Royaume du Maroc
Ministère de l'Education Nationale, du Préscolaire et des Sports
Centre National de l'Évaluation et des Examens

Examen National du Brevet de Technicien Supérieur
Session de Mai 2023
- Corrigé -

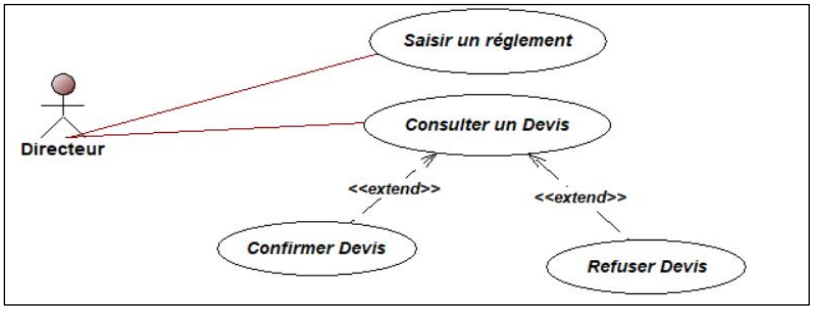
Filière :	Développement des Systèmes d'Information - DSI -	Durée :	4 heures
Épreuve :	Conception des Applications Informatiques	Coefficient :	50

ÉTUDE DE CAS : LA SOCIÉTÉ « MATBAKHI »

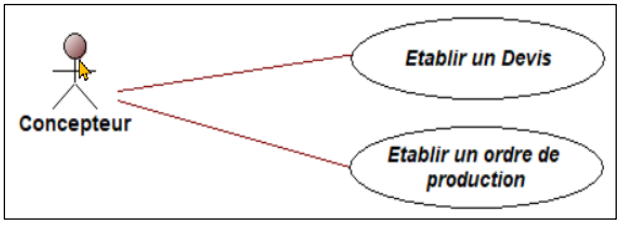
DOSSIER I : GESTION DE PRODUCTION *(14 points)*

❖ CRÉATION ET LANCEMENT D'UN PROJET

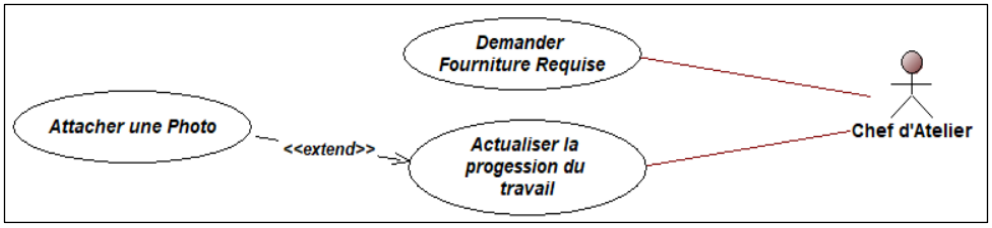
1. Donner le diagramme des cas d'utilisation pour les acteurs : *(3 pts)*
Directeur, Concepteur et Chef d'atelier.



