**JOURNAL DE BORD**

**Pour le PORTFOLIO :**

**Présentation :**

**J’ai eu l'opportunité de rejoindre une équipe de l’entreprise Berger-Levrault, BL.Résident, un système de facturation utilisé dans les maisons de retraite. Cette application est notamment une composante de BL.Sénior, un logiciel complet qui intègre également BL.Soins pour le suivi médical et paramédical et BL.Planning pour la gestion des plannings.**

**Mon rôle au sein de l'équipe était principalement axé sur le support aux utilisateurs, en particulier la résolution des problèmes rencontrés, que ce soit des bogues ou des difficultés d'utilisation. J'ai également eu l'occasion de contribuer à quelques évolutions du logiciel. Pendant mon stage, j'ai appris à comprendre le cheminement des plaintes concernant les bogues, ainsi que les processus de résolution de ces problèmes. Cette expérience m'a permis de développer mes compétences en résolution de problèmes, en communication interne et en collaboration au sein d'une équipe multidisciplinaire.**

**Compétences :**

* **Intégration dans une équipe, processus agiles**
* **Réalisation de tests**
* **Corrections d’erreurs mineures (SonarQube)**
* **Débogage directement depuis le web**

**Méthodologie agile :**

* **DSM**
* **Sprint**
* **Sprint Planning**
* **Rétro Sprint**

**Sujets sur lesquels j’ai travaillés : (missions)**

**Exemple de la réalisation d’un ticket sur ? :**

* **Le client annonce un bug au support (N1)**
* **Le support (N1) fait remonter aux personnes plus compétentes réalisant beaucoup de tests (N2)**
* **Les testeurs (N2) créent un ticket pour amener une résolution de ce bug par un développeur (N3)**
* **Le ticket m’ait affecté :**
  + **Prise de connaissance du ticket (évaluation du temps à prendre durant le « Sprint Poker »)**
  + **Reproduction du bug sur l’application**
  + **Recherche de l’emplacement du bug dans le code**
  + **Résolution du bug**
  + **Test de la fonctionnalité**
* **Envoi en « Recette » 🡪 Vérification de la résolution par un membre de l’équipe chargé des tests**
* **Envoi sur la branche de la prochaine mise à jour (environ 2 par ans)**

**Lien du compte-rendu word**

**MISSIONS :**

* **Amélioration des Tests Unitaires :**
  + **Identifier des parties du code qui ont une couverture de test insuffisante. Mettre ensuite en œuvre des tests unitaires supplémentaires pour augmenter la couverture.**
  + **Analyser les tests unitaires existants pour détecter des zones potentielles d'amélioration.**
* **Gestion des Vulnérabilités et des Erreurs SonarQube :**
  + **Analyser les rapports SonarQube pour identifier des vulnérabilités, des erreurs ou des codes enfreignant les bonnes pratiques. Proposer et implémenter des solutions pour corriger ces problèmes.**
* **À la suite du SPRINT du 31/01/2024 :**
  + **Résolution de bugs réels sur la solution (retours de recette + bugs présents) :**
    - **DLOI - Facturation - Gestion des ressources - Ajout - Problèmes de dates**
    - **CJAC - Référentiel - Structures - Duplication Etablissement - Erreur bloquante à l'enregistrement sur environnement BL.senior**
    - **CJAC - Structures - Assistant - Duplication d'unité avec APL - Manque les lignes de type "Aide au logement" (APL, ALS, ALF)**
    - **DLOI - Facturation - Emission de facture - Facture et frais annexes**

**Responsable :**

* **Bastien JACOUD** (présent à l’entreprise le mercredi et vendredi)
  + **Mail :** Bastien.Jacoud@berger-levrault.com

**Sonar :**

[1011872 Correction SonarQube - SANTE - Confluence (berger-levrault.com)](https://projetbl.berger-levrault.com/confluence/display/SAN/1011872+Correction+SonarQube)

**Tests Unitaires :**

[1011871 Mise à niveau des TU - SANTE - Confluence (berger-levrault.com)](https://projetbl.berger-levrault.com/confluence/pages/viewpage.action?pageId=136816987)

**Faux positifs :**

**Compteurprecedent + compteurAArchiver**

EditionMSDASDueManager



EcritureComptableManager



ClotureManager

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

Description générée automatiquement

AbstractImportExportAplManager

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

Description générée automatiquement

IntegrationFactureInteropResultDTO (Impossible de changer le setter car serialize)

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

LotNumerotation

ToFuture() au lieu de ToList()

// BLException

INVALIDOPERATIONEXCEPTION

MANAGEEXCEPTION

Com sur integrationfactureinteropResult

// Sonarqube – faux positif

**RECETTE :**

Les modifs sonar pour recette :

* BL.Sante.Usager.Editions.Managers/OpenXml/EditionListeAttenteManager.cs L193
* BL.Sante.Usager.Models/Helper/MouvementFacturationHelper.cs L928
* BL.Sante.Usager.Models.Aurelia/usagers/models/helper/indicateurHelper.ts L30
* BL.Sante.Usager.UseCases/Interop/UsagerSanteMSInteropUseCase.cs L1050
* Encaissementdecaissementhelper.ts bug sonar à tester
  + Test de non regression sur l’encaissement mspa des factures usagers avec la gestion des acomptes
* MouvementFacturationHelper.cs
  + Test de non regression

**Question pour hafiz :**

Ligne 1639 sur récup d’une date qui peut être null

Ecriturecomptable

+ Qu’est-ce que les packages ?

+ Lancement du webhost

Abstractimportexportaplmanager 🡪 vérif faux positifs

Faire les if (exception) 🡪 après réaliser les vérifications

C:\DEV\Sante.Portail\DevArea\Sources\BL.Core.Portail\BL.Sante.Web>

cmd.exe /c npm run BL.Dev --color=always

PRESENTATION

Finance.Managers/EncaissementDecaissement/Clotures/ClotureManager.cs Ligne195

TESTS TERMINES, A TESTER

Archivage/IArchivageInfoUsagerDataContract

CertDC/IEnvoiCertDCDataContract

Contributions/IDetailContributionDataContract

Documents/IDocumentDestinataireDataContract

Documents/IDocumentGenereDataContract

Documents/IDocumentMssEnvoyerDataContract

Documents/IFichierUsagerDataContract

TESTS EN COURS

DossierUniqueUsagers/IPreAdmissionDataContract

08/01/2024

* Arrivée aux locaux
* DSM (Daily Scrum Meeting) à 9h30 pendant 20 minutes :
  + Présentations aux collaborateurs
  + Echanges sur ce que l’on à fait la journée précédente et sur ce que l’on va faire dans la journée
* Installation de l’environnement de travail (VS, extensions…)

09/01/2024

* DSM :
  + J’ai fait :
    - Toute l’installation de l’environnement
    - Un ticket GLPI qui va être accepté
  + Je vais faire :
    - Voir avec Bastien Jacoud ce qu’il y a à faire
* Exercices C# pour la reprise en main du langage
* Finalisation de l’environnement, accès aux projets (manque encore des accès suite à une erreur d’installation10/01/2024

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

Description générée automatiquement10/01/2024

- DSM remplacé par la réunion de sprint (Planification sur 3 semaines) :

* + Explications sur le fonctionnement :
    - Chef de produit dit pourquoi
    - Les développeurs disent comment
  + Rappel pour les développeurs par Bastien :
    - Sur les temps de développements
    - Sur les besoins des tests
    - Suivi de la DOD (Definition Of Done)
      * Une image contenant texte, capture d’écran, Police, nombre

        Description générée automatiquement
    - Suivi de la DOR (Definition Of Ready)
      * Une image contenant texte, capture d’écran, Police, document

        Description générée automatiquement
    - Gestion des commentaires et des statuts :
      * Commenter les tickets
      * Documenter les améliorations et blocages en temps réel pour faciliter la compréhension du développement
  + A tour de rôles, les collaborateurs disent ce qui ne va pas et montrent des corrections...
  + Conversations sur les méthodes de travail (agilité) :
    - Demandes d’ateliers conceptions
    - Demande d’ajout d’un producteur de ticket à la réunion pour lui montrer les besoins lors d’un envoi du ticket (besoin de précisions pour limiter les risques de problèmes externes)
* QUESTIONS :
  + Le DOD c’est tous les commentaires et tests qui montre que le développement est fonctionnel ? (Ce qu’il y a en dessous des tickets ?)
  + RECETTE/RETOUR/AUTRES RECITS dans l’excel (à quoi ça sert ?)
  + Différence entre usagers/patients/résidents ?
  + Mettre en place le pair programming ?

