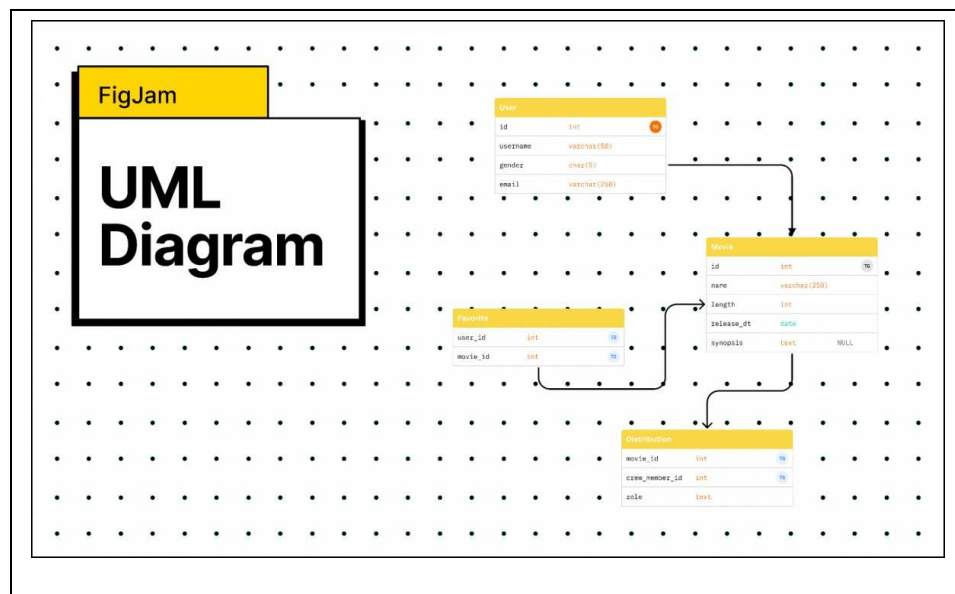




Projet:

Modelisation et Conception oriente objet



Presenter par:

-Jatsa harrisson Nerisse

-Diallo Souleymane

Note Importante

Comme discuté lors de la présentations nous avons apporté quelques modifications au code sources en nettoyant le code superflu. Nous n'avions pas montré certaines fonctionnalités avec le menu de l'application mais comme on l'avait mentionné des methodes étaient implémentées pour ce travail.

Ces méthodes se retrouvent dans la classe MenuUtils qui se trouve dans le package utils ainsi que la classe TimeLogSystem qui se trouve dans le package backend.

Bien entendu il y'a d'autres fonctionnalites dans les autres classes tels que les tests réussis mais les changements majeurs se trouvent dans les classes cités ci-dessus.

Sommaire

I. Introduction :

Le projet "TimeLog" représente un travail en équipe visant à appliquer les principes de l'analyse et de la conception orientées objet, avec un accent particulier sur l'approche unifiée et UML. Le développement du logiciel se fera en Java, utilisant l'éditeur de code VS Code, et offre une occasion d'expérimenter avec des outils logiciels performants tels que Git et GitHub, couramment utilisés dans le développement professionnel. La collaboration entre les membres de chaque équipe est essentielle pour garantir le respect des délais impartis. Les étudiants seront amenés à apprendre à travailler en équipe, à se répartir équitablement les activités, à développer leur sens de la communication et à synchroniser leur travail pour être productifs dans leurs efforts. Un chef d'équipe sera désigné pour coordonner le travail et veiller à la progression harmonieuse du projet, ce rôle étant tournant et préalablement déterminé pour chaque membre de l'équipe.

II. Objectif du projet TimeLog

Le système TimeLog offrira divers services via une interface en ligne de commande, permettant à l'utilisateur de choisir des éléments de menu et de fournir des valeurs à partir de listes d'options. Parmi les services proposés figurent la génération de rapports d'état pour chaque projet et l'ensemble des projets, les rapports de salaire pour chaque employé sur une période spécifiée, la production de talons de paie, la modification des paramètres et certaines données par l'administrateur, la gestion des disciplines de travail et l'enregistrement des informations dans des fichiers texte au format Json.

En outre, le projet TimeLog devra être capable de s'interfacer avec un futur sous-système de production et d'impression de chèques de paie aux deux semaines. Pour ce faire, il devra envoyer une liste d'objets de type PayInfo au sous-système nommé Payroll. Cet objet PayInfo contiendra les informations nécessaires pour une période de paie, notamment l'identifiant de l'employé, le nombre d'heures de base, le nombre d'heures supplémentaires, les taux horaires de base et supplémentaires, ainsi que les dates de la période de paie considérée.

Le logiciel TimeLog, conçu pour une compagnie de développement de logiciels X, permettra une assistance automatisée pour le calcul du temps, des salaires et le contrôle budgétaire des projets. Chaque employé pourra se connecter sur une machine dédiée et signaler le début et la fin de son travail sur une activité en spécifiant le projet et la discipline de travail. Le système enregistrera l'heure et la date de ces signalements. Le logiciel TimeLog adopte différentes disciplines d'activités,

telles que le design1 (haut niveau), le design2 (détaillé), l'implémentation, le test et le déploiement, inspirées du processus unifié de développement de logiciels vu en classe. Un projet sera caractérisé par sa date de début, date de fin, le nombre d'heures budgétées pour chaque discipline, ainsi qu'un nom et un numéro d'identification. De même, chaque employé sera caractérisé par son historique de taux horaires de base et de taux pour le temps supplémentaire, son numéro d'identification (ID), son nom, sa date d'embauche, son éventuelle date de départ, son numéro d'assurance sociale et son poste.

La mise en œuvre du projet TimeLog doit comprendre trois projets (projet1, projet2, projet3) avec leur budget pour toutes les disciplines, ainsi que trois employés (employé1, employé2, employé3) ayant chacun un nom d'utilisateur et un ID, et un compte administrateur (admin) avec les privilèges appropriés. Chaque employé sera assigné à deux projets parmi les trois, avec des durées de travail spécifiées pour chaque jour, sur une période de deux semaines. L'initialisation avec des exemples simples est demandée pour faciliter les tests et assurer l'uniformité des projets des étudiants.

III. Outils logiciels utilisés (Git, GitHub, VS Code)

Le recours à l'outil de gestion de version Git est fortement conseillé pour gérer efficacement le développement du code source et permettre un retour rapide à une version fonctionnelle en cas de problèmes. Pour faciliter le développement parallèle et l'intégration automatique des différentes portions de code, l'utilisation de l'outil de gestion de contrôle GitHub est requise, sachant que GitHub est la partie serveur de Git et permet le travail collaboratif sur le cloud. Chaque équipe devra ouvrir un compte gratuit sur GitHub et fournir un accès en lecture dans le rapport faisant partie des livrables du projet. La correction du projet prendra en compte les traces de collaboration telles que les "commit", les "issues", les "Pull requests" et les "tags" sur le repos sur GitHub, ainsi que l'historique de production du code, qui ne devra pas être réalisé hâtivement la veille de la soumission, mais montrer les débuts significatifs de la production à partir du 08 juillet.

IV. Travail en équipe et collaboration entre les membres

Ce projet a été réalisé par Harrisson Nerisse et Diallo Souleymane. Une contribution significative a été un élément essentiel pour mener à bien ce projet ambitieux.

Diallo Souleymane a joué un rôle essentiel dans le développement de la composante technique du projet. Grâce à ses compétences exceptionnelles en programmation, il a été en mesure de concevoir et d'implémenter des solutions innovantes qui ont grandement amélioré la fonctionnalité globale de notre application. Sa persévérance et son dévouement à résoudre des problèmes complexes ont été une source d'inspiration pour l'équipe. Diallo a contribué à la réalisation du code, En fin de compte, son engagement total dans le projet a eu un impact significatif sur le succès de notre équipe.

