trois bases fondamentales :

- Le **moteur JavaScript V8** développé par Google, qui permet d'exécuter du code JavaScript à l'intérieur de Google Chrome et, grâce à Node, directement **sur le serveur**.
- Une **boucle d'événements, appelée aussi event loop NodeJS**, permettant d'exécuter plusieurs opérations simultanées de façon **asynchrone** et **non bloquante** en tirant profit des multiples fils d'exécution (*multithreading*) des noyaux des processeurs modernes.
- Une **API de bas niveau** basée sur la structure **entrées-sorties (I/O)** dénommée **libuv**, qui permet d'adopter une approche de **programmation événementielle**. [15]

## 2.3. Niveau de base de données MongoDB

MangoDB :

MongoDB est un SGBD NoSql open source et multiplateforme. Il s'agit d'une base de données orientée document, ce qui signifie que les données sont enregistrées à l'aide de collections et de documents, au lieu des tables et des lignes, comme dans une base de données relationnelle. Cela facilite et accélère l'intégration des données dans les applications.

MongoDB stocke les données au format JSON binaire qui permet l'échange rapide de données entre le client et le serveur. IL peut être utilisé pour le stockage de gros volumes de données, ce qui le rend hautement évolutif. Il est très simple de transformer des données JavaScript vers Mongo DB et inversement grâce à des librairies tel que mongoose par exemple.[15]

### Mongoose :

Mongoose est une bibliothèque ODM (Object Data Modeling) pour MongoDB et Node.js. Elle gère les relations entre les données, fournit la validation de schéma et elle est utilisée pour traduire entre les objets dans le code et la représentation de ces objets dans MongoDB.[16]

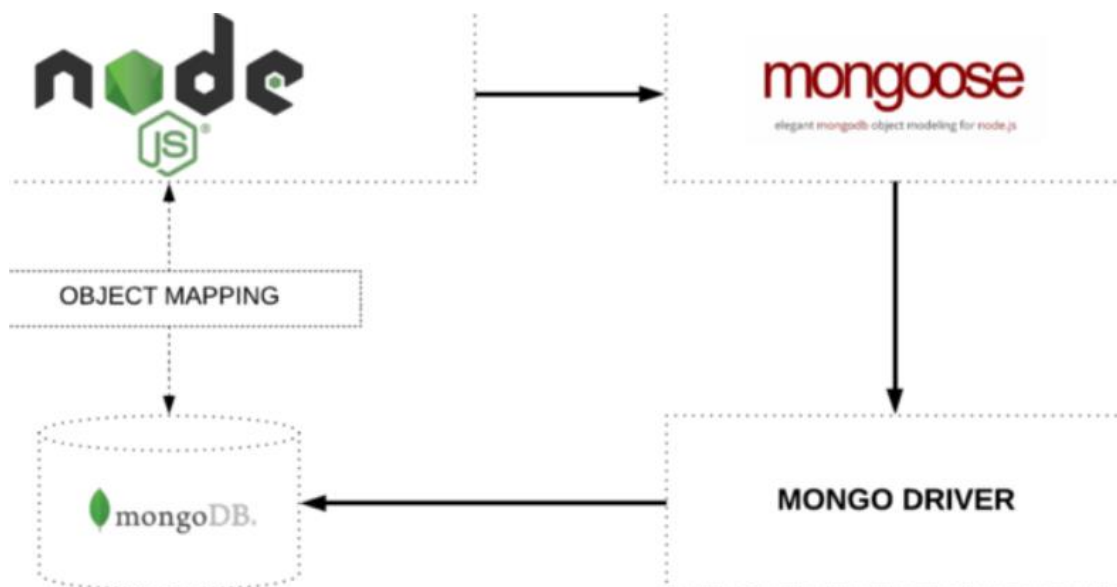


Figure 18: Mappage d'objets entre Node et MongoDB géré via Mongoose.

