****



**Abdoulaye WANE**

Rapport de Stage (2023-2024)

**Lycée Privé Saint-Bénigne de Dijon**

**Projets : Outil permettant l’analyse de logs sur une machine de conditionnement et module sur un outil existant d’envoi de notes.**

****

**Table des matières :**

[**Glossaire** 3](#_Toc162523570)

[I. Présentation de l’organisation et des projets du stage. 4](#_Toc162523571)

[I.1.Présentation du groupe IDEMIA : Leader mondial de l’identité numérique. 4](#_Toc162523572)

[I.1.1 Fonctionnement du support régional. 4](#_Toc162523573)

[I.1.1.1. Missions et Activités 4](#_Toc162523574)

[I.1.1.2. Compétences Clés 5](#_Toc162523575)

[I.1.1.3. Structure du Support 5](#_Toc162523576)

[I.2.Présentation des projets 6](#_Toc162523577)

[I.2.1 Kernlogextractor 6](#_Toc162523578)

[1.2.1.1 Les Machines 6](#_Toc162523579)

[1.2.1.1.1. Fonctionnement simplifié : 6](#_Toc162523580)

[1.2.1.1.2. Avantages de la machine Kern et Casillo : 6](#_Toc162523581)

[1.2.1.2. L’Application 7](#_Toc162523582)

[1.2.1.2.1. Fonctionnalités 7](#_Toc162523583)

[1.2.1.2.2. Traitement des fichiers de log 7](#_Toc162523584)

[1.2.1.2.3. Traitement des fichiers de log Cassilo 7](#_Toc162523585)

[1.2.1.2.4. Création de la timeline 8](#_Toc162523586)

[I.2.2 NDT (Note delivery Tool) 8](#_Toc162523587)

[1.2.2.1. L'Application NDT 8](#_Toc162523588)

[II. Les tâches et les missions réalisées. 10](#_Toc162523589)

[II.1. Logiciels, technologies et langages utilisés 10](#_Toc162523590)

[II.2. En quoi consistait ma mission 11](#_Toc162523591)

[II.2.1. Kernlogextractor 12](#_Toc162523592)

[II.2.1.1. Développement de la fonctionnalité de sélection du mode Kern/Cassilo 12](#_Toc162523593)

[II.2.1.2. Développement de la fonctionnalité de lecture et d'extraction des fichiers de logs Cassilo 15](#_Toc162523594)

[II.2.1.3. Développement de la fonctionnalité de parsing des logs Cassilo 16](#_Toc162523595)

[II.2.2.NDT (Note Delivery Tool) 18](#_Toc162523596)

[II.2.2.1. Développement du système de notifications 18](#_Toc162523597)

[II.2.2.2. Développement de la fonctionnalité de statut "En transit" 23](#_Toc162523598)

[II.2.2.3. Développement de la fonctionnalité de gestion des listes de diffusion 24](#_Toc162523599)

[II.2.2.4. Développement de la fonctionnalité de modification de la hauteur des lignes 26](#_Toc162523600)

[II.2.2.5. Développement de la fonctionnalité d'envoi à toute la production 27](#_Toc162523601)

[II.2.2.6 Développement du statut "OK mais pas d'action nécessaire" 28](#_Toc162523602)

[II.2.2.7 Développement de la fonctionnalité de retraitement d'une note côté récepteur 29](#_Toc162523603)

[III. Immersion professionnelle 29](#_Toc162523604)

[**1. Amélioration de la gestion des notifications :** 30](#_Toc162523605)

[**2. Suivi des actions et collaboration entre les équipes :** 30](#_Toc162523606)

[**3. Ergonomie et accessibilité de l'application :** 30](#_Toc162523607)

[**4. Déploiement des nouvelles versions des applications** 31](#_Toc162523608)

[VI. Conclusion et remerciements 32](#_Toc162523609)

## **Glossaire**

**A**

* **Application NDT :** Note Delivery Tool, application de messagerie interne pour la communication entre le service client et la zone de production.
* **Azure DevOps :** Plateforme de développement et d'opérations cloud de Microsoft.

**B**

* **Bouton radio :** Élément d'interface utilisateur permettant de choisir une option parmi un ensemble de choix exclusifs.
* **C# :** Langage de programmation orienté objet développé par Microsoft.

**E**

* **Enumération :** Type de données permettant de définir un ensemble de valeurs nommées et constantes.
* **Excel :** Logiciel tableur développé par Microsoft.

**F**

* **Fichier de log :** Fichier contenant des informations sur l'activité d'un système ou d'une application.
* **Fonctionnalité :** Action ou ensemble d'actions qu'un système ou une application est capable de réaliser.

**H**

* **HSA** : High Security Area

**I**

* **IDEMIA :** Groupe français leader mondial de l'identité numérique.
* **Interface utilisateur :** Ensemble des éléments d'un système ou d'une application permettant à l'utilisateur d'interagir avec celui-ci.

**L**

* **Liste de diffusion :** Groupe de destinataires prédéfinis pour l'envoi de messages.

**M**

* **Machine Kern et Casillo :** Machines automatisant l'insertion des cartes dans les enveloppes.
* **Mode Kern/Cassilo :** Modes de fonctionnement de l'application KernLogExtractor pour traiter les fichiers de logs des machines Kern et Casillo.
* **Module :** Composant logiciel indépendant pouvant être intégré à un système plus large.

**N**

* **Note :** Message envoyé via l'application NDT sender.

**O**

* **Opérateur :** Utilisateur de l'application NDT.

**P**

* **Production :** Ensemble des services et systèmes utilisés pour la production d'un produit ou d'un service.

**R**

* **Retraitement d'une note :** Action de relancer le traitement d'une note côté récepteur.

**S**

* **Statut :** État d'une note ou d'un fichier dans l'application NDT.
* **Système de notifications :** Ensemble de composants permettant d'avertir l'utilisateur de nouveaux événements ou de changements d'état.

**T**

* **Timeline :** Représentation chronologique des activités et des inactivités de la machine Kern ou Cassilo .
* **TulPep :** Type de notification apparaissant en bas à droite de l'écran.

**U**

* **Utilisateur :** Personne utilisant l'application NDT.

**W**

* **Winforms :** Framework d'interface utilisateur graphique pour le développement d'applications Windows.

**X**

* **XML :** Langage de balisage permettant de représenter des données structurées.

**Y**

* **Aucun :** Mot-clé utilisé dans l'application NDT pour signifier qu'aucune action n'est nécessaire.

**Z**

* **Zone de production :** Zone où les produits ou services sont fabriqués ou assemblés.

## I. Présentation de l’organisation et des projets du stage.

### I.1.Présentation du groupe IDEMIA : Leader mondial de l’identité numérique.

**IDEMIA** est un groupe français leader mondial de l'identité numérique, présent dans plus de 180 pays et employant plus de 15 000 personnes. Créé en 2017 par la fusion d'Oberthur Technologies et de Morpho, IDEMIA s'est imposé comme un acteur incontournable de la sécurité numérique en proposant des solutions innovantes et de haute qualité.

