II. Éléments de corrections (CAI) :

DOSSIER I : SERVICE DE DISTRIBUTION DU MATÉRIEL INFORMATIQUE

❖ **PLANIFICATION DU PROJET**

I - 10. Diagramme de GANTT. (*voir document réponse DR1, Figure 1*) *(2 pts)*

I - 11. Diagramme de PERT. (voir *document réponse DR1, Figure 2*) *(2 pts)*

I - 12. Calcule des marges totales. *(1,5 pt)*

MT(A)=1-0-1=0 MT(B)=4-1-2=1 MT(C)=2-1-1=0

MT(D)=5-3-1=1 MT(E)=7-4-2=1 MT(F)=7-2-5=0 MT(G)=8-7-1=0

I - 13. Déterminer le chemin critique. (voir *diagramme de PERT*). *(1 pt)*

A-C-F-G.

I - 14. Si la tâche F subit un retard de 1 journée, quel impact aurait-il sur la réalisation et l'avancement du projet. *(1 pt)*

On a MT(F)=0. Donc si la tâche F subit un retard de 1 journée alors le projet sera retardé aussi, et ne se terminera pas dans le délai prévu.

I - 15. les trois facteurs sont : Qualité, temps et le budget. *(0,5 pt)*

❖ **ESTIMATION DE CHARGE**

I - 16. Déterminer la charge de réalisation du projet. *(1,5 pt)*

Le projet comporte 20000 lignes de code. Donc c'est un projet simple.

$C=3,2(20)^{1,05}=74,34\approx 74$ mois/homme

I - 17. Déterminer la durée de réalisation du projet. *(1,5 pt)*

$D=2,5(74)^{0,38}=12,83\approx 13$ mois

I - 18. Déduire la taille moyenne de l'équipe qui doit travailler le projet. *(1 pt)*

Taille de l'équipe=74/13=5,69≈6 hommes

DOSSIER II : SUIVI DE LA CARRIÈRE DES SALARIÉS

II - 1. Trois facteurs qui influencent la qualité d'un logicielle. *(0,75 pt)*

Portabilité
Réutilisabilité
Interopérabilité
Maintenabilité
Flexibilité
Testabilité
Correctude
Fiabilité
Efficacité
Utilisabilité

II - 9. Comparaison entre le test fonctionnel (*boite noire*) et le test structurel (*boite blanche*). *(1,25 pt)*

Le test est l'exécution ou l'évaluation d'un système ou d'un composant par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus.

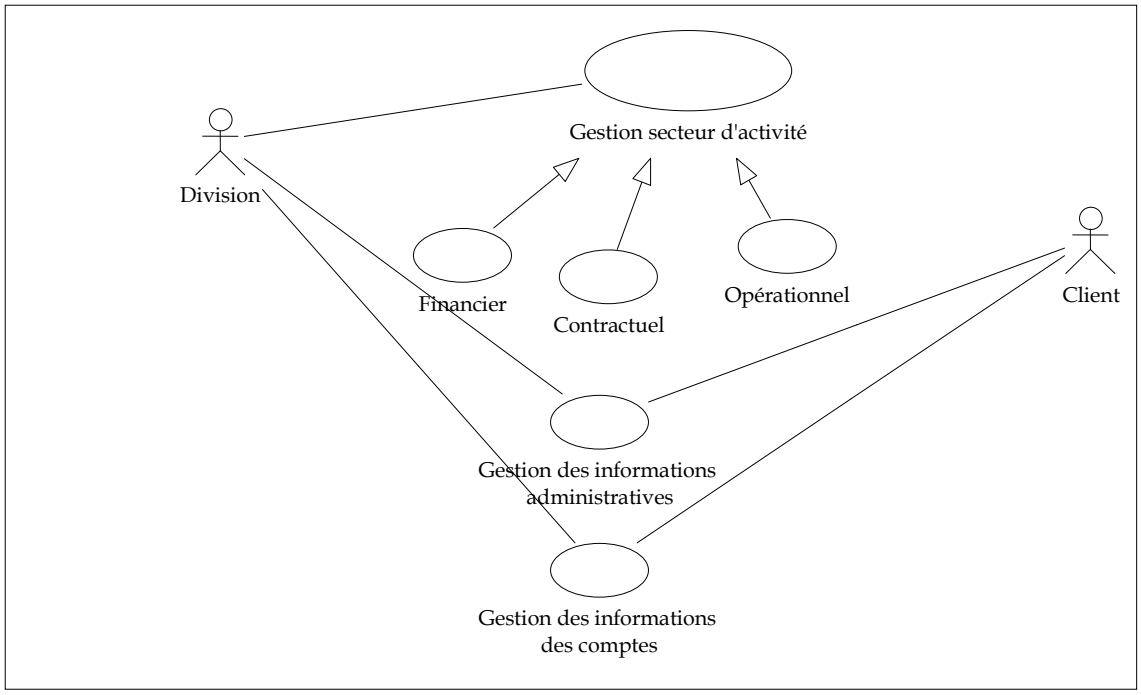
Le test fonctionnel est basé sur les **spécifications** du programme, par contre le test structurel est basé sur **l'analyse** du programme et cela nécessite **le code source** du programme.

