00000000000

**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

Université Abderrahmane Mira de Bejaia

Faculté des Sciences Exactes

Département d'Informatique



**RAPPORT DE PROGET AGL**

Développement rapide d’une application Web pour la gestion d’une bibliothèque.

Année universitaire : 2023/2024

**Réaliser par :**

* BERKATI Farah
* BENYOUB Adel
* AYADENE Meriem
* ARAR Manal

**Etudiants en M2GL (G1)**

Table des matières

**Tapez le titre du chapitre (niveau 1)1**

Tapez le titre du chapitre (niveau 2)2

Tapez le titre du chapitre (niveau 3)3

**Tapez le titre du chapitre (niveau 1)4**

Tapez le titre du chapitre (niveau 2)5

Tapez le titre du chapitre (niveau 3)6

# Bibliographie

**Aucune source spécifiée dans le document actif.**

**INTRODUCTION GENERAL :**

Dans le monde numérique d'aujourd'hui, la gestion efficace de l'information est essentielle, surtout dans des domaines cruciaux comme les bibliothèques. Les bibliothèques, en tant que piliers de connaissances, ont évolué au fil des ans pour devenir des centres d'apprentissage dynamiques, nécessitant des systèmes de gestion modernes et sophistiqués. Notre projet ambitieux de développement d'une application web pour la gestion d'une bibliothèque répond à cette nécessité croissante en exploitant les dernières avancées en matière de génie logiciel.

L'objectif central de notre projet est de créer une application de gestion de bibliothèque robuste, conviviale et hautement efficace. Pour atteindre cet objectif, nous allons mettre en œuvre les Ateliers de Génie Logiciel (AGL), un ensemble d'outils et de méthodologies qui vont révolutionner notre approche du développement logiciel. Les AGL offrent une structure qui favorise la collaboration, automatise les tâches répétitives et garantit la qualité, accélérant ainsi le cycle de développement.

**CHAPITRE 1 :**

**PRESENTATION DES AGL’S**

**Introduction :**

Dans ce premier chapitre nous allons présenter les concepts essentiels des Ateliers de Génie Logiciel (AGL), et comprendre leur importance dans le contexte du développement des applications et logiciels. Nous allons examiner les avantages clés des AGL en matière d'efficacité, de collaboration et de qualité des logiciels.

1. **Définition du terme Atelier de Génie Logiciel (AGL) :**

Un ensemble cohérent d'outils informatiques formant un environnement d'aide à la conception, au développement et à la mise au point de logiciels d'application spécialisés. On retrouvera par exemple dans un AGL des dictionnaires de données, des outils permettant de réaliser des diagrammes, pour faciliter la phase d'analyse et de conception des applications. Puis des générateurs de code ainsi que des aides à la mise au point (encore appelés débogueurs) viendront accélérer la production et la finalisation de l'application.

# **Catégories d'AGL :**

Les AGL peuvent être classés selon plusieurs aspects :

* Richesse du support : ensemble d'outils, outils intégrés, aide à la démarche.
* Type de problèmes : logiciels embarqués, temps réel, "business applications", applications métiers
* Type de projet d’ingénierie logicielle : développement logiciel (cycle de vie), intégration de systèmes, système à base de connaissance.
* Ampleur du projet : complexité, nombres de participants, durée ...
* Gestion des ressources du projet : les considérations managériales des ressources mises en œuvre dans le projet sont-elles prises en compte (planification, ordonnancement, …).
* Phase du cycle de développement prises en compte : conception et/ou développement.

# **Critères d'adoption d'un AGL :**

Choisir d'utiliser un AGL pose certains questionnements :

* Investissement de ressources : Coût d'adoption d'une technologie AGL.
* Aide et Support technique disponible : évaluation à long terme de l'exploitation du logiciel.
* Méthodes et processus de GL existants dans l'entreprise : adéquation entre ce qui est fait par les 'acteurs' et ce qui est proposé par les outils
* Montée en charge : aussi bien en termes d'ampleur du projet que de la performance des applications générées avec l'outil.

