**Plateforme de développement collaboratif des logiciels à base des méthodes**

**Agile/Scrum**

## **Introduction générale**:

Le logiciel est devenu un facteur de succès pour toutes les entreprises, ces derniers passent à une fabrication ou le composant logiciel est primordial dans leurs environnements, ils imposent certain critères pour le développement de leur logiciel qui est devenue leur projet initial, ou il faut bien gérer leur budget pour un bon rendement. Les entreprises de développements cherchent toujours la meilleure façon de gérer leur projet, en intégrant un environnement sain est idéal pour ses développeurs et leur limiter tout interruption inutile pendant leurs travail pour gagner en temps et en performance.

L’entreprise présente la collaboration entre ses employées, se qui pousse leurs chefs à penser sérieusement à limiter les conflits entre les employées se qui infecte fortement le rendement de ces entreprises, alors on essaye toujours de trouver un système qui assure une bonne entente, pour assurer le rendement exiger, se qui assure la survie de l’entreprise.

### Contexte :

Le développement collaboratif de logiciel rentre dans les concepts liés à l’industrialisation de la production et la fabrication des applicatives informatiques. La Gestion du cycle de vie des Application - ou ALM pour Application Lifecycle Management en anglais – peut être définie de plusieurs façons :

* La gestion coordonnée du cycle de développement d’une application logicielle…
* …. De son idée initiale à sa livraison et sa maintenance…
* ..impliquant tous les acteurs participant de prés ou de loin au projet...
* … ou qu’ils soient…
* …tout en appliquant les processus propres à l’entreprise et politiques de sécurité

### Problématique :

dans son domaine d’expertise sur l’intégration et le développement spécifique des applications métiers d’entreprise confronte à des problèmes de collaboration et de coordination entre les acteurs du projet (maitre d’ouvrage, maitre d’œuvre) dans son cycle de vie (de définition des besoins jusqu'au déploiement et la maintenance).

Les principales difficultés auxquelles elle fait face sont :

* Collaboration et coordination entre acteurs qui participent sur le cycle de vie de réalisation logiciel (analyse, concepteur, développeur, testeur, intégrateur, validateur, police de qualité…),
* Enrichissement des modèles (version) et le retour en arrière (itération),
* Validation et approbation des itérations,
* Planification et suivi des taches,
* Maitrise des méthodes agile de développement collaboratif des logiciels,

De plus partout dans le monde Les projets informatique avait connue des échecs récurrents, ces échecs son du au mauvais pilotage de ces projets, si un projet est livrés en retard ou que ce dernier à dépasser son budget et considérer comme un projet échouer.

On introduisant des nouvelles méthodes on limite ces échec récemment le taux de la réussite a un peu augmenter même si les cause de ces échec son bien connues comme : le besoin mal exprimer, le manque de communication ou encore le manque de réactivité.

### Objectifs :

**Icon Software ®** souhaite de mettre en place à ces développeurs une plateforme complète de développement collaboratif pour la conception et la réalisation des projets logiciel selon la méthode agile Scrum.

L’objectif de ce projet est de :

* Etude sur les méthodes Agile et Scrum,
* La spécification des métadonnées de chaque étape dans le cycle de vie,
* Conception et modélisation de la plateforme,

### Organisation de travail :

Notre mémoire est organisé en quatre chapitres : Chapitre 1 : Analyse

Chapitre 2 : Conception

conclusion

**Sommaire**

Titre

Introduction générale

### Chapitre 1 : Analyse

* 1. Présentation d’ALM (Application Lifecycle Management)
  2. Méthodologie
  3. Introduction aux méthodes Agiles
  4. La méthode Scrum
     1. Les rôles dans Scrum
     2. Déroulement de Scrum

### Chapitre 2 : Conception

* 1. UML
  2. Diagramme de classe
  3. Cas d’utilisation
  4. Diagramme de séquence

**Conclusion** Conclusion générale Perspectif

**Chapitre 1 : Analyse**

### Présentation d’ALM (Application Lifecycle Management) :

ALM consiste la gestion du Cycle de vie des applications, et leur pilotage depuis son idée initial à sa livraison et sa maintenance pendants tout la vie de l’application, il rapproche les acteurs participant de près ou de loin donc il fourni des outils adaptés pour tous les personnes impliquées dans le projet.

