Une image contenant texte

Description générée automatiquement 

Rapport de Projet de fin d’études

Présenté pour l’obtention du

Diplôme National de Licence Fondamentale

Mention Sciences de l’Informatique

*Thème :*

**Gestion des actifs au sein de DATACENTER**

*Réalisé par :*

**Anis Tagoug**

Encadrant (s) : M. Marwouene kachroudi– Université centrale

M. Markrem mannai- Tunisie Telecom

*Travail proposé et réalisé en collaboration avec*

Tunisie Telecom

Année universitaire : 2020/2021

# Présentation du cadre du projet

## Introduction

Ce chapitre a pour objectif de situer notre projet dans son contexte général. Ainsi, nous commençons par la présentation de l’organisme d’accueil. Ensuite, nous décrivons brièvement le sujet, les objectifs à atteindre et la méthodologie de travail adoptée.

## Présentation de l’organisme d’accueil :

Le présent projet est réalisé dans le cadre de la préparation d’un mémoire de fin d’études présenté en vue de l’obtention de License fondamentale de sciences de l’informatique pour l’année 2020/2021 . Nous présentons dans ce paragraphe l’organisme d’accueil ainsi que ses secteurs d’activité

### Fiche d’identité : \*

### Une image contenant flèche Description générée automatiquement

* L’office national des télécommunications est créé suite à la promulgation de la loi N°36 du 17 avril 1995.  
  L’office a ensuite changé de statut juridique, en vertu du décret N°30 du 5 avril 2004, pour devenir une société anonyme dénommée « Tunisie Telecom ».
* En juillet 2006, il a été procédé à l’ouverture du capital de Tunisie Telecom à hauteur de 35% en faveur du consortium émirati TeCom-DIG.  
  Cette opération vise à améliorer la rentabilité de Tunisie Telecom et à lui permettre de se hisser parmi les grands opérateurs internationaux.
* Depuis sa création, Tunisie Telecom œuvre à consolider l’infrastructure des télécoms en Tunisie, à améliorer le taux de couverture et à renforcer sa compétitivité.  
  Elle contribue également, activement à la promotion de l’usage des TIC et au développement des sociétés innovantes dans le domaine des télécoms.
* Pionnière du secteur des télécoms en Tunisie, Tunisie Telecom a établi un ensemble de valeurs définitoires qui place le client au centre de ses priorités.  
  L’adoption de ces valeurs se traduit en particulier par une amélioration continue des standards de l’entreprise et de la qualité des services.
* Tunisie Telecom compte dans ses rangs plus de 6 millions d’abonnés dans la téléphonie fixe et mobile.

### Tunisie Telecom se compose de 24 directions régionales, de 140 Espaces TT et points de vente et de plus de 13 mille points de vente privés. Elle emploie plus de 6000 agents.

### Organigrammes Du Tunisie Telecom :

### 

Figure : organigrammes Tunisie Telecom

Tunisie Telecom comprend vingt-quatre Directions Régionales disposant chacune de structures opérationnelles et de fonctions supports propres.

### Secteurs d’activités :

**Tunisie Télécom** propose des services dans le domaine des **télécommunications** fixes et mobiles. En juin 2006, il est fort de 1 259 000 abonnés au réseau fixe (RTCP), dont il détient le monopole, et de

3 265 000 abonnés au réseau GSM, la première ligne ayant été inaugurée le 20 mars 1998.



## Présentation du sujet :

Mon projet intitulé « Gestion des actifs au sein de DATACENTER » consiste à la manipulation et la réalisation d’une application web permettant d’offrir les principales fonctions d’une gestion efficace et simplifiée des composants du DATACENTER.

## Choix méthodologique :

Dans cette section nous présentons la méthodologie adoptée, le cycle de développement et  le formalisme de conception.

### La méthodologie Scrum :

Nous avons choisi la méthodologie Scrum pour la conception et le développement de notre application.

1. **Introduction à la méthodologie Scrum :**

Le terme Scrum (qui signifie « mêlée » en rugby) se rapproche plus d’une gestion de ressources humaines plutôt que d’une réelle méthode de développement. Il s’agit ici de ne pas oublier le côté humain du développement.

Les principales caractéristiques de Scrum sont :

* + - * + Identifier les changements très tôt.
        + Donner toute confiance aux développeurs et les laisser faire leur travail sans pression ni contrainte.
        + Faire des itérations variantes (généralement de 30 jours), appelées aussi « sprints » pour laisser le temps de coder. Chaque itération a un objectif bien précis ou « backlog » et fournit une nouvelle fonctionnalité testée (une démonstration est faite à la fin de chaque sprint).
        + Faire des réunions tous les jours (Daily meeting) et chaque semaine (Weekly meeting) pour encadrer les équipes et recaler les objectifs.
        1. **Le choix de la méthodologie Scrum :**

Outre la méthode formelle, Scrum, peut être vue comme un Framework méthodologique dont l’implémentation doit être ajustée en fonction des caractéristiques techniques, organisationnelles et culturelles des projets qu’on souhaite les mettre en œuvre (Scrum, au demeurant, ne limite pas son champ d’application aux seuls projets informatiques : ses principes sont applicables pour toute autre activité visant à produire un résultat)

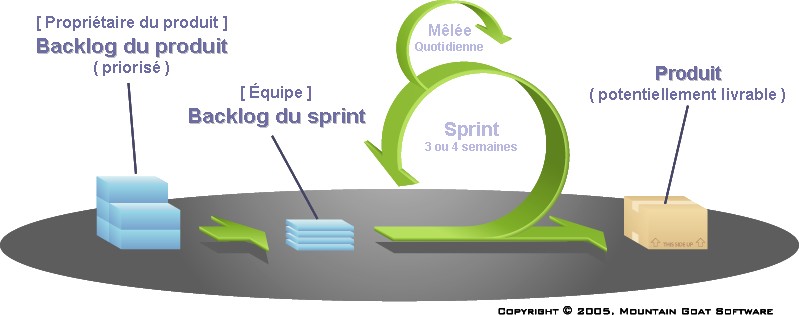


FIGURE 1.1 – Schéma illustratif de SCRUM

Dans ses grandes lignes, Scrum définit un jeu minimal d’acteurs, de cérémonies et d’artefacts qui permettent de relever les défis principaux du développement incrémental : la planification, la gestion du temps et la gestion des obstacles. Scrum est entièrement piloté par la Valeur Métier.

La gestion des risques, en particulier, est réalisée à travers ce prisme. Scrum identifie trois acteurs :

* **Le Product Owner** ((Directeur de Produit), qui possède l’expertise fonctionnelle et est à

même de réaliser les arbitrages nécessaires à la priorisation des développements. Son rôle est absolument essentiel et son respect des règles du jeu est la pierre angulaire du succès d’un projet agile.

* **Membre de l’Equipe**, et dont la tâche principale est d’optimiser la capacité de production

de l’Equipe en l’aidant à travailler de façon autonome et à s’améliorer constamment. Il est également le garant de la bonne implémentation de Scrum.

* **L’Equipe, dont la taille doit être réduite**, et qui prend en charge le développement du

Produit (planification, conception, codage, tests, documentation) sans spécialisation des rôles. La particularité d’une Equipe Scrum est d’être « auto-organisée », et donc dépourvue de hiérarchie. Cet aspect constitue une rupture radicale avec les approches managériales traditionnelles, qui privilégient un contrôle centralisé généralement incarné par le Chef de Projet.

Les avantages cités ci-dessus se révèlent particulièrement, bien adaptés à mon projet de fin d’études dont les objectifs et la limitation temporelle sont parfaitement délimités et connus. La pratique de la méthode Scrum nous a donné l’opportunité d’être intégré au sein de ce processus et de participer aux cycles de développement. Après avoir fait le choix de la méthodologie, qui est une étape primordiale dans le cycle de développement d’un produit informatique, nous exposerons dans le paragraphe suivant la problématique a laquelle nous sommes amenés à développer une solution.

### Formalisme adopté

UML (Unified Modeling Language) est un langage de modélisation graphique orienté objet de troisième génération à base de pictogrammes. Dans le cadre de notre spécification, j’ai opté pour le choix UML parce que cette méthodologie offre les avantages suivants :

- la possibilité de modification

-la réutilisabilité

-la modularité.

Telles sont les qualités reconnues de cette approche.

UML fournit un moyen astucieux permettant de représenter diverses projections d’une même représentation grâce à ses différents diagrammes. En effet, il couvre l’aspect statique et dynamique d’un système selon ses différents diagrammes. Il définit pour cela dix diagrammes qui sont subdivisés en des vues statiques (qui représentent « physiquement » le système à modéliser au moyen de diagrammes d’objets, de classes, de paquetages, de cas d’utilisation, de composants, de déploiement et enfin d’architecture) et des vues dynamiques (qui traduisent le fonctionnement du système au moyen de diagrammes de séquences, de collaboration, d’états de transitions et d’activités).

Pour éviter de surcharger le rapport et d’entrer dans les fins détails techniques, nous nous ne contenterons de présenter quelques diagrammes que nous avons jugés utiles pour comprendre le projet à savoir :

* Le diagramme des cas d’utilisation : Il permet de structurer les besoins des utilisateurs et les objectifs correspondants de notre système en identifiant ses utilisateurs et leurs interactions,
* Le diagramme des séquences : Il permet une représentation temporelle des objets et de leurs interactions,
* Le diagramme des classes : Il permet de présenter les classes et les interfaces de notre système ainsi que les différentes relations entre celles-ci.

## Conclusion

Ce chapitre a donné l’occasion de présenter dans un premier temps la société « Tunisie Telecom » puis le cadre du sujet, ainsi que les objectifs de notre travail vise à atteindre. En vue de suivre un avancement logique dans ce rapport, une étude théorique concernant l’état de l’art fera l’objet du prochain chapitre.

**Chapitre 2**

# Etat de l’art et étude de l’existant

## Introduction

Dans ce chapitre, nous entament une étude concernant le système de gestion des actifs existants adoptés par Tunisie Telecom et ses limites. Puis, nous mettons l’accent sur quelques solutions concurrentielles existantes sur le marché qui peuvent répondre aux besoins de Tunisie Telecom

## Situation Actuelle

La société Tunisie Telecom utilise Excel comme logiciel pour la gestion de ces serveurs …

.

### Excel :

### -introduction

Excel est un outil de calcul ; il comprend de nombreuses fonctions intégrées (scientifiques, financières, statistiques, etc.) et remplace avantageusement une calculatrice. Excel permet d’élaborer des modèles de calcul, et de les réutiliser.

* -fonctionnalités  
  Présentation de données, en tableaux (c’est un tableur).
* Présentation graphique de données (c’est un grapheur).
* Gestion de données : il permet de gérer une « base de données » locale et d’effectuer des opérations de tri, de sélection, de suppression de doublons et d’extraction de données.

### 

### Les Limites des solutions existantes

### les limites de Excel (à rédigé ) :

**1. Excel n'est pas sécurisé.**

La sécurité est l'une des principales raisons pour lesquelles de nombreuses entreprises préfèrent créer une application Web à partir de données Excel plutôt que d'exécuter leurs opérations sur une feuille de calcul.

**2. Excel est sujet aux erreurs.**

Contrairement aux applications Web, qui peuvent être personnalisées pour limiter les vues des utilisateurs sur des données spécifiques, les feuilles de calcul exposent les formules à tout le monde, ce qui peut entraîner des falsifications et des erreurs, intentionnelles ou non.

**3. Les feuilles de calcul Excel sont difficiles à suivre :**

Il y’a de fortes chances que la plupart de vos collègues envoient des feuilles de calcul Excel pour partager des informations. Mais les choses peuvent devenir incontrôlables assez rapidement : trop de versions du fichier, vous ne savez jamais qui a apporté quelles modifications, et l'emplacement de la dernière version est un mystère.

En déplaçant vos données d'Excel vers des applications Web, vous fournissez un accès à plusieurs utilisateurs qui doivent effectuer leurs tâches simultanément. Les utilisateurs n'obtiennent que les vues spécifiques dont ils ont besoin, ce qui leur permet d'effectuer des calculs et de créer des rapports sans altérer l'intégrité des données de base.

