



Filière :	Développement des Systèmes d'Information - DSI
Épreuve :	Conception des Applications Informatiques - CAI

Durée :	4 heures
Coefficient :	50

## ÉTUDE DE CAS : GESTION DU RÉSEAU D'EAU POTABLE

### DOSSIER I : HÉBERGEMENT DU SITE WEB

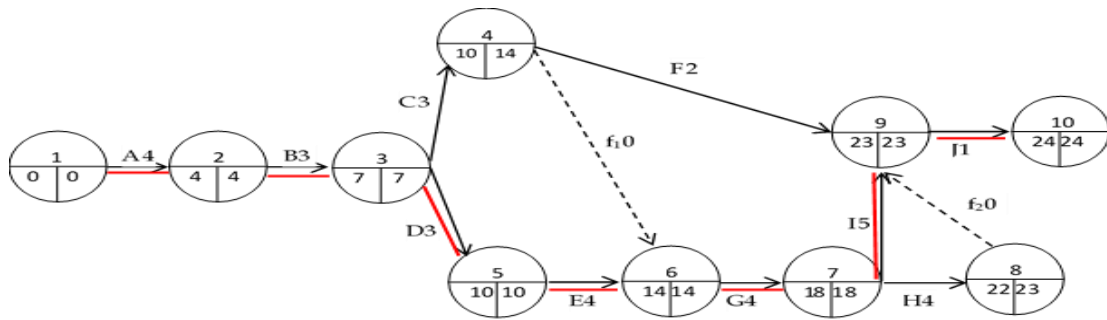
(14 pts)

- La régie : la maîtrise d'ouvrage  
Le prestataire « **M@RIMAF** » : la maîtrise d'œuvre (1 pt)
- Donner le tableau des niveaux. (1,5 pt)

Niveau	1	2	3	4	5	6	7	8
Tâches	A	B	C,D	E,F	G	H	I	J

- Élaborer le diagramme de PERT en indiquant les dates au plus tôt et au plus. (3 pts)

Le réseau de PERT



- Calculer les marges totales. (2 pts)

MT(A)=4-0-4=0	MT(F)=23-10-2=11
MT(B)=7-4-3=0	MT(G)=18-14-4=0
MT(C)=14-7-3=4	MT(H)=23-18-4=1
MT(D)=10-7-3=0	MT(I)=23-18-5=0
MT(E)=14-10-4=0	MT(J)=24-23-1=0

- Indiquer le chemin critique. (1 pt)

A-B-D-E-G-I-J

- Déduire le temps minimal de réalisation de ce projet. (1,5 pt)

Le temps minimal de la réalisation du projet est la somme des durées des tâches du chemin critique :  
Temps minimal=4+3+3+4+4+5+1=24 jours  
Elle correspond aussi à la date de fin de la dernière étape du diagramme de PERT

- Calculer la charge globale du projet en jour homme ? (1,5 pt)

Charge globale

$$\begin{aligned}
 &= \sum_{i=1}^n \text{durée}_i \times \text{capacité}_i \\
 &= 4 \times 3 + 3 \times 2 + 3 \times 3 + 3 \times 3 + 4 \times 2 + 2 \times 2 + 4 \times 3 + 4 \times 3 + 5 \times 2 + 1 \times 1 \\
 &= 12 + 6 + 9 + 9 + 8 + 4 + 12 + 12 + 10 + 1 = 83 \text{ jour homme}
 \end{aligned}$$

8. En respectant les informations du tableau ci-dessus et en débutant les tâches au plus tôt, compléter l'histogramme suivant : (1,5 pt)

