

Projet des Parcours GSI et SIE

GSI : Génie des Systèmes d'Information

SIE : Système d'Information d'Entreprise

Rapport jour 3

Sommaire :

- I. Présentation du livrable
- II. Algo de recherche

I. Présentation du livrable

L'objectif de cette partie est d'implémenter la fonctionnalité Recherche dans notre réseau social afin de trouver un nouvel ami. Nous avons décidé de traiter les recherches suivantes :

a) Sur le nom et/ou prénom ; par exemple :

- a. `rechercher("colin", "matthias")`
- b. `rechercher("Colin", null)`
- c. `rechercher(null, "Matthias")`

b) Sur l'étape d'un parcours scolaire ou professionnel par nom, ville et/ou année ; par exemple :

- a. `rechercher("Eisti", "Pau", 2008)`
- b. `rechercher("Eisti", null, 2010)`

d) Sur les amis d'amis ; il s'agit de parcourir les amis d'une personne et proposer les amis de ceux-ci qui ne sont pas déjà des amis ; on peut procéder de manière récursive en parcourant les amis des amis des amis et ainsi de suite ; pour éviter des recherches trop longue et se retrouver avec des amis aux 34ème degré avec qui on a aucune affinité, il faut mettre une limite dans la recherche sous la forme d'un seuil (nombre de résultat, distance franchie par exemple) ; on le traitera de la façon suivante:

- a. vous n'avez pas d'évaluation quantitative ; vous pouvez affecter un poids de 1 à chaque lien d'amitié et trouver les n chemins les plus courts vers les personnes n'étant pas déjà vos amis ;

Les résultats des recherches seront présentés sous la forme d'une liste d'amis potentiels où l'utilisateur pourra piocher pour agrandir son cercle d'amis.

Toutes les recherches et manipulations se feront sur les éléments de la mémoire vive.

Par la suite, nous procéderons à une batterie de tests pour permettre de valider notre recherche sur un réseau social chargé à partir d'une base de données.

II. Algo de recherche

a. Recherche par nom et/ou prénom

```
indice: Entier;  
compteur: Entier;  
resultat: Liste personne;  
reseau: liste personne;  
taille : entier;  
nom : char;  
prenom : char;
```

```
indice = 0;  
compteur = 0;  
taille = taille du reseau;
```

```
Tant que (indice < taille) faire
```

```
    Si (nom == reseau[indice].nom) Et (prenom == reseau[indice].prenom) faire  
        resultat[0].nom = reseau[indice].nom;  
        resultat[0].prenom = reseau[indice].prenom;  
        compteur = 1;  
        Tant que (compteur < taille) Faire  
            resultat[compteur].nom = NULL;  
            resultat[compteur].prenom = NULL;  
            compteur ++;  
        Fin Tant que  
    retourner resultat;
```

```
    Fin Si
```

```
    Si (nom == reseau[indice].nom) Ou (prenom == reseau[indice].prenom) faire  
        resultat[compteur].nom = reseau[indice].nom;  
        resultat[compteur].prenom = reseau[indice].prenom;  
        compteur ++;
```

```
    Fin Si
```

```
    indice ++;
```

```
Fin Tant que  
retourner resultat;
```

b. Recherche par Etablissement et/ou ville et /ou période

```
indice: Entier;
compteur: Entier;
resultat: Liste personne;
resultat2: Liste personne;
reseau: liste personne;
taille : entier;
etablissement : Char;
ville : Char;
periode : Entier;

indice = 0;
compteur = 0;
resultat = NULL;
resultat2 = NULL;
taille = taille du reseau;

Si (etablissement /= NULL) Et (ville /= NULL) Et (periode /=NULL) Faire
    Tant que (indice < taille) faire
        Si ( ville == reseau[indice].parcours.ville) Et (periode ==
reseau[indice].parcours.periode) faire
            resultat[compteur] = reseau[indice];
            compteur ++;
        Fin Si
        indice ++;
    Fin Tant que
    taille = compteur - 1;
    compteur = 0;
    indice = 0;
    Tant que (indice < taille) faire
        Si ( etablissement == resultat[indice].parcours.parcoursScolaire.etablissement) Ou
(etablissement == resultat[indice].parcours.parcoursProfessionnel.entreprise) Faire
            resultat2[compteur] = resultat[indice];
            compteur ++;
        Fin si
        indice ++;
    Fin Tant que
    retourner resultat2;
Fin Si

Sinon Si (etablissement = NULL) Et (ville /= NULL) Et (periode /=NULL)
    Tant que (indice < taille) faire
        Si ( ville == reseau[indice].parcours.ville) Et (periode ==
reseau[indice].parcours.periode) faire
            resultat[compteur] = reseau[indice];
            compteur ++;
        Fin Si
        indice ++;
```

```
        Fin Tant que
        retourner resultat;
    Fin Sinon Si

    Sinon Si (etablissement /= NULL) Et (ville = NULL) Et (periode /=NULL)
        Tant que (indice < taille) faire
            Si ( etablissement == resultat[indice].parcours.parcoursScolaire.etablissement) Et
(periode == reseau[indice].parcours.periode) faire
                resultat[compteur] = reseau[indice];
                compteur ++;
            Fin Si
            indice ++;
        Fin Tant que
        retourner resultat;
    Fin Sinon Si

    Sinon Si (etablissement /= NULL) Et (ville /= NULL) Et (periode =NULL)
        Tant que (indice < taille) faire
            Si ( ville == reseau[indice].parcours.ville) Et ( etablissement ==
resultat[indice].parcours.parcoursScolaire.etablissement) faire
                resultat[compteur] = reseau[indice];
                compteur ++;
            Fin Si
            indice ++;
        Fin Tant que
        retourner resultat;
    Fin Sinon Si

    Sinon
        Tant que (indice < taille) faire
            Si ( ville == reseau[indice].parcours.ville) Ou ( etablissement ==
resultat[indice].parcours.parcoursScolaire.etablissement) Ou (periode ==
reseau[indice].parcours.periode) faire
                resultat[compteur] = reseau[indice];
                compteur ++;
            Fin Si
            indice ++;
        Fin Tant que
        retourner resultat;
    Fin Si

    retourner resultat;
```