**4. Excel est difficile à mettre à l'échelle.**

S'appuyer sur Excel pour effectuer les opérations quotidiennes n'est pas viable pour une entreprise en croissance. Le temps et les efforts nécessaires pour conserver ces documents sont un goulot d'étranglement pour l'efficacité et la productivité du travail.

Le logiciel ne prend pas en charge le partage de données ou les notifications en temps réel, ce qui rend difficile pour les équipes de travailler ensemble et de prendre des mesures immédiates. Les fichiers prennent également une éternité à charger à mesure qu'ils grossissent, donc plus vous avez de données, plus vous avez besoin de temps pour faire avancer les choses.

## Les Solutions disponibles

La recherche logicielle s’est concentrée sur Internet, par le biais de forums ayant déjà évoqué le sujet, du moteur de recherche Google, et du site Wikipédia qui propose une liste de logiciels de gestion de DATACENTER, libres ou propriétaires.

Une fois la liste de logiciels établie, les sites internet concernés sont consultés pour avoir les informations souhaitées. Les démonstrations en ligne permettent également de faire des tests.

### Les critères de recherche

Pour qu’une solution soit retenue, on doit vérifier les critères suivants :

* + - * Ouverture et possibilité de mise à jour (facile à comprendre et à modifier pour l’intégration de nouvelles fonctionnalités ou d’autres mises à jour).
      * Plate-forme remplissant le maximum de tâches.

### Quelques plateformes étudiées

Dans cette partie, nous allons présenter les applications pouvant répondre aux exigences de Tunisie Telecom les plus importantes et les plus satisfaisantes aux critères de recherche.

Manage engine (opManager)

OpManager, solution de confiance

pour l’analyse du réseau

OpManager offre des fonctionnalités complètes d’analyse réseau et permet un suivi global du réseau. Analyse la performance de réseau en temps réel et son contrôle avec une détection proactive des pannes. On peux garantir la continuité d’activité en maintenant une haute disponibilité et gérer des  
environnements informatiques hétérogènes. OpManager offre aussi une évolutivité selon les besoins du réseau sans nuire à la fiabilité.

: Analyse réseau en temps réel :

Analyse et optimisation de la performance du réseau en temps réel avec des tableaux de bord et des graphiques dynamiques personnalisables. Mesure des indicateurs de performance clés et maintien la disponibilité des périphériques.

Analyse des serveurs physiques et virtuels :

Analyse des indicateurs de performance clés comme la disponibilité ou l’utilisation du processeur, de l’espace disque et de la mémoire, et obtention d’un suivi complet des serveurs physiques et virtuels.

Analyse basée sur des seuils

Détection, identification et correction des problèmes du réseau avec des alertes basées sur des seuils. Définition de plusieurs seuils pour chaque indicateur de performance et réception des notifications instantanées.

1-4 : Tableaux de bord personnalisables :

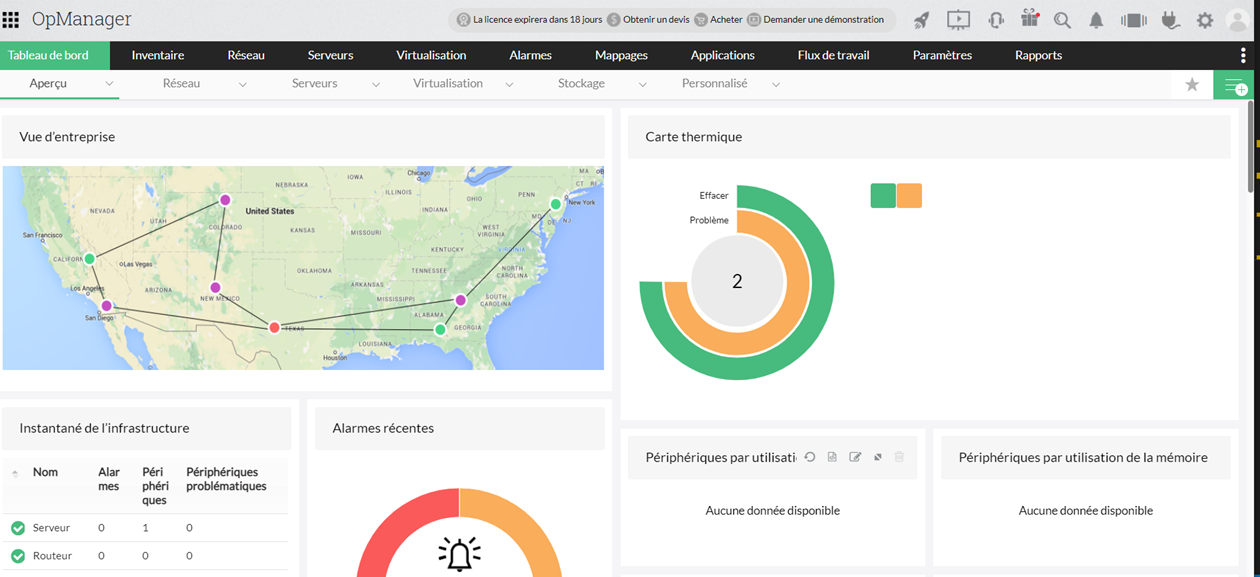
Aperçu synthétique personnalisable de plusieurs éléments stratégiques du réseau, avec des widgets sur mesure et des graphiques de performance en temps réel.

1-5 : Analyse des liaisons WAN :

Analysez des indicateurs clés comme la latence, la gigue, la durée RTT et la perte de paquets pour éliminer les problèmes réseau. Affichage de la performance par saut pour déterminer l’origine de la latence et y remédier rapidement.

1-6 : Facile à configurer et abordable :

OpManager n’exige aucune procédure d’installation complexe et inclut des bases de données et des serveurs Web. La tarification transparente garantit l’absence de coûts cachés



Wattdesign ( 6SigmaAcess) :

6SigmaAccess est une solution simple permettant de visualiser la salle, de localiser un équipement et d’afficher les ressources disponibles

2-1 : Conçue pour de multiples utilisateurs

L’appli web de gestion pour le data center permet à plusieurs utilisateurs de travailler simultanément sur le même modèle. En parallèle, elle met à jour les modèles en temps réel pour minimiser les conflits.

2-2 :  Déploiement intelligent

Déploiement de l’informatique en fonction de la puissance électrique disponible, de l’espace, du refroidissement et de la charge au sol. L’outil donne une réponse sur la disponibilité des ressources afin qu’on puisse prendre de meilleures décisions de déploiement.

2-3 : Application Web

Chargement du modèle 3D du centre de données pour visualiser les assets à l’aide d’un outil Web sécurisé. Ainsi, on planifie et on place les équipements à venir.

2-4 : La simulation des flux d’air en renfort

Grâce à la puissance du moteur d’aide à la décision basé sur l’ingénierie par simulation numérique, 6SigmaAccess permet de prédire l’impact des changements avant leur mise en œuvre

2-5 : Gestion des assets

On oublie les tableaux Excel et les modèles Visio ! on Passe à un modèle 3D complet et gère les informations relatives aux ressources informatiques.

2-6 : Affichage de l’espace, puissance, refroidissement, ports réseau

On visualise facilement l’espace, le refroidissement, la puissance, les ports réseau disponibles au niveau des baies pour comprendre la répartition actuelle des ressources.

2-7 Tableaux de bord :

On exploite les tableaux de bord préconfigurés et personnalisables de puissance électrique, d’espace, de refroidissement, de ports réseau pour analyser l’état du centre de données.

Solution Envisagée :

Conclusion

Ce chapitre nous a permis de préciser le cadre dans lequel se situe ce projet, ainsi que ses objectifs visés. En plus, nous avons eu l’occasion d’étudier des concepts qui seront utilisés dans le cadre du projet. Enfin, nous avons présenté une étude de l’existant pour pouvoir cibler les points importants de l’application. Dans le chapitre suivant, nous allons spécifier les besoins fonctionnels et non fonctionnels de notre application. Nous détaillerons ensuite quelques cas d’utilisation que nous avons pu dégager.

**Chapitre 3 : Spécification et analyse des besoins**

**Intr**oduction

Dans tout système, les fonctionnalités doivent être mises en relation avec un ensemble de besoins utilisateurs. Ces besoins définissent les fonctions que les utilisateurs s’attendent à voir par le système. La spécification des besoins forme les fondations sur lesquelles l’architecture du système est construite. Tout au long de ce chapitre nous allons définir les besoins, identifier les acteurs et les activités par lesquels nous déduisons assez facilement les cas d’utilisation de notre application.

1. Etude des Besoins

L’analyse du sujet nous a permis de cerner les fonctionnalités de la gestion de DATACENTER

* 1. Les Acteurs du système : A voir

L’administrateur : un utilisateur ou un groupe d’utilisateurs définit le privilège dans le site web en ajoutant / enlevant et modifiant l’accès de chaque utilisateur.

L’administrateur a aussi tous les droits de gestion de la plateforme de DATACENTER.

L’Employé :

Ce sont les employés de la société qui ont des droits d’accès bien définis par l’administrateur pour la Plateforme

.

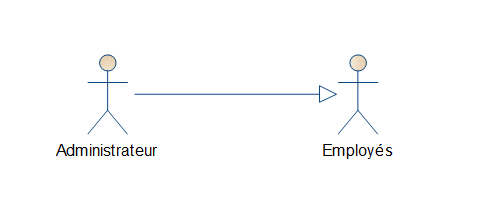


Figure : Les Acteurs de l'application

* 1. Besoins Fonctionnels :

Les besoins fonctionnels doivent répondre aux exigences du système en termes de fonctionnalités. Ils constituent une sorte de contrat par rapport au comportement du système.

* + 1. Les Besoins Fonctionnels pour l’Employé :
* Trouver l’emplacement d’un serveur dans le DATACENTER.
* Lister les emplacements possibles pour rajouter un serveur.
* Gérer les composants du DATACENTER.
* Définir les composants principaux de DATACENTER.
* Faire des rapports précis et une analyse des informations sur les opérations en calculant :
* Taux d’énergie électrique consommé pour le fonctionnement de DATACENTER
* Le degré de refroidissement et d’humidité enregistré par rapport aux seuils tolérés.
* La Pourcentage des serveurs up.
* Le nombre de matériels par constructeur.
* Le nombre de serveurs par système d’exploitation
* Consulté le dashboard
  + 1. Les Besoins Fonctionnels pour l’administrateur :

L’administrateur a les mêmes fonctionnalités de l’employé et il peut gérer l’accès des utilisateur(s).

* 1. Besoins non fonctionnels
* Sécurité :

Seuls les administrateurs pourront consulter les informations personnelles des membres, Les droits d’accès doivent être bien respectés, Le serveur de l’application doit être protégé contre les menaces du web comme « SQL injection ».

* Performance :

-Temps de réponse

– le chargement de l’application, ouverture d’écran et des délais de rafraîchissement.

-En temps de traitement

– fonctions, calculs, importations/exportations de données.  
-L’interrogation de données et rapports

– temps de chargement initial et des chargements suivants.

* Portabilité :

Compatibilité avec diverses plateformes, facilité de remplacement d’autres systèmes en place, facilité d’installation et de désinstallation de l’application.

* Ergonomie :

-Les standards d’ergonomie – la densité d’éléments sur les écrans, la disposition et le flux, les couleurs, l’Interface Utilisateur, les raccourcis clavier.  
- Internationalisation / besoins de localisation – langages, orthographe, claviers, formats de papier, etc.

1. **Les Diagrammes des cas d’utilisation :**

Cette phase a pour objectif de décrire le comportement attendu de l’application. Pour cela, nous nous basons sur les diagrammes des cas d’utilisation qui représentent un élément essentiel de la modélisation UML. Laquelle permet de modéliser les fonctionnalités de l’application du point de vue besoin utilisateur. Elle sert aussi à définir le produit à développer et le modéliser.

