Cahier des charges de l'application INFOWARE - Service RH
Partie développement

Logiciel de gestion des salariés de l'entreprise INFOWARE

Sommaire

Le contexte

Organisation cliente : INFOWARE, une entreprise de distribution de matériel informatique.
Prestataire : l'entreprise STESIO, dans laquelle vous êtes stagiaire, est l'ESN (entreprise de services du numérique) mandatée par INFOWARE pour développer son projet.

Le projet du Service RH

INFOWARE-SIRH est la future application de gestion des ressources humaines de l'entreprise INFOWARE.
Elle reposera sur une base de données existante implantée sur un serveur MariaDB.
INFOWARE souhaite initier ce projet en développant une fonctionnalité de gestion des données relatives aux salariés sous la forme d'un CRUD.

Wikipédia : "l'acronyme informatique anglais CRUD (pour create, read, update, delete) (parfois appelé SCRUD avec un "S" pour search) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données".

Un prototype incomplet de l'application existe et vous est fourni. Vous êtes chargés de poursuivre sa mise au point.

Votre tuteur a organisé votre travail en différentes étapes qu'il vous présente sous la forme d'un diagramme des tâches PERT (Cf. "les étapes de la mission").

Contraintes de travail

- Vous travaillez en binôme
- Vous développez en Java avec l'EDI NetBeans.
- Votre code devra respecter les normes et standards définis lors de votre formation.
- Chaque vue (classe JFrame, ou JDialog) gérera les événements relatifs à ses contrôles graphiques.
- Les données seront gérées à l'aide de classes situées dans un paquetage "metier"
- L'accès aux données stockées dans le SGBDR local MariaDB se fera par l'intermédiaire de classes spécialisées dites DAO (Data Access Object).
- Les méthodes des classes DAO fourniront des objets des classes métier.
- Votre tuteur vous demande de rendre compte de votre travail à l'issue de chaque étape sur la plateforme web Moodle.

Consignes de remise :

Pour chaque étape, remettez une archive zip contenant au minimum :

- le projet NetBeans
- le compte-rendu au format PDF

L'archive zip devra être nommée de la façon suivante : G_n°_de_groupe_NOM1_NOM2_E_n°_d_etape.zip.
D'autres productions pourront être ajoutées à l'archive en fonction des demandes propres à chaque étape.

Ressources

- Les travaux se dérouleront sur les plateformes de développement individuelles.
- Des ressources seront fournies sous la forme de fichiers et accessibles depuis la plateforme Moodle :
  - un projet NetBeans initial à compléter, comprenant le script SQL de création de la base de données (dans le paquetage « sql »)
  - des tutoriels pour :
    - gérer une base de données depuis Netbeans (sans passer par PhpMyAdmin) : ouvrir une connexion vers une base de données, requêter, exploiter les résultats de la requête ;
    - apprendre à **ouvrir une seconde fenêtre à partir d'une première en lui transmettant des données ;**

Évaluation du projet

L'évaluation prendra en compte les éléments suivants :

- la présentation du projet en binôme avec test et explication du code ;
- les comptes-rendus d'étape remis sur Moodle.

