

## ÉTUDE DE CAS : « ENFANT MALIN » pour accessoires d'enfants et jouets

### DOSSIER I : ÉTUDE DU PROJET DE LA SOCIÉTÉ

(12 pts)

#### 1. QUESTIONS PRÉLIMINAIRES

1 - 1. Définir la maîtrise d'ouvrage dans un projet informatique ?

(1 pt)

La maîtrise d'ouvrage (MOA), aussi dénommée maître d'ouvrage est l'entité porteuse d'un besoin, définissant l'objectif d'un projet, son calendrier et le budget consacré à ce projet. Le résultat attendu du projet est la réalisation d'un produit, appelé ouvrage.

La maîtrise d'ouvrage est à l'origine de l'idée de base du projet et représente, à ce titre, les utilisateurs finaux à qui l'ouvrage est destiné.

1 - 2. Quelle est la différence entre le test unitaire et le test d'intégration ?

(1 pt)

Dans le test unitaire, on vérifie le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module ») ; dans le test d'intégration, chacun des modules indépendants du logiciel est assemblé et testé dans l'ensemble.

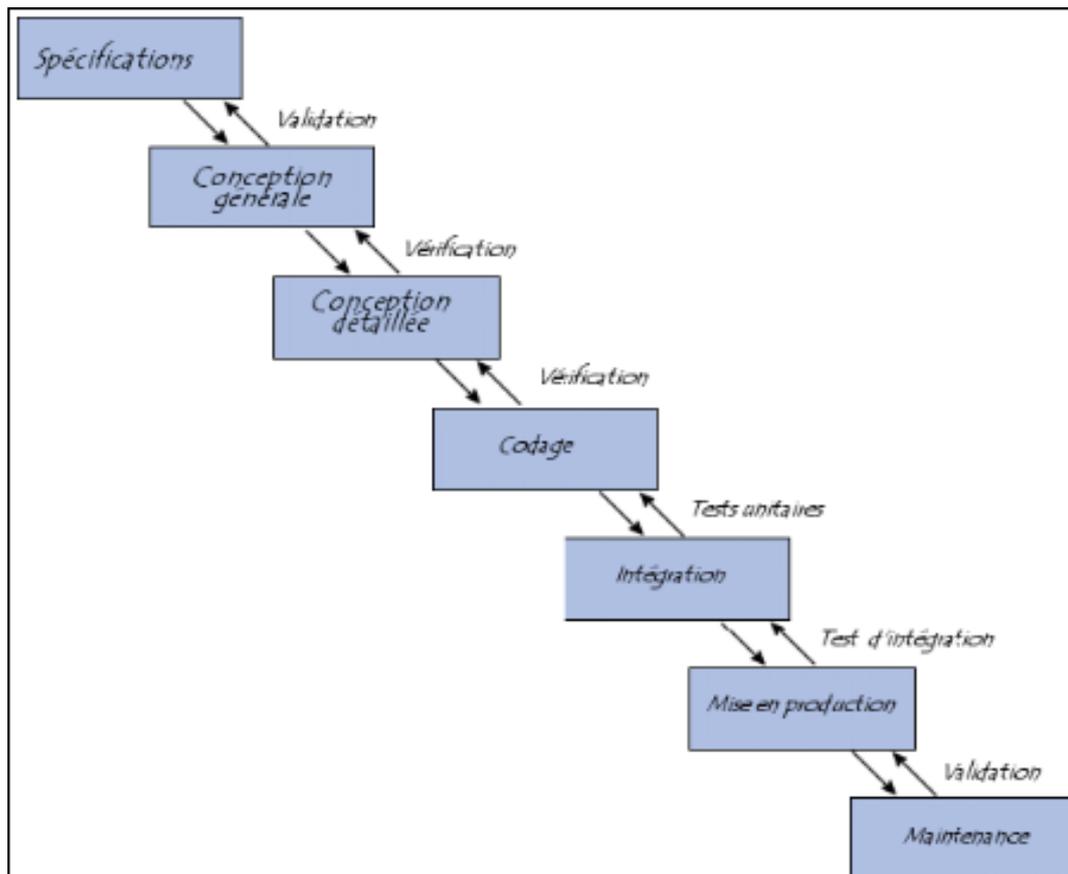
1 - 3. En quelle(s) étape(s), se trouvent les tests (unitaire et d'intégration) dans le cycle de vie en cascade ?