ALM recouvre plusieurs grandes étapes :

* + - Gestion des exigences
    - Gestion des changements
    - Gestion de version
    - Gestion de projet
    - Gestion de la qualité
    - Gestion des livrables
    - Gestion Documentaire
    - Collaboration sociale

Pour éviter certain problèmes comme la multiplication des outils qui provoque le non intégration ou non compatibilités des logiciels qui vont donner un manque des liens,

ALM à regrouper tous ces outils dans une seule application pour que tous ces outils arrivent à travailler ensemble d’une façon directe, avec une communication souple entre tous les services, avec un accès au données qui sont centralisées et partager pour éviter les échange de fichier manuel qui peuvent provoquer des pertes de données ou qui peuvent prendre beaucoup de temps pour arriver à destination, ainsi on évite toutes les installations et configuration des outils dans les différents postes par une seule installation et ça va faciliter l’administration de l’application, en cas d’un nouveau projet il suffit juste de créer ce dernier

et ajouter des membres depuis l’annuaire de l’entreprise, maintenant on peut travailler dans un environnement ou la distance na pas d’importance tous les membres son connecter a la plate forme ou qu’ils sont, et peuvent être ajoure par l’outil de suivi d’avancement des projets.

Chaque entreprise a ça façon de travailler mais elle doit suivre une méthode pour être en conformité avec les normes, il existe plusieurs méthodes de gestion de projet qui peuvent être mises en place avec ALM comme : Cycle en V, CMMI, Cobit, ITIL, et les Méthodes agiles.

Dans notre projet on va parler des méthodes agiles/Scrum

### Méthodologies :

Pour cette partie nous nous intéresserons à donner un aperçu sur les méthodologies agiles en général. Puis, on va entamer une étude détaillée sur la méthodologie Scrum, celle utilisée dans le présent projet.

### Introduction aux méthodes Agile:

C’est des méthodes de développement, qui arrive a réduit le cycle de vie de logiciel pour développer une version minimal, puis en intégrant les fonctionnalités par un processus

itératif basé sur une écoute du client qui va assurer une grande réactivité à ses demandes et des tests tout au long du cycle de développement pour assuré que le logiciel répond au besoin réel du client.

Principe des méthodes Agile :

* + - La communication de l’équipe avant les processus et les outils.
    - Le fonctionnement du projet est plus prioritaire que sa documentation technique.
    - L’implication du client plutôt qu’un contrat.
    - La planification et flexible et soumise a des changements.

### La méthode Agile Scrum :

Scrum est une méthode agile de gestion de projet, utilisé pour organiser les équipes pour produire une grande valeur métier dans un temps court avec une meilleure qualité, elle découpe le projet en itérations nommées Sprints, un sprint peut avoir une durée de 2 a 4 semaines.

On donne l’estimation des taches avant chaque sprint, pour permettre une sélection des taches qu’on va inclure dans un sprint, cette liste des taches choisi pour un sprint s’appel le backlog, on peut éventuellement ajouter de nouvelle taches en cours de projet en dehors des sprints, un sprint se compose par des mêlée quotidienne, dans laquelle les collaborateurs indiquent les taches effectuées la veille, les difficultés rencontré, et les taches a entamer prochainement, cela permet d’évaluer l’avancement, et de partager les ressources selon les besoins.

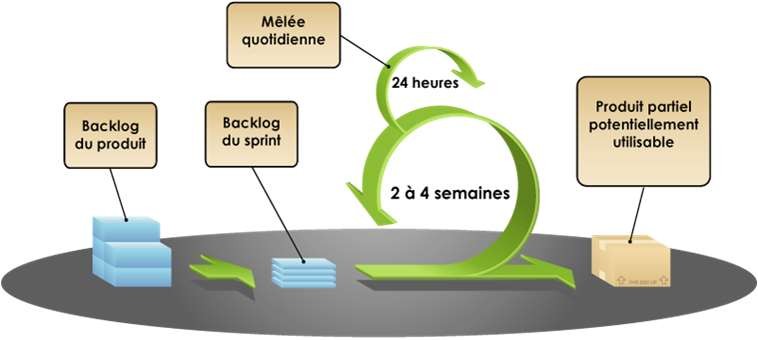


FIGURE-1 : les étapes de la méthode Scrum

### 1. Les rôles dans Scrum :

Chef de projet (Product Owner) :

C’est le directeur de produit, il représente les clients et les utilisateurs.