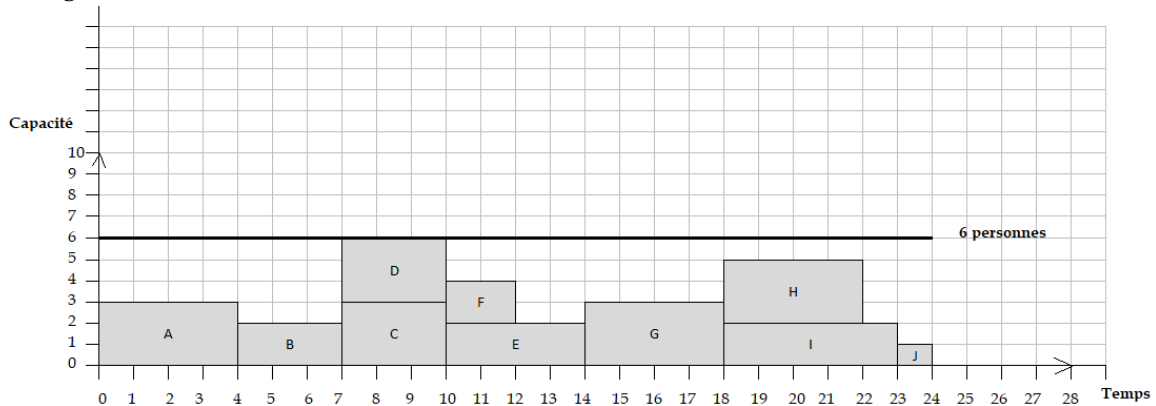
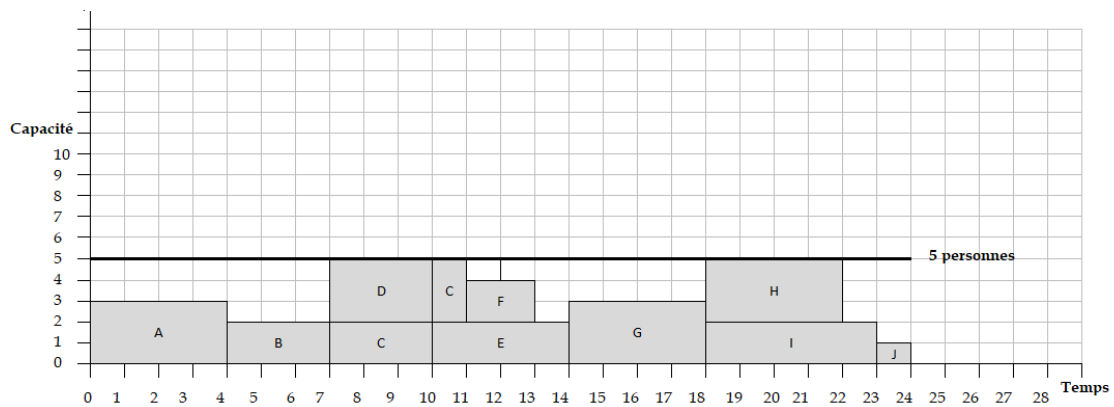


Figure 1 : histogramme des charges et capacité

9. Proposer une modification de l'ordonnancement initial satisfaisant la contrainte évoquée ci-dessus sans modifier la durée totale du projet. Donner le nom de cette technique. (1 pt)

On a :  $MT(C) = 4 \text{ jours}$  et  $MT(F) = 11 \text{ jours}$  alors : le déplacement de la tâche C, de 3 jours, et la tâche F d'une journée permet de satisfaire la contrainte « 5 personnes max par jour » sans modifier la durée totale du projet.