### 3. Node Package Manager (NPM):

Aujourd'hui, **toutes** les librairies et Framework JavaScript s'installent avec npm. Un package manager est un gestionnaire de dépendances. npm permet d'installer toutes les librairies, Framework et outils dont a besoin un projet JavaScript. C'est l'équivalent de Composer en PHP, Maven en Java, et NuGet en C#.

L'outil s'appelle Node Package Manager car il a été inventé au départ pour Node.js. Mais aujourd'hui, npm est devenu le package manager pour tout l'écosystème JavaScript, y compris hors Node.js. Il suffit donc d'installer Node pour installer npm . [18]

#### **4. Cas d'utilisation de la pile MERN :**

Comme toute pile Web, on peut créer tout ce qu'on veut dans MERN, même si elle est parfaitement adaptée aux cas lourds JSON, cloud natifs et dotés d'interfaces Web dynamiques. Quelques exemples pourraient être:

- Gestion des flux de travail
- Agrégation de nouvelles
- Applications et calendriers Todo
- Forums interactifs / produits sociaux. [19]

#### **5. Pourquoi choisir la pile MERN?**

Mongo DB, la base de données de documents à la racine de la pile MERN a été conçue pour stocker les données JSON de manière native (elle utilise techniquement une version binaire de JSON appelée BSON ), et tout, de son interface de ligne de commande à son langage de requête (MQL, ou Mongo DB Query Language), sont construits sur JSON et JavaScript.

Mongo DB fonctionne extrêmement bien avec Node.js, et facilite le stockage, la manipulation et la représentation des données JSON à chaque niveau de l'application. Pour les applications Cloud-natives, Mongo DB Atlas facilite encore plus la tâche en offrant un cluster Mongo DB mise à l'échelle automatique sur le fournisseur Cloud choisis, en quelques clics.

Express.js (fonctionnant sur Node.js) et React.js rendent l'application JavaScript / JSON MERN complète. La combinaison signifie que les données JSON circulent naturellement de l'avant vers l'arrière, ce qui les rend rapides à construire et raisonnablement simples à déboguer. De plus, on a besoin que d'un seul langage de programmation et de la structure du document JSON pour comprendre l'ensemble du système. [19]

## 6. Captures d'écran :

### 6.1. Page d'accueil :

C'est la première page qui s'affiche après avoir lancé l'application, c'est la page principale de notre site sur laquelle l'internaute visiteur se fait sa première impression sur notre entreprise (GTS développement).elle renseigne sur le contenu de notre application « Plateforme de suivi de chantier ».