**Le groupe se compose de trois divisions principales :**

* **Identity & Access Management (IAM) :** Cette division fournit des solutions d'authentification et de gestion des accès pour les entreprises et les gouvernements.
* **Public Security & Identity :** Cette division propose des solutions biométriques et d'identification pour les forces de l'ordre et les administrations publiques.
* **Secure Transactions :** Cette division développe des solutions de paiement et de sécurité pour les transactions électroniques.

**IDEMIA s'engage à protéger l'identité de ses clients et à garantir la sécurité de leurs données.** Le groupe s'appuie sur une expertise de pointe en biométrie, cryptologie et intelligence artificielle pour proposer des solutions innovantes et répondant aux besoins de ses clients.

**IDEMIA est un groupe en pleine croissance et qui s'investit dans l'innovation.** Le groupe est un acteur majeur de la transformation numérique et s'engage à contribuer à la construction d'un monde plus sûr et plus fluide.

#### I.1.1 Fonctionnement du support régional.

##### I.1.1.1. Missions et Activités

Le support régional d'IDEMIA à Dijon se distingue par sa diversité d'interventions sur différents sites et auprès de clients variés. Ses missions principales englobent l'assistance technique, avec un support aux utilisateurs sur une panoplie d'outils métiers, le déploiement de solutions et la formation des utilisateurs, ainsi que la prestation de services de consulting pour répondre aux besoins spécifiques des clients. Il se démarque également par son expertise technique, se traduisant par le développement d'outils internes pour l'analyse des problèmes et l'optimisation des solutions, et une étroite collaboration avec les équipes de développement. Enfin, la relation client occupe une place centrale, avec une communication étroite et une adaptation constante aux besoins et cultures des clients.

##### I.1.1.2. Compétences Clés

Les compétences recherchées au sein du support régional d'IDEMIA sont diverses. Sur le plan technique, une maîtrise des outils métiers est indispensable, de même que la capacité à analyser et résoudre des problèmes complexes et des connaissances en développement logiciel. Sur le plan relationnel, les candidats doivent faire preuve d'aisance dans la communication avec des interlocuteurs variés, d'adaptabilité et de respect des différences culturelles, ainsi que d'un esprit d'équipe prononcé. Enfin, la maîtrise de l'anglais est un atout majeur, étant la langue de communication interne.

I.1.1.b. Chiffres Clés

Le support régional d'IDEMIA compte 30 personnes au sein de son équipe, dont 8 personnes pour la région Europe Moyen-Orient Afrique.

##### I.1.1.3. Structure du Support

La structure du support régional est organisée en équipes spécialisées par zone géographique ou par domaine d'expertise, favorisant ainsi une meilleure répartition des compétences et une prise en charge plus efficace des clients.

Le support régional d'IDEMIA à Dijon représente un pilier essentiel de l'entreprise, alliant expertise technique et relation client. Il offre aux candidats une opportunité d'évolution dans un environnement international et stimulant, où les compétences techniques et relationnelles sont indispensables pour réussir.

### I.2.Présentation des projets

#### I.2.1 Kernlogextractor

##### 1.2.1.1 Les Machines

**La machine Kern et Casillo : automatisation de l'insertion des cartes dans les enveloppes.**

**Kern** et **Casillo** sont des machines qui automatisent l'insertion des cartes dans les enveloppes. Dotées d'une technologie de pointe, elles permettent de gagner du temps et d'améliorer la précision de ce processus.

###### 1.2.1.1.1. Fonctionnement simplifié :

1. **Chargement des cartes et des enveloppes :** Les cartes et les enveloppes sont placées dans des feeders dédiés.
2. **Insertion précise :** La machine sélectionne l'enveloppe correspondante et insère la carte avec précision.
3. **Fermeture et empilage :** Les enveloppes sont ensuite fermées et empilées pour une expédition rapide.

###### 1.2.1.1.2. Avantages de la machine Kern et Casillo :

* **Gain de temps :** Permet de traiter un grand nombre de cartes en un temps record, réduisant considérablement la charge de travail.
* **Amélioration de la précision :** Réduit les erreurs humaines liées à l'insertion manuelle des cartes.

Exemple de la machine Kern :



##### 1.2.1.2. L’Application

L'application KernLogExtractor extrait les plages d'activité et d'inactivité de la machine Kern en analysant ses journaux. Elle remplace la saisie déclarative quotidienne des opérateurs par une extraction automatique des informations.

###### 1.2.1.2.1. Fonctionnalités

* **Affichage d'une grille**
  + Opérateur
  + Client (SG, BNP...)
  + Heure de début
  + Heure de fin
  + Durée
  + Nombre de cartes appairées
* **Insertion des durées d'inactivité** (en rouge)
* **Total d'activité et nombre de cartes** de la journée

###### 1.2.1.2.2. Traitement des fichiers de log

* **Deux types de fichiers :**
  + PersonalizedMatchingLog\*.txt (plus simples)
  + MatchingLog\*.txt (plus détaillés)
* **Informations extraites :**
  + Nom de l'opérateur
  + Timestamp
  + Timestamp de lecture des données carte
  + Données lues sur l'encart et la carte

###### 1.2.1.2.3. Traitement des fichiers de log Cassilo

* **Fichiers .txt** avec "Firm: " dans la première ligne
* **Informations extraites :**
  + Nom du setup (client)
  + Nom de l'opérateur (peu fiable)
  + Timestamp pour chaque carte

###### 1.2.1.2.4. Création de la timeline

* **Détection des intervalles d'activité/inactivité**
* **Consolidation des intervalles** à partir de plusieurs fichiers
* **Combler les trous** avec des périodes d'inactivité configurables (30 secondes, 10 minutes...)

Ci-dessous nous avons l’interface de l’application, dont la Timeline est alimenté avec les logs de la machine Kern :

**Date 
09102/2023 
See 'Ficxfs inactivity 
F rom 
202302M 14:42 
202302.109 09:31 
202302M 095895 
202302M 
202302.109 
2023.02M 102198 
Cl Custom range 
From 19902/2023 
'600212023 
Search 
Custorner 
cards • 
. 2430 
05:00 
1245 
Nb cards 
1749 
Duration 
ooh ogm 41s 
ooh ogm 035 
ooh 23m 03s 
Total activiW • 
00:41 /07:45 **

#### I.2.2 NDT (Note delivery Tool)

##### 1.2.2.1. L'Application NDT

**Objectif :**

* Faciliter la communication entre le service client et la zone de production
* Répondre aux besoins spécifiques de chaque groupe d'utilisateurs
* NDT est mis en place car il y a une interdiction formelle de texte libre qui transitent de la HSA vers l’office.
* Elle permet de substituer à une application de messagerie classique type mail

qui est interdite en zone HSA.

