Contexte et projet
INFOWARE - SIRH
logiciel de gestion des salariés de l'entreprise Infoware

Le contexte

L'organisation cliente : INFOWARE est une entreprise de distribution de matériel informatique.
Le 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 *INFOWARE - SIRH*

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 Oracle..
INFOWARE souhaite initier ce projet en développant une fonctionnalité de gestion des données relatives aux salariés.

La société STESIO accueille chaque année des étudiants de BTS SIO en stage. Vous êtes actuellement deux stagiaires au sein de l'entreprise et c'est à vous qu'est confiée la demande de la société INFOWARE. Ce premier travail sera l'occasion, pour votre tuteur, de vous évaluer et, ainsi, de vous confier un projet en accord avec les capacités que vous aurez démontrées durant cette première semaine.

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

- Vous avez 2,5 jours pour réaliser le projet
- Vous travaillez en binômes
- 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) contiendra une classe "Ecouteur" interne qui gérera les événements relatifs aux contrôles graphiques de la vue.
- L'accès aux données stockées dans le SGBDR local Oracle se fera par l'intermédiaire de classes DAO.
- Les méthodes des classes DAO fourniront des objets des classes métier.
- Les classes "Ecouteur" utiliseront les données 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.

Horaires de l'entreprise STESIO et planning de la semaine

	Lundi 21/05/2018	Mardi 22/05/2018	Mercredi 23/05/2018	Jeudi 24/05/2018	Vendredi 25/05/2018
8h30- 12h00	**Férié**	• Constitution des équipes • Début du projet	Projet	Fin du projet	9h00-12h00 :bilan, stage, anglais, EDM, CGE
13h30 - 17h00		Projet	Projet	Evaluations individuelles	Libre

Ressources

- Les travaux se dérouleront sur les plateformes de développement individuelles Ubuntu virtualisées.
- Les fichiers fournis (scripts SQL, …) seront accessibles depuis la plateforme Moodle

Evaluation du projet

Chaque étudiant sera évalué individuellement :

- présentation et test du projet, explication du code ;
- dossiers remis sur Moodle par binôme.

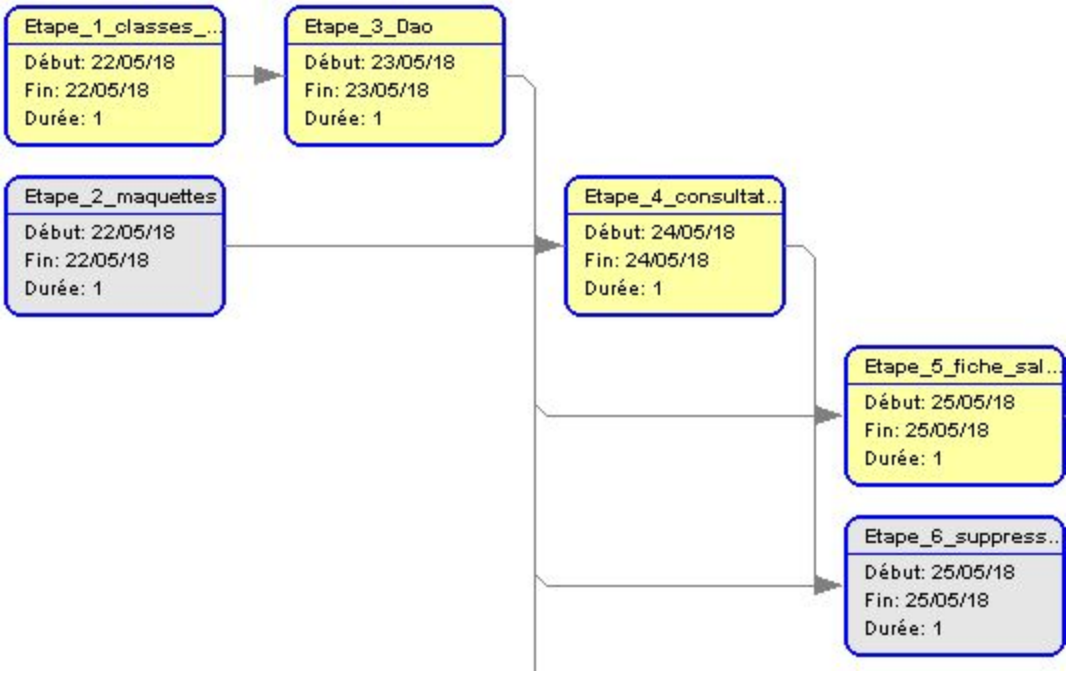
Activités et compétences du référentiel mises en œuvre