* 1. Le cas d’utilisation générale :

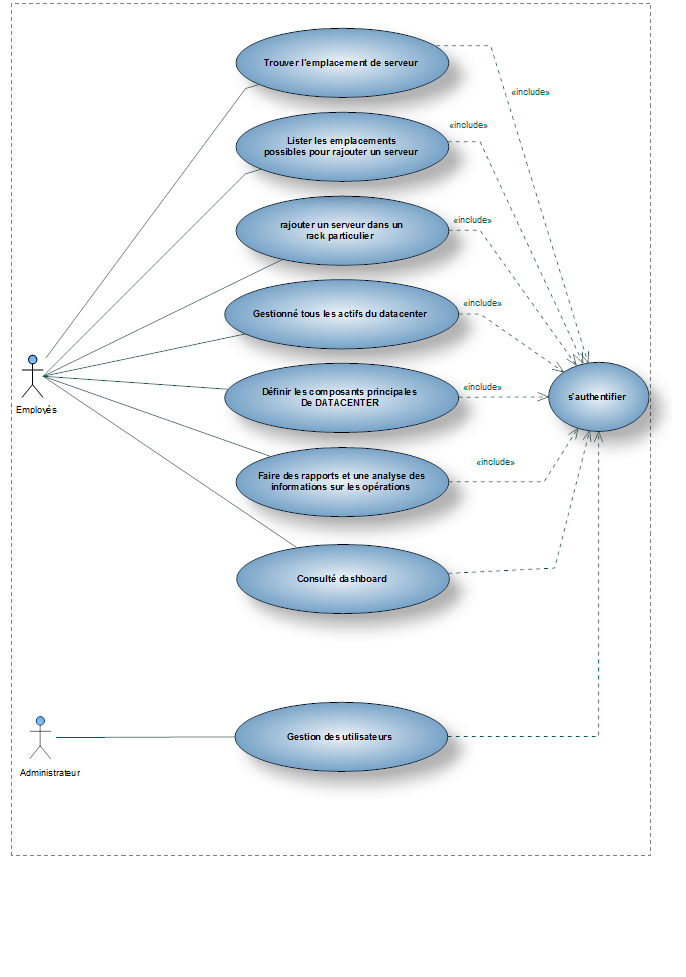


Figure : Digramme de cas d'utilisation générale

La figure ci-dessus présente le diagramme des cas d’utilisation générale. Nous avons vu les différents acteurs du système (administrateur, employé). Nous détaillerons ensuite les cas d’utilisation accessibles pour chaque utilisateur.

* 1. Les cas d’utilisation de l’administrateur :

La figure ci-dessous présente le rôle de l’administrateur de l’application. En effet, il est capable de :

-Trouver l’emplacement de serveur(s).

-Lister les emplacements possibles pour rajouter un serveur.

-rajouter un serveur dans un rack particulier.

-Gérer les composants de DATACENTER.

-Définir les composants principaux de DATACENTER.

-Faire des rapports et une analyse des informations sur les opérations

-Gérer le droit d’accès pour chaque utilisateur.

-Consulté le Dashboard

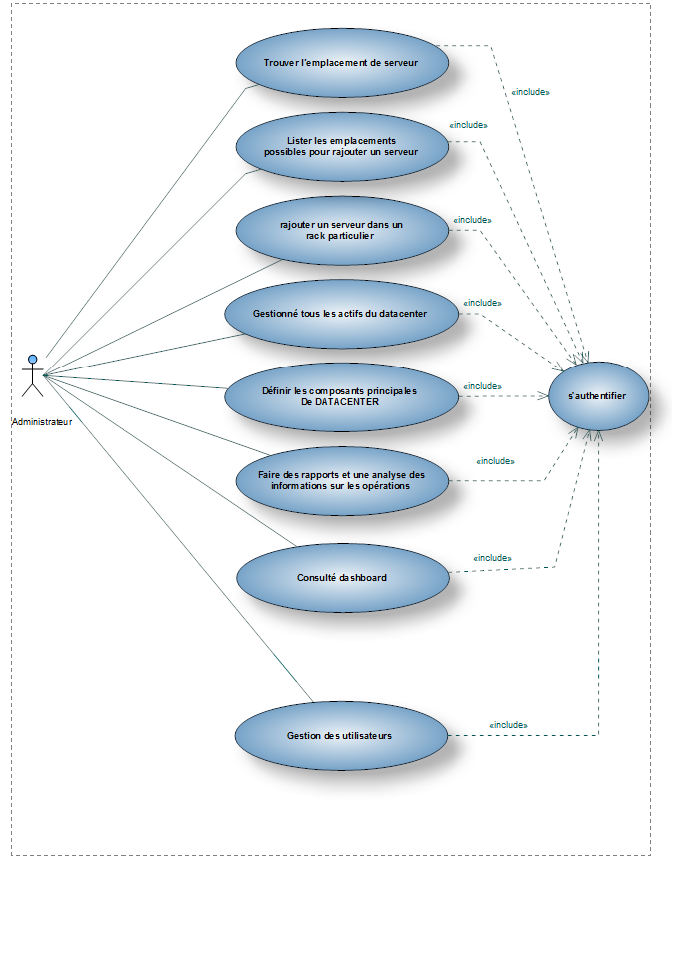


Figure : Le diagramme de cas d'utilisation de l'administrateur

* 1. Le cas d’utilisation de l’employé :

L’employé a presque les mêmes droits que l’administrateur comme le présente la figure ci-dessous où l’employé est capable de :

-Trouver l’emplacement du serveur.

-Lister les emplacements possibles pour rajouter un serveur.

-rajouter un serveur dans un rack particulier

-Gérer les matériels du DATACENTER

-Définir les composants principaux De DATACENTER.

-Consulté le Dashboard

-Faire des rapports précis et une analyse des informations sur les opérations

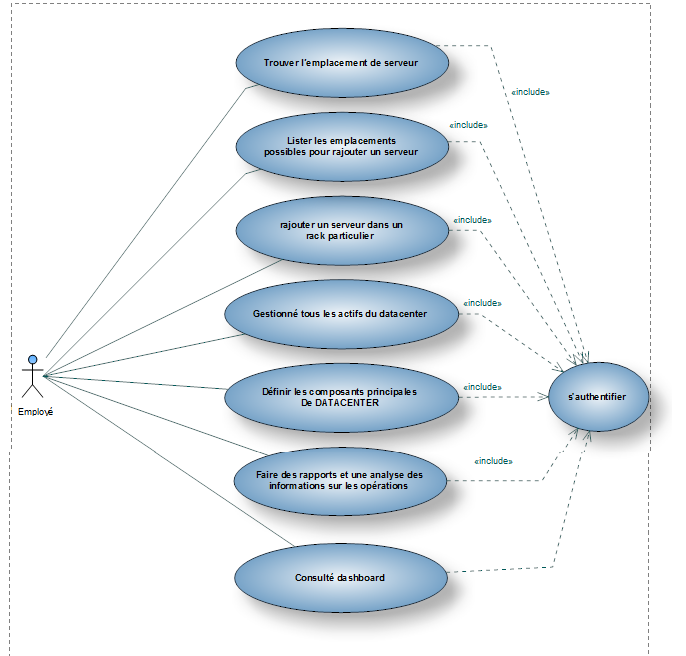


Figure : Le diagramme des cas d'utilisation de l’employé

1. Diagramme des séquences :

Avec les diagrammes de séquences, l’UML fournit un moyen graphique pour représenter les

Interactions entre les objets à travers le temps [N1]. Ces diagrammes montrent typiquement

un acteur, les objets et les acteurs avec lesquels il interagit au cours de l’exécution du cas

d’utilisation. Dans ce paragraphe, nous présentons quelques diagrammes de séquences

permettant de décrire les différentes interactions entre l’utilisateur et l’application .

Diagramme de séquence pour le scénario d’authentification

Avant l’accès à l’application, l’utilisateur doit s’authentifier en introduisant son identifiant et

son mot de passe. Après vérification, si l’utilisateur est accepté, il aura accès au système et les

informations personnelles de sa session serons prise en compte sinon un message d’erreur

s’affichera.

-Une session avec le profil de l’utilisateur est crée. Cette session garde trace du parcours de

l’utilisateur dans le site.

– Renvoyer l’utilisateur vers la page d’authentification pour retaper de nouveau les

informations demandées. L’opération d’identification doit s’effectuer dans un espace sécurisé

pour garantir la confidentialité des informations transmises.

-Si l’utilisateur est un employé de la société, le système le redirige vers l’espace des

Employées, sinon le système le redirige vers l’espace administratif.

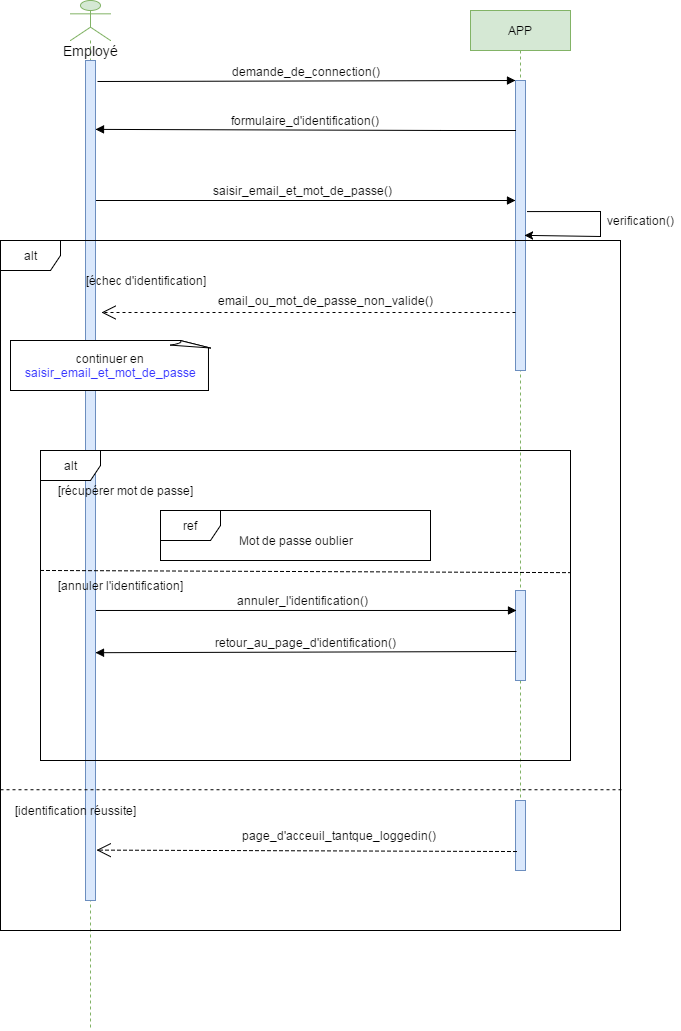
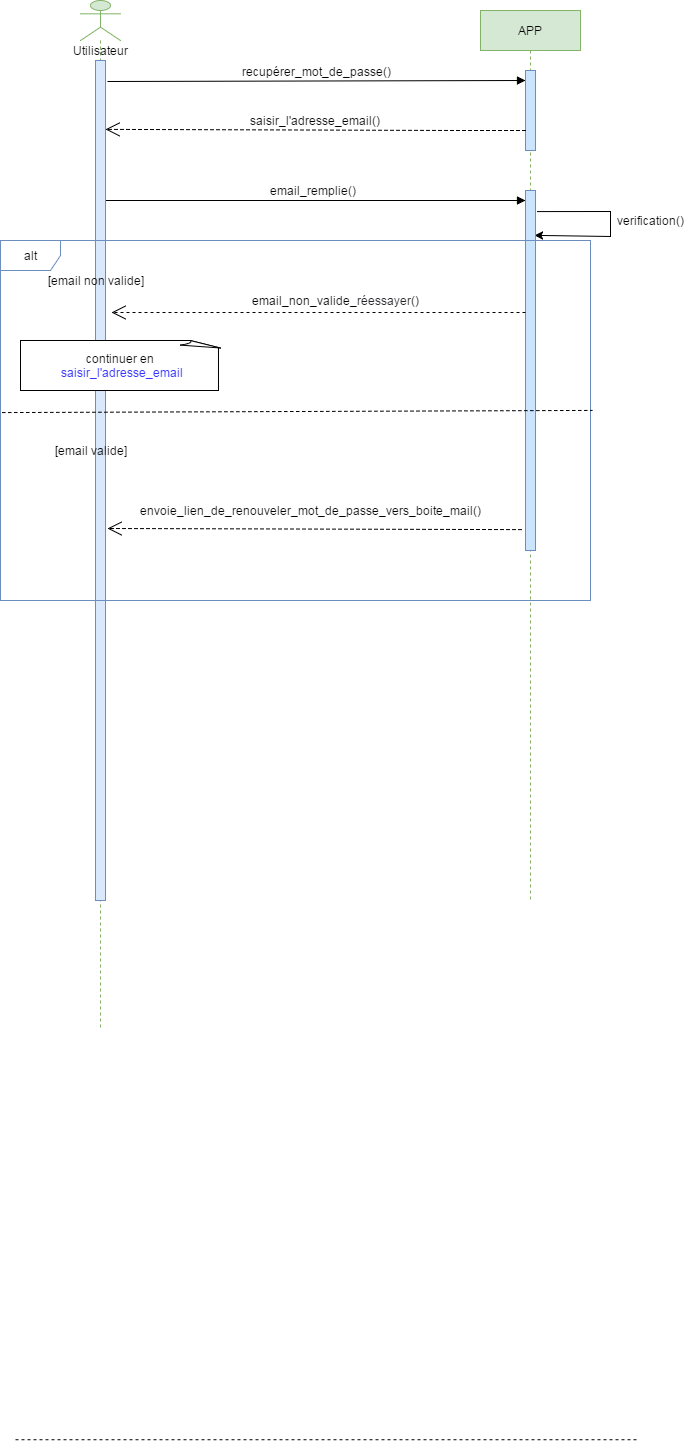
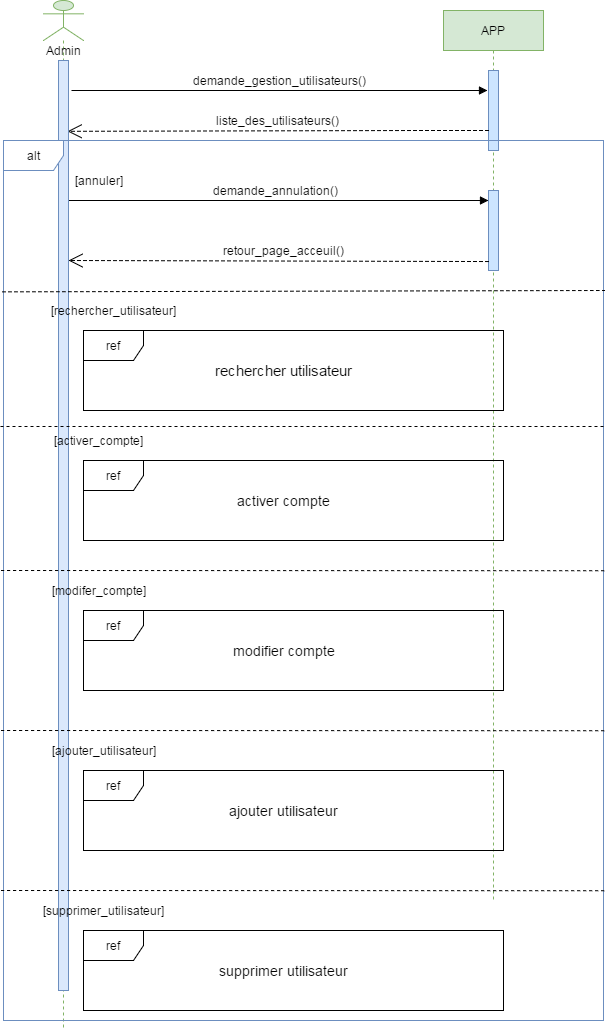
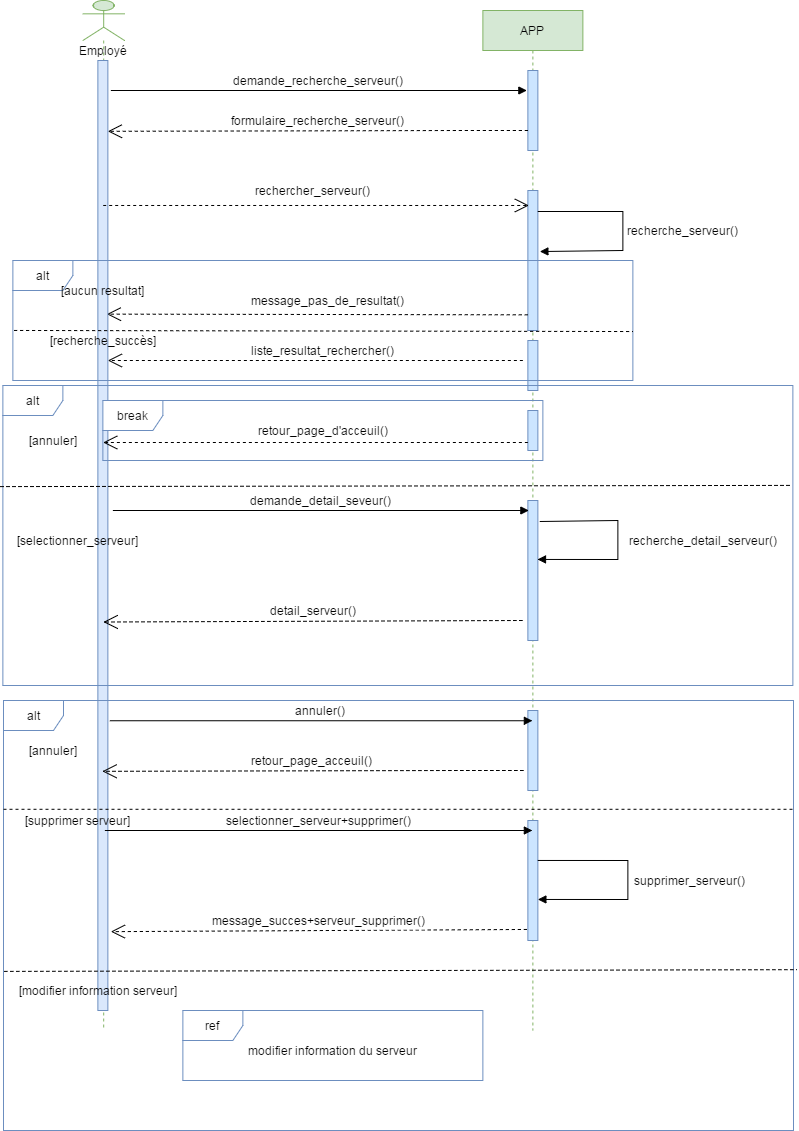
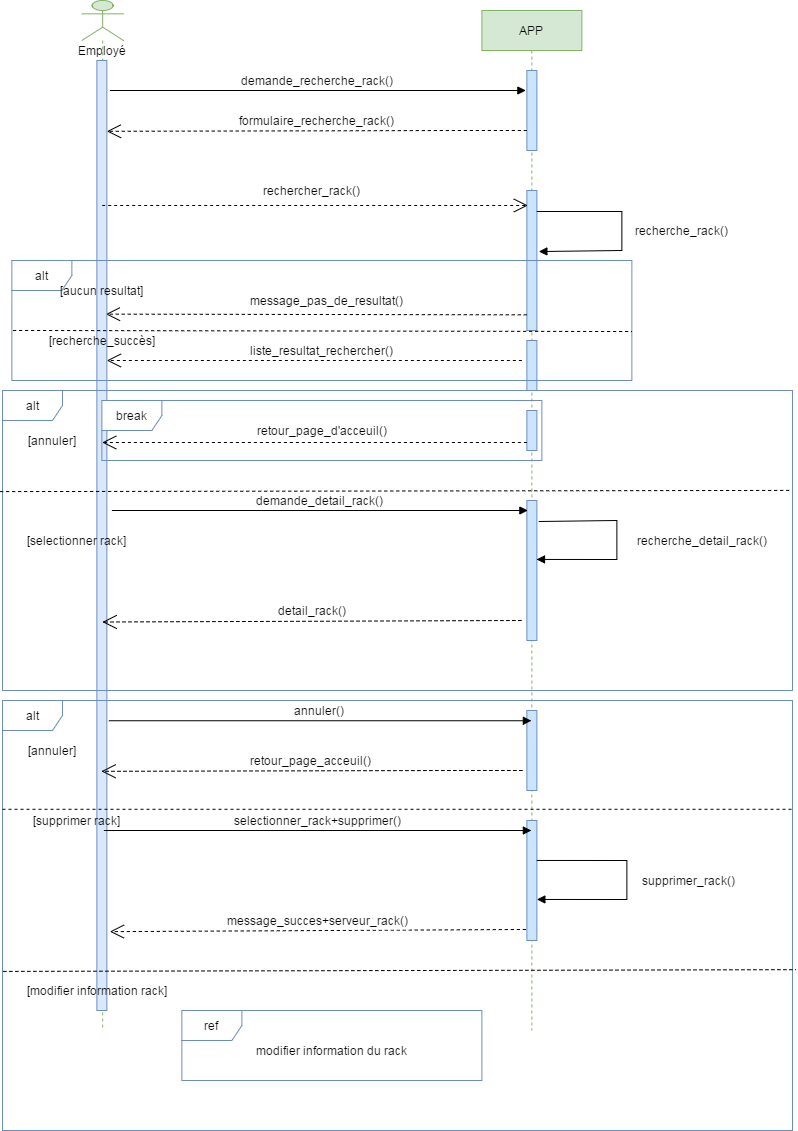
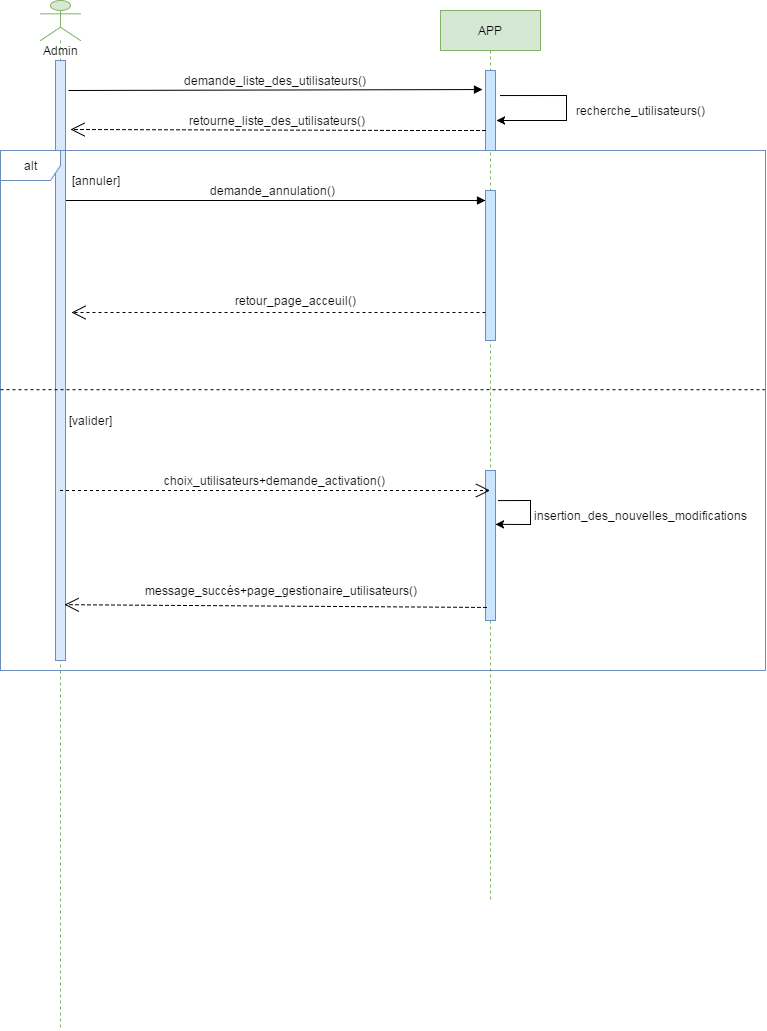
