

RAPPORT DE PROJET DE FIN D'ETUDE

Présenté en vue de l'obtention de la

LICENCE EN SCIENCES DE L'INFORMATIQUE

Parcours : Génie Logiciel et Système d'Information

Développement d'une plateforme web de gestion des projets

Par

MOHAMED KHALIL SAKIS

Réalisé au sein du "Centre National de l'Informatique"



Président :

Rapporteur :

Année Universitaire : 2024 - 2025

Table des matières

Remerciements	4
Introduction générale	5
1 Cadre du projet	7
1.1 Introduction	7
1.2 Présentation de l'organisme d'accueil	7
1.3 Etude de l'existant	8
1.4 Critique de l'existant	10
1.5 Solution proposée	11
1.6 Conclusion	12
2 Spécification des besoins	13
2.1 Introduction	13
2.2 Identification des besoins fonctionnels	13
2.3 Identification des besoins non fonctionnels	14
2.4 Identification des acteurs	14
2.5 Diagramme de cas d'utilisation globale	15
2.6 Backlog de produit	17
2.7 Identification des besoins techniques	17
2.7.1 Environnement de travail	18
2.7.2 Environnement logiciel	18
2.7.3 Technologies utilisées	19
2.8 Méthodologie de conception	21
2.8.1 Méthodologie Agile	21
2.8.2 Scrum	21
2.9 Conclusion	22

Table des figures

1.1	Logo de la société	7
1.2	www.atlassian.com/fr/software/jira	8
1.3	azure.microsoft.com/en-us/products/devops	9
1.4	tuleap.org	9
1.5	redmine.org	10
2.6	Diagramme de cas d'utilisation	16
2.7	Processus Scrum [19]	21

Liste des tableaux

2.1	Description détaillée des acteurs	15
2.2	Backlog de produit	17
2.3	Caractéristiques du PC	18
2.4	Environnement Logiciel	19
2.5	Technologies utilisées	20

Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui nous ont accompagné et soutenu tout au long de ce stage.

En premier lieu, nous remercions chaleureusement nos enseignants de l'Institut Supérieur des Technologies de l'Information et des Communications de Borj Cedria pour leur engagement, leur disponibilité et les connaissances qui nous ont été transmises. Leur expertise et leur dévouement ont été essentiels à notre apprentissage et à l'accomplissement de notre travail.

Nous souhaitons également exprimer notre reconnaissance envers notre encadrant professionnel au sein de YYY, Mr. XXX. Ses précieux conseils, son suivi attentif et son expertise ont permis la bonne réalisation de ce projet dans les meilleures conditions.

À tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce projet et à notre développement au cours de ce stage, nous adressons nos sincères remerciements.

Finalement, nous remercions les membres du jury qui ont accepté d'évaluer notre travail.

Introduction générale

Dans un monde où l'innovation technologique se succède à un rythme effréné, les organisations font face à des défis de plus en plus complexes dans la gestion de leurs activités. La capacité à coordonner efficacement les ressources, à suivre les progrès et à garantir la qualité des résultats est devenue essentielle pour conserver un avantage concurrentiel. Les systèmes de gestion de projets jouent un rôle clé dans ce contexte en fournissant des outils permettant de structurer les processus, d'améliorer la collaboration entre les équipes et d'optimiser les délais d'exécution.

Cependant, la complexité croissante des projets informatiques, conjuguée à la diversité des acteurs impliqués et à la multiplicité des processus, rend indispensable l'adoption d'une solution centralisée et intuitive. Une telle plateforme doit non seulement faciliter la soumission des projets, mais également permettre leur évaluation par des équipes de test et leur validation par des superviseurs. Elle doit offrir une visibilité complète sur l'état d'avancement des projets et les retours des différents intervenants, garantissant ainsi une gestion rigoureuse et conforme aux standards de qualité.

Dans ce contexte, nous avons entrepris de développer une plateforme interne de gestion de projets pour notre organisme d'accueil. Cette solution vise à simplifier et structurer le processus de soumission, d'évaluation et d'approbation des projets élaborés par l'équipe de développement. Concrètement, les développeurs peuvent y soumettre leurs travaux, qui sont ensuite évalués par des testeurs et validés par des superviseurs, selon un flux de travail prédéfini. En centralisant ces étapes, notre plateforme ambitionne d'améliorer la transparence, la collaboration et l'efficacité au sein de l'organisation.

Pour répondre efficacement à ces besoins, nous adoptons une méthodologie de conception agile, en utilisant le framework **SCRUM**. Cette approche itérative et incrémentale nous permet de collaborer étroitement avec les parties prenantes et de livrer des fonctionnalités à valeur ajoutée de manière progressive.

La structure de notre rapport se divise en quatre chapitres, chacun détaillant un aspect fondamental de notre analyse :

- Chapitre 1 intitulé « **Cadre du projet** » dont nous allons présenter l'organisme d'accueil, définir les objectifs et les attentes du projet, tout en faisant une étude de l'existant afin de proposer une solution adaptée aux besoins de l'utilisateur.
- Chapitre 2 intitulé « **Spécification des besoins** » dans lequel nous allons identifier les besoins fonctionnels et non fonctionnels, ainsi que les acteurs et les cas

d'utilisation.

- Chapitre 3 intitulé « **Release 1 : Mise en Œuvre des fonctions de base d'utilisation** » qui décrit le premier release.

Nous présentons dans ce chapitre le raffinement, la conception et la réalisation du sprint 0 et 1.

- Chapitre 4 intitulé « **Release 2 : Fonctionnalités avancées de gestion et d'approbation** » qui présente le deuxième release.

Nous exposons dans ce chapitre le raffinement, la conception et la réalisation du sprint 2 et 3.

Enfin, nous synthétisons notre rapport par une conclusion générale et perspectives.

Chapitre 1

Cadre du projet

1.1 Introduction

Dans notre premier chapitre, nous établirons le cadre général de notre projet de fin d'étude. Nous débuterons par une présentation de l'organisme d'accueil dans lequel se déroule notre projet. Ensuite, nous réaliserons une étude de l'existant pour identifier les besoins non satisfaits et les insuffisances du système actuel. Enfin, nous proposerons une solution adaptée aux défis recensés.

1.2 Présentation de l'organisme d'accueil

Le Centre National de l'Informatique (CNI), créé le 30 décembre 1975, est un établissement public à caractère non administratif placé sous la tutelle du Ministère des Technologies de la Communication. Doté de la personnalité civile et de l'autonomie financière, le CNI opère dans les domaines de l'informatique et des technologies de la communication.

Certifié ISO 9001, le CNI développe et maintient des solutions adaptées aux besoins de l'administration, tout en offrant des services d'hébergement sécurisé et de continuité d'activité. Il assure également la formation des utilisateurs et un suivi technique pour garantir un déploiement optimal des applications.

Acteur clé de la transformation numérique en Tunisie, le CNI contribue à la modernisation de l'administration en proposant des outils technologiques fiables et performants.



FIGURE 1.1 – Logo de la société

1.3 Etude de l'existant

L'étude de l'existant constitue une étape primordiale dans notre démarche, nous permettant d'examiner en profondeur les plateformes de gestion de projets informatiques déjà présentes sur le marché. Cette analyse comparative a pour but d'identifier les points forts et les faiblesses des solutions actuelles, en mettant en lumière les fonctionnalités qui répondent aux attentes des utilisateurs et en évaluant les tendances ainsi que les pratiques couramment adoptées.

Les outils de gestion de projets actuels, bien que variés, peinent souvent à concilier simplicité, personnalisation et centralisation des processus. Si certaines plateformes proposent des fonctionnalités de soumission, de testing et d'approbation, elles restent généralement complexes à configurer, coûteuses pour les petites structures, ou trop génériques pour répondre aux besoins spécifiques des organisations informatiques. Notre solution se démarque en offrant un workflow unifié et intuitif, conçu pour fluidifier le cycle de validation des projets informatiques. En s'appuyant sur des technologies modernes et open-source, elle allie accessibilité, sécurité et adaptabilité, tout en garantissant une traçabilité complète des étapes d'approbation entre les différents acteurs.

Des exemples réels de plateformes en ligne dédiées à la gestion de projets informatiques