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

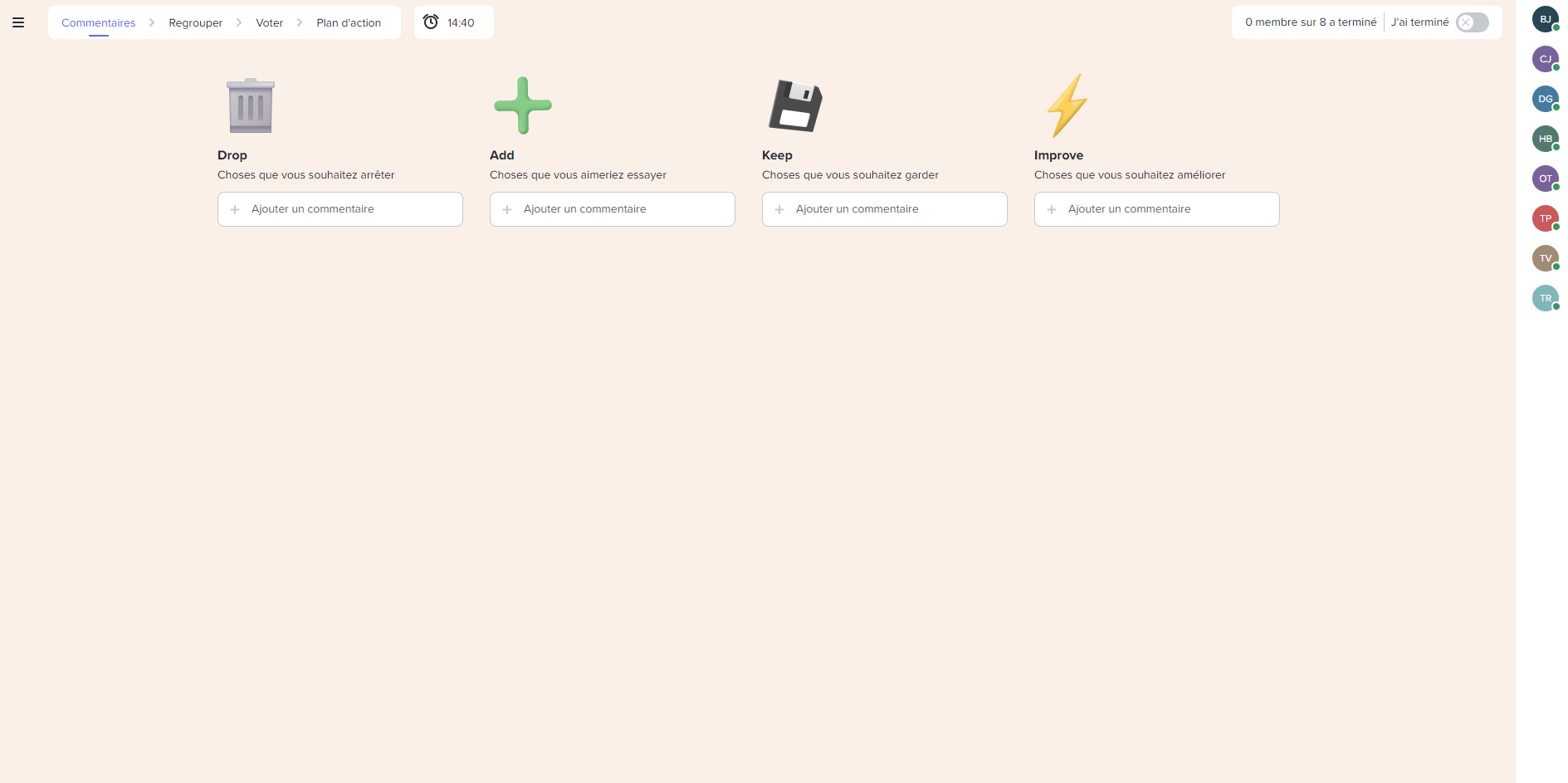
Description générée automatiquement

(Planning des réunions du mois)

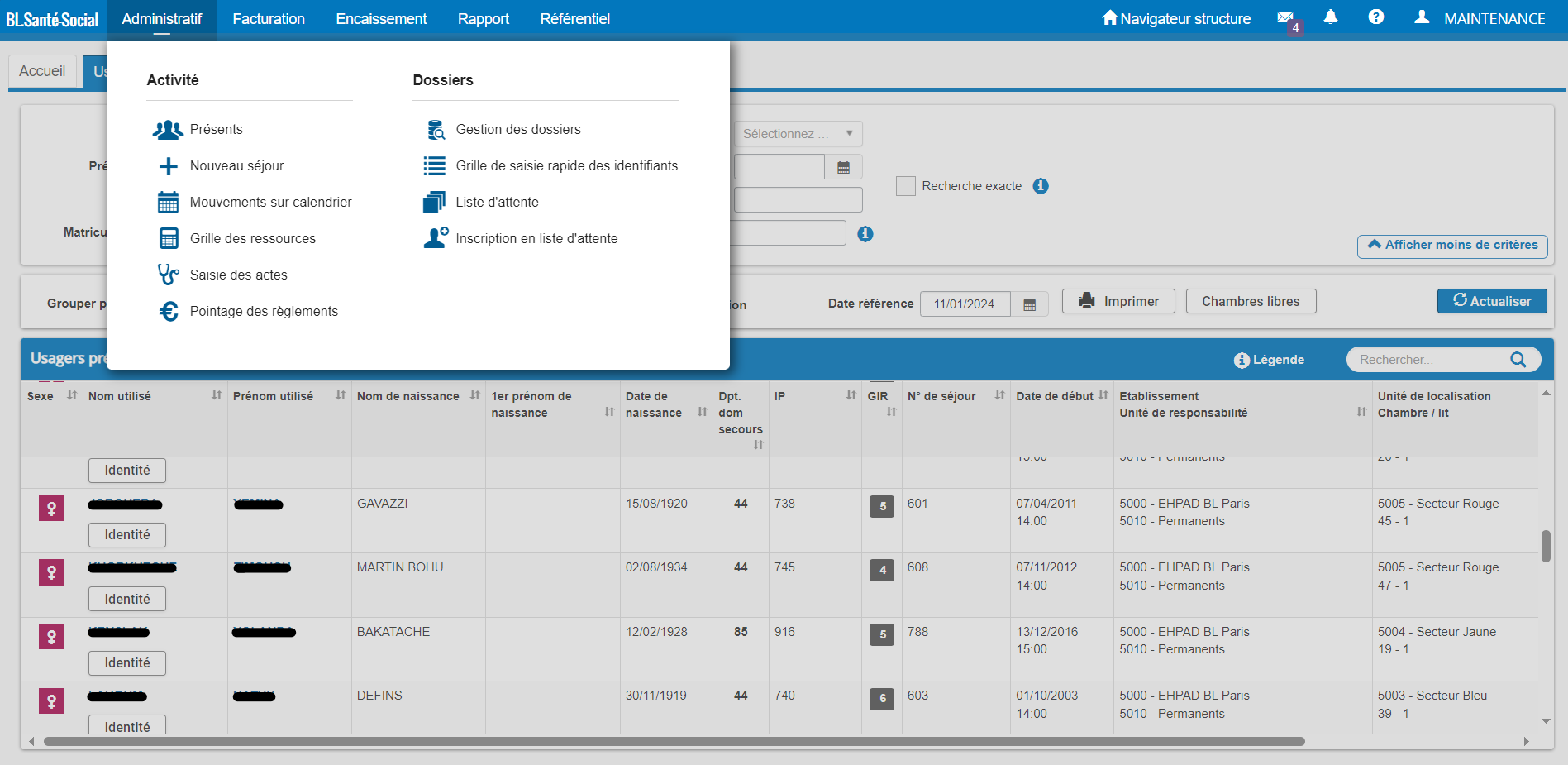
* Poker Planning (planification des prochains sprints) :
  + Présentation de chaque ticket
  + Estimation du temps à passer sur chaque tâche via un vote de chacun (devs) après présentation (en jours ou en points d’efforts)
  + Pour moi (TU pdt 1,5j et Correctifs SONAR pdt 3j) en trinôme avec Youssef et Azze Edine 🡪 Présentation en fin de réunion de ces aspects
* Maj terminée de BL Resident, compte dotnet remis en place avec le MD5

11/01/2024

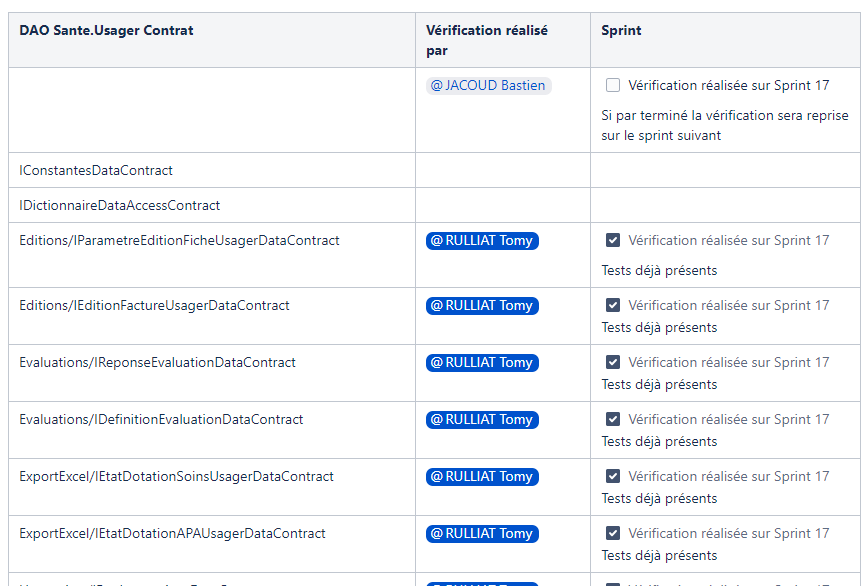
* Lecture d’un mail envoyé sur le retour du sprint avec tous les éléments discutés
* DSM :
  + Comme la **dernière** journée à été faite de réunion, rien n’a été fait, clarification sur quelques aspects du sprint
* Rétro Sprint :
  + Outil Neatro
  + Passer en revue ce qui a bien fonctionné lors du sprint qui vient de se terminer
  + Identifier les axes d’amélioration, élaborer un plan pour les améliorations
  + Elaborer un plan pour les améliorations
  + Obtenir un retour d’information
  + Chaque collaborateur complète les 4 cases, vote de ce qu’il y a de plus important puis bilan final avec plan d’action (voir ci-dessous)

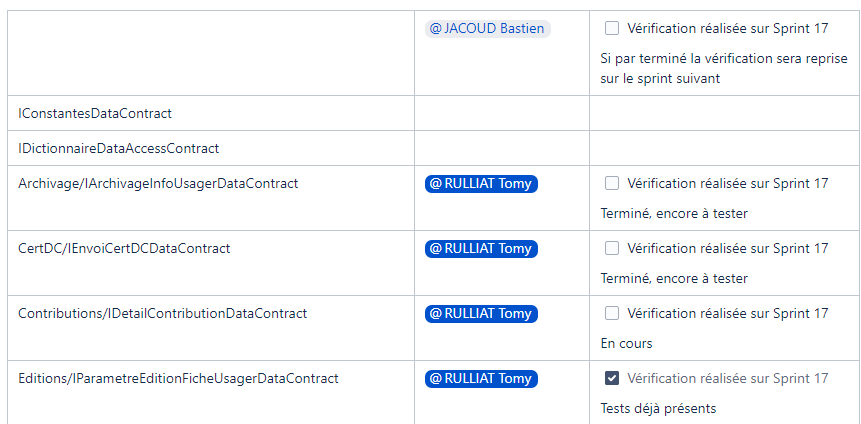




* Accès à l’environnement de développement grâce à Hafiz (BDD avec accès au site)

12/01/2024

* DSM :
  + J’ai fait :
    - Environnement git et tfs
  + Je vais faire :
    - Attente du ticket GLPI de gaetan pour l’accès à la base de test
    - Finalisation du git
* Besoin de faire remonter les commits sur 2 environnements car période de transition :
  + PFS (Désir de ne plus l’utiliser)
  + GitLab (Nouveau à l’entreprise)
* Pour TFS (besoin de connexion à la base de test : Ticket GLPI par Gaetan REVOL)
* Pour Git (Besoin d’accès au Nexus.forge.berger-levrault.com)
* Mon travail sera de créer et réaliser un grand nombre de tests manquants sur toutes les méthodes de la solution de la section « Usager » dans un premier temps
  + Comme je n’ai pas les accès et donc que je ne peux rien créer, mon premier travail est de repérer les éléments de tests déjà effectués.
  + 
* J’ai commencé l’écriture de tests (AVEC FLUENT ASSERTIONS)
  + Une image contenant texte, capture d’écran, logiciel, Police

    Description générée automatiquement
  + 

15/01/2024

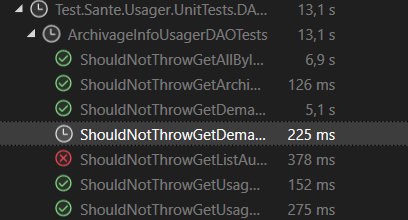
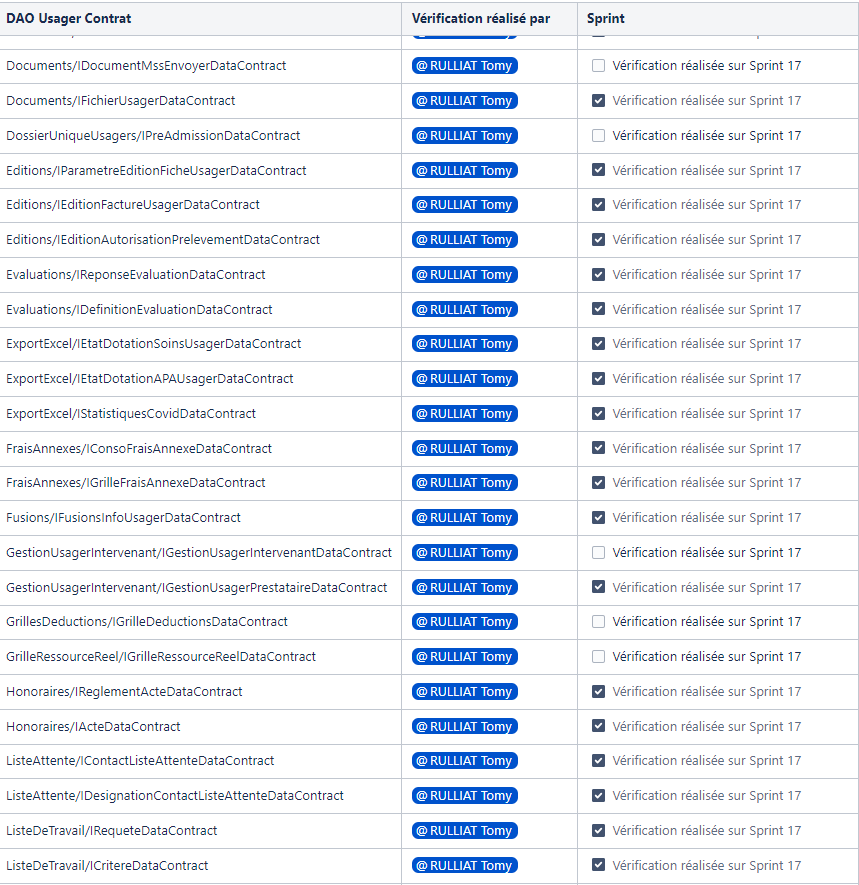
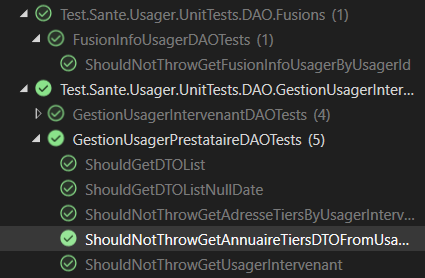
* DSM :
  + J’ai fait :
    - Vérification des DAO déjà testés de Sante.Usager
    - Début d’écriture de tests
    - Ticket GLPI bien créé par Gaetan pr l’accès à la base de tests 🡪 en attente
    - Découverte de méthodes SAVE et DELETE dans les DAO (fait-on des tests)
      * Exemple Dans Sante.Usager/Synthese/DroitSocial/DroitSocialDAO.cs
      * Une image contenant texte, capture d’écran, Police

        Description générée automatiquement
      * REPONSE : En discuter demain avec laurent à la réunion spécialisée en tests mais dans l’idée oui, il faut tout tester, à voir pour tester SAVE et DELETE en même temps
    - Lorsque l’on a des méthodes privées, doit-on les tester ? car l’on ne peut pas n’étant pas renseigner dans les DataContracts
      * Exemple dans Sante.Usager/Data.Providers(ou Contracts)/Archivage…
  + Je vais faire :
    - Continuer les tests + les vérifier en fonction des accès
* Mise à jour des packages nugget pour GIT
* Une image contenant texte, capture d’écran, Police

  Description générée automatiquement
* Problème de connexion lors du lancement des tests malgré les accès :
  + La méthode d'initialisation Test.Sante.Usager.UnitTests.DAO.Documents.FichierUsagerDAOTests.TestInitialize a levé une exception. Spring.Objects.Factory.ObjectCreationException : Spring.Objects.Factory.ObjectCreationException: Error thrown by a dependency of object 'Spring.Transaction.Interceptor.TransactionInterceptor#0' : Initialization of object failed : 28000: aucune entr�e dans pg\_hba.conf pour l'h�te � 10.43.100.19 �, utilisateur � dotnet �,base de donn�es � blsante �, SSL inactif while resolving 'transactionManager' to 'transactionManager' defined in 'assembly [BL.Core.Fwt.Spring.Nhibernate, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null], resource [BL.Core.Fwt.Spring.Nhibernate.ApplicationContext-Sante.xml] line 48' ---> Npgsql.PostgresException: 28000: aucune entr�e dans pg\_hba.conf pour l'h�te � 10.43.100.19 �, utilisateur � dotnet �,base de donn�es � blsante �, SSL inactif.
* Réalisation d’une note personnelle pour ne pas me perdre :  
  Une image contenant texte, capture d’écran, Police

  Description générée automatiquement

16/01/2024

* DSM :
  + J’ai fait :
    - Continuer les tests
    - Bloqué sur un problème de lancement des tests malgré l’accès à la base de test 🡪 Hafiz
  + Je vais faire :
    - La réu puis écriture de tests (+ commencer à regarder SONAR)
* Réunion avec Laurent sur les TU et SONAR :
  + Questions :
    - Les méthodes SAVE, UPDATE et DELETE
  + Infos :
* Réunion avec David pour ces soucis de connexion à la base de test 🡪 rien trouvé
* Tests possibles, début de la vérification :
  + 
  + 
  + 

17/01/2024

* DSM :
  + J’ai fait :
    - Les tests peuvent enfin être vérifiés donc c’est ce que j’ai fait
  + Je vais faire :
    - Correction de tests qui ne sont pas passés
    - Peut-être commencer à regarder du Sonar ?
* Une partie de ce que j’ai modifié pour l’instant :
  + 
* Exemple de test que j’ai créé:
  + Une image contenant texte, capture d’écran, logiciel

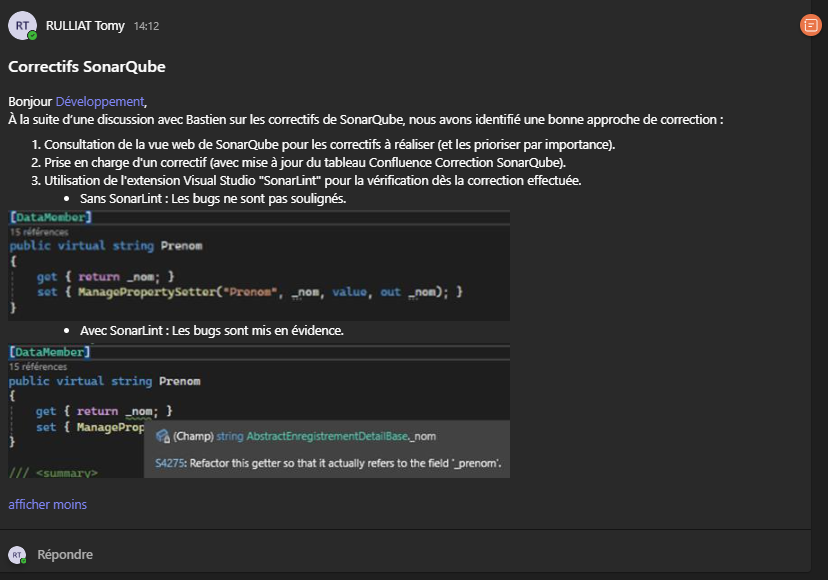
    Description générée automatiquement
  + Une image contenant texte, capture d’écran

