Travail de fin d’études

**Projet Java 18-19**

Présenté par

**Sean Prévot, Christopher Denis, Luca Genco**

En vue de l’obtention du grade de

**Bachelier en informatique de Gestion**

Introduction

Le projet sur lequel porte cette analyse est une application de direction des ressources humaines, elle devra être fonctionnelles selon les demandes de l’utilisateur final ainsi qu’assez design et simple pour pouvoir être utilisé par un Utilisateur non informaticien.

Voici donc l’analyse liée à ce projet et rédigé par les l’équipe de programmation.

Analyse

1.1 Choix d'un nom pour la nouvelle application

Le nom choisi par notre groupe est « Twinkle ».

1.2 Définition du système

Le directeur des DRH nous demande de lui écrire une application de bureau permettant la gestion de son personnel.

Il s’en servira pour encoder les informations des personnes qu’il engage ainsi que pour leur évaluation en fin d’année. Il sera le seul à l’utilisé (pas dispo sur le web) et sa base de données se trouvera en local.

Le système permettra au DRH d’ajouter de nouveaux employés ainsi que de leur préciser sa fonction. Il permettra aussi d’afficher une liste triée de ces dit employées ainsi que d’afficher uniquement les employés appartenant à une certaine fonction.

Le DRH ajoutera, via le système, une note à chaque employé pour son évaluation annuelle de ses compétences sur les 4 types : Techniques, Organisationnelles, Relationnelles\_&\_Sociales et adaptation.

* 1. Glossaire des terminologies

Scénario nominale : Un événement qui se passe comme on l’a prévu au départ.

Scénario alternatif : Un évènement ne se passe pas comme on l’a prévu dans le programme.

Ergonomie : Facilité d’utilisation du programme, maniabilité du programme.

Acteur : Personne concerné par la fonctionnalité.

Besoin en IHM : Interface graphique nécessaire pour une fonctionnalité.

Précondition : Condition nécessaire avant l’exécution du programme.

Base de données : Serveur informatique qui stocke des données pour une application.

Local : Serveur local disponible sur un pc afin de procéder à des tests (Base de données local).

2. Décrire et énumérer les exigences de l’utilisateur (les « requirements »)

R1 – Ajouter un employé : le DRH doit être capable d’ajouter des employés sur base d’informations.

R2 – Préciser une fonction : le DRH doit avoir la possibilité de préciser la fonction d’un employé. Un employé ne pourra occuper qu’une seule fonction à la fois. Une fonction peut être occupée par plusieurs employés.

R3 – Lister les employés : le DRH doit pouvoir lister les différents employés enregistrés.

R4 – Trier sur nom : le DRH doit pouvoir trier la liste (voir R3) sur le nom.

R5 – Trier sur prénom : le DRH doit pouvoir trier la liste (voir R3) sur le prénom.

R6 – Trier sur date d’engagement : le DRH doit pouvoir trier la liste (voir R3) sur la date d’engagement.

R7 – Ajouter une évaluation : le DRH doit pouvoir ajouter une évaluation sur les différentes compétences de chaque employé (allant de 1 à 10).

R8 – Modifier une évaluation : le DRH doit pouvoir modifier une évaluation d’une compétence de chaque employé (allant de 1 à 10).

R9 – Afficher les employés par fonction : le DRH doit pouvoir sélectionner une fonction et afficher les employés occupant cette fonction.

R20 – Authentification : le DRH n’aura pas là possibilité/nécessité de s’authentifier pour utiliser l’application (il sera le seul à l’utiliser).

R21 – Date d’évaluation : la date d’évaluation ne devra pas être mémorisée.

R22 – Supprimer une évaluation : le DRH ne pourra pas supprimer une évaluation.

R23 – Modifier un employé : le DRH ne pourra pas modifier les informations d’un employé.

R24 – Supprimer un employé : le DRH ne pourra pas supprimer un employé.

R25 – Fonctions et compétences : le DRH ne pourra pas supprimer, modifier ou ajouter des fonctions et/ou des compétences à la base de données.

