

Rapport de Projet "File rouge"

Développement d'un système de gestion de l'apprentissage : " Validation des projets réalisés"

SPÉCIALITÉ :

Développement Mobile - Mode Bootcamp

Supervisé par : ESSARRAJ FOUAD

Réalisé par : YASMINE DAIFANE

Année de formation 2023/2024

Table des matières

Table des matières.....	2
Listes des figures.....	3
Remerciement.....	4
Introduction.....	5
Cahier de charge.....	6
Contexte de projet.....	6
Objectifs de formation.....	7
Processus de développement et méthodes.....	8
2TUP (Two Tracks Unified Process).....	8
Design thinking.....	9
Développement agiles.....	11
Méthode agile.....	11
Méthode Scrum.....	11
Planification.....	13
Diagramme de Gantt.....	13
Branche fonctionnelle.....	14
Empathie.....	14
Définir le problème.....	15
Idéation.....	16
Diagramme de cas d'utilisation.....	16
Branche technique.....	17
Capture des besoins techniques.....	17
Architecture de l'application :.....	17
Prototype.....	18
Conception.....	19
Diagramme de classes.....	19
Diagramme de packages.....	20
Maquette.....	21
Réalisation.....	22
Technologies Utilisées.....	22
Outils utilisés.....	24
Conclusion.....	25
Annexe 1 : Planification.....	26
Annexe 2 : Maquettes.....	26

Listes des figures

Figure 1: Introduction.....	5
Figure 2: Context de projet.....	6
Figure 3: Schéma Objectifs de formation.....	7
Figure 4: 2TUP.....	8
Figure 5: Design thinking.....	9
Figure 6:Méthode agile.....	11
Figure 7: Les phases d'une méthode Scrum.....	12
Figure 8: Diagramme de Gantt.....	13
Figure 9 : Carte d'empathie.....	14
Figure 10 : Diagramme de cas d'utilisation.....	16
Figure 11: Architecture de l'application.....	18
Figure 12:Conception - pkg_validations.....	19
Figure 13 : Diagramme packages - pkg_global.....	20
Figure 14 :Maquette - pkg Réalisations-projets.....	21
Figure 15 :Maquette - pkg validations.....	21
Figure 16 : Backlog pkg_validations.....	26
Figure 17 :Maquette - pkg validations.....	26

Remerciement

Je tiens tout d'abord à exprimer ma profonde gratitude à Dieu pour m'avoir accordé la force, la détermination et la patience nécessaires à l'accomplissement de ce projet de fin d'études.

Je remercie également chaleureusement mon encadrant, Monsieur Fouad, pour son soutien indéfectible, ses précieux conseils et sa patience tout au long de cette aventure. Son expertise et sa disponibilité ont été des atouts majeurs pour la réussite de ce travail.

Je n'oublie pas mes collègues et amis, qui m'ont apporté leur aide, leur encouragement et leur soutien moral. Leur camaraderie et leur collaboration ont grandement enrichi mon expérience.

Enfin, je dédie une pensée spéciale à ma mère, dont l'amour, les sacrifices et le soutien constant ont été une source d'inspiration et de motivation. Merci de m'avoir toujours encouragé à donner le meilleur de moi-même.

Introduction

Soli-LMS (Learning Management System) est un système de gestion de la formation puissant et polyvalent conçu pour satisfaire les besoins variés des organismes de formation et des établissements d'enseignement. Soli-LMS offre une gestion efficace de toutes les étapes de la formation, de la création des projets à l'évaluation finale des apprenants. Les formateurs ont la capacité de créer et d'organiser des projets de formation, de définir des objectifs précis, de structurer les activités pédagogiques et de fournir les ressources nécessaires aux apprenants. De leur côté, les apprenants bénéficient d'un suivi personnalisé de leur progression, ont la possibilité de consulter leurs notes et commentaires, et peuvent accéder facilement aux ressources pédagogiques disponibles. Soli-LMS facilite également la gestion des ressources humaines, des modules de formation, des compétences, et des parcours d'autoformation, offrant ainsi une solution complète pour une gestion intégrée et efficace des formations.

Dans le cadre de notre projet, je me concentre principalement sur le développement et l'amélioration du package de validation. La mise en place de ce package est vitale pour s'assurer que les projets de formation respectent les normes de qualité attendues, et que les apprenants reçoivent des évaluations justes et constructives. En validant chaque compétence associée à ces projets, les formateurs peuvent valider les projets réalisés par les apprenants grâce au package de validation.

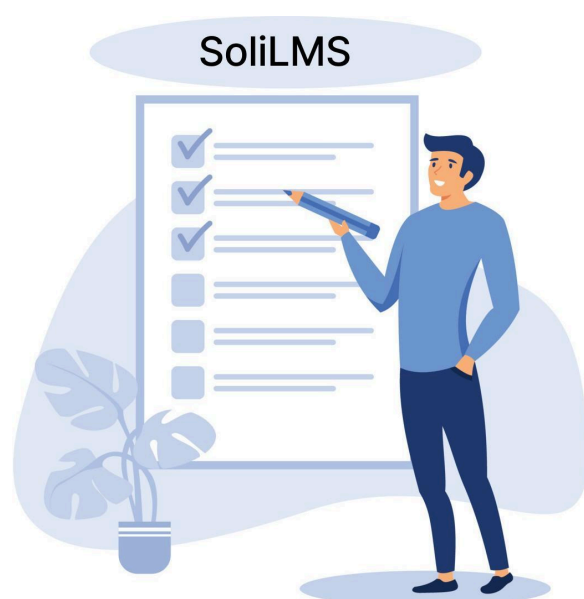


Figure 1: Introduction

Cahier de charge

Dans ce chapitre, nous allons explorer en détail le contexte du projet ainsi que les objectifs spécifiques de la formation. Nous expliquerons les raisons qui ont motivé cette initiative, en examinant les défis et opportunités actuels de l'entreprise. Ensuite, nous définirons clairement les compétences et connaissances que les participants doivent acquérir, ainsi que l'impact attendu sur leur performance professionnelle .

Contexte de projet

Le Centre Solidaires Digital SOLICODE a un besoin pressant d'une solution de gestion de formation moderne et performante. Pour répondre aux exigences croissantes de qualité et d'accessibilité des formations, le centre souhaite centraliser et optimiser ses processus pédagogiques.

Soli-LMS a été développé pour répondre à ces besoins spécifiques.



Cette plateforme permettra au centre de structurer et de gérer efficacement l'ensemble du cycle de formation, de la création des projets à l'évaluation des apprenants, tout en offrant une interface conviviale et des fonctionnalités complètes pour améliorer la qualité et l'efficacité des formations.

Figure 2: Contexte de projet

Objectifs de formation

Ce projet Soli-LMS vise à mettre en pratique et à présenter mes compétences en développement web, conception et gestion de projet. La plateforme sera conviviale et responsive, garantissant une expérience optimale sur divers appareils. Elle utilisera Laravel et MySQL pour une gestion efficace des données, simplifiant ainsi les processus et offrant des possibilités d'améliorations continues.