    Description générée automatiquement
  + Une image contenant texte, capture d’écran, Police

    Description générée automatiquement
  + Maintenance/DroitPortabiliteRGPDDAOTests entre autres réalisé à 100%

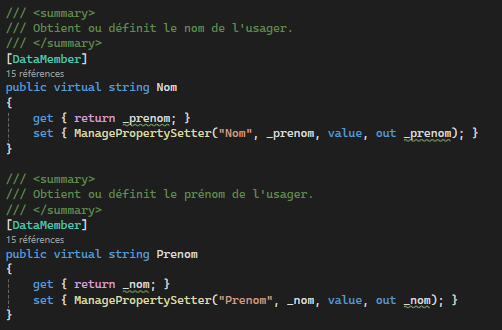
18/01/2024

* DSM :
  + J’ai fait :
    - Les TU peuvent enfin être vérifiés
    - Continuer les tests
  + Je vais faire :
    - Continuer les tests
* Questions pour hafiz :
  + Générer un package / archiver les tests
  + Voir pour l’exécution sous git
  + Sonar ? Il n’y a plus rien ? [Projects (berger-levrault.com)](https://sonarqube.forge.berger-levrault.com/projects?search=pfs-) 🡪 RESOLU
* Documentation GitFlow
* Découverte de SonarLint pour les correctifs SONARQUBE :
  + Une image contenant texte, capture d’écran, Police

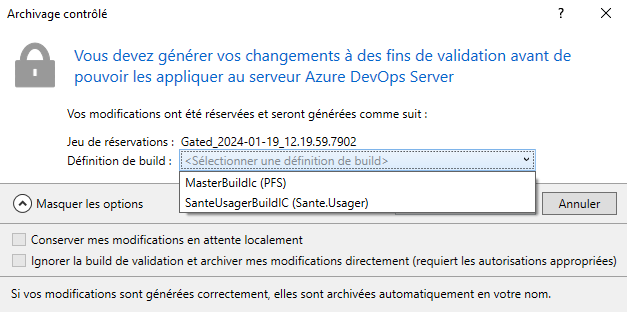
    Description générée automatiquement
* Diffusion d’information sur le Teams des développeurs sur la démarche de correction de bugs SonarQube
* 

19/01/2024

* Entrevue avec Bastien
  + Saisie des temps
  + Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

    Description générée automatiquement
* DSM :
  + J’ai fait :
    - Tests
    - Début de SONAR
  + Je vais faire :
    - Voir avec Bastien pour la saisie des temps et d’autres questions
    - Correctifs SonarQube
* Discussion avec hafiz
  + Aide sur l’archivage des tests
  + Les tests sur GIT
  + TOUJOURS vérifier les fichiers avant d’archiver
  + Explication des FL :
    - 1. Le client à un problème
    - 2. Le support n’a pas la réponse
    - 3. L’analyste metier n’a toujours pas la réponse 🡪 Affecte un ticket aux développeurs
    - 4. Les développeurs corrigent le bug
* Bug SonarQube :
  + 
* Archivage des tests et du bug Sonar après vérification des changements :
  + Une image contenant texte, capture d’écran, logiciel

    Description générée automatiquement
  + Une image contenant texte, capture d’écran, Police, ligne

    Description générée automatiquement
  + 
  + Vérification des tests après archivage au cas où
    - Une image contenant texte, capture d’écran, Police, logiciel

      Description générée automatiquement
* Corrections SonarQube :

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

Description générée automatiquement

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

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

22/01/2024

* DSM :
  + J’ai fait :
    - Archivage des TU
    - Correctifs sonar pendant une demie-journée
  + Je vais faire :
    - Archivage via Git des correctifs Sonar et des TU
    - Correctifs SonarQube
    - Tickets pour les tests Git
* Les tests peuvent maintenant être exécutés sur Git
* Vérifier les tests
* Aufaite, pour les BDD, utilisation du logiciel DBeaver :
  + Une image contenant texte, capture d’écran, nombre, Police

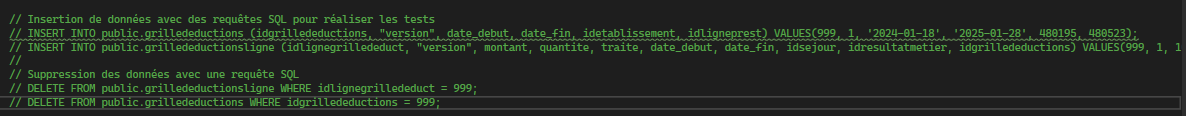
    Description générée automatiquement
  + Une image contenant texte, capture d’écran, logiciel, nombre

    Description générée automatiquement

23/01/2024

* DSM :
  + J’ai fait :
    - Vérification et archivage des TU sur Git
    - Correctifs Sonar
  + Questions :
    - Le SonarQube réalise une vérification quand ? la dernière était vendredi dernier (Réponse : à chaque build 🡪 Mes builds Git n’étaient donc pas passés)
    - Accès aux projets SonarQube
  + Je vais faire :
    - Journée Sonar encore
* Transports moins long (40min au lieu de plus d’1h)

24/01/2024

* DSM :
  + Point d’équipe :
    - Le travail du développeur ne s’arrête pas qu’au développement :
      * Prendre du temps pour voir avec le recetteur
      * S’inquiéter des échéances
      * S’organiser
  + J’ai fait :
    - Sonar
    - TU parce que je n’avais pas les accès aux autres projets Sonar
  + Je vais faire :
    - SonarQube
* COMMENTAIRES PR L’EXECUTION DES TU
* 

25/01/2024

* Vérifier correctifs Sonar sur l’appli
* DSM :
  + J’ai fait :
    - Correctifs SONAR
    - Tests unitaires
  + Je vais faire :
    - Branches Git tout au propre (Tout vérifier)
    - Correctifs SONAR
  + Question :
    - Question fonctionnelle git 🡪 quand fait-on les Merge Request pour des branches de TU ou de Sonar (Fin du jour, fin du sprint ?) et est-ce qu’on coche le fait que la branche soit supprimée quand c’est accepté pour des branches de TU ou de Sonar ?
* Mise en place d’une branche feature pour les résolutions que je fais :
* Une image contenant texte, capture d’écran, Police

  Description générée automatiquement
  + Mes commits se feront sur cette branche 🡪 Lorsque tout est vérifier, envoie d’une « Merge Request » 🡪 Vérification par un tiers puis envoi sur la branche Develop 🡪 Lorsque tous les devs ont tout envoyé sur la branche Develop et que tout fonctionne, envoi sur la branche Main (Branche destiné à la version Client Finale)

26/01/2024

* MARK AS READY 🡪 Revue de code (1h) avec David
* DSM :
  + J’ai fait :
    - Continuer un peu les TU
    - Corrections Sonar, j’ai vu aussi ce matin avec David à la suite de la MR
  + Je vais faire :
    - Corrections Sonar (Ce que j’ai déjà fait + nouveautés)
    - Tester sur l’application web
* PB de faux positifs SonarQube
* 
* Ceci met en erreur car parametreEdition peut être null alors que :
* 
* J’ai appris comment correctement renvoyer une erreur (pour les boucles, l’effectuer avant la boucle pour ne pas charger le programme) :
* 

29/01/2024

* DSM :
  + J’ai fait :
    - Corrections Sonar Sante.Finance, on a vu avec David 🡪 on va enfin pouvoir Merge sur la branche develop
  + Je vais faire :
    - J’ai terminé mon sprint 🡪 Repartir sur des TU
* Participants du DSM :
* Une image contenant texte, capture d’écran

  Description générée automatiquement

30/01/2024

* DSM :
  + J’ai fait :
    - Finaliser la Merge des correctifs Sonar avec David
    - TU de Sante.Usager
  + Je vais faire :
    - TU de Sante.Usager 🡪 Que je vais finaliser aujourd’hui.
* Changements des TU :
  + 
* Préparation d’une petite présentation de mon travail en vue des réunions de Sprint du 31/01 et 01/02 :
  + TU :
    - Les tests unitaires de la solution Sante.Usagers tous effectués :  
      [1011871 Mise à niveau des TU - SANTE - Confluence (berger-levrault.com)](https://projetbl.berger-levrault.com/confluence/pages/viewpage.action?pageId=136816987)
    - Exemple :
      * Voir (TU/Fonction) 🡪 Environ 250 tests unitaires créés
  + SONAR :
    - Les correctifs de bugs SonarQube de la solution Sante.Finance (passage de 45 bugs à seulement 19 🡪 Correction de 26 bugs dont beaucoup de faux positifs)  
      [1011872 Correction SonarQube - SANTE - Confluence (berger-levrault.com)](https://projetbl.berger-levrault.com/confluence/pages/viewpage.action?spaceKey=SAN&title=1011872+Correction+SonarQube)
    - Explications :
      * Ces corrections n’affectent pas le fonctionnement du code mais :   
        1. Elles permettent une meilleure lisibilité et compréhension  
        2. Elles permettent d’éviter de possibles prochains bugs  
        3. Elles permettent de mettre à jour les anciennes pratiques qui ne se font plus
    - Exemple :
      * Voir (Before/After) Lorsque l’on modifiait le prenom, il modifiait la variable \_nom