LA GESTION DES CLIENTS :

II - 10. Les acteurs qui agissent sur cette partie du système. *(1 pt)*

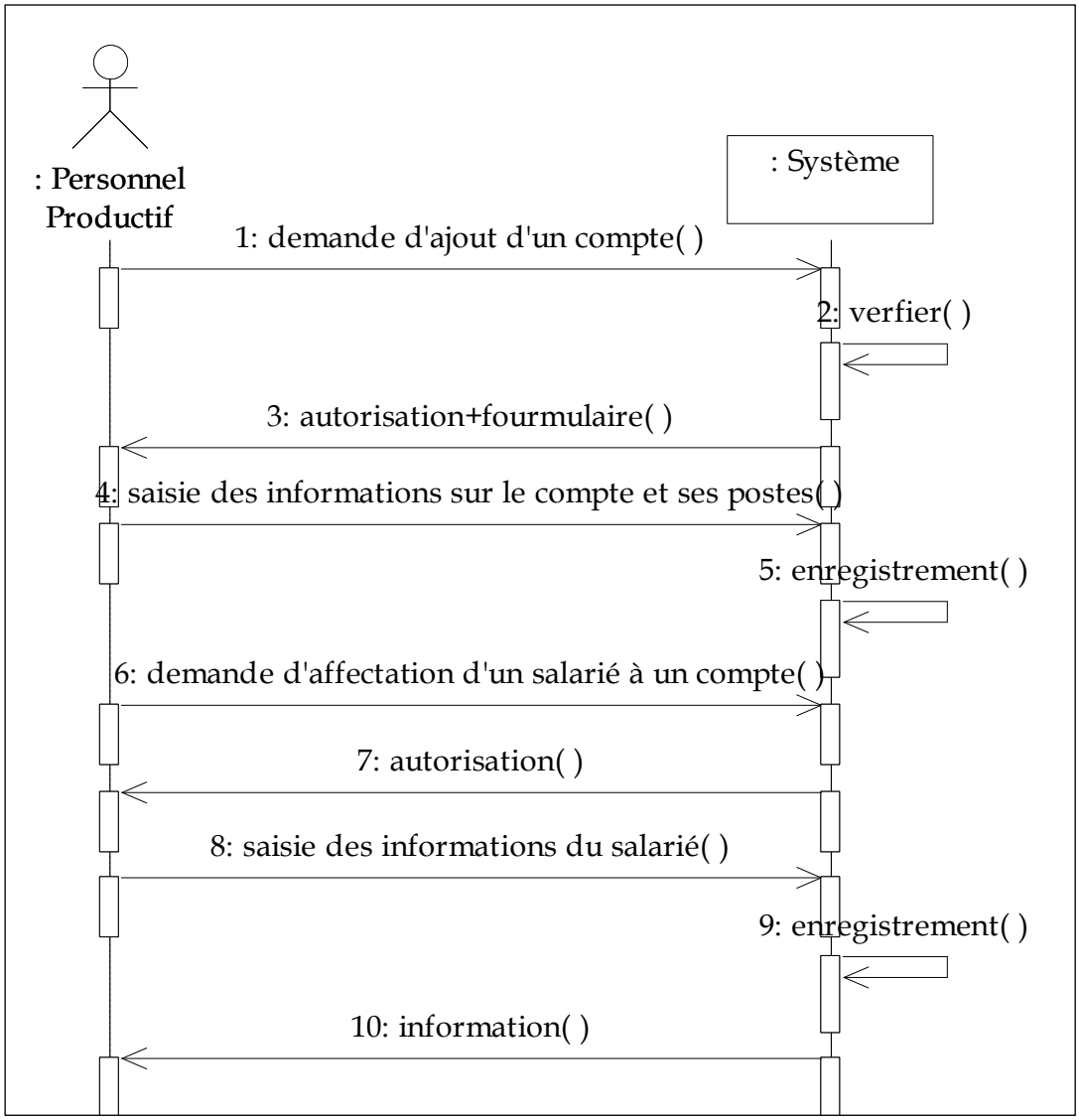
Client et la division.