Test unitaire dans l'étape de programmation

(0,5 pt)

Test d'intégration dans l'étape intégration.

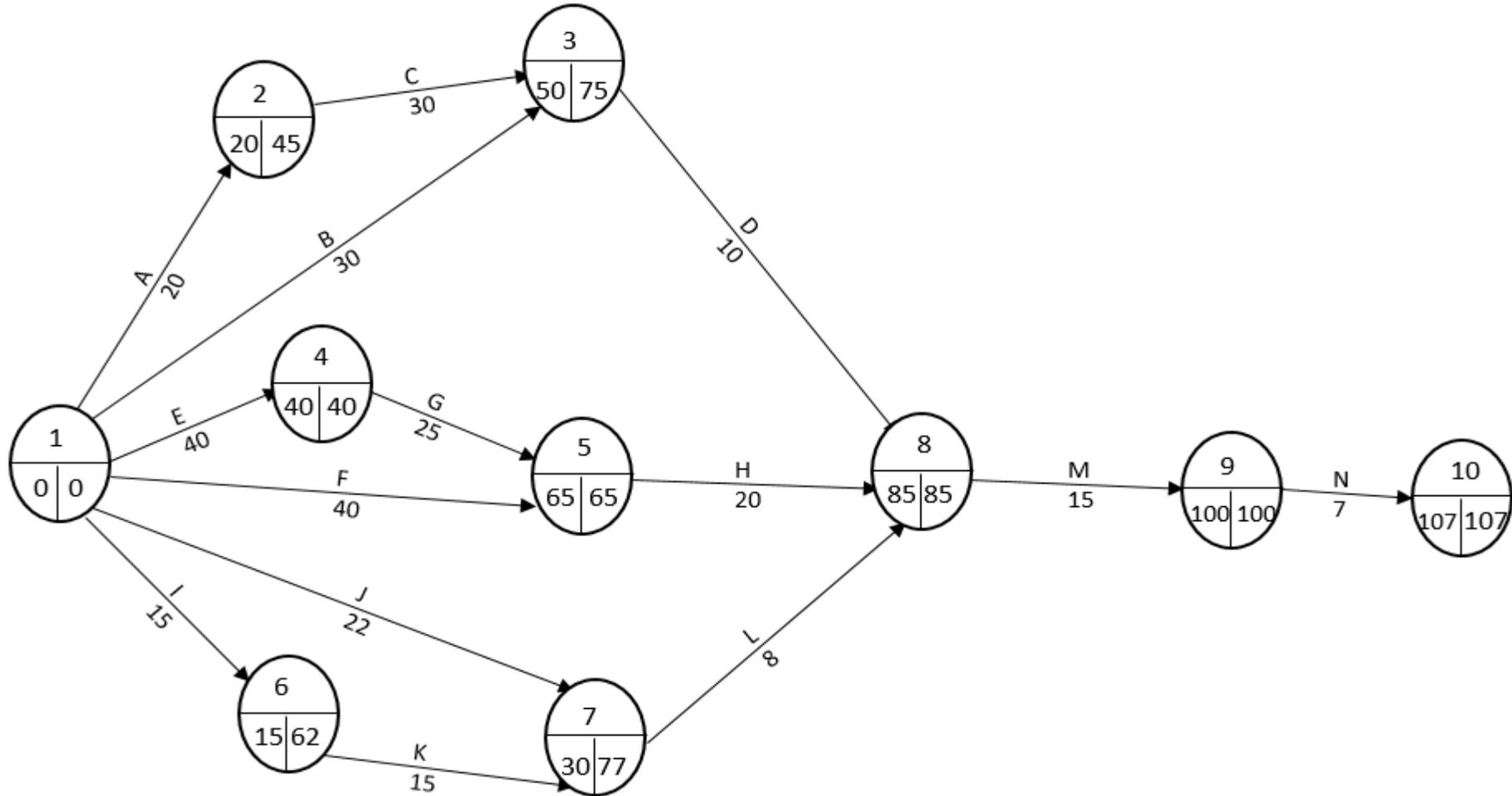
(0,5 pt)