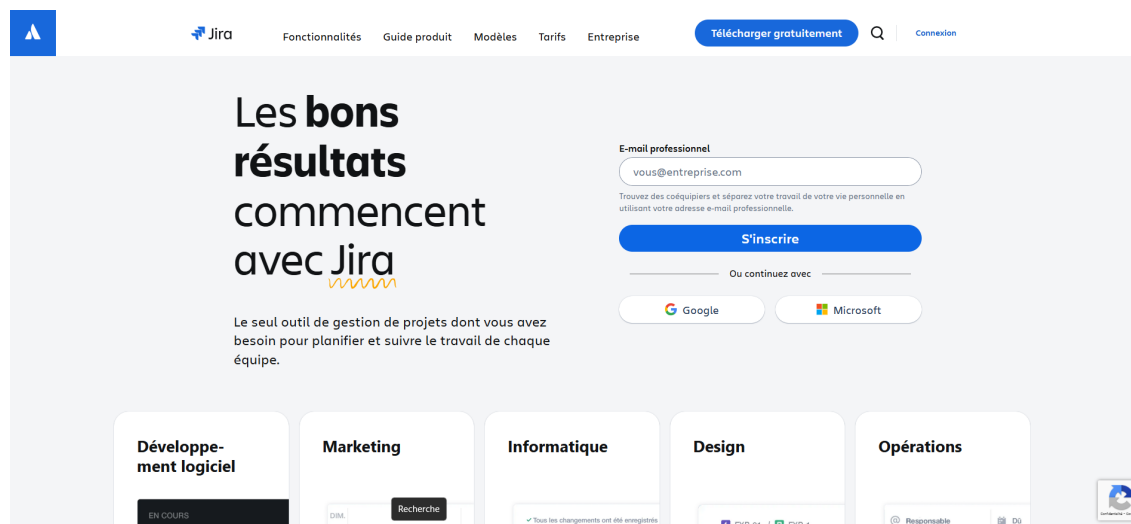
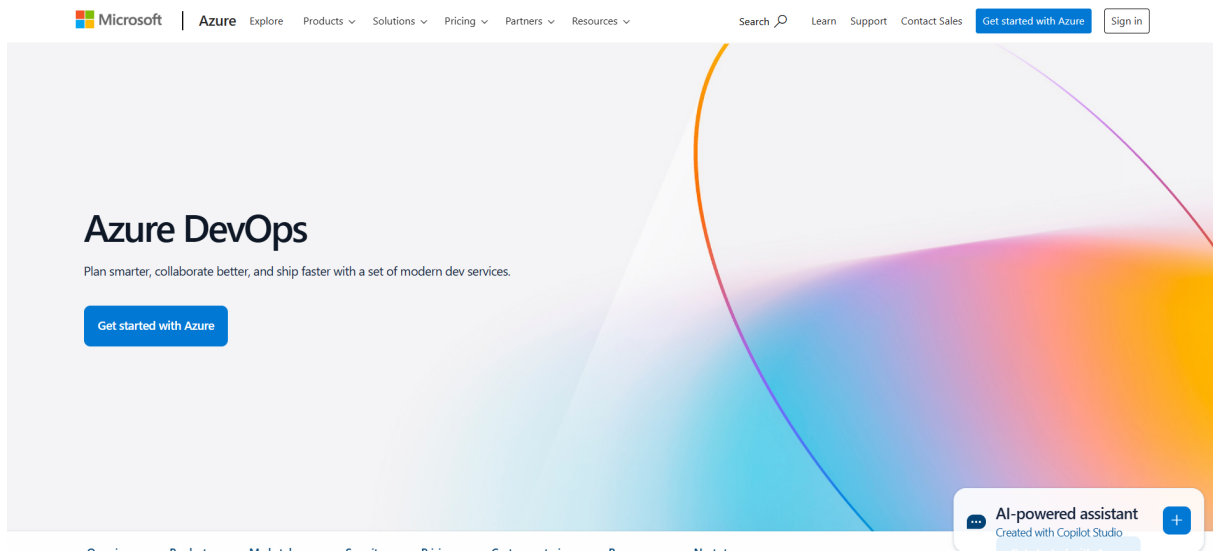
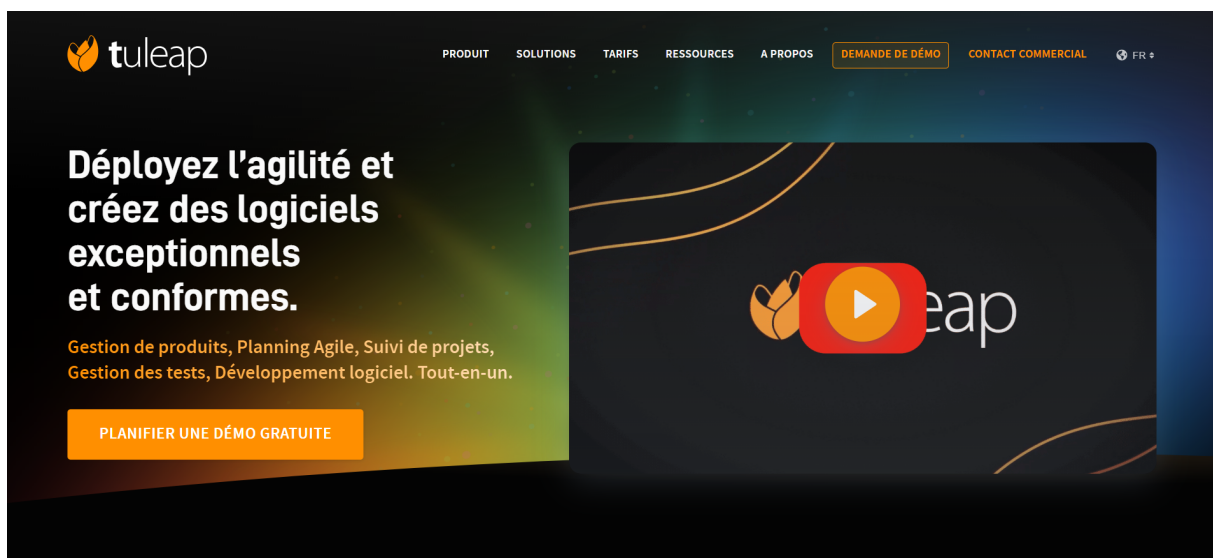


FIGURE 1.2 – www.atlassian.com/fr/software/jira

- **Jira** : Un outil de gestion de projet Agile populaire pour les équipes de développement, permettant de planifier, suivre et livrer des projets efficacement.

FIGURE 1.3 – azure.microsoft.com/en-us/products/devops

- **Microsoft Azure DevOps** : Une plateforme complète pour la gestion du cycle de vie des applications, incluant le développement, le déploiement et la gestion des projets.

FIGURE 1.4 – tuleap.org

- **Tuleap** : Une plateforme open source de gestion de projets qui intègre des outils de développement, de gestion de versions et de suivi de bugs.