II - 11. Le diagramme de cas d'utilisation correspondant. *(2 pts)*



LA GESTION DES POSTES :

II - 12. Diagramme de séquence. (*Niveau Système*) *(2 pts)*



LE SUIVI DES QUALIFICATIONS :

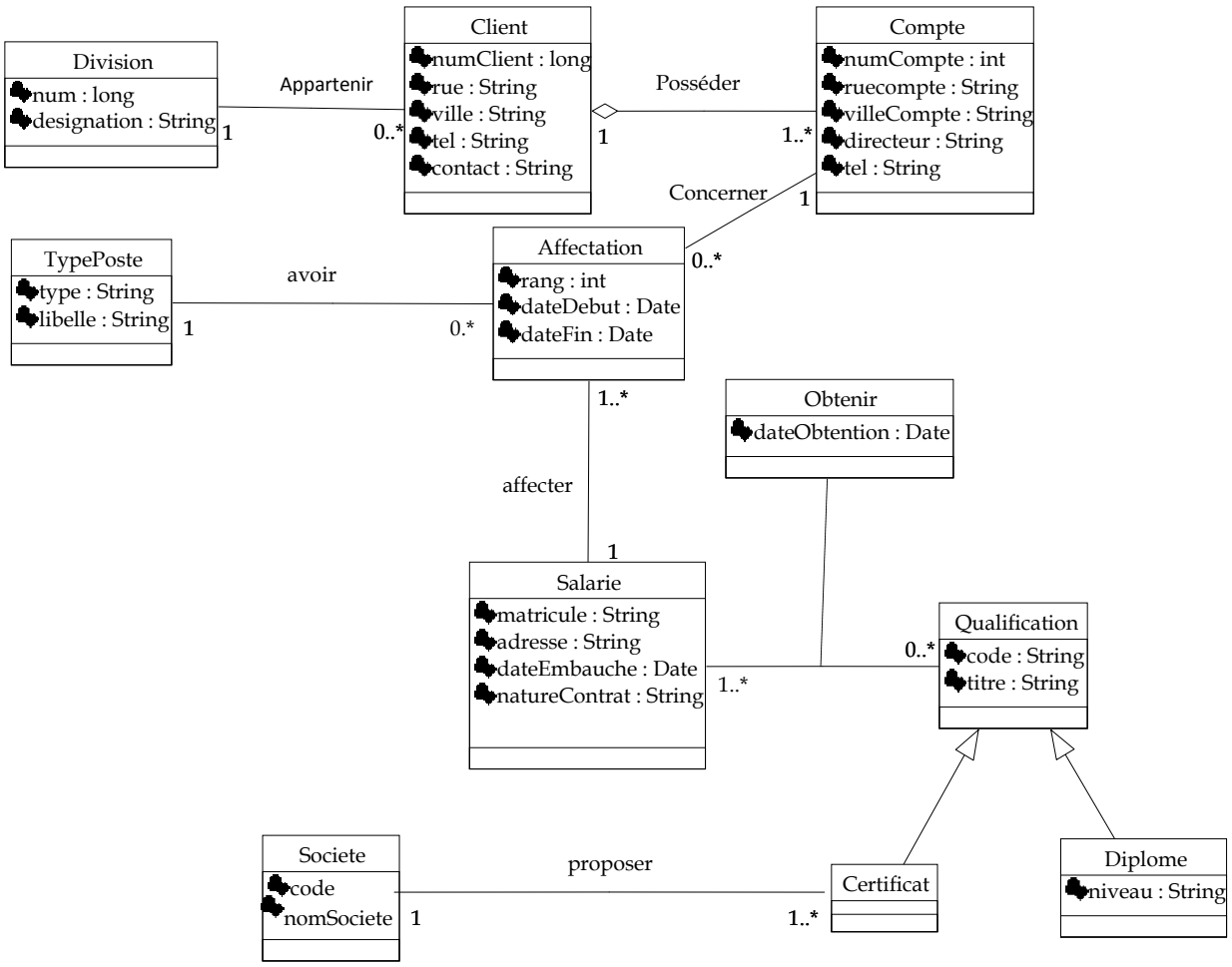
II - 13. Le type d'association qui relie les classes « **COMPTE** » et « **CLIENT** » est une **composition.**