31/01/2024

* Nouveau SPRINT :
  + Une image contenant texte, ligne, Police, nombre

    Description générée automatiquement
* DSM remplacé par la réunion de SPRINT (3h) :
  + Point sur l’organisation de l’équipe (sprint plus long : 4 semaines en raison de vacances)
  + Présentation rapide de mes travaux (sans montrer de code)
    - Discussion autour de ces travaux : besoins d’envoyer en RECETTE (test) pour certaines corrections Sonar 🡪 Quand il y a de réels changements sur les méthodes
* Réalisation des corrections de mes tests unitaires à la suite de la revue de code avec David GRELLETY
* Réunion d’après-midi POKER PLANNING (2h30) :
  + Présentation des tickets de chacun + estimation de la durée de réalisation 🡪 Mes tickets :
    - BLRES-11055 : Déjà terminé par Thomas PADILLA (pas d’estimation)
    - BLRES-11271 : Erreur lors de la duplication d’un établissement 🡪 L’ajout est bloqué (estimé 1 jour)
    - BLRES-11082 : Lors de la duplication d’un établissement, Lorsque "Aide au logement" est cochée, les lignes de prestation de sous-type "APL", "ALS", "ALF" sont absentes (estimé 3 jours)
    - BLRES-11276 : Erreur dans le cumul de frais annexes, lorsque deux frais annexes similaires avec un prix différent 🡪 en mode détaillé, une seule ligne s’affiche au lieu de deux (estimé 1 jour) voir avec azze edine
* Estimation en collaboration :
  + Une image contenant texte, capture d’écran, Police, logiciel

    Description générée automatiquement

01/02/2024

* DSM :
  + J’ai fait :
    - Réunions
    - Merge Request des TU sante.usager acceptée et passée en branche Develop



* + Je vais faire :
    - Réunion Rétro Sprint
    - Reprise des TU de sante.finance
* Reprise des TU Sante.Finance
  + Pour l’instant :
    - 
  + Je commence par une mise de tous les fichiers dans le tableau pour ne pas en oublier :
    - Une image contenant texte, capture d’écran, nombre, document

      Description générée automatiquement
* Changement du fichier de SPRINT (MAJ des jours consacrés à chaque tâches) suit au Poker Planning du 31/01 :

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

Description générée automatiquement

* Rétro SPRINT avec l’outil NEATRO :
  + Une image contenant texte, capture d’écran, Police

    Description générée automatiquement
  + Etablissement des actions à prévoir :

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

Description générée automatiquement

02/02/2024

* DSM :
  + J’ai fait :
    - Réunion Rétro SPRINT
    - Reprise des TU de sante.finance
  + Je vais faire :
    - Faire ma journée TU de sante.finance

05/02/2024

* DSM :
  + J’ai fait :
    - Sprint TU terminé, il ne me manque plus qu’à voir la merge request et ce sera OK
  + Je vais faire :
    - (Répété au-dessus) Merge Request des TU sante.finance
    - Débuter les correctifs Sonar
* Une image contenant texte, capture d’écran, Police, nombre

  Description générée automatiquement
* Une image contenant texte, capture d’écran, Police, ligne

  Description générée automatiquement
* Une image contenant texte, capture d’écran, Police, nombre

  Description générée automatiquement
* Lors de la recherche de projet sur SonarQube, je ne vois pas tous les projets BL.Résident 🡪 Envoi d’un mail à l’administrateur des droits
* Une image contenant texte, capture d’écran

  Description générée automatiquement
* Découverte d’une fonction qui ne semble plus être utilisée depuis 2013 🡪 Re vérification par un autre utilisateur 🡪 Démarche de suppression du fichier

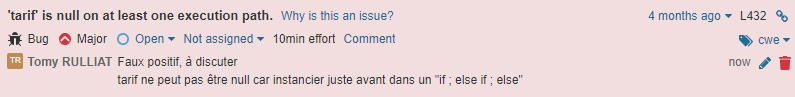
Fonction helper invalidoperationexception

Get {valueforserialisation ;}

06/02/2024

* DSM :
  + Question bloquante :
    - Que fait-on des faux positifs SonarQube ? Est-ce que c’est bien à cause de la version trop ancienne ? Et si oui, est-ce qu’on va l’avoir bientôt ?
    - Réponse : Il y a des droits (que l’on n’a pas encore) pour signaler les faux positifs afin qu’ils ne soient plus référencés dans SonarQube
  + J’ai fait :
    - Correctifs Sonar et Merge Request
  + Je vais faire :
    - Continuer sur SonarQube
* Encore beaucoup de faux positifs :
* Une image contenant texte, Police, capture d’écran

  Description générée automatiquement
* Une image contenant texte, capture d’écran, Police

  Description générée automatiquement
* 

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

Description générée automatiquement

Passage de 5 à 6 pour créer une nouvelle maj pour les packages nugget (incrémenter le nugget)

07/02/2024

* DSM :
  + J’ai fait :
    - Correctifs Sonar Sante.fwm et Sante.Usager
  + Je vais faire :
    - Terminé ce que je peux faire au niveau des correctifs sonar
    - Corriger les merge request
    - Corriger un petit test
    - Corriger l’accès au web
* Le site ne se lançait pas, après avoir vu avec David, tout refonctionne :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* Pour les MERGE REQUEST :
  + voir les security spots (http 🡪 pas grave si ?)
  + <https://sonarqube.forge.berger-levrault.com/project/issues?id=PFS-BL.Sante.Finance&issues=AYsjvfoB5dm-NEr-iLHe&open=AYsjvfoB5dm-NEr-iLHe>  
    qu’en faire ?
* Changement de bdd pour avoir de la matière pour la future correction des tests :

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

Description générée automatiquement

08/02/2024

* DSM :
  + J’ai fait :
    - Correctifs Sonar
  + Je vais faire :
    - Terminé les correctifs Sonar
    - Merge request
    - Peut-être commencer le ticket BLRES-11271
* Editionlisteattentemanager 🡪 colomn += 5
* Grille deduction 🡪 faut tester

Y’a-t-il une possibilité que civilite soit null, et sinon si c’est possible, doit-on initialiser code à 0 ducoup ?

Est-ce qu’on bloque si civilite est null ou alors on lance un autre code (0 ?)

Réponse : demander à hafiz car peut-être utilisée dans une autre solution

Teams avec hafiz sur :

* + Suppression ou non d’une fonction
  + Un bug sonarQube qui peut affecter d’autres projets
  + A quoi sert une ligne
  + En commençant les FL, comment savoir où se trouve l’erreur dans le code ?
  + (S’il y a du temps) 🡪 Un peu d’aide pour réaliser le portail web via git (car je n’ai que TFS pour l’instant)

09/02/2024

* DSM :
  + J’ai fait :
    - Fin du sprint de correctifs Sonar
    - Passer du temps sur les merge request
  + Je vais faire :
    - Le ticket BLRES-11271 🡪 Si un dev à un peu de temps pour me montrer comment retrouver le bug dans toute l’architecture et peut m’aiguiller un peu
* Aujourd’hui, je m’attaque à un vrai bug, BLRES-11271 :
  + Lors de la duplication d’un établissement (déjà existant) dans une entreprise (déjà existante), un message d’erreur s’affiche :

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

Description générée automatiquement

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

Description générée automatiquement

Une image contenant texte, capture d’écran, Page web, Police

Description générée automatiquement

J’ai appris comment fonctionne le support dans l’équipe :

* N1 (téléphone) essayer de comprendre les besoins du client et de les dépanner quand ce n’est pas trop compliqué
* N2 avec les infos du N1, voir s’il est possible de faire quelque chose d’un peu plus technique
* N3 (Support BL.Résident) Récupération de la base du client sur un poste local pour refaire le bug et pour le corriger (correction via le code, appel au client pour un changement au niveau de la base ou pour des problèmes type UPDATER, soucis de versions…)

Résolution du ticket :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

(Ajout de la dernière ligne)

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

Description générée automatiquement

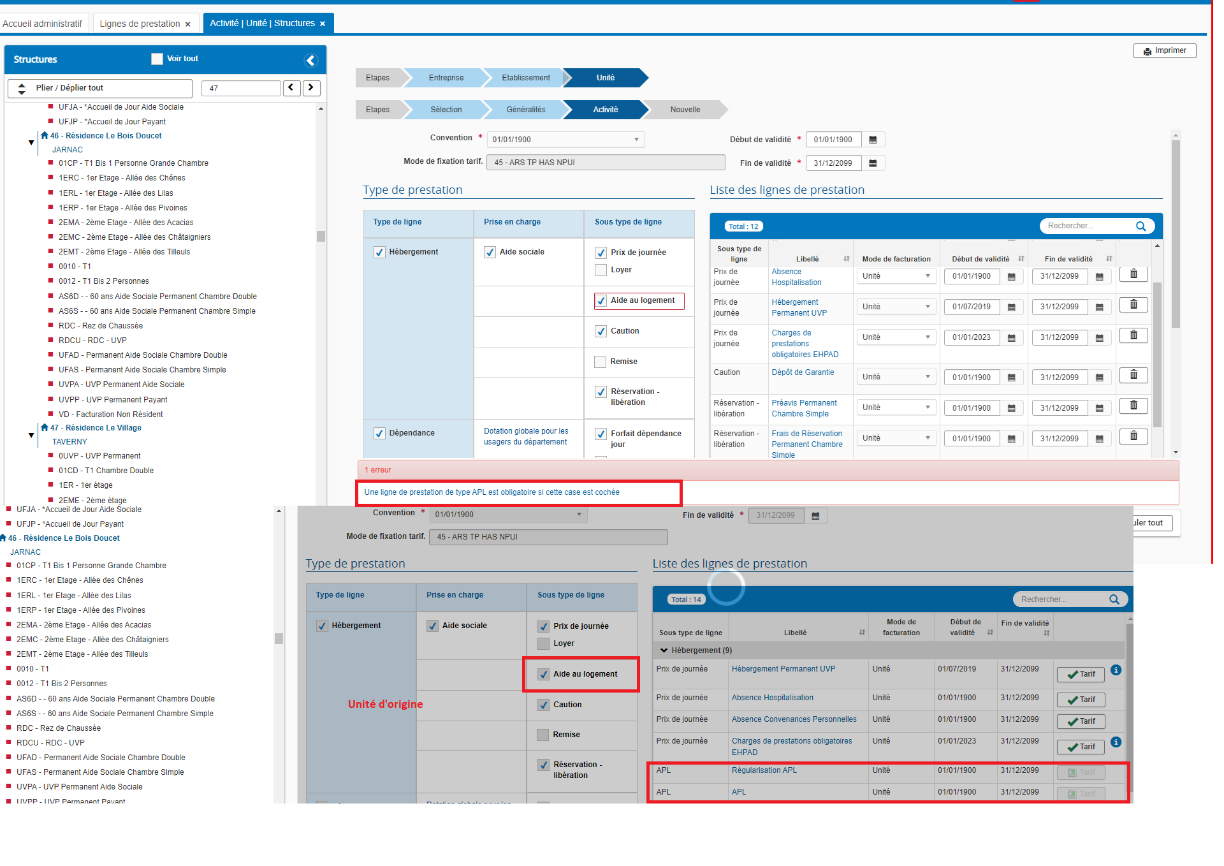
(Ajout du secteuractivite)