* + - Définit les fonctionnalités du produit
    - Responsable du retour sur investissement
    - Définit les priorités dans le backlog en fonction de la valeur « métier »
    - Ajuste les fonctionnalités et les priorités à chaque sprint si nécessaire
    - Accepte ou rejette les résultats Chef d’équipe (Scrum master):

C’est l’intermédiaire entre le chef de projet et l’équipe et joue le rôle de coache de l’équipe,

il est chargée de veiller à la mise en application de la méthode et au respect de ses objectifs.

* + - Représente le management du projet
    - Responsable de faire appliquer par l’équipe les valeurs et les pratiques de Scrum
    - Surveille l’application des et des valeurs Scrum par l’équipe
    - Élimine les obstacles
    - S'assure que l'équipe est complètement fonctionnelle et productive
    - Facilite une coopération poussée entre tous les rôles et fonctions
    - Protège l'équipe des interférences extérieures Equipe Scrum (Development Team) :

Composer de 5 a 10 personnes (Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.)

* + - A pleine temps sur le projet
    - L’équipe s’organise par elle-même
    - La composition de l’équipe ne doit pas changer pendant un sprint

### 2. Déroulement de Scrum :

Planification du sprint :

* + - L’équipe choisit, à partir de backlog de produit les éléments qu’elle s’engage à finir.
    - La liste des taches est créée :
      * Les taches son identifier et estimées (en jours-homme)
      * Collectivement, pas seulement le chef d’équipe

### La mêlée (Scrum quotidien) :

* + - Paramètres
      * Tous les jours
      * 15 minutes
      * Debout
    - Il n’est pas fait pour résoudre les problèmes
      * Tous le monde est invité
      * Seuls les membres d’équipe peuvent parler
    - Il permet d’éviter d’organiser d’autres réunions

Chacun répond à 3 questions:

1. Qu’est ce que t’as fait hier ?
2. Que vas-tu faire aujourd’hui ?
3. Y’a-t’il des obstacles qui te freine ?

### Revue de sprint (La démo) :

A la fin de chaque sprint on fait une démo, elle pousse l’équipe à terminer leur travail et le livrer

* + - L’équipe présente se qu’elle a fait pendant le sprint
    - Toute l’équipe participe
    - On invite les autre à découvrir se que l’équipe fait

### Rétrospective du sprint:

C’est une réunion qui vise a amélioré les sprints et ne plu faire les mêmes erreurs

* + - Se fait a la fin de chaque sprint
    - Tout l’équipe participe même le chef de projet éventuellement client et autre intervenant
    - Chacun donne son avis
      * ce qu’il pense être bien
      * ce qu’il pense qui peut être mieux
      * ou ce qu’il devrait être différent pour le prochain sprint

À la fin on garde les suggestions qui peuvent améliorer le prochain sprint.

# Chapitre 2: Conception

La phase de conception consiste à donner une idée plus claire de l’analyse de notre solution envisagée a travers des diagrammes UML.

* 1. **UML (*Unified Modeling Language*) :**

UML est un langage de modélisation graphique, il est apparu dans le cadre de la conception orientée objet, couramment utilisé dans les projets logiciels, UML est utilisé pour spécifier, visualiser et construire les documents nécessaires au bon développement d'un logiciel orienté objet. Il offre un standard de modélisation, pour représenter l'architecture logicielle.

### Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations qui lie c’est dernier, on trouve comme lien l’association, généralisation, dépendance…etc.