1 pt

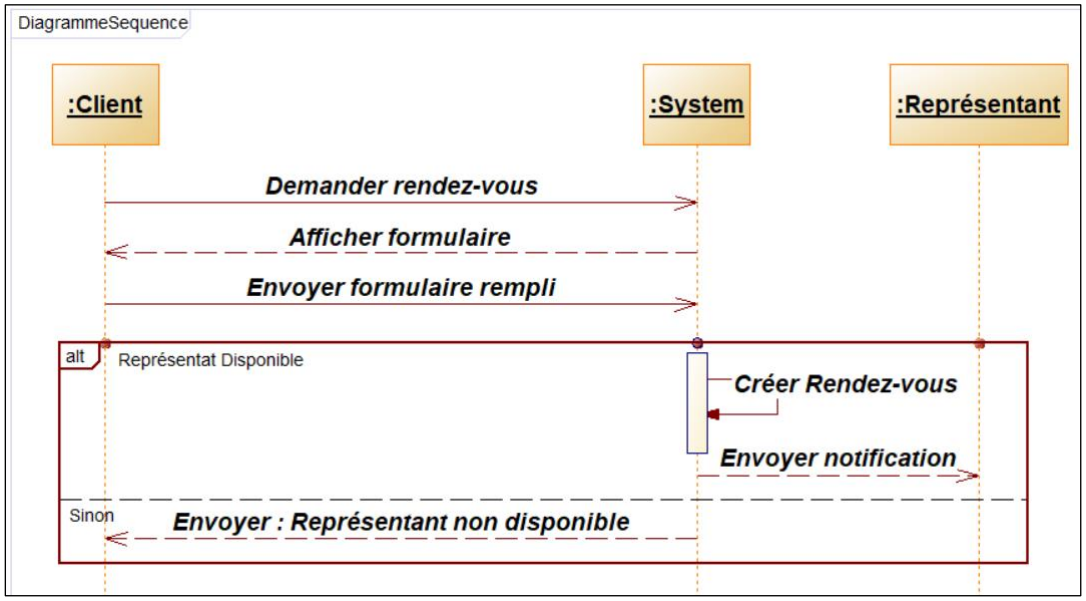


1 pt

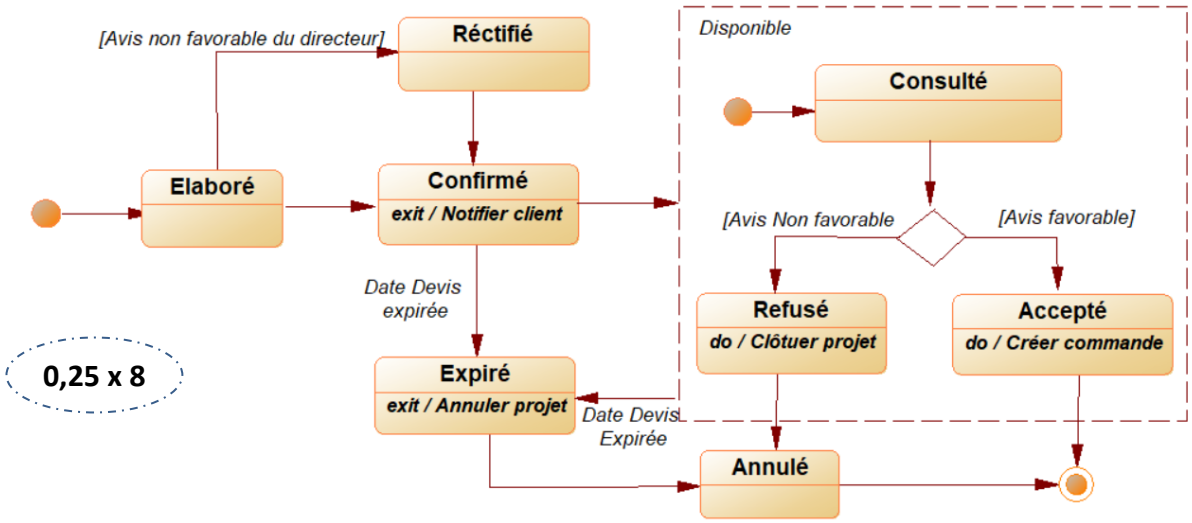


1 pt

2. Donner le diagramme de séquence correspondant au cas d'utilisation suivant : *(3 pts)*

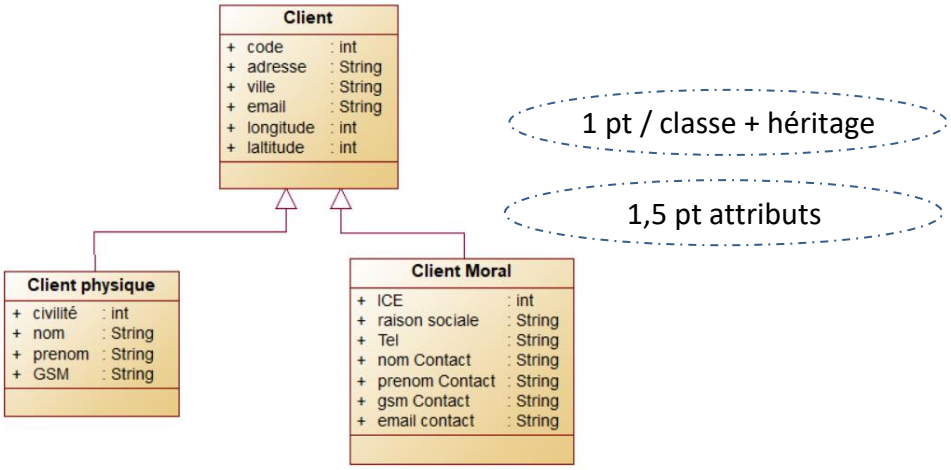


