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)**

**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**

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.
* **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.

**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**

# **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 :**

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.

# **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.

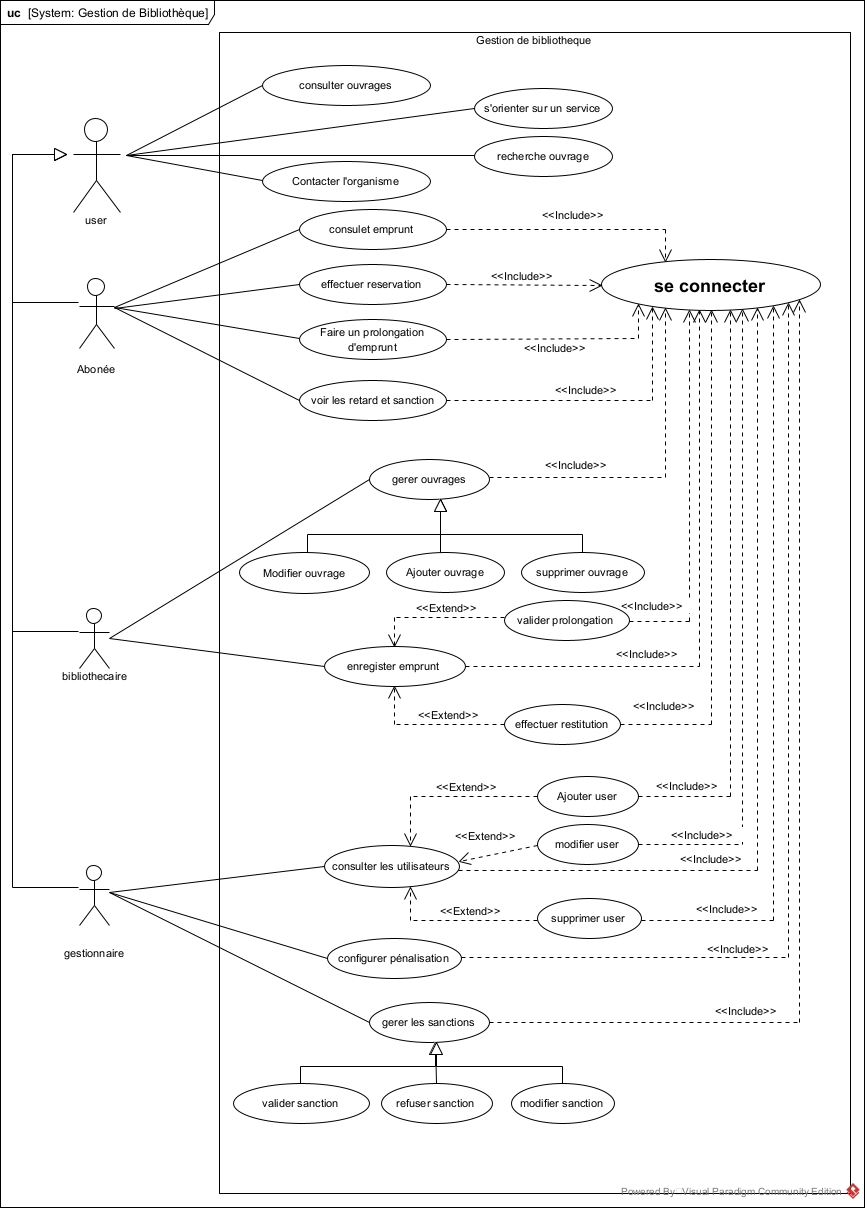
# **Disponibilité** : Assurer un accès continu pendant les heures d'ouverture avec une maintenance minimale.

# **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.

**CHAPITRE 4 :**

**PLANNIFICATION DE PROJET ET PESENTATION DES UTILES**

****

**CHAPITRE 5 :**

**REALISATION**

**CONCLUSION GENERALE:**

**RESUME:**