Activités et compétences du référentiel mises en œuvre

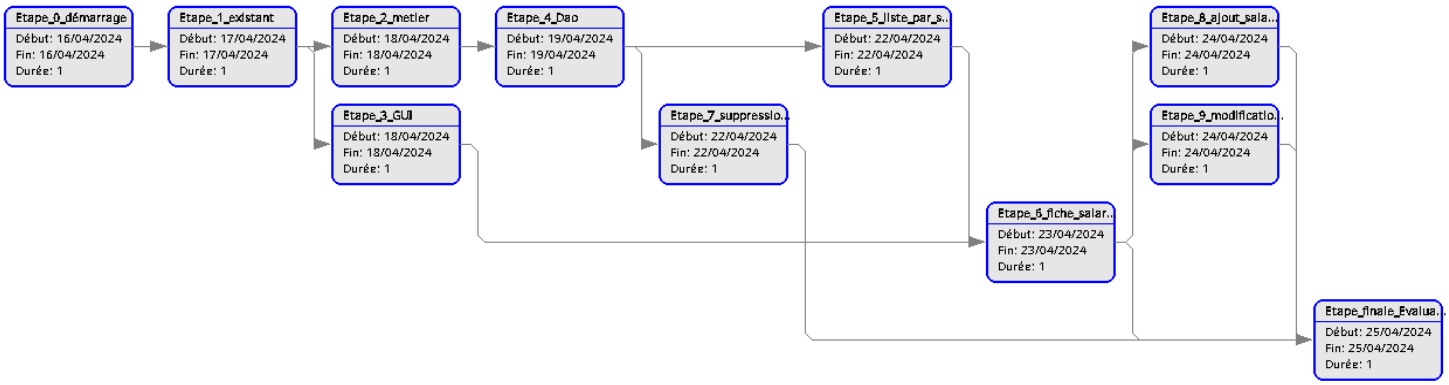
- **BLOC1 - Support et mise à disposition de services informatiques**
  - B1.2 - Répondre aux incidents et aux demandes d'assistance et d'évolution
    - B1.2.3 - Traiter des demandes concernant les applications
  - B1.4 - Travailler en mode projet
    - B1.4.1 - Analyser les objectifs et les modalités d'organisation d'un projet
    - B1.4.2 - Planifier les activités
- **BLOC2_SLAM - Conception et développement d'applications**
  - B2.1_SLAM - Concevoir et développer une solution applicative
    - B2.1.5_SLAM - Identifier, développer, utiliser ou adapter des composants logiciels
    - B2.1.7_SLAM - Utiliser des composants d'accès aux données
    - B2.1.9_SLAM - Réaliser les tests nécessaires à la validation ou à la mise en production d'éléments adaptés ou développés
    - B2.1.10_SLAM - Rédiger des documentations technique et d'utilisation d'une solution applicative
    - B2.1.11_SLAM - Exploiter les fonctionnalités d'un environnement de développement et de tests

Les étapes du projet

Votre tuteur a prévu des étapes intermédiaires pour mener à bien votre mission.
Le diagramme de PERT ci-dessous montre que certaines étapes peuvent être menées en parallèle ;
ne pas tenir compte des dates de début/fin figurant sur les tâches.

Vous n'aurez peut-être pas le temps de réaliser toutes les étapes.

- Etape_0_démarrage
- Etape_1_existant
- Etape_2_metier
- Etape_3_GUI
- Etape_4_Dao
- Etape_5_liste_par_service
- Etape_6_fiche_salarie
- Etape_7_suppression_salarie
- Etape_8_ajout_salarie
- Etape_9_modification_salarie
- Etape_finale_Evaluation



Étape n°0 - constitution des équipes

L'étape zéro consiste à :

- former des binômes ;
- définir et initialiser les outils de travail en commun : partage des fichiers (ex : OneDrive), gestion du projet (ex : Trello), outil de messagerie (ex : Teams), ...

À l'issue de l'étape : communiquez aux enseignants la composition de votre binôme ainsi que le choix de vos outils collaboratifs.

Étape n°1 – prise en main de l'existant

Un projet Netbeans vous a été fourni.

Travail à faire

- Créez la base de données sur le serveur local MariaDB : en tant qu'administrateur, exécutez le script de création de la base de données SIRH (le script SQL est fourni) ; ce script crée également un utilisateur 'sirh_util' (mdp="secret") possédant tous les droits sur la base de données 'SIRH' uniquement. C'est cet utilisateur qui est utilisé par l'application pour la connexion à la BDD (et non 'root').
- Ouvrez le projet puis associez-lui éventuellement les bibliothèques nécessaires ainsi que leurs documentations type javadoc respectives.
- Procédez à un test fonctionnel de cette application pour résoudre les problèmes éventuels et déterminer les fonctionnalités disponibles.