Harrison Nerisse, de son côté, a apporté une perspective unique au projet grâce à ses compétences en rédaction et à sa créativité dans la conception de diagrammes et de modèles conceptuels. Son implication dans la rédaction de la documentation technique et la création de diagrammes UML ont grandement amélioré la clarté et la compréhension du projet. En plus de ses talents en rédaction, Harrison a également contribué en tant qu'assistant dans le développement du code, montrant sa polyvalence et son désir de contribuer dans divers domaines. Son attitude proactive et sa capacité à s'adapter aux besoins changeants du projet ont été un atout précieux pour l'équipe.

Le pourcentage de travail de chaque étudiant peut être estimé comme suit :

Diallo Souleymane :

Travail technique et programmation : 80%

Conception de diagrammes et de modèles : 30%

Participation aux réunions et discussions : 100%

Tests et débogage : 50%

Harrison Nerisse :

Rédaction de la documentation : 80%

Conception de diagrammes et de modèles : 70%

Contribution au développement du code : 20%

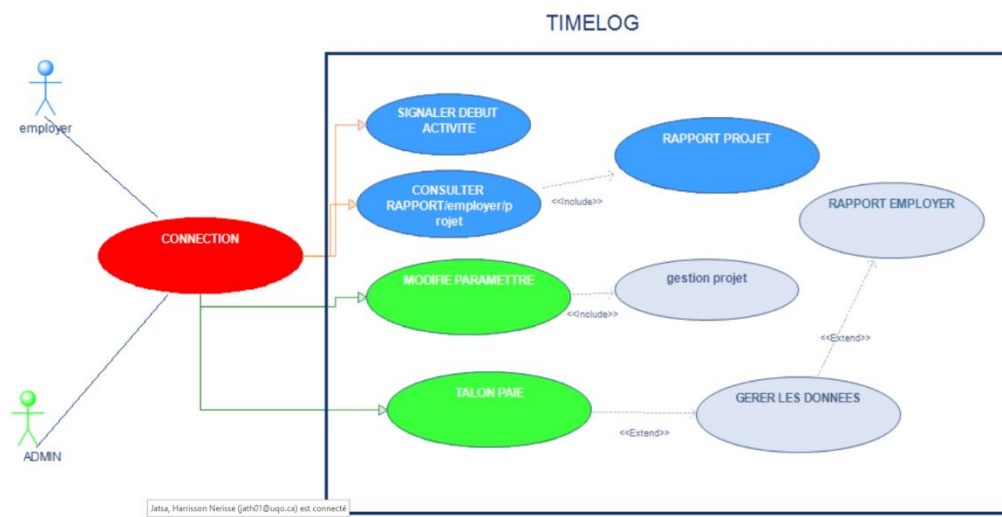
Participation aux réunions et discussions : 100%

Tests et révisions : 50%

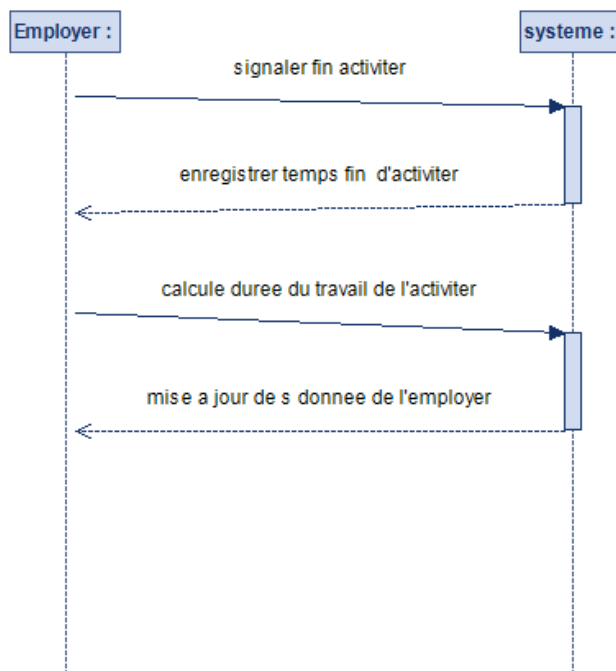
Lien Github: <https://github.com/soul-diallo/TimeLog-Github-Diallo-Jatsa>

V. Modèle du domaine

SCHEMA DE CAS D'UTILISATION:



1. DSS du cas d'utilisation "Signaler fin d'activité"



2. Diagramme d'interaction d'une opération du DSS

Dans ce diagram, nous avons trois objets principaux impliqués : l'employé, le projet et le système de suivi du temps et de gestion de projet. L'interaction se déroule en plusieurs étapes :

L'employé sélectionne l'option pour générer un rapport d'état de projet.

