UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS AUTOMNE 2023
Dpt. Informatique et Ingénierie PROJET DE SESSION

PRÉSENTÉ À JAMAL ABD-ALI

COMME EXIGENCE PARTIELLE DU COURS
Modélisation et Conception orientée Objet (INF1163 - 20)

Système TIMELOG

Pour assistance automatisée de calcul de temps, de salaires et de contrôle de budget

Par

MARYAM MOUATADID (MOUM06519400)
KEN LESSARD-GERBER (LESK26269809)
DARIN JOREL FENGYEP NGANLEU (FEND65040008)

Table des Matières

Timelog

Le projet TimeLog a pour objectif d'appliquer les principes de l'analyse et de la conception orientées objet, en utilisant l'approche unifiée et UML, pour une implémentation en Java. Son objectif principal est de créer un logiciel permettant à une entreprise de développement de logiciels de gérer efficacement le temps, les salaires et le contrôle budgétaire de ses projets. TimeLog est un système installé sur une machine dédiée, accessible en ligne de commande. Le système a pour objectif de permettre aux employés d'une organisation de gérer efficacement et de suivre le temps passé sur différents projets. Il vise à fournir une solution automatisée pour l'enregistrement des activités des employés, la gestion des projets, le suivi du temps passé sur chaque activité, et la génération de rapports pertinents pour évaluer l'avancement des projets et la productivité des employés. Une brève spécification des besoins est résumée ci-dessous :

Besoin	Spécification
Gérer projet(s)	Le système doit permettre la création, la modification et la suppression de projets. Cette fonctionnalité est essentielle pour organiser les activités des employés en fonction des projets sur lesquels ils travaillent. Le système doit permettre l'ajout de nouvelles informations, la modification des détails existants et la possibilité d'écrire et lire des archives de projets au format Json.
Gérer employé(s)	Le système doit permettre la création, la modification et la suppression de comptes employé, avec des droits administrateurs. Le système doit permettre l'ajout de nouvelles informations, la modification des détails existants et la possibilité d'écrire et lire des archives de projets au format Json.
Enregistrer activité	Le système doit permettre aux employés d'enregistrer leurs activités, y compris le début et la fin d'une activité, ainsi que des détails tels que le projet associé et la discipline de travail.
Générer rapport	Le système doit permettre la génération de rapports pour évaluer l'avancement des projets et la productivité des employés - ainsi que des rapports récapitulatifs sur les salaires et déductions.

Conception et Modélisation

Dans le cadre du projet TimeLog, notre approche de conception a été guidée par les principes de l'analyse et de la conception orientées objet, en utilisant l'approche unifiée et UML. Nous avons produit plusieurs artefacts pour détailler la structure et le comportement du système, fournissant ainsi une base solide pour son développement et son utilisation.

Modèle du Domaine - ou modèle conceptuel, a pour but de représenter simplement les concepts importants et représentatifs du problème. Nous avons identifié les principales entités en utilisant telles que Projet, Administrateur, Employé, Activité, Discipline, Rapport et Timelog en définissant leurs attributs et leurs associations. Ce modèle fournit une abstraction des concepts clés du domaine – qui servira de point de départ pour la conception détaillée des classes, de l'artefact suivant.

Diagramme des Classes : offre une visualisation plus détaillée de la structure statique du système. Nous avons défini les classes, leurs attributs, méthodes, et les relations entre elles. L'objectif était de représenter de manière exhaustive les entités identifiées dans le modèle du domaine, ainsi que leurs interactions.

Le diagramme des classes comprend par ailleurs toutes les classes utilitaires que le système utilise. Notamment, les classes "Lecture" et "Ecriture" qui permettent respectivement la lecture et l'écriture d'objets Employe et/ou Projet dans des fichiers de format Json.