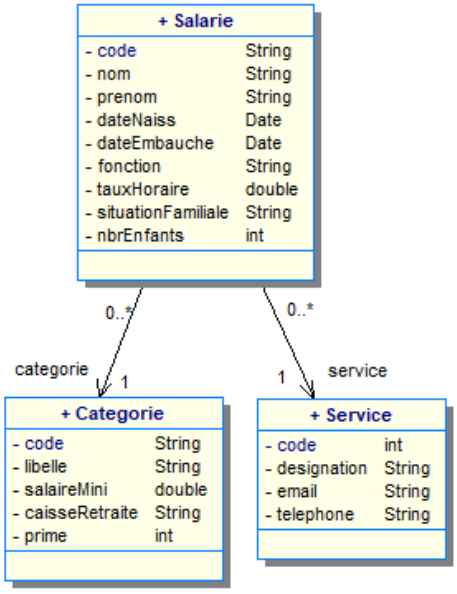
A remettre à l'issue de l'étape

Dans un fichier PDF respectant la nomenclature, remettez un compte-rendu de l'étape comportant les éléments suivants :

- une trace d'exécution du script SQL, une copie d'écran montrant que la base de données a bien été créée et qu'elle est accessible avec l'utilisateur 'sirh_util' ;
- un rapport de test fonctionnel : copie.s d'écran.s montrant les fonctionnalités disponibles, accompagnées de vos commentaires ;

Étape n°2 – compléter les classes métier

Voici un extrait du diagramme des classes métier de l'application établi par votre tuteur :
Les classes Salarie et Service on déjà été codées.



Travail à faire

- Testez les classes métier existantes à l'aide des classes de tests unitaires correspondantes (paquetage test.metier).
- Vous devez coder en java la classe métier Categorie, ainsi que sa classe de test unitaire, et montrer que les tests sont conformes aux attentes.
- Complétez la classe Salarie pour tenir compte de l'association Salarie-Categorie ;
  des erreurs de conception à cette étape pouvant affecter de façon importante la suite de votre développement, il vous est conseillé de montrer votre code à un enseignant avant de commencer l'étape suivante.
- Modifiez la classe de test unitaire de Salarie pour tenir compte de l'évolution de cette classe et exécutez la à nouveau.

Contraintes

- Une classe métier doit fournir (a minima) les méthodes suivantes :
  - un constructeur avec des paramètres permettant de valoriser les attributs ;
  - un accesseur et un mutateur pour chaque attribut ;
  - une méthode toString permettant de contrôler l'état d'une instance de la classe.
- Les classes métier sont regroupées dans un paquetage `modele.metier` .
- Les classes de test unitaire sont regroupées dans un paquetage `test.metier` .

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests unitaires (trace d'exécution des classes de test commentée) ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Étape n°3 - maquettage d'une interface graphique

Cette étape a pour but de concevoir les interfaces graphiques utilisateurs (GUI) de l'application. Afin de vous permettre de comprendre l'utilité de chacune d'elles, on vous fournit le diagramme de cas d'utilisations ainsi que certains scénarios représentatifs du mode d'utilisation de l'application (Cf. "Modélisation fonctionnelle" sur cette page et la suivante).

Travail à faire

Vous devez concevoir l'interface graphique suivante : fenêtre de consultation des données relatives à **un salarié**.

Remarques

- utilisez le concepteur de NetBeans pour concevoir votre maquette ; ainsi, son code Java sera généré et elle sera prête à être intégrée lors des étapes suivantes ;

Contraintes

- les dates seront affichées au format jj/mm/aaaa ;
- les objets graphiques seront nommés conformément aux normes définies lors de votre formation.