1.2.2.1.1. **Fonctionnalités :**

* **Gestion des notifications :**
  + Clignotement de l'icône
  + Tous les membres d’un même service voient les notes envoyées et les notes reçues
  + Le traitement d’une note peut se faire depuis n’importe quel poste du service
* **Gestion des destinataires :**
  + Sélection individuelle ou groupée
* **Suivi des notes :**
  + Accusé de réception

**À noter :**

Il n’y a pas d’infrastructure de type base de données ou de serveur central.

**Les trois types d’application NDT :**

* **NDT (sender) - Service Client :**

**Une image contenant cercle, Caractère coloré, vert, Graphique

Description générée automatiquement**

* + Destinée à l'envoi de notes vers la zone de production.
  + Les utilisateurs du service client ont exprimé le besoin d'un clignotement plus évident de l'icône de l'application à chaque notification.

Ci-dessous voici l’interface de l’application NDT sender existante :

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

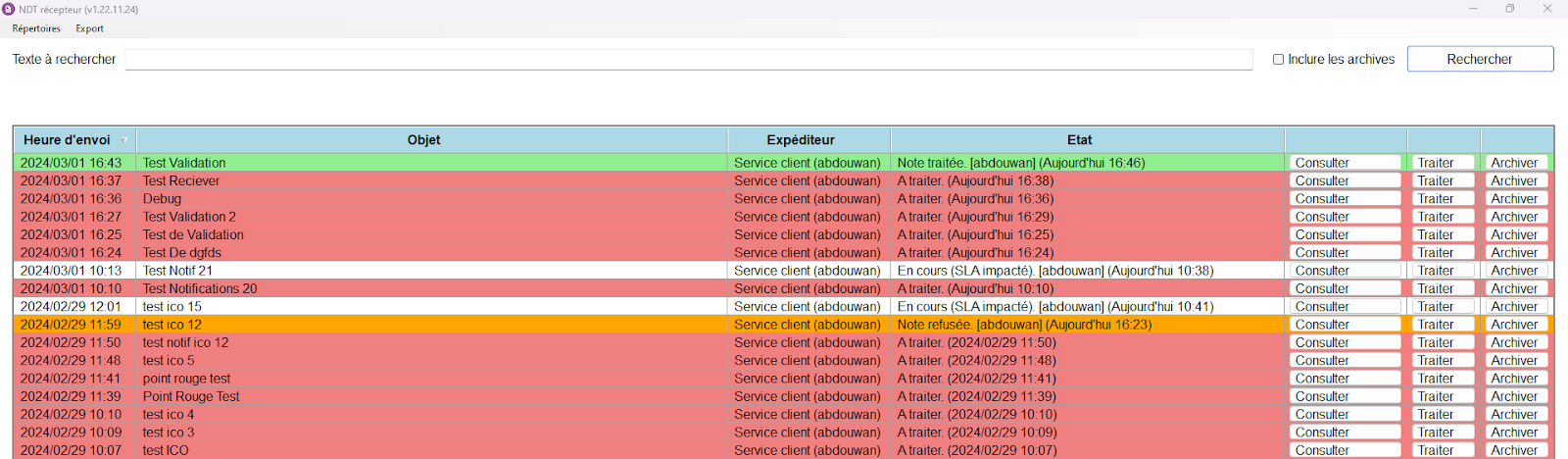
* **NDT (receiver) :**

Une image contenant cercle, violet, Graphique, Lilas

Description générée automatiquement

* + Dédiée à la réception des messages du service client, avec la possibilité de réponse restreinte à des messages prédéfinis.
  + Les utilisateurs de la zone de production ont demandé que les messages soient envoyés à tous les destinataires concernés, sans avoir à sélectionner individuellement les destinataires.
  + Ils souhaitent également recevoir une notification plus visible, avec un clignotement d'icône plus prononcé pour toutes les notifications, indépendamment de la nature de la note.

Ci-dessous voici l’interface de l’application NDT receiver existante :



* **NDT (Viewer) :**



* Elle permet d’avoir un aperçu global des notes qui sont reçu côté sender et côté receiver.
* Dans notre développement sauf au niveau de l’interface graphique, il n’y aura pas spécialement de changement au niveau des fonctionnalités

Ci-dessous voici l’interface de l’application NDT Viewer existante :

Une image contenant texte, capture d’écran, ligne, logiciel

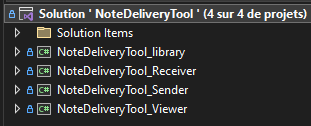
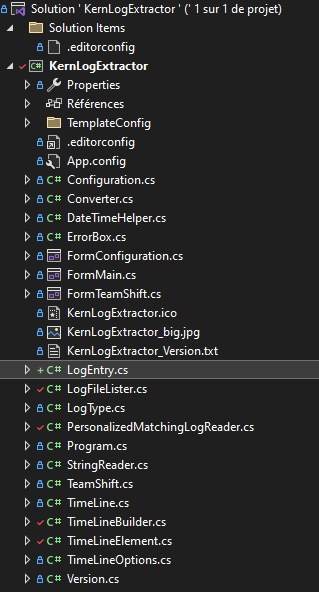
Description générée automatiquement

## II. Les tâches et les missions réalisées.

### II.1. Logiciels, technologies et langages utilisés

* C# (langage de programmation)
* Framework GUI WinForms (Windows Forms)
* XML
* Visual Studio 2022 (Environnement de développement)
* Azure DevOps
* Excel

Ci-dessous nous avons les deux solutions sur environnement de développement qui représentes les deux projets, respectivement KernlogExtractor et ensuite Note Delivery Tool :



### II.2. En quoi consistait ma mission

D’abord pour le projet KernLogExtractor, notre mission consistait à développer une nouvelle fonctionnalité sur l’application existante. Au niveau de l’interface on doit être capable de lire et d'extraire des informations à partir des fichiers de logs Casillo. Nous avons intégré deux radio bouton dans l'interface graphique pour sélectionner entre Kern et Casillo, et implémenter la modification de la configuration en fonction de ce choix. Nous avons également créé une fonction générique pour gérer les logs de la Kern et de la Cassilo, en sélectionnant les bons fichiers, en listant les sous-répertoires entre les dates de début et de fin spécifiées, et en créant une méthode pour parser les fichiers de log de la Cassilo. Pour finir, nous avons assemblé ces fonctionnalités dans une fonction générique et ajouté une méthode prenant en paramètre tous les fichiers parsés et retournant une liste d'éléments TimeLineElement.