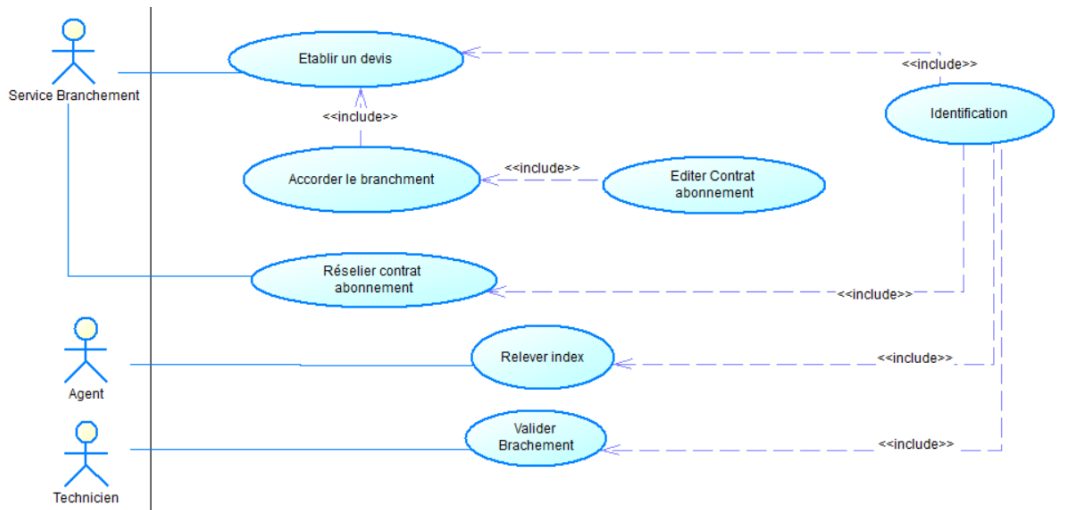
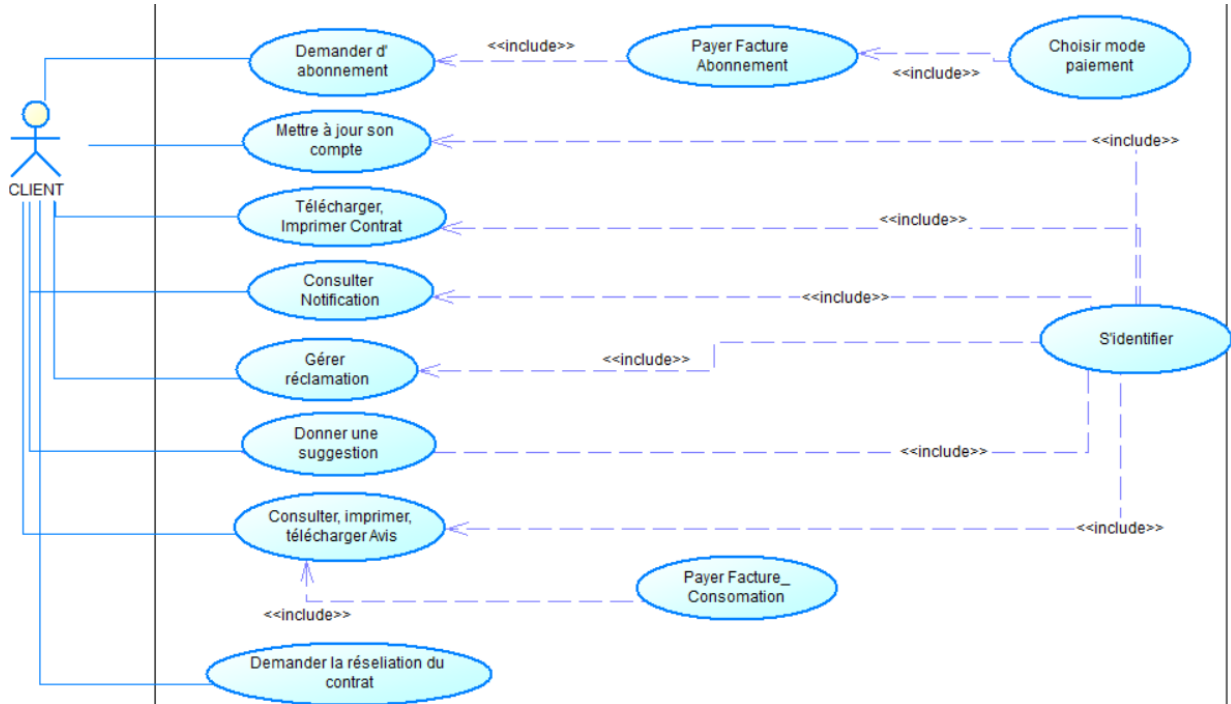
C'est la technique de lissage :  $\forall t \text{ on doit avoir } \text{capacité}(t) \geq \text{charge}(t)$