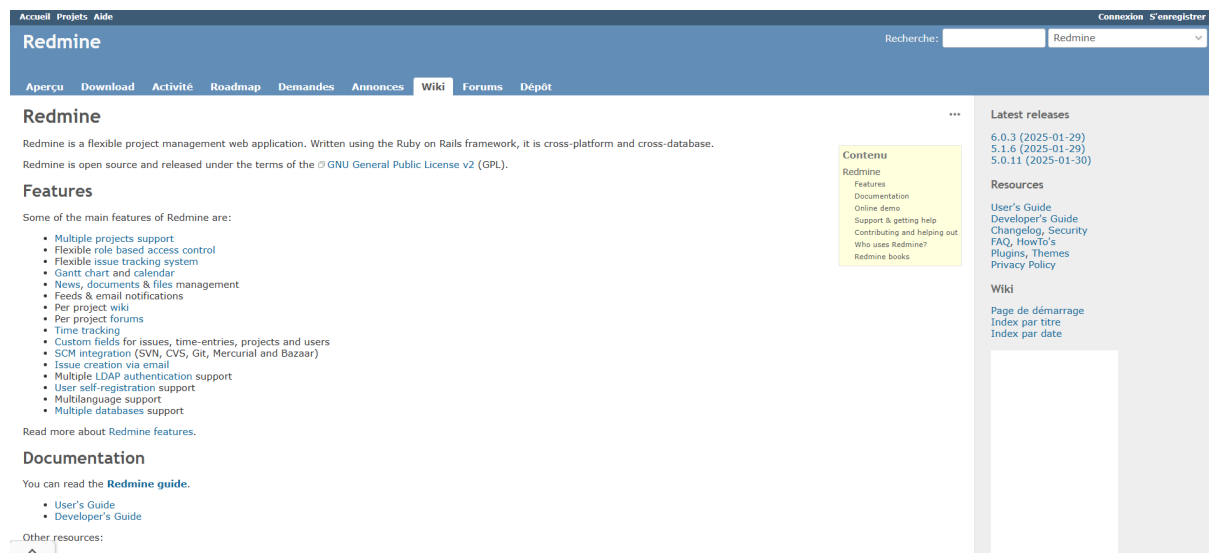


FIGURE 1.5 – redmine.org

- **Redmine** : Un gestionnaire de projets open source qui permet de suivre les tâches, les bugs et les versions, avec des fonctionnalités de gestion de projet flexibles.

1.4 Critique de l'existant

Les plateformes de gestion de projets informatiques actuelles présentent plusieurs limitations qui entravent l'expérience utilisateur et diminuent la satisfaction des parties prenantes impliquées dans le développement et la gestion des projets. Parmi les insuffisances les plus notables, nous pouvons citer :

- **Complexité et courbe d'apprentissage élevée** : Les plateformes peuvent être difficiles à maîtriser pour les utilisateurs débutants, nécessitant souvent une formation pour une utilisation optimale. De plus, leurs interfaces sont souvent surchargées de fonctionnalités, ce qui complique la prise en main et l'adoption par les nouveaux utilisateurs.
- **Coûts élevés** : Certaines plateformes imposent des licences ou des abonnements onéreux, ce qui peut être un frein pour les petites structures. Même les outils open-source génèrent des coûts indirects (hébergement, maintenance, personnalisation).
- **Intégration limitées** : Bien que ces plateformes offrent des intégrations avec divers outils tiers, certaines intégrations sont limitées, ce qui peut entraîner des silos d'information et des inefficacités dans la gestion des projets.
- **Scalabilité et Adaptabilité** : La scalabilité et la rapidité d'adaptation aux changements dans les processus métier sont souvent des défis pour ces plateformes. Les équipes ont besoin de solutions qui peuvent évoluer rapidement avec leurs besoins, ce qui n'est pas toujours possible avec les outils existants.

- **Manque de flexibilité dans les workflows** : Les workflows prédéfinis ne correspondent pas toujours aux processus métiers spécifiques des organisations, notamment pour les approbations multi-niveaux ou les validations externes.
- **Sécurité et conformité** : Les solutions cloud, comme Azure DevOps, posent des défis en matière de gouvernance des données, notamment pour les organisations soumises à des réglementations strictes, tandis que les versions open-source, telles que Tuleap, nécessitent une configuration manuelle pour garantir un niveau de sécurité adapté.

Ces insuffisances soulignent l'importance de proposer une solution mieux adaptée, offrant une expérience utilisateur plus fluide, intuitive et personnalisée, afin de répondre pleinement à leurs attentes.

1.5 Solution proposée

Pour répondre aux besoins non satisfaits des solutions existantes et pallier leurs limites, nous proposons de développer une plateforme web innovante de gestion de projets. Cette solution se positionne comme une alternative intuitive, économique et adaptée aux besoins spécifiques des organisations informatiques. Conçue pour simplifier le cycle de vie des projets, de la soumission à l'approbation finale, elle intègre des fonctionnalités innovantes et une architecture moderne, répondant aux enjeux de collaboration, de transparence et de sécurité. Pour atteindre ces objectifs, notre plateforme s'appuie sur les principes suivants :

- **Une interface intuitive et épurée** : Conçue pour offrir une expérience utilisateur fluide, elle réduit la courbe d'apprentissage et permet une navigation simplifiée, évitant aux utilisateurs d'être submergés par une surcharge d'options.
- **Un workflow personnalisable** : Nous proposons des processus modulables, adaptés aux spécificités de chaque organisation, garantissant ainsi une gestion fluide des projets, de la soumission à l'approbation finale.
- **Une sécurité et une gouvernance des données renforcées** : Le chiffrement des données et une gestion fine des accès assurent une protection optimale des informations, répondant aux exigences des organisations sensibles.
- **Un modèle de coût abordable** : Notre solution adopte une tarification adaptable aux besoins des petites structures et des startups, tout en proposant des fonctionnalités avancées pour les grandes entreprises.
- **Une plateforme évolutive et adaptable** : Grâce à une architecture modulaire et des mises à jour régulières, notre plateforme s'adapte rapidement aux changements dans les processus métier, garantissant une évolutivité continue.