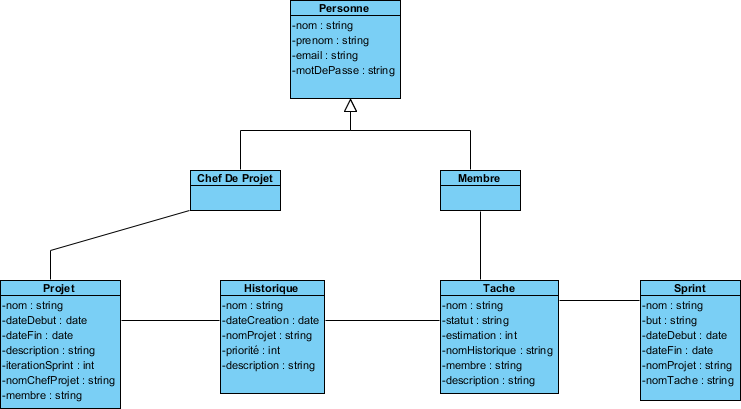


FIGURE-2 : Diagramme de classe de l’application aykonScrum

### Cas d’utilisation (Use case) :

Un cas d’utilisation représente les relations fonctionnelles entre les acteurs et le système, il permet de représenter visuellement une séquence d’actions réalisées par un système,

représenté par une boite rectangulaire, produisant un résultat sur un acteur, l’acteur

représente le rôle d’une entité externe (utilisateur humain dans notre cas) qui interagit avec le système, il est noter qu’un utilisateur peut jouer plusieurs rôles vis-à-vis du système ce qui amène a le modélisé par plusieurs acteurs.

UML propose des conceptions issus de bonne pratique qui permet de décrire, dans des documents lisibles par tous, la finalité des interactions système et de ses utilisateurs.

Pour avoir une idées détailler et claire de notre système on a met un cas d’utilisation globale, avec tous les acteurs du système ainsi que des cas d’utilisation de chaque acteur.

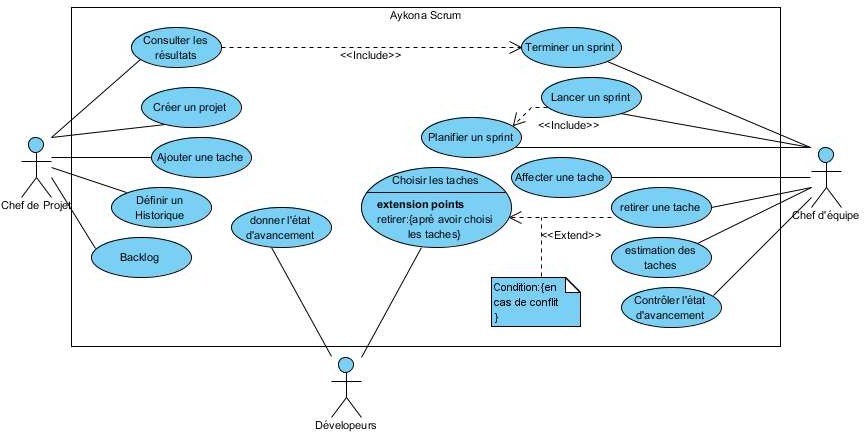


FIGURE-3 : cas d'utilisation globale d’Aykona Scrum

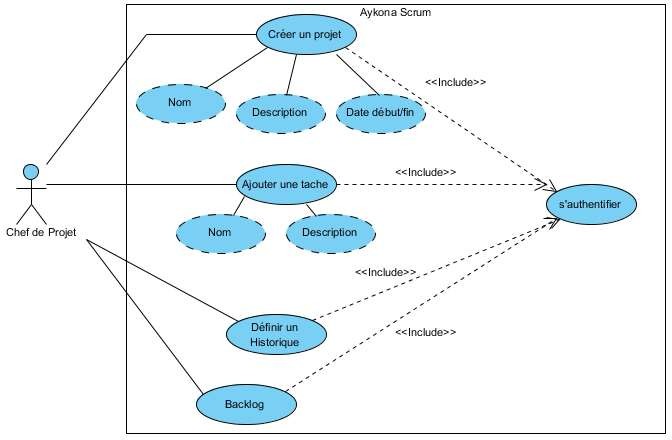
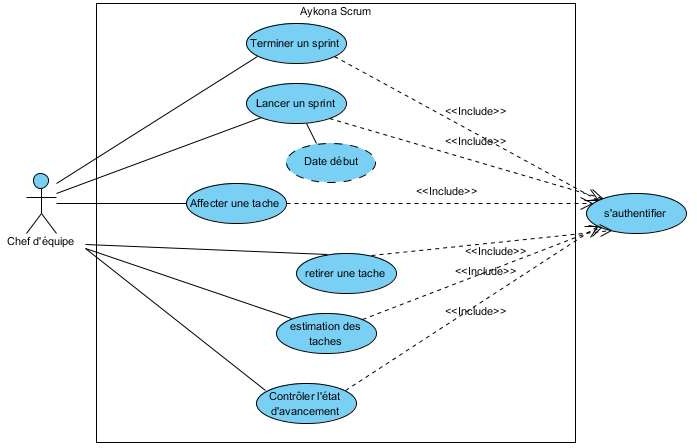