1. **Les avantages des AGL :**

* **Accélération du Développement :** Les AGL automatisent de nombreuses tâches de développement, ce qui accélère le processus de création de logiciels complexes.
* **Collaboration :** Les AGL favorisent la collaboration entre les membres de l'équipe grâce à des outils de communication en temps réel et de partage de code.
* **Gestion des Versions :** Les AGL intègrent des systèmes de gestion de versions comme Git, ce qui permet de suivre l'évolution du code source, de gérer les branches et de fusionner les modifications de manière transparente.
* **Amélioration de la Qualité :** Les AGL offrent des outils de test automatisés, d'analyse statique du code et de détection des erreurs, ce qui contribue à améliorer la qualité du logiciel.

1. **Flexibilité et Adaptabilité :** Les AGL sont flexibles et peuvent être adaptés aux besoins spécifiques du projet, permettant ainsi aux équipes de travailler selon leurs préférences et leurs méthodologies.
2. **Les différents types des ateliers génie logiciel :**

On distingue essentiellement deux types d'AGL selon la nature des outils intégrés :

1. **Les environnements de conception (upper-case) :** Axés sur l'analyse et la conception, ces ateliers intègrent des outils tels que des éditeurs de diagrammes (avec vérification syntaxique), des dictionnaires de données et des générateurs de code.

**Exemple** : **BOUML** est un logiciel de création de diagrammes UML. Permet de dessiner des diagrammes suivant la norme UML 2.0, Il est multiplateformes.

1. **Les environnements de développement (lower-case**) : ces ateliers se concentrent sur les phases d'implémentation et de test du processus logiciel. Ils intègrent généralement des éditeurs (éventuellement dirigés par la syntaxe), des générateurs d'interfaces homme/machine, des SGBD, des compilateurs, optimiseurs, debuggers.

**Exemple** : **WebDev** (pour la conception d’application web) et **WinDev** Mobile (pour la conception d’application mobile)

**Conclusion :**

Nous avons vu dans cette section, les fondamentales des AGLs en précisant leurs types, leurs avantages dans le développement rapide des logiciels de qualité, et on peut déduire que même si ces AGL ont atteint un niveau élevé, tant en termes de couverture des différentes étapes du processus logiciel, ils restent un sujet controversé et sont loin d'être au niveau attendu.

**CHAPITRE 2 :**

**PRESENTATION DE LA METHODE SCRUM**

## **Définition de la méthode Scrum:**

La méthode Scrum est une méthode agile de gestion de projets informatiques privilégiant la communication, et facilitant les réorientations opportunes. C'est désormais la méthode privilégiée pour les démarches dites "agiles». Fort de son succès dans l'univers informatique, elle est maintenant déployée en entreprise comme nouvelle organisation du fonctionnement en "mode projet.

1. **Principes Fondamentaux de Scrum :**

Scrum est une méthodologie Agile basée sur la transparence, l'inspection et l'adaptation. Les principes clés de Scrum sont la collaboration, la communication fréquente et le développement itératif. Il encourage l'auto-organisation des équipes, la flexibilité face aux changements de priorité et l'attention constante à la qualité.

1. **Rôles dans Scrum :**

* **Product Owner :** Le Product Owner est le gardien de la vision du produit. Il définit les fonctionnalités du produit, priorise le backlog et s'assure que l'équipe de développement comprend clairement les exigences.
* **Scrum Master :** Le Scrum Master est le facilitateur du processus Scrum. Il s'assure que l'équipe suit les principes de Scrum, supprime les obstacles, et facilite les réunions Scrum, aidant ainsi l'équipe à rester concentrée et à progresser.
* **Équipe de Développement :** L'équipe de développement est composée de professionnels qui réalisent le travail. Ils sont auto-organisés, interfonctionnels et ont tous les talents nécessaires pour créer le produit.

1. **Artefacts Scrum :**

* **Product Backlog :** Le Product Backlog est une liste dynamique et priorisée des fonctionnalités à développer. Il est géré par le Product Owner et représente les besoins du client pour le produit.
* **Sprint Backlog :** Le Sprint Backlog est une liste des tâches que l'équipe de développement prévoit de réaliser pendant le sprint en cours. Il est créé à partir du Product Backlog et est géré par l'équipe de développement.
* **Increment :** L'Increment est la somme de toutes les fonctionnalités achevées au terme d'un sprint. Il doit être potentiellement livrable, c'est-à-dire qu'il doit être fonctionnel et prêt à être mis en production.