La plateforme offrira une interface conviviale aux clients pour accéder facilement aux services proposés. Elle intégrera des fonctionnalités favorisant une communication fluide et une évaluation précise des projets des apprenants, contribuant ainsi à leur développement de compétences de manière progressive .



Figure 3: Schéma Objectifs de formation

Processus de développement et méthodes

Le processus de développement logiciel est une approche structurée pour créer ou améliorer un système logiciel. Ce chapitre est divisé en deux parties : la première partie aborde le 2TUP (Two-Track Unified Process), tandis que la deuxième partie se concentre sur le Design Thinking.

2TUP (Two Tracks Unified Process)

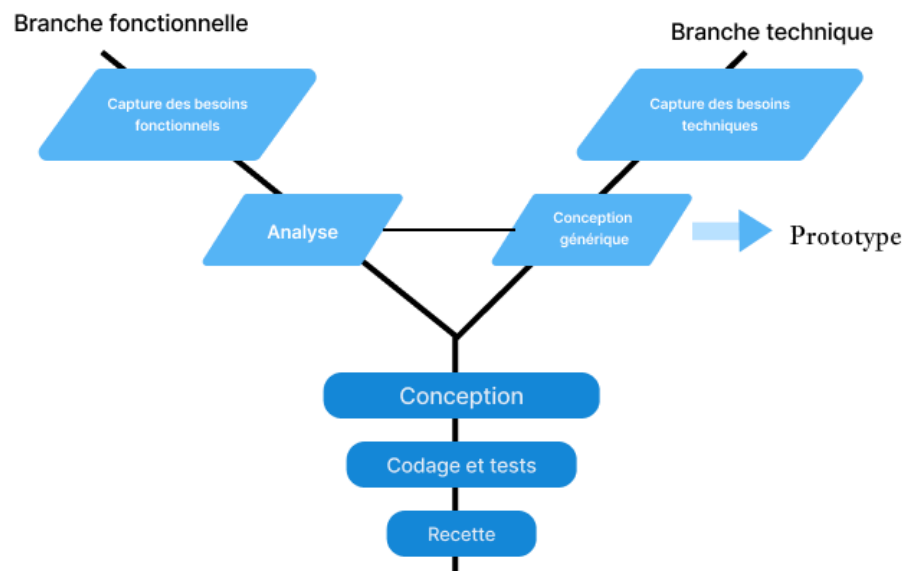


Figure 4: 2TUP

2TUP est un processus de développement logiciel qui implémente le Processus Unifié. Il commence par une étude préliminaire qui consiste essentiellement :

- cahier des charges .
- modéliser le contexte .
- identifier les acteurs qui vont interagir avec le système à construire

Le processus s'articule ensuite autour de 3 phases essentielles :

- **Branche fonctionnelle** qui consiste en la modélisation et le maquettage pour clarifier les besoins fonctionnels.
- **Branche Technique** qui recense toutes les contraintes à respecter pour réaliser le système. Elle définit ensuite les composants nécessaires à la construction de l'architecture technique.
- **La phase de réalisation** consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins. Arrive par la suite l'étape de codage et enfin l'étape de recette, qui consiste à valider les fonctions du système développé.

Design thinking

Le design thinking est une approche de l'innovation centrée sur l'humain. C'est une méthode ou un processus de conception globale, centré sur l'utilisateur (ou l'humain), en vue de réaliser des services ou produits innovants.

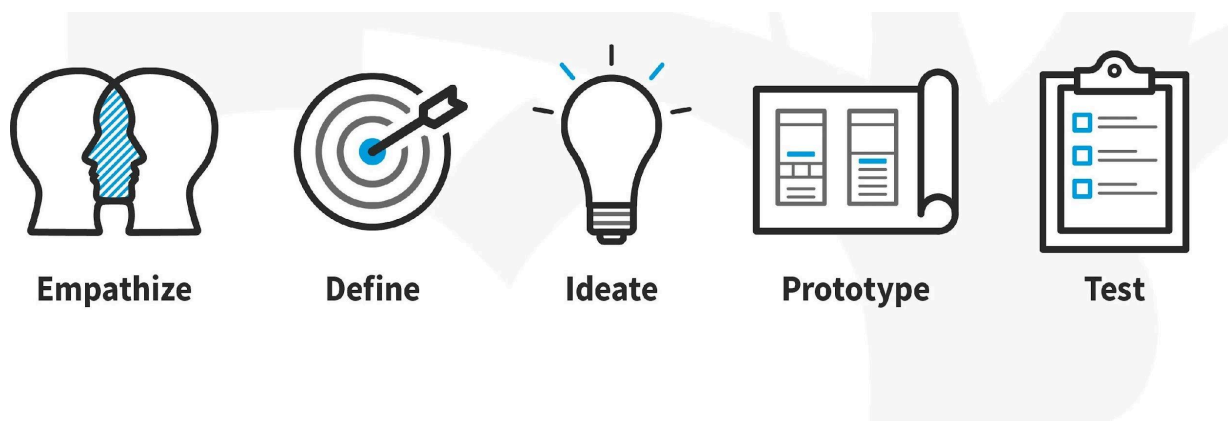


Figure 5: Design thinking

Empathie

Comprendre les besoins et les perspectives des utilisateurs concernés par le problème à résoudre. Cela implique d'observer, d'écouter et d'interagir avec les utilisateurs pour acquérir une compréhension approfondie de leurs besoins, motivations et défis.

Définir le problème

Définir clairement le problème à résoudre en se basant sur les informations recueillies lors de la phase d'empathie.

Idéation

Générer un grand nombre d'idées créatives pour aborder le problème identifié. Encourager la pensée divergente, où toutes les idées sont les bienvenues, sans jugement, afin de favoriser l'exploration de différentes solutions possibles.

Prototype

Créer des prototypes tangibles ou des représentations visuelles des solutions potentielles. Les prototypes peuvent être simples et rapides à réaliser, mais doivent être suffisamment concrets pour recueillir des retours et les itérer.

Test

Tester les prototypes auprès d'utilisateurs réels. Les premières maquettes comportent nécessairement des erreurs, qu'il faut identifier au plus vite et dont il faut ensuite tirer des enseignements.

Développement agiles

Le développement Agile est une approche flexible et itérative qui vise à fournir des versions fonctionnelles du logiciel de manière régulière, en répondant aux besoins changeants des utilisateurs. Dans ce chapitre, nous allons d'abord expliquer les principes fondamentaux de l'Agilité, ses avantages et ses limites. Ensuite, nous allons nous concentrer sur Scrum, l'une des méthodes Agile les plus populaires, en expliquant ses rôles clés, ses événements, ses artefacts et les bonnes pratiques pour réussir un projet Scrum.

Méthode agile

La méthode agile, également connue sous le nom de développement agile ou gestion de projet agile, est une approche itérative et collaborative de la gestion de projets et de développement logiciel. Elle met l'accent sur l'adaptabilité, la flexibilité et la livraison continue de produits ou de fonctionnalités.

L'objectif principal des méthodes agiles est de permettre aux équipes de répondre de manière plus efficace aux besoins changeants des clients, de réduire les risques liés au développement de produits, d'améliorer la qualité du produit final et d'optimiser la satisfaction des parties prenantes.