DOSSIER II : GESTION DE CONSOMMATION DE L'EAU

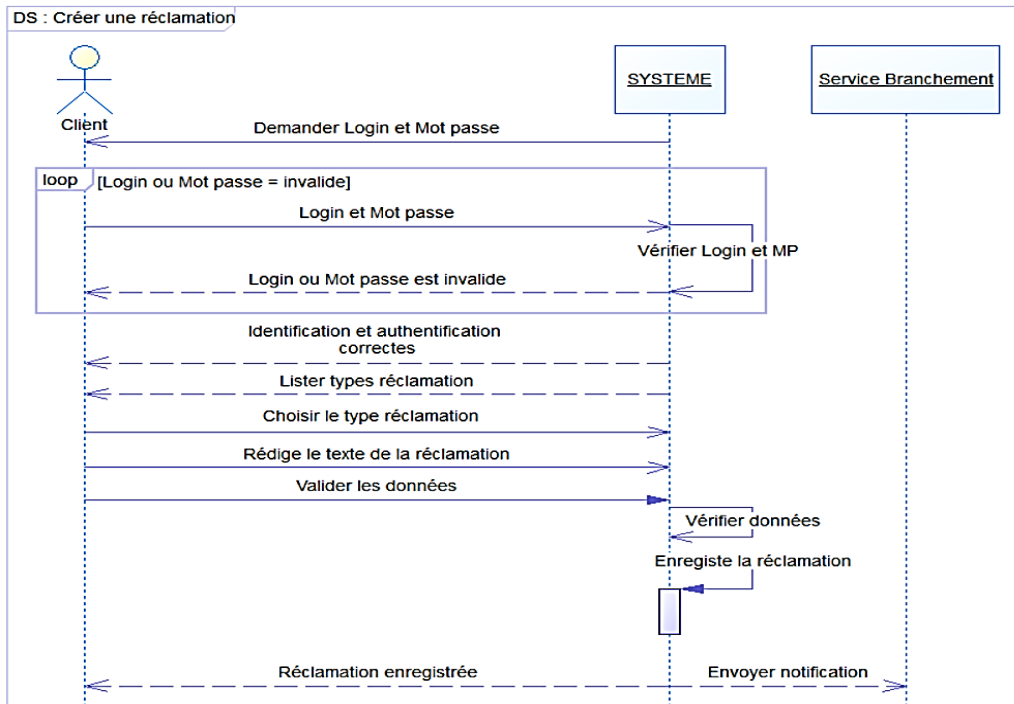
(12 pts)

1. Les acteurs qui agissent sur le système étudié. (1 pt)  
 Client (*abonné*), service branchement, technicien, agent
2. Élaborer le diagramme des cas d'utilisation correspondant. (4 pts)