1. **Événements Scrum :**

* **Sprint :** Un Sprint est une période de temps fixe (généralement deux à quatre semaines) pendant laquelle l'équipe de développement travaille pour créer un Increment potentiellement livrable.
* **Sprint Planning :** Au début de chaque Sprint, l'équipe de développement et le Product Owner se réunissent pour définir les objectifs du Sprint et sélectionner les tâches du Product Backlog à réaliser.
* **Daily Scrum :** Chaque jour pendant le Sprint, l'équipe de développement se réunit pour partager les progrès, discuter des obstacles et planifier le travail pour les prochaines 24 heures.
* **Sprint Review :** À la fin de chaque Sprint, l'équipe de développement présente l'Increment lors d'une réunion appelée Sprint Review. Le Product Owner évalue l'Increment et décide des prochains objectifs.
* **Sprint Retrospective :** Après la Sprint Review, l'équipe de développement se réunit en Sprint Retrospective pour réfléchir sur le Sprint écoulé, identifier ce qui a bien fonctionné et ce qui peut être amélioré, afin d'ajuster leurs pratiques pour les Sprints suivants.

**CHAPITRE 3 :**

**SPECIFICATION DES EXIGENCE DU SYSTEME**

**Introduction :**

# La spécification des exigences, également connue sous le nom de documentation, est un processus consistant à noter toutes les exigences du système et de l'utilisateur sous la forme d'un document. Ces exigences doivent être claires, complètes, exhaustives et cohérentes

# **Exigences fonctionnelles selon chaque acteur de système :**

**Bibliothécaires :**

1. **Ajout de livres :** Les bibliothécaires doivent pouvoir ajouter de nouveaux en fournissant des détails tels que le titre, l'auteur, le genre et le numéro d'inventaire.
2. **Mise à jour des informations :** Les bibliothécaires doivent être en mesure de mettre à jour les informations sur les livres existants, y compris les détails sur les exemplaires disponibles et empruntés.
3. **Gestion des exemplaires :** Les bibliothécaires doivent pouvoir ajouter de nouveaux exemplaires d'ouvrages, spécifiant leur référence et leur localisation dans les rayons.
4. **Gestion des emprunts et des restitutions :** Les bibliothécaires doivent pouvoir enregistrer les emprunts d'ouvrages par les abonnés, y compris la date d'emprunt et la date de retour prévue. Ils doivent également enregistrer les restitutions d'ouvrages, marquant ainsi la fin des emprunts.
5. **Génération de rapports d'emprunt :** Les bibliothécaires doivent pouvoir générer un rapport quotidien de l'ensemble des emprunts (date, heure, code) et un rapport quotidien détaillé de tous les emprunts de la journée, incluant les informations sur les abonnés, les ouvrages, les dates et les heures.

**Membres de la Bibliothèque (lecteur/Abonné) :**

1. **Recherche de livres** : Les membres de la bibliothèque doivent pouvoir rechercher des livres dans la base de données en utilisant des critères tels que le titre, l'auteur, ou la catégorie. Ils doivent pouvoir consulter les détails des exemplaires disponibles, y compris le nombre d'exemplaires disponibles et leur emplacement dans la bibliothèque.
2. **Emprunts d'ouvrages :** Les membres de la bibliothèque doivent pouvoir emprunter des ouvrages en utilisant leur identifiant d'abonné. Le système doit vérifier le nombre maximum d'ouvrages empruntables (trois) par abonné.
3. **Prolongation d'emprunts :** Les membres de la bibliothèque doivent pouvoir demander la prolongation exceptionnelle du délai d'emprunt d'un ouvrage de trois à cinq semaines. La prolongation doit être autorisée uniquement si aucune réservation n'a été faite pour l'ouvrage.
4. **Gestion des réservations :** Les membres de la bibliothèque doivent pouvoir réserver des ouvrages qui sont actuellement empruntés par d'autres membres. Ils doivent également pouvoir annuler des réservations.