Car la classe « **COMPTE** » n'a aucun rôle sans l'instance de la classe « **CLIENT** ». *(1 pt)*

II - 14. Les cardinalités utilisées entre les classes « **DIVISION** » et « **CLIENT** ». *(1 pt)*

Un client appartient à une et 1 seule division alors qu'une division peut comporter plusieurs clients.

II - 15. Compléter le diagramme de classes répondant aux exigences du système. *(3 pts)*



DOSSIER III : GESTION DES RÉMUNÉRATIONS *(16 pts)*

1. Donner le code en SQL permettant de :

1.1 Créer la base de données **`BD_SALARIÉS`** . (*avec valeurs par défauts*) *(1 pt)*

```
Create database BD_SALARIÉS;
```

1.2 Créer les 3 tables de cette base (*respecter toutes les contraintes*). *(3 pts)*

```
Create table T_SALARIE(Matricule Int Primary Key,Nom varchar(50),
     Prenom varchar(50), Date_Naissance Date,Date_Embauche Date,
     Salaire_Annuel real,Type_Poste varchar(40));

Create table T_EVALUATION(Matricule int,Date_Evaluation Date,
     Niveau Varchar(50),Augm_Prop real,Prime_Prop real,
      Constraint PK_Eval PrimaryKey (Matricule,Date_Evaluation),
      Constraint FK_S ForeignKey (Matricule)References T_SALARIE(Matricule));

Create Table T_EVOLUTION(Matricule Int,Date_Mise_A_Jour Date,
      Ancien_Salaire_Ann real,Pourc_Augm Real,Prime real,Motif varchar(100),
      Constraint PK_Evo PrimaryKey (Matricule,Date_Mise_A_Jour),
      Constraint FK_SalForeignKey (Matricule)References T_SALARIE(Matricule));
```

2. Formuler en SQL la requête permettant d'obtenir les informations suivantes :***salaire annuel maximal, salaire annuel minimal et moyenne des salaires annuels par type de poste***. *(2 pts)*

```
SELECT  Type_Poste,Max(Salaire_Annuel),Min(Salaire_Annuel),
        Avg(Salaire_Annuel)
FROM    T_SALARIE
GROUPBY Type_Poste;
```

3. Expliquer le rôle de la requête suivante : *(1 pt)*

```
SELECT      Type_Poste, Count(*)
FROM  T_SALARIE
GROUP BY Type_Poste
HAVING  Count(*)=(SELECT Max(Count(*))
        FROM T_SALARIE
        GROUP BY Type_Poste);
```

La requête affiche le (ou les) type du poste et le (ou les) nombre de salarié du (ou des) poste qui compte le plus grand nombre de salariés.

4. Créer en SQL la fonction « **`F_Nbe_Salariés`** » permettant d'obtenir le nombre de salariés ayant plus d'un nombre d'ans, donné en paramètre, dans l'année et dont l'évaluation en **`2014`** est « **`faible`** ». *(2,5 pts)*

```
CREATE FUNCTION F_Nbe_Salariés(@annee int)
RETURNSintAS
BEGIN
       DECLARE @Nombre int
       SELECT @Nombre=Count(Matricule) FROM T_SALARIE
       WHERE (YEAR(GetDATE())-YEAR(Date_Naissance))> @annee
       AND    Matricule IN(SELECT Matricule FROM T_EVALUATION
       WHERE YEAR(Date_Evaluation)=YEAR(GETDATE())AND Niveau='faible')
       Return @Nombre
END
GO
```