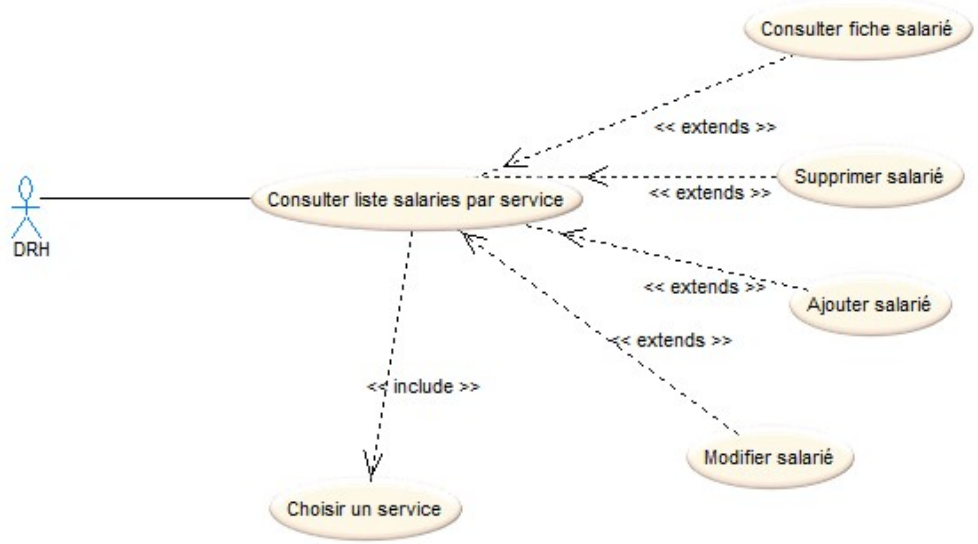
A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ou bien la référence du dépôt de code accessible par les évaluateurs ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - la maquette illustrée avec des exemples de données ;
  - un commentaire sur les choix de conception (types de composants, …) ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Modélisation fonctionnelle

DIAGRAMME DE CAS D'UTILISATION



SCENARIOS

SCÉNARIOS DU CAS D'UTILISATION C1 - "Consulter tous les salariés"

Acteur : DRH

Pré-conditions :

- *la fenêtre de consultation de la liste des salariés est affichée avec la liste de tous les salariés*

C1-SN - Scénario nominal

Post-condition : *néant.*

La liste est affichée et est complète dès l'ouverture de la fenêtre.
Pour chaque salarié, on affiche la désignation de son service.

SCÉNARIOS DU CAS D'UTILISATION C2 - "Consulter les salariés par service"

Acteur : DRH

Pré-conditions :

- *la fenêtre de consultation de la liste des salariés est affichée avec la liste de tous les salariés.*
- *une liste déroulante propose la liste des services, plus une option "tous les services"*

C2-SN - Scénario nominal

Post-condition : *néant.*

Acteur	Système étudié
1-Le DRH sélectionne un service dans la liste déroulante	
	2-L'application recherche et affiche la liste des salariés du service.

Le scénario est terminé.

SCÉNARIOS DU CAS D'UTILISATION C3 - "Supprimer un salarié"

Acteur : DRH

Pré-conditions :

- *la fenêtre de consultation de la liste des salariés est affichée avec la liste des salariés du service actuellement sélectionné*

C3-SN - Scénario nominal

Post-condition : *la liste des salariés est mise à jour.*

Acteur	Système étudié
1-Le DRH sélectionne un salarié dans la liste 2-Le DRH clique sur le bouton « Supprimer »	
	3-L'application affiche une demande de confirmation rappelant l'identité du salarié
4-Le DRH confirme la suppression	
	5-L'application supprime les données relatives au salarié dans la base de données 6 – L'application affiche de nouveau la liste des salariés du service, mise à jour.

Le scénario est terminé.

SCÉNARIOS DU CAS D'UTILISATION C4 - "Consulter fiche salarié"

Acteur : DRH

Pré-conditions :

- *la fenêtre de consultation de la liste des salariés pour un service donné est affichée avec la liste des salariés du service actuellement sélectionné*

C4-SN - Scénario nominal

Post-condition : *néant.*

Acteur	Système étudié
1-Le DRH sélectionne un salarié dans la liste 2-Le DRH clique sur le bouton « Consulter »	
	3-L'application affiche une fenêtre graphique supplémentaire comportant l'ensemble des informations relatives au salarié
4-Le DRH clique sur le bouton « retour »	
	5-L'application affiche de nouveau la fenêtre de consultation de la liste des salariés

Le scénario est terminé.