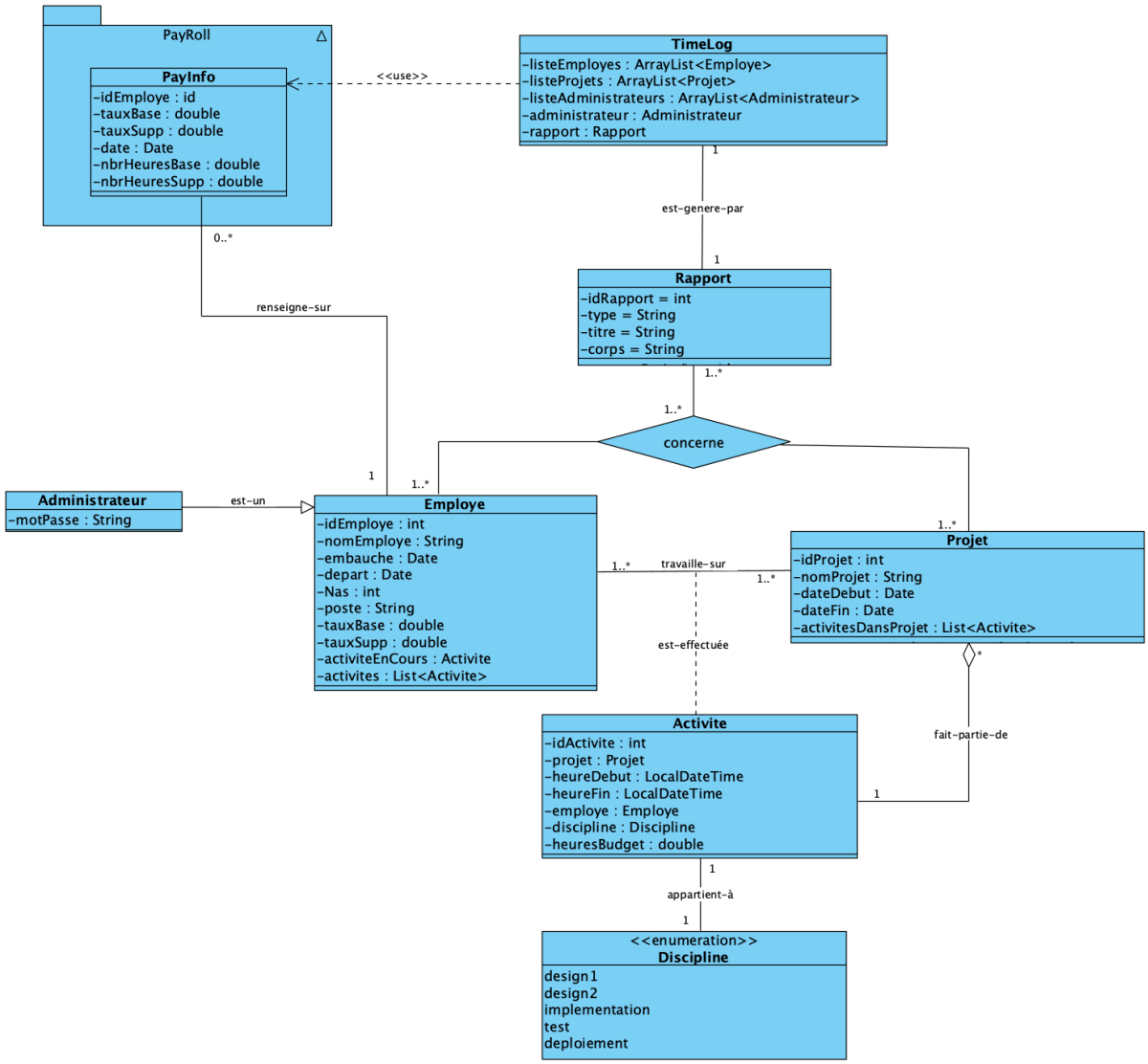
Ce diagramme prend également en considération l'interfaçage avec le futur sous-système de production et impression de chèques de paie *"PayRoll"*.

Les employés (objets de la classe Employe) sont reliés aux objets de type PayInfo (classe PayInfo). Timelog utilise une interface PayrollInterface, dont les opérations sont réalisées par le sous-système PayRoll, pour communiquer une liste d'objets PayInfo par employé(s), afin d'accéder aux opérations de production et impression de Payroll.