FIGURE- 4: cas d'utilisation chef projet



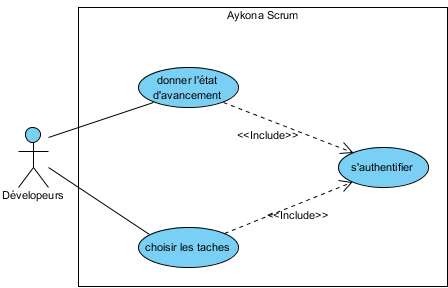
FIGURE- 5: cas d'utilisation chef d'équipe

FIGURE- 6: cas d'utilisation développeu

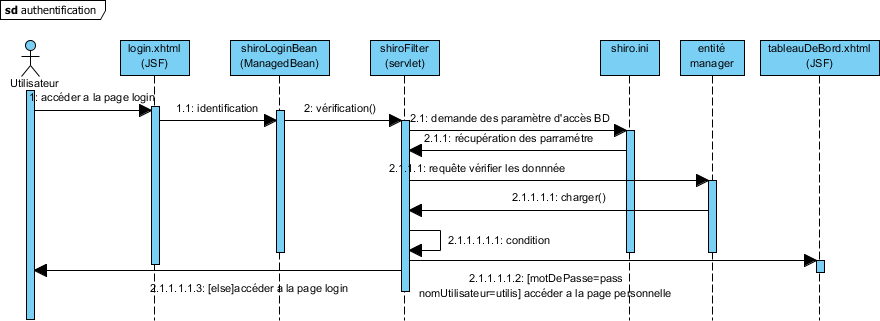
### Diagramme de séquence:

Il représente graphiquement la chronologie des échanges de message avec le système ou au sein du système, ces éléments sont :

* + - Acteurs
    - Objets (instances)
    - Message (cas d’utilisation, appels d’opération)

Il permet de décrire la réalisation d’un cas d’utilisation sur le système décrit par le diagramme de classes

Les diagrammes de séquence de notre application sont les suivant :



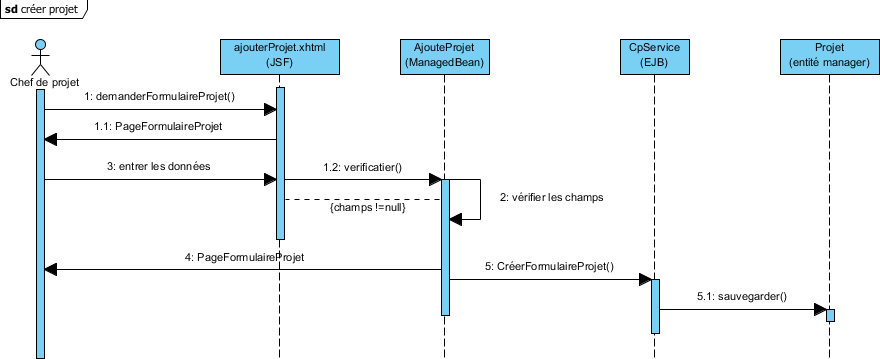
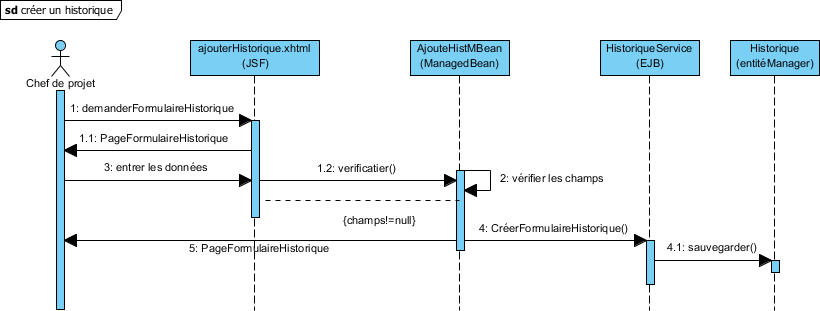
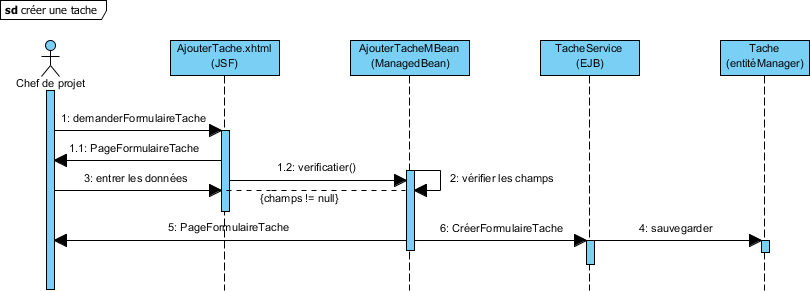
 FIGURE- 7: diagramme de séquence d’authentification