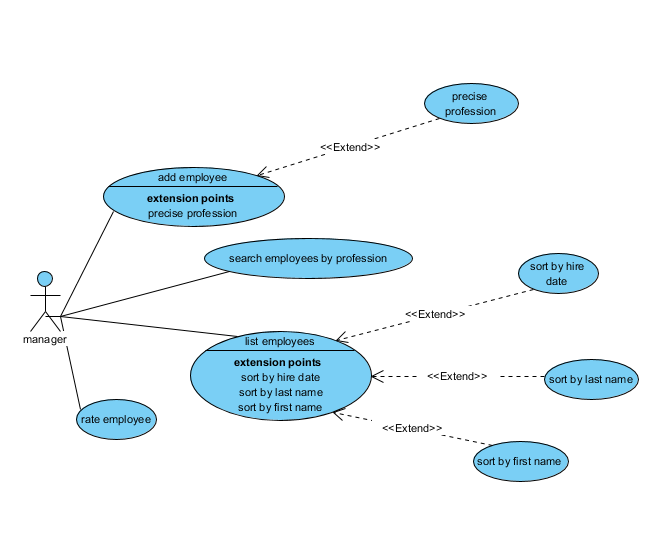
R26 – Rémunérations : les montants des rémunérations ne seront pas mémorisés.

3. Décrire et énumérer les cas d'utilisation par itération

Add employee : Cette fonctionnalité permet de créer un nouvel employé dans le système.  
list employee : Cette fonctionnalité permet de lister les employés existants dans le système.  
rate employee : Cette fonctionnalité permet d’évaluer un employé sur ses compétences.  
search by profession : Cette fonctionnalité permet de lister des employés selon leur fonctions dans l’entreprise.  
sort by lastname : Cette fonctionnalité se trouve dans l’onglet « lister employé » et permet de trier les employé sur leur nom.

sort by firstname : Cette fonctionnalité se trouve dans l’onglet « lister employé » et permet de trier les employé sur leur prénom.  
sort by hire date : Cette fonctionnalité se trouve dans l’onglet « lister employé » et permet de trier les employé sur leur date d’engagemen

4. Élaborer le diagramme des cas d'utilisation



5. Distribuer les rôles de chaque membre du groupe pour les itérations futures

*Itération 1*

Tout le monde :

1.1 Choix d’un nom pour la nouvelle application.

1.4 Détails des acteurs ainsi que de leurs responsabilités respectives.

5. Distribuer les rôles des membres pour les futures itérations.

6. Dessiner l’interface homme machine.

7. Présenter le modèle complet des Uses Cases.

9. Fournir le diagramme de composants décrivant l’architecture du système.

Sean :

1.2 Définition du système.

8. Fournir un diagramme de classes au niveau conceptuel des classes du domaines.

Luca :

1.3 Glossaire des terminologies.

3. Décrire et énumérer les cas d'utilisation par itération.

4. Élaborer le diagramme des cas d'utilisation.

Christopher :

2. Décrire et énumérer les exigences de l’utilisateur (les « requirements »).

10. Présenter un schéma relationnel (modèle logique) de votre base de données.

*Itération 2*

Tout le monde :

Analyse

Sean :

Couche Métier.

Couche Domaine.

Luca :

Couche Persistance.

Christopher :

Couche Présentation (Home, Add).

Couche Logique.

*Itération 3*

Tout le monde :

Analyse

Sean :

Couche IHM.

Couche logique.

Luca :

Couche métier.

Christopher :

Couche persistance.

*Itération 4*

Tout le monde :

Analyse

Sean :

Couche Persistance.

Luca :