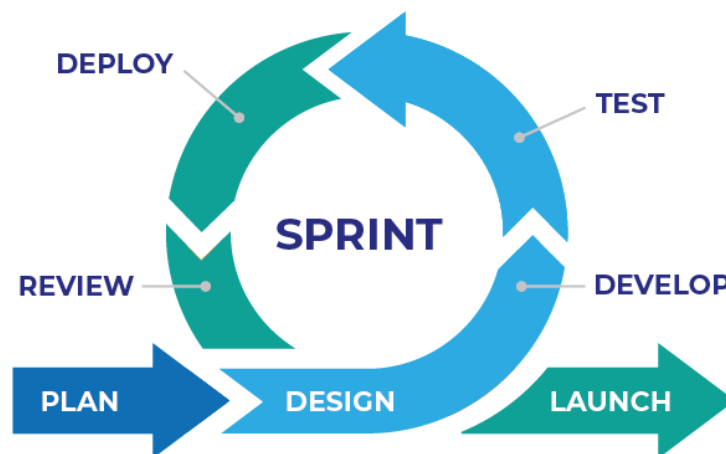


Figure 6: Méthode agile

Méthode Scrum

Scrum est une méthodologie de gestion de projet agile qui se base sur des cycles de travail itératifs appelés « sprints ». Il implique un Product Owner, un Scrum Master, une équipe de développement et utilise des événements tels que les réunions quotidiennes, les revues de sprint et les rétrospectives de sprint pour favoriser la collaboration, la transparence et la livraison régulière de fonctionnalités de haute qualité. Le Scrum permet aux équipes de répondre rapidement aux changements et de livrer de la valeur plus rapidement.

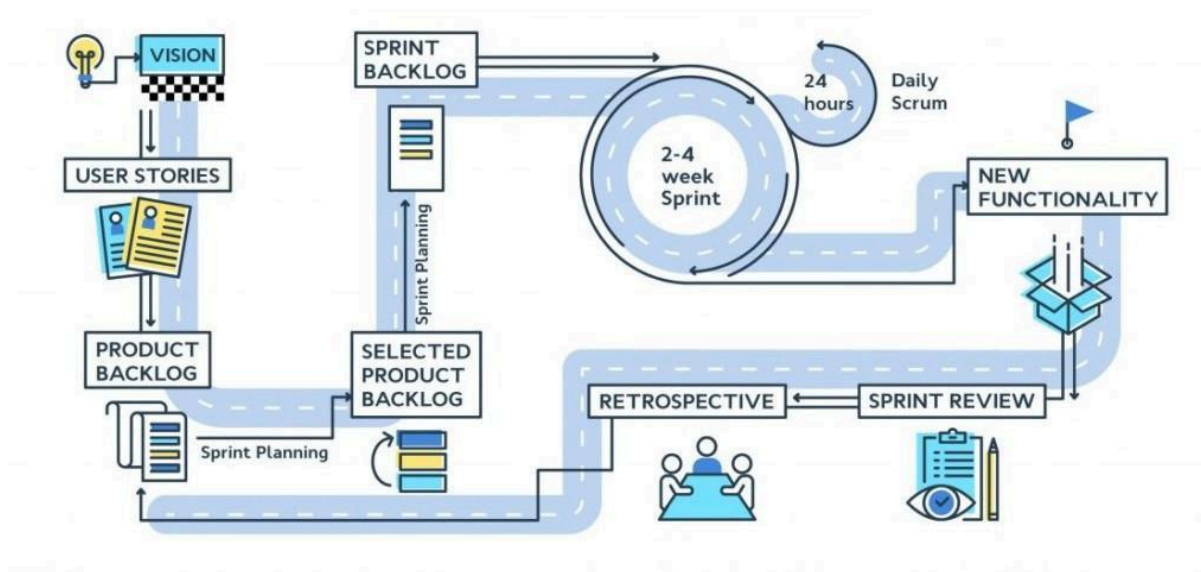


Figure 7: Les phases d'une méthode Scrum

Planification

Pour assurer la réussite de notre projet, il est impératif de créer un plan détaillé qui énumère toutes les tâches à réaliser en utilisant la méthode que nous avons présentée précédemment. Ce plan nous permettra d'obtenir une vue d'ensemble du projet, de définir les objectifs clairement définis et d'identifier les ressources requises pour chaque tâche.

Diagramme de Gantt

Dans cette partie de mon travail , j'ai élaboré un diagramme de Gantt pour tracer la trajectoire de développement de mon package de validation. Ce diagramme offre une représentation visuelle claire des différentes étapes du projet et de leur calendrier, permettant ainsi une gestion efficace du temps et une meilleure compréhension de la progression des tâches .