Ensuite, pour le projet NoteDeliveryTool (NDT), notre mission consistait à apporter plusieurs améliorations à l'outil de gestion des notes. Dès le début, des ajustements ont été nécessaires au niveau de l’interface graphique en corrigeant quelques bugs de positionnement des éléments de la form. Un système de notifications a été mis en place pour tous les types de messages, avec des notifications visuelles pour le récepteur et l'expéditeur. La visibilité des notifications a été améliorée en ajoutant une fenêtre contextuelle de notification et en ajustant la hauteur des lignes pour un affichage plus lisible. La couleur de l'état "En transit" a été modifiée pour le rendre plus visible, et une fonctionnalité a été ajoutée pour afficher la durée pendant laquelle un message est en transit. La gestion des listes de diffusion a également été prise en charge, permettant d'ajouter et de supprimer des membres d'une liste de diffusion dans le fichier de configuration. Enfin, une option a été ajoutée pour envoyer des notes à l'ensemble de la production, et un nouveau statut "OK mais pas d'action nécessaire" a été implémenté. De plus, une reprise de traitement de note côté récepteur a été réalisée pour mettre à jour l'utilisateur ayant traité la note, même avec le même statut.

#### II.2.1. Kernlogextractor

##### II.2.1.1. Développement de la fonctionnalité de sélection du mode Kern/Cassilo

**Objectif :**

Ajouter deux radios bouton dans l'interface graphique pour choisir si l'application fonctionne en mode Kern ou Cassilo.

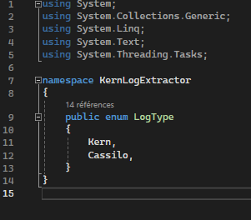
**Contraintes :**

* Les boutons doivent être bien distingués sur l’interface existante.
* Le choix du mode doit impacter la configuration de l'application.
* L'application doit être capable de lire et de modifier le fichier de configuration.

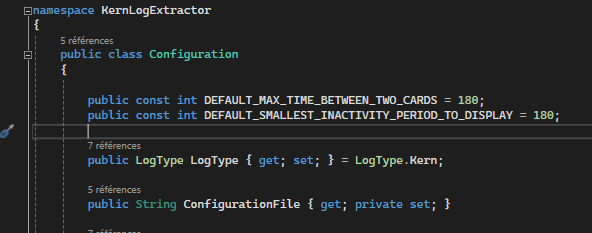
**Développement :**

**1. Ajout d’un énuméré « LogType » dans le fichier LogType.cs.**

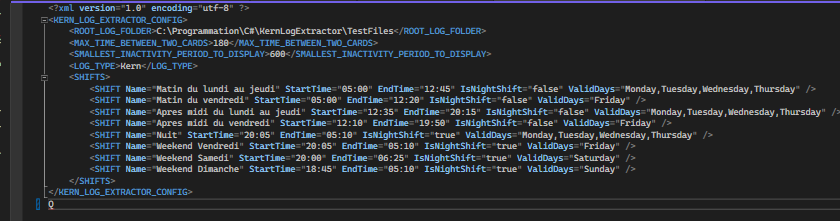
Dans notre cas nous avons opté pour l’ajout d’un énuméré pour différencier les logs d’une machine Kern d’une machine Cassilo.

****

Puis cela implique la modification de la classe Configuration, car on veut qu’elle soit par défaut en mode Kern :

( ( き て ま よ ま 1 」 言 新 も 11 ー ご 」 に こ 
~ 籃 1 ・ 新 1 

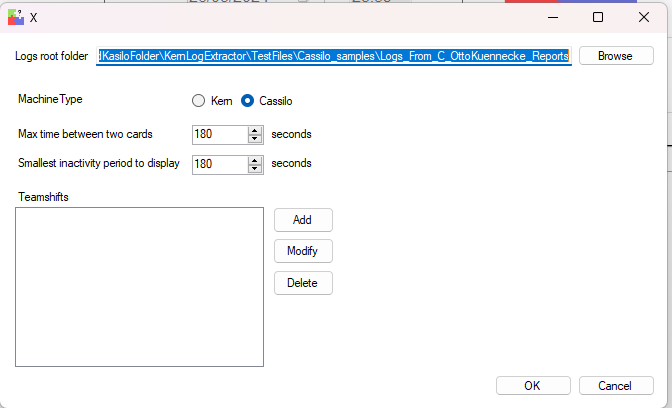
Pour ensuite modifier le fichier de configuration XML en ajoutant la balise <LOG\_TYPE>Kern</LOG\_TYPE>:



**2. Ajout des radios boutons dans l'interface graphique :**

* Identifier un emplacement adéquat pour le bouton dans l'interface.
* Choisir un style de bouton cohérent avec l'interface existante.
* Définir le texte du bouton.

Étant donné que nous ouvrons une boîte de dialogue, les actions sur les radios bouton ne sont enregistrés qu'après avoir cliqué sur le bouton “OK”.

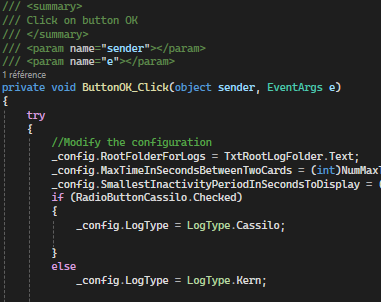


Donc dans la méthode ButtonOK\_Click on s’assure que :

RadioButtonCassilo.Checked : Vérifie si le bouton radio "RadioButtonCassilo" est coché.

Si oui, la propriété "LogType" de l'objet de configuration est définie à "LogType.Cassilo".

Sinon (bouton radio non coché), la propriété "LogType" de l'objet de configuration est définie à "LogType.Kern". Alors la configuration prendra en compte le fait qu’il existe 2 types de deux de logs possibles (Cassilo et Kern).

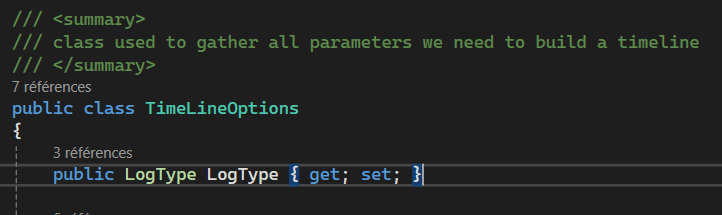


**2. Changement de la configuration en fonction du mode choisi :**

* Lire la configuration du fichier de configuration au démarrage de l'application.
* Modifier la configuration en fonction du mode sélectionné par l'utilisateur.
* Enregistrer la configuration modifiée dans le fichier de configuration.

**3. Ajout d'un champ dans le type TimeLineOptions :**

* Ajout d’une propriété de type “LogType” à la structure TimeLineOptions pour indiquer le mode sélectionné (Kern ou Cassilo).



**4. Test du mode Kern par défaut :**

* Vérifier si la configuration du fichier est absente au démarrage de l'application.
* Si la configuration est absente, activer le mode Kern par défaut.

**Résultats obtenus :**

* Bouton "Kern/Cassilo" ajouté à l'interface graphique.
* Fonctionnalité de changement de mode implémentée.
* Configuration de l'application modifiée en fonction du mode choisi.
* Mode Kern activé par défaut si la configuration est absente.

##### II.2.1.2. Développement de la fonctionnalité de lecture et d'extraction des fichiers de logs Cassilo

**Objectif :**

Nous voulons lire les fichiers de logs Cassilo et extraire les informations, telles que le nom du client, les heures de début et de fin de production des cartes pour les afficher dans l'application.