**Gestionnaire/Administrateur :**

1. **Gestion des abonnés :** Le gestionnaire doit pouvoir gérer les inscriptions des abonnés, enregistrer leurs informations personnelles et définir la durée de validité de leur abonnement. Il doit avoir la possibilité de définir le statut d'abonné pour les étudiants, les enseignants et les abonnés externes.
2. **Gestion des abonnés externes :** Le gestionnaire doit limiter le nombre d'abonnés externes à environ 10% des inscrits chaque année. Il doit permettre aux abonnés externes de s'inscrire exceptionnellement en échange d'une cotisation.
3. **Gestion des pénalités :** L'administrateur doit être en mesure de gérer les pénalités imposées aux emprunteurs en cas de retards ou de non-restitution d'ouvrages dans les délais. Le système doit bloquer temporairement l'accès aux emprunts pour les emprunteurs pénalisés.
4. **Exigences non fonctionnelles :**

# **Sécurité :** Garantir la confidentialité et la sécurité des données des abonnés.

# **Performances** : Assurer des temps de réponse rapides et une gestion efficace des opérations.

# **Convivialité :** Fournir une interface utilisateur intuitive et conviviale.

# **Extensibilité :** Permettre l'ajout facile de nouvelles fonctionnalités à l'avenir.

# **Maintenabilité :** Faciliter la maintenance, les mises à jour et les corrections de bugs.

# **Disponibilité** : Assurer un accès continu pendant les heures d'ouverture avec une maintenance minimale.

# **Scalabilité** : S'adapter à la variation de la charge en ajoutant des ressources au besoin.

# **Intégration** : S'intégrer avec d'autres systèmes et bases de données universitaires.

**Conclusion :**

Une spécification des exigences est un document qui décrit les besoins spécifiques d'un projet ou d'un système. La spécification des exigences est importante car elle sert de base à tous les travaux futurs sur le projet. La spécification des exigences logicielles (SRS) est différente de la spécification des exigences métier (BRS), bien qu'elles soient liées

**CHAPITRE 4 :**

**PLANNIFICATION DE PROJET ET PESENTATION DES UTILES**

# **Introduction :**

La planification est un élément essentiel pour la réussite de tout projet, et notre projet de gestion de bibliothèque ne fait pas exception. Notre approche combine deux méthodes complémentaires : la planification Agile avec Scrum et la gestion traditionnelle de projet.

Dans la première partie de notre rapport, nous explorerons les principes agiles avec un focus sur les rôles Scrum, les sprints, et le Product Backlog.

 La seconde partie se concentre sur la planification traditionnelle, avec la définition des tâches, le Diagramme de Pert pour visualiser les dépendances, et le Diagramme de Gantt pour la gestion temporelle. Cette combinaison vise à assurer une planification robuste et adaptable pour mener à bien notre projet de gestion de bibliothèque.

1. **Planification Agile avec Scrum :**
2. **présentation d’outils (ClickUp) :**

****

Figure 2: Logo ClickUp.

ClickUp est une plateforme de gestion de projet en ligne qui vise à simplifier et à centraliser la collaboration au sein des équipes. Conçu pour répondre aux besoins variés des projets, ClickUp offre une approche flexible et personnalisable pour la gestion des tâches, la collaboration et la communication

ClickUp propose un éventail de fonctionnalités qui peuvent être adaptées aux principes de la gestion agile. En raison de ses caractéristiques polyvalentes et adaptatives, nous avons conclu que ClickUp serait un choix optimal pour répondre aux besoins spécifiques de notre projet. La flexibilité offerte par la définition des sprints et Product Backlog, permet une gestion agile efficace. De plus, la capacité à suivre en temps réel l'avancement des tâches, à assigner des responsabilités et à gérer les dépendances, offre une adaptabilité cruciale. Cette souplesse, associée à une interface conviviale, favorise une collaboration transparente au sein de l'équipe et facilite la prise de décision proactive.

1. **Les rôles Scrum :**

Dans le cadre de la méthodologie Scrum, l'identification des personnes pour chaque rôle est une étape cruciale pour assurer la clarté des responsabilités et la cohésion de l'équipe. Voici une suggestion de personnes pour chaque rôle :

* **Product owner :** Définir la vision du projet, prendre des décisions éclairées sur les fonctionnalités à développer c’est le rôle de Mm Djebbar.
* **Scrum master :** Personne chargée de faciliter l'application de Scrum au sein de l'équipe dans notre cas ça sera le membre BERKATI Farah
* **Équipe de Développement :** Responsable de la réalisation des éléments du Product Backlog lors des sprints. Représenter par les tous les s de group

1. **planification des sprints :**

La planification des sprints est une composante clé de la méthodologie Scrum, où un sprint est défini comme une itération de développement de courte durée, généralement d'une à quatre semaines. Chaque sprint a pour objectif de livrer une version incrémentale du produit, ajoutant de nouvelles fonctionnalités ou améliorations. Ces périodes définies permettent à l'équipe de se concentrer sur des objectifs spécifiques, favorisant la flexibilité et l'adaptation aux changements.