Cette solution complète vise non seulement à faciliter le processus de gestion des projets informatiques pour ces organisations, mais également à offrir aux entreprises du secteur une opportunité d'optimiser leur productivité, de mieux structurer leurs processus et de renforcer la collaboration entre les équipes. En intégrant des fonctionnalités adaptées aux exigences modernes, elle permet une meilleure visibilité sur l'état d'avancement des projets, une allocation plus efficace des ressources et une prise de décision éclairée.

1.6 Conclusion

Dans ce chapitre, nous avons posé les bases du projet en fournissant un aperçu général de son cadre. Nous avons d'abord présenté l'organisme d'accueil, puis décrit le projet en analysant les solutions existantes dans le domaine de gestion des projets informatiques. Cette étude a permis d'identifier les lacunes des systèmes actuels, orientant ainsi notre réflexion vers une solution plus adaptée, pratique et innovante.

Nous avons ensuite détaillé la solution proposée, conçue pour répondre aux besoins des utilisateurs et de l'entreprise.

Le prochain chapitre sera consacré à la définition des spécifications des besoins, où nous aborderons les différents besoins fonctionnels et non fonctionnels de notre système.

Chapitre 2

Spécification des besoins

2.1 Introduction

Dans ce chapitre, nous allons définir avec précision les exigences de notre projet. Nous commencerons par analyser les besoins fonctionnels et non fonctionnels, qui sont cruciaux pour le développement de notre solution. Ensuite, nous identifierons les différents acteurs impliqués dans le système ainsi que leurs interactions avec celui-ci. Enfin, nous présenterons le backlog produit et le diagramme global des cas d'utilisation, permettant ainsi de structurer le processus de développement et d'assurer que notre application répond de manière efficace aux attentes des utilisateurs.

2.2 Identification des besoins fonctionnels

Un besoin fonctionnel est défini comme une exigence décrivant une action que le système doit pouvoir accomplir, sans prendre en compte de contraintes physiques. Ces besoins sont formulés du point de vue de l'utilisateur.[4] Notre système offre la capacité de :

- S'authentifier
- Échanger des messages
- Accéder aux statistiques
- Administrer ses données personnelles
- Gérer des tâches
- Affecter des développeurs à une tâche
- Gérer les projets
- Gérer les utilisateurs
- Gérer la phase de test
- Gérer les notes internes
- Consulter les tâches à accomplir
- Déposer son projet
- Tester des projets

2.3 Identification des besoins non fonctionnels

Les besoins non fonctionnels sont des critères de qualité essentiels pour l'exécution des besoins fonctionnels, ayant un impact indirect sur les résultats et la performance de l'utilisateur. Leur importance réside dans le fait qu'ils ne doivent en aucun cas être ignorés.[4] Pour garantir cela, il est nécessaire de satisfaire aux exigences suivantes :

- **Sécurité** : Notre plateforme doit garantir la confidentialité des données des utilisateurs ainsi qu'une protection solide de leurs informations personnelles, y compris les noms, adresses, numéros de téléphone et toutes autres données à caractère personnel. Cela nécessite la mise en œuvre de mesures de sécurité avancées, telles que des protocoles de cryptage, des pare-feu et des contrôles d'accès, afin de prévenir tout accès non autorisé.
- **Disponibilité** : Les services doivent être accessibles 24 heures sur 24 et 7 jours sur 7, permettant ainsi aux utilisateurs de réserver des hébergements à tout moment. Les temps d'arrêt doivent être minimisés afin d'éviter toute interruption pour les utilisateurs.
- **Compatibilité** : L'application doit être compatible avec divers navigateurs et appareils, permettant aux utilisateurs d'accéder au site depuis n'importe quelle plateforme. Des tests doivent être effectués sur plusieurs navigateurs et dispositifs pour garantir un fonctionnement optimal.
- **Performance et Utilisabilité** : Le site doit être rapide, réactif et intuitif, évitant ainsi les temps d'attente superflus et assurant une navigation fluide. Les pages doivent se charger rapidement, les menus et boutons clairement étiquetés, et les instructions faciles à suivre pour garantir une expérience utilisateur idéale.
- **Fiabilité** : Notre plateforme s'engage à garantir la fiabilité absolue de chaque service. Chaque processus est soigneusement conçu pour assurer l'intégrité des données et la continuité du service, offrant ainsi à nos utilisateurs une expérience sans faille et une confiance totale dans notre système.
- **Les erreurs** : L'application doit être conçue pour détecter et gérer les erreurs de manière efficace, en fournissant des messages d'erreur clairs et bien structurés. Ces messages orientent l'utilisateur, l'aidant à comprendre rapidement la nature du problème et les étapes à suivre pour le résoudre.