- P1 : Production de services
  - A1.1.1 Analyse du cahier des charges d'un service à produire
    - C1.1.1.2 Identifier les fonctionnalités attendues du service à produire
  - 1.4.1 Participation à un projet
    - C1.4.1.2 Rendre compte de son activité
- P4 - Conception et maintenance de solutions applicatives
  - A4.1.2 Conception ou adaptation de l'interface utilisateur d'une solution applicative
    - C4.1.2.2 Maquetter un élément de la solution applicative
  - A4.1.3 Conception ou adaptation d'une base de données
    - C4.1.3.4 Manipuler les données liées à la solution applicative à travers un langage de requête
  - A4.1.7 Développement, utilisation ou adaptation de composants logiciels
    - C4.1.7.1 Développer les éléments d'une solution
    - C4.1.7.2 Créer un composant logiciel
    - C4.1.7.4 Utiliser des composants d'accès aux données
  - A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés
    - C4.1.8.1 Élaborer et réaliser des tests unitaires
  - A4.1.10 Rédaction d'une documentation d'utilisation
    - C4.1.10.1 Rédiger la documentation d'utilisation, une aide en ligne, une FAQ

Les étapes de la mission

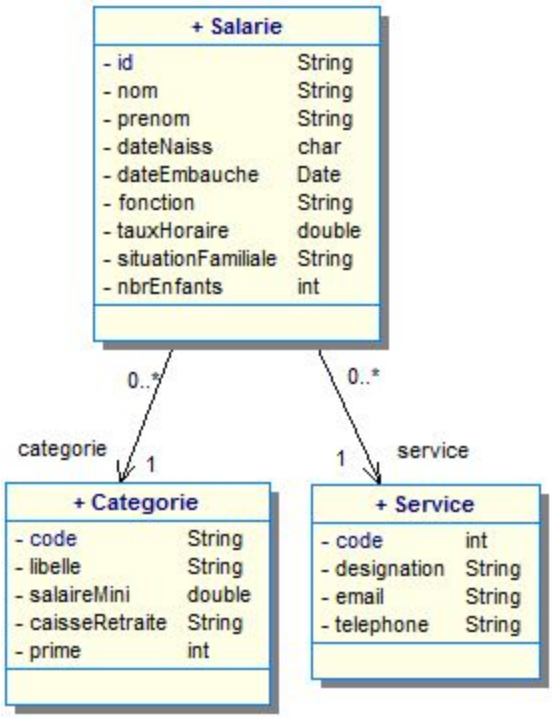
Vous êtes chargés de mettre au point un CRUD des salariés de l'entreprise.
Wikipédia : "l'acronyme informatique anglais CRUD (pour create, read, update, delete) (parfois appelé SCRUD avec un "S" pour search) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage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

Votre tuteur a prévu des étapes intermédiaires pour mener à bien votre mission.
Certaines étapes peuvent être menées en parallèle.



Étape n°1 - classes métier

Voici un extrait du diagramme des classes métier de l'application :



Travail à faire

- Vous devez coder en java les classes métier du diagramme ci-dessus, ainsi que des classes de test unitaire montrant que ces classes fonctionnent correctement.
- Des erreurs de conception à cette étape pouvant affecter de façon importante la suite de votre développement, il vous est conseillé de montrer le code de vos classes à un enseignant avant de commencer l'étape n°3.

Contraintes

- Chaque 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` .

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans contenant le code 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°2 - maquettage des interfaces graphiques

Cette étape a pour but de concevoir les interfaces graphiques utilisateurs 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 deux scénarios représentatifs du mode d'utilisation de l'application (Cf. "Modélisation fonctionnelle" sur les deux pages suivantes).

Travail à faire

Vous devez concevoir les interfaces graphiques suivantes :

- fenêtre de consultation de la liste des salariés pour un service donné ;
- fenêtre de consultation des données relatives à un salarié.

Remarques

- utilisez le concepteur de NetBeans pour concevoir vos maquettes ; ainsi, elles seront prêtes à intégrer aux étapes suivantes ;
- il n'est pas demandé de concevoir l'authentification du DRH.

Contraintes

- si vous utilisez un composant JTable en consultation, les cellules ne doivent pas être éditables
- les dates seront affichées au format jj/mm/aaaa ;
- les objets graphiques seront nommés conformément aux normes définies lors de votre formation.

