***Consigne du projet***

**Étape 1 : Spécification**

1. Rédiger les fonctionnalités de votre logiciel ( Description textuelle bien détaillé qui intègre tous les règles métier : voir exemple en cours)
2. Réaliser une maquette du projet qui décrit tous les écrans de l’application et leurs enchainement (Donnez tous les détails : voir exemple en cours)
3. Réaliser un dictionnaire donnée à partir des maquettes et du cahier de charge
4. Réaliser un diagramme MCD

**2 Conception : Un exemple sera réalisé en classe**

Nous proposons de réaliser une application en trois couches :

Couche persistance : cette couche est responsable de la communication avec la base des donnés.

Couche Métier : cette couche est responsable de la logique métier et communique avec la couche persistance pour lire et effectuer les mises à jour dans la base des données.

Couche Présentation : cette couche contient tous les interfaces graphiques et la logique de navigation. Cette couche implémente MVC et communique avec le couche métier.

1. Donnez les classes en détailles ( attributs et méthodes) de la couche persistance en respectant les règles suivantes :
   1. Chaque table dans la base de données est représenté par une classe DAO
   2. Chaque table DAO offre les services CRUD au minimum de la table qu’elle représente
   3. Ajouter une classe qui est responsable de la connexion avec la BD avec le pattern singleton
2. Donnez les classes en détailles ( attributs et méthodes) de la couche métier en respectant les règles suivantes :
   1. Cette couche est divisée en deux packages : POJO et Gestion
   2. POJO : classe Métier = Table BD avec getteurs et setteurs
   3. Service : classe qui gère les opération CRUD et la logique métier des POJO ( Collections du POJO + Méthodes CRUD)
   4. Les classe Service sont les seuls qui communiquent avec la couche persistance.
3. Donnez les classes en détailles ( attributs et méthodes) de la couche présentation en suivant les règles suivants :
   1. Chaque écran dans la maquette doit être présenté avec une classe
   2. Ajouter un classe modèle pour chaque écran ( cette classe contient les données de l’écran avec getteurs et setteurs)
   3. Ajouter un classe contrôleur pour gérer le CRUD de l’ensemble des écrans d’une table.
   4. Respecter le pattern MVC pour les classes précédents
   5. Seuls les classes contrôleurs ont le droit de communiquer avec la couche métier, un contrôleur peut appeler un autre contrôleur.

**3-Codage et Tests**

Créer un fichier Excel qui décrit tous les tests réalisés et leurs résultats. Nous suivrons la stratégie et l’ordre suivant:

1. Couche Persitance : Tester chaque classe et chaque requête
2. Tester les classes POJO ( constreucteurs, get, set ) avec une méthode afficher
3. Tester les méthodes des classes Service sans appel à la bases des données avec une méthode afficher
4. Tester les méthodes des classes Service avec appel à la bases des données avec une méthode afficher et en consultant la base de données
5. Tester les écrans sans faire appel aux contrôleurs et sans navigation entre les écrans
6. Tester la navigation entre écrans pour chaque cas d’utilisation
7. Tester les contrôleurs en faisant appel aux classes service
8. Tester chaque cas d’utilisation en complet de bout en bout

|  |  |  |  |
| --- | --- | --- | --- |
| Étape | Date | Livrable | remarques |
| Spécification | 13 Mars | Document spécification | Un document de 10 pages au minimum |
| Back office | 3 avril | Code | présentation |
| Front office | 17 avril | Code | présentation |
| Reporting | 8 Mai | Code | présentation |
| Soutenance | 15 Mai | Présentation Latex | Durée 10 à 15 min |