2.4 Identification des acteurs

Un acteur, qu'il s'agisse d'une personne physique ou morale, est une entité impliquée ou influencée par l'action ou le projet en question. Il est donc crucial de définir clairement l'action ou la série d'actions concernées afin d'identifier les acteurs et leur rôle respectif. Dans le cadre de notre plateforme, nous avons identifié deux acteurs qui interagissent directement avec notre système :

- **Superviseur** : C'est l'administrateur qui possède toutes les permissions de contrôle pour gérer les utilisateurs et les ressources, selon les critères établis.
- **Développeur** : Tout utilisateur de la plateforme a pour responsabilité d'exécuter les tâches qui lui sont assignées par le superviseur.

- **Testeur** : Toute personne ayant le rôle de testeur est chargée de valider la qualité et la conformité des projets développés. Il n'intervient que lorsqu'un projet lui est attribué par le superviseur.

Acteur	Rôle
Superviseur	<ul style="list-style-type: none"> — S'authentifier — Échanger des messages — Accéder aux statistiques — Administrer ses données personnelles — Gérer des tâches — Affecter des développeurs à une tâche — Gérer les projets — Gérer les utilisateurs — Gérer la phase de test — Gérer les notes internes
Développeur	<ul style="list-style-type: none"> — S'authentifier — Échanger des messages — Accéder aux statistiques — Administrer ses données personnelles — Consulter les tâches à accomplir — Déposer son projet
Testeur	<ul style="list-style-type: none"> — S'authentifier — Échanger des messages — Accéder aux statistiques — Administrer ses données personnelles — Tester des projets

TABLE 2.1 – Description détaillée des acteurs

2.5 Diagramme de cas d'utilisation globale

Le diagramme de cas d'utilisation vise à identifier les attentes de chaque utilisateur vis-à-vis du système. Notre analyse des besoins repose sur la visualisation des interactions entre les utilisateurs et les fonctionnalités envisagées du système.[5]