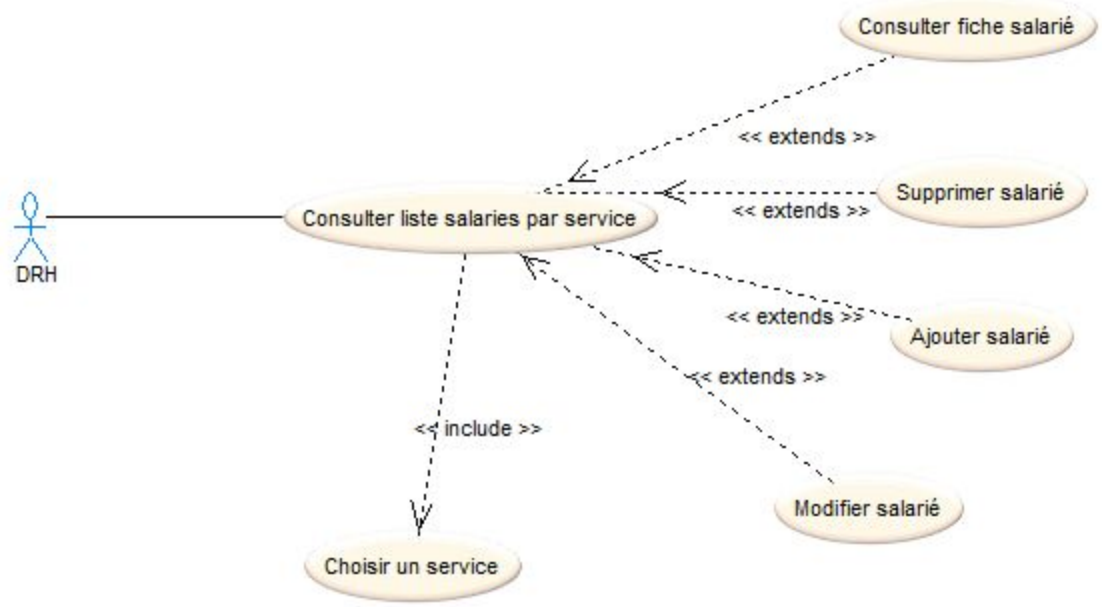
A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans contenant la définition des classes des fenêtres graphiques ;
- un compte-rendu de l'étape comportant les éléments suivants :
    - les maquettes illustrées 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



SCÉNARIO NOMINAL DU CAS D'UTILISATION C1 - "Consulter les salariés par service"
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 premier service*

C1-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.

SCÉNARIO NOMINAL DU CAS D'UTILISATION C3 - "Supprimer 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*

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.

Étape n°3 - classes d'accès aux données (DAO)

Il s'agit de programmer l'accès à la base de données.

Travail à faire

- Créez la base de données sur le serveur local Oracle (service oracle-xe). En tant qu'administrateur, créez un nouvel utilisateur INFOWARE en lui attribuant les rôles « CONNECT » et « RESOURCE » ; puis, dans le schéma de cet utilisateur, exécutez le script de création des tables (les scripts SQL sont fournis).
- Pour chaque classe métier, vous devez coder en java une classe DAO (Data Access Object) qui permet de faire le lien entre les enregistrements de la base de données relationnelle et les objets métier instanciés.
- A chaque classe DAO, est associée une classe de test unitaire montrant le fonctionnement normal de ses méthodes sur un jeu d'essai.

Contraintes

- Chaque classe DAO doit fournir (a minima) les méthodes suivantes :
    - une méthode getOneById qui retourne un objets métier d'après son identifiant ;
    - une méthode getAll qui retourne une collection d'objets métier tirés de l'ensemble des enregistrements de la (des) tables correspondante(s).
- Vous pouvez éventuellement anticiper les besoins et ajouter les méthodes qui vous semblent nécessaires aux 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* .
- Le code permettant la connexion à la base de données doit être regroupé dans une classe commune à toutes les classes DAO, afin d'éviter de le dupliquer.
- L'application doit se connecter avec l'utilisateur INFOWARE qui ne possède de droits que sur ses propres tables.

Remarques

Il est conseillé (mais pas imposé) d'utiliser NetBeans pour gérer la base de données (mode opératoire vu en TP SLAM2). En effet, l'usage simultané de deux applications gourmandes en ressources comme NetBeans et SqlDeveloper risque de ralentir beaucoup votre machine virtuelle.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans contenant le code 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°4 - 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°2 en y intégrant des classes internes de type "Listener" qui utilisent le code des classes DAO et métier.
L'étape n°4 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 « C1-SN - Consulter les salariés par service » (Cf. étape 2), ajoutez le code nécessaire à la classe de la fenêtre graphique correspondante.

Contraintes

- Vous devez ajouter une classe interne permettant d'écouter les événements et d'y réagir.
- La liste des services est remplie à l'instanciation de la classe, d'après le contenu de la base de données.
- Quitter la fenêtre met fin à l'application.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans contenant le code 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°5 - fonctionnalité de consultation des données relatives à un salarié

L'étape n°5 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 (page suivante).

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

Contraintes
- Le code événementiel d'une classe graphique doit figurer dans classe interne.
- Les informations affichées 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 contenant le code 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).

SCÉNARIO NOMINAL DU CAS D'UTILISATION C2 - "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*

C2-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°6 - fonctionnalité de suppression d'un salarié

L'étape n°6 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 à la classe de la fenêtre graphique correspondante.

Contraintes

Le code événementiel d'une classe graphique doit figurer dans classe interne.

A remettre à l'issue de l'étape

Dans une archive zip respectant la nomenclature :

- le répertoire de votre projet NetBeans contenant le code 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).