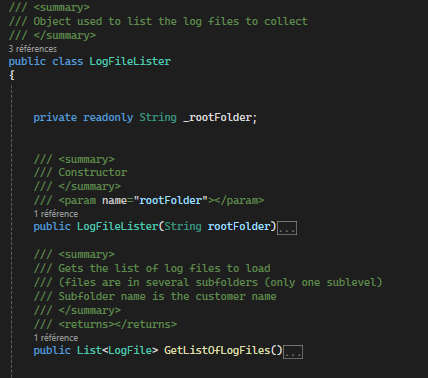
**Développement :**

**1. Analyse des fichiers de logs Cassilo :**

* Étude du format des fichiers de logs Cassilo.
* Identification des différents types d'informations présentes dans les fichiers de logs.
* Définition des règles d'extraction des informations.

**2. Conception de la logique d'extraction des informations :**

Dans notre cas, nous avons utilisé la classe LogFileLister pour lister les fichiers .txt dans les sous-dossiers du dossier racine des journaux. :



**Résultats obtenus :**

* Fonctionnalité de lecture et d'extraction des fichiers de logs Cassilo implémentée.

*Une image contenant Police, texte, logo, blanc

Description générée automatiquement*

Après le choix de la configuration en mode Cassilo :

*Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement*

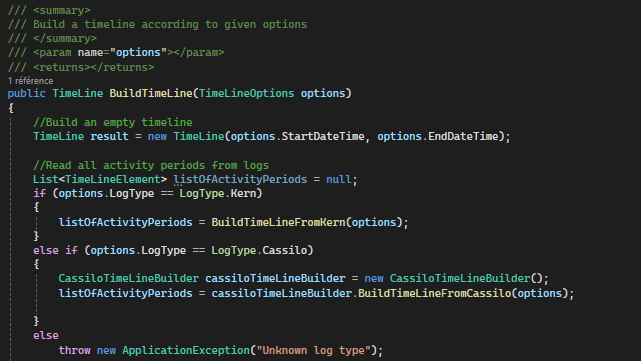
##### II.2.1.3. Développement de la fonctionnalité de parsing des logs Cassilo

**Objectif :**

En tant que développeur, je veux pouvoir parser les logs de la machine Cassilo.

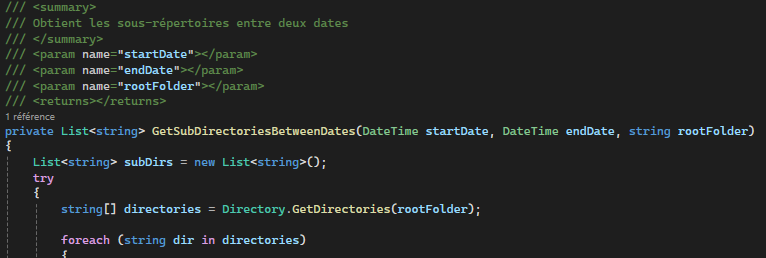
**Développement :**

**1. Utiliser une fonction générique pour gérer les logs de la Kern et de la Cassilo :**

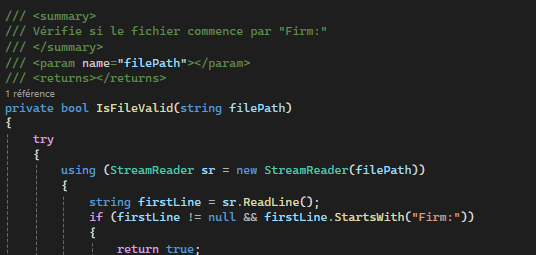


**2. Sélection des bons fichiers :**

Définition de la méthode GetSubDirectoriesBetweenDates pour lister les sous-répertoires compris entre la date de début et la date de fin (StarteDate -1 et EndDate +1).

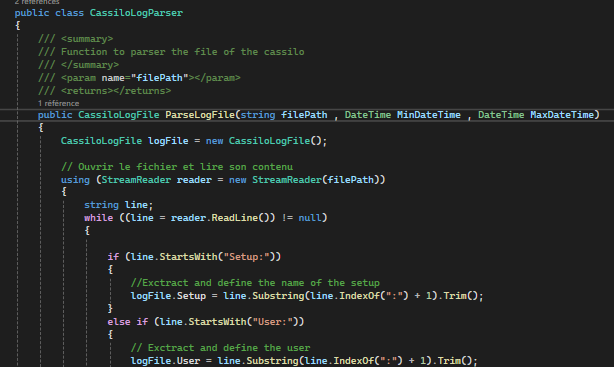


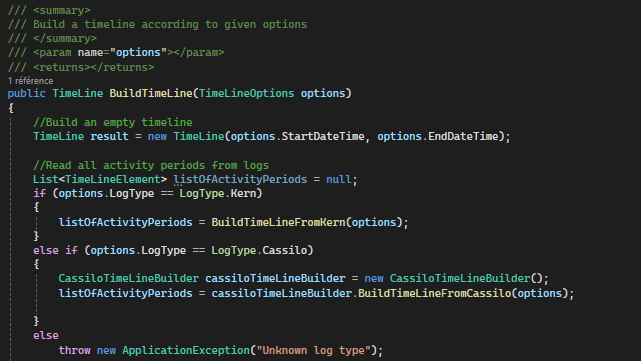
Définition de la méthode IsValid pour lister les fichiers valides (extension .txt et nom commençant par "Firm :").



**3. Parsing des fichiers de log de la Cassilo : ajout de la classe CassiloLogParser**

* On ajoute la méthode ParseLogFile pour parser les fichiers de log de la Cassilo.



**4. Assemblage des fonctions dans la fonction générique « «BuildTimeLine »:**

Cette méthode permet de faire les actions suivantes :

* Si le type de log est LogType.Kern, la méthode BuildTimeLineFromKern(options) est appelée pour construire la liste d'activité à partir des logs de type Kern et l'affecte à listOfActivityPeriods.
* Si le type de log est LogType.Cassilo, une nouvelle instance de CassiloTimeLineBuilder est créée. Ensuite, la méthode BuildTimeLineFromCassilo(options) de cette instance est appelée pour construire la liste d'activité à partir des logs de type Cassilo et l'affecte à listOfActivityPeriods.

Enfin on combine les résultats des différentes fonctions pour obtenir une liste complète d'objets TimeLineElement.

**Résultats obtenus :**

* Fonctionnalité de parsing des logs Cassilo implémentée.
* Informations extraites des logs et stockées dans une liste d'objets TimeLineElement.
* Fonction générique pour gérer les logs de la Kern et de la Cassilo.
* Informations extraites des fichiers de logs et affichées dans l'application.

#### II.2.2.NDT (Note Delivery Tool)

##### II.2.2.1. Développement du système de notifications

**Objectif :**

Mettre en place un système de notifications pour tous les types de messages dans l'application.