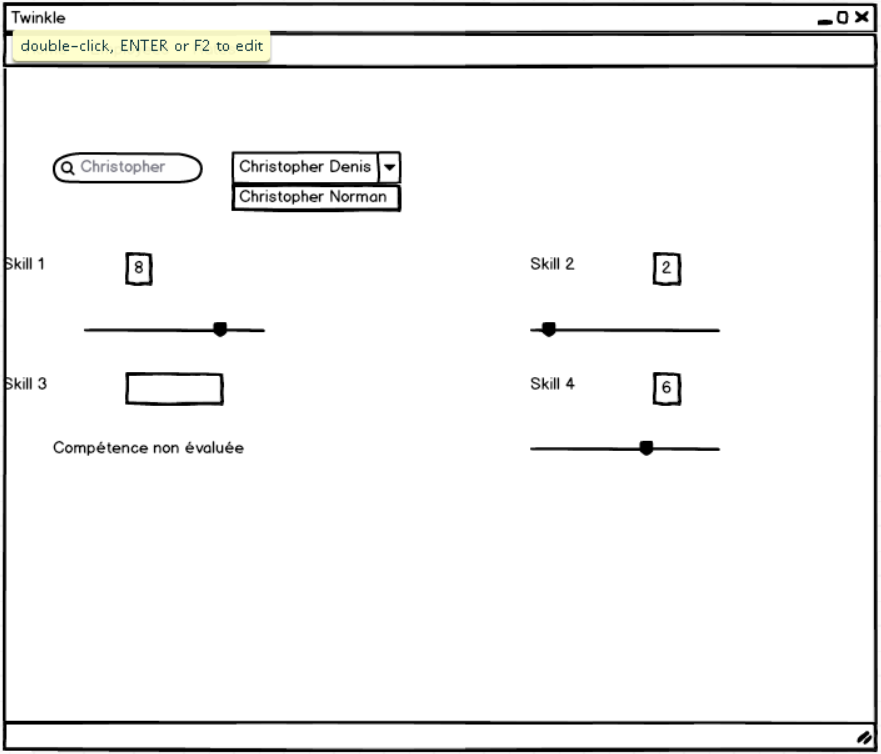
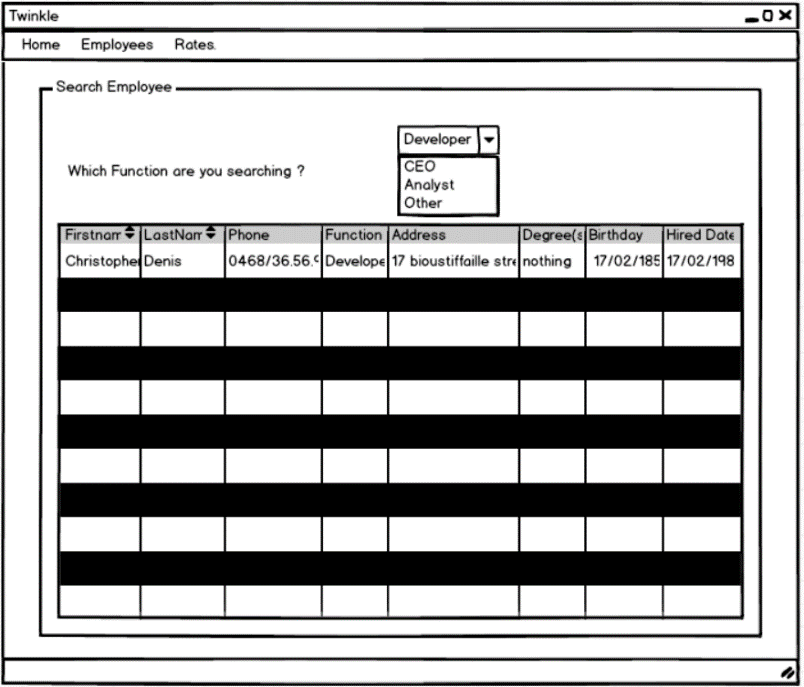
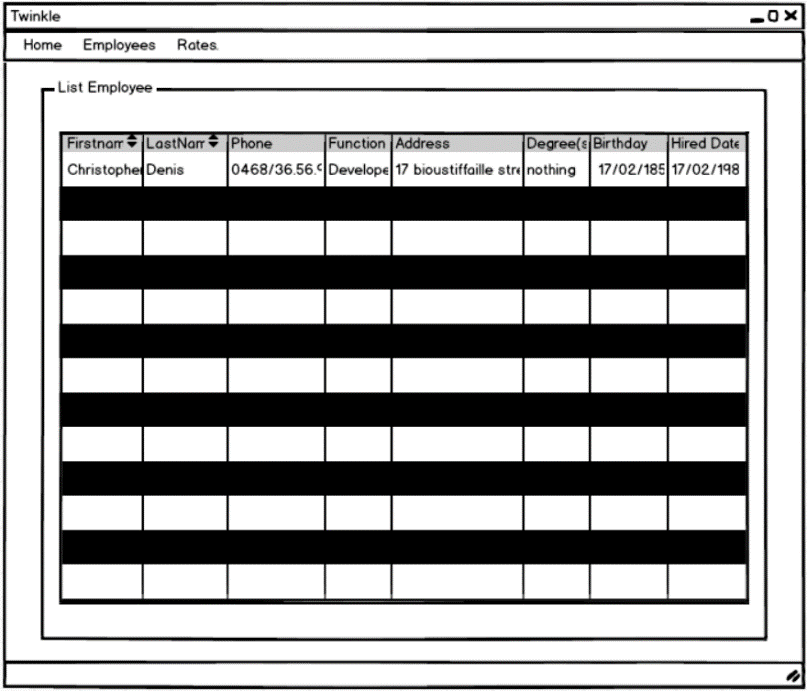