3. Reprendre et compléter le diagramme d'état-transition de l'objet « **devis** ». *(2 pts)*



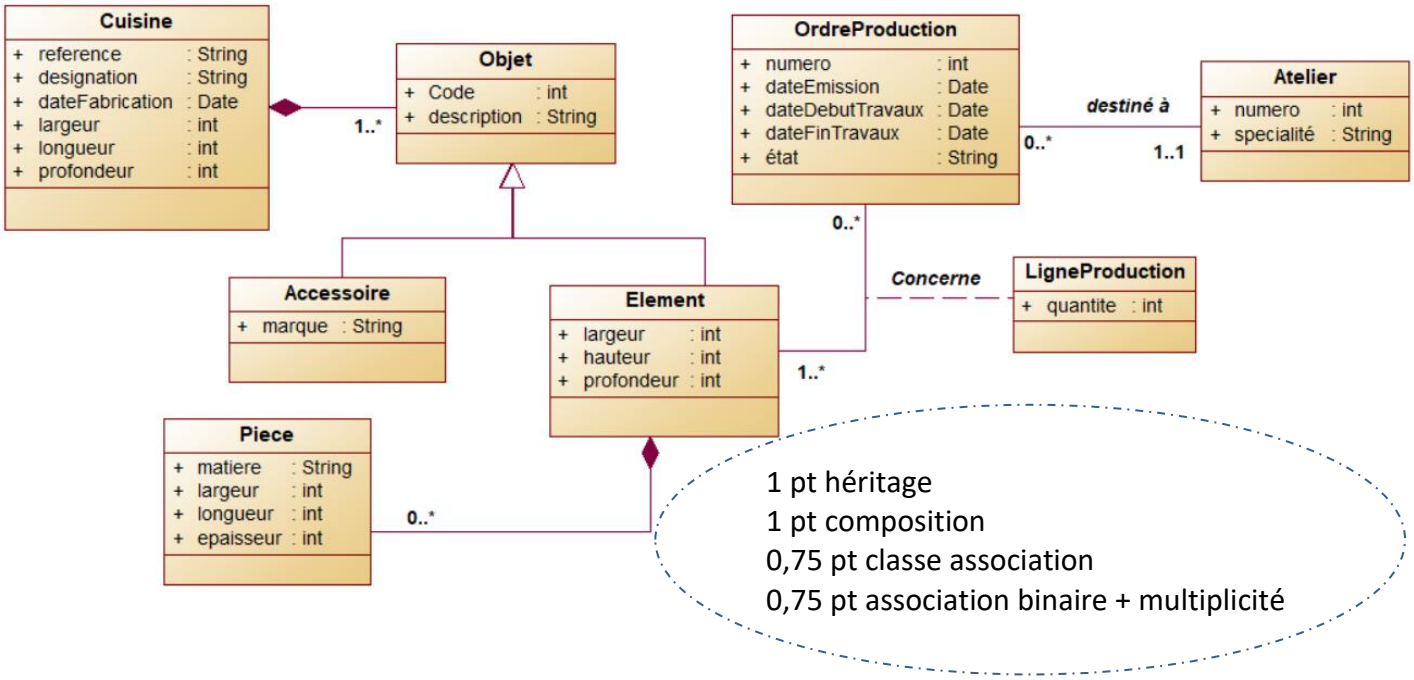
❖ GESTION DES CLIENTS

4.



❖ PRODUCTION DES CUISINES

5.