Lors du ticket, je tombe sur une autre erreur :  
Une image contenant texte, capture d’écran, Police, Page web

Description générée automatiquement

J’ai appris à débuger une appli web avec l’outil de développement + « Sources »

12/02/2024

* DSM :
  + J’ai fait :
    - Réalisation du ticket BLRES-11271 avec l’aide de David,mais lorsque je l’ai corrigé, je suis tombé sur une erreur côté GIT sur des mappings qui ne correspondaient pas, j’ai demandé à Youssef OABI et j’attends sa correction
  + Je vais faire :
    - Attaquer le ticket BLRES-11276 🡪 Azze-Edine, je crois que tu as travaillé sur la duplication des lignes de prestations de sous-type APA (si tu vois de quoi je parle, la FL SANBLRESFL-2560) 🡪 Je dois faire la même chose pour les lignes de prestations APL, si tu as un peu de temps pour m’aider
* Le ticket BLRES-11276 :
  + 
  + Une image contenant texte, Police, capture d’écran

    Description générée automatiquement
  + Pour info, j’ai des aides APL + APA sur l’unité « EN01 – RESIDENCES … / 1000 – EPHAD BL LYON / 1003 – Blordiere B »
* Une image contenant texte, capture d’écran, nombre, logiciel

  Description générée automatiquement
* Pour les aides sociales APL, elles ne se dupliquent pas quand il y a « prise en charge Aide Sociale » de coché en mspu
* En mspa, celles décochés s’affichent quand c’est coché, et inversement
* Question, comment corriger ? comment savoir si un etablissement est pa ou pu

En pu, pas de mode de calcul, tjr ressource deduite de l’aide sociale (forcément prise en charge, enlevé si décocher)

En pa, 2 choix, ressource aide sociale et ressource deduite de l’aide sociale (toujours le copier)

Dupliquer si c’est cohérent avec la nouvelle unité

Ne pas dupliquer si l’établissement ne les accepte pas

* Réalisation de présentation de code pour expliquer les avancés, les corrections effectuées

Débogage : Une image contenant texte, logiciel, Icône d’ordinateur, Page web

Description générée automatiquement

Pour refresh l’application WEB, prendre les dll (bl.sante.Fwm.Managers/bin/debug/net48) et le mettre dans (BL.Sante.Web.Host/bin)

Ticket résolu 11276 :

Une image contenant texte, logiciel, Icône d’ordinateur, nombre

Description générée automatiquement

Le ticket 11271 aussi résolu

Reporté sur la dev TFS

Reste a fusionner et lancer les builds

Le temps pris sur chaque ticket est suivi :

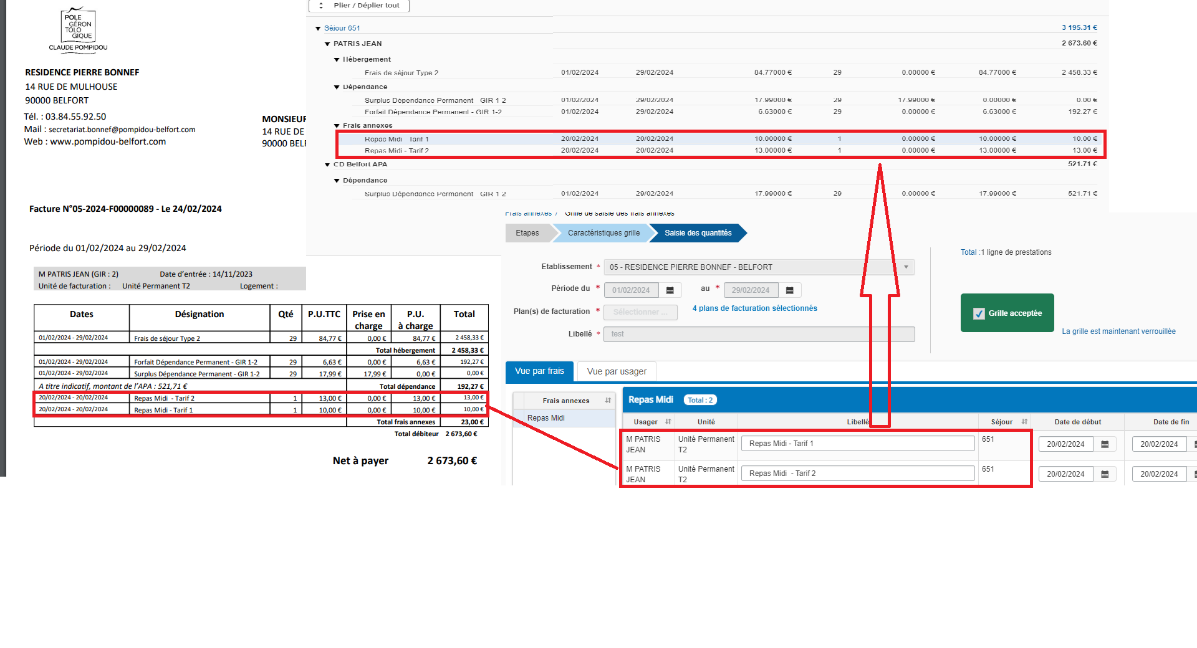
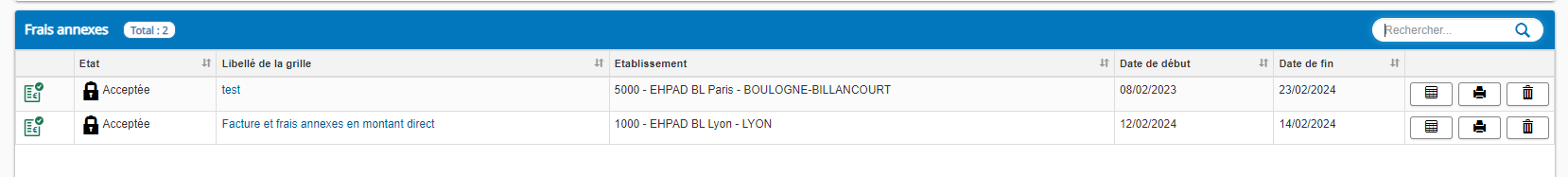
Une image contenant texte, capture d’écran

Description générée automatiquement

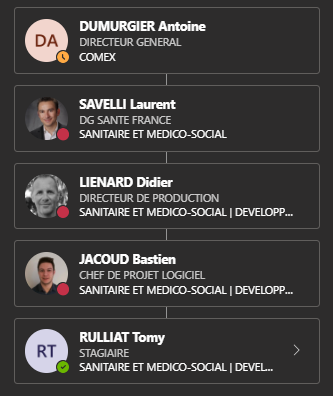
Une image contenant texte, capture d’écran

Description générée automatiquement

13/02/2024

* DSM :
  + J’ai fait :
    - J’ai résolu les tickets 11276 et 11271
  + Je vais faire :
    - Il ne me reste plus qu’a fusionner et lancer les builds
    - Attaquer le ticket 11082
* Ticket 11082 :
  + 
* Ce qu'il faut corriger c'est que :
  + En mode de décompte détaillé, les lignes de montant direct doivent être distinguées sur la facture, si elles ont même libellé (exactement comme dans le cas où les libellés sont différents)
  + En mode de décompte cumulé, les lignes de montant direct doivent être globalisées sur la facture, si elles ont même libellé, comme dans les versions antérieures
* J’ai déjà commencé à créer deux tests :
  + 
  + « test » à 2 entrées :
    - « Repas midi – Tarif 1 » à 13 euros
    - « Reaps midi – Tarif 2 » à 10 euros
  + « Facture et frais annexes en montant direct » a aussi 2 entrées :
    - « Repas midi » à 13 euros
    - « Repas midi » à 10 euros
* Pour tester, il faut que je puisse afficher les factures, comment faire ?
* Où se passe le traitement ? sante.fwm.edition ?