Le système demande à l'employé de choisir le projet pour lequel il souhaite générer le rapport.

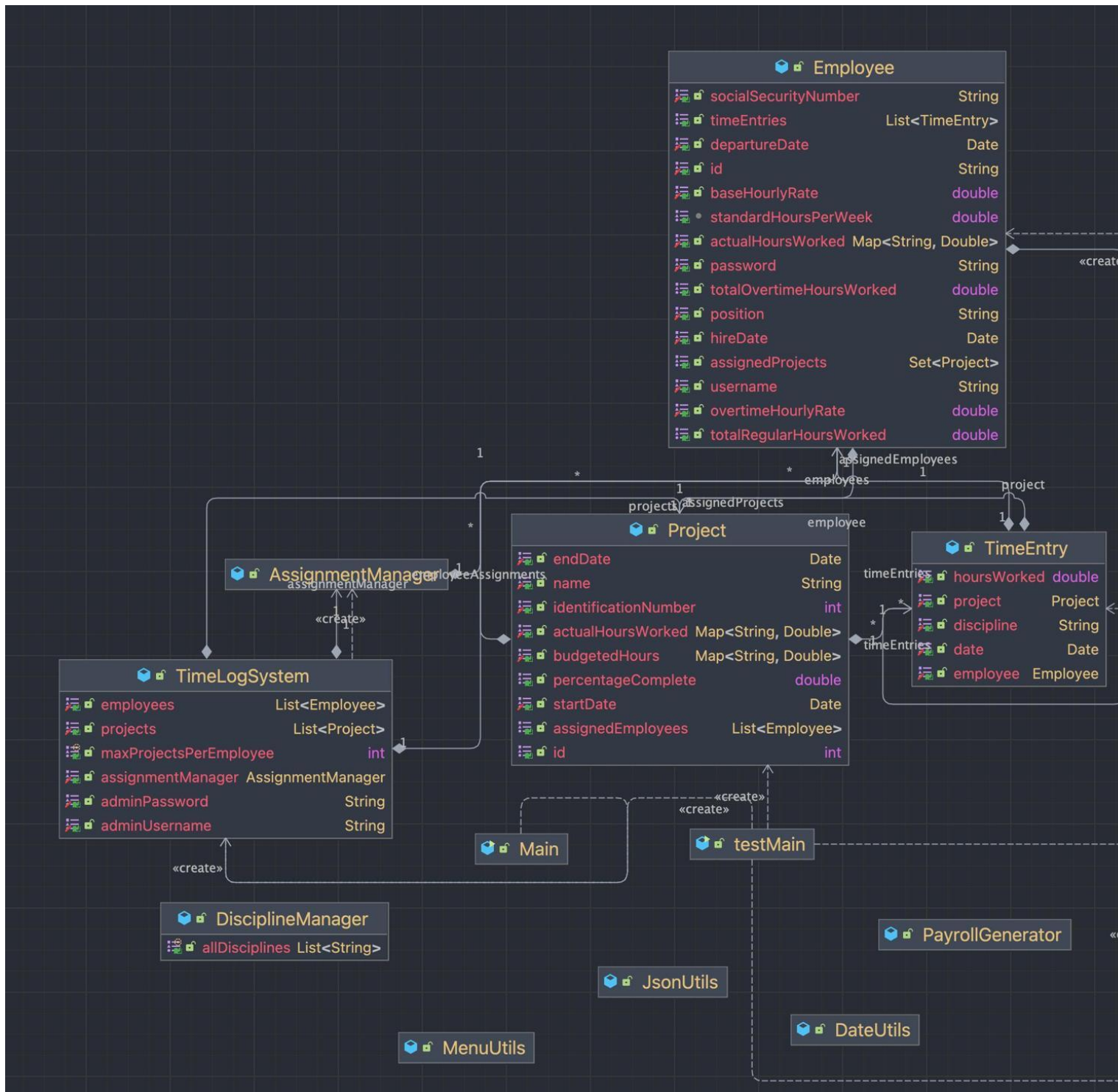
L'employé choisit le projet (project1 dans cet exemple).