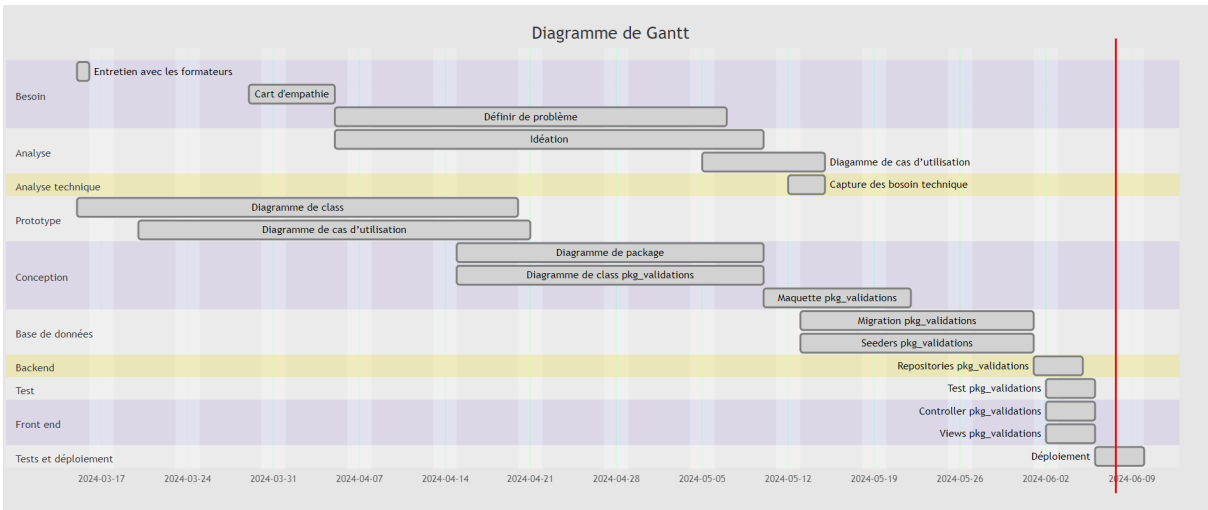


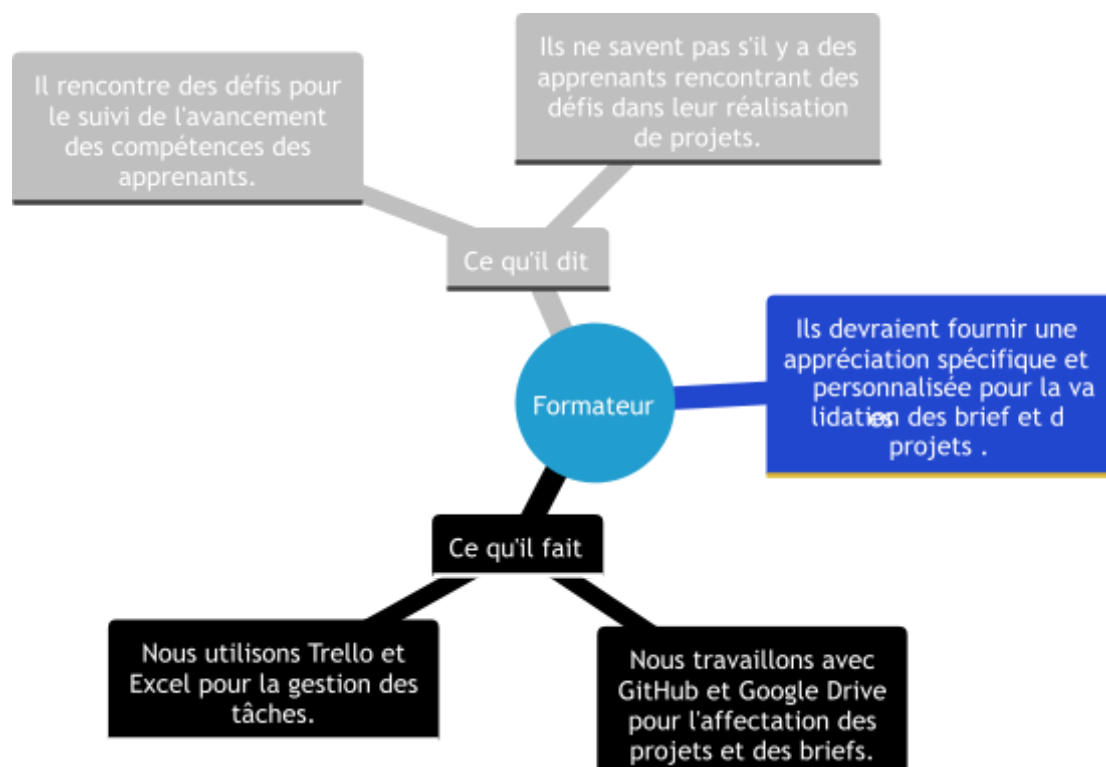
Figure 8: Diagramme de Gantt

Branche fonctionnelle

Dans cette phase de capture des besoins fonctionnels de la méthode 2TUP, en utilisant le processus de développement comme la carte d'empathie et la définition du problème, ainsi que des techniques d'idéation pour identifier les besoins des utilisateurs. aussi, l'utilisation d'UML(diagrammes de cas d'utilisation et les diagrammes de classe).

Empathie

Une séance d'empathie a été réalisée avec les formateurs de centre Solicode le 19 janvier 2024.



Carte d'empathie de Formateur

Figure 9 : Carte d'empathie

Persona : Formateur

Objectifs et Responsabilités

- Assurer le bon déroulement des projets et des briefs assignés aux apprenants.
- Suivre et évaluer l'avancement des compétences des apprenants.
- Fournir un feedback spécifique et personnalisé pour valider les briefs et les projets.

Défis et Frictions

- Manque de visibilité sur les progrès individuels des apprenants.
- Difficultés à détecter les apprenants en difficulté sans feedback détaillé.
- Temps et ressources limités pour fournir un feedback personnalisé à chaque apprenant.

Besoin et Attentes

- Outils ou méthodes pour un suivi plus efficace de l'avancement des compétences.
- Mécanismes pour identifier rapidement les apprenants en difficulté.
- Processus clairs et efficaces pour fournir un feedback détaillé et personnalisé.

Définir le problème

Le centre Solicode manque d'un système efficace de gestion des projets de formation, ce qui entraîne des difficultés dans la mise en place de critères uniformes d'évaluation, l'intégration des outils de gestion et de communication, et la cohérence des appréciations de validation des compétences des apprenants.

Idéation

Créer une application web conviviale pour le centre Solicode qui permettra :

- La validation des projets des apprenants.
- Aux formateurs de valider chaque compétence ciblée dans un projet ou un brief.
- L'intégration de appréciation de validation personnalisables.
- L'inclusion de champs permettant aux formateurs d'ajouter des remarques et des commentaires détaillés.

Diagramme de cas d'utilisation

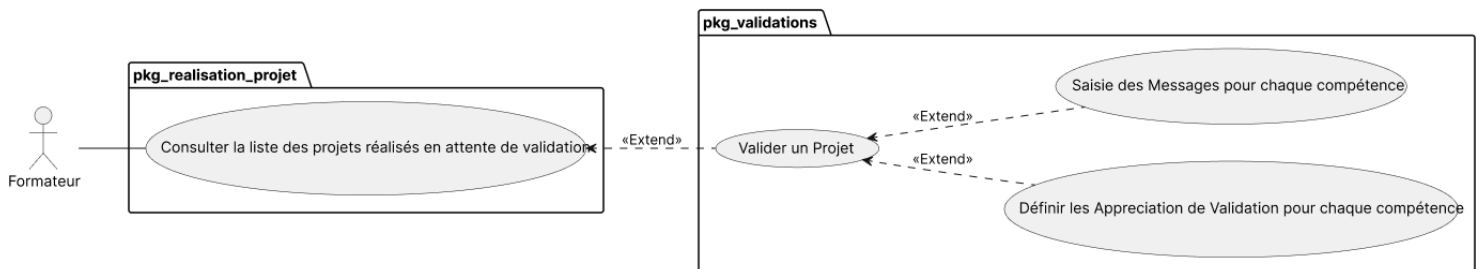


Figure 10 : Diagramme de cas d'utilisation

Branche technique

La branche technique fait référence à la partie du projet qui est axée sur les aspects techniques et de développement. Cette branche englobe les activités liées à la conception, la programmation, et la mise en œuvre des fonctionnalités et des composants du projet. Elle implique généralement l'utilisation d'outils de développement, la gestion des versions du code source, et la collaboration entre les développeurs. La branche technique joue un rôle essentiel dans la création d'un produit final de haute qualité, en veillant à ce que les spécifications techniques soient respectées et que les fonctionnalités soient correctement implémentées.

Capture des besoins techniques

Pour le développement de l'application Soli-LMS, nous avons choisi Laravel comme framework backend pour sa robustesse et ses fonctionnalités avancées, associé à MySQL pour la gestion des données en raison de sa fiabilité. Pour l'interface, AdminLTE a été sélectionné pour son design moderne et ses fonctionnalités avancées, assurant une expérience utilisateur conviviale. En intégrant ces technologies avec des design patterns comme les repositories, notre objectif est de développer une application solide, performante et facile à utiliser, répondant de manière efficace et optimale aux besoins de gestion de formation. L'utilisation des repositories garantit une séparation claire des préoccupations entre le backend et le frontend, améliorant ainsi la maintenabilité, la scalabilité et la performance globale de l'application.