Au cours de notre projet, nous avons planifié et exécuté quatre sprints successifs. Ces sprints sont clairement détaillés dans le Product Backlog, démontrant notre engagement envers la livraison continue de fonctionnalités de valeur. Chaque sprint a permis à l'équipe de développement de travailler sur des éléments prioritaires du Product Backlog, favorisant ainsi une approche itérative et incrémentale pour atteindre les objectifs du projet.

1. **Product backlog :**

Le Product Backlog, élément central de la méthodologie Scrum, est une liste dynamique et priorisée de toutes les fonctionnalités à développer. Il agit comme un carnet de commandes évolutif, reflétant la vision globale du produit et s'ajustant continuellement en fonction des changements de priorités, des retours d'utilisateurs et des exigences émergentes.

 Les fonctionnalités du produit sont formulées sous forme de “User Stories”, courtes et centrées sur les utilisateurs, offrant ainsi une compréhension claire des besoins à satisfaire.

La priorisation constante du Product Backlog permet de maximiser la valeur livrée à chaque itération et de garantir que l'équipe de développement se concentre sur les éléments les plus cruciaux pour le succès du projet.

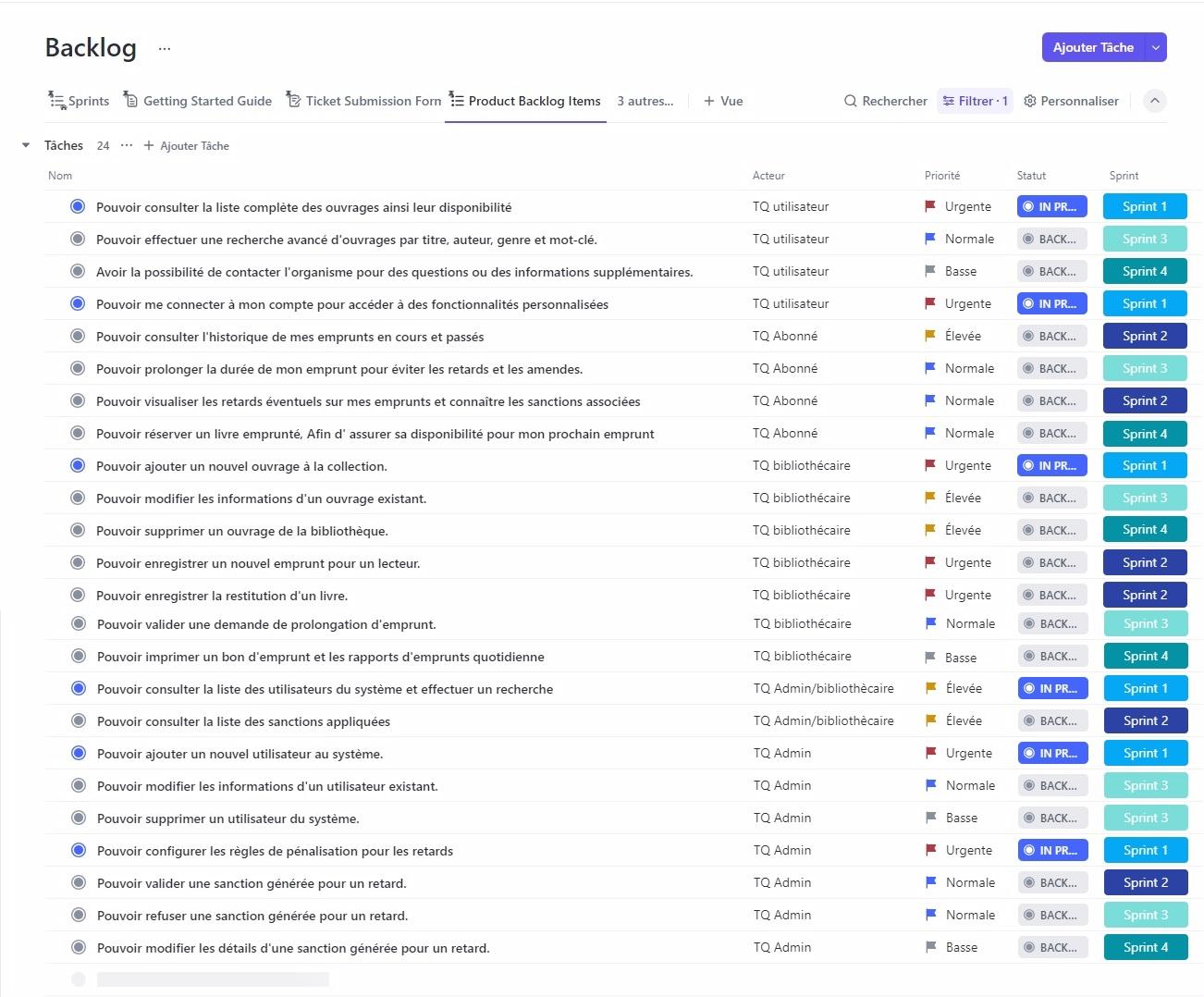
****

Figure 3: Product Backlog.

1. **Planification et gestion de projet :**
2. **Définition des tâches :**

La définition des tâches dans un projet constitue un pilier essentiel de la gestion, apportant une clarté cruciale sur les objectifs spécifiques, simplifiant la répartition des responsabilités, permettant une planification temporelle réaliste, identifiant les interdépendances entre les activités, facilitant le suivi de l'avancement, et renforçant la communication au sein de l'équipe. Ces aspects fondamentaux établissent une base robuste pour une gestion proactive et couronnée de succès du projet.

 Le tableau ci-dessous offre une représentation claire et structurée des tâches essentielles de notre projet