**Contraintes :**

* Les notifications doivent être claires et visuellement distinctes pour chaque type de message.
* Les notifications doivent être minimales et ne pas perturber l'utilisateur.

**Développement :**

**1. Définition des types de notifications en fonction des couleurs existantes pour les l’état des notifications de l’interface graphique :**

**Pour le sender :**

|  |  |  |
| --- | --- | --- |
| Statut | Signification | Couleur indicative de la notification |
| **Note reçue** | La note a été reçue mais n'a pas encore été traitée. | - |
| **OK, on va le faire** | La note a été prise en charge et est en cours de traitement. | - |
| **OK, on va le faire mais délai dépassé** | La note est en cours de traitement, mais le délai (SLA) n'est pas respecté. | Orange |
| **Note refusée** | La note a été refusée, la demande ne sera pas traitée. | Rouge |
| **OK c'est fait** | La note a été traitée avec succès. | Vert |
| **OK c'est fait mais délai dépassé** | La note a été traitée, mais le délai (SLA) n'a pas été respecté. | Orange |
| **Le service n'est pas concerné** | La note a été reçue par erreur, elle ne concerne pas ce service. | Orange |
| **Aucune action nécessaire (État que l’on va expliquer plus tard )** | La note a été traitée et aucune action supplémentaire n'est requise de la part de ce service. | Vert |
| **Etat indéterminé** | Le statut de la note n'a pas pu être déterminé, une attention particulière est requise. | Rouge |

**Pour le receiver :**

|  |  |  |
| --- | --- | --- |
| Statut de transfert | Description | Couleur d'affichage |
| Envoyé | La note est en train d'être acheminé vers son destinataire. | Bleu |
| A traiter | La note est arrivé et nécessite un traitement. (Heure de dernière mise à jour fournie) | Rouge |
| En cours | Le traitement du fichier a commencé par [Destinataire]. (Heure de dernière mise à jour fournie) | - |
| En cours (retard) | Le traitement du fichier a commencé par [Destinataire] mais accuse un retard. (Heure de dernière mise à jour fournie) | Orange |
| Refusé | La note a été refusé par [Destinataire]. (Heure de dernière mise à jour fournie) | Orange |
| Traité | La note a été traité par [Destinataire]. (Heure de dernière mise à jour fournie) | Vert |
| Traité (retard) | La note a été traité par [Destinataire] mais accuse un retard. (Heure de dernière mise à jour fournie) | Vert |
| Service non concerné | Le service ne prend pas en charge cette note . (Heure de dernière mise à jour fournie) | Vert |
| Aucune action nécessaire | Aucune action n'est requise de notre part. Le fichier a été traité par [Destinataire]. (Heure de dernière mise à jour fournie) | Vert |
| Indéterminé | Le statut du fichier est inconnu. (Heure de dernière mise à jour fournie) | Rouge |

* Définir le contenu et le style de chaque type de notification en fonction de la couleur du type de message. Ci-dessous par exemple nous avons la couleur verte pour une notification qui à été traitée avec le statut « Ok , c’est fait  » , puis nous avons le titre du type de NDT et enfin nous avons une icone qui représente l’application .

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**2. Implémentation des notifications :**

Dans notre contexte, il existait déjà un système de notification, raison pour laquelle nous allons utiliser les méthodes déjà existantes qui permette de déclencher les notifications.

**3. Intégration des notifications dans l’application :**

* Afficher les notifications dans une zone dédiée de l'interface utilisateur (par exemple, en bas à droite de l'écran).
* Mettre à jour les notifications en fonction des actions de l'utilisateur (Ok, c’est fait , Note refusé , etc.).

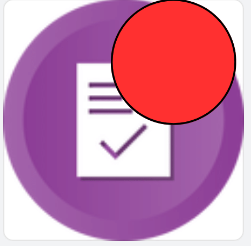
**4. Gestion des notifications pour le récepteur :**

* **Notification d'une nouvelle note:**
  + Afficher un TulPep en bas à droite de l'écran avec l'icône d'une enveloppe violette et le texte "Vous avez reçu une nouvelle note".

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

* + Afficher un petit rond rouge sur l'icône de l’application dans la liste des notes.



* + Masquer le rond rouge et le TulPep une fois que la note a été lue
* **Mise à jour d'une note :**
  + Afficher un TulPep avec le nouveau contenu de la note.
  + Mettre à jour le statut de la note dans la liste des notes.
* **Suppression d'une note :**
  + Avoir la possibilité de supprimer la notification de la note.

**5. Gestion des notifications pour le sender :**

* **Remplacer la notification actuelle par un TulPep :**
  + Afficher un TulPep avec le nouveau statut de la note (. Une image contenant texte, capture d’écran, Police

    Description générée automatiquement
* **Distinguer la notification en fonction du nouveau statut :**
  + Utiliser des couleurs et des icônes différentes pour chaque statut. (Dans la partie développement de notre système de notification , il y a le tableau indicatif pour les types de notifications)
* **Afficher un petit rond rouge sur l'icône de la note dans la liste des notes.**

Une image contenant cercle, Graphique, Caractère coloré, capture d’écran

Description générée automatiquement

* **Masquer le rond rouge et le TulPep une fois que la note a été prise en compte.**

**Résultats obtenus:**

* Système de notifications implémenté pour tous les types de messages.
* Notifications claires et visuellement distinctes pour chaque type de message.
* Système flexible et extensible pour supporter de nouveaux types de messages.

##### II.2.2.2. Développement de la fonctionnalité de statut "En transit"

**Objectif :**

Améliorer la gestion du statut "En transit" des messages pour les rendre plus facilement identifiables et fournir des informations plus claires à l'utilisateur.

**Développement :**

**1. Nouvelle couleur pour "En transit" :**

* Définir une nouvelle couleur mauve clair pour le statut "En transit".



* Appliquer la nouvelle couleur à tous les éléments de l'interface utilisateur qui indiquent le statut "En transit".

**2. Affichage d'informations plus claires sur l'état "En transit" :**

* Afficher la durée pendant laquelle un message est en transit.

**3. Passage au rouge après un certain temps :**

* Définir un délai après lequel un message en transit devient rouge.
* Afficher une notification à l'utilisateur lorsque le message devient rouge.

**Résultats obtenus :**

* Statut "En transit" plus facilement identifiable grâce à la nouvelle couleur.
* Informations plus claires et plus complètes sur l'état "En transit".

##### II.2.2.3. Développement de la fonctionnalité de gestion des listes de diffusion

**Objectif :**

Gérer des listes de diffusion pour permettre l'envoi de messages à un groupe de services en même temps.

**Contraintes :**

* Les listes de diffusion doivent être configurables et modifiables.
* L'interface utilisateur doit être intuitive et facile à utiliser.
* La gestion des listes de diffusion doit être performante et fiable.