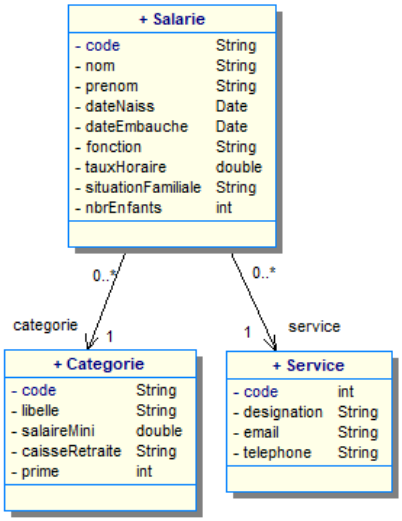
Étape n°4 - classes d'accès aux données (DAO)

Le code source est structuré selon ce qu'on nomme un design pattern. Il est architecturé en couches, chaque couche ayant son propre rôle :

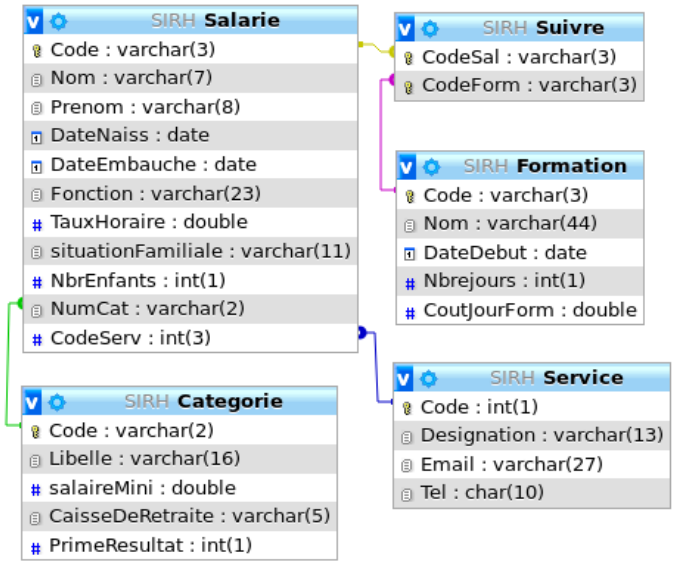
- Interfaces graphiques : interfaces utilisateurs avec contrôles ;
- Classes métiers : structurent les données traitées par le programme (Cf. Diagramme UML des classes métier ci-dessous);
- Classes Dao : c'est une couche intermédiaire entre les classes métier et la base de données ; c'est là que sont exécutées les requêtes SQL
    - Pour chaque classe métier, il existe une classe DAO
    - Les méthodes qui exécutent un SELECT transforment les données récupérées depuis la base de données en objets métier ;
    - Les méthodes qui exécutent un INSERT ou un UPDATE utilisent les objets métier pour créer ou modifier les enregistrements de la base de données ;

Cette étape consiste à de tester les classes d'accès à la base de données (couche DAO).

Diagramme des classes métier actuel (rappel)



Modèle relationnel de la base de données SIRH
(toutes las tables ne sont pas utilisées dans l'application)



A chaque classe DAO, correspond une classe de tests unitaires montrant le fonctionnement normal de ses méthodes sur un jeu d'essai (paquetage `test.dao`). Les classes DaoService, DaoSalarie, et TestDaoService existent déjà.

Travail à faire

- Testez la classe DaoService.
- Codez et mettez au point les classes DaoCategorie et TestDaoCategorie.
- Si nécessaire, adaptez la classe DaoSalarie pour tenir compte de la catégorie du salarié.
- Codez et mettez au point la classe TestDaoSalarie .