DOSSIER II : ÉTUDE DU PROJET *(12 points)*

❖ PLANIFICATION :

1) Que représente la société **Matbakhi** et l'entreprise **MiniSoft** dans ce projet.

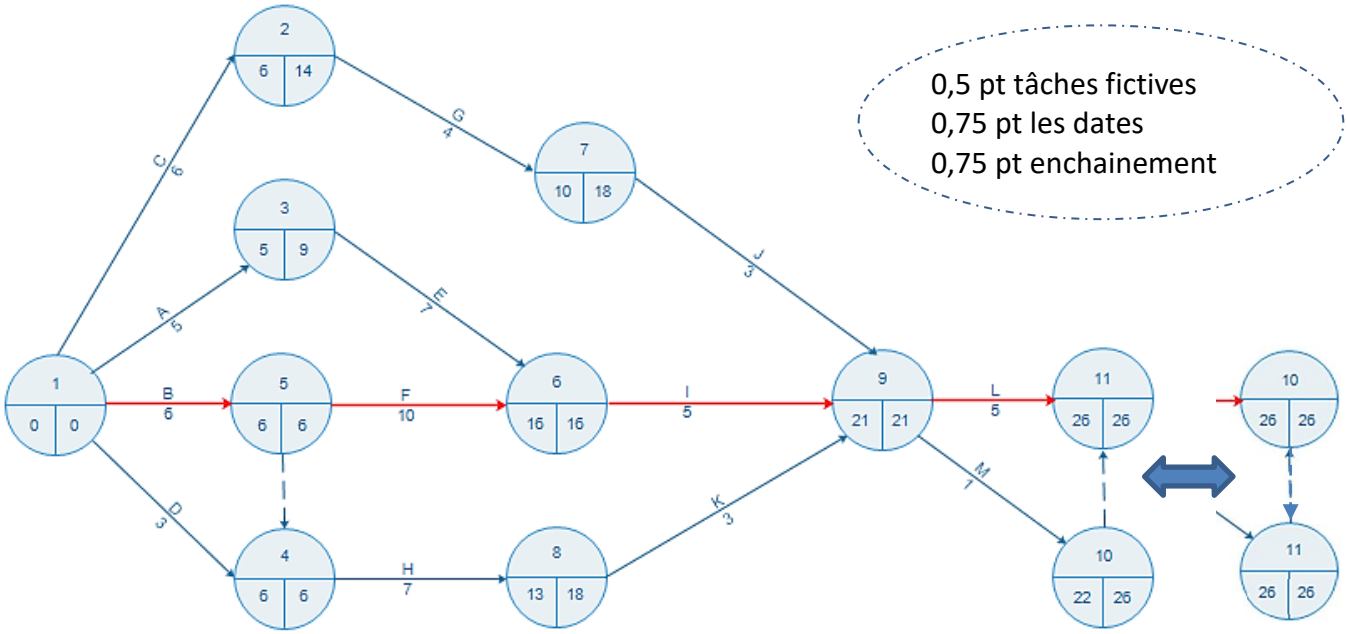
- Matbakhi : maitrise d'ouvrage *(0,5 pt)*
- MiniSoft : maitrise d'œuvre *(0,5 pt)*

❖ SUIVI :

2) *(1 pt)*

Niveau	Tâches
1	A, B, C, D
2	E, F, G, H
3	I, J, K
4	L, M

3)