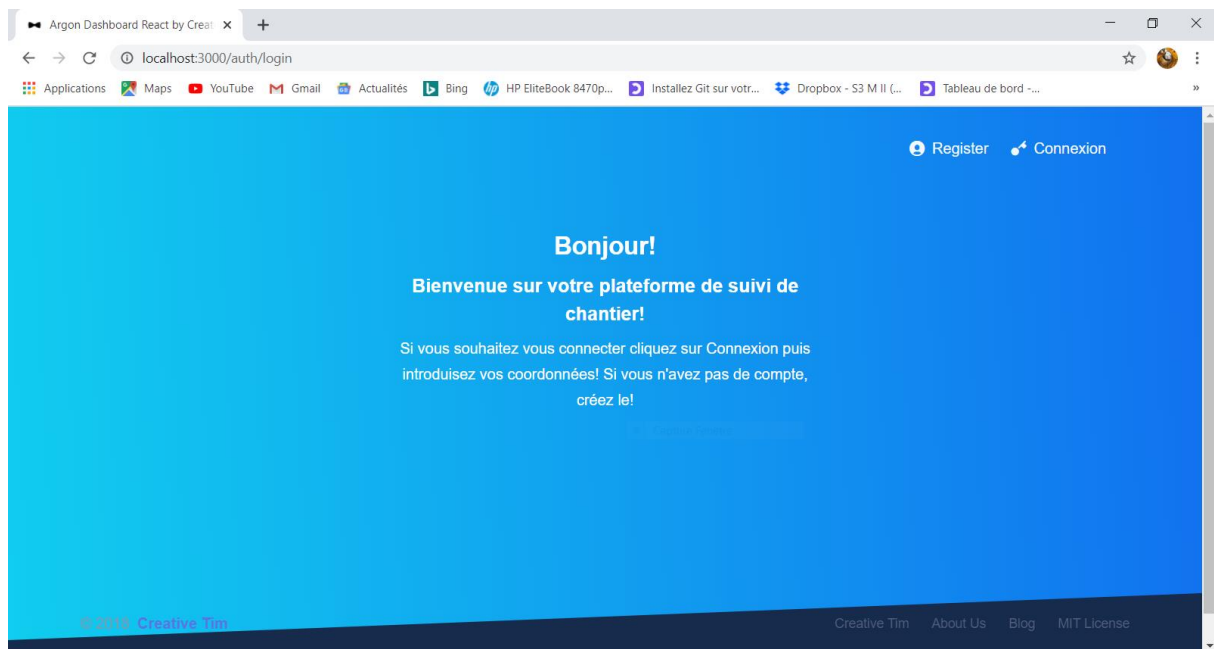


Figure 19:Page d'accueil

### 6.2. Tableau de bord :

C'est la page principale de l'application où sont affichées les informations les plus essentielles. Le tableau de bord désigne la représentation visuelle des informations essentielles concernant notre projet « suivi de chantier ». Il permet donc de mesurer la performance de