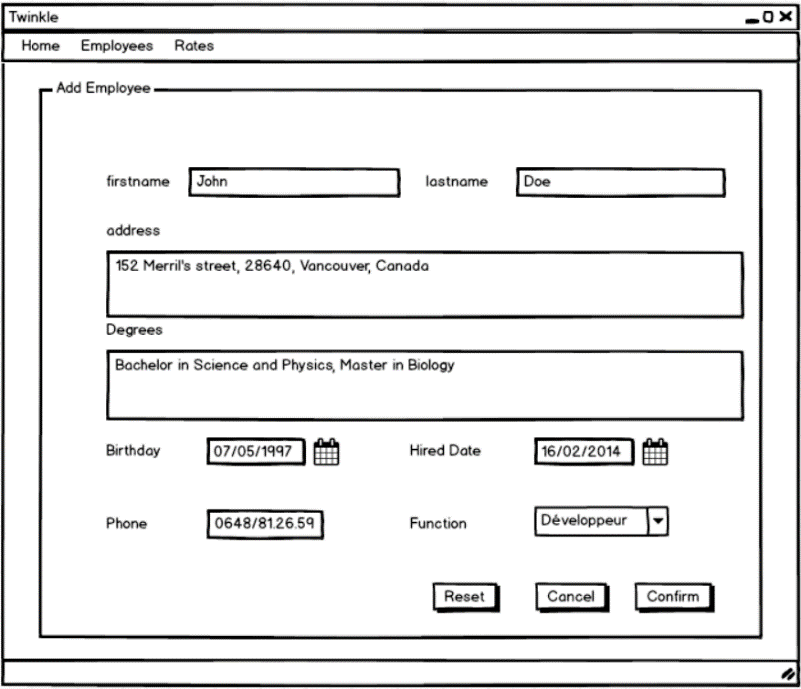
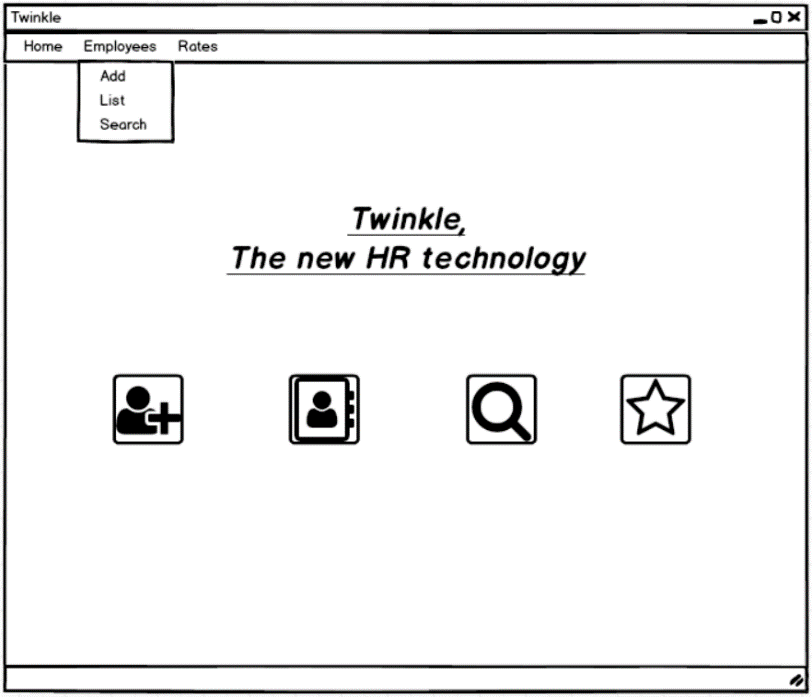
Couche IHM.