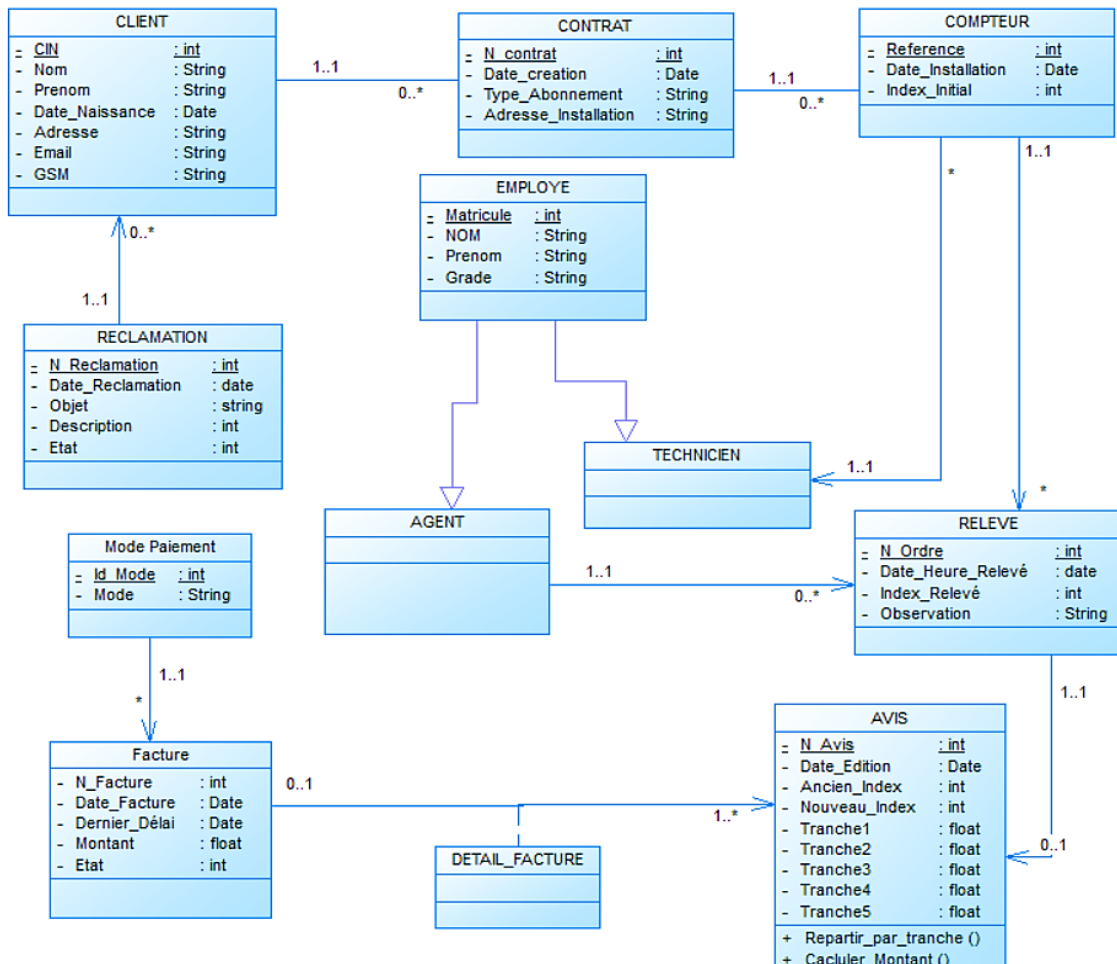
3. La spécification du scénario nominal de "créer une réclamation" est décrite par :  
 Décrire ce cas d'utilisation par un diagramme de séquence.

(2 pts)



4. Établir le diagramme de classes permettant de représenter les informations du système étudié.  
 Indiquer les attributs et les multiplicités.

(5 pts)



**DOSSIER III : GESTION DES ABONNEMENTS PROVISOIRES**

(14 pts)

1. Donner le code Transact SQL permettant la création de cette base de données « **Gs\_compteur** » en utilisant les propriétés suivantes : (2 pts)

	Nom Logique	Croissance	Taille Initiale	Taille maximale	Emplacement
Fichier de données	Compteur_data	5%	10 MB	1 GB	D:\Regie\Data\
Fichier journal	Compteur_log	10 MB	5 MB	Illimitée	D:\Regie\Log\

```
CREATE DATABASE Gs_compteur
ON PRIMARY
    (NAME = Compteur_data, FILENAME = 'D:\Regie\Data \ Compteur_data.mdf', SIZE = 5MB,
    MAXSIZE = 1 GB, FILEGROWTH = 5 %)
LOG ON
    ( NAME = Compteur_log, FILENAME = D:\DataBases\BDAudit.Idf', SIZE = 5MB,
    MAXSIZE = UNLIMITED, FILEGROWTH = 10MB )
```