Architecture de l'application :

L'architecture de notre application suit le modèle Modèle-Vue-Contrôleur (MVC) avec une couche supplémentaire pour le Repository Pattern. Le Modèle-Vue-Contrôleur est un modèle d'architecture logicielle qui divise une application en trois parties principales : le modèle (qui représente les données et la logique de l'application), la vue (qui affiche les données au client) et le contrôleur (qui gère les entrées utilisateur et coordonne les interactions entre le modèle et la vue).

Le Repository Pattern ajoute une couche supplémentaire d'abstraction qui permet de séparer la logique de stockage des données de la logique métier de l'application. Cette couche est responsable de l'accès aux données, de leur persistance et de leur récupération. Cette abstraction permet une meilleure maintenabilité et évolutivité de l'application. En utilisant cette architecture, nous avons pu séparer les différentes responsabilités de l'application, ce qui facilite la maintenance et l'évolution de celle-ci.

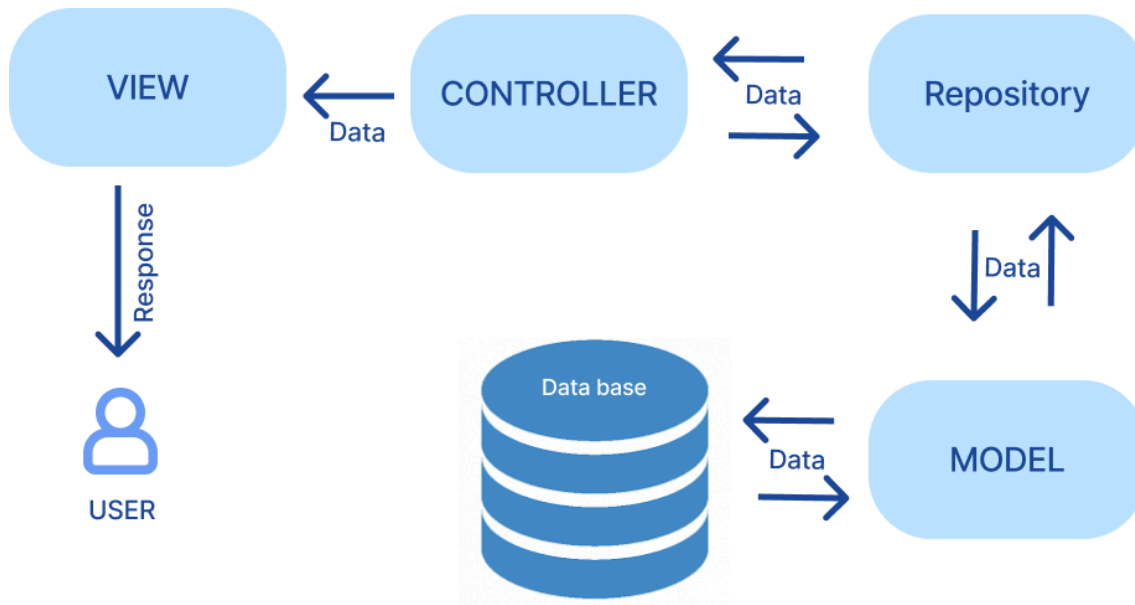


Figure 11: Architecture de l'application

Prototype

Avant de démarrer la réalisation de notre projet Soli-LMS, nous avons mis en place un prototype fonctionnel détaillé qui a été développé sur trois sprints distincts :

- Gestion des Projets (Sprint 1) : Nous avons débuté avec la création du sprint de gestion des projets, qui comprend les opérations CRUD basiques pour la gestion des projets. Ces opérations incluent la création, la lecture, la mise à jour et la suppression de projets, ainsi que leur association avec les utilisateurs et les tâches. Nous avons également intégré des fonctionnalités d'importation et d'exportation de données pour permettre aux utilisateurs de manipuler facilement les données et d'effectuer des recherches dynamiques avec Ajax .
- Gestion des Tâches (Sprint 2) : Bien que prévu dans le planning, ce sprint n'a pas encore été réalisé. Il sera dédié à la gestion des tâches au sein des projets, y compris la création, l'assignation, le suivi et la collaboration sur les différentes tâches.
- Gestion de Stock (Sprint 3) : Également en attente de réalisation, ce sprint sera axé sur la gestion des stocks et des ressources matérielles nécessaires à la réalisation des projets. Cela comprend la gestion des stocks, des commandes, des entrées/sorties de matériel, et d'autres fonctionnalités liées à la gestion des ressources.

En parallèle, le prototype intègre un système d'authentification et d'autorisation utilisant Spatie pour assurer la sécurité et la gestion des droits d'accès des utilisateurs. Ce processus itératif nous a permis de valider rapidement les principales fonctionnalités de l'application, de recueillir des retours d'utilisateurs pour optimiser l'expérience utilisateur, et de garantir une transition fluide vers la phase de développement complète.

Conception

Ce chapitre présente le diagramme de classe et les maquettes de l'interface utilisateur du package Validation. Le diagramme de classe offre une visualisation structurée des différentes classes et de leurs relations spécifiques à ce package. Les maquettes permettent de visualiser concrètement l'apparence et l'agencement de l'interface utilisateur, en mettant en évidence les fonctionnalités clés liées à la validation des projets réalisés par les apprenants. Ces éléments sont essentiels pour comprendre l'architecture du package Validation et pour guider le processus de développement, garantissant une conception cohérente et conviviale.