Couche Logique.

Christopher :

Couche Métier.

6. Dessiner à l'aide d'un logiciel les prototypes d'interface Homme-Machine de votre application



7. Présenter le modèle complet (scénario principal inclus ainsi que les alternatives) des cas d'utilisation relatifs à l’itération suivante

*Itération 2* :

**Création d’un employé**

Acteur concerné : Manager

Requirement : R1

Auteur : Luca

Précondition : L’utilisateur doit être dans la fenêtre « add employee »

Scénario Nominal :

1. Le RH clique sur l’icône ajouter pour ouvrir le formulaire d’ajout d’employé.
2. Le RH remplis les différents champs.
3. Le RH valide les champs et clique sur le bouton valider.

Scénario Alternatif 1 :

1. Scénario nominal.
2. Le RH ne remplis pas un ou plusieurs champs obligatoires.
3. Le RH clique sur le bouton valider.
4. Fin : Un message d’erreur est affiché signalant que tous les champs sont obligatoires.

Scénario Alternatif 2 :

1. Scénario nominal.
2. Le RH remplis les différents champs.
3. Le RH clique sur le bouton annuler.
4. Fin : Les données sont toujours dans la vue et la fenêtre a été fermé

Ergonomie : Une fois le bouton annuler les données sont toujours dans la vue, il ne faut donc pas retaper les données si un ajout a été déjà encodé.

*Itération 3* :

**Lister des employés**

Acteur concerné : Manager

Requirement : R3

Auteur : Luca

Précondition : L’utilisateur doit être dans la fenêtre « list employee »

Scénario nominal :

1. Le RH clique sur l’icône « lister » pour ouvrir un tableau listant les employés.
2. Il peut trier les employés selon leur prénom, leur nom et la date d’engagement.

Scénario alternatif :

1. Le programme n’est pas ouvert.
2. Il n’y a aucun employé dans la base de données.

Fin : Nominal : La liste s’affiche.

Alternatif : Aucun employé ne s’affiche.

*Itération 3 :*

**Lister les employés selon leurs fonctions**

Acteur concerné : Manager

Requirement : R9

Auteur : Luca

Précondition : L’utilisateur doit être dans la fenêtre « search employee »

Scénario nominal :

1. Le RH clique sur le menu rechercher un employé.
2. L’employé choisi la fonction désirer dans la liste déroulante pour afficher tous les employés qui occupe cette fonction.

Scénario alternatif :

1. Scénario nominal
2. Aucun employé n’existe dans la base de données.

Fin : Nominal : La liste d’employé s’affiche selon la fonction choisie.

Alternatif : La liste des employés est vide, aucun employé n’est affiché.

*Itération 4 :*

**Évaluer l’employé**

Acteur concerné : Manager

Requirement : R7, R8

Auteur : Luca

Précondition : L’utilisateur doit être dans la fenêtre « rate employee »

Scénario nominal :

1. Le RH clique sur l’icône « évaluer ».
2. Une liste déroulante apparait et on saisit l’employé à évaluer ou on le sélectionne.
3. Une fois l’employé sélectionner on évaluer la compétence voulue de 0 à 10.

Scénario alternatif :