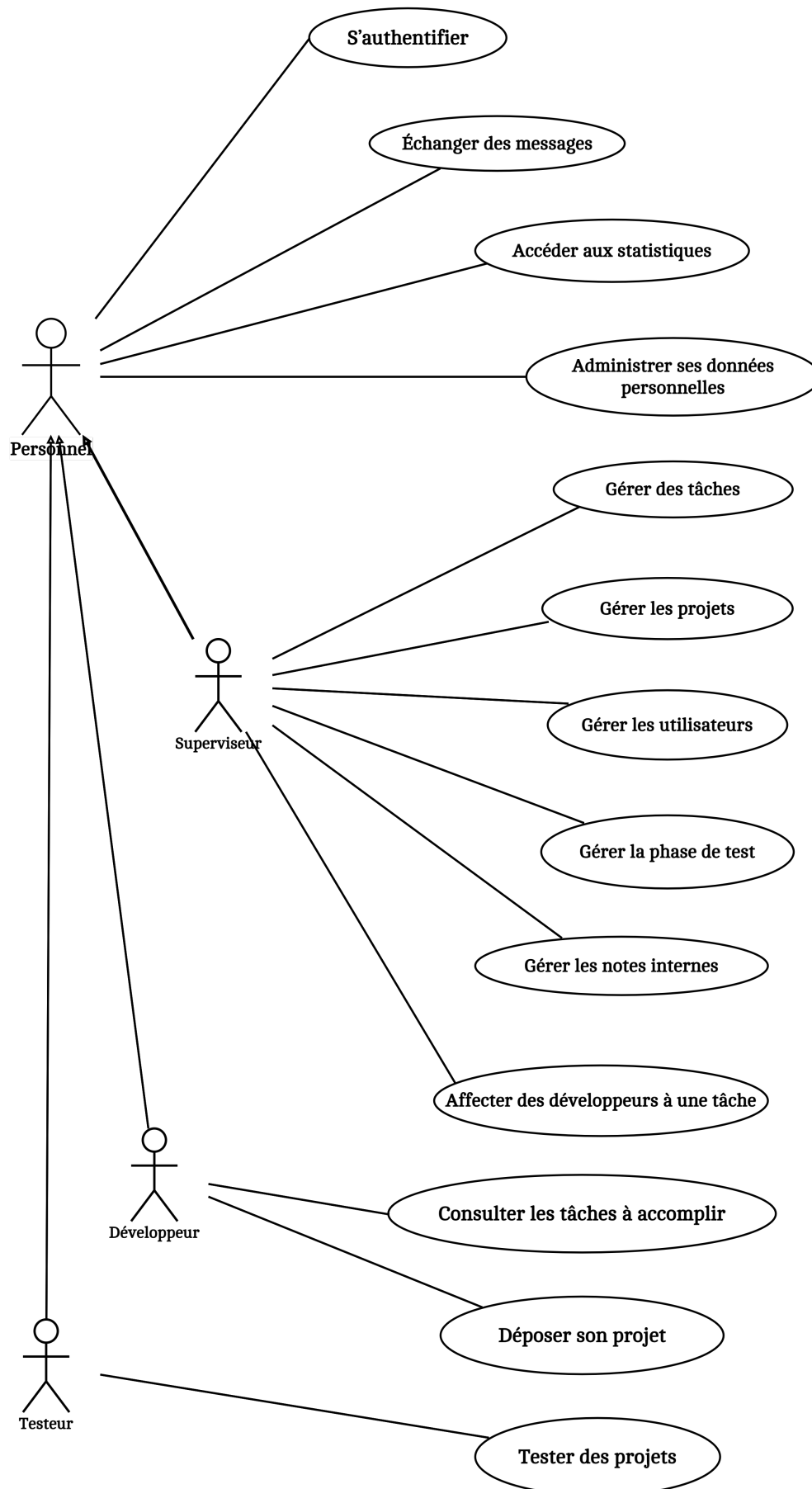


FIGURE 2.6 – Diagramme de cas d'utilisation

2.6 Backlog de produit

Après avoir déterminé les exigences fonctionnelles de notre système, nous exposons dans cette section le backlog du produit.

Backlog de produit	Priorité	Estimation	Planification
En tant que personnel, je peux m'authentifier.	1	Moyen	Sprint 0
En tant que personnel, je peux échanger des messages.	1	Moyen	Sprint 0
En tant que personnel, je peux accéder aux statistiques.	1	Moyen	Sprint 0
En tant que personnel, je peux administrer mes données personnelles.	1	Moyen	Sprint 0
En tant que superviseur, je peux gérer des tâches.	1	Moyen	Sprint 1
En tant que superviseur, je peux affecter des développeurs à une tâche.	1	Moyen	Sprint 1
En tant que superviseur, je peux gérer les projets.	2	Moyen	Sprint 1
En tant que superviseur, je peux gérer les utilisateurs.	2	Moyen	Sprint 1
En tant que superviseur, je peux gérer la phase de test.	2	Moyen	Sprint 1
En tant que superviseur, je peux gérer les notes internes.	2	Moyen	Sprint 1
En tant que développeur, je peux consulter les tâches à accomplir.	2	Moyen	Sprint 2
En tant que développeur, je peux déposer mon projet.	3	Moyen	Sprint 2
En tant que testeur, je peux tester des projets.	3	Moyen	Sprint 2

TABLE 2.2 – Backlog de produit

2.7 Identification des besoins techniques

Un besoin technique se réfère aux conditions, ressources et outils nécessaires pour assurer le bon fonctionnement et la mise en œuvre de notre projet. Dans ce chapitre, nous allons explorer les différents aspects de l'environnement de travail et de l'environnement logiciel, qui jouent un rôle important dans la réalisation de notre solution.[6]

2.7.1 Environnement de travail

Le développement de cette plateforme est réalisé à l'aide d'un ordinateur portable disposant des caractéristiques décrites dans le tableau 2.3.

Caractéristique	Détails
Système d'exploitation	Windows 11
Type du système	Système d'exploitation 64 bits, processeur x64
Processeur	Intel Core i7
Mémoire RAM	32 Go
Stockage	256 Go SSD et 1 To HDD
Carte graphique	NVIDIA GeForce GTX 1650

TABLE 2.3 – Caractéristiques du PC

2.7.2 Environnement logiciel

Dans notre projet, nous avons utilisé les logiciels suivants :

Logiciel	Définition
	Zoom est une plateforme de vidéoconférence multiplateforme offrant réunions, webinaires, partage d'écran, chat de groupe et enregistrement, facilitant le travail à distance et les échanges en ligne. [7]
	Postman est une plateforme pour la gestion et le test des API. Elle permet aux développeurs de créer, exécuter et documenter des requêtes HTTP de manière intuitive, sans compétences en programmation. Avec une interface graphique conviviale, Postman facilite l'organisation des requêtes, le partage entre équipes et l'automatisation des tests, tout en offrant des outils pour simuler des environnements et générer de la documentation API.[8]
	Node.js est un environnement d'exécution JavaScript côté serveur basé sur le moteur V8 de Google Chrome. Il permet aux développeurs de créer des applications web rapides et évolutives en utilisant JavaScript.[9]
	IntelliJ IDEA est un IDE professionnel pour Java et Kotlin, alliant complétion intelligente, refactorisation et analyse statique à une intégration poussée d'outils de versionnage, de build et de débogage, le tout dans une interface ergonomique, personnalisable et multiplateforme pour accélérer le développement et la gestion de projets. [10]
	MySQL est un système de gestion de base de données relationnelle open-source largement utilisé dans le développement d'applications web. Il offre une performance élevée, une fiabilité et une facilité d'utilisation pour stocker et gérer les données.[11]
	Visual Studio Code est un éditeur de code source léger, extensible et multiplateforme développé par Microsoft. Il offre une interface conviviale, une intégration avec de nombreux outils et une large gamme d'extensions pour améliorer la productivité des développeurs.[12]

TABLE 2.4 – Environnement Logiciel

2.7.3 Technologies utilisées

Dans notre projet, nous avons utilisé les technologies suivantes :