Remarques

- Chaque classe DAO fournit (a minima) les méthodes suivantes :
  - une méthode getOneById qui retourne un objet métier d'après son identifiant ;
  - une méthode getAll qui retourne une collection d'objets métier tirés de l'ensemble des enregistrements de la.des table.s correspondante.s.
- Ces classes peuvent être complétées en fonction des besoins : vous pourrez ajouter les méthodes qui vous semblent nécessaires à l'occasion des étapes suivantes.
- Les classes DAO seront regroupées dans un paquetage ***modele.dao*** ; leurs noms portent le préfixe « Dao ».
- Les classes de test unitaire seront situées dans le paquetage ***test.dao****;* leurs noms portent le préfixe « Test».
- Le code permettant la connexion à la base de données se trouve dans une classe commune à toutes les classes DAO, afin d'éviter de le dupliquer : ConnexionBDD (classe fournie).
- Les paramètres de connexion sont fixés dans le fichier "sirhJdbc.proprerties" (fourni). L'application se connecte avec l'utilisateur "sirh_util" qui ne possède de droits que sur ses propres tables.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests unitaires (trace d'exécution des classes de test commentée) ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Étape n°5 - fonctionnalité de consultation des salariés d'un service

Cette étape et les suivantes sont destinées à programmer le comportement des interfaces graphiques utilisateurs conçues à l'étape n°3.

Un ticket de demande d'évolution du code a été émis :

Si on se réfère au scénario du cas d'utilisation « C2-SN - Consulter les salariés par service » (Cf. étape 3), la sélection d'un service dans la liste déroulante doit filtrer la liste des salariés en fonction de l'appartenance de ceux-ci au service.

Or, actuellement, cette fonctionnalité est inopérante.

Travail à faire

Déterminer la cause du problème et y remédier.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests fonctionnels :
    - copies d'écran montrant le comportement de l'application lors du test des scénarios prévus ;
    - interprétation des résultats obtenus ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Étape n°6 - fonctionnalité de consultation des données relatives à un salarié

L'étape n°6 concerne à la fois la fenêtre de consultation de la liste des salariés pour un service donné et la fenêtre de consultation des données relatives à un salarié :
Un bouton de commande “Consulter” doit permettre l'apparition d'une fenêtre affichant les informations sur le salarié actuellement sélectionné. Ces informations proviennent de la consultation de la base de données. Le nom du service et le libellé de la catégorie doivent y figurer.
La description du scénario nominal de ce cas d’utilisation vous est fournie : scénario du cas d'utilisation « C4-N - Consulter fiche salarié » (Cf. étape 3),

Travail à faire
Ajoutez le code nécessaire dans les classes des fenêtres graphiques concernées.

Contraintes
- Les informations affichées sur la fenêtre de consultation des données relatives à un salarié ne sont pas modifiables.
- Quitter la fenêtre de consultation des données relatives à un salarié permet de retourner à la fenêtre initiale (affichage de la liste des salariés du service).

A remettre à l’issue de l’étape
Dans une archive zip respectant la nomenclature :
- le répertoire de votre projet NetBeans ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests fonctionnel :
    - copies d'écran montrant le comportement de l'application lors du test des scénarios prévus ;
    - interprétation des résultats obtenus ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Étape n°7 - fonctionnalité de suppression d’un salarié

L'étape n°7 concerne la fenêtre de consultation de la liste des salariés pour un service donné.

Travail à faire
En vous référant au scénario du cas d'utilisation « C3-SN - Supprimer un salarié » (Cf. étape 2), ajoutez le code nécessaire dans la classe de la fenêtre graphique correspondante.

Contraintes
L’application doit demander confirmation avant de procéder à la suppression effective du salarié.

A remettre à l’issue de l’étape
Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests fonctionnel :
    - copies d'écran montrant le comportement de l'application lors du test des scénarios prévus ;
    - interprétation des résultats obtenus ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Étape n°8 - fonctionnalité d'ajout d'un nouveau salarié

L'étape n°8 concerne à la fois la fenêtre de consultation de la liste des salariés pour un service donné et la fenêtre de consultation des données relatives à un salarié :
Un clic sur le bouton "Ajouter" de la fenêtre de consultation de la liste des salariés provoque l'affichage d'un formulaire de saisie des informations relatives à un nouveau salarié.