l'entreprise et d'évaluer son organisation en vue d'atteindre les objectifs bien définis.

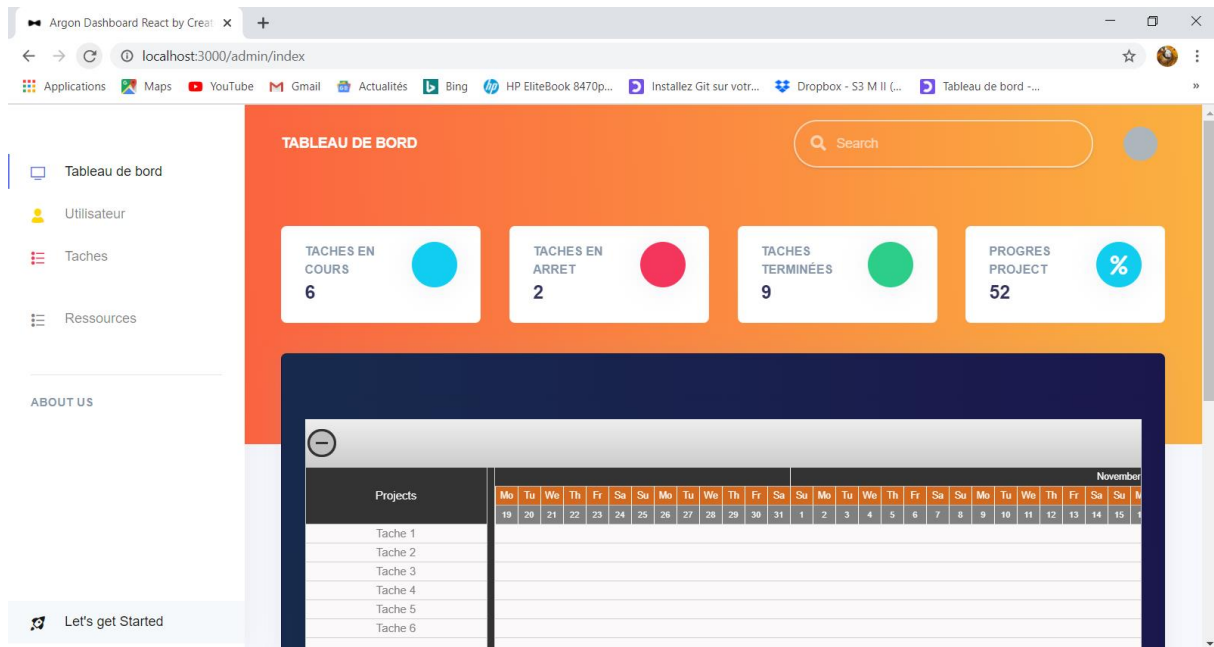


Figure 20:Tableau de bord

### 6.3. Gantt :

Le diagramme de Gantt permet de représenter graphiquement l'état d'avancement des différentes tâches qui constituent le projet ainsi que leurs durées. C'est le planning général de notre projet.