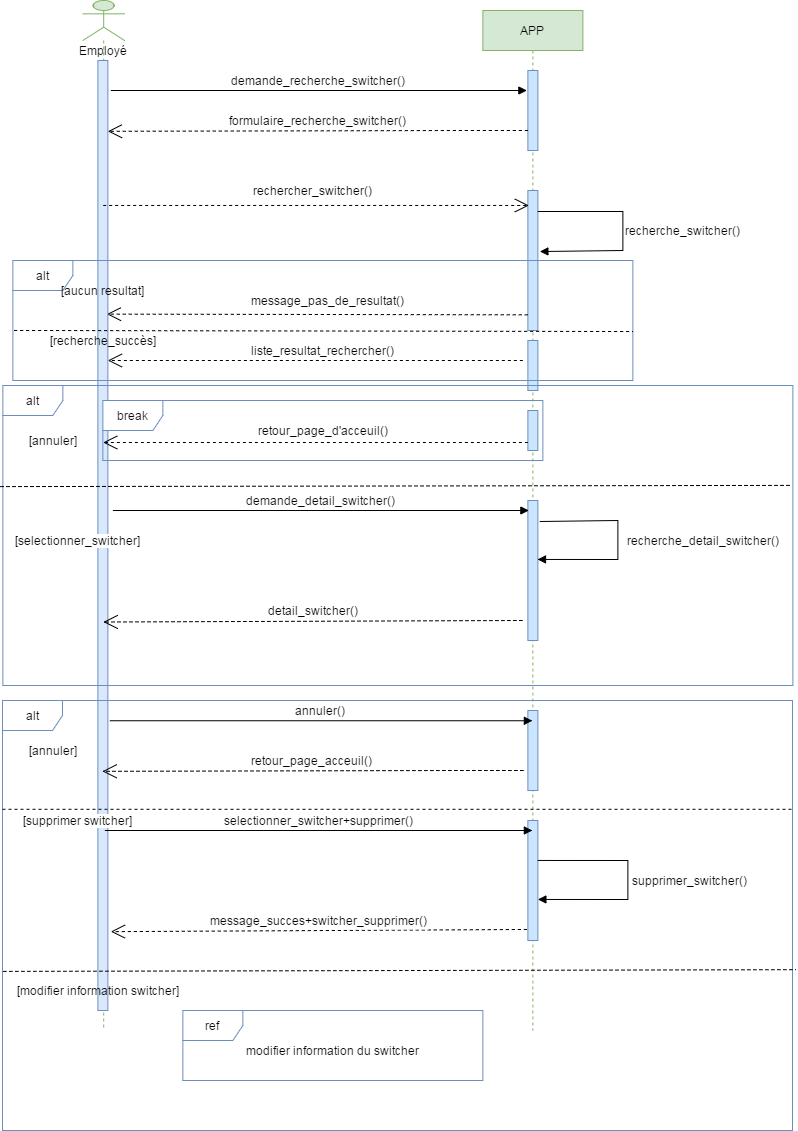
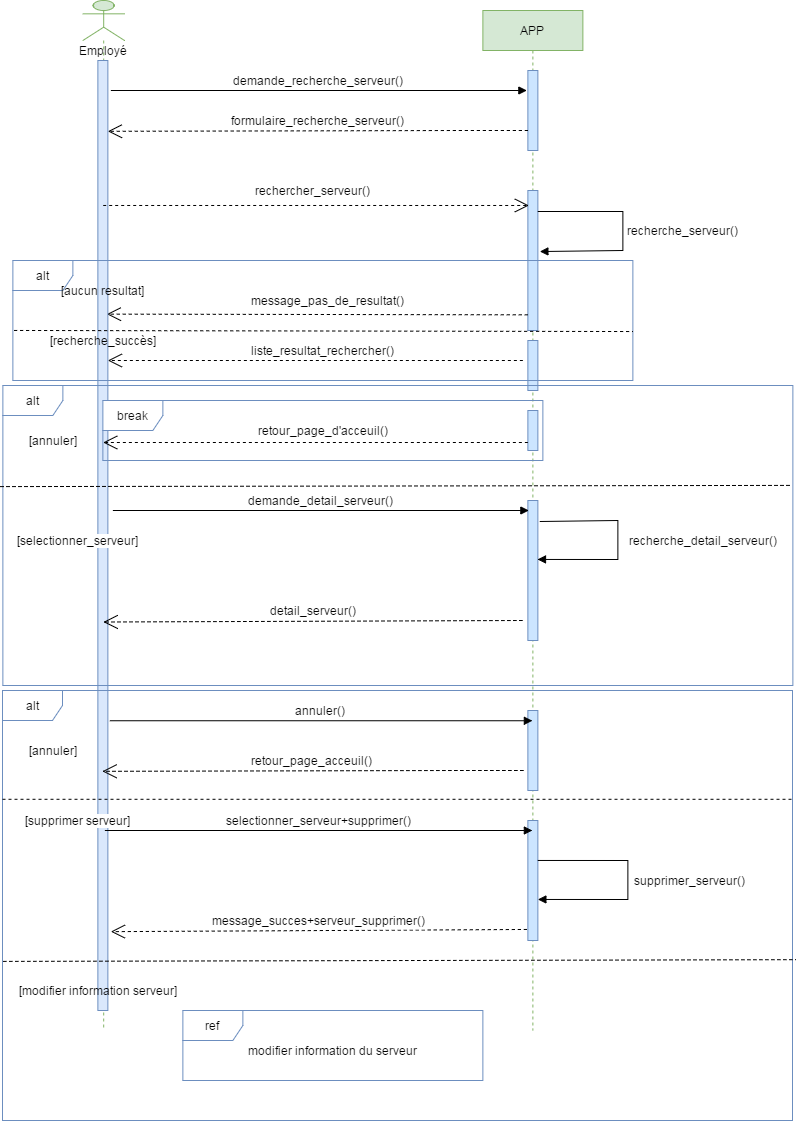
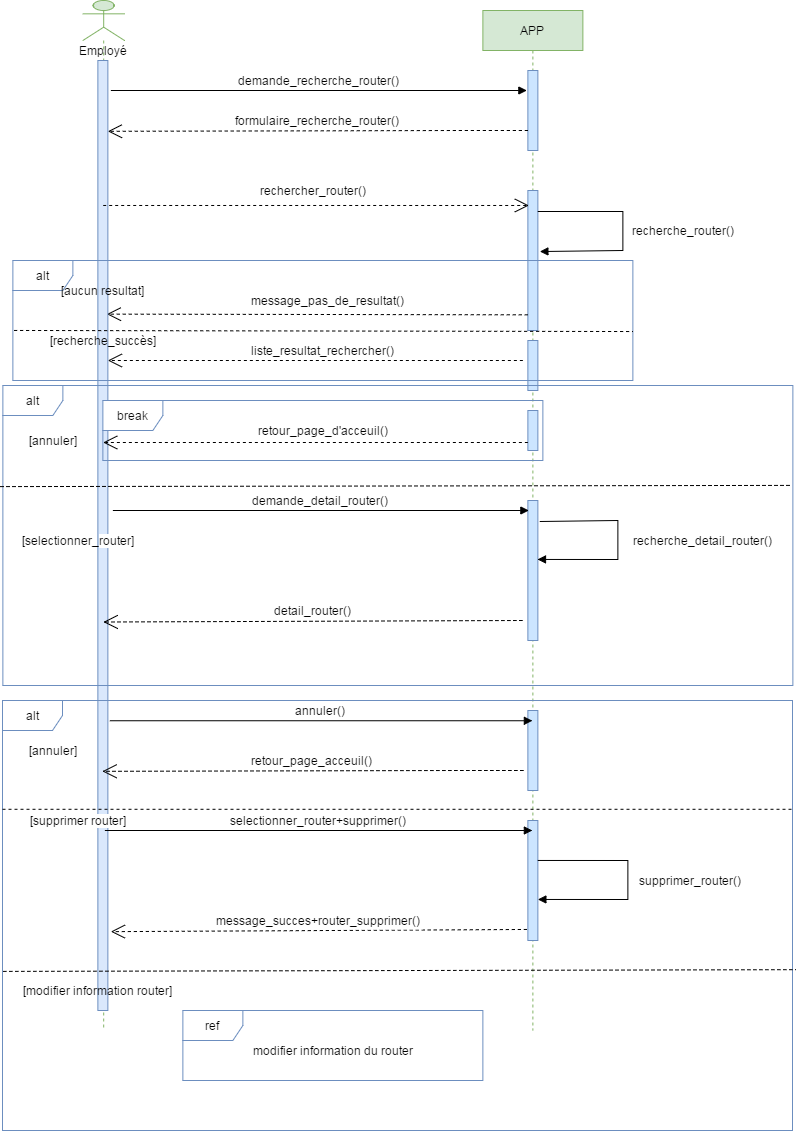
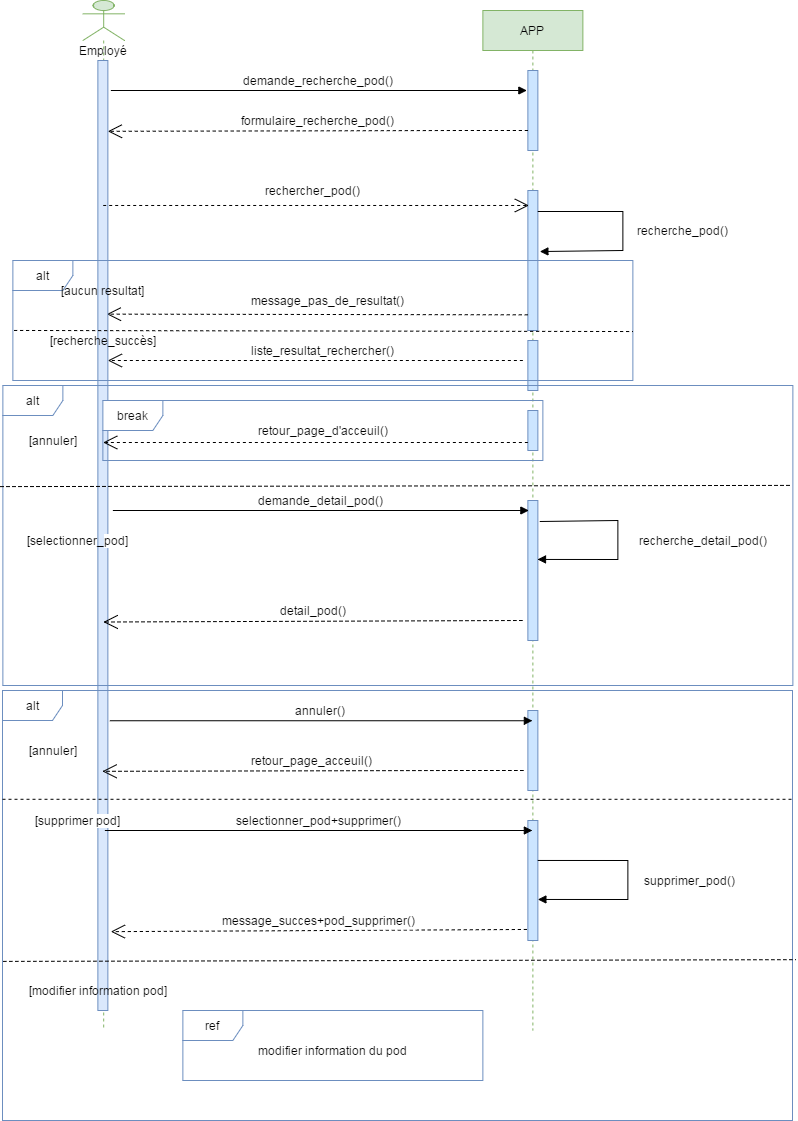
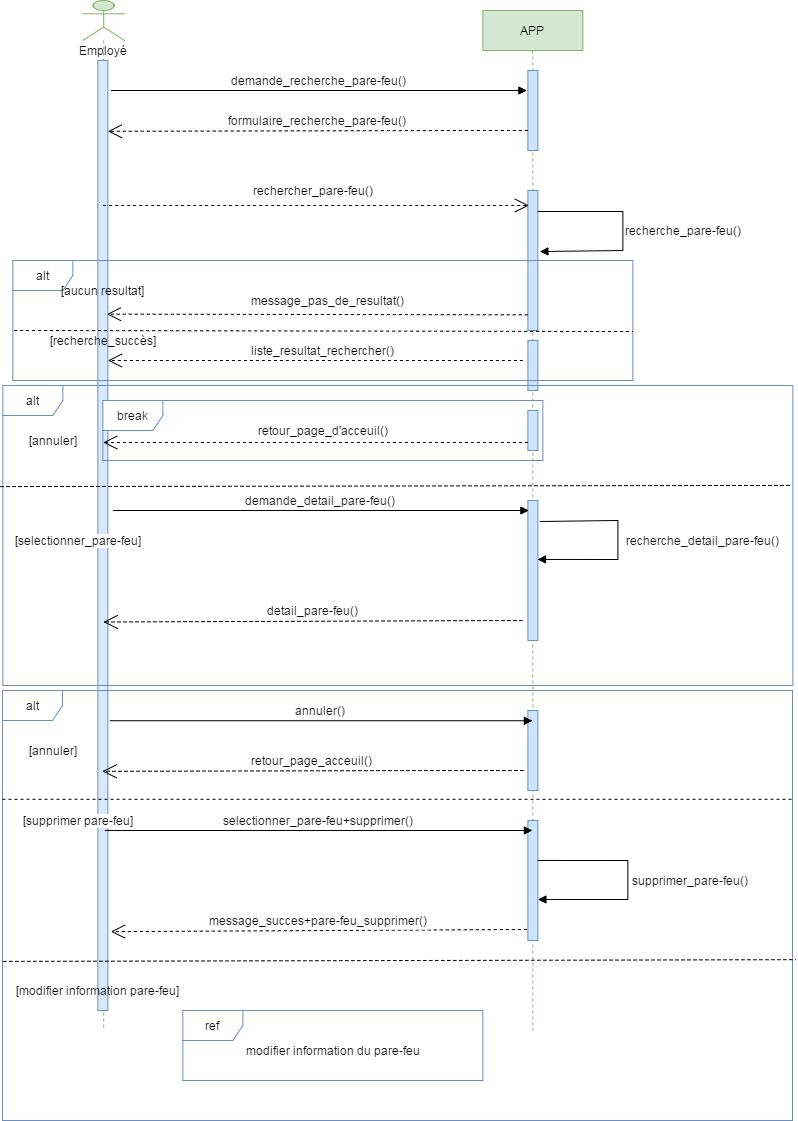
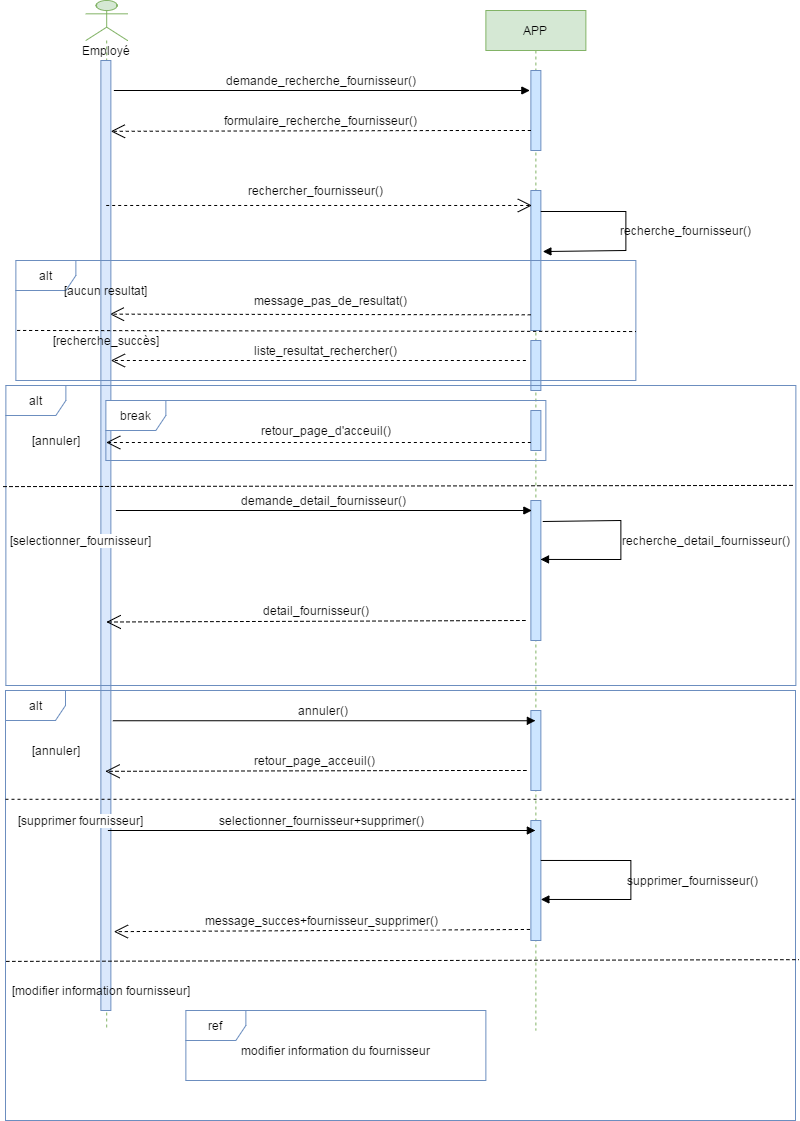