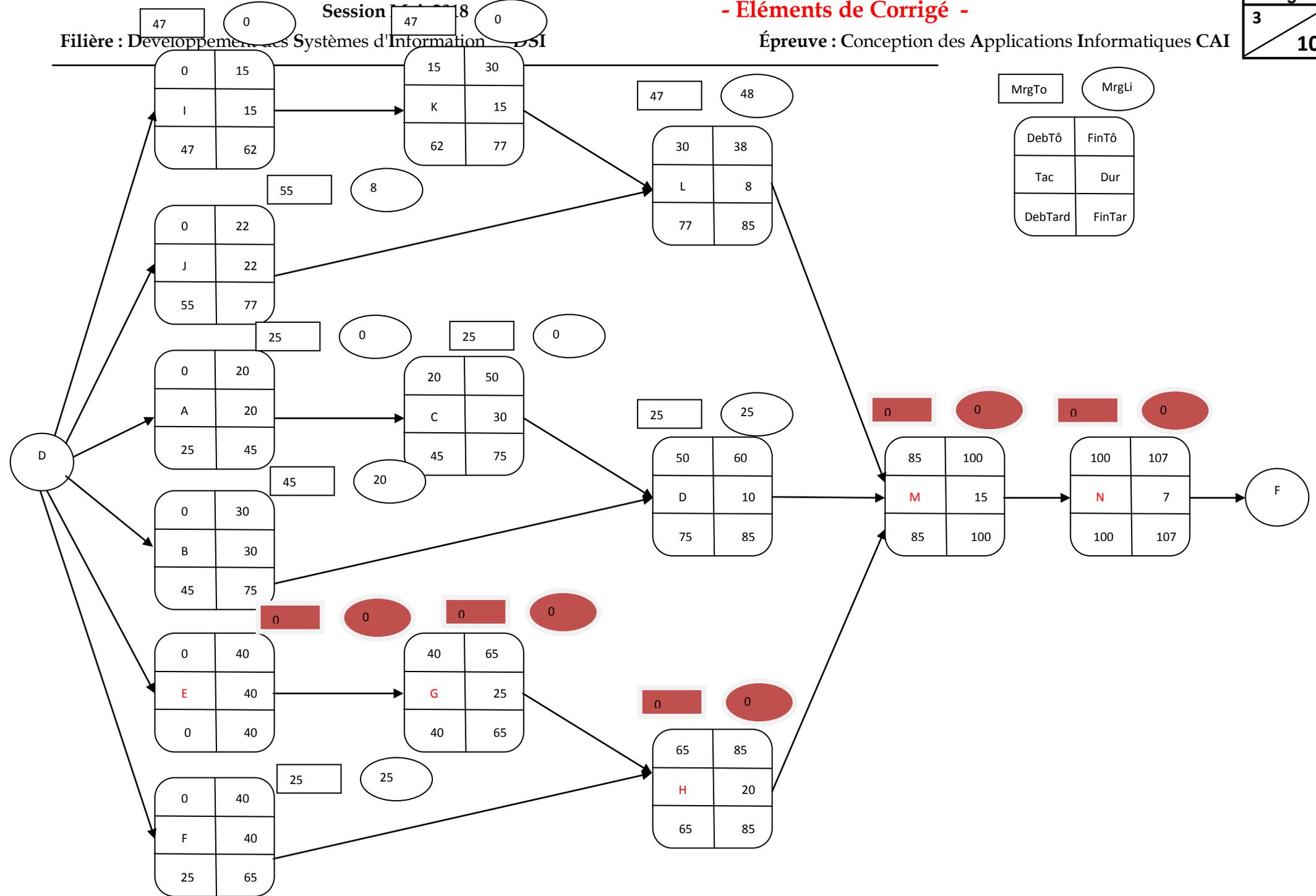


2. PLANIFICATION DU PROJET

2 - 1. Diagramme de PERT.

(2 pts)





2 - 2. Déduire la durée minimale du projet. (0,5 pt)

La durée minimale du projet est de 107 jours

2 - 3. Représenter pour les tâches (D et K), sur le tableau suivant, les dates au plutôt et au plus tard ainsi que les marges totale et libre. (1 pt)

Tâche	Au plutôt		Au plus tard		Marge totale	Marge Libre
	Début	Fin	Début	Fin		
D	50	60	75	85	25	25
K	15	30	62	77	47	0