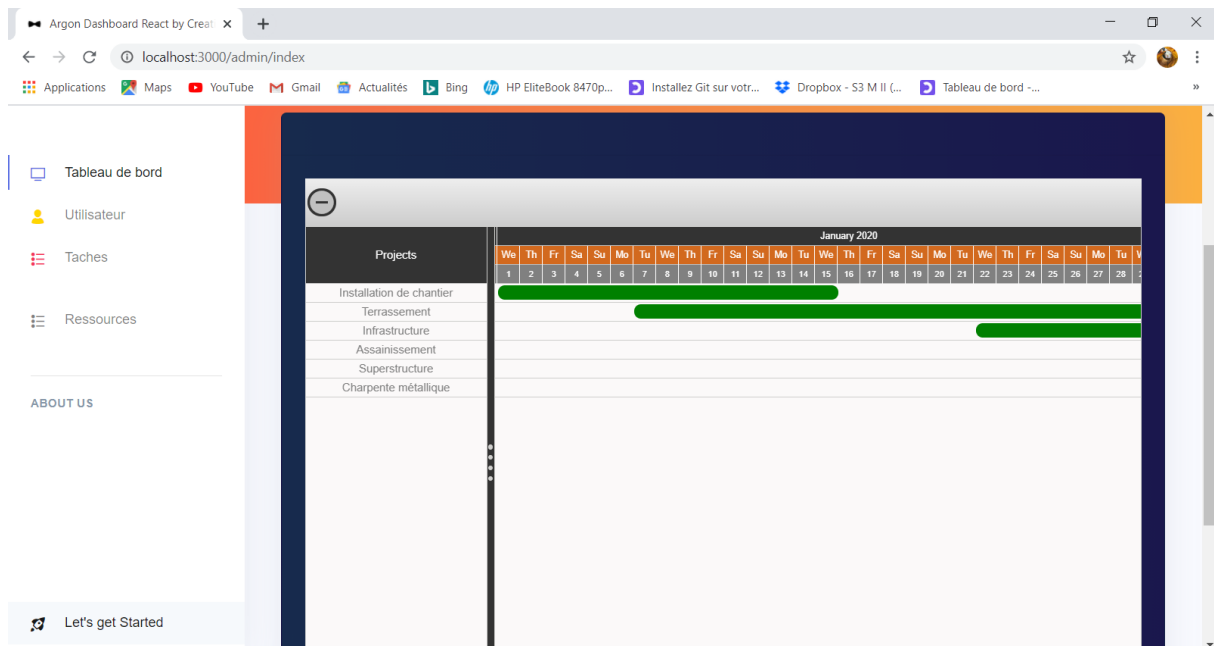
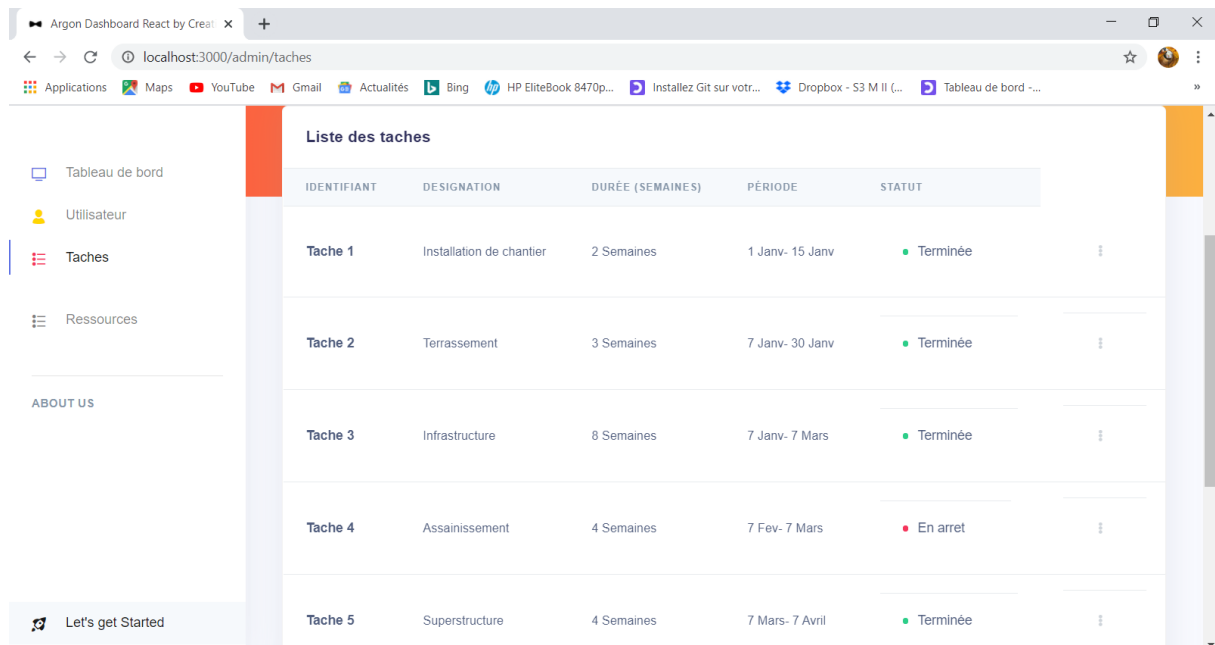