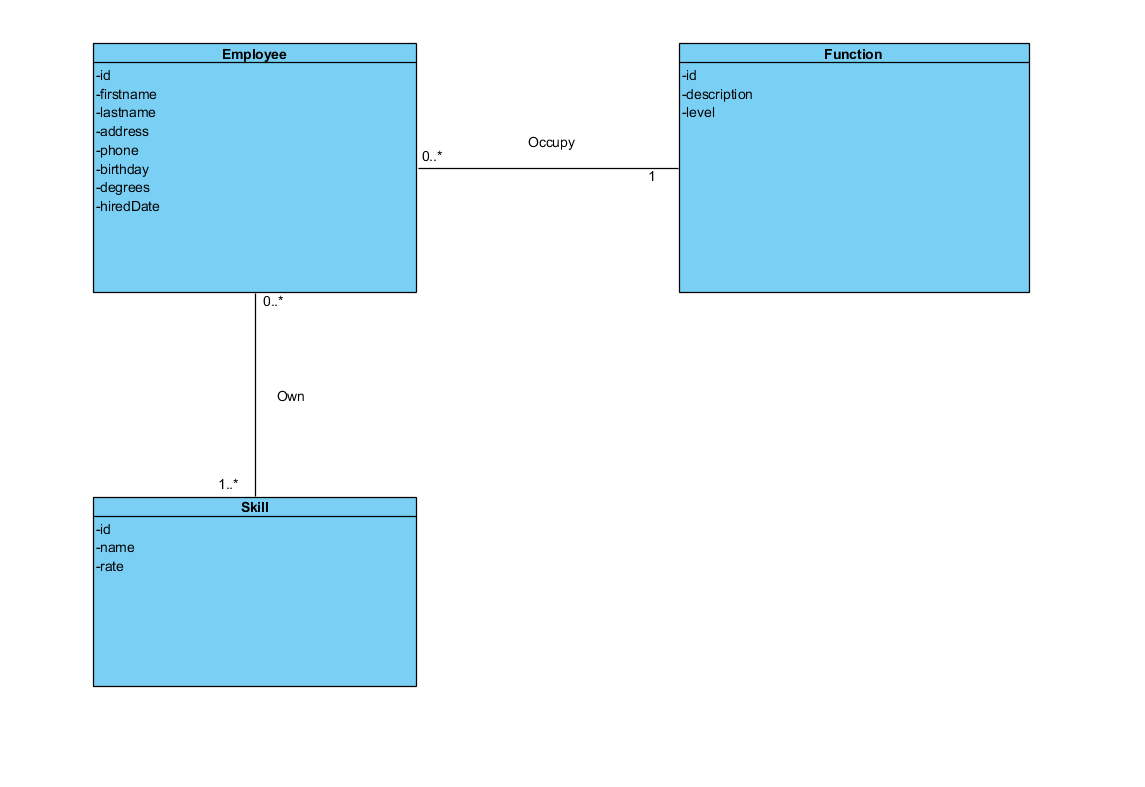
1. La liste d’employé est vide
2. Aucun employé n’a été trouvé

