

ECF

Développer une application Desktop pour une entreprise

Critères d'évaluation

Les compétences suivantes seront évaluées :

- Conception
 - La liste des fonctionnalités
 - Le maquettage
 - