PAC_2024_PROJET2_GROUPE1 Home

Dossier d'analyse fonctionnelle

00 - Récapitulatif du projet

Fonctionnalités

- Gestion du contrat employé
- Gestion des prestations et absence (horaire)
- Gestion fiches de paies

Utilisateurs et droits

- Directeur ⇒ super administrateur qui a accès à tous les sites
- Manager ⇒ tous les droits sauf modifier leurs données et uniquement accès à son site
- Employés ⇒ accès à leurs données

Process de l'application

1. Un utilisateur se connecte
2. L'accès aux fonctionnalité s'effectuera sur base d'un token
3. Différentes fonctionnalités possible en fonction des rôles :

- Employé
  - Consulter ses fiches de paie
  - Consulter son contrat
  - Se déconnecter
- Manager
  - Idem que employé
  - Modifier/supprimer les contrats sauf le sien
- Directeur
  - Idem que manager
  - Peut modifier toutes les données

5. Synchronisation des données :

- Récupération des données d'Euro-Core lorsqu'on effectue une requête
- Traitement effectué par Euro-Rolls (modification salaire)
- Envoi des données sur le serveur euro-core
- Reception d'une réponse de chez euro-core pour confirmation

01 - Briques logicielles

Front-end

- **Framework/Library**: Angular pour construire une interface utilisateur.
- **Dashboard :** personnalisé pour les utilisateurs (proposant des actions pertinentes en fonctions de l'utilisateur connecté)
- **Formulaire :** pour soumettre une absence (certificat médical) ou une demande de congé

Back-end

- **Runtime et Langage**: NodeJS (Javascript)
- **Framework** : NestJS
- **ORM** : TypeORM
- Définition des classes et des méthodes
- Définition des différentes routes nécessaires
- Implémenter CRUD pour les différentes entités
- Gestion des erreurs lors de l'envoi des données
- Gérer la soumission du formulaire (lors de l'envoi vers euro-core)

Authentification et Sécurité

- **Authentication**: JWT (JSON Web Tokens) pour gérer les sessions utilisateurs.
- **Hashing**: BCrypt pour le hashage des mots de passe.
- **Role-Based Access Control (RBAC)**: Gestion des rôles et des permissions.

Test unitaire/test d'intégration

- Mocha

Documentation et deploiement

- **Documentation:** Swagger
- **CI/CD :** Git/Github/Jenkins

02 - Utilisateurs et fonctionnalités

Employé

- Consulter ses fiches de paie
- Consulter son contrat
- Ajouter des Factures
- Soumettre demande de congés / certificat médical
- Se connecter / déconnecter

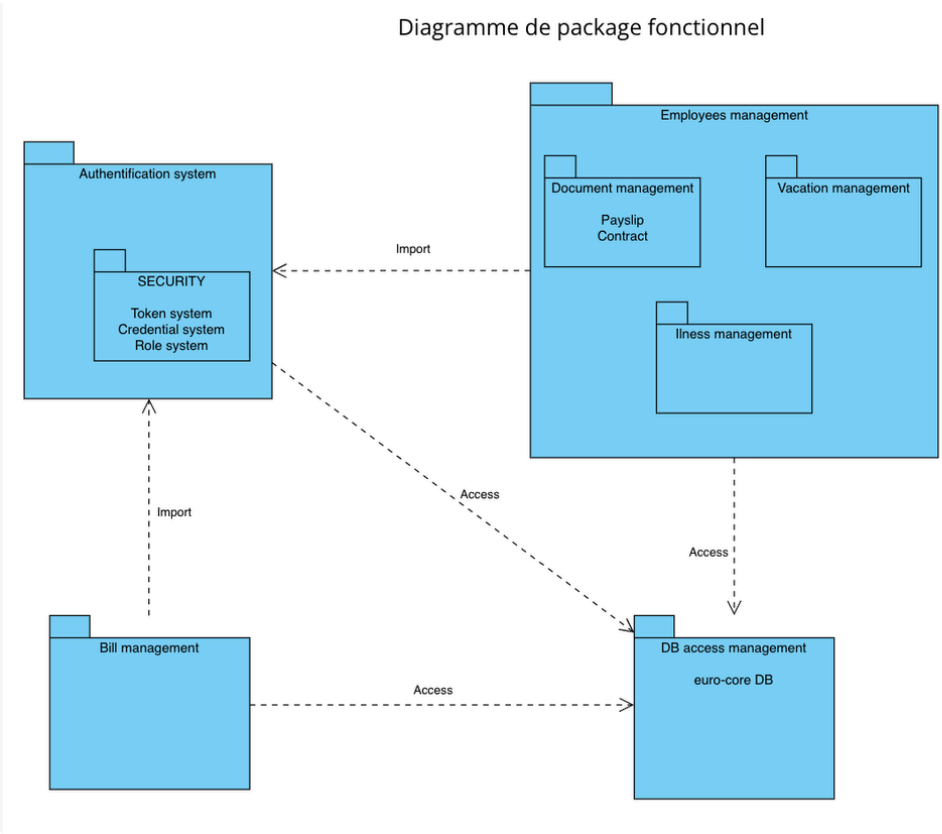
Manager

- Consulter les fiches de paie des employés de son site
- Consulter / ajouter / modifier les contrats des employés de son site mis à part son propre contrat
- Ajouter / supprimer / modifier des factures de son site
- Soumettre demande de congés / certificat médical
- Approuver demandes de congés / certificats médicaux des employés de son site
- Se connecter / déconnecter

Directeur

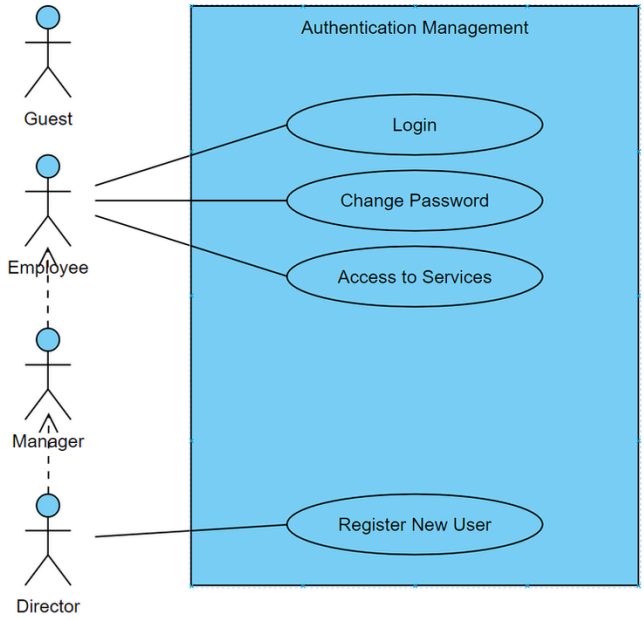
- Consulter les fiches de paie
- Consulter / ajouter / modifier les contrats
- Approuver demandes de congés / certificats médicaux
- Ajouter / supprimer / modifier des factures
- Se connecter / déconnecter

03 - Diagramme de package fonctionnel



04 - Use Case

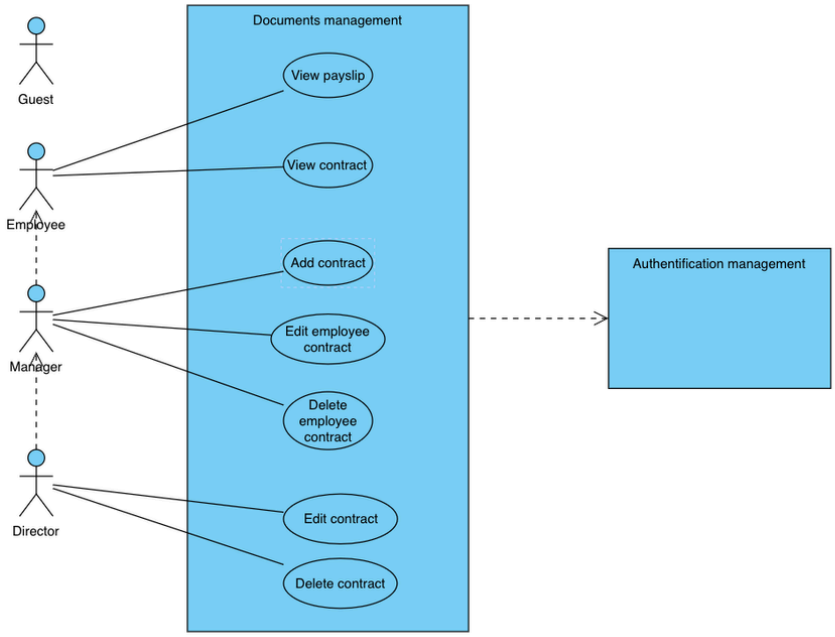
Connexion


Authentication Management
Guest
Employee
Manager
Director
Login
Change Password
Access to Services
Register New User

Ajouter Employé

Nom	Ajouter un employé
Description/Contexte	Permet au manager d'ajouter un employé
Acteur	Directeur
Donnée en entrée	Information sur le nouvel employé(nom, prénom, adresse, localité,...)
Précondition	Le Token du directeur est valide
Postcondition	Le nouvel utilisateur est ajouté à la base de données
Donnée en sortie	Message de confirmation
Scénario principal	1. Le Directeur sélectionne l'option "Ajouter nouvel employé" dans le menu 2. Le directeur insère les données de l'employé dans le formulaire 3. Il appuie sur le bouton valider. 4. Le système affiche un message de confirmation indiquant que l'utilisateur a été créé avec succès
Scénario échec	Le Token de l'utilisateur n'est pas valide et/ou le format des données n'est pas valide.
Scénario alternatif	Le directeur peut choisir d'annuler l'opération à tout moment avant la validation. Le système retourne à l'état précédent sans ajouter une nouvel utilisateur à la base de données.

Documents


Documents management
View payslip
View contract
Add contract
Edit employee contract
Delete employee contract
Edit contract
Delete contract
Guest
Employee
Manager
Director
Authentification management

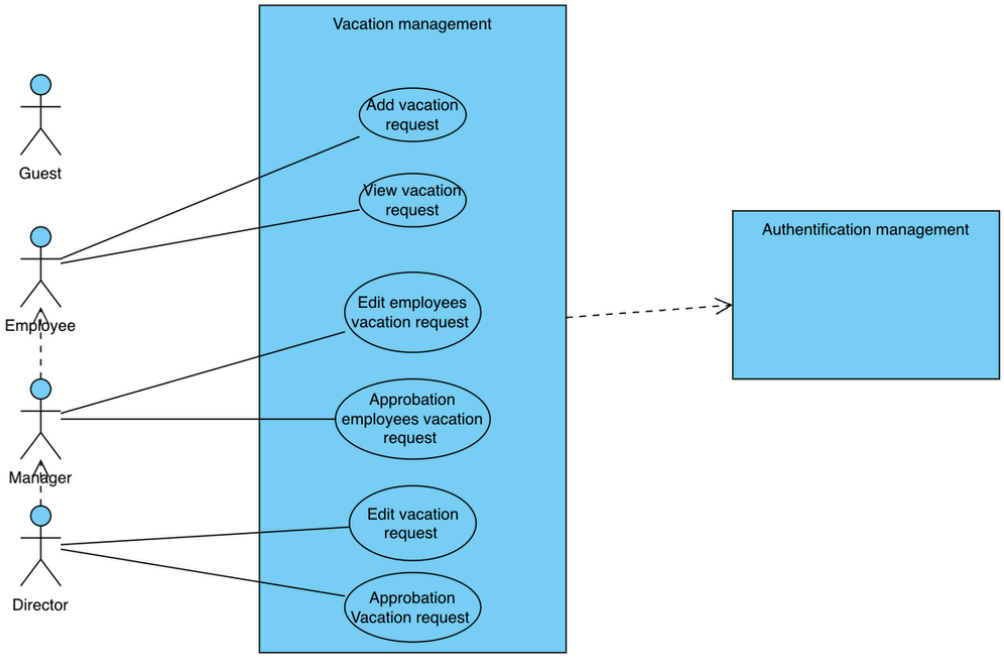
Consulter fiche de paie

Nom	Consulter fiche de paie
Description/Contexte	Permet à un employé de consulter ses fiches de paie
Acteur	Employé
Donnée en entrée	• Identifiant de l'employé (automatiquement dérivé de la session active) • Mois et année de la fiche de paie souhaitée
Précondition	• L'employé doit être authentifié et connecté à l'application. • Les fiches de paie doivent être préalablement générées et disponibles dans le système.
Postcondition	• La fiche de paie demandée est affichée à l'écran. • Un enregistrement de la consultation de la fiche de paie est créé dans le système pour des fins de traçabilité.
Donnée en sortie	Détails de la fiche de paie
Scénario principal	1. L'employé navigue vers la section "Documents personnels". 2. L'employé sélectionne l'option "Fiches de paie". 3. L'application affiche une liste de fiches de paie disponibles. 4. L'employé sélectionne une fiche de paie pour un mois et une année spécifiques. 5. L'application vérifie les droits d'accès de l'employé pour cette fiche de paie. 6. L'application récupère les détails de la fiche de paie depuis la base de données 7. L'application affiche la fiche de paie à l'écran. 8. Un enregistrement de la consultation est créé pour des fins de traçabilité.
Scénario echec	7a. Le système ne trouve pas la fiche de paie demandée dans la base de données. Le système affiche un message d'erreur indiquant que la fiche de paie n'est pas disponible. Le cas d'utilisation s'arrête.
Scénario alternatif	4a. L'application affiche un message indiquant qu'aucune fiche de paie n'est disponible pour l'employé. Le cas d'utilisation s'arrête. 6a. Le système détecte que l'employé n'a pas les droits nécessaires pour consulter la fiche de paie demandée. Le système affiche un message d'erreur indiquant un problème d'autorisation. Le cas d'utilisation s'arrête.

Ajouter un nouveau contrat

Nom	Ajouter un Nouveau Contrat
Description/Contexte	Permet à un manager d'ajouter un nouveau contrat pour un employé de son site
Acteur	Manager
Donnée en entrée	Toutes les informations sur le contrat (nom de l'employé, poste, date de début, date de fin, salaire, avantages)
Précondition	• Le manager est authentifié et connecté à l'application. • Le manager a les droits d'accès nécessaires pour ajouter des contrats.
Postcondition	• Le nouveau contrat est ajouté au système et est disponible pour consultation. • Une confirmation est affichée au manager
Donnée en sortie	• Confirmation de l'ajout du contrat. • Détails du contrat ajouté.
Scénario principal	1. Le manager navigue vers la section "Gestion des contrats". 2. Le manager sélectionne l'option "Ajouter un nouveau contrat". 3. Le système affiche un formulaire pour l'ajout d'un nouveau contrat. 4. Le manager remplit les informations du contrat (nom de l'employé, poste, date de début, date de fin, salaire, avantages). 5. Le manager soumet le formulaire. 6. Le système vérifie les droits d'accès du manager et valide les données du formulaire. 7. Le système enregistre le nouveau contrat dans la base de données. 8. Le système affiche une confirmation de l'ajout du contrat au manager. 9. Le système envoie une notification à l'employé informant de l'ajout du nouveau contrat.
Scénario echec	7a. Le système détecte que le manager n'a pas les droits nécessaires pour ajouter un nouveau contrat. Le système affiche un message d'erreur indiquant un problème d'autorisation. Le cas d'utilisation s'arrête. 7b. Le système détecte une erreur dans les données du formulaire (par exemple, une date invalide ou un champ manquant). Le système affiche un message d'erreur demandant au manager de corriger les données. Retour à l'étape 4.
Scénario alternatif	10a. Si la notification est désactivée pour cet employé, le système n'envoie pas de notification, mais le contrat est ajouté normalement. Le cas d'utilisation reprend à l'étape 9.

Demandes de Congés



Ajouter demande de congé

Nom	Ajouter une Demande de Congé
Description/Contexte	Permet à l'employé d'ajouter une nouvelle demande de congé.
Acteur	Employé
Donnée en entrée	Informations sur la demande de congé (date de début, date de fin, type de congé)
Précondition	• L'employé est authentifié et connecté à l'application.
Postcondition	• La nouvelle demande de congé est enregistrée dans le système et est soumise pour approbation. • Une confirmation est affichée à l'employé.
Donnée en sortie	• Confirmation de l'ajout de la demande de congé. • Détails de la demande de congé ajoutée.
Scénario principal	1. L'employé navigue vers la section "Demandes de Congé". 2. L'employé sélectionne l'option "Ajouter une demande de congé". 3. Le système affiche le nombre de congés restants au-dessus du formulaire pour que l'employé puisse vérifier. 4. L'employé choisit le type de congé parmi les options disponibles (vacances, jours fériés, récupération). 5. L'employé remplit les autres informations de la demande de congé (date de début, date de fin). 6. L'employé soumet le formulaire. 7. Le système vérifie si l'employé dispose de suffisamment de jours de congé pour la demande. 8. Le système valide les données du formulaire et enregistre la nouvelle demande de congé dans la base de données. 9. Le système affiche une confirmation de l'ajout de la demande de congé à l'employé. 10. La demande de congé est soumise pour approbation.
Scénario echec	7a l'employé n'a pas suffisamment de jours de congé, le système affiche un message d'erreur indiquant que l'employé n'a pas assez de jours de congé pour cette demande. Retour à l'étape 5 pour que l'employé puisse modifier sa demande. 10a. Le système détecte une erreur dans les données du formulaire (par exemple, une date invalide ou un champ manquant). Le système affiche un message d'erreur demandant à l'employé de corriger les données. Retour à l'étape 5.
Scénario alternatif	6a. L'employé choisit une demande de congé pour des dates spécifiques qui nécessitent une approbation spéciale (par exemple, pendant une période de pic d'activité). Le système affiche un message indiquant que la demande de congé nécessite une approbation spéciale et informe l'employé des étapes à suivre.

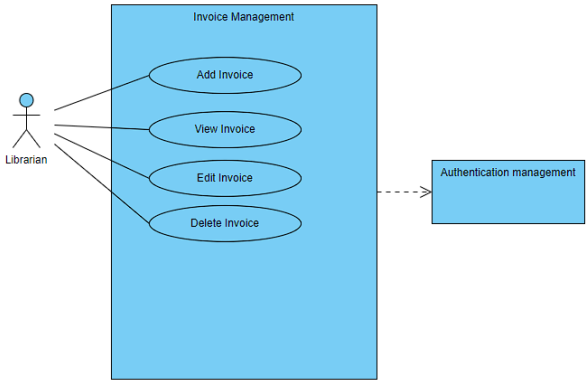
Editer demande de congé (manager)

Nom	Éditer une Demande de Congé
Description/Contexte	Permet à un manager d'éditer une demande de congé existante soumise par un employé.
Acteur	Manager
Donnée en entrée	• Identifiant de l'employé (nom, prénom)
Précondition	• Le manager est authentifié et connecté à l'application. • Une demande de congé existante soumise par un employé est disponible pour édition.
Postcondition	• La demande de congé est mise à jour avec les modifications apportées par le manager. • Une confirmation de la mise à jour est affichée au manager. • Une confirmation de la mise à jour est envoyée à l'employé.
Donnée en sortie	• Confirmation de la mise à jour de la demande de congé. • Détails de la demande de congé mise à jour.
Scénario principal	1. Le manager navigue vers la section "Demandes de Congé". 2. Le manager recherche la demande de congé à éditer. 3. Le manager sélectionne l'option "Éditer" à côté de la demande de congé souhaitée. 4. Le système affiche le formulaire de modification de la demande de congé avec les informations existantes pré-remplies. 5. Le manager modifie les informations de la demande de congé selon les besoins 6. Le manager soumet le formulaire de modification. 7. Le système valide les modifications et met à jour la demande de congé dans la base de données. 8. Le système affiche une confirmation de la mise à jour de la demande de congé au manager. 9. La demande de congé mise à jour est soumise pour approbation.
Scénario echec	3a. Le manager ne parvient pas à trouver la demande de congé à éditer. Le système affiche un message d'erreur indiquant que la demande de congé n'a pas été trouvée. Le cas d'utilisation s'arrête. 7a. Le système détecte une erreur dans les données du formulaire (par exemple, une date invalide ou un champ manquant, jours de congés insuffisant). Le système affiche un message d'erreur demandant au manager de corriger les données. Retour à l'étape 5.
Scénario alternatif	7b. Le manager décide d'annuler la modification de la demande de congé. Le système annule les modifications et retourne à l'affichage des demandes de congé.

Approbation demande de congé

Nom	Approuver une demande de congé
Description/Contexte	Permet au directeur d'approuver une demande de congé soumise par un employé ou un manager
Acteur	Directeur
Donnée en entrée	• Informations sur la demande de congé à accepter (date de début, date de fin, type de congé) • Motif de refus (le cas échéant)
Précondition	• Le directeur est authentifié et connecté à l'application. • Une demande de congé soumise par un employé ou manager est disponible pour acceptation.
Postcondition	• La demande de congé est acceptée et mise à jour dans le système (si acceptée). • La demande de congé est refusée avec indication du motif (si refusée). • Une confirmation de l'action entreprise est envoyée au directeur et à l'employé concerné.
Donnée en sortie	• Confirmation de l'action entreprise (acceptation ou refus) pour le directeur. • Notification envoyée à l'employé contenant le résultat de l'action et, en cas de refus, le motif de refus.
Scénario principal	1. Le directeur accède à la liste des demandes de congé soumises. 2. Le directeur recherche la demande de congé à traiter. 3. Le directeur approuve la demande de congé 4. Le système enregistre l'acceptation de la demande de congé dans la base de données. 5. Le système envoie une confirmation de l'acceptation au directeur et à l'employé concerné.
Scénario echec	2a. Le directeur ne parvient pas à trouver la demande de congé à accepter. Le système affiche un message d'erreur indiquant que la demande de congé n'a pas été trouvée. Le cas d'utilisation s'arrête.
Scénario alternatif	3a Le directeur refuse la demande de congé. Le directeur saisit un motif de refus dans un formulaire dédié. Le système enregistre le refus de la demande de congé avec le motif dans la base de données. Le système envoie une confirmation du refus avec le motif au directeur et à l'employé concerné.

Factures


Invoice Management
Add Invoice
View Invoice
Edit Invoice
Delete Invoice
Librarian
Authentication management

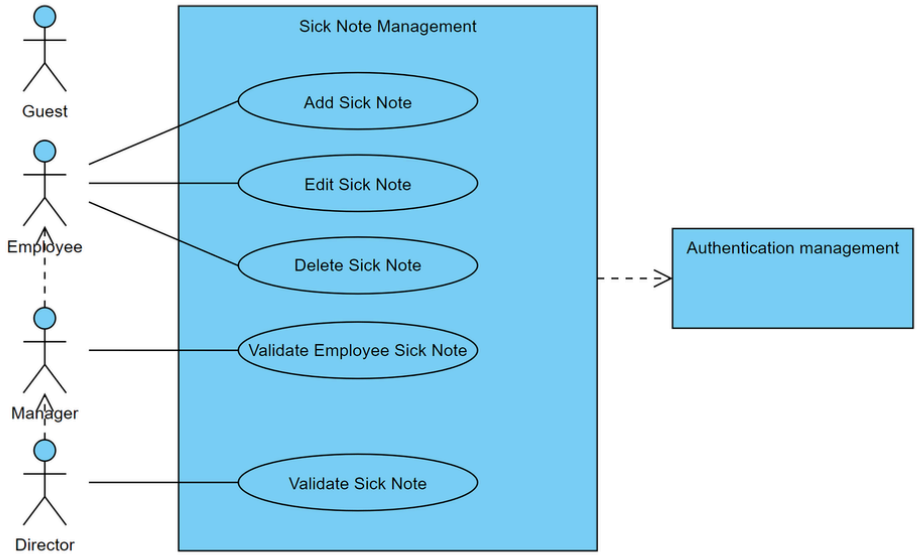
Ajouter Facture

Nom	Ajouter une facture
Description/Contexte	Permet aux utilisateurs d'ajouter une nouvelle facture dans la base de données
Acteur	Employé/Manageur/Directeur
Donnée en entrée	Informations sur la facture (prix,destinataire,service,date)
Précondition	L'utilisateur possède un Token valide
Postcondition	La facture est ajoutée à la base de données
Donnée en sortie	Message de confirmation
Scénario principal	1. L'utilisateur sélectionne l'option "Ajouter une facture" dans le menu 2. Le système affiche un formulaire de saisie des informations de la facture en fonction des droits de l'utilisateur 3. L'utilisateur remplit les informations et valide le formulaire 4. Le système enregistre les informations de la facture dans la base de données 5. Le système affiche un message de confirmation indiquant que la facture a été ajoutée avec succès
Scénario échec	Les données saisies ne sont pas valides ou il y a une erreur lors de l'enregistrement. Le système affiche un message d'erreur et invite le bibliothécaire à corriger les informations
Scénario alternatif	L'utilisateur peut choisir d'annuler l'opération à tout moment avant la validation. Le système retourne à l'état précédent sans supprimer la facture

Supprimer Facture

Nom	Supprimer une facture
Description/Contexte	Permet aux managers/directeur de supprimer une facture en fonction de leurs droits
Acteur	Bibliothécaire
Donnée en entrée	Informations sur la facture (id, service donné, prix, site, date, destinataire)
Précondition	L'utilisateur possède un Token valide
Postcondition	La facture est supprimée
Donnée en sortie	Message de confirmation
Scénario principal	1. L'utilisateur sélectionne l'option "Supprimer facture" à côté du document dans la liste 2. Le système affiche la facture suivie d'un message de confirmation 3. l'utilisateur confirme la suppression de la facture 4. Le système affiche un message de confirmation indiquant que la facture a été supprimée avec succes
Scénario echec	Le Token de l'utilisateur n'est pas valide et/ou il ne dispose pas des droits nécessaire à cette action ou le document n'existe plus et un message d'erreur le signale
Scénario alternatif	L'utilisateur peut choisir d'annuler l'opération à tout moment avant la validation. Le système retourne à l'état précédent sans supprimer la facture

Certificats médicaux



Valider C.M.

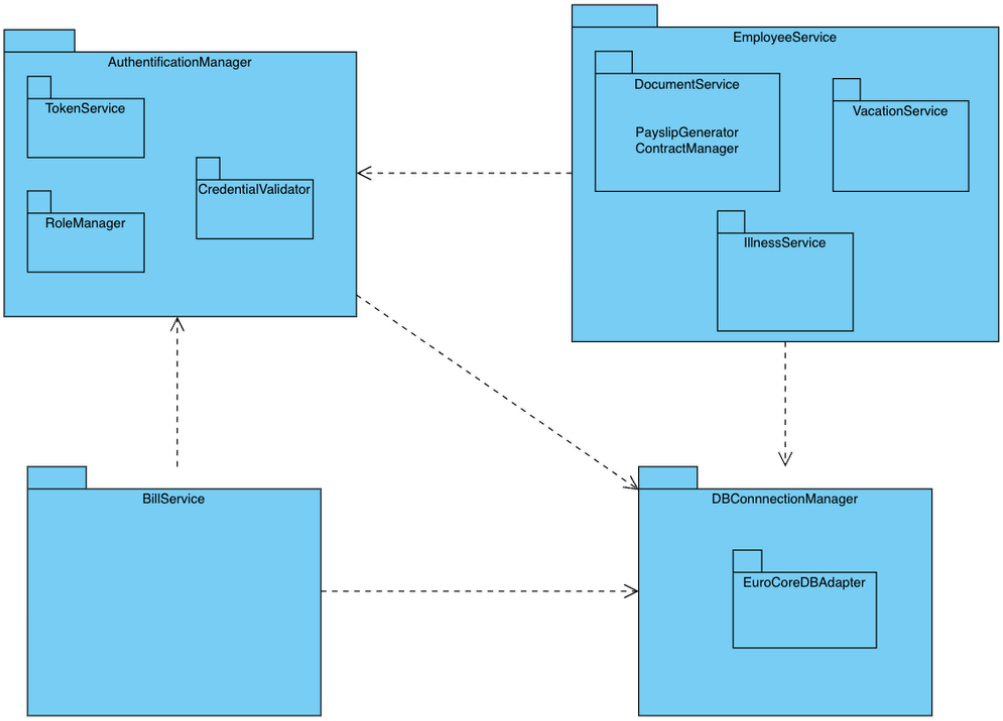
Nom	Valider un certificat médical
Description/Contexte	Permet au manageur/directeur de valider un certificat médical
Acteur	Manageur/Directeur
Donnée en entrée	Information sur le certificat médical (date, durée, raison, document, identifiant unique de l'employé)
Précondition	L'utilisateur possède un Token valide
Postcondition	Le statut du certificat médical est modifié dans la base de données.
Donnée en sortie	Message de confirmation
Scénario principal	1. Le Manageur/Directeur sélectionne l'option "Gestion des absences" dans le menu 2. L'utilisateur sélectionne le certificat médical qu'il désire valider 3. L'utilisateur appuie sur le bouton valider. 4. Le système affiche un message de confirmation indiquant que le certificat médical a été validé avec succès
Scénario echec	Le Token de l'utilisateur n'est pas valide et/ou il ne dispose pas des droits nécessaire à cette action ou le document n'existe plus et un message d'erreur le signale.
Scénario alternatif	L'utilisateur peut choisir d'annuler l'opération à tout moment avant la validation. Le système retourne à l'état précédent sans modifier le statut de validité du certificat médical

Editer C.M.

Nom	Editer un certificat médical
Description/Contexte	Permet a l'employé/manageur/directeur de valider un certificat médical
Acteur	Employé/Manageur/Directeur
Donnée en entrée	Information sur le certificat médical (date, durée, raison, document, identifiant unique de l'employé)
Précondition	L'utilisateur possède un Token valide
Postcondition	Le statut du certificat médical est modifié dans la base de données.
Donnée en sortie	Message de confirmation
Scénario principal	1. L' Employé/Manageur/Directeur sélectionne l'option "Gestion des absences" dans le menu 2. L'utilisateur sélectionne le certificat médical qu'il désire modifier 3. L'utilisateur modifie les données du certificat médical 4. L'utilisateur appuie sur le bouton valider. 5. Le système affiche un message de confirmation indiquant que le certificat médical a été validé avec succès
Scénario echec	Le Token de l'utilisateur n'est pas valide et/ou le format des données n'est pas valide.
Scénario alternatif	L'utilisateur peut choisir d'annuler l'opération à tout moment avant la validation. Le système retourne à l'état précédent sans éditer l'état du certificat médical

Dossier d'analyse technique

01 - Diagramme de package technique



02 - Architecture logicielle

Microservice

Nous avons opté pour une architecture en micro-service.

Avantages

- Modularité et maintenabilité : chaque fonctionnalité (gestion des fiches de paie, contrat, ...) peut être développée, déployée et maintenue indépendamment. Cela permet aux équipes de se concentrer sur des domaines sans interférer avec les autres services
- Evolutivité : il sera possible de mettre à jour et faire évoluer chaque service sans devoir redéployer l'entièreté de l'application
- Sécurité : une panne dans un microservice n'empechera pas l'exécution des autres

Mise en oeuvre

1. Services

- **Service de gestion des documents :** responsable de la consultation, modification et création des contrats/fiches de paie
- **Service de gestion des congés :** responsable de la demande, gestion et approbation des congés
- **Service de gestion des maladies :** responsable de la demande, consultation, modification et approbation des maladies
- **Service de gestion de la facturation :** responsable de l'ajout, la modification et la consultation des factures
- **Service d'Authentification :** responsable de la connexion

2. Communication

- API REST : les services communiqueront via des API REST sécurisée permettant une interraction fluide et sécurisée entre eux

Structure applicative

La structure applicative choisie favorise la modularité, la maintenabilité et la scalabilité.

En utilisant des frameworks modernes, des ORM pour la gestion des données, et des pratiques de sécurité robustes nous obtenons les avantages suivants:

- Séparation claire des responsabilités
- Performance
- Facilité à tester
- Facilité de déploiement
- Evolutive

Technologies

1. **Front-End : Angular**

Etant donné notre choix pour la structure en micro-service, nous avons décidé d'opter pour Angular pour le front-end. Angular offre une structure claire grâce à son architecture basée sur des modules/composants ⇒ facilite la maintenabilité et évolution de l'application

2. **Back-End : NestJS**

NestJS suit également une architecture modulaire et est inspiré par angular, ce qui permet une organisation modulaire des services back-end. De plus, NestJS intègre facilement TypeORM, ce qui permet de simplifier la gestion des bases de données relationnelles.

L'utilisation de TypeScript sur ces deux technologies présente également l'avantage de permettre la détection des erreurs au moment de la compilation grâce au typage statique, ce qui réduit les bugs et améliore la qualité du code.

03 - Synchronisation des données

Pour répondre aux exigences de ce projet, nous allons synchroniser nos données avec celles d'Euro-Core en utilisant un repository local au sein de notre application.

Chaque classe de notre application héritera d'une classe `BaseEntity` qui contiendra les propriétés suivantes :

- `Id`
- `CreationDate`
- `LastModificationDate`

Une des stratégies que nous adopterons sera la mise en cache de nos données dans un objet. Afin de maintenir des performances optimales sur le long terme, nous supprimerons les entités dont la date de dernière modification dépasse quatre semaines.

Si une entité nécessaire n'est pas présente dans notre repository local, nous effectuerons une requête HTTP GET pour la récupérer et l'ajouter à notre repository.

Nous allons également exporter ces entités dans des fichiers .JSON afin de conserver les données en cas de coupure du serveur. Cette approche permettra d'éviter des requêtes inutiles en redémarrant le système avec les données locales déjà disponibles, garantissant ainsi une meilleure résilience et une efficacité accrue.

Intégrité des Données

Euro Core devra ajouter des routes permettant de vérifier l'intégrité des données. Ces routes prendront en paramètre l'Id de l'objet ainsi que la `LastModificationDate`.

- **Si la date de dernière modification est identique** à celle envoyée par Euro Roll, nous enverrons une réponse contenant un message spécifiant que l'entité est bien à jour.
- **Si la date de dernière modification n'est pas identique** à celle envoyée par Euro Roll, nous enverrons une réponse contenant un message spécifiant qu'il y a une non-intégrité des données.
- **Si l'Id de l'objet ne correspond pas** à une entité dans la base de données, nous enverrons un message disant que l'objet n'existe pas.

Synchronisation des Données

1. **Entité à Jour** : Si l'entité est déjà à jour, nous ne ferons rien et renverrons l'entité présente dans le repository local de Euro Roll.
2. **Données Non Identiques** : Si la donnée n'est pas identique à celle d'Euro Core, nous effectuerons une requête HTTP GET à Euro Core afin de réactualiser la donnée. Lors de la réception, nous mettrons à jour le repository local ainsi que le fichier JSON.
3. **Objet Non Existant** : Si la réponse de la requête indique que l'objet n'existe plus dans la base de données d'Euro Core, nous enverrons une erreur à l'utilisateur lui indiquant que l'objet n'existe plus, et le supprimerons du repository local ainsi que du fichier JSON.

Vérifications Lors d'une Requête Post / Update

Avant la modification de l'objet, nous effectuerons les vérifications d'intégrité des données mentionnées ci-dessus. Lorsque l'utilisateur aura modifié la donnée et tentera d'effectuer une requête, nous réévaluerons l'intégrité des données :

- **Si la donnée n'a pas été altérée**, le processus se déroulera normalement.
- **Dans le cas contraire**, une erreur sera envoyée à l'utilisateur indiquant que la donnée a été modifiée depuis la dernière réception.

Diagramme d'activité gestion d'une donnée

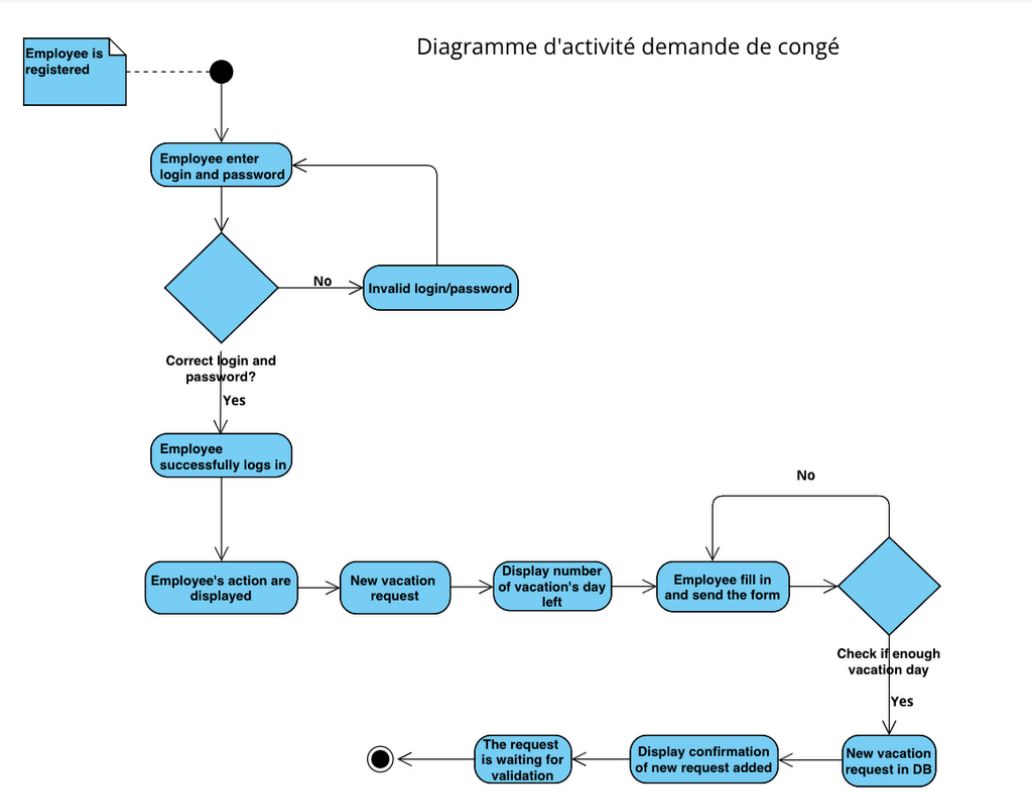
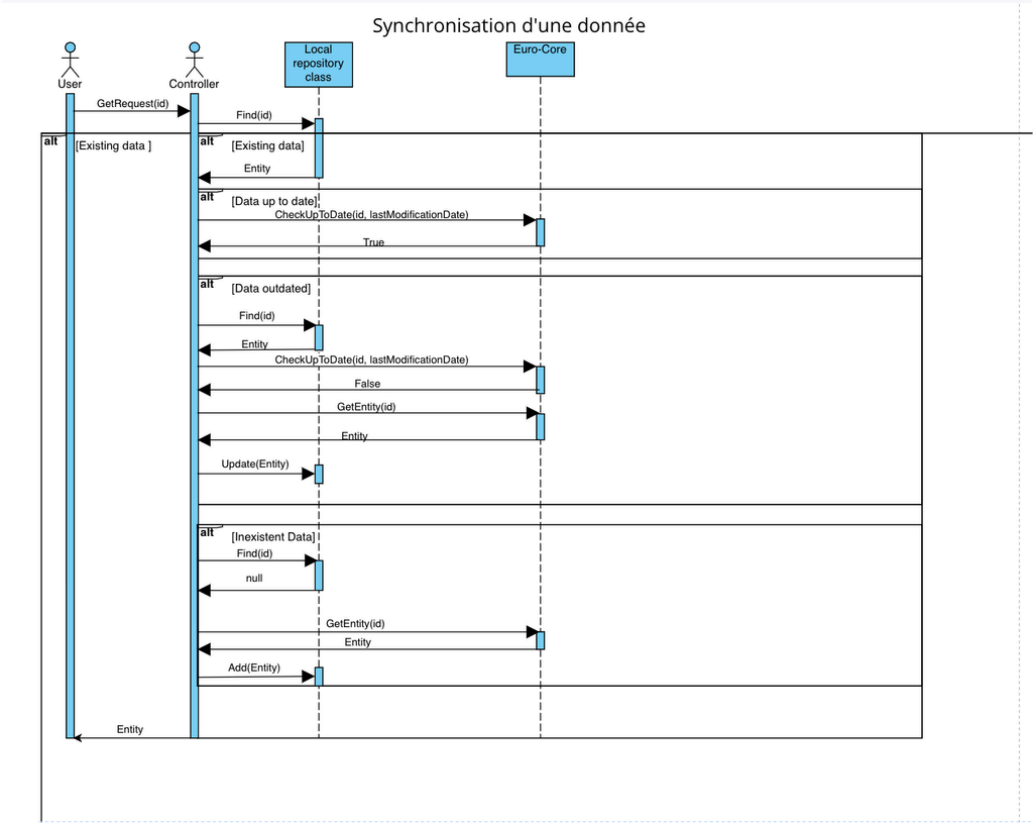
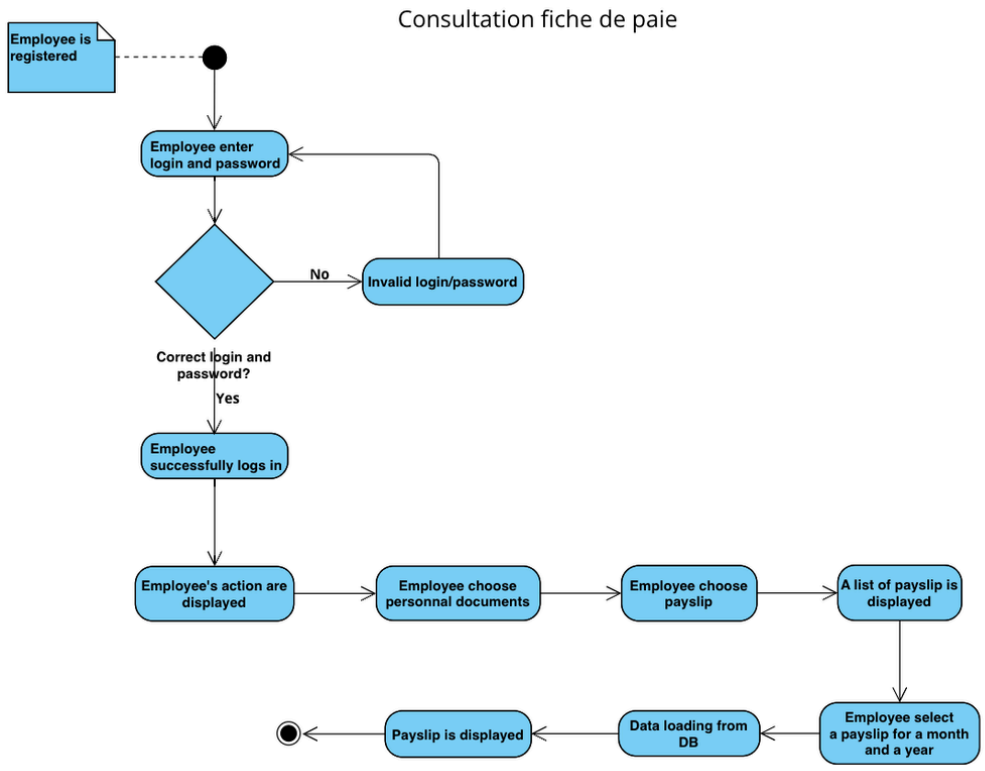


Diagramme de séquence synchronisation donnée

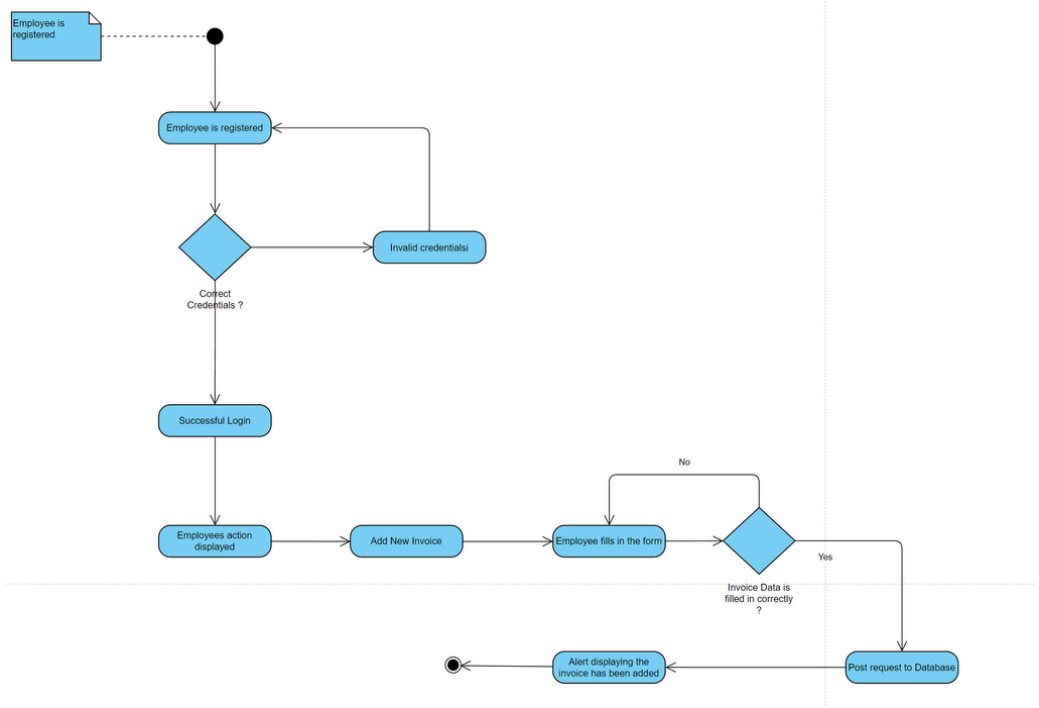


04 - Diagramme Séquence/activité

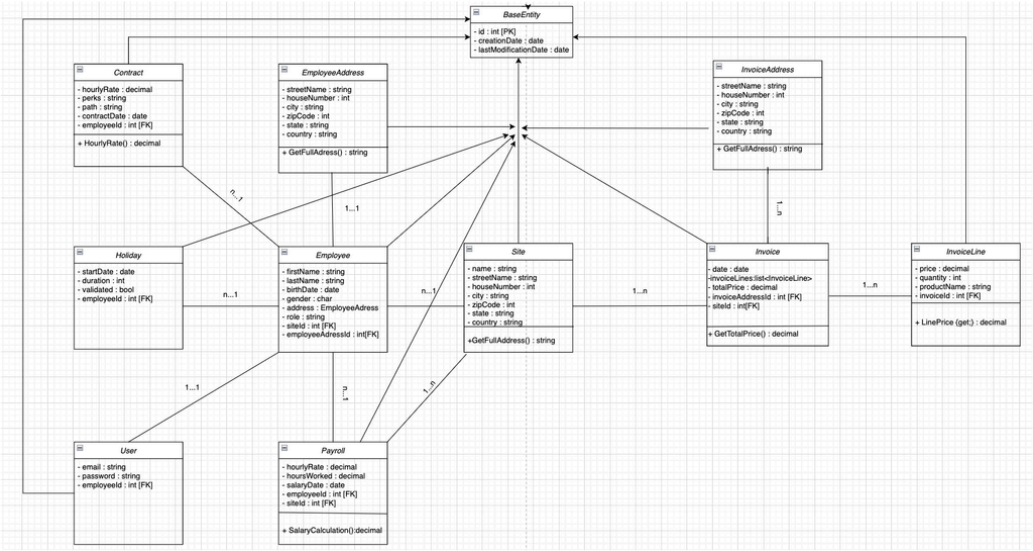
Consultation fiche de paie



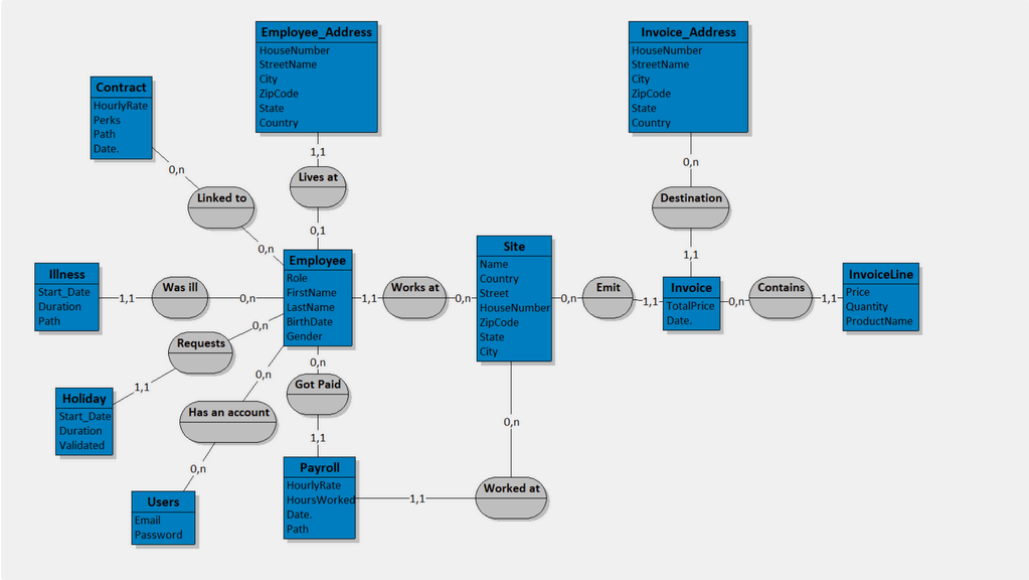
Ajouter une facture



05 - Diagramme de classe



06 - MCD

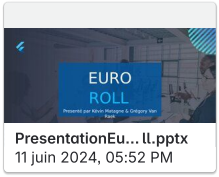

Employee_Address
HouseNumber
StreetName
City
ZipCode
State
Country
Invoice_Address
HouseNumber
StreetName
City
ZipCode
State
Country
Contract
HourlyRate
Perks
Path
Date.
1,1
0,n
Lives at
Linked to
Destination
0,1
0,n
1,1
Illness
Start_Date
Duration
Path
Was ill
Employee
Role
FirstName
LastName
BirthDate
Gender
Works at
Site
Name
Country
Street
HouseNumber
ZipCode
State
City
Emit
Invoice
TotalPrice
Date.
Contains
InvoiceLine
Price
Quantity
ProductName
1,1
0,n
Requests
Holiday
Start_Date
Duration
Validated
Got Paid
Has an account
Payroll
HourlyRate
HoursWorked
Date.
Path
Worked at
Users
Email
Password

Gestion de projet

Jira

https://vanraekgregory.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog Can't find link

Reunions client



PresentationEu... ll.pptx
11 juin 2024, 05:52 PM

1 - 21/05/2024 - Clarification du projet

1. **Processus métier**

- Comment fonctionne le process actuellement ?
  - Actuellement nous sommes sur une ancienne application rh. Nous aimerions moderniser et surtout faire une application qui fait partie de l'écosystème.
  - J'entends par là, l'intégration avec Euro Core (pour la base de donnée)
- Quel est l'ensemble des interactions employé ? (gestion absences, suivi des heures travaillées, traitement paie, ...)
  - Au niveau employé il faudra pouvoir gérer:
    - Son contrat
    - Ses prestations (horaire)
    - Ses absences (horaire et certificat medical)
    - fiche de paie
- Comment fonctionne la facturation des services interne et externe ?
  - C'est un document standard qui récupèrent les données selon le point suivant pour fournir le justificatif

2. **Utilisateurs**

- Quels sont les différents users ? (administrateurs, manager, employés, responsable rh, ...)
  - un directeur qui est aussi le super admin
  - Des managers qui ont des droits quasi complet sauf qu'ils ne pourront pas gérer leur propres données.
  - Les employés qui ont un accès à leur propre donnée.
- Que peuvent-ils faire ?
  - voir point précédent
- Quels sont leurs droits ?
  - voir point précédent

3. **Synchronisation des données**

- Comment se fera la synchronisation des données entre euro core et euro roll ?
  - A vous de proposer une solution
  - Il faut savoir que les données maitres sont celle d'euro core.
  - A vous de proposer une stratégie.