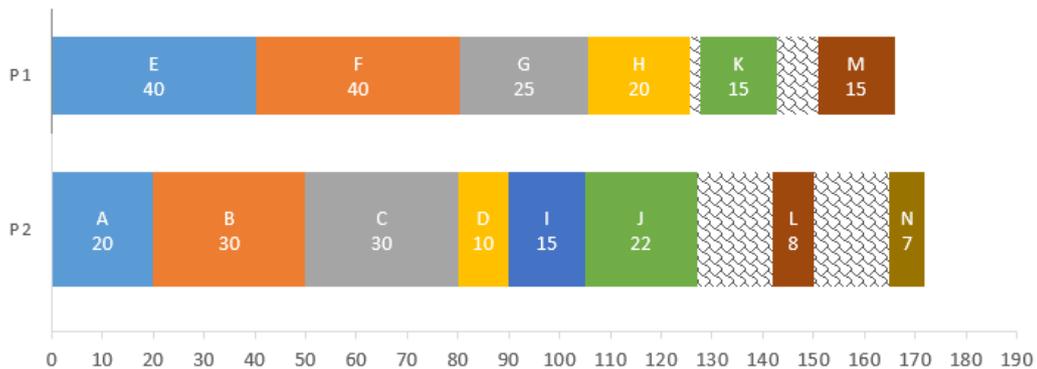
2 - 4. Le chemin critique. (0,5 pt)

**E, G, H, M, N**

2 - 5. Le diagramme de GANTT. (1,5 pt)

Les tâches de chaque lot doivent se suivre dans l'ordre suivant : conception, jeu de test, programmation et test.

NB. Il peut exister d'autres variantes de solutions : l'essentiel est de respecter les contraintes d'enchaînement des tâches dans le diagramme de PERT.



P1	0	40	80	105	125	127	142	150	165
	E		F		G	H	K	M	

P2	0	20	50	80	90	105	127	142	150	165	172
	A	B		C		D	I	J	L	N	

2 - 6. La durée effective du projet est de 172 jours (0,5 pt)

### 3. ESTIMATION DES CHARGES

3 - 1. Calculer la charge (*en Mois Homme*) de la programmation de chacun des 3 lots (*Lot1, Lot2 et Lot3*) en utilisant la méthode **COCOMO** : (1 pt)

Tous les lots correspondent à des projets simple ( $kisl < 50\,000$ )

Lot	Kisl	Charge estimée (programmation)
Lot 1	1.00	3.20 mois homme ( MH)
Lot 2	1.25	4.04 mois homme ( MH)
Lot 3	0.50	1.55 mois homme ( MH)

3 - 2. Calculer la charge des autres étapes de chacun des 3 Lots (*Lot1, Lot2 et Lot3*) en utilisant le tableau des ratios suivant : (1 pt)

Lot	Conception ( MH)	Jeu de test ( MH)	Programmation ( MH)	Test ( MH)	Charge lot ( MH)
Lot 1	0.53	0.96	3.20	0.64	5.33
Lot 2	0.67	1.21	4.04	0.81	6.73
Lot 3	0.26	0.47	1.55	0.31	2.58

3 - 3. Calculer la charge de l'intégration et de la validation de tout le projet (*Lot4 et Lot5*), sachant qu'elle constitue **25%** des charges de la programmation. (0,5 pt)

La charge totale de programmation étant de  $3.20 + 4.04 + 1.55 = 8.79$  ( MH)

D'où la charge d'intégration et de validation est de  $8.79 * 0.25 = 2.20$  ( MH)

3 - 4. Calculer la charge totale du projet. (0,5 pt)

