ECF

Développer une interface utilisateur de type Desktop

Critères d’évaluation : les compétences suivantes seront évaluées :

* La liste des fonctionnalités
* Le diagramme de Use Case
* Le maquettage
* Le diagramme de classes des classes Métier, Contrôleur, Dao
* L’algorithmie
* Les bases du langage java et les conventions de nommage
* Les concepts de la Programmation Orientée Objet
* Le développement avec une architecture MVC
* La documentation en Javadoc et les commentaires techniques
* Les collections Java
* La gestion des exceptions
* La création de logs
* Le développement des tests unitaires avec JUnit 5 sur les classes Métier
* L’utilisation du Design Pattern Singleton pour la connexion à la base de données
* L’utilisation de Git et d’un dépôt distant GitHub

**Cahier des charges**

Gestion de clients-prospects pour l’entreprise REVERSO

L’entreprise REVERSO veut pouvoir créer, modifier, supprimer ou afficher ses clients ou ses prospects.

* Sur la page d’accueil :
* L’utilisateur devra choisir entre la gestion des clients et la gestion des prospects
* Il choisira ensuite entre la création, la modification, la suppression ou l’affichage. Un label devra lui indiquer s’il gère des clients ou des prospects
* L’utilisateur pourra quitter l’application ou revenir au choix de clients/prospects à tout moment
* Si l’utilisateur choisit la modification ou la suppression, une liste déroulante apparaîtra sur cette page pour lui permettre de choisir la société suivant sa raison sociale.
* Pour la création, suppression, modification, une nouvelle page s’affichera :
* Pour la création, tous les champs seront vides
* Pour la modification, tous les champs seront affichés et pourront être modifiés hormis l’identifiant
* Pour la suppression, tous les champs seront affichés mais aucun champ ne pourra être modifié. Une boite de dialogue demandera confirmation de la suppression après validation
* L’utilisateur devra pouvoir quitter l’application ou revenir à la page d’accueil à tout moment
* Pour l’affichage, une nouvelle page s’affichera :
* Celle-ci affichera les clients de l’entreprise ou ses prospects suivant le choix fait en page d’accueil. La liste sera triée par raison sociale du client ou du prospect
* Toutes les données des clients ou des prospects devront apparaître, sauf les commentaires
* L’utilisateur devra pouvoir quitter l’application ou revenir à la page d’accueil à tout moment

Le client aura comme données : un identifiant, sa raison sociale, son adresse, son numéro de téléphone, son adresse mail de contact, son chiffre d’affaires et son nombre d’employés. L’utilisateur pourra rentrer des commentaires concernant le client.

Le prospect aura comme données : un identifiant, sa raison sociale, son adresse, son numéro de téléphone, son adresse mail de contact, la date où il a été prospecté et s’il a semblé intéressé. L’utilisateur pourra rentrer des commentaires concernant le prospect.

Spécifications techniques

* Le développement se fera en Java avec la bibliothèque graphique Swing
* L’affichage de la liste des clients ou des prospects se fera avec une JTable
* Une Javadoc sera fournie dans un package dédié
* La persistance des données se fera à l’aide de mySql ou SQLServer

Spécifications fonctionnelles

1. Créez une classe Societe abstraite avec les attributs suivants :

* Identifiant : entier
* Raison sociale : chaîne de caractères
* Numéro de rue : chaine de caractères
* Nom de rue : chaîne de caractères
* Code postal : chaîne de caractères
* Ville : chaîne de caractères
* Téléphone : chaîne de caractères
* Adresse mail : chaîne de caractères
* Commentaires : chaîne de caractères

Contraintes :

* La raison sociale devra être saisie et **unique (géré par contrainte dans la base de données)**
* Tous les champs de l’adresse devront être renseignés
* Le téléphone devra être renseigné et avoir au moins 10 caractères
* L’adresse mail devra être renseignée et avoir au moins le caractère « @ » (utilisez un regex)
* Les commentaires ne seront pas obligatoires
* Vous pouvez créer une classe Adresse

1. Créez une classe Client héritant de la classe Societe avec comme attributs supplémentaires :

* Le chiffre d’affaires du client : double
* Le nombre d’employés : entier

Contraintes :

* Le chiffre d’affaires devra être renseigné et être supérieur à 200
* Le nombre d’employés devra être renseigné et être strictement supérieur à zéro

1. Créez une classe Prospect héritant de la classe Societe avec comme attributs supplémentaires :

* La date de prospection : format date (LocalDate)
* Prospect intéressé : chaîne de caractères ou entier

Contraintes :

* La date doit être renseignée et avoir le format « jj/mm/aaaa »
* Les valeurs affichées de « Prospect intéressé » seront « oui » ou « non ». Gérer une énumération et une liste déroulante dans le formulaire ou des CheckBox

1. **Il ne devra y avoir qu’une seule liste déroulante** dans la page d’accueil, qui n’affichera que les clients ou les prospects suivant le choix de l’utilisateur.
2. **L’application aura 3 interfaces utilisateur** : Il ne devra y avoir qu’une seule Frame pour la création, modification et suppression des clients et des prospects. Celle-ci devra donc s’adapter aux choix de l’utilisateur.
3. **Il ne devra y avoir qu’une seule interface utilisateur** pour afficher la liste des clients ou des prospects. Celle-ci devra donc s’adapter au choix de l’utilisateur.
4. Créez une base de données et 2 tables (client et prospect)
5. Dans le package DAO

- Créez une classe dans le package DAO pour la classe Client et une classe dans le package DAO pour la classe Prospect

- Dans ces 2 classes, créer des méthodes ayant le même nom pour :

* Lire toute la table (findAll)
* Lire un enregistrement à partir de son nom (findByName)
* Insérer un enregistrement (create)
* Modifier un enregistrement (update)
* Supprimer un enregistrement (delete)

Vous devez donc avoir au moins 5 méthodes dans votre classe DAO Client et au moins 5 méthodes dans votre classe DAO Prospect

1. Stocker les messages appropriés dans un fichier log

**Travail demandé :**

* La liste des fonctionnalités, le diagramme de Use Case, le maquettage et le diagramme de classes (Métier, contrôleur, DAO), le MCD et le MPD me seront envoyés **sur ma boite mail office 365 (**[**sylvie.touchot@afpa.fr**](mailto:sylvie.touchot@afpa.fr)**) pour validation**.

Le maquettage de l’application et le diagramme de Use Case seront réalisés sur papier ou avec un utilitaire de votre choix.

Le diagramme de classes, comportant les attributs et les méthodes pour chaque classe, le MCD et le MPD seront effectués avec un utilitaire de votre choix.

* Le développement ne commencera qu’après validation de la partie conception.
* Le dossier complet me sera envoyé au plus tard **le vendredi 8 mars 2024 à 12h** sur ma boite mail office 365 ([**sylvie.touchot@afpa.fr**](mailto:sylvie.touchot@afpa.fr)), ainsi qu’un accès à votre dépôt Github.

Contenu du dossier :

* La liste des fonctionnalités
* Le maquettage
* Le diagramme de Use Case
* Le diagramme de classes (Métier, Contrôleur, DAO)
* Le MCD et le MPD
* Le script de la base de données
* Le dossier du projet (y compris le fichier de paramétrage de la base de données, le fichier log, le .gitignore)