4) La durée minimale du projet étant **26 semaines**. *(0,5 pt)*

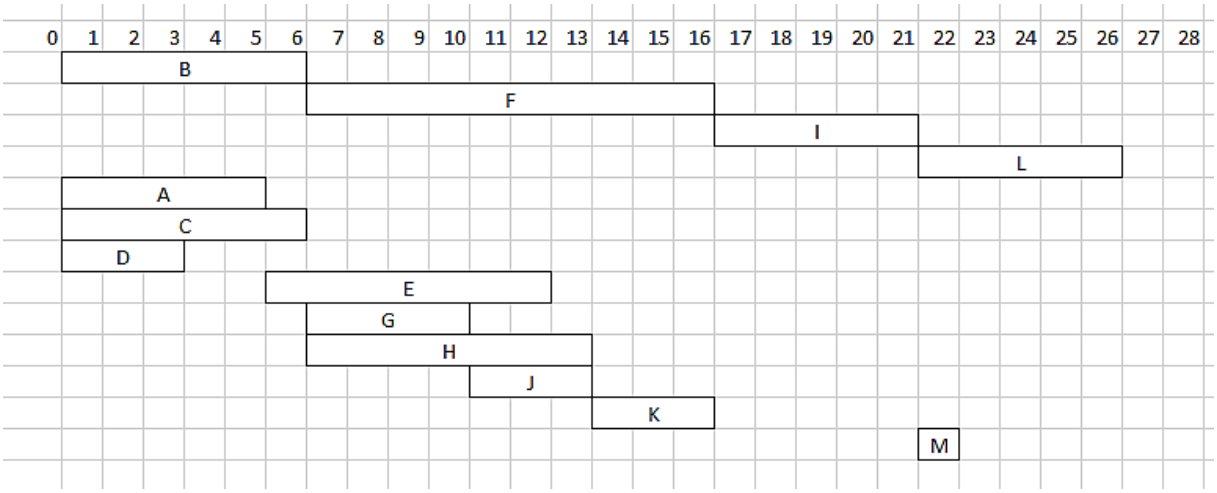
5) Les marges : *(0,25 pt x 6)*

	A	B	C	D	E	F	G	H	I	J	K	L	M
MT	4	0	8	3	4	0	8	5	0	8	5	0	4
ML	0	0	0	3	4	0	0	0	0	8	5	0	4

6) Le chemin critique : **B, F, I, L** *(1 pt)*

7) Le retard de **4 semaines** sur la tâche **D** entraine un retard d'une (***retard-marge* = *1***) semaine sur le délai du projet (***MT* = *3***) et la tâche **H** (***ML* = *3***). *(0,5 pt x 2)*

8) *(1 pt)*



9)

9.1.

a) Le coût réel : CR = 5000+8000+6000+4000+5000+3000 = **31 000 Dh.** *(0,5 pt)*

b) La valeur planifiée : selon le diagramme de Gantt, il fallait : *(0,5 pt)*

- terminer les tâches A, B, C et D.
- passer 2 semaines de travail pour les tâches F, G, et H.
- passer 3 semaines e travail pour la tâche E.

d'où :

VP = 5000+6000+9000+4500+(14000*3/7)+(30000*2/10)+ (6000*2/4)+(14000*2/7)

VP = 5000+6000+9000+4500+6000+6000+3000+4000

VP = **43 500 Dh**

c) La valeur acquise : VA = 100%A+100%B+100%C+100%D+40%E+50%G *(0,5 pt)*

= 5000+6000+9000+4500+5600+3000

VA = **33 100 Dh**

d) L'écart de cout EC = VA – CR = 33100-31000= **2100Dh** *(0,25 pt)*

e) L'écart de délai ED = VA – VP = **-10400 Dh** *(0,25 pt)*

9.2. On est en retard car ED <0. *(0,25 pt)*

9.3. On est en sous-consommation car EC >0. *(0,25 pt)*

DOSSIER III : GESTION COMMERCIALE *(14 points)*

1. Donner le code Transact SQL permettant la création de la base de données « **GS-Commerciale** » en considérant les paramètres suivants : *(2 pts)*

```
USE master ;
GO
CREATE DATABASE GS_Commerciale
ON    ( NAME = Commerce_data,
      FILENAME = 'E:\BD_Comm\Data\Commerce.mdf',  SIZE = 10MB,
      MAXSIZE = UNLIMITED,  FILEGROWTH = 3 )
LOG ON  ( NAME = Commerce_log,
      FILENAME = 'E:\BD_Comm\Log\Commerce.ldf',
      SIZE = 5MB,  MAXSIZE = 1GB,
      FILEGROWTH = 10MB ) ;
GO
```

2. Donner le code Transact SQL permettant la création de la table **« PROJET »** en respectant les contraintes suivantes : *(2,5 pts)*

- La date de clôture est postérieure à la date de visite.
- Le champ « *numProj* » est auto-incrémenté.