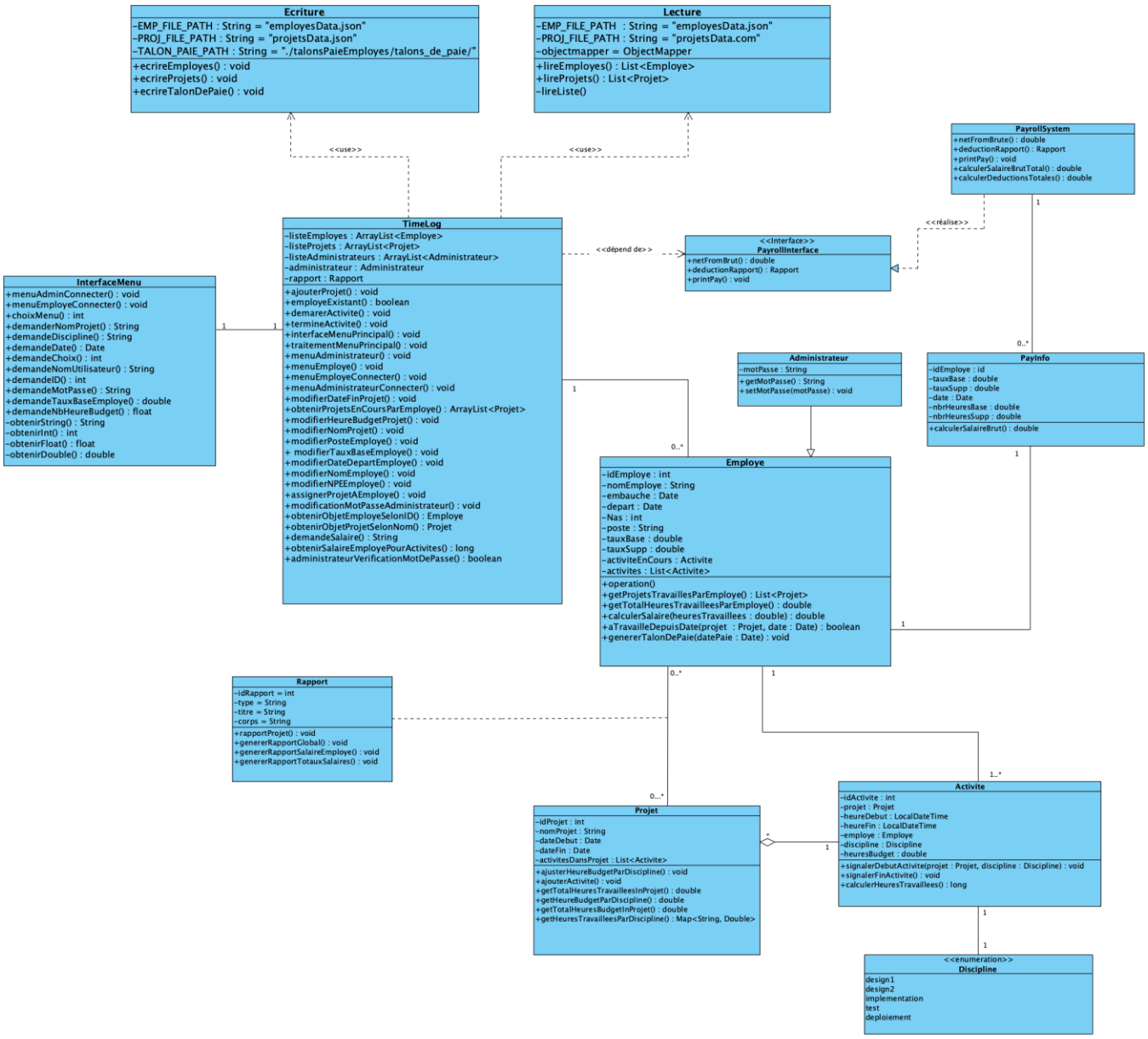
Pour le cas d'utilisation "Signaler Fin d'Activité", nous avons élaboré une Spécification de Système Distribué (DSS). Cela inclut une description détaillée des préconditions, postconditions et scénarios principaux, ainsi que des extensions potentielles. Le DSS fournit une vue approfondie des interactions nécessaires entre les acteurs et le système pour accomplir cette tâche spécifique.

