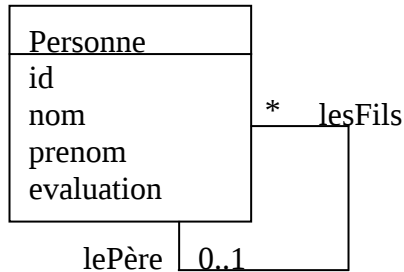


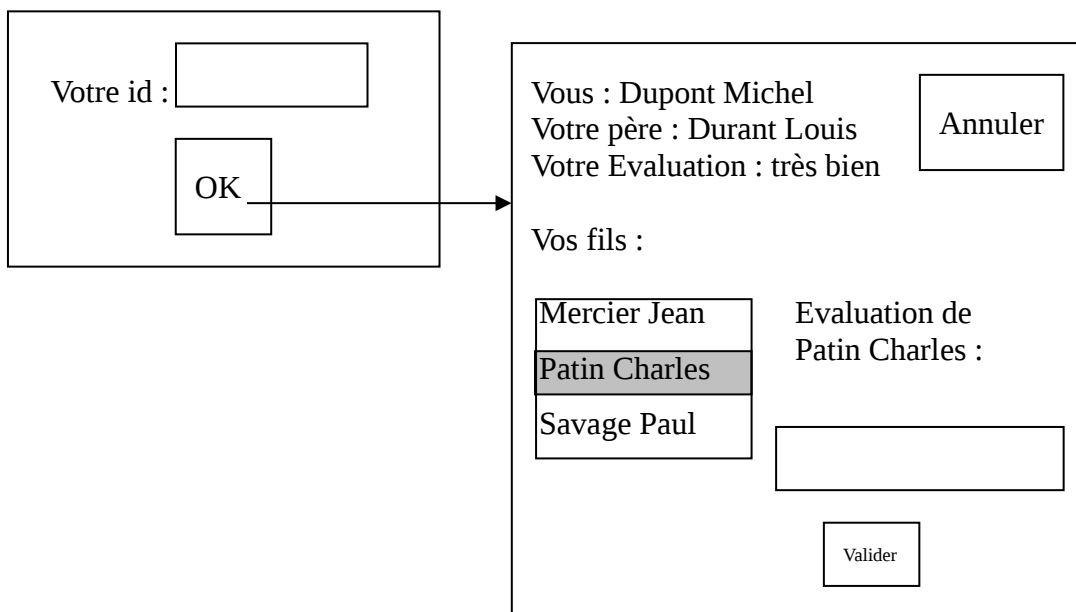
TD Révision COO – M1 MIAGE

Au cours de chaque année, l'activité professionnelle de chaque personnel de l'université doit être évaluée par une personne de l'université appelée son père. Chaque personnel a au plus un père mais peut avoir à évaluer plusieurs personnes : ses fils. On veut construire une application permettant à une personne de consulter son évaluation (faite par son père) et de mettre à jour l'évaluation de ses fils. Une personne est représentée par un identifiant, ses nom et prénom. Une évaluation est une chaîne de caractères. Le modèle du domaine choisit est le suivant :



On suppose que les informations sur les personnels et leurs liens hiérarchiques sont créées et stockées dans une base de données relationnelle de nom « personnels ».

L'application qu'on veut construire est constituée de 2 écrans. Le premier permet à une personne de s'identifier (en donnant son identifiant) et le second permet de consulter les informations de la personne identifiée, le nom et le prénom de son père (quand il existe), son évaluation (quand elle existe) et la liste des noms et prénoms de ses fils (quand ils existent). En sélectionnant une personne dans la liste, la personne identifiée peut alors saisir ou modifier l'évaluation de la personne sélectionnée.



Vous n'avez pas besoin de faire d'interface graphique pour ajouter des personnes (vous pouvez soit les entrer directement dans la base, ou bien faire une fonction d'initialisation).

1. Faire le diagramme de classes UML du domaine.
2. Donner le schéma SQL (en incluant les clés étrangères, et clés primaires)
3. Implémenter la couche Persistance , avec :
 - Remontées paresseuses (avec *Virtual Proxy*, c-à-d qu'à partir d'un objet *Personne* on doit pouvoir accéder de manière transparente à ses pères/fils sans devoir refaire un appel à la Persistance).
 - Écriture en B.D.D. lorsqu'on valide sur le panneau de droite, à l'aide du patron *Unit Of Work*.
4. Implémenter la partie interface graphique avec Java Swing.