Diagramme de classes

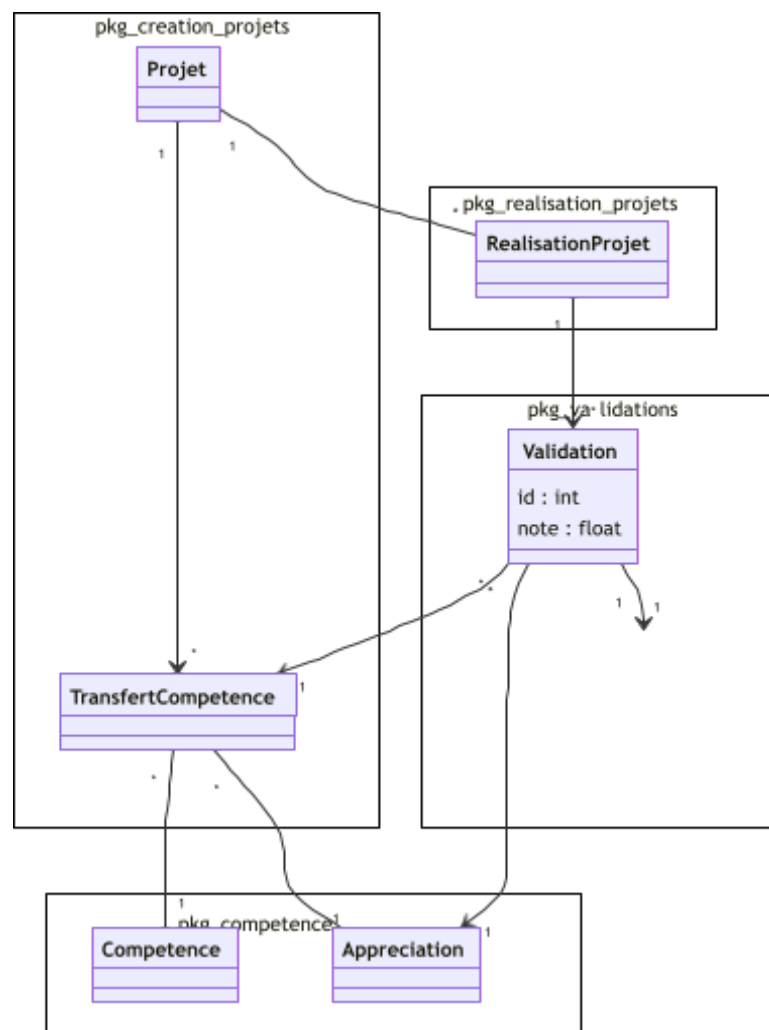


Figure 12:Conception - pkg_validations

Diagramme de packages

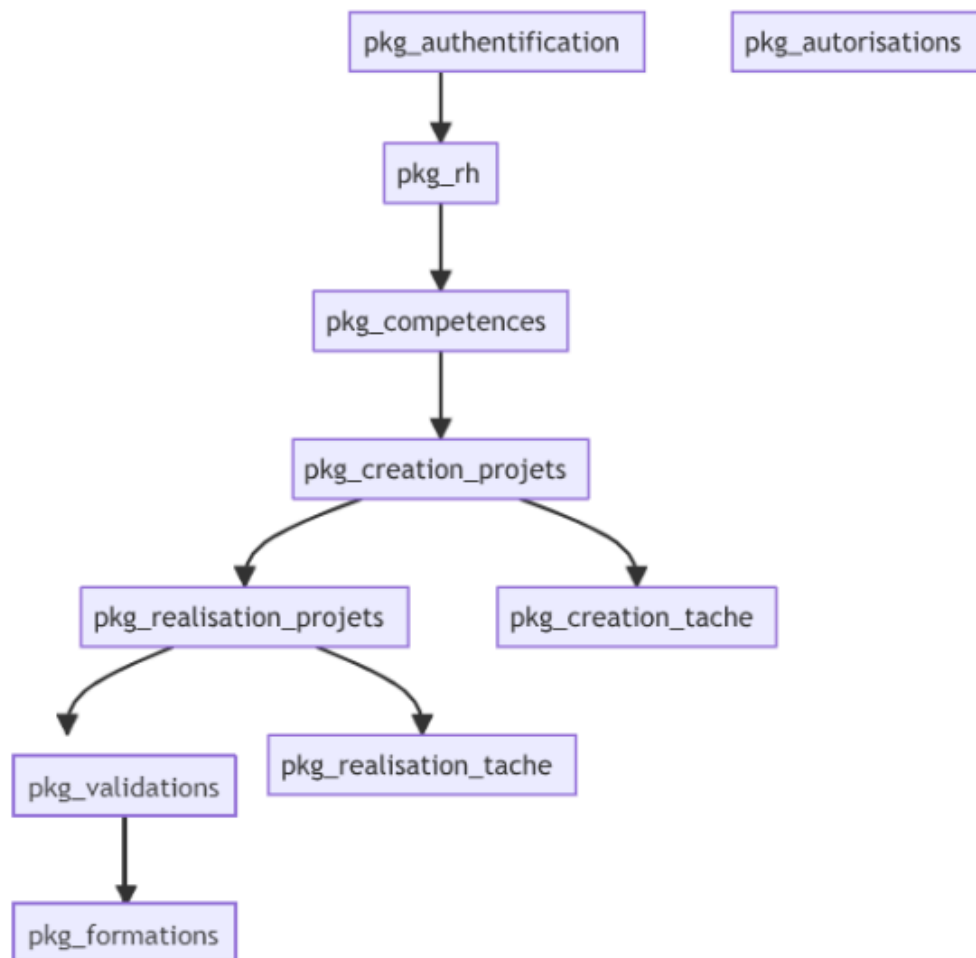


Figure 13 : Diagramme packages - *pkg_global*

Maquette

Maquette partie formateur qui permet la validation les réalisation des apprenants :