Figure :Diagramme de séquence pour le scénario d’authentification





Conclusion :

On a terminé le chapitre spécification et nous allons toucher la conception du l’application en utilisant le diagrammes de classe et tous .

Chapitre 4 : Conception

Introduction :

Le présent chapitre sera consacré à la présentation de l’étape de conception de notre système. La conception est un processus créatif qui permet de décrire la manière non ambigüe du fonctionnement désiré du système afin d’en faciliter la réalisation et la maintenance. A cet effet, nous présentons dans une première partie une conception générale de notre application ensuite dans la deuxième partie nous détaillons la conception de l’application et de la base de données utilisée.

1. Conception Générale :
   1. Conception architecturale :

L’architecture de la plateforme est « Microservices » qui utilise le model MVC pour chaque service

* + 1. L’architecture microservices :

1. Définition :

les microservices sont une technique de [développement logiciel](https://fr.wikipedia.org/wiki/D%C3%A9veloppement_logiciel) — une variante du [style architectural](https://fr.wikipedia.org/wiki/Patron_d%27architecture) de l'[architecture orientée services](https://fr.wikipedia.org/wiki/Architecture_orient%C3%A9e_services) (SOA) — qui structure une application comme un ensemble de services [faiblement couplés](https://fr.wikipedia.org/wiki/Couplage_(informatique)). Les microservices indépendants communiquent les uns avec les autres en utilisant des [API](https://fr.wikipedia.org/wiki/Interface_de_programmation) indépendantes du langage de programmation.

Des API [REST](https://fr.wikipedia.org/wiki/Representational_State_Transfer) sont souvent employées pour relier chaque microservice aux autres. Un avantage avancé est que lors d'un besoin critique de mise à jour d'une ressource, seul le microservice contenant cette ressource sera mis à jour, l'ensemble de l'application restant compatible avec la modification, contrairement à la totalité de l'application dans une architecture classique, par exemple une [architecture trois tiers](https://fr.wikipedia.org/wiki/Architecture_trois_tiers). Cependant, le coût de mise en place, en raison des compétences requises, est parfois plus élevé.

1. Les microservices :

Chaque micro-service peut être:

* Développé indépendamment
* Avoir une base de données indépendante
* testé indépendamment
* déployé indépendamment

(Technologie différente des autres)

Les applications de microservices permettent aux développeurs de décomposer plus facilement leur travail en petites équipes indépendantes et d’intégrer ce travail au fur et à mesure.

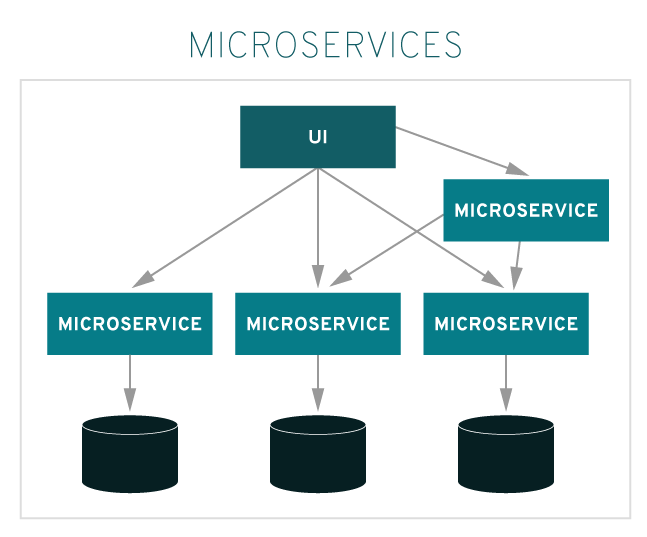


Figure : Architecture Microservices

2.le modèle MVC :

1. L’architecture MVC :
   1. Définition :

L'architecture *Modèle/Vue/Contrôleur* (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le *modèle*, la *vue* et le *contrôleur* ayant chacun un rôle précis dans l'interface.

modèle : données (accès et mise à jour)

vue : interface utilisateur (entrées et sorties)

contrôleur : gestion des événements et synchronisation

1. Model (modèle) :

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient.

Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur).

1. View (vue) :

La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d’une entrée, boutons, …). Ces différents évènements sont envoyés au contrôleur.

1. Contrôleur :

Le contrôleur est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Certains évènements de l'utilisateur ne concernent pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier.

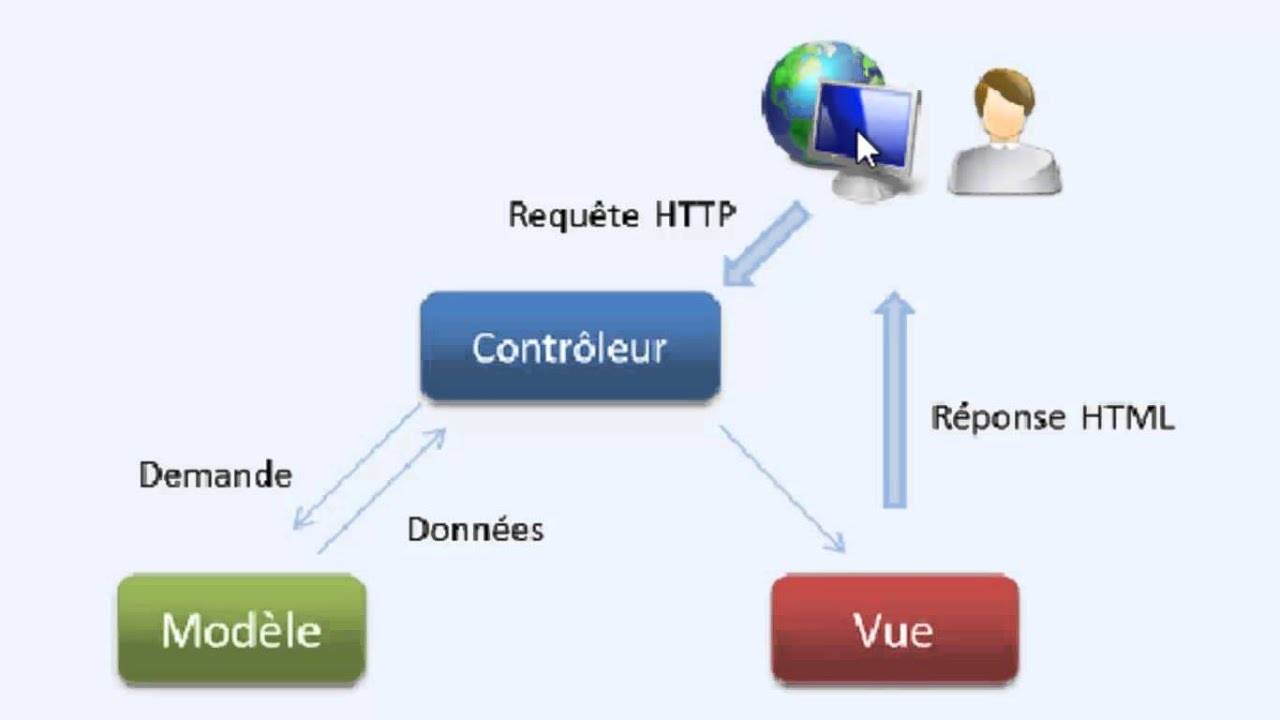


Figure : Architecture MVC

* + 1. Architecture retenue :

Contrairement à l'architecture monolithique traditionnelle, l'architecture de microservices est une approche dans laquelle l'application est structurée via plusieurs petits services autonomes qui fonctionnent ensemble.