SCÉNARIOS DU CAS D'UTILISATION C5- "Ajouter un salarié"

Acteur : DRH
Pré-conditions :

- *la fenêtre de consultation de la liste des salariés pour un service donné est affichée avec la liste des salariés du service actuellement sélectionné*

C5-SN - Scénario nominal
Post-condition : *la liste des salariés du service choisi est affichée et à jour.*

Acteur	Système étudié
1-Le DRH clique sur le bouton « Ajouter»	
	2-L'application affiche une fenêtre graphique supplémentaire comportant un formulaire de saisie vide
3- Le DRH saisit les informations relatives au salarié 4-Le DRH clique sur le bouton « Valider»	
	6-L'application ajoute le salarié à la base de données dans le service 5-L'application affiche de nouveau la fenêtre de consultation de la liste des salariés et la liste est mise à jour

Le scénario est terminé.

Travail à faire

- adaptez les interfaces graphiques en ajoutant les composants (boutons de commande, …) nécessaires ;
- ajoutez le code nécessaire dans les classes des fenêtres graphiques concernées.

Contraintes

- Le service est prérempli et non modifiable sur le formulaire (déjà sélectionné auparavant)
- La catégorie sera choisie dans une liste déroulante.
- La date pourra être saisie à l'aide d'un composant graphique spécialisé de type JCalendar
- Quitter la fenêtre de saisie des données relatives à un salarié permet de retourner à la fenêtre initiale (affichage de la liste des salariés du service) sans ajouter de salarié.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ou bien la référence du dépôt de code accessible par les évaluateurs ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests fonctionnel :
    - copies d'écran montrant le comportement de l'application lors du test des scénarios prévus ;
    - interprétation des résultats obtenus ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).

Étape n°9 - fonctionnalité de modification d'un salarié

L'étape n°8 concerne à la fois la fenêtre de consultation de la liste des salariés pour un service donné et la fenêtre de consultation des données relatives à un salarié :
Un clic sur le bouton "Modifier" de la fenêtre de consultation de la liste des salariés provoque l'affichage d'un formulaire de modification des informations relatives au salarié sélectionné.

SCÉNARIOS DU CAS D'UTILISATION C6 - "Modifier un salarié"

Acteur : DRH
Pré-conditions :

- *la fenêtre de consultation de la liste des salariés pour un service donné est affichée avec la liste des salariés du service actuellement sélectionné*

C6-SN - Scénario nominal
Post-condition : *la liste des salariés du service choisi est affichée et à jour.*

Acteur	Système étudié
1-Le DRH clique sur le bouton « Modifier	
	2-L'application affiche une fenêtre graphique supplémentaire comportant un formulaire de saisie prérempli avec les informations relatives au salarié sélectionné
3- Le DRH modifie les informations relatives au salarié 4-Le DRH clique sur le bouton « Valider»	
	6-L'application met à jour le salarié à la base de données dans le service 5-L'application affiche de nouveau la fenêtre de consultation de la liste des salariés et la liste est mise à jour

Le scénario est terminé.

Travail à faire

- Adaptez les interfaces graphiques en ajoutant les composants (boutons de commande, …) nécessaires ;
- Ajoutez le code nécessaire dans les classes des fenêtres graphiques concernées.

Contraintes

- Le service est modifiable et sera choisi dans une liste déroulante.
- La catégorie sera choisie dans une liste déroulante.
- La date pourra être saisie à l'aide d'un composant graphique spécialisé de type JCalendar
- Quitter la fenêtre de saisie des données relatives à un salarié permet de retourner à la fenêtre initiale (affichage de la liste des salariés du service) sans modifier le salarié.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans ou bien la référence du dépôt de code accessible par les évaluateurs ; le code source est normalisé et commenté ;
- un compte-rendu de l'étape comportant les éléments suivants :
  - vos explications ;
  - un rapport de tests fonctionnel :
    - copies d'écran montrant le comportement de l'application lors du test des scénarios prévus ;
    - interprétation des résultats obtenus ;
  - un bilan de l'étape (fait, non fait, difficultés rencontrées).