

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

**INF33307– Assurance de la qualité et gestion de projets informatiques**

**Automne 2023**

**Travail pratique #2**

**Professeur: Ismaïl Khriss**

# 1 Conditions de réalisation

Le travail doit s’effectuer en groupe de deux au maximum. **Travail (Document en format Word) à remettre via le site du cours au plus tard le 13 novembre à 23h59**.

# 2 Travaux à réaliser

En se basant sur l’énoncé du TP1, sa solution et le modèle de l’architecture du système (voir section 4), répondez aux questions suivantes (**expliquez vos résultats**):

## 2.1 Estimation basée sur les problèmes

Calculez l’effort total et la durée en utilisant COCOMO.

## 2.2 Estimation basée sur les processus

Le cadre de développement est le modèle en cascade. Calculez la durée des activités ci-dessous en utilisant les informations et formules suivantes:

* Communication : l’effort estimé est de 2 semaines.
* Planification : l’effort estimé est de 1 semaine.
* Analyse: l’effort estimé est de 2 semaines.
* Conception générale: l’effort estimé est de 1 semaine.
* Conception détaillée: l’effort de conception détaillée d’une classe est proportionnel à la longueur estimée du code de cette classe. Productivité de conception détaillée: 5000 lignes/mois.
* Codage et tests unitaires: productivité: 1000 lignes/mois.
* Tests d’intégration: l’effort est proportionnel au nombre de classes du système, le coefficient de proportionnalité est 0.12 mois, soit si le système comporte n classes, l’effort est de 0.12 \* n mois.

## 2.3 Planification : conditions 1

Construisez un réseau CPM sur lequel sont indiqués le chemin critique et sa durée ainsi que le dia­gramme de Gantt des activités et des ressources humaines, sous les conditions suivantes:

* Communication terminée avant le début de la planification.
* Planification terminée avant le début de l’analyse.
* Analyse terminée avant le début de la conception générale.
* Conception générale terminée avant le début de la conception détaillée.
* Pour chaque sous-système (il s’agit des sous-systèmes déduits de l’architecture du système (voir section 4) soit *SS\_Interface*, *SS\_Logique* et *SS\_AccèsDonnées*), on peut commencer les activités de codage et tests unitaires dès que la con­ception détaillée de toutes les classes du sous-système est terminée,
* En revanche, il faut attendre que toutes les activités de codage et tests unitaires de tous les sous-systèmes soient terminées pour commencer les tests d’intégration.
* Les ressources humaines suivantes sont disponibles: communication (1 personne), planification (1 personne), analyse (2 personnes), conception générale (1 personne), conception détaillée (3 personnes, une par sous-système), codage et tests unitaires (3 personnes, une par sous-système), tests d’intégration (1 personne).

## 2.4 Exploitation des jeux : conditions 2

Pouvez-vous raccourcir le chemin critique en modifiant l’affectation des ressources aux activités de conception détaillée et de codage et tests unitaires. Les autres conditions de conditions 1 restant en vigueur? Si oui, montrez le nouveau diagramme de Gantt des activités et des ressources humaines.

# 3 Barème

Question 2.1 10 points

Question 2.2 25 points

Question 2.3 40 points

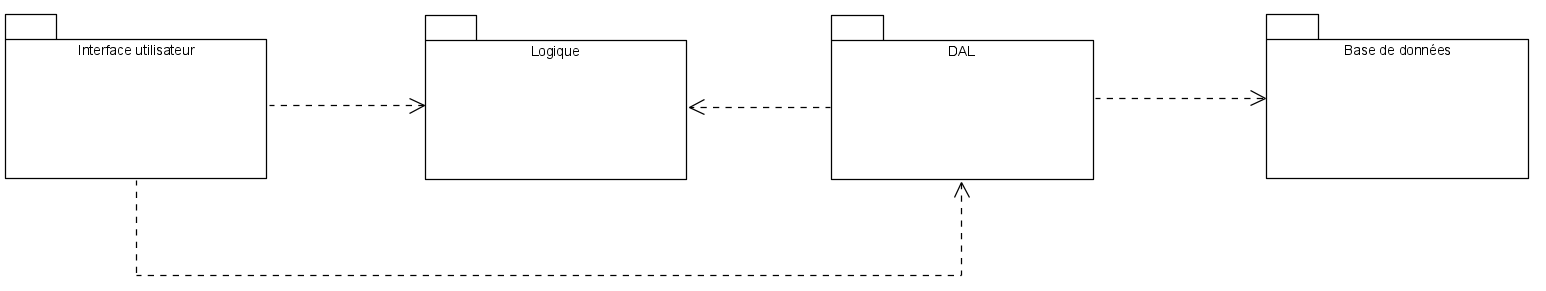
Question 2.4 15 points

Qualité de la présentation et du français 10 points

# 4 Architecture du système

Le système de gestion des accès aux bâtiments aura une architecture en couches (voir le diagramme ci-dessous) et est composé en trois sous-systèmes :

* Le premier sous-système *SS\_Interface* contient la couche *Interface utilisateur.*
* Le deuxième sous-système *SS\_Logique* contient la couche *Logique.*
* Le troisième sous-système *SS\_AccèsDonnées* contient les couches *DAL* et *Base de données*.



La couche logique contient toutes les classes du modèle d’analyse. La couche d’interface utilisateur (paquetage *Interface*) contient tous les formulaires du système. La couche DAL contient les classes qui font le lien entre les classes du système et la base de données servant à la persistance des données du système. Nous supposons que chaque classe de la couche logique va donner lieu à une interface et à une classe. Par exemple, si la couche logique contient la classe *Hôpital*, la classe DAL va contenir une interface qu’on peut appeler *IHôpital* et une classe qu’on peut appeler *HôpitalMapper*. La couche Base de données contient une seule classe qui joue le rôle d’une façade interagissant avec la base de données.