Le diagramme d'interaction illustre dynamiquement le déroulement d'une opération spécifique, en l'occurrence, la signalisation de la fin d'une activité. Nous avons identifié les acteurs principaux, les étapes clés et les interactions entre eux. Cela permet de visualiser le flux de contrôle et de données pendant l'exécution de l'opération.

1. Modèle du domaine



2. Diagramme des classes



3. Cas d'utilisation "Signaler d'activité"

Titre : Signaler fin d'activité
Acteur principal : Employé
Acteur secondaire : -
Description : Ce cas d'utilisation permet à un usager "Employé" de signaler la fin d'une activité dans le système TimeLog. Il utilise l'interface en ligne de commande pour enregistrer l'heure de fin de l'activité, ainsi que d'autres informations pertinentes à celle-ci.
Préconditions :

- L'employé dispose d'un nom d'usager et un ID.
- L'employé a déjà signalé le début de cette activité.

Postconditions :

- L'heure de fin de l'activité est enregistrée dans le système.

– Le statut de l'activité est mis à jour dans TimeLog pour indiquer qu'elle est terminée.

Scénario principal

Actions des acteurs	Réponses du système
1. Ce cas d'utilisation débute lorsqu'un employé souhaite signaler la fin de son temps de travail sur projet.	
2. L'employé se connecte au système avec son nom d'usager et son ID.	3. Le système vérifie l'identité de l'employé.
	4. Le système valide les informations de l'employé et autorise l'accès à l'employé
	5. Le système affiche le menu principal des opérations possibles.
6. L'employé choisit de signaler la fin de l'activité.	7. Le système affiche la liste des projets en cours de l'employé.
8. L'employé sélectionne le projet qu'il souhaite terminer.	9. Le système demande à l'employé de confirmer son action.
10. L'employé exécute la commande de confirmation.	11. TimeLog enregistre l'heure de fin de l'activité en utilisant l'heure et la date actuelles du système.
	12. TimeLog met à jour le statut de l'activité pour indiquer qu'elle est terminée.
	13. Le système génère un message de confirmation pour indiquer que l'opération de l'employé a été enregistrée avec succès.
14. L'employé termine sa session sur TimeLog.	

Extensions (scénarios alternatifs)

A. Le Système TimeLog n'est pas accessible :

1. En cas de panne ou de maintenance, l'accès au système TimeLog est momentanément indisponible pour les usagers.
2. Un message s'affiche sur l'état du système, indiquant aux usagers l'heure et la date de remise en service du système.

2A. Le délai pour identifier l'employé est dépassé :

1. Le système avertit l'employé et quitte.

3A. Le système détecte une erreur dans les identifiants de l'employé :

1. Le système affiche un message d'erreur d'authentification.
2. Le système soumet le formulaire de connexion à nouveau à l'employé.
3. Le scénario principal reprend au point 2.

9A. L'employé annule l'opération de terminaison d'activité :