Technologie	Définition
	Angular est un framework basé sur TypeScript, développé par Google, qui permet de créer des applications web monopage performantes grâce à une architecture par composants. Il facilite le développement en offrant des fonctionnalités avancées comme la liaison de données, la gestion des composants et l'injection de dépendances.[13]
	Spring Boot est un framework Java qui simplifie la création d'applications Spring autonomes et prêtes pour la production, en réduisant la configuration et en offrant des fonctionnalités comme l'auto-configuration, un serveur web embarqué et une gestion facilitée des dépendances. Il permet aux développeurs de se concentrer sur la logique métier tout en accélérant le développement et le déploiement d'applications web ou de microservices. [14]
	HTML (HyperText Markup Language) est le langage de balisage standard utilisé pour créer des pages web. Il définit la structure et le contenu des pages en utilisant des balises. [15]
	CSS (Cascading Style Sheets) est un langage de feuille de style utilisé pour contrôler la présentation visuelle des pages web écrites en HTML. Il permet de définir la mise en forme, la couleur, la disposition et d'autres aspects visuels d'un site web. [16]
	TypeScript (TS) est un langage de programmation open-source, qui étend JavaScript en ajoutant un typage statique optionnel pour améliorer la sécurité et la maintenabilité du code, tout en restant compatible avec tout code JavaScript valide. [17]

TABLE 2.5 – Technologies utilisées

2.8 Méthodologie de conception

2.8.1 Méthodologie Agile

La méthodologie Agile est une approche itérative et collaborative pour la gestion de projets qui se concentre sur la livraison continue de produits fonctionnels, en mettant l'accent sur la flexibilité, l'adaptabilité et la communication constante entre les membres de l'équipe. Contrairement aux méthodologies traditionnelles, l'Agile favorise des cycles de développement courts, des retours fréquents des parties prenantes et une capacité à s'adapter rapidement aux changements. En intégrant des principes tels que la collaboration, la réactivité aux besoins changeants du projet et la livraison incrémentielle, l'approche Agile permet aux équipes de travailler de manière plus efficace et de produire des résultats de haute qualité tout en répondant aux attentes du client de manière continue.[18]

2.8.2 Scrum

Scrum est un cadre de travail Agile spécifique qui se concentre sur la gestion de projets complexes en mettant l'accent sur la collaboration, la transparence et l'adaptabilité. Basé sur des itérations courtes appelées «**sprints**», généralement de deux à quatre semaines, Scrum encourage les équipes à se concentrer sur des objectifs clairs et à livrer des fonctionnalités utilisables à la fin de chaque sprint. Les rôles clés dans Scrum incluent le Product Owner, chargé de définir les besoins du produit, le Scrum Master, responsable de faciliter le processus et d'éliminer les obstacles, et l'équipe de développement, chargée de réaliser le travail. Les cérémonies régulières telles que la planification du sprint, la revue de sprint et la rétrospective aident à maintenir la transparence et à favoriser l'amélioration continue au sein de l'équipe.

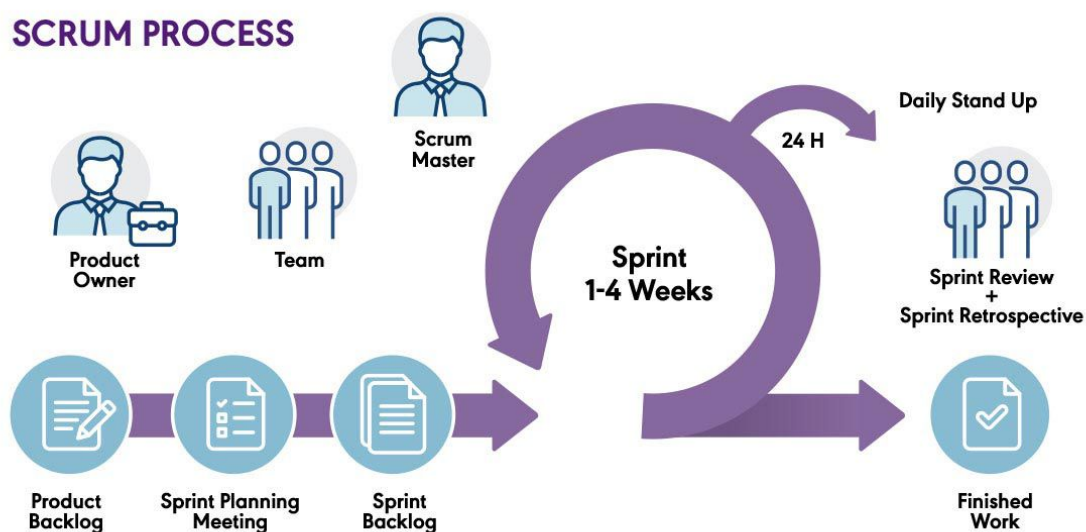


FIGURE 2.7 – Processus Scrum [19]

2.9 Conclusion

Dans ce chapitre, nous avons présenté le diagramme de cas d'utilisation, facilitant ainsi la décomposition fonctionnelle de notre système.

Dans le prochain chapitre, nous aborderons le premier release en détaillant le processus de raffinement, la conception et la réalisation.