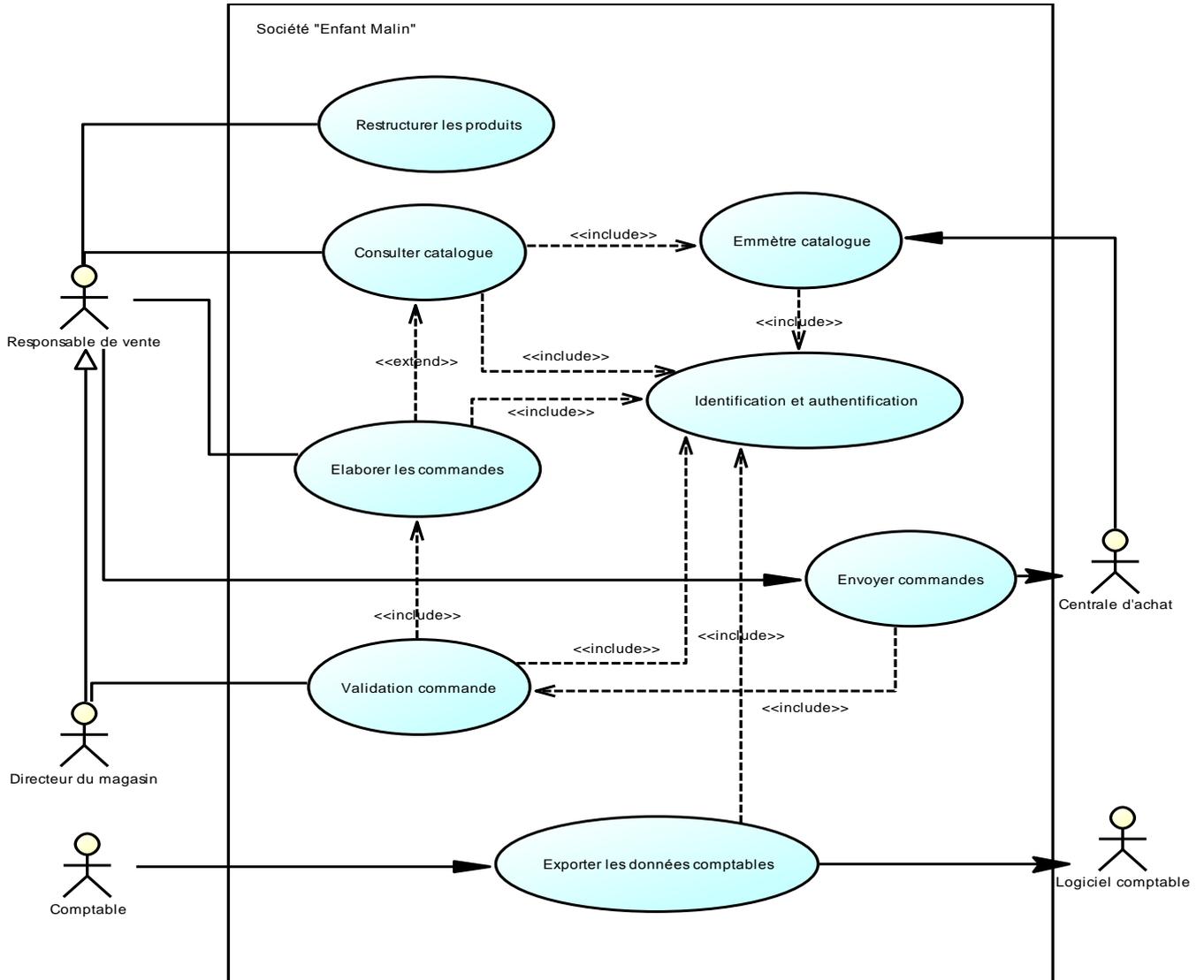
La charge totale du projet est de  $5.30 + 6.74 + 2.58 + 2.20 = 16.84$  mois homme ( MH)

DOSSIER II : L'APPROVISIONNEMENT DES MAGASINS & STOCK

(14 pts)

1. Diagramme de cas d'utilisation :

(4 pts)



1 pt : acteurs

0,5 pt : héritage d'acteur

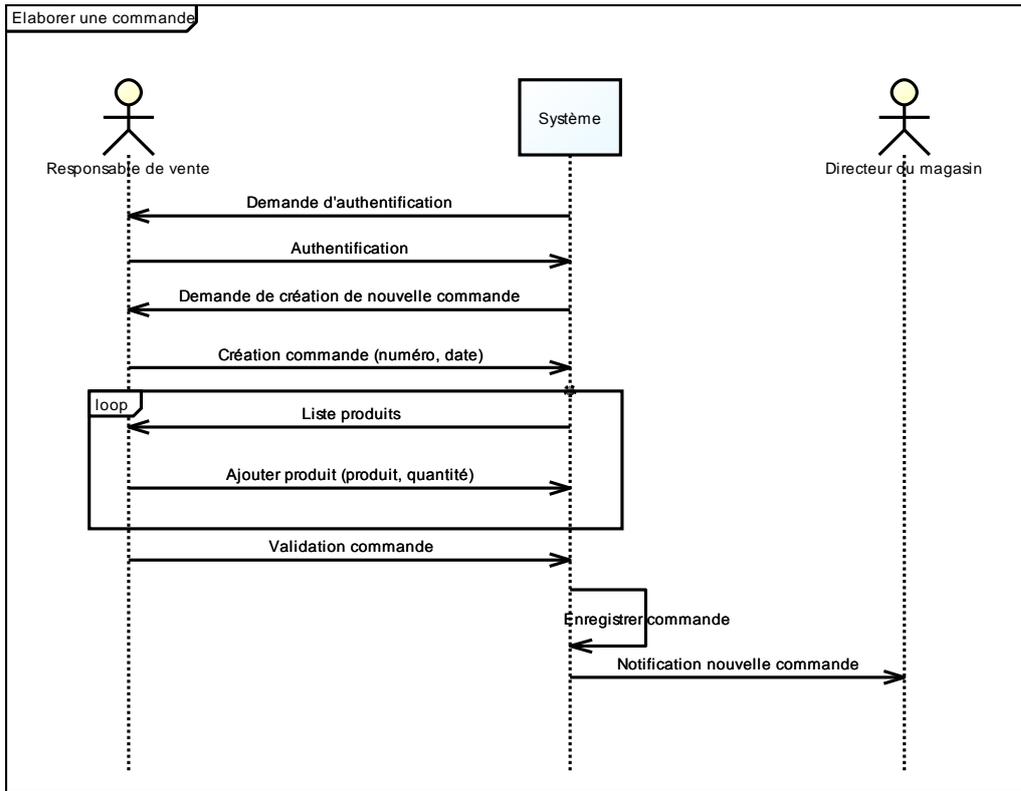
0,5 pt : identification et authentification