**Développement :**

**1. Création d'une liste de diffusion:**

* Définir une interface utilisateur pour la création d'une liste de diffusion.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

**3. Création de boutons dynamiques pour les listes de diffusion:**

* Développer une fonction pour créer des boutons dynamiques dans l'interface utilisateur pour représenter les listes de diffusion.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

* Associer chaque bouton à une liste de diffusion spécifique.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

**Résultats obtenus :**

* Fonctionnalité de gestion des listes de diffusion implémentée.
* Possibilité de créer, modifier et supprimer des listes de diffusion.
* Possibilité d'ajouter et de supprimer des membres d'une liste de diffusion.
* Boutons dynamiques créés pour représenter les listes de diffusion.

##### II.2.2.4. Développement de la fonctionnalité de modification de la hauteur des lignes

**Objectif :**

Modifier la hauteur des lignes du tableau pour améliorer la lisibilité des informations.

**Contraintes :**

* La modification de la hauteur des lignes ne doit pas affecter la performance de l'application.

**Développement :**

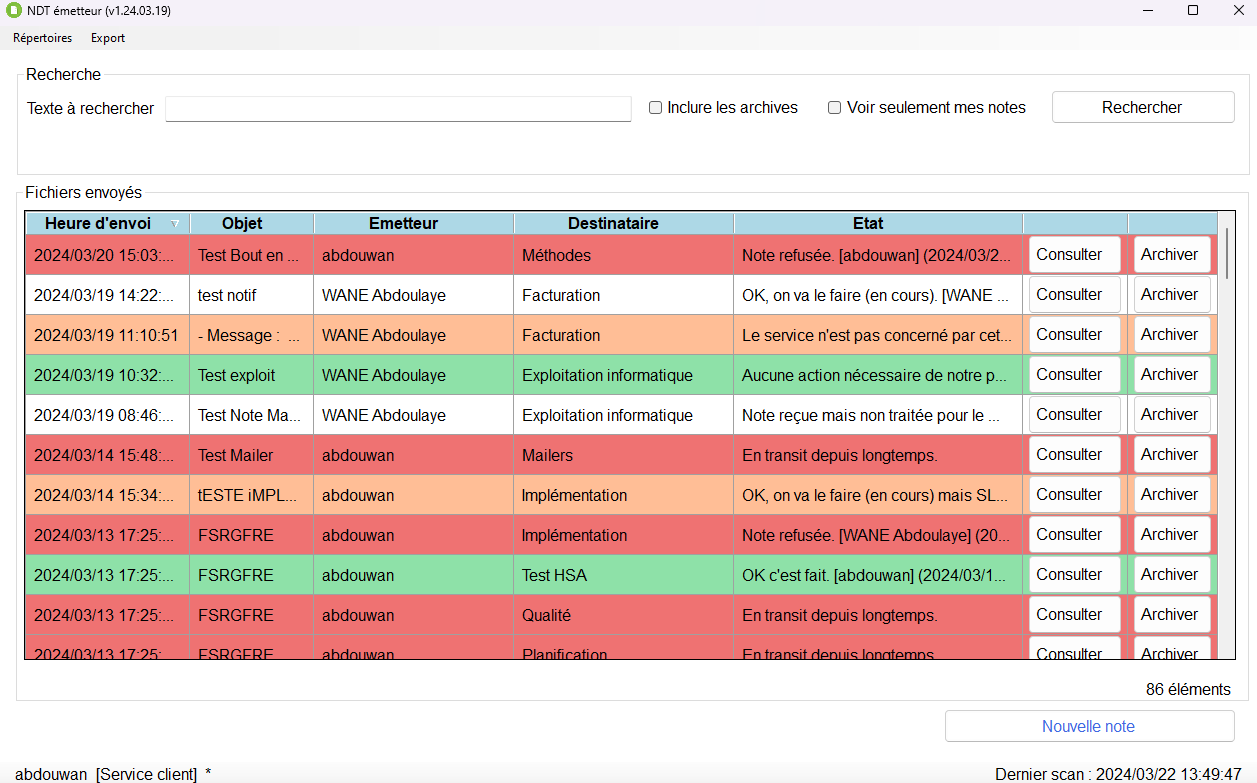
**1. Identification de la propriété des lignes gérant la hauteur :**

* Rechercher dans la documentation Winforms la propriété des lignes qui permet de modifier leur hauteur.

**Résultats obtenus :**

* Fonctionnalité de modification de la hauteur des lignes implémentée.
* Interface utilisateur mise à jour pour refléter les modifications apportées à la hauteur des lignes.

Ci-dessous voici respectivement les nouvelles hauteurs des lignes pour le sender ainsi que le receiver.

 Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

##### II.2.2.5. Développement de la fonctionnalité d'envoi à toute la production

**Objectif :**

Ajouter des options pour envoyer des messages à tout ou partie de la production.

**Développement :**

**1. Ajout de boutons dans l'interface :**

* Ajouter trois boutons dans l'interface utilisateur :
  + "Tout" : pour envoyer le message à tous les services.
  + "Aucun" : pour ne pas envoyer le message à aucun service.
  + "Toute la production" : pour envoyer le message à tous les services de production.

**2. Implémentation de la logique du bouton "Tout" :**

* Développer une fonction pour sélectionner tous les services dans la liste.
* Mettre à jour l'interface utilisateur pour indiquer que tous les services sont sélectionnés.

**3. Implémentation de la logique du bouton "Aucun" :**

* Développer une fonction pour désélectionner tous les services dans la liste.
* Mettre à jour l'interface utilisateur pour indiquer qu'aucun service n'est sélectionné.

**4. Ajout d'une propriété "Production" dans la configuration des services :**

* Ajouter une propriété booléenne "Production" à la structure de données de configuration des services.
* Mettre à jour le code pour charger et enregistrer la propriété "Production".

**5. Implémentation du bouton "Toute la production" :**

* Développer une fonction pour sélectionner tous les services dont la propriété "Production" est vraie.
* Mettre à jour l'interface utilisateur pour indiquer que tous les services de production sont sélectionnés.

**Résultats obtenus :**

* Fonctionnalité d'envoi à tout ou partie de la production implémentée.
* Possibilité de sélectionner tous les services, aucun service ou tous les services de production.
* Interface utilisateur mise à jour pour refléter les sélections effectuées.

##### II.2.2.6 Développement du statut "OK mais pas d'action nécessaire"

**Objectif :**

Ajouter un nouveau statut "OK mais pas d'action nécessaire" pour les fichiers traités.

**Contraintes :**

* Le nouveau statut doit être intégré dans l'énumération EnumFileStatus.
* L'interface utilisateur doit afficher le nouveau statut de manière claire et distincte.
* Le code doit être robuste et performant.

**Développement :**

**1. Ajout du statut "NothingToDo" à l'énumération** EnumFileStatus**:**

* Définir une nouvelle valeur NothingToDo dans l'énumération EnumFileStatus.
* Implémenter la méthode ToString () pour retourner la chaîne "NOTHING TO DO".
* Implémenter la méthode Parse () pour retourner la valeur NothingToDo à partir de la chaîne "NOTHING TO DO".
* Définir la propriété IsFinalStatus à « True » pour le statut NothingToDo.

