***Cahier des charges***

1. ***Introduction :***

Développement d’une application qui effectuera la validation du temps travaillé par des employés d’une entreprise dans le respect des règles donnée par l’entreprise. Chaque employer doit remplir une feuille de temps hebdomadaire.

1. ***Répartition des tâches à chacun des membres de l'équipe.***

|  |  |
| --- | --- |
| ***Le membre*** | ***Sa Tache*** |
| Belil Zakaria | * Responsable de projet « product owner ». * Répartition des taches pour les membres d’équipe. * Vérification d’évolution de travaille de chaque membre à chaque Daily Scrum Meeting. * Création du projet sur visuel studio * Vérification de diagramme de classe réaliser par l’équipe * Programmation |
| Chabane Rima | * « Scrum Master » aide les autres membres d’équipe dans la résolution des problèmes durant le travail sur le projet. * Création des classe (journeeDeTravaille, les classe dériver de chaque employer, semaine de travail) * Programmation |
| Alimiraoussaid Samira | * Création des fichier jSon * Faire le cahier des charges * Programmation |
| Lamouchi Mohamed Melek | * Programmation * Creation de la classe validateurFeuilledeTemps |

1. ***Technologies choisies pour le projet :***

Pour le projet on a choisi d’utiliser le langage de programmation C#.

1. ***Échéancier du sprint :***

***Le 10-04-2023(***durée de la rencontre 1h :30***)***

Première rencontre de l’équipe : explication de projet et répartition des taches pour chaque membre de l’équipe.

***Le 11-04-2023(***durée de la rencontre 3h***)***

Établissement des Exigences Fonctionnelles et non fonctionnelles, liste des diagramme UML

***Le 12-04-2023(***durée de la rencontre 2h***)***

Vérification d’état d’avancement des taches reparti précédemment.

***Le 14-04-2023(***durée de la rencontre 1h***)***

Vérification d’état d’avancement des taches reparti précédemment. Et répartition des nouvelles taches.

***Le 15-04-2023(***durée de la rencontre 2h***)***

Vérification d’état d’avancement des taches reparti précédemment. Plus un travail de groupe sur le code.

***Le 17-04-2023(***durée de la rencontre 2h***)***

Rencontre pour vérification et résolution des problèmes rencontrée par certain membre de groupe sur l’utilisation git et github.

1. ***Exigences Fonctionnelles :***

* L’application doit permettre à l'utilisateur de saisir un fichier JSON représentant la feuille de temps d'un employé.
* L’application doit valider la feuille de temps en vérifiant qu'elle respecte les règles énoncées pour chaque employé, telles que le nombre d'heures travaillées sur chaque projet, les heures de travail régulières et les heures supplémentaires.
* Si la feuille de temps ne respecte pas les règles, l'application doit afficher les erreurs spécifiques à l'utilisateur.
* L’application doit générer un fichier JSON de sortie contenant les erreurs détectées (s'il y en a) pour une utilisation ultérieure ou une présentation à d'autres parties prenantes.
* L’application doit être conviviale et facile à utiliser, avec des instructions claires pour guider l'utilisateur tout au long du processus de validation.

1. ***Exigences non fonctionnelles :***
   1. Performance : L'application doit être capable de valider les feuilles de temps des employés en un temps raisonnable, même pour de grandes quantités de données.
   2. Sécurité : Les données des employés doivent être protégées contre les accès non autorisés.
   3. Fiabilité : L'application doit être robuste et capable de traiter les erreurs de manière efficace pour minimiser les temps d'arrêt et les pertes de données.
   4. Convivialité : L'application doit être facile à utiliser pour les utilisateurs, avec une interface utilisateur intuitive et conviviale.

Dispose d’un menu contextuel

1. ***Liste des diagrammes UML :***
   * + - ***Le diagramme de cas d’utilisation***

***Une image contenant diagramme

Description générée automatiquement***

* ***Le Diagramme de classe***

Une image contenant graphique, diagramme

Description générée automatiquement