Le système accède aux détails du projet (tel que le nom, les heures travaillées, les heures budgétées) en utilisant l'objet Project.

Le système génère le rapport d'état du projet en utilisant la méthode generateProjectStatusReport() de l'objet Project.

Le rapport est affiché à l'employé, montrant les informations sur le projet et les heures de travail.

3. Diagramme de classes



Lorsqu'on démarre le projet, on tombe sur l'interface principale, qui nous demande de nous identifier. En fonction du type d'authentification, nous aurons un menu différent .

Si on est administrateur on peut gérer les employé, c'est-à-dire, créer, modifier , supprimer, on peut aussi gérer les projets, et générer les rapport.

```
=== Menu Principal ===  
1. Rapport d'état d'un projet  
2. Rapport d'état global  
3. Rapport de salaire d'un employé  
0. Quitter  
Veuillez entrer votre choix : |
```

```
=== Menu Administrateur ===  
1. Gérer les projets  
2. Gérer les employés  
0. Retour au Menu Principal  
Veuillez entrer votre choix : 1
```

Si l'on est utilisateur, on peut ajouter une activiter c'est-à-dire , préciser l'heure de depart et arrive, et générer aussi des rapports.

Tous ces operations se font sous clavier, dans un terminal

VII. Code source adéquatement documenté

Notre code source va comprendre 11 classes, que nous allons détailler le fonctionnement :

1. La classe main :

Ce code représente un programme informatique qui aide à suivre le temps de travail des employés dans une entreprise. Imaginons que vous travaillez dans une entreprise où vous devez enregistrer quand vous commencez à travailler, quand vous arrêtez et d'autres informations importantes. Ce programme vous aide à faire cela de manière organisée.

Lorsque vous exécutez le programme, il vous demande ce que vous voulez faire en vous montrant des options. Vous pouvez choisir de vous connecter en tant qu'administrateur ou en tant qu'employé.

Si vous choisissez d'être administrateur, il vous demandera votre nom d'utilisateur et votre mot de passe. Une fois que vous êtes connecté, vous pouvez faire des choses comme gérer les projets de l'entreprise ou les informations sur les employés.

```
=== Menu Principal ===
1. Rapport d'état d'un projet
2. Rapport d'état global
3. Rapport de salaire d'un employé
0. Quitter
Veuillez entrer votre choix : 1
Nom d'utilisateur : admin
Mot de passe : admin
```

Si vous choisissez d'être un employé, il vous demandera également votre nom d'utilisateur et votre mot de passe. Une fois que vous êtes connecté, vous pouvez signaler quand vous commencez à travailler et quand vous arrêtez, comme appuyer sur un bouton pour enregistrer ces moments. Vous pouvez également demander au programme de générer un rapport sur votre salaire. Si à un moment donné on ne peut plus utiliser le programme, vous pouvez choisir de le quitter, et toutes les informations seront enregistrées pour la prochaine fois que vous l'utiliserez.

2. La classe employer

Cette partie du code représente une sorte de "fiche d'information" pour un employé d'une entreprise. Imaginez que chaque employé possède une carte sur laquelle sont consignées des informations importantes le concernant.

Chaque carte (ou objet) "Employé" contient des informations telles que le nom d'utilisateur de l'employé, son identifiant unique, le taux horaire normal de paiement, le taux horaire pour les heures supplémentaires, la date d'embauche de l'employé, etc. La fiche est également utilisée pour effectuer des calculs.

Le formulaire peut également être utilisé pour effectuer des calculs. Par exemple, il peut calculer le montant de la rémunération d'un employé en fonction des heures travaillées et du taux horaire. Si le salarié a effectué des heures supplémentaires, la feuille de calcul peut également calculer le paiement de ces heures, en utilisant un taux horaire différent.