14/02/2024

* DSM :
  + J’ai fait :
    - J’ai suivi la doc de thomas pour le déploiement des packages nuggets
    - J’ai commencé à essayer de reproduire le bug du ticket 11082
    - Réunion
  + Je vais faire :
    - Fusion des 2 tickets avec hafiz
    - Regarder les dernières corrections Sonar avec David
    - Ticket 11082 sur les émissions de facture
    - Regarder les modifs Sonar pour recette
* 

Les modifs sonar pour recette :

* BL.Sante.Usager.Editions.Managers/OpenXml/EditionListeAttenteManager.cs L193
* BL.Sante.Usager.Models/Helper/MouvementFacturationHelper.cs L928
* BL.Sante.Usager.Models.Aurelia/usagers/models/helper/indicateurHelper.ts L30
* BL.Sante.Usager.UseCases/Interop/UsagerSanteMSInteropUseCase.cs L1050
* Encaissementdecaissementhelper.ts bug sonar à tester
  + Test de non regression sur l’encaissement mspa des factures usagers avec la gestion des acomptes
* MouvementFacturationHelper.cs
  + Test de non regression
* Concrètement pour le ticket :
  + Trouver la variable de décompte Détaillé (référentiel/lignes de prestations/«repas») :
    - Si mode de décompte détaillé :
      * Si les lignes ont le même libellé, les distinguer
      * Si les lignes n’ont pas le même libellé, les distinguer
    - Si mode de décompte cumulé :
      * Si les lignes ont le même libellé, les regrouper
      * Si les lignes n’ont pas le même libellé, les distinguer
  + La variable de l’objet « LignePrestation » :
    - 
    - Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

      Description générée automatiquement
    - Une image contenant texte, Police, Logiciel multimédia, logiciel

      Description générée automatiquement
  + L’enregistrement se fait par la fonction EnregistrerGrille du manager FraisAnnexe.cs de la solution BL.Sante.Usager
* Build des 2 premiers tickets pour la livraison en branche RECETTE
* Une image contenant texte, Police, capture d’écran

  Description générée automatiquement
* Prise de connaissance de la doc NeoLoad de Dorian
* Une image contenant texte, Logiciel multimédia, Police, logiciel

  Description générée automatiquement
* NeoLoad est un logiciel de test de charge qui permet aux équipes DevOps de concevoir et d'exécuter des tests de performance sans code sur les applications système.

Encaissementdecaissementhelper.ts bug sonar à tester

Test de non regression sur l’encaissement mspa des factures usagers avec la gestion des acomptes

MouvementFacturationHelper.cs

Test de non regression s

Passage des tickets en « IMPLEMENTED » signifiant que je les ai corrigé en indiquant la version :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

15/02/2024

* DSM :
  + J’ai fait :
    - Finalisé les corrections Sonar après la vérif des faux positifs de David
    - Commencer à regarder les modifs Sonar à tester
    - Ticket 11082 sur les émissions de facture
  + Je vais faire :
    - Retest le ticket BLRES-11271 (qui n’est pas reproductible en local🡪Voir avec un dev)
    - Le ticket 11082
* Ticket BLRES-11276 résolu :
* Une image contenant texte, capture d’écran, Police, nombre

  Description générée automatiquement

16/02/2024

* DSM :
  + J’ai fait :
    - Passer beaucoup de temps sur la mise en place d’un env RTM pour pouvoir tester le ticket que christine m’a demandé en 20230501 et pour pouvoir commencer à corriger mon dernier ticket
  + Je vais faire :
    - Enfin commencer à corriger le ticket sur les emissions de factures
* Pb dans le téléchargement des factures :



* Sur les factures test, repas doit être détaillé et petit déjeuner doit être cumulé :
* Une image contenant texte, capture d’écran, nombre, Parallèle

  Description générée automatiquement
* Ci-dessus, « Repas » doit être sur 2 lignes, « Petit Déjeuner » sur une ligne
* Une image contenant texte, capture d’écran, nombre, Parallèle

  Description générée automatiquement
* Ci-dessus, du temps que les frais annexes n’ont pas le même libellé, les lignes doivent toutes apparaîtrent
  + Juste corriger, si « Detaille » et le même nom, alors afficher 2 lignes
* Présentation de Thomas VIAL sur neoload

19/02/2024

* Phillipe : Document HL7 de test pour le projet de l’année dernière
* Lucy : voir les applications
* DSM :
  + Bloquant :
    - J’ai un peu de mal à trouvé l’origine d’un bug, si un dev aura un peu de temps pour m’aider
  + J’ai fait :
    - Présentation NeoLoad
    - Ticket BLRES-11082 sur les émissions de factures
  + Je vais faire :
    - Ticket sur les émissions de factures
* Lien :
* Une image contenant texte, capture d’écran, logiciel, nombre

  Description générée automatiquement
* Avec :
* Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

  Description générée automatiquement
* Créer un type pour avoir un type de retour ?
* Changer le linq
* Template de facture :
* Une image contenant texte, capture d’écran, nombre, ligne

  Description générée automatiquement

20/02/2024

* DSM :
  + J’ai fait :
    - (Avec l’aide d’hafiz) J’ai enfin réussi à trouver le bug du ticket sur les émissions de facture, je pense avoir réussi à le corriger
  + Je vais faire :
    - Finaliser mon ticket, tester
    - Tester et corriger aussi le ticket que Christine m’a demandé sur la duplication d’établissement
* Correction du ticket sur les émissions de factures :  
  Une image contenant texte, capture d’écran, logiciel

  Description générée automatiquement
* Avant (cela n’allait pas car Repas est défini en « cumulé » et doit donc s’afficher sur plusieurs lignes) :  
  Une image contenant texte, capture d’écran, nombre, Police

  Description générée automatiquement
* Après correction (Ici il y a 4 repas de tarifs différents) :  
  Une image contenant texte, capture d’écran, nombre, Police

  Description générée automatiquement

Test

Fusion rtm

Fusion rtm vers recette

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

  Description générée automatiquement

21/02/2024

* DSM :
  + J’ai fait :
    - Au niveau du ticket (BLRES-11271) que Christine m’avait demandé de regarder, on a regardé avec David, tout est bon
    - Correction du ticket sur les émissions de factures, il ne me reste plus qu’a le reporté sous GIT
  + Je vais faire :
    - [FAIT] Reporté sous GIT
    - Repasser sur les correctifs Sonar pour voir ce qu’il y a à recetter
    - Que faire ? TU ?
    - Call avec Thomas padilla pour lui montrer les corrections Sonar
    - Call avec Ouiam pour lui montrer les prérequis pour recetter le ticket 11082
    - [FAIT] Repasser sur les tickets pour com : « conforme à ce qui est demandé », « voir synthèse dev »…

22/02/2024

* DSM :
  + J’ai fait :
    - Reporté le ticket 11082 sous git
    - Passer un moment avec ouiam pour recetter le ticket
    - TU de sante.finance
  + Je vais faire :
    - Regarder la merge request sur les correctifs sonar sante.fwm avec hafiz
    - TU

23/02/2024

* DSM :
  + J’ai fait :
    - Appel avec thomas sur les questions qu’il avait au niveau de SonarQube
    - TU
  + Je vais faire :
    - Merge avec hafiz si tu as du temps comme ça, c’est fait avant que je parte ducoup.
    - TU
* FINI

Comme je fini ce soir, je voulais vous remercier de m’avoir accueilli, intégré et de m’avoir aidé

J’ai beaucoup appris et y’avais une bonne entente donc merci pour ça, à ce soir pour hafiz et peut-être à la prochaine pour tous les autres

POUR LES FICHIERS HL7 (stage 1) :

L'application que vous avez développée, permettant la conversion de fichiers HL7ORU en format XML et PDF tout en anonymisant les données personnelles des patients, présente plusieurs avantages dans le contexte d'une entreprise de logiciel travaillant dans le domaine de la santé. Voici une explication détaillée de l'intérêt de cette conversion et des raisons pour lesquelles on utilise des fichiers HL7ORU :

1. **Interopérabilité des Systèmes de Santé :**

* Les fichiers HL7 (Health Level Seven) sont des normes de communication utilisées dans le domaine de la santé pour permettre l'interopérabilité entre différents systèmes d'information médicale. Le format HL7ORU (Observation Result Unsolicited) est spécifiquement conçu pour les résultats d'observations non sollicités, tels que les résultats de tests médicaux.
* En convertissant ces fichiers HL7ORU en formats plus courants comme XML et PDF, vous facilitez l'intégration des données médicales dans d'autres systèmes et applications, améliorant ainsi la collaboration et l'échange d'informations entre les différents acteurs du secteur de la santé.

1. **Accessibilité des Données :**

* La conversion en formats plus conviviaux comme XML et PDF permet aux professionnels de la santé d'accéder plus facilement aux données médicales. Les fichiers XML sont structurés et peuvent être traités plus facilement par des applications informatiques, tandis que les fichiers PDF offrent une visualisation standardisée et facile à partager.

1. **Confidentialité des Données :**

* Anonymiser les données personnelles des patients avant la conversion est essentiel pour respecter les normes de confidentialité et de protection des données de santé, telles que la norme HIPAA (Health Insurance Portability and Accountability Act) aux États-Unis. Cela garantit que les informations sensibles des patients ne sont pas divulguées involontairement lors de la transmission ou du partage des résultats d'observations.

1. **Présentation Visuelle des Résultats :**

* La conversion en format PDF offre une présentation visuelle standardisée des résultats d'observations médicales. Cela peut être particulièrement utile pour les rapports médicaux destinés aux patients, aux médecins ou à d'autres professionnels de la santé, améliorant ainsi la compréhension et la communication des informations médicales.

1. **Stockage et Archivage :**

* Les fichiers XML et PDF sont adaptés au stockage à long terme et à l'archivage des données médicales. La conversion permet donc de créer des archives électroniques plus accessibles et durables, facilitant la gestion des données de santé sur le long terme.

En résumé, la conversion de fichiers HL7ORU en formats plus conviviaux et la prise en compte de l'anonymisation des données contribuent à améliorer l'interopérabilité, l'accessibilité, la confidentialité et la gestion des données médicales dans le domaine de la santé.