0,5 pt : extension

1,5 pt : autres

2. Diagramme de séquence.

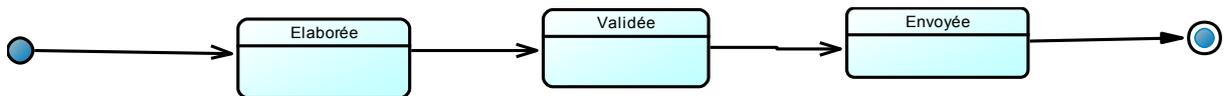
(3 pts)



1 pt : acteurs  
1 pt : loop  
1 pt : autres

3. Diagramme d'états de transition de l'objet **commande**.

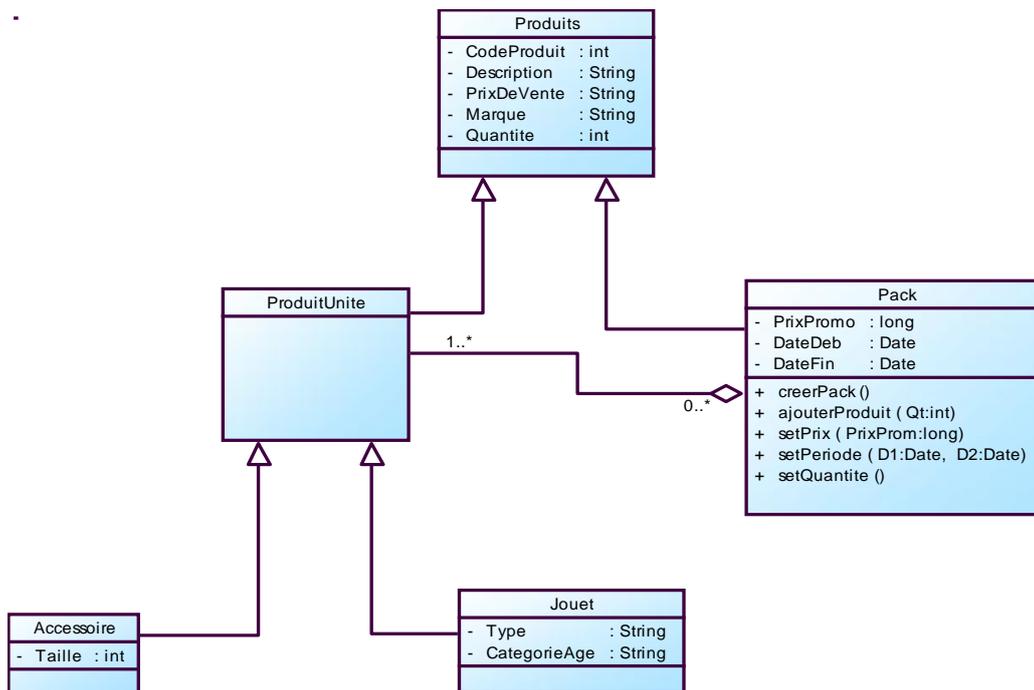
(2 pts)