FIGURE- 8: diagramme de séquence créer projet



 FIGURE- 9: diagramme de séquence créer un historique

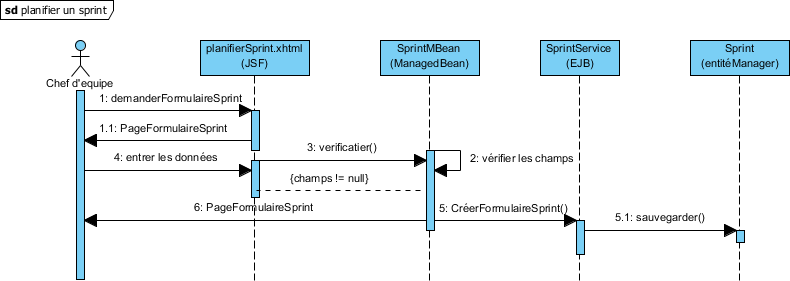
 FIGURE- 10: diagramme de séquence créer tache

FIGURE- 11: diagramme de séquence planifier un sprint

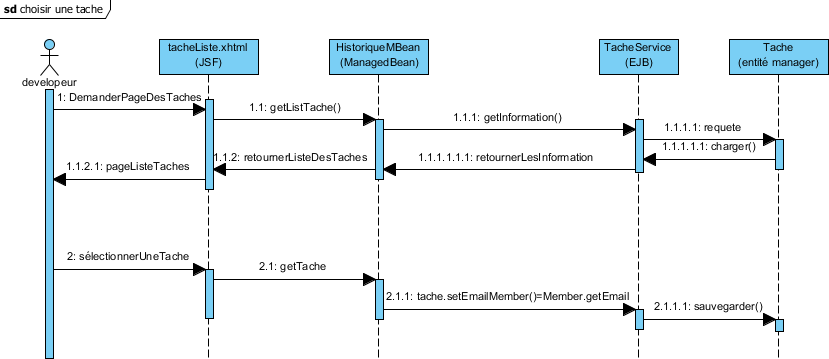


FIGURE- 12: diagramme de séquence choisir une tache

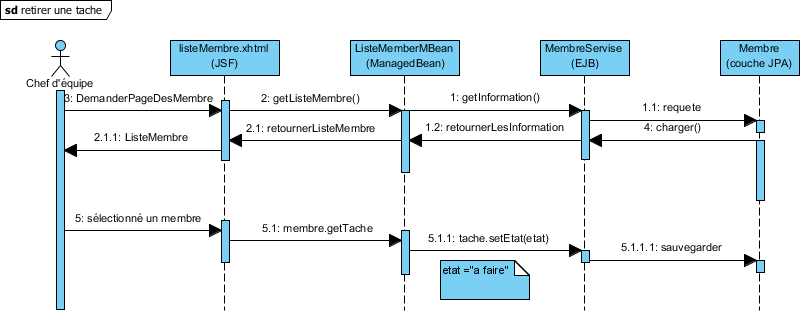


FIGURE- 13: diagramme de séquence retirer une tache

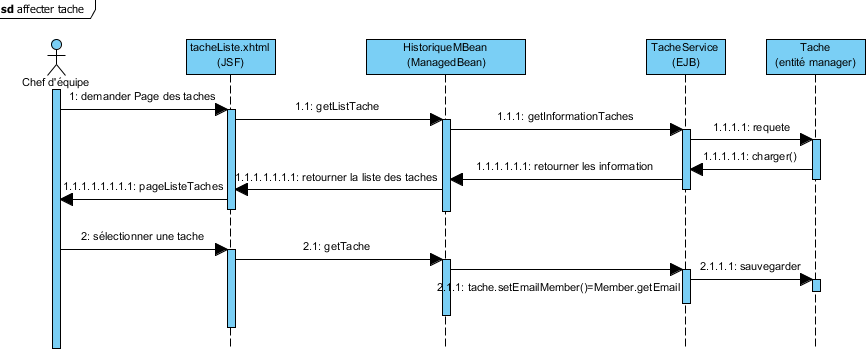


FIGURE- 14: diagramme de séquence affecter tache

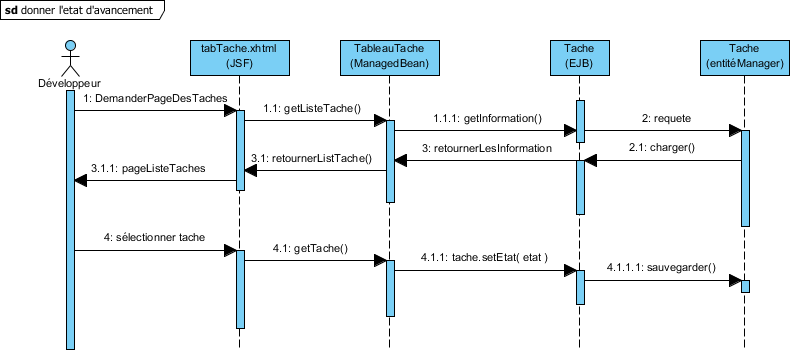


FIGURE- 15: diagramme de séquence donner l'état d'avancement

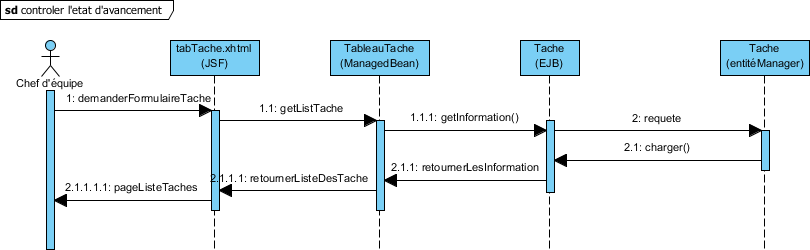


FIGURE- 16: diagramme de séquence contrôler l'état d'avancement

**Conclusion**

**Conclusion générale :**

Notre projet nous a permet d’approcher l’environnement de l’entreprise, et de comprendre les rôles de chacun dans une entreprise, plus précisément dans le développement informatique , ça nous a permet d’avoir une idée sur les problèmes, qui peuvent bien

infecter l’environnement de travail, les conflit entre les membre, ou les problèmes qui peuvent bien retarder ou échoué des projets informatique, tout ces informations nous ont aider a comprendre des techniques employer pour minimiser les pertes , et de bien gériez les projet, dans une équipe qui travail en collaboration pour produire un résultat satisfaisant, le travail collaboratif est une source de créativité il permet d’améliorer et de rendre le travail flexible pour un meilleur résultat, ces étude nous ont permet d’apprendre des nouveaux concept pour planifier des projets en minimisant leur cout.

### Perspectif :

Les point qu’on peut ajouter dans notre application pour l’améliorer serra de pouvoir générer des rapports de chaque projet, et dans le future on souhaite qu’elle devient une application complète pour la gestion de cycle de vie des logiciel en lui implémentant tout ses fonctionnalité, par exemple la gestion des changements qui consiste a stoker et tracer tous les changement effectués par les membres dans l’évolution de leur travail, et la gestion de version pour permettre aux ingénieurs de développer ensemble et fusionner leur travail en tout sécurité, gestion documentaire et d’apporter de nouveaux approche qui améliore le travail collaboratif.