Le formulaire garde également la trace des heures effectivement travaillées par le salarié, en les associant à des dates spécifiques. Cela peut s'avérer utile pour tenir un registre précis des heures de travail.

En outre, le formulaire comporte des méthodes permettant d'obtenir des informations spécifiques, telles que le total des heures normales travaillées par l'employé et le total des heures supplémentaires travaillées.

Dans l'ensemble, cette partie du code permet d'organiser et de gérer les informations relatives à un employé, ainsi que d'effectuer des calculs liés à ses heures de travail et à son salaire. Il s'agit en quelque sorte d'une fiche d'information pour l'employé, qui peut également effectuer des calculs automatiques pour simplifier le suivi des heures travaillées et le paiement.

3. Discipline manager

Cette classe représente un responsable de discipline dans le contexte d'un projet ou d'une entreprise. Imaginez que vous travaillez sur différents aspects d'un projet, tels que la conception, la mise en œuvre, les tests, etc. Chacun de ces aspects est appelé "discipline".

Ce gestionnaire de disciplines dispose d'une liste prédéfinie de disciplines déjà définies par défaut. Ces disciplines sont stockées dans une liste, comme une liste de noms.

Lorsque vous utilisez ce gestionnaire, vous pouvez demander une liste de toutes les disciplines disponibles en appelant la méthode **getAllDisciplines()**. Cette méthode renvoie une copie de la liste des disciplines, ce qui signifie que vous obtiendrez la liste des noms de disciplines prédéfinis.

4. Assignment manager

Cette classe représente un gestionnaire d'affectation pour les employés au sein des projets. Imaginez que vous travaillez dans une entreprise où différents employés sont affectés à différents projets pour accomplir leur travail. Ce gestionnaire permet d'organiser ces affectations.

Le gestionnaire d'affectation dispose d'une liste de correspondances entre les employés et les projets. Chaque employé peut être affecté à plusieurs projets, et chaque employé a une liste de projets qui lui sont assignés.

Vous pouvez utiliser ce gestionnaire pour :

Affecter un employé à un projet spécifique en appelant la méthode `assignEmployeeToProject(employee, project)`. Si le salarié n'a pas encore été affecté à un projet, une nouvelle entrée est créée pour ce salarié dans la liste des affectations. Si le salarié a déjà été affecté à d'autres projets, il sera ajouté à la liste.

Pour obtenir la liste des projets auxquels un salarié est affecté, appelez la méthode `getEmployeeAssignments(employee)`. Vous obtiendrez ainsi la liste des projets dans lesquels l'employé est impliqué.

Toutefois, le nombre de projets auxquels un employé peut être affecté est limité. Si cette limite est atteinte, le gestionnaire affiche un message indiquant que l'employé ne peut pas être affecté à d'autres projets.

5. Json utils

Cette classe concerne la gestion de fichiers au format JSON. Imaginez que JSON soit un moyen de stocker des informations d'une manière structurée et lisible par les ordinateurs. Le code utilise des outils spéciaux pour travailler avec ces fichiers JSON.

Il y a quelques actions principales que ce code peut effectuer :

Sauvegarder des objets dans un fichier JSON :

La méthode **saveToJson** permet de prendre un objet (comme un employé ou un projet) et le sauvegarde dans un fichier JSON spécifié. Cela se fait en transformant l'objet en texte JSON et en l'écrivant dans le fichier.

Charger des objets à partir d'un fichier JSON :

La méthode **loadFromJson** fait l'inverse de la précédente. Elle lit un fichier JSON spécifié et en extrait les données pour créer un objet Java. Cela peut être utile pour récupérer des informations stockées dans des fichiers.

Convertir des objets en format JSON :

La méthode **toJson** permet de prendre un objet Java et de le convertir en format JSON sous forme de texte. Cela peut être utile pour envoyer des données à d'autres parties d'un programme ou pour stocker temporairement des données.

