

PFR : Memory (CDC)

Agnes Courde - Needemand - 2024

Partie 1 :

Client.....	2
Cahier des charges :.....	3
Contexte et Objectifs.....	3
Fonctionnalités Clés du Logiciel de l'application web.....	3
Fonctionnalité avancées.....	5
Objectifs de l'Entreprise Prestataire.....	6
Bénéfices Attendus.....	6
Contraintes technologiques.....	7
Application Web.....	7
Application Annexe en python.....	7
Dossier de conception.....	8
Livrables.....	8
Instructions.....	9

Client

Memory est une start up qui fournit une plateforme basée sur l'IA. L'entreprise Memory, est spécialisée dans le monde de l'événementiel. Leur plateforme permet la gestion complète d'un bar à boissons, de la création d'un menu à la gestion des stocks et des salariés intérimaires et saisonniers. L'entreprise a une dizaine d'années d'ancienneté.

Memory souhaite améliorer la collaboration entre **ses développeurs**. Elle décide de demander à une entreprise spécialisée de créer un logiciel interne pour gérer les notes et les feedbacks entre les différents membres de son équipe de développeurs.

L'entreprise Memory **vous contacte** pour vous demander d'ajouter une nouvelle fonctionnalité dans leur écosystème.

Cahier des charges :

Contexte et Objectifs

Cette startup est composée d'une équipe de développeurs répartis sur plusieurs projets. L'objectif principal est de faciliter le partage d'informations, d'idées, de problèmes techniques, et de suggestions de manière centralisée et structurée. Le logiciel doit permettre à chaque développeur de laisser des notes pour les autres membres de l'équipe, que ce soit pour signaler un problème, partager une astuce, ou demander des conseils.

Fonctionnalités Clés du Logiciel de l'application web

1. Création et Partage de Notes :

Les développeurs peuvent créer des notes sous forme de texte, images, ou même de code. Les notes peuvent être partagées avec un ou plusieurs développeurs spécifiques ou avec toute l'équipe.

2. Classification et Étiquetage :

Les notes peuvent être classées par projet, langage de programmation, ou type de problème. Des étiquettes peuvent être ajoutées pour faciliter la recherche ultérieure (ex : "bug", "optimisation", "documentation").

3. Recherche Avancée :

Une fonction de recherche avancée permet de retrouver rapidement des notes spécifiques en fonction des mots-clés, des projets, ou des contributeurs.

4. Commentaires et Discussions :

Chaque note peut être commentée par d'autres membres de l'équipe pour lancer des discussions ou proposer des solutions. Un système de fil de discussion permet de garder les échanges organisés.

5. Profil des contributeurs

Chaque contributeur peut détailler son profil en incluant ses compétences techniques. Ses compétences permettront de mieux attribuer les développeurs à chaque projet ou résolution de bug.

6. Tableau de Bord Personnalisé :

Chaque développeur dispose d'un tableau de bord personnalisé où il peut voir entre autre :

- Les notes les plus récentes
- Les tâches qui lui sont assignées.
- Les projets sur lesquels il travaille.
- La liste des contributeurs qui ont au moins un projet en commun
- Les discussions en cours

Fonctionnalité avancées

Notifications :

Les développeurs reçoivent des notifications lorsque de nouvelles notes les concernant ou lorsque des commentaires sont ajoutés à une note qu'ils suivent.

Métriques de participation :

Définissez des indicateurs pertinents pour mesurer la participation (nombre de notes, qualité des contributions, résolution de problèmes, etc.). Réfléchissez à la façon dont ces métriques seront calculées et mises à jour.

Un administrateur peut générer des rapports de participation individuels et par projet.

Confidentialité et Accès aux Données :

Déterminez qui aura accès aux profils détaillés et aux rapports de participation. Proposez des niveaux d'accès différents selon les rôles dans l'équipe. Alternant, Dev Senior, Chef de projet ...

Prenez en compte les aspects de sécurité, notamment pour la gestion des utilisateurs et le partage des notes.

Objectifs de l'Entreprise Prestataire

L'entreprise à qui la startup fait appel pour développer ce logiciel devra :

- Proposer une interface utilisateur **intuitive et ergonomique**.
- Proposer une application complémentaire de type desktop
- Assurer la sécurité et la confidentialité des informations partagées.
- Prendre en compte la synchronisation des données entre les différents composants.