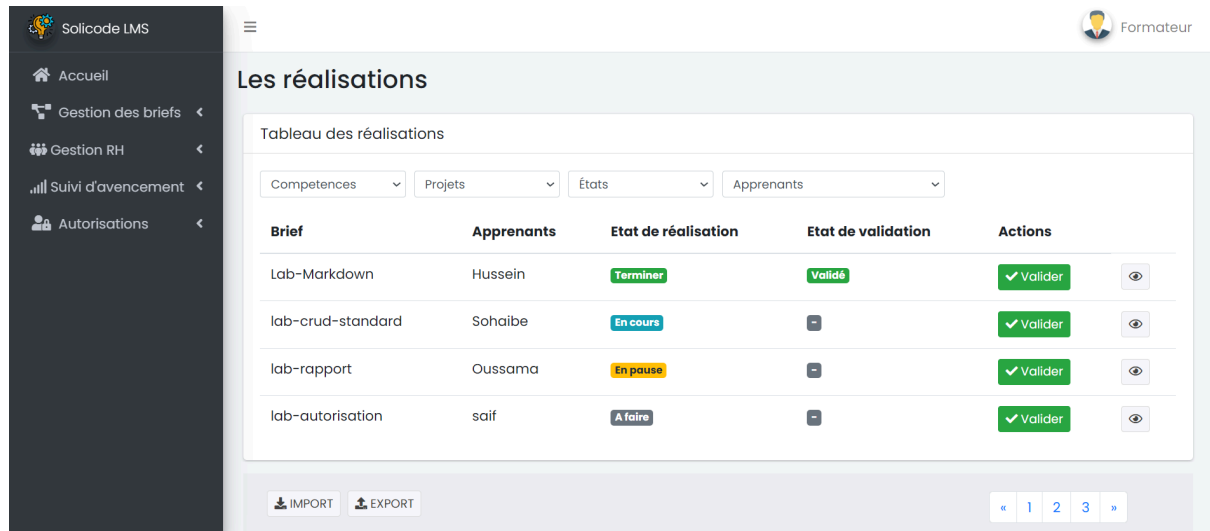


Figure 14 :Maquette - pkg Réalisations-projets

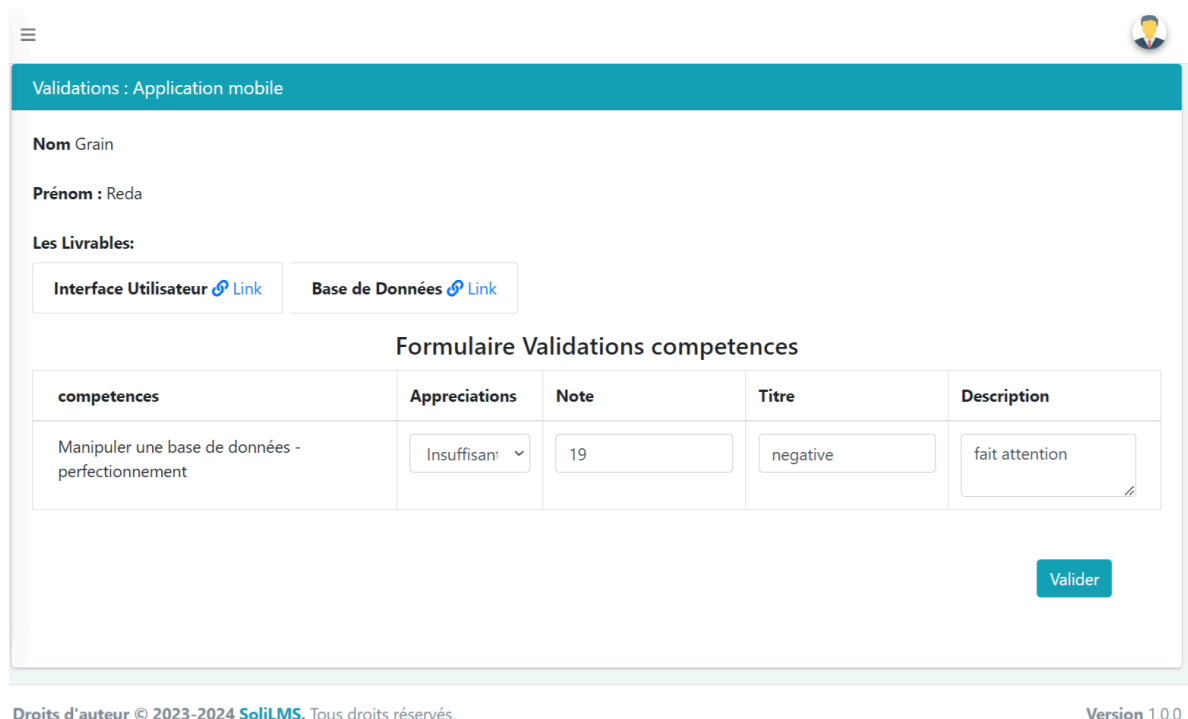


Figure 15 :Maquette - pkg validations

NB : Pour voir les autres maquettes, veuillez vous référer à l'Annexe 2 .

Réalisation

Ce chapitre met en valeur les technologies et outils utilisés dans la réalisation du projet Soli-LMS. Il permet de présenter les différentes technologies mises en œuvre et de visualiser le résultat obtenu.

Technologies Utilisées

HTML

(Hypertext Markup Language) : Langage de balisage standard utilisé pour créer la structure et présenter le contenu des pages web.



CSS

(Cascading Style Sheets) : Langage de feuille de style utilisé pour décrire la présentation et la mise en forme des documents HTML.



AdminLTE

est un modèle d'interface utilisateur open-source pour les applications web, reconnu pour son design moderne et ses fonctionnalités avancées, facilitant la création de tableaux de bord et de panneaux d'administration intuitifs et réactifs.



JavaScript

C'est un langage de programmation interprété de haut niveau principalement utilisé pour ajouter de l'interactivité et un comportement dynamique aux sites web.



MySQL

C'est un système de gestion de base de données relationnelle (SGBDR) open-source largement utilisé pour stocker et gérer des données dans diverses applications web.



Laravel

est un framework web PHP open-source reconnu pour sa simplicité, son efficacité et sa richesse en fonctionnalités, facilitant le développement rapide et la création d'applications web modernes.



UML

Le langage de modélisation unifié (UML) est un langage de modélisation standardisé utilisé pour représenter visuellement les systèmes logiciels. Il fournit un ensemble de diagrammes et de symboles pour illustrer la structure, le comportement et les relations des différents composants d'une application logicielle.



jQuery

C'est une bibliothèque JavaScript rapide, légère et riche en fonctionnalités. Elle simplifie la traversée des documents HTML, la gestion des événements et l'animation, rendant ainsi plus facile la création de pages web interactives et dynamiques .



Outils utilisés

GitHub

est une plateforme web qui offre un hébergement pour les dépôts Git. Elle propose des fonctionnalités comme le contrôle de version, les outils de collaboration, le suivi des problèmes et la gestion de projet, ce qui la rend populaire pour les projets de développement logiciel open-source et privés.



Git:

est un système de contrôle de version distribué qui permet à plusieurs développeurs de collaborer efficacement sur un projet. Il suit les modifications des fichiers, facilite les branches et les fusions, et fournit un historique de toutes les modifications.



Figma

est un outil de conception collaboratif utilisé pour créer des interfaces utilisateur et des prototypes. Il permet la collaboration en temps réel, le partage de designs et la collecte de retours.



VSC

Visual Studio Code est un éditeur de code source léger avec des fonctionnalités puissantes et des options de personnalisation étendues. Il est largement utilisé par les développeurs pour écrire et éditer du code dans différents langages de programmation.



Jekyll

est un générateur de site statique open-source. Il convertit des fichiers texte simples en sites web statiques et blogs, offrant une solution rapide et sécurisée pour créer des sites sans la complexité des CMS dynamiques.



Mermaid :

est un langage de balisage simplifié pour créer des diagrammes et des graphiques, offrant une façon intuitive de représenter des concepts visuellement dans les rapports et les documents.



PlantUML

est un langage de modélisation textuel permettant de générer des diagrammes UML tels que les diagrammes de classes, de cas d'utilisation et de cartes d'empathie, offrant une approche efficace pour la représentation graphique de structures et de processus.