Écrire du texte JSON dans un fichier :

La méthode **writeJsonToFile** prend du texte JSON et l'écrit dans un fichier. Cela peut être utile si vous avez déjà les données au format JSON et que vous souhaitez simplement les stocker dans un fichier.

Dans l'ensemble, ce code simplifie le processus de conversion entre les objets Java et le format JSON, ce qui est utile lorsque vous voulez stocker ou échanger des données structurées avec d'autres parties de votre programme ou même d'autres programmes. C'est un peu comme avoir un traducteur qui vous permet de convertir les informations dans un format que les ordinateurs comprennent bien.

6. Menu utils

cette classe représente un ensemble d'outils qui aident les utilisateurs à interagir avec le système de suivi du temps de travail et de gestion de projet. Plutôt que de manipuler directement le code, les utilisateurs peuvent utiliser ces outils pour entrer des informations et obtenir des rapports spécifiques.

Les fonctions incluses dans MenuUtils fournissent différentes fonctionnalités :

Afficher des menus :

Les méthodes comme `displayUserTypeMenu()` et `displayAdminMenu()` affichent des menus à l'écran avec des options pour les utilisateurs. Ces menus donnent des choix comme générer des rapports, gérer les projets, les employés, etc.

Entrer des informations :

Les méthodes comme `promptForUsername()`, `promptForPassword()`, `promptForProjectName()`, etc. invitent les utilisateurs à entrer des informations spécifiques, comme leur nom d'utilisateur, leur mot de passe, le nom du projet, etc.

Afficher des rapports :

Les méthodes comme `displayProjectStatusReport()`, `displayGlobalProjectStatusReport()`, etc. génèrent et affichent des rapports sur l'état des projets, les heures travaillées, les salaires, etc. Ces rapports permettent aux utilisateurs de voir les informations sous une forme lisible.

Fournir des messages de confirmation ou d'erreur :

Les méthodes comme `displayAssignmentConfirmation()` affichent des messages pour confirmer une action (comme l'assignation d'un employé à un projet) ou afficher des messages d'erreur (comme lorsque le nombre maximum de projets est dépassé).

7. Payroll generator

Cette classe représente un générateur de rapports de paie et d'état de projet. Imaginez que vous avez besoin de produire des rapports pour les employés et les projets, et ce code vous aide à automatiser ce processus.

Génération de Rapport de Paie (`generatePayrollReport`):

Cette méthode prend en entrée un employé, une date de début et une date de fin. Elle génère un rapport de paie pour cet employé sur la période spécifiée.

Le rapport affiche le nom de l'employé, la période de temps couverte, et pour chaque discipline travaillée :

Le nombre d'heures travaillées.

Le salaire de base calculé en fonction des heures travaillées.

Le salaire supplémentaire (heures supplémentaires) - cette partie doit encore être implémentée.

Le rapport affiche également le total du salaire de base, du salaire supplémentaire, du salaire brut et du salaire net.

Génération de Rapport d'État de Projet (`generateProjectStatusReport`) :

Cette méthode prend en entrée un projet. Elle génère un rapport d'état pour ce projet.

Le rapport affiche le nom du projet, son identifiant, la date de début et la date de fin.

Pour chaque discipline travaillée dans le projet, le rapport affiche :

Le nom de la discipline.

Le nombre d'heures travaillées dans cette discipline.

Le nombre d'heures budgétées pour cette discipline.

Le pourcentage d'avancement, mais cette partie est actuellement en commentaire.

Le rapport donne un aperçu global de l'état du projet en termes d'heures travaillées et de budget.

8. Project

Cette classe appelée "Project", qui est utilisée pour stocker et gérer des informations sur un projet. Imaginez que vous ayez plusieurs projets et que vous ayez besoin d'un moyen organisé de stocker leurs détails.

Stockage des détails du projet :

La classe a plusieurs attributs tels que startDate (date de début du projet), endDate (date de fin du projet), budgetedHours (heures budgétées pour chaque discipline), name (nom du projet) et identificationNumber (numéro d'identification du projet).

Ces attributs sont utilisés pour stocker des informations spécifiques à chaque projet, comme quand il commence, quand il se termine, combien d'heures sont prévues pour chaque discipline, etc.