Figure 21:Gantt

## 6.4. Liste des tâches :

Est un tableau contenant les différentes tâches qui constituent le projet ainsi que toutes les informations nécessaires les concernant en précisant celles qui sont terminées, celles qui sont en arrêt et celles qui sont en cours d'exécution.



IDENTIFIANT	DESIGNATION	DURÉE (SEMAINES)	PÉRIODE	STATUT
Tache 1	Installation de chantier	2 Semaines	1 Janv- 15 Janv	Terminée
Tache 2	Terrassement	3 Semaines	7 Janv- 30 Janv	Terminée
Tache 3	Infrastructure	8 Semaines	7 Janv- 7 Mars	Terminée
Tache 4	Assainissement	4 Semaines	7 Fev- 7 Mars	En arrêt
Tache 5	Superstructure	4 Semaines	7 Mars- 7 Avril	Terminée

Figure 22:Liste des tâches

## 6.5. Détails tâche :

En cliquant sur le bouton Détails une fenêtre s'affiche .elle représente toutes les informations concernant le nombre de ressources humaines ainsi que le type et la quantité matérielle utilisées.



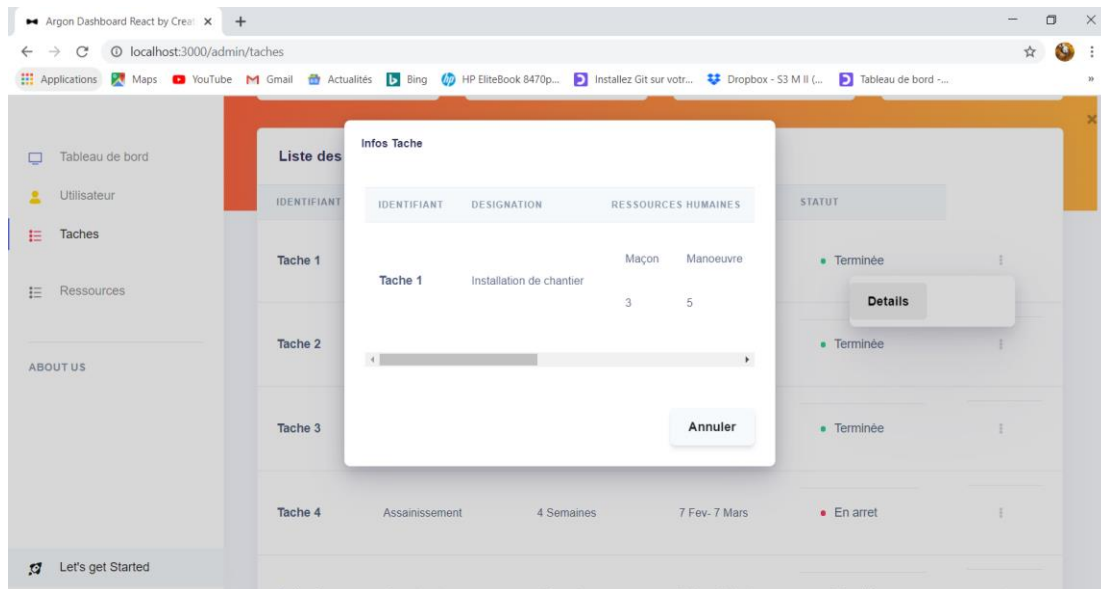


Figure 23:Détails taches

## 6.6. Liste des ressources :

Est un tableau contenant les détails sur les types et les quantités des ressources humaines, matérielles et matériaux utilisées durant tout le projet.

Ressources humaines	Nombre	Ressources Materiaux	Volume/Quantité	Ressources Matérielles	Quantité
Chef de projet	1	Ciment	47 tonnes	Bulldozer	1
Chef de chantier	1	Sable	70 m³	Pelle hydraulique	1
Conducteur de travaux	1	Graviers	140 m³	Niveleuse	1
Topographe	1	Ronds à béton	23 tonnes	Compacteur pneumatique	1
Metreur verificateur	1	Bois de coffrage	17.70 m³	Compacteur cylindre	1
Conducteur rétro-chargeur	1	Charpente metalique	271 tonnes	Grue mobile	1
Conducteur de camion	4	Parpaing	13000 unités	Camion à benne	2

Figure 24:Liste des ressources.

## 6.7. Espace utilisateur :

Est l'espace réservé à l'utilisateur de notre système qui peut consulter ses informations personnelles et également modifier son profil.