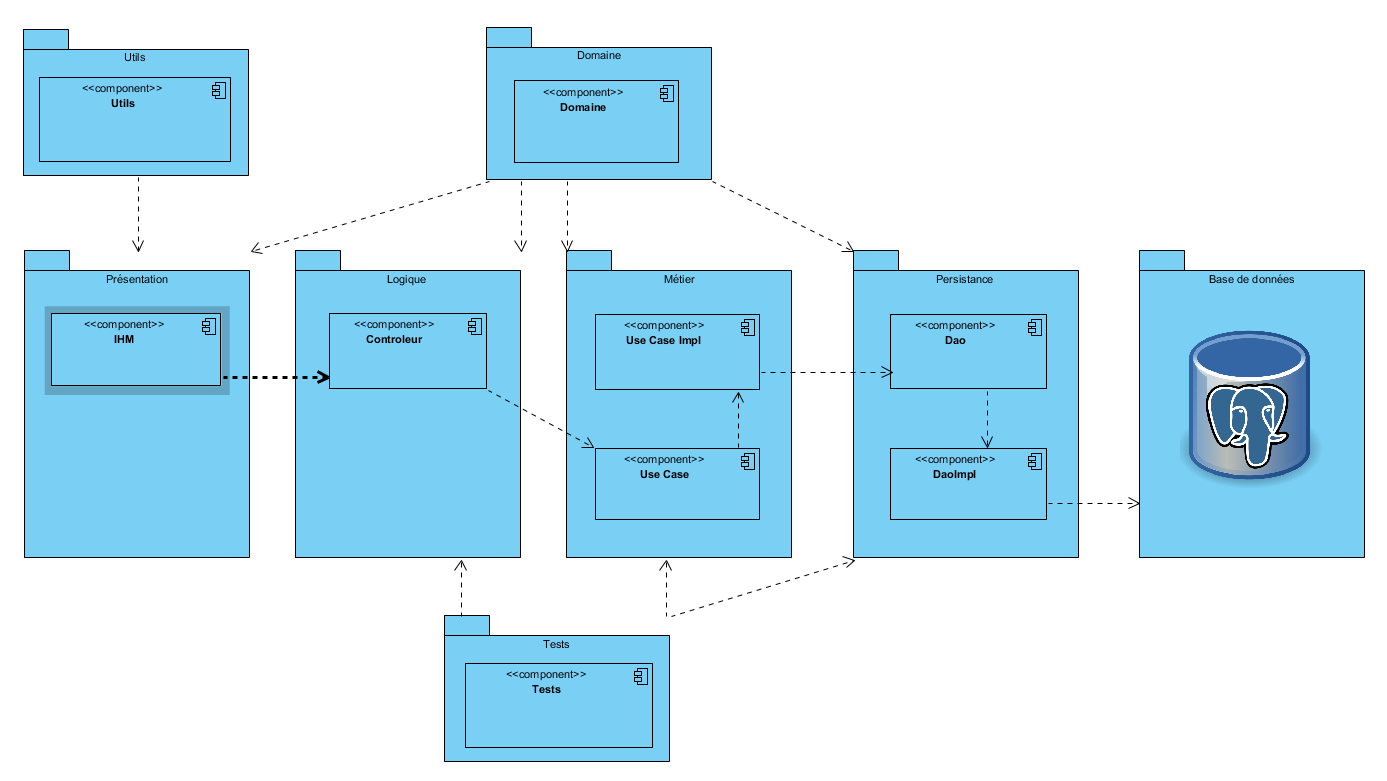
Fin : Nominal : Un employé à bien été trouvé et est bien noté sur la compétence sélectionner, la note est stockée en base de données.

Alternatif : L’employé n’as pas été trouvé. Aucun changement n’a été opéré.

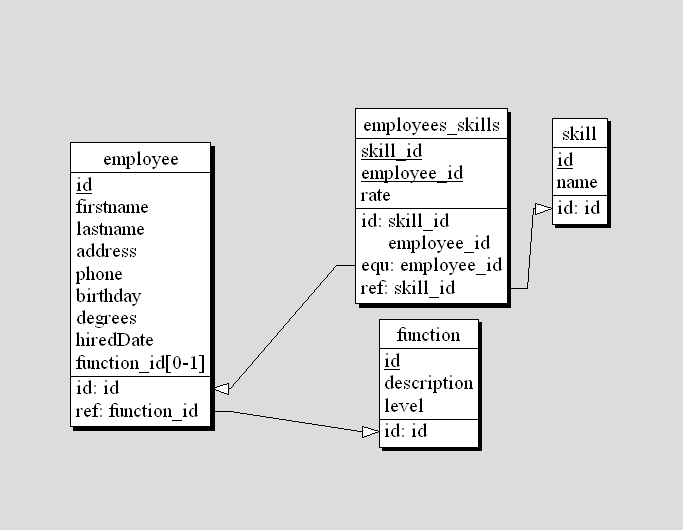
Add employee : Cette fonctionnalité permet de créer un nouvel employé dans le système.  
list employee : Cette fonctionnalité permet de lister les employés existants dans le système.  
rate employee : Cette fonctionnalité permet d’évaluer un employé sur ses compétences.  
search by profession : Cette fonctionnalité permet de lister des employés selon leur fonctions dans l’entreprise.  
sort by lastname : Cette fonctionnalité se trouve dans l’onglet « lister employé » et permet de trier les employé sur leur nom.

sort by firstname : Cette fonctionnalité se trouve dans l’onglet « lister employé » et permet de trier les employé sur leur prénom.  
sort by hire date : Cette fonctionnalité se trouve dans l’onglet « lister employé » et permet de trier les employé sur leur date d’engagement.

8. Fournir un diagramme de classes de niveau conceptuel des classes du domaine ne reprenant que les attributs, pas les méthodes, ni les constructeurs

9. Fournir un diagramme de composants décrivant l’architecture du système. Il doit reprendre les packages de l’application.