**2. Affichage du statut "NothingToDo" dans le récepteur :**

* Ajouter le statut NothingToDo à la liste des statuts possibles.
* Pour le statut NothingToDo, afficher l'opérateur, la date et une couleur verte.

**3. Affichage du statut "NothingToDo" dans l'expéditeur :**

* Dans la méthode ElementInDisplayedList, pour le statut NothingToDo, afficher le texte "Aucune action nécessaire", l'opérateur et la date, avec une couleur verte.

**4. Affichage du statut "NothingToDo" dans la visionneuse :**

* Dans l'interface graphique, pour le statut NothingToDo, afficher le texte en vert avec l'opérateur et la date.

**Résultats obtenus :**

* Nouveau statut "NothingToDo" ajouté à l'énumération EnumFileStatus.
* Le statut "NothingToDo" est affiché de manière claire et distincte dans l'interface utilisateur du récepteur, de l'expéditeur et de la visionneuse.

##### II.2.2.7 Développement de la fonctionnalité de retraitement d'une note côté récepteur

**Objectif :**

Permettre à l'utilisateur de retraiter une note côté récepteur, même si le statut de la note n'a pas changé.

**Développement :**

**1. Mise à jour de l'utilisateur et de la date de traitement :**

* Développer une fonction pour mettre à jour l'utilisateur et la date de traitement d'une note.
* Appeler cette fonction lorsque l'utilisateur clique sur le bouton "Retraiter" une note.

**2. Affichage des informations de mise à jour :**

* Mettre à jour l'interface utilisateur pour afficher l'utilisateur et la date de traitement mis à jour.
* Afficher une notification à l'utilisateur pour l'informer que la note a été retraitée.

**Résultats obtenus :**

* Fonctionnalité de retraitement d'une note côté récepteur implémentée.
* L'utilisateur et la date de traitement d'une note peuvent être mis à jour même si le statut de la note n'a pas changé.
* L'interface utilisateur est rafraîchi pour afficher les informations de mise à jour.

## III. Immersion professionnelle

**Alex (mon principale suiveur technique durant le stage) et moi avons effectué une liste des demandes des utilisateurs au sein de l'usine de fabrication des cartes et en office. L'un des volets importants de notre mission était d'étudier l'application NDT (Note Delivery Tool) utilisée pour la communication interne entre le service client et la zone de production.**

**Nous avons eu des échanges avec plusieurs utilisateurs notamment avec Aline (en zone HSA), chef d’atelier au sein de la production, plusieurs points d'amélioration de l'application NDT ont été soulevés :**

### **1. Amélioration de la gestion des notifications :**

* **Cas problématique :** Aline a reçu deux notes avec le même numéro dans la même journée, mais n'a pas vu la seconde. Cela soulève le problème de la distinction entre les notifications et le risque de manquer des informations importantes.
* **Propositions :**
  + **Ajout d'un accusé de réception pour les notes** : le service client serait notifié lorsque la note est lue et validée par le destinataire.
  + **Amélioration de la visibilité des notifications** : les notifications pourraient être plus visibles sur l'interface de l'application.

### **2. Suivi des actions et collaboration entre les équipes :**

* **Problème actuel :** il n'y a pas de moyen de savoir si les chefs d'équipe ont pris connaissance des notes et effectué les actions requises.
* **Propositions :**
  + **Envoi des notes à un groupe de destinataires** : les notes concernant les DHL pourraient être envoyées au standard et non pas uniquement au chef d'équipe conditionnement.

### **3. Ergonomie et accessibilité de l'application :**

* **Problème actuel :** les chefs d'équipe doivent se connecter à un PC spécifique pour consulter les notes, ce qui n'est pas pratique.
* **Proposition :**
  + **Déploiement de l'application NDT sur les postes de travail** : les chefs d'équipe et les opératrices pourraient accéder aux notes directement depuis leur poste de travail.

**En plus de ces points spécifiques, nous avons également relevé des demandes d'amélioration générales :**

* **Ajout d'un champ pour le motif d'un refus** : les opérateurs pourraient expliquer les raisons du refus d'une note.

### **4. Déploiement des nouvelles versions des applications**

**En collaboration avec l'équipe IT, nous avons finalisé le développement des fonctionnalités attendues sur le cahier des charges et avons effectué le déploiement des nouvelles versions de l’applications :**

* **Application NDT :** outil de communication interne pour la transmission d'informations entre le service client et la zone de production.
* **Application KernLogExtractor :** outil d'analyse des fichiers de log des machines Kern et Casillo.

**Le déploiement s'est déroulé en deux phases :**

* **Phase 1 :** installation des nouvelles versions des applications sur les postes de travail en zone Office et en zone HSA.
* **Phase 2 :** tests et validation des nouvelles fonctionnalités par les utilisateurs.

**L'ensemble des opérations s'est déroulé sans incident majeur et les nouvelles versions des applications sont désormais disponibles et opérationnelles.**

**Ce déploiement a permis d'apporter des améliorations significatives en termes de performance, de sécurité et d'ergonomie. Les utilisateurs ont d'ores et déjà exprimé leur satisfaction quant aux nouvelles fonctionnalités et à la facilité d'utilisation des applications.**

**Ce succès est le résultat d'une collaboration étroite entre les équipes métiers et l'équipe IT, et témoigne de la capacité de notre équipe à mener à bien des projets complexes et à répondre aux besoins spécifiques de nos clients.**

## VI. Conclusion et remerciements

Ce stage a été une expérience enrichissante et formatrice qui m'a permis d'acquérir de nouvelles compétences et de découvrir le monde professionnel en profondeur. J'ai pu mettre en pratique mes connaissances théoriques et apprendre de nouvelles techniques dans le domaine de l'analyse et du développement d'applications.

Je suis reconnaissant envers toutes les personnes qui ont contribué à la réussite de mon stage et qui m'ont apporté leur soutien et leurs conseils. Emeline Emeriau, mon maître de stage, pour sa patience, sa pédagogie et son implication dans mon apprentissage. Alexandre Donnet, mon accompagnateur technique, pour sa disponibilité, son expertise et ses précieux conseils. Lysa Bezzon, pour son accueil chaleureux et pour m'avoir permis d'intégrer l'équipe. Aurélie Jouveneau, Mustapha Doudou et Gregory Lamia, pour leur collaboration et leur aide précieuse.

Tous les membres du groupe IDEMIA de Dijon qui m'ont apporté leur soutien et leur encouragement tout au long de mon stage.

Ce stage a été une étape importante dans mon parcours professionnel et je suis convaincu que les compétences acquises et les expériences vécues me seront utiles pour la suite de ma carrière. Je suis reconnaissant envers toutes les personnes qui ont contribué à la réussite de mon stage et je leur adresse mes plus sincères remerciements.