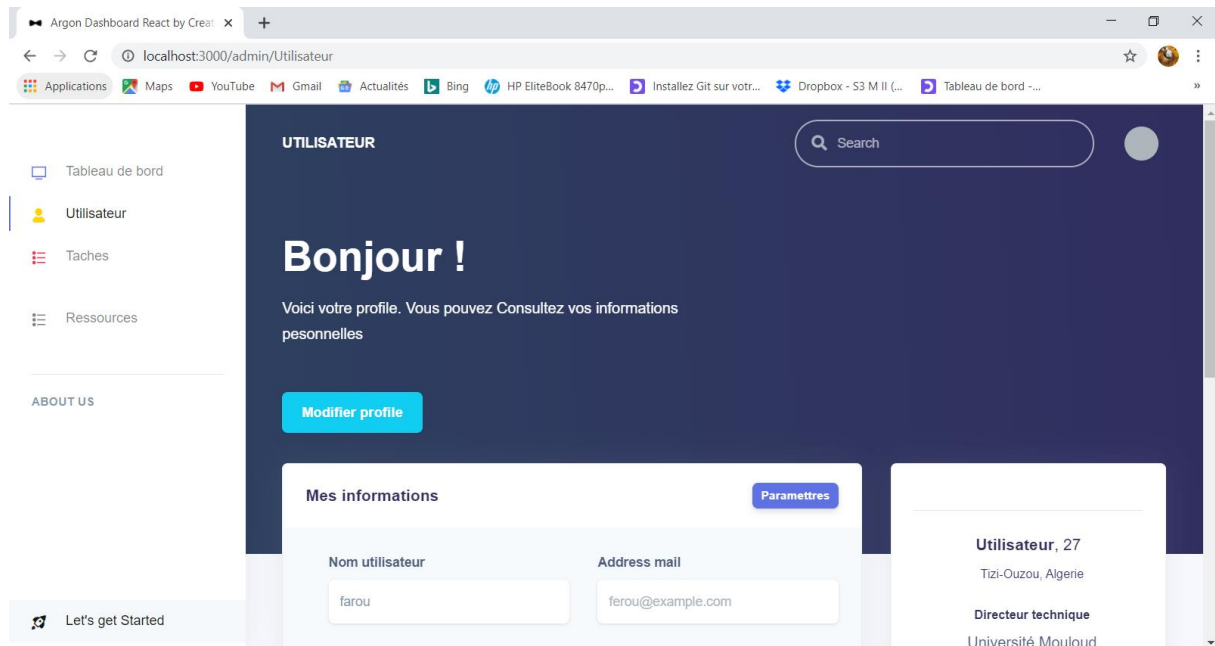


Figure 25:Espace utilisateur

## Conclusion

Nous sommes arrivés à l'étape cruciale de notre projet qui s'avère indispensable ,du fait qu'elle nous a permis de concrétiser nos exigences et atteindre les objectifs imposés durant les phases précédentes .

Pour ce faire, nous avons eu besoin d'un ensemble d'outils de développement afin de concevoir notre application et par la même proposer une solution informatique pour la gestion et le suivi des projets.

## **Conclusion générale**

Notre projet de fin d'études a eu pour objectif : la conception et la réalisation d'une application de gestion et de suivi des projets de l'entreprise .à ce stade de réalisation, on a pu développer une solution informatique optimale.

Pour ce faire, nous avons commencé dans un premier lieu par comprendre le contexte général de notre application et identifier les différents objectifs et exigences. Ensuite, une étude détaillée du cahier des charges afin de comprendre les besoins des utilisateurs du système. Par la suite, nous avons commencé à développer les différentes vues des objets.

Ce que nous tirons de ce mémoire est que le processus d'informatisation permet non seulement d'améliorer les conditions de travail mais aussi de le promouvoir en terme de quantité et de qualité.

Finalement, cette expérience nous a offert des opportunités d'intégrer l'environnement de l'entreprise et améliorer nos capacités dans notre vie professionnelle tout en enrichissant nos connaissances.

## Bibliographie

- [1] Abdelkrim, D. L. *ORGANISATION DE CHANTIER ET GESTION DE PROJET* .
- [2] <https://methodesbtp.com/articles/le-planning/>
- [3] <https://fr.wikipedia.org/wiki/Construction>.
- [4] <https://www.nutcache.com/fr/blog/pourquoi-planifier-un-projet-objectifs-de-la-planification/>.
- [5] <https://espeduc.net/orientation/profession/bpe>.
- [6] <https://www.divalto.com/definition-logiciel-erp/>
- [7] <https://www.choisirmonerp.com/erp/definition-d-un-erp>
- [8] <https://fablain.developpez.com/tutoriel/presenterp/>  
JournalduNet.com.
- [9] <https://www.kpulse.fr/blog/conseils-pratiques/qu-est-ce-qu-un-erp-et-pourquoi-en-avez-vous-besoin>
- [10] <https://web.maths.unsw.edu.au/~lafaye/CCM/merise/concintro.htm>
- [11] <https://ineumann.developpez.com/tutoriels/merise/initiation-merise/>
- [12] <http://spiralconnect.univ-lyon1.fr/webapp/page/page.html?id=9877>
- [13] <https://fr.wikipedia.org/wiki/Node.js>
- [14] responsive-mind.fr. (s.d.). *cycle-de-developpement-informatique*.
- [15] ionos.fr/digitalguide. (s.d.).
- [16] <https://medium.com/@axel.marciano/votre-premi%C3%A8re-application-en-react-node-express-mongodb-5ab0dc531091>.
- [17] <https://atinux.developpez.com/tutoriels/javascript/mongodb-nodejs-mongoose/>
- [18] <https://formationjavascript.com/guides/npm-node-package-manager/>

[19] <https://www.mongodb.com/mern-stack>