Méthodes pour accéder aux détails du projet :

Il y a des méthodes get et set pour chaque attribut, ce qui permet d'accéder et de modifier les détails du projet de manière contrôlée.

Calcul du pourcentage d'avancement :

La méthode getPercentageComplete() calcule et retourne le pourcentage d'avancement du projet en fonction des heures réellement travaillées par rapport aux heures budgétées.

Gestion des heures réellement travaillées :

Il y a des méthodes calculateTotalHoursForDiscipline() et budgetedHoursForDiscipline() pour calculer respectivement le total des heures travaillées pour une discipline donnée et les heures budgétées pour cette discipline.

Génération de Rapport d'État de Projet :

La méthode generateProjectStatusReport() génère et affiche un rapport d'état pour le projet. Pour chaque discipline, elle affiche les heures travaillées, les heures budgétées et le pourcentage d'avancement. Cela donne une vue globale de l'état du projet pour chaque discipline.

9. Test main

Une classe de test appelée "testMain". Il démontre comment utiliser la classe TimeLogSystem et d'autres composants pour créer des employés fictifs, des projets fictifs, générer des rapports et effectuer des opérations d'entrée/sortie de données.

Initialisation du système :

Une instance de TimeLogSystem est créée en utilisant le nom d'utilisateur et le mot de passe "admin".

Création d'employés fictifs :

Trois employés fictifs (employee1, employee2 et employee3) sont créés avec des détails tels que le nom d'utilisateur, l'ID, le taux horaire de base, le taux horaire des heures supplémentaires, etc. Ces employés sont ajoutés au système en utilisant la méthode addEmployee().

Création de projets fictifs :

Trois projets fictifs (project1, project2 et project3) sont créés avec des détails tels que la date de début, la date de fin, les heures budgétées pour chaque discipline, etc. Ces projets sont ajoutés au système en utilisant la méthode addProject().

Génération de Rapport d'État de Projet :

Un rapport d'état est généré pour project1 en utilisant la méthode generateProjectStatusReport().
Conversion et écriture en JSON :

Les employés du système sont convertis en format JSON à l'aide de la méthode toJson() de la classe JsonUtils, puis écrits dans un fichier appelé "employees.json".

Les projets du système sont également convertis en format JSON et écrits dans un fichier appelé "projects.json".

Affichage des informations des projets au format JSON :

Les informations JSON de chaque projet du système sont affichées à l'écran.

10. Time log system

Une classe appelée "TimeLogSystem", qui fonctionne comme un système de suivi du temps pour les employés et les projets. Voici ce que fait la classe en général :

Initialisation du système :

Lorsque vous créez une instance de cette classe, vous devez fournir un nom d'utilisateur et un mot de passe pour l'administrateur.

Gestion des employés et des projets :

Vous pouvez ajouter, supprimer et mettre à jour des employés et des projets dans le système.

La classe stocke les employés et les projets dans des listes.

Assignation d'employés à des projets :

Vous pouvez assigner un employé à un projet en utilisant l'objet AssignmentManager.

Authentification d'utilisateurs :

Vous pouvez vérifier si un nom d'utilisateur et un mot de passe correspondent à ceux de l'administrateur ou d'un employé.

Calcul des salaires :

Vous pouvez calculer le salaire d'un employé pour une période spécifique en fonction des entrées de temps enregistrées.

Le salaire est calculé en fonction des taux horaires de base et des taux horaires des heures supplémentaires.

Calcul des salaires totaux :

Vous pouvez calculer le salaire total en fonction des taux horaires et des heures travaillées pour tous les employés.

Gestion des projets pour les employés :

Vous pouvez obtenir une liste des projets assignés à un employé spécifique.

VIII. CONCLUSION

En conclusion, le projet TimeLog offre une opportunité d'apprentissage pratique pour nous les étudiants, en nous confrontant à des aspects essentiels du développement logiciel en équipe et en leur permettant d'appliquer les concepts d'analyse et de conception orientées objet, tout en utilisant des outils logiciels modernes et performants.