1. Le système annule l'opération et retourne au menu principal.
2. La fin d'activité n'est pas enregistrée.

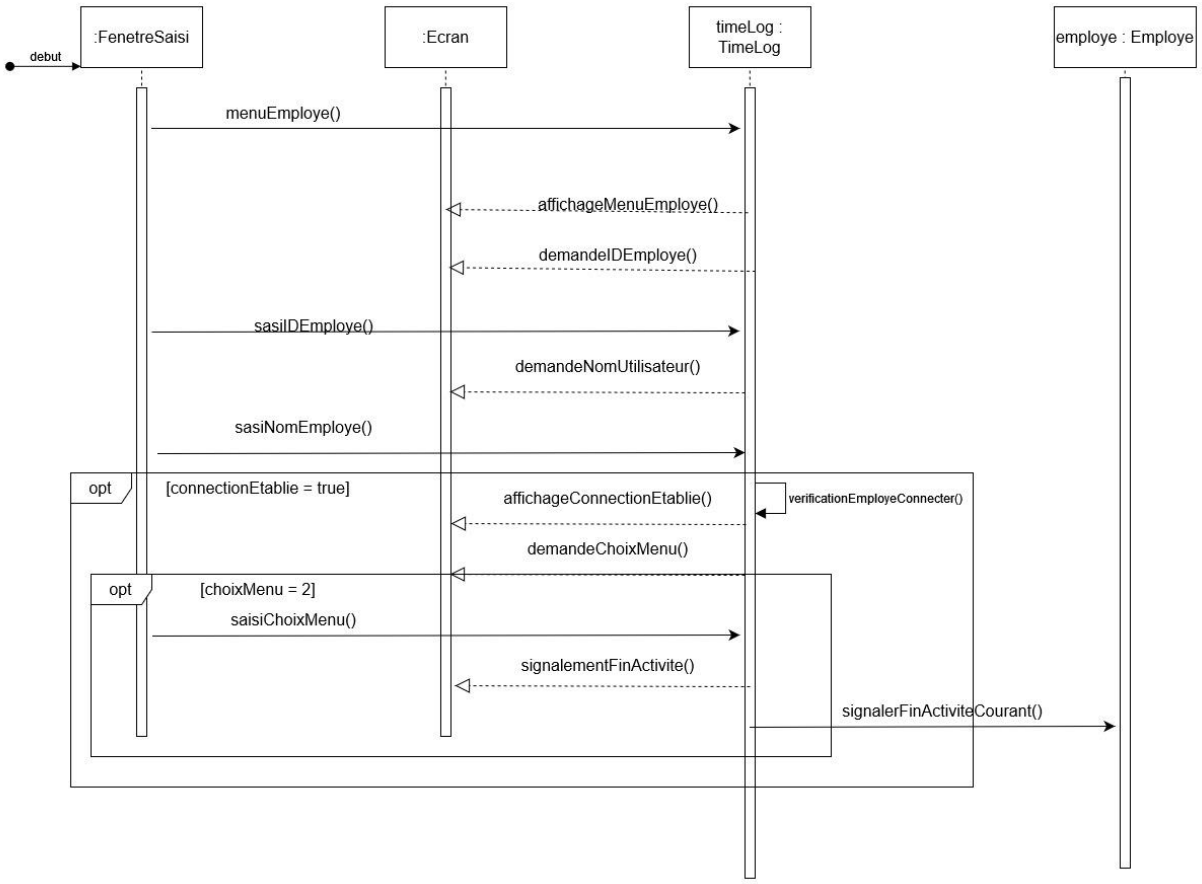
3. Le scénario principal reprend au point 5.

4. Diagramme de séquence système pour le cas d'utilisation "Signaler fin d'activité"

TimeLog: Scénario Signaler une fin d'activité

1. L'employe se arrive au systeme TimeLog se connecte à l'aide de l'interface.
2. Le systeme affiche un menu de connection.
3. L'employe choisi l'option de connection approprié.
4. Le systeme demande une entrée de ID.
5. L'employe sasi son ID.
6. Le systeme demande une entrée de nom.
7. L'employe saisi son nom d'utilisateur.
8. Le systeme demande un choix pour le menu.
9. L'employe saisi l'entrée 2 pour signer une fin d'activité.
10. Le systeme traite l'information afin de terminer l'activite en cours.

5. Diagramme d'interaction d'une opération du DSS "Signaler fin d'activité"



Exécution

Pour compiler et exécuter TimeLog :

- Ouvrir l'invite de commande
- Naviguer vers le répertoire du projet : cd chemin/vers/le/projet/
- Compiler les fichiers sources Java : javac src/*.java

Exécution à partie du fichier .exe

- Double clic sur le fichier program.exe
- Le point de d'entrée du programme est la fonction main() du fichier Program.java.

Sous-Sytème PayRoll

Le sous-système Payroll est une partie du projet qui gère la production et l'impression des éléments de paie des employés. Il comprend plusieurs classes, notamment **PayInfo**, **Payroll(interface)**, **PayrollSystem.** Ce sous-système interagit éventuellement avec d'autres classes connexes.
PayInfo : Cette classe stocke les informations liées à la paie d'un employé, telles que le nombre d'heures de base, le nombre d'heures supplémentaires, les taux horaires, etc. Elle dispose d'une méthode *calculerSalaireBrut()* qui prend en compte les heures de base et supplémentaires avec les taux horaires correspondants pour calculer le salaire brut.

PayrollInterface : Cette interface définit trois méthodes : *netFromBrute, deductionRapport,* et *printPay*. Ces méthodes servent à calculer le salaire net total, générer un rapport de déductions, et afficher les chèques de paie.
PayrollSystem : Cette classe implémente l'interface *PayrollInterface* et fournit des implémentations pour les méthodes définies dans cette interface. Elle contient des méthodes pour calculer le salaire net total à partir des informations de paie, générer un rapport de déductions, et afficher les chèques de paie. Elle utilise la classe Rapport pour stocker les informations sur les déductions.
TimeLog : Cette classe gère l'interaction avec l'utilisateur via le menu console. Elle utilise le PayrollSystem pour générer des rapports, calculer les salaires, etc.

1. Illustration de l'interfaçage avec le futur système

Voici une illustration générale de la manière dont on peut interfacer ces deux systèmes :

- Création d'une instance de PayrollSystem dans la classe TimeLog: ceci peut être fait lors de l'initialisation de la classe.

```
public TimeLog() {
    // Initialisation des listes de projets et d'employés
    projets = new ArrayList<>();
    employes = new ArrayList<>();

    // Initialisation du système de paie
    payrollSystem = new PayrollSystem();
```

- Utiliser l'instance de PayrollSystem pour appeler les méthodes nécessaires dans les différentes parties où cela est approprié (calcul salaire : appelant la méthode netFromBrute de la classe payrollSystem, Afficher le Rapport de déduction et imprimer la fiche de paie avec la méthode PrintPay de la classe payrollSystem)

Annexes

1. Contribution et Collaboration

- Ken a contribué l'implémentation des classe java comme par exemple, Timelog, InterfaceMenu, Program et en partie certaine classe comme Employe et Activite. De plus, il a aussi effectué certain test comme **test_9,test_10,test_11,test_12, test_13, test_14 et test_15** dans le fichier *code/target/test-*classes. Il a également participé à la conception de certain diagramme comme le diagramme de séquence du cas d'utilisation Signaler fin d'activité et le DSS de ce cas. Pour ajouter, il a également fait une compilation du système afin de créer un fichier exécutable du système TimeLog.
- Maryam a contribué à l'implémentation des classes Java comme par exemple, Rapport, Projet, Employe, Lecture, Ecriture, Activite en partie les classes comme TimeLog, InterfaceMenu et Program. De plus, elle a également effectué les diagrammes tels que le diagramme du modèle, le diagramme de classes et les diagrammes des cas d'utilisation. Pour ajouter, elle a également effectué les parties de Description de projet, Conception et Modélisation, Cas d'utilisation et la partie Exécution dans le rapport.
- Darin a contribué à la modélisation et implémentation du sous-système PayRoll. Notamment les classes java tel que : PayInfo, PayrollSystem et l'interface PayrollInterface , Activité, et Employé. A noter également, l'ajustement TimeLog pour l'interaction avec la classe PayrollSystem. Il a également travaillé sur la classe rapport avant modification et nouvelle approche de résolution.

En somme, le travail collaboratif sur le projet TimeLog s'est réparti sur plusieurs rencontres durant la session d'étude - au travers un serveur Discord réservé au développement de ce projet. Les trois membres ont contribué aux différentes étapes du projet, notamment la modélisation, l'implémentation et la validation et la maintenance. Aussi, ont-ils contribué à la rédaction de ce document de synthèse.

2. GITHUB

Voici le lien GitHub concernant le projet:

https://github.com/itsmimzi/ProjetINF1163-2023/tree/Demo-VersionFinal

Tous les diagrammes présents dans le fichier de retrouve aussi dans le répertoire sur GitHub.