10. Présenter un schéma relationnel (modèle logique) de votre base de données



11. Tests fonctionnels

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | DESCRIPTION | PROCÉDURE | RÉSULTAT ATTENDU | RÉSULTAT OBTENU |
| 1 | Ajout employé | Cliquer sur employees puis add, entrer les valeurs suivantes :  Lastname : Laravel, FirstName :Angular,  Address :Rue du Rivage, 29 Moignelée,  Degrees : Informatique de gestion,  Birthday : 02/06/1998,  Hire date : 19/05/2019,  Taper 0498/83.04.33,  Choisir la fonction analyste dans la liste | L’employé à bien été ajouté et le message suivant apparait : « Laravel Angular has been added » |  |
| 2 | Ajout employé | Cliquer sur employees puis add, entrer les valeurs suivantes :  Lastname : Denis, FirstName :Christopher,  Address :Rue du Rivage, 29 Moignelée,  Degrees : Informatique de gestion,  Birthday : 02/06/1998,  Hire date : 19/05/2019,  Taper 0498830433,  Choisir la fonction analyste dans la liste | Le message suivant apparait : « invalid phone format » |  |
| 3 | Lister employés | Cliquer sur employees puis sur list | La liste des employés s’affiche |  |
| 4 | Lister employés | Cliquer sur employees puis sur list | La liste d’employé est vide |  |
| 5 | Rechercher employé | Cliquer sur employees puis sur search, sélectionner une fonction | La liste d’employé s’affiche en fonction de la fonction sélectionnée |  |
| 6 | Rechercher employé | Cliquer sur employees puis sur search, sélectionner une fonction | Aucun employé n’est disponible pour cette fonction |  |
| 7 | Évaluer un employé | Cliquer sur rate, choisir Corentin Simon et évaluer la compétence adaptation avec une valeur de 5 | Le message suivant apparait : « Skill has been evalued » |  |
| 8 | Évaluer un employé | Cliquer sur rate, écrire Denis dans la liste d’employés | Un résultat à bien été trouvé dans la liste déroulante « 2 ~ Denis Christopher » |  |
| 9 | Évaluer un employé | Cliquer sur rate, rechercher Albert dans la liste d’employés | Aucun employé n’a été trouvé dans la liste |  |

12. Descriptions des tests

Nous avons réalisé des tests fonctionnels, unitaires et d’intégrations sur l’ensemble des usecases (addEmployee, ListEmployees, …). Tous les tests se sont déroulés avec succès.

13. Problèmes rencontrés et leurs solutions

Nous n’avons rencontré aucun problème.

14. Présentation de l’application (technique)

14.1. Qu’est-ce qui marche ?

L’ensemble des fonctionnalités demandées fonctionne.

14.2. Qu’est-ce qui ne fonctionne pas ? Pourquoi ?

Rien, tout fonctionne.

14.3. Éventuelles améliorations (design)

* Un design plus professionnel
* Un design plus épuré
* Un deisgn plus flat

14.4. Avis sur le projet (organisation)

Tout le monde a travaillé sur une couche différente lors de chaque itération (tout le monde a travaillé sur chaque couche).

14.5. Impressions

Le projet est certes peu ambitieux, mais il nous a permis de bien comprendre le système de programmation en couche et ce sans devoir négliger certains aspects de l’application

14.6 Regrets

Nous aurions aimé pouvoir ajouter quelques fonctionnalités tels que la suppression et la modification d’un employé.

14.7. Difficultés du projet

Nous n’avons rencontré aucune difficulté. Le projet reprenait ce que l’on a appris en cours et nous avons eu le temps d’ajouter certaines « fonctionnalités » au projet comme un « datepicker ».

14.8. Qu’est-ce qui pourrait améliorer le projet

Pour améliorer le projet, nous pourrions utiliser JavaFX pour implémenter une interface plus développée.

14.9. Temps de travail investi

* Christopher : 26 heures
* Sean : 20 heures
* Luca : 24 heures

Conclusion

Après toute cette Analyse, nous pouvons assurer le projet comme réussi, la team de programmation à fait de son mieux pour fournir un programme design, facile d’utilisation et efficace.

En espérant qu’il vous convienne.