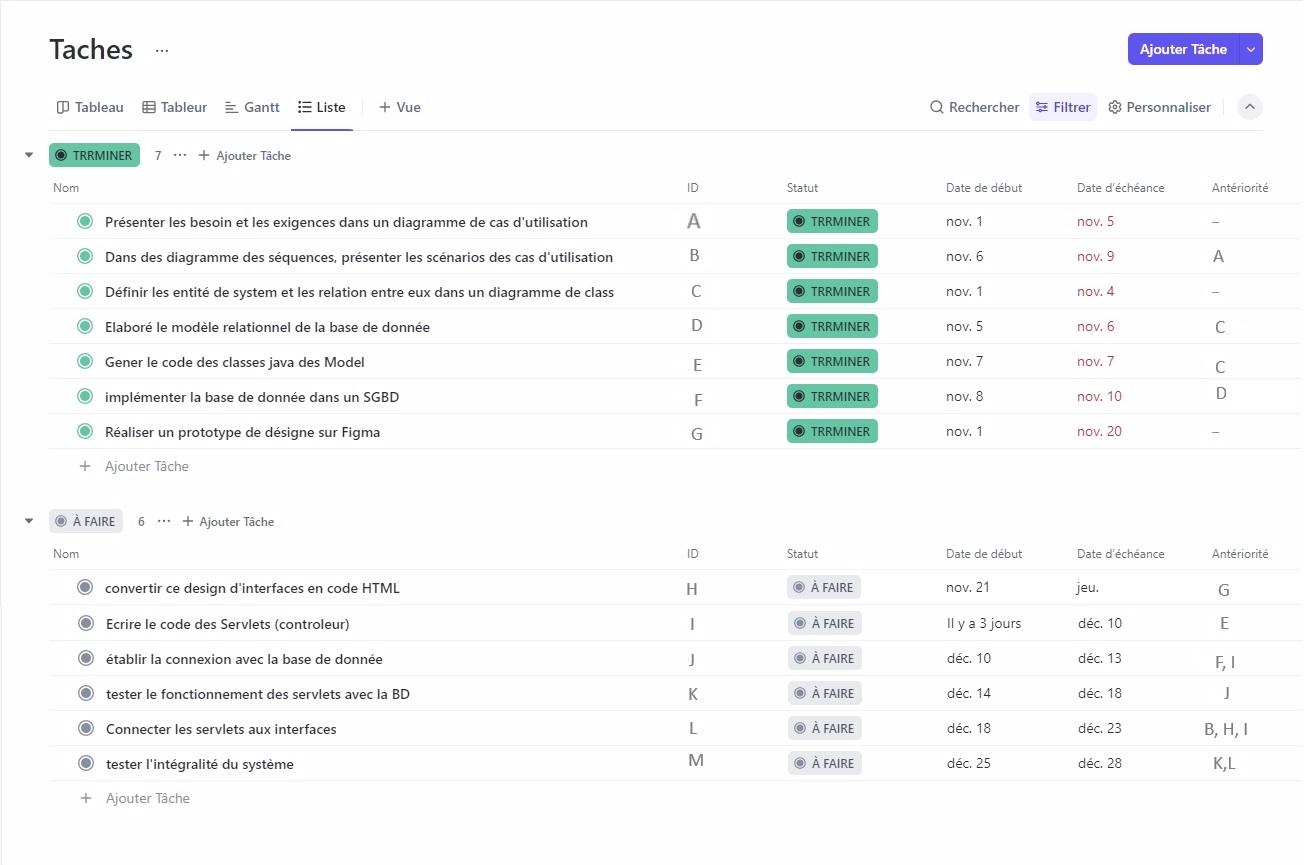
: ****

Figure 4: Liste des tâches.

1. **Diagramme de PERT**

Le diagramme de Pert, acronyme de Program Evaluation and Review Technique, est un outil de gestion de projet qui permet de représenter graphiquement les différentes étapes et les dépendances entre les tâches.

Chaque tâche est représentée par un nœud, et les flèches entre les nœuds indiquent les dépendances et le flux logique du projet. Le diagramme de Pert offre une visualisation claire du chemin critique, c'est-à-dire la séquence d'activités qui détermine la durée totale du projet. Il permet également d'identifier les tâches qui peuvent être effectuées en parallèle.

Le diagramme ci-dessous offre une représentation visuelle des tâches de projet définies dans le tableau des tâches