- Récupération des données de manières récurrente = ? (à quelle fréquence et de quelle manière)
  - voir point précédent
- Quelles sont les technologies utilisées par Euro core ? Comment Euro roll doit être implémentée vis à vis de celle-ci ?
  - L'application est développée en java mais cela ne doit pas impacté vos choix.

4. **Autre**

- Euro-roll devra pouvoir générer des rapports ? Si oui quelle sorte ?
  - Non les rapports sont dans une autre applications
- Doit-on gérer le calcul des salaires ? Si oui, comment gérer le fait qu'on doit gérer plusieurs législation ?
  - Oui, à vous de trouver la solution
- Doit-il y avoir une interaction avec la comptabilité ? (injection des frais automatique)
  - Certainement, il y a un ensemble de coût lié au fonctionnement qui doit apparaitre sur les facturations.

2 - 23/05/2024

Interraction employé

- Fonctionnalité pour gérer le calcul des salaires ? Gérer automatiquement ? ⇒ on récupère les données d'euro-field
- L'employé peut-il gérer son horaire seul ? Y a t-il un systeme de pointage ? oui mais géré par euro-field
- Pas d'utilisateur responsable rh ? Comment se fait le calcul des salaires ? pas besoin (euro-field)
- Comment fonctionne la facturation des services interne et externe ? Pas claire la réponse (C'est un document standard qui récupèrent les données selon le point suivant pour fournir le justificatif) ⇒ outil dans lequel on rentre les données à facturer et qui génère une facture
- Qui peut s'occuper de la facturation ?
- Pour le calcul des salaires, on peut faire un calcul type sur base de celui belge et adapter les taux en fonction des pays ?
- Si ok, on peut imaginer faire 1 méthode par calcul de salaire et 1 génération de rapport par pays
- Doit-il y avoir une interaction avec la comptabilité ? (injection des frais automatique) (Certainement, il y a un ensemble de coût lié au fonctionnement qui doit apparaitre sur les facturations)
- La base de donnée est dans euro-core ou bien on a une base de donnée propre ? on choisit

manager a acces uniquement au site belgique s'il travaille en belgique

pas d'acces rh

on sait quel service a produit quoi pour quelle recette

3 - 28/05/2024

- mcd de ce qui est déjà fait ? Notamment pour savoir ce qui est en place au niveau des tables de la DB euro-core
- diagramme de package
  - DB management ? Comment représenter les accès vers la db euro-core ? doit on le représenter ou bien sous entendu que c'est géré dans les package bill/employees ?
  - pour authentification, les users sont propres à notre application ou bien on partage des users commun avec les autres applications ?

4 - 31/05/2024

- Ajouter la partie synchronisation dans les scenarios de nos use case ?