1 pt : états début et fin  
1 pt : autres

4. Diagramme de classes.

(5 pts)



2 pts : classes  
 1 pt : héritage  
 1 pt : Composition  
 1 pt : autres

**DOSSIER III : GESTION DES VENTES**

(14 pts)

1.

```

CREATE DATABASE BD_Ordre
ON
( NAME = Ordre_livraison,
  FILENAME = E:\Srv_Bd\Ordre_livraison.mdf')
LOG ON
( NAME = Ordre_livraison_log,
  FILENAME = E:\Srv_Bd\Ordre_livraison_log.ldf') ;
GO ;
    
```

0,5 pt

0,5 pt

0,5 pt

2.

```

USE BD_Ordre;
GO
CREATE TABLE CLIENT(id_client INT IDENTITY(10,10) PRIMARYKEY,
  nom NVARCHAR(50) not null, prenom NVARCHAR(50)not null ,
  adresse NVARCHAR(250),telephone NVARCHAR(50), email NVARCHAR(80));
CREATE TABLE ORDRE_LIVRAISON (num_ordre INT PRIMARY KEY,id_client INT,
  Date_ordre DATE,Date_livraison DATE, Prix MONEY,
  Etat_livraison NVARCHAR(20)CHECK IN ('En cours','Livré','Annulé'),
  CONSTRAINT Fk_Client FOREIGNKEY (id_client)
  REFERENCES CLIENT(id_client) );
    
```

1 pt

1 pt

0,5 pt

0,5 pt

3. `CREATE PROC Sp_liste_ordres (@id_clt INT)AS` 0,5 pt  
`BEGIN`  
`IF exists (SELECT * FROM CLIENT WHERE id_client= @id_clt)` 1 pt  
`SELECT num_ordre,id_client,date_ordre`  
`FROM ordre_livraison WHERE id_client= @id_clt`  
`ELSE`  
`RAISERROR('ce client n'existe pas',15, 1 )` 0,5 pt  
`END;`

4. `CREATE FUNCTION fn_nb_Livraison(@id_clt INT)RETURNS INT` 0,5 pt  
`AS`  
`BEGIN`  
`DECLARE @nb INT= 0`  
`IF exists(SELECT * FROM client WHERE id_client= @id_clt)` 0,5 pt  
`BEGIN`  
`SET @nb=(SELECT COUNT(*) FROM ordre_livraison` 0,5 pt  
`WHERE Id_client= @id_clt)`  
`END`  
`RETURN @nb` 0,5 pt  
`END;`

5. a) Création d'un Trigger qui doit se déclencher avant l'insertion et la mise à jour d'une saisie dans la table « **ORDRE\_LIVRAISON** » afin de vérifier que la date de livraison est antérieure (*supérieure*) à la date d'ordre ou création une contrainte check sur le champ "**date\_livraison**". 0,5 pt
- b) Donner le code de la solution proposée. (1 pt)

```
CREATE TRIGGER Verifier_Date ON ORDRE_LIVRAISON
FOR insert,update AS
BEGIN
    DECLARE @date_Livraison date, @date_ordre DATE
    Select @date_Livraison=date_Livraison From inserted
    select @date_ordre=date_ordre From ORDRE_LIVRAISONO,inserted I
    Where O.num_ordre=inserted.num_ordre
    if (DATEDIFF(day,@date_ordre,@date_Livraison)<0)
    Begin
    Raiserror('La date de livraison ne peut pas être avant la date d'ordre',15,120)
    Rollback
    End
END
```

6.

```
CREATE FUNCTION Fn_Resultat(@annee int) RETURNS TABLE AS          0,5 pt
RETURN
    SELECT Id_Client ,COUNT(num_ordre) AS [Nombreordre],
    SUM(prix) AS [Montant total]          0,5 pt
    FROM ORDRE_LIVRAISON O, CLIENT C
    WHERE YEAR(date_ordre)= @annee          0,5 pt
    And C.Id_Client=O.Id_Client Group By Id_Client ;          0,5 pt
```

7.

```
CREATE TRIGGER Tr_Verifier on ORDRE_LIVRAISON          0,5 pt
FOR delete AS
BEGIN
    DECLARE @ID_Clt INT
    Select @ID_Clt =Id_Client From deleted          1 pt
    IF exists(SELECT * FROM Ordre_Livraison WHERE id_client= @ID_Clt)
    BEGIN
        RollBack          0,5 pt
    END
END
END
```