Cette architecture a plusieurs avantages puisqu’elle :

* Rend possible un déploiement continu et automatisé.
* Permet aux développeurs de prendre des décisions appropriées et spécifiques au service.
* Chaque microservice possède sa propre base de données.
* Permet aux entreprises d'optimiser les ressources pour le développement et les applications.
* Optimise le dimensionnement et s'intègre facilement aux services tiers.

L’architecture microservices utilise le modèle MVC dans chaque microservice car le modèle MVC a une :

* Maintenance facile et ajout de fonctionnalités.
* Réduit le temps de développement du projet.
* Aide à créer des logiciels fiables avec des architectures testées.
* Facilite la réutilisation du code.
* Meilleure intégration de l'équipe .
* Réduction de la complexité du code.
  1. Conception détaillée :

Cette section sera consacrée à la conception détaillée de notre application, nous débutons avec les diagrammes de classes

1.2.1 Diagramme de classes :

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction.

1.2.1.1 Diagramme de classe générale :

Une image contenant texte, signe, vente, capture d’écran

Description générée automatiquement

1.2.2 : conception des bases de données :

Cette partie est consacrée à la conception des bases de données utilisées…

-SQL :

Pour La gestion du premier microservice (user) j’ai utilisé la base de données SQL avec le logiciel MySQL

SQL :

Le langage **SQL** (Structured Query Language) est un langage informatique utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données.

                Les instructions SQL couvrent 4 domaines :

* [Langage de définition de données](https://fr.wikipedia.org/wiki/Langage_de_d%C3%A9finition_de_donn%C3%A9es),
* [Langage de manipulation de données](https://fr.wikipedia.org/wiki/Langage_de_manipulation_de_donn%C3%A9es),
* [Langage de contrôle de données](https://fr.wikipedia.org/wiki/Langage_de_contr%C3%B4le_de_donn%C3%A9es),
* [Langage de contrôle des transactions](https://fr.wikipedia.org/wiki/Langage_de_contr%C3%B4le_des_transactions).

My SQL :



Figure : Logiciel de base de donnée MySQL

MySQL est un serveur de [bases de données relationnelles](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle) [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language) développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est [multi-thread](https://fr.wikipedia.org/wiki/Processus_l%C3%A9ger" \o "Processus léger) et multi-utilisateur.

-NoSQL :

NoSQL correspond à « **not only SQL** » et c’est en effet ce que ce modèle de base de données veut être : non pas une contrepartie, mais bien un enrichissement et **complément utile des bases de données SQL relationnelles traditionnelles**. Ce faisant, les bases de données NoSQL dépassent les limites des systèmes relationnels et exploitent un modèle de base de données alternatif. Cela ne veut toutefois pas dire qu’aucun système SQL n’est utilisé. Il existe de nombreuses variantes combinées au sein desquelles les deux solutions peuvent être utilisées et qui restent toutefois englobées sous l’étiquette NoSQL.

* Organisent les gros volumes de données au moyen de techniques flexibles,.

* Dans la mesure où les systèmes NoSQL exploitent un cluster matériel et un serveur Cloud, les **capacités**sont réparties de manière **uniforme** et la base de données fonctionne avec fluidité.
* Contrairement aux bases de données relationnelles qui perdent rapidement de la puissance à mesure que le volume de données augmente, les bases de données NoSQL s’imposent comme une solution performante, flexible et évolutive pour le stockage de grosses quantités de données.
* L’une des particularités des systèmes NoSQL est, en outre, leur capacité de **dimensionnement horizontal.** Les bases de données SQL relationnelles sont dimensionnées verticalement.
* Leur entière performance repose sur un seul et même serveur.

 Les solutions NoSQL répartissent généralement leurs données sur plusieurs serveurs. En cas d’augmentation du volume de données, de **nouveaux serveurs** peuvent alors facilement être **ajoutés**.

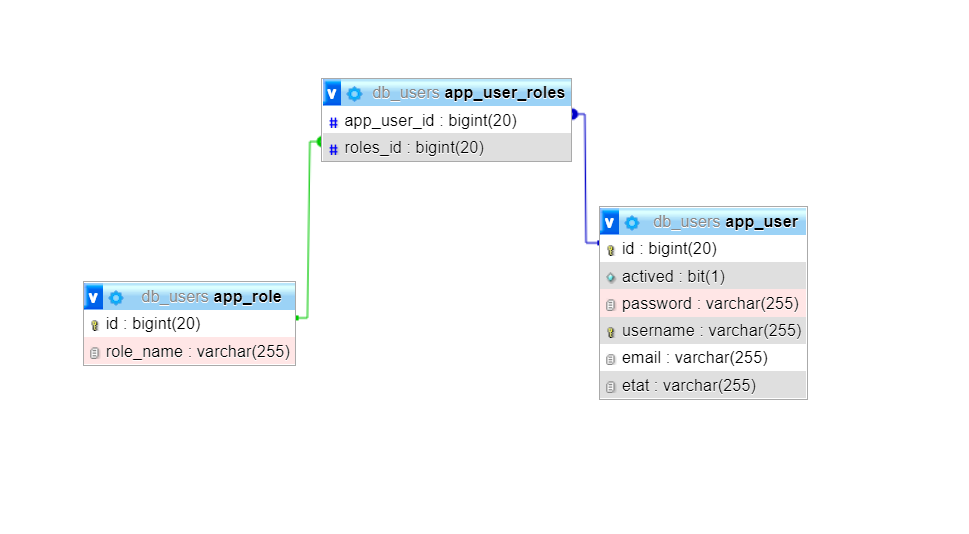
-MongoDB :



MongoDB est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) [orienté documents](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_orient%C3%A9e_documents), [répartissable sur un nombre quelconque d'ordinateurs](https://fr.wikipedia.org/wiki/Scalability) et ne nécessitant pas de schéma prédéfini de données. Il est écrit en [C++](https://fr.wikipedia.org/wiki/C%2B%2B). Le serveur et les outils sont distribués sous [licence SSPL](https://fr.wikipedia.org/wiki/Server_Side_Public_License), les pilotes sous [licence Apache](https://fr.wikipedia.org/wiki/Licence_Apache) et la documentation sous [licence Creative Commons](https://fr.wikipedia.org/wiki/Licence_Creative_Commons)[4](https://fr.wikipedia.org/wiki/MongoDB#cite_note-licensing-4). Il fait partie de la mouvance [NoSQL](https://fr.wikipedia.org/wiki/NoSQL).

1.2.2.1 : Base de donnée User :

Concepteur de la base de données SQL :



Vue globale des tables SQL :



1.2.2.2 : Base de données DATACENTER :

Pour La gestion du deuxième microservice (DATACENTER) j’ai utilisé la base de données NoSQL avec le logiciel MongoDB .

Concepteur base de données DATACENTER NoSQL :

Une image contenant table

Description générée automatiquement

## Conclusion

Tout au long de ce chapitre, nous avons présenté une conception détaillée de notre application. J’ai commencé par la description de l’architecture globale de l’application, puis j’ai présenté une vue statique de l’application moyennant les diagrammes de classes. Ensuite, j’ai cité les différentes méthodes de bases de données que j’ai choisies. Enfin nous avons pu construire les bases de données de l’application avec ses différentes tables et collections.

A présent, nous sommes capables d’entamer la partie réalisation.

Chapitre 4 : Réalisation

Introduction :

Dans ce paragraphe nous allons décrit sommairement et conformément à la méthodologie SCRUM, les étapes de réalisation de notre projet, ensuite nous allons présenter la réalisation par rapport au produit final en décrivant l’environnement de travail et les technologies utilisées pour la mise en place de notre plateforme. Enfin, nous présentons quelques captures d’écran de l’application réalisée.

1.Réalisation par rapport au backlog

- Le déroulement des sprints :

3.Environnement de travail :

Cette partie présente l’environnement matériel mis à la disposition du présent projet ainsi

que l’environnement logiciel que nous avons exploité et nous présentons les choix

technologiques en les justifiant.

4.Environnement matériel :

Pc Asus TUF i7-10ème génération 16Mo RAM Disque dur SSD 512 Go

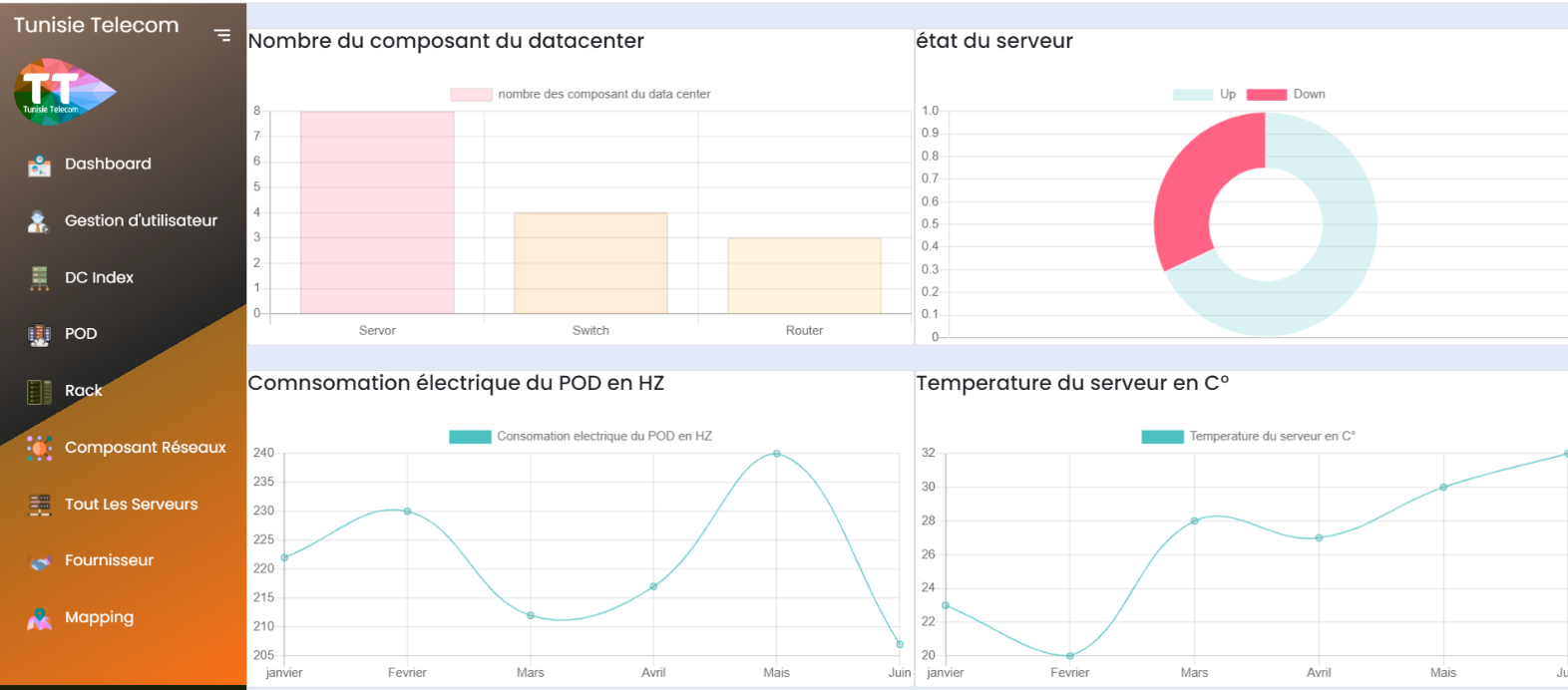
5.Environnement logiciel :