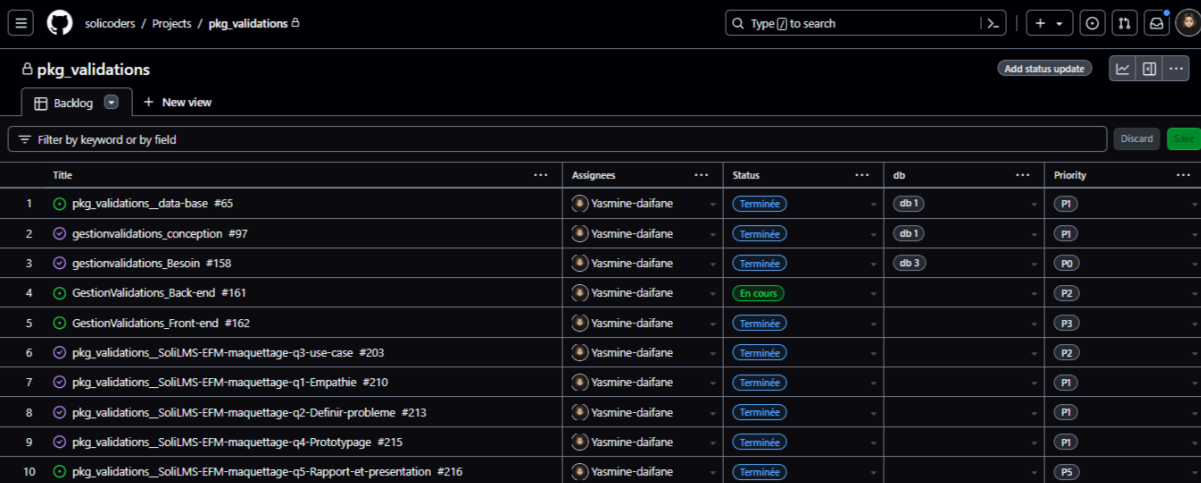


Conclusion

En conclusion, le projet Soli-LMS représente une avancée significative dans le domaine de la gestion de formation en offrant une plateforme complète et polyvalente pour répondre aux besoins des organismes de formation et des établissements d'enseignement. Réalisé en groupe, chaque membre de l'équipe a contribué avec son expertise à développer des packages spécifiques, tels que le package de Validation, qui sont essentiels pour assurer le bon fonctionnement et la performance de Soli-LMS. Doté d'un ensemble complet de fonctionnalités, Soli-LMS facilite la création, la gestion et l'évaluation des projets de formation, tout en offrant une expérience utilisateur conviviale et adaptable à divers appareils. L'intégration de technologies modernes telles que Laravel, MySQL et AdminLTE renforce la robustesse et la performance de la plateforme, assurant ainsi une gestion efficace des ressources humaines, des projets et des compétences des apprenants. Dans l'ensemble, Soli-LMS représente une solution complète pour la gestion de la formation et la progression des apprenants.

Le package de Validation au sein du projet Soli-LMS constitue une composante essentielle pour garantir l'équité, la précision et la cohérence dans l'évaluation des projets des apprenants. Grâce au diagramme de classe et aux maquettes d'interface utilisateur présentés, ce package offre une visualisation claire et structurée des processus de validation, facilitant ainsi la compréhension et la gestion des critères d'évaluation. Réalisé par un membre spécifique de l'équipe, le package de Validation contribue de manière significative à l'efficacité et à la qualité de l'évaluation des projets au sein de Soli-LMS. La combinaison de niveaux de validation personnalisables et d'un système de commentaires détaillés permet aux formateurs d'évaluer avec précision les compétences des apprenants et de fournir des retours constructifs pour favoriser leur progression. En résumé, le package de Validation joue un rôle crucial dans le succès et la performance globale de Soli-LMS.

Annexe 1 : Planification

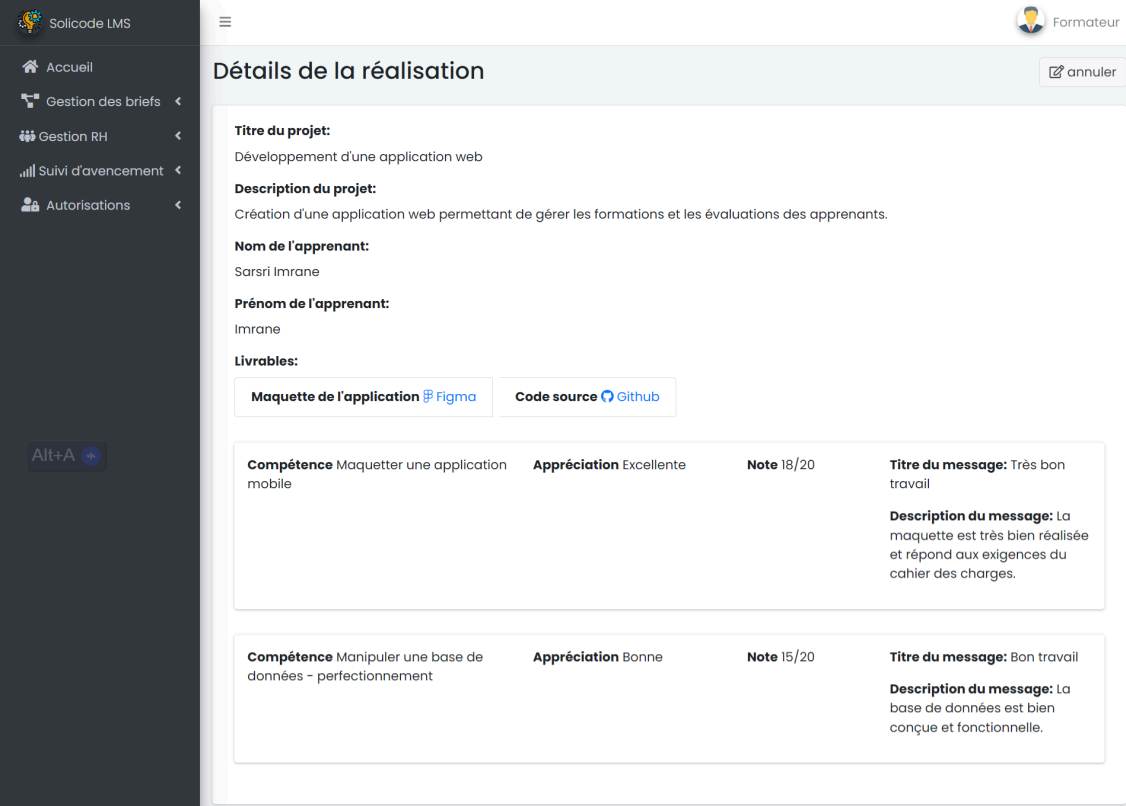


Title	Assignees	Status	db	Priority
1 pkg_validations_data-base #65	Yasmine-daifane	Terminée	db 1	P1
2 gestionvalidations_conception #97	Yasmine-daifane	Terminée	db 1	P1
3 gestionvalidations_Besoin #158	Yasmine-daifane	Terminée	db 3	P0
4 GestionValidations_Back-end #161	Yasmine-daifane	En cours		P2
5 GestionValidations_Front-end #162	Yasmine-daifane	Terminée		P3
6 pkg_validations_SolilMS-ERM-maquette-q3-use-case #203	Yasmine-daifane	Terminée		P2
7 pkg_validations_SolilMS-ERM-maquette-q1-Empathie #210	Yasmine-daifane	Terminée		P1
8 pkg_validations_SolilMS-ERM-maquette-q2-Definir-probleme #213	Yasmine-daifane	Terminée		P1
9 pkg_validations_SolilMS-ERM-maquette-q4-Prototypage #215	Yasmine-daifane	Terminée		P1
10 pkg_validations_SolilMS-ERM-maquette-q5-Rapport-et-presentation #216	Yasmine-daifane	Terminée		P5

Figure 16 : Backlog pkg_validations

Annexe 2 : Maquettes

Maquette partie formateur qui permet de visualiser en détail les informations de la validation :



Solicode LMS | **Formateur**

Détails de la réalisation [annuler]

Titre du projet:
Développement d'une application web

Description du projet:
Création d'une application web permettant de gérer les formations et les évaluations des apprenants.

Nom de l'apprenant:
Sarsri Imrane

Prénom de l'apprenant:
Imrane

Livrables:

- Maquette de l'application [Figma](#)
- Code source [Github](#)

Compétence	Appréciation	Note	Titre du message
Maquetter une application mobile	Excellente	18/20	Très bon travail
Manipuler une base de données - perfectionnement	Bonne	15/20	Bon travail

Description du message: La maquette est très bien réalisée et répond aux exigences du cahier des charges.

Description du message: La base de données est bien conçue et fonctionnelle.

Figure 17 : Maquette - pkg_validations