2. Donner le code Transact SQL permettant la création des tables « **COMPTEUR** » et « **ABONNEMENT** » de cette base de données en respectant les contraintes suivantes : (3 pts)

```
CREATE TABLE ABONNEMENT (Reference varchar(20) PRIMARY KEY, Date_abonnement Date, Date_Fin Date,
Entreprise INT, Adresse_installation varchar(150),
Etat_Abonnement varchar(50) CHECK IN ('En cours', 'Clôturé', 'Annulé', 'Résilié'),
Constraint FK_Ent_Ab Foreign Key (Entreprise) References Entreprise (Id_Client));

CREATE TABLE COMPTEUR (Ref_Compteur int Primary Key, Ref_Abn varchar(20),
Date_installation Date, Marque varchar(150),
Etat_Compteur varchar(50) CHECK IN ('En service', 'Hors service'),
Constraint FK_Cmp_Ab Foreign Key (Ref_Abn) References ABONNEMENT(Reference));
```

3. Donner le code Transact SQL permettant la création d'une fonction qui retourne la raison sociale d'entreprise en la cherchant par sa référence d'abonnement. (2 pts)

```
Create function Fn-trouver_Abonné(@Ref as varchar(20)) returns varchar(50) AS
BEGIN
    DECLARE @RS varchar(50), @ID INT
    Select @ID=Entreprise From ABONNEMENT Where Reference=@Ref
    SELECT @RS=Raison_sociale From ENTREPRISE where Id_Client=@ID
    RETURN @RS
END
```

4. Donner le code de la procédure « **Sp\_Liste\_Index** » permettant d'afficher la liste des relevés (date et index) concernant un compteur donné. (1,5 pt)

```
Create proc Sp_Liste-Index ( @Ref As varchar(20) ) AS
BEGIN
    If Exists (Select * From RELEVE Where Compteur=@Ref)
        Select Date_Releve, Index_Releve From RELEVE Where Compteur =@Ref
    Else
        raiserror('ce Compteur n''existe pas',15, 1 )
END
```

5. Donner le code de la fonction table « **fn\_lst\_abonnements** » permettant de lister des abonnements avec indication du nom d'entreprise (*raison sociale*) et du nombre de compteurs installés entre deux dates données en argument. (1,5 pt)

```
Create function fn_lst_abonnements (@du date, @Au date)
Returns @T_Abonnement Table (@Nom varchar(50), @NbreCompteurs INT) As
Begin
    INSERT INTO @T_Abonnement (@Nom, @ NbreCompteurs)
    SELECT Reference, Raison_sociale, COUNT(*) FROM COMPTEUR C, ABONNEMENT A, ENTREPRISE E
    WHERE C.Ref_Abn = A.reference AND E.id_client = A.Entrepise
    AND date_installation >= @du AND date_installation <= @Au
    GROUP BY Reference, Raison_sociale
    RETURN
End
```

6. Donner le code de la vue « **V\_MaxCompteur** » permettant de présenter pour chaque compteur d'eau la référence d'abonnement, la référence du compteur et l'index maximum relevé. (1 pt)

```
CREATE VIEW V_MaxCompteur as
SELECT Ref_Abn, Ref_Compteur, MAX(index_Releve) AS Max_Index FROM RELEVE R, COMPTEUR C
WHERE R.Compteur=C.Ref_Abn
GROUP BY Ref_Abn, Ref_Compteur
```

7. Donner le code de la fonction « **Fn\_consommation\_total** » permettant d'afficher la consommation totale, de tous les compteurs confondus, d'un abonnement donné depuis sa souscription. (1,5 pt)

```
Create function Fn_consommation_total ( @Ref As varchar(20) ) returns AS
Begin
    SELECT SUM(Max_Index) FROM V_MaxCompteur WHERE Ref_Abn = @Ref
    RETURN
End
```

8. Donner le code permettant la création d'un trigger (Trg\_verifier) qui permet d'empêcher la suppression et la modification d'un abonnement clôturé. (1,5 pt)

```
Createtriggger Trg_verifier on ABONNEMENT For Delete Update As
BEGIN
    If Exists (Select * From ABONNEMENT Where Reference= deleted.Reference
    And Etat_Abonnement='clôturé')
        Rollback ;
END
```