Bénéfices Attendus

- **Amélioration de la Collaboration** : Le logiciel permettra aux développeurs de mieux partager leurs connaissances et de résoudre les problèmes plus rapidement.
- **Centralisation de l'Information** : Toutes les notes et discussions seront centralisées, évitant ainsi la dispersion de l'information sur différents canaux.
- **Gain de Temps** : La recherche facilitée et la classification des notes permettront aux développeurs de retrouver rapidement les informations dont ils ont besoin.

Ce logiciel interne aidera la startup à fluidifier les échanges entre ses développeurs, à mieux capitaliser sur les connaissances partagées, et à renforcer la cohésion au sein de l'équipe.

Contraintes technologiques

Application Web

- Le frontend doit être conçu principalement en JavaScript, avec la possibilité d'utiliser React ou vueJS.
- Vos serveurs étant en PHP, le backend doit être conçu en PHP..
- La base de données est déployée sur un SGBD MySQL.

Application Annexe en python

Dans un second temps, il vous sera demandé de réaliser une application en python, qui fera appel à la même API que l'application précédente.

Application desktop complémentaire en Python (choisir au moins une option) :

- Gestion de notes : créations rapide de prise de notes, avec une synchronisation manuelle.
- Messagerie avec une synchronisation automatique des discussions présentes sur le site web.
- Répertoire des développeurs, avec gestion des compétences, application destiné au responsable des ressources humaines.

Cette application devra forcément se synchroniser avec l'application web.

Dossier de conception

Réaliser un dossier de conception complet pour répondre à l'appel d'offres de Memory (gestion de notes). Un dossier de conception est un ensemble de documents qui décrit les particularités du projet et aide à la planification et la réalisation du projet.

Dans votre analyse du cahier des charges, réfléchissez à l'ensemble des acteurs de l'application et aux différents rôles.

Livrables

Votre dossier de conception devra inclure les éléments suivants :

1. **Conception graphique :**
 - a. **Maquette** : Créez une maquette visuelle de l'interface utilisateur de votre application, en tenant compte des fonctionnalités clés.
 - b. **User Stories** : Rédigez au moins 5 user stories décrivant les principales fonctionnalités de votre système du point de vue de l'utilisateur. (Storyboard avec des commentaires)
2. **Conception du modèle**
 - a. **Modèle Conceptuel de Données (MCD)** : Élaborez un MCD représentant la structure logique de vos données.
 - b. **Modèle Logique de Données (MLD)** : Transformez votre MCD en MLD.
 - c. **Modèle Physique de Données (MPD)** : Créez un MPD basé sur votre MLD, adapté à MySQL
3. **Conception fonctionnelle :**
 - a. **Diagramme de classe** : Élaborez un diagramme de classe UML représentant la structure statique de votre système, en vous concentrant sur les entités principales (ex: Utilisateur, Note, Projet, Commentaire)..
 - b. **Diagramme de séquence** : Créez au moins un diagramme de séquence UML pour un scénario clé de votre application.
4. **Conception d'architecture**
 - a. **Diagramme de composants** : Développez un diagramme de composants UML montrant l'organisation et les dépendances des composants logiciels majeurs. Montrez l'organisation des différentes parties du système (frontend, backend, base de données).
 - b. **Diagramme de dépendance** : Représentez les relations entre les différentes technologies, bibliothèques et composants logiciels de votre projet.

- c. **Diagramme de déploiement** : Concevez un diagramme de déploiement UML illustrant l'architecture physique de votre système.

Instructions

Assurez-vous de la **cohérence** entre tous les diagrammes et modèles.

- Le nom des classes, méthodes composants ... doivent être les mêmes dans tous les diagrammes.
- Les fonctionnalités décrites dans les user stories doivent se retrouver dans les diagrammes UML. Par exemple, une user story parlant de "création de compte" devrait avoir une correspondance dans le diagramme de séquence et les méthodes du diagramme de classe.
- Le diagramme de déploiement et le diagramme de composants doivent refléter la même architecture système. Les composants majeurs doivent se retrouver dans les deux diagrammes de manière cohérente.
- Les éléments d'interface utilisateur présents dans la maquette doivent pouvoir être expliqués par les fonctionnalités modélisées dans les diagrammes UML.
- Les entités présentes dans le MCD doivent correspondre aux classes principales du diagramme de classe. Le MLD et le MPD doivent être des traductions fidèles du MCD.

Accompagnez chaque diagramme d'une brève explication (1-2 paragraphes) **justifiant vos choix** de conception.