Au cas ou :  
using BL.Sante.Usager.Data.Providers.CertDC;

using BL.Sante.Usager.Models.CertDC;

using FluentAssertions;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using NHibernate;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Test.Sante.Finance.UnitTests;

using static BL.Sante.Usager.Models.CertDC.EnvoiCertDC;

namespace Test.Sante.Usager.UnitTests.DAO.CertDC

{

[TestClass]

internal class EnvoiCertDCDAOTests : TestBase

{

private ISessionFactory sessionFactory;

private ISession session;

private EnvoiCertDCDAO envoiCertDCDAO;

public void ShouldNotThrowSaveAndClean()

{

EnvoiCertDC envoiCertDC = new EnvoiCertDC();

// Initialiser envoiCertDC avec les données nécessaires pour le test : a vérifier

Action act = () => envoiCertDCDAO.SaveAndClean(envoiCertDC);

// Assert

act.ShouldNotThrow("Aucune exception ne doit être levée");

// Vérifier que l'enregistrement a été effectué avec succès

EnvoiCertDC savedEnvoiCertDC = envoiCertDCDAO.GetByUsagerId(envoiCertDC.Sejour.Usager.Id);

Assert.IsNotNull(savedEnvoiCertDC, "L'enregistrement ne doit pas être null");

}

[TestMethod]

public void ShouldNotThrowGetStatut()

{

long usagerId = 1; // ID d'usager valide dans la BDD

// Act

EnvoiCertDCStatut result = 0;

Action act = () => result = envoiCertDCDAO.GetStatut(usagerId);

// Assert

act.ShouldNotThrow("Aucune exception ne doit être levée");

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

}

[TestMethod]

public void ShouldNotThrowGetByUsagerId()

{

long usagerId = 1; // ID d'usager valide dans votre base de données

// Act

EnvoiCertDC result = null;

Action act = () => result = envoiCertDCDAO.GetByUsagerId(usagerId);

// Assert

act.ShouldNotThrow("Aucune exception ne doit être levée");

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

}

}

}

using BL.Core.Fwm.Models;

using BL.Sante.Usager.Data.Contracts;

using BL.Sante.Usager.Models.DTO.Documents;

using BL.Sante.Usager.Models;

using FluentAssertions;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Test.Sante.Finance.UnitTests;

using BL.Sante.Usager.Data.Contracts.Archivage;

using BL.Sante.Usager.Models.Archivage;

using BL.Sante.Usager.Models.DTO.Archivage;

using BL.Core.Fwt.Models;

namespace Test.Sante.Usager.UnitTests.DAO.Archivage

{

[TestClass]

internal class ArchivageInfoUsagerDAOTests : TestBase

{

private IArchivageInfoUsagerDataContract ArchivageInfoUsagerDAO;

[TestMethod]

public void ShouldNotThrowGetArchivageInfoUsagerByUsagerId()

{

long usagerId = 1;

ArchivageInfoUsager result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetArchivageInfoUsagerByUsagerId(usagerId);

// Assert

act.ShouldNotThrow();

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

Assert.AreEqual(usagerId, result?.Id, "L'Id de l'usager dans le résultat doit correspondre à l'Id attendu");

}

[TestMethod]

public void ShouldNotThrowGetAllByIds()

{

IList<long> idList = new List<long> { 1, 2, 3 };

IList<ArchivageInfoUsager> result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetAllByIds(idList);

// Assert

act.ShouldNotThrow();

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

Assert.AreEqual(idList.Count,result.Count,$"Le nombre d'éléments dans le résultat doit correspondre au nombre attendu. IDs attendus : {string.Join(", ", idList)}");

}

[TestMethod]

public void ShouldNotThrowGetUsagersForArchivageAutomatique()

{

int seuilArchivage = 42;

IList<ArchivageAutomatiqueInfoDTO> result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetUsagersForArchivageAutomatique(seuilArchivage);

// Assert

act.ShouldNotThrow();

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

// Vérifiez que chaque élément dans le résultat a une valeur d'IdUsager valide

foreach (var dto in result)

{

Assert.IsTrue(dto.IdUsager > 0, $"L'IdUsager de l'élément dans le résultat doit être supérieur à zéro. Valeur actuelle : {dto.IdUsager}");

}

// Vérifiez que chaque élément dans le résultat a une valeur d'IdArchivageInfo valide

foreach (var dto in result)

{

Assert.IsTrue(dto.IdArchivageInfo >= 0, $"L'IdArchivageInfo de l'élément dans le résultat doit être supérieur ou égal à zéro. Valeur actuelle : {dto.IdArchivageInfo}");

}

}

[TestMethod]

public void ShouldNotThrowGetUsagersForReactivationAutomatique()

{

int seuilArchivage = 42;

IList<ReactivationAutomatiqueInfoDTO> result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetUsagersForReactivationAutomatique(seuilArchivage);

// Assert

act.ShouldNotThrow();

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

// Vérifier que chaque élément dans le résultat a une valeur d'IdUsager valide

foreach (var dto in result)

{

Assert.IsTrue(dto.IdUsager > 0, $"L'IdUsager de l'élément dans le résultat doit être supérieur à zéro. Valeur actuelle : {dto.IdUsager}");

}

// Vérifier que chaque élément dans le résultat a une valeur d'IdArchivageInfo valide

foreach (var dto in result)

{

Assert.IsTrue(dto.IdArchivageInfo >= 0, $"L'IdArchivageInfo de l'élément dans le résultat doit être supérieur ou égal à zéro. Valeur actuelle : {dto.IdArchivageInfo}");

}

// Vérifier que chaque élément dans le résultat a une valeur d'ArchivageInfoUsagerPrecedent valide

foreach (var dto in result)

{

Assert.IsNotNull(dto.ArchivageInfoUsagerPrecedent, $"ArchivageInfoUsagerPrecedent de l'élément dans le résultat ne doit pas être null");

Assert.IsTrue(dto.ArchivageInfoUsagerPrecedent.Id > 0, $"L'Id de ArchivageInfoUsagerPrecedent doit être supérieur à zéro. Valeur actuelle : {dto.ArchivageInfoUsagerPrecedent.Id}");

}

}

[TestMethod]

public void ShouldNotThrowGetDemandeurArchivageById()

{

long idDemandeurArchivage = 1; // à ajuster selon les besoins

AbstractDemandeurArchivage result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetDemandeurArchivageById<DemandeurArchivagePAP>(idDemandeurArchivage);

// Assert

act.ShouldNotThrow("Aucune exception ne doit être levée");

// Le résultat ne doit pas être null

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

// L'identifiant du demandeurArchivage correspond à celui recherché

Assert.AreEqual(idDemandeurArchivage, result.Id, "L'identifiant du résultat doit correspondre à l'identifiant recherché");

// Vérifier la propriété TypeAdresse

Assert.IsTrue(!string.IsNullOrEmpty(result.TypeAdresse), "Le demandeurArchivage doit avoir une valeur non nulle ou non vide pour la propriété TypeAdresse");

}

[TestMethod]

public void ShouldNotThrowGetDemandeurArchivageIdByDemandeurId()

{

long idDemandeur = 1; // ajuster l'identifiant selon les besoins

AbstractDemandeurArchivage result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetDemandeurArchivageIdByDemandeurId<DemandeurArchivagePAP>(idDemandeur);

// Assert

act.ShouldNotThrow("Aucune exception ne doit être levée");

// Assert que le résultat ne doit pas être null

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

// Exemple d'assertion pour vérifier l'identifiant

Assert.AreEqual(idDemandeur, result.Id, "L'identifiant du résultat doit correspondre à l'identifiant recherché");

// Vérifier la propriété TypeAdresse

Assert.IsTrue(!string.IsNullOrEmpty(result.TypeAdresse), "Le demandeurArchivage doit avoir une valeur non nulle ou non vide pour la propriété TypeAdresse");

}

[TestMethod]

public void ShouldNotThrowGetListAuditableByCriteria()

{

var listCriteria = new List<AuditableFieldDTO>{

new AuditableFieldDTO { Nom = "Nature", Value = "ArchivageAutomatique" },

new AuditableFieldDTO { Nom = "unNom", Value = "uneValeur" },

new AuditableFieldDTO { Nom = "Username", Value = "JohnDoe" }

};

IList<AuditableDTO> result = null;

Action act = () => result = ArchivageInfoUsagerDAO.GetListAuditableByCriteria(listCriteria);

// Assert

act.ShouldNotThrow("Aucune exception ne doit être levée");

// Le résultat ne doit pas être null

Assert.IsNotNull(result, "Le résultat ne doit pas être null");

// La liste résultante ne doit pas être vide si les critères sont définis

if (listCriteria.Any())

{

Assert.IsTrue(result.Any(), "La liste résultante doit contenir au moins un élément");

}

// Les éléments de la liste doivent avoir des informations correctes

foreach (var auditableDto in result)

{

Assert.IsNotNull(auditableDto.DateAction, "La date d'action ne doit pas être null");

Assert.IsNotNull(auditableDto.TypeAction, "Le type d'action ne doit pas être null");

Assert.IsNotNull(auditableDto.GroupValueObjectAuditable, "Le groupe d'objets auditables ne doit pas être null");

Assert.IsNotNull(auditableDto.Username, "Le nom d'utilisateur ne doit pas être null");

Assert.IsNotNull(auditableDto.InfosComplementaire, "Les informations complémentaires ne doivent pas être null");

}

}

}

}

UPDATE de FileConverter :

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

Description générée automatiquement



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

Description générée automatiquement

Une image contenant texte, Appareils électroniques, capture d’écran, Police

Description générée automatiquement

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

Description générée automatiquement



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

Description générée automatiquement