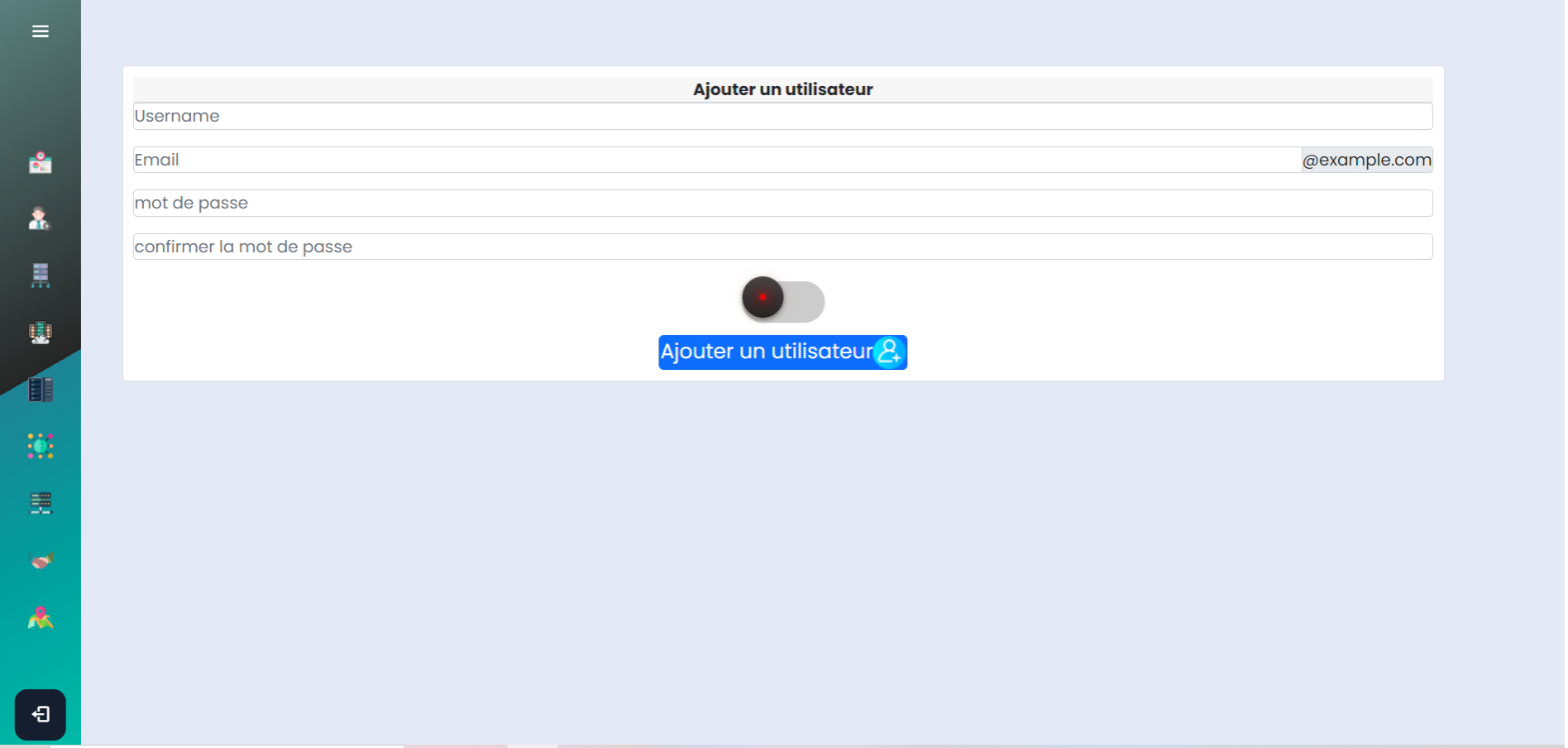
J’ai utilisé Intelliji IDE et Vs code

Vu du site  :

Une image contenant texte, carte de visite

Description générée automatiquement

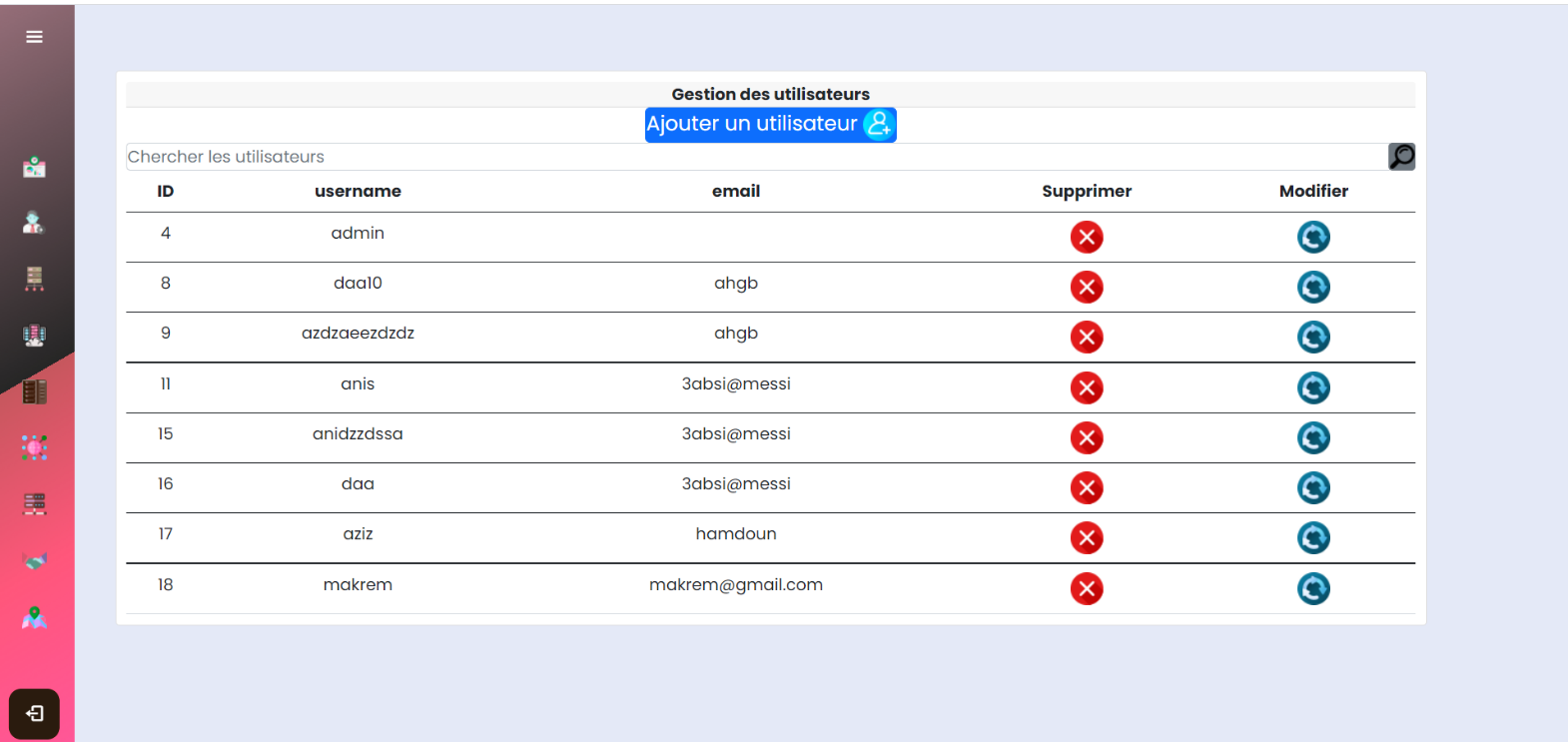
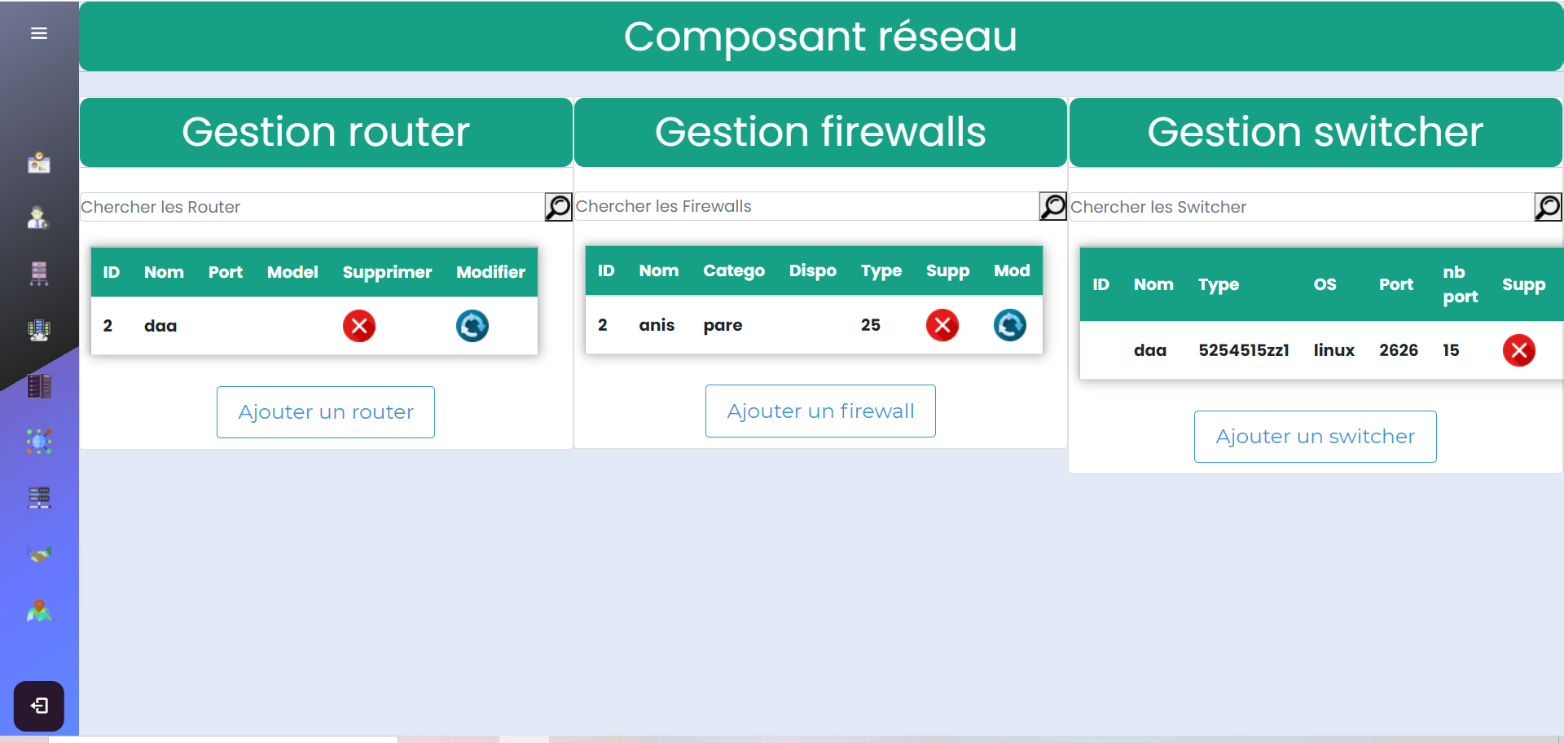












Conclusion : Pour mieux faire une gestion des actifs de DATACNTER on a réussit de faire une application tous les besoins demander et qui manipule tous les composants du DATACENTER pour un accès et une gestion facile et efficaces et sécurisé