NB : On suppose que les autres tables sont déjà créées.

```
CREATE TABLE PROJET (
      numProjet int IDENTITY(1,1) Primary Key, Date_visite Date, Date_cloture Date,
      Description Text, Etat_Projet nvarchar(20), client int, Representant int,
      Constraint Chk_Date Check (Date_visite <= Date_cloture) ,
      Constraint PK_Pr_Clt Foreign KEY (client) REFERENCES Client(Code_client) ,
      Constraint FK_Pr_Rep Foreign KEY (Representant) REFERENCES Representant (Matricule) );
```

3. Donner le code Transact SQL permettant la création de la fonction **« fn_Montant_Devis() »** qui renvoie le montant total HT d'un devis qui vaut la somme des totaux des lignes du devis et du prix d'installation. Le total d'une ligne étant le produit de la quantité et du prix unitaire. *(1,5 pt)*

```
Create function fn_Montant_Devis (@devis  int)
returns money as
   begin
   declare @total, @Mt_Inst float
   select  @Mt_Inst = Prix_Installation from Devis where numDevis = @devis
   select  @total = SUM(quantite*prix), Devis from Ligne_Devis group by Devis
         having Devis = @devis
         if(@total is not null)
               set @total = @total+ @Mt_Inst
         return @total
    end;
```

4. Donner le code Transact SQL permettant la création de la fonction **« fn_Montant_Cmd() »** qui renvoie le montant total HT d'une commande qui est égal au montant du devis correspondant. *(1,5 pt)*

```
CREATE FUNCTION fn_Montant_Cmd (@cmd AS INT) RETURNS FLOAT AS
BEGIN
   DECLARE @p INT
   SET @p =  (SELECT Devis FROM commande WHERE numCmd = @cmd)
   RETURN dbo.fn_Montant_Devis(@p) ;
END
```

5. Donner le code Transact SQL pour créer la fonction **table « fn_Lignes_Cmd()»** permettant de lister les lignes d'une commande qui sont les mêmes lignes du devis correspondant. *(1,5 pt)*

```
CREATE FUNCTION fn_Lignes_Cmd (@num_commande AS INT) RETURNS TABLE
   As RETURN (
   SELECT * FROM ligne_Devis WHERE Devis= (SELECT Devis FROM Commande
                                           WHERE numCmd = @num_commande)
);
```

6. Donner le code Transact SQL pour créer la procédure **« sp_Creer_Cmd() »** permettant de générer une commande à partir d'un devis donné sachant que : *(2 pts)*

- La date de la commande est celle du système ;
- L'état par défaut, d'une commande, est « *en cours* » ;
- Le numéro de la commande est auto-incrémenté.

```
CREATE PROCEDURE sp_Creer_Cmd (@devis AS INT) AS
BEGIN
   INSERT INTO Commande (Date_Cmd, Etat_Cmd, Devis)
                 VALUES (GetDate(),'En cours', @devis)
END
```

7. Donner le code Transact SQL pour créer la vue **« v_Clients_Actifs »** permettant de lister les codes, noms, prénoms et villes des clients ayant au moins un projet en cours de réalisation. *(1,5 pt)*

```
CREATE VIEW v_Clients_Actifs AS
  SELECT code_client, nom, prenom, ville FROM Client
  WHERE code_client in(SELECT Client FROM Projet WHERE Etat_projet='En cours')
```

8. Donner le code Transact SQL pour créer le trigger **« tr_Projet_Clt »** permettant d'empêcher la suppression d'un projet dont l'état est « ***clôturé*** ». *(1,5 pt)*

```
CREATE TRIGGER tgr_Projet_clt ON Projet INSTEAD OF DELETE AS
 BEGIN
     IF exists(SELECT * FROM DELETED WHERE Etat_Projet = 'clôturé' )
         Delete from Projet where NumProjet in (select NumProjet from Deleted);
    Else
         Raiserror('impossible de supprimer un projet non clôturé ',12,1);
 End
```