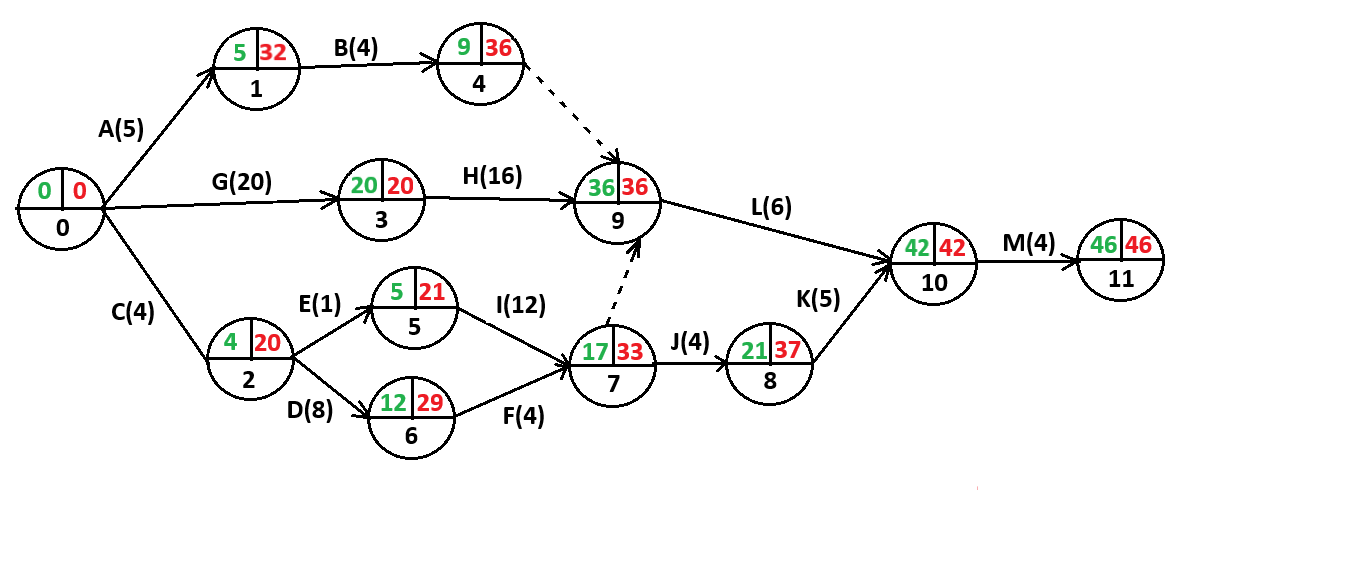
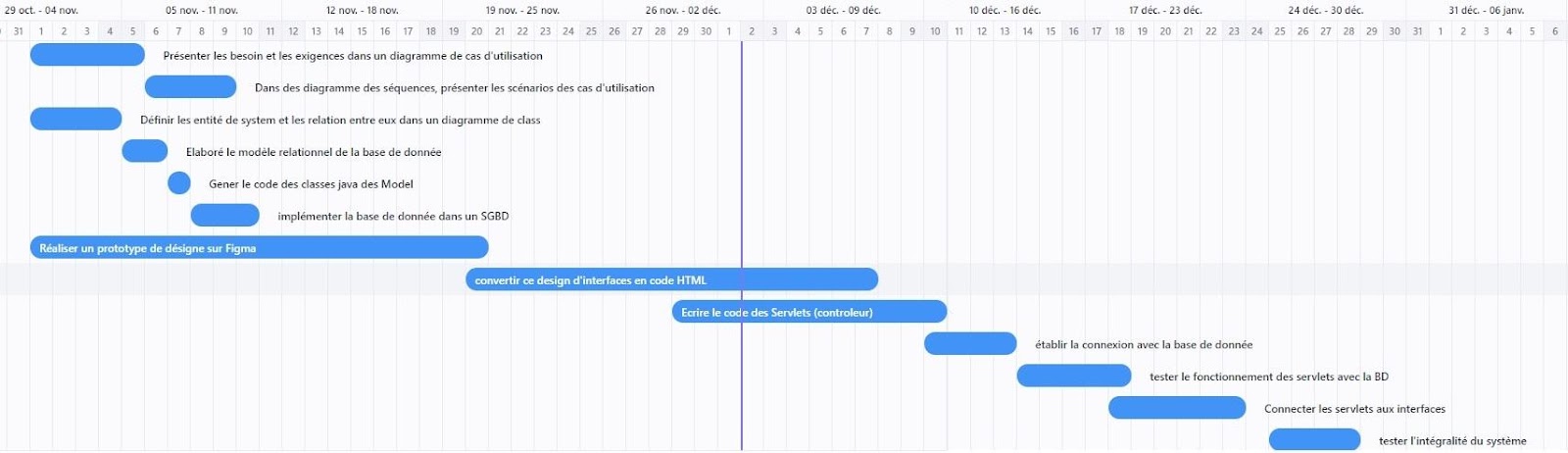
****

Figure 5: Diagramme de PERT.

1. **Diagramme de Gantt**

Le diagramme de Gantt est un outil de gestion de projet visuel qui illustre les tâches du projet sur une ligne de temps. Il offre une représentation graphique des activités planifiées, de leur séquence et de leur durée. Chaque tâche est représentée par une barre horizontale dont la longueur indique la durée prévue, permettant ainsi de visualiser clairement le début, la fin et le chevauchement des différentes phases du projet.

Le diagramme de Gantt généré à partir du tableau des tâches offre une vue chronologique des activités planifiées. Chaque tâche est représentée par une barre horizontale positionnée sur la ligne de temps. Cette représentation visuelle facilite la compréhension des échéances, des chevauchements et des périodes critiques du projet.

****Figure 6: Diagramme de Gantt.

# Conclusion :

La combinaison stratégique de méthodes agiles comme Scrum avec des outils traditionnels a abouti à une planification solide et flexible pour notre projet de gestion de bibliothèque. Cette approche hybride a favorisé une collaboration fluide, une focalisation précise sur les objectifs tout en restant adaptable aux évolutions, assurant ainsi une gestion proactive et réussie.

**CHAPITRE 5 :**

**REALISATION**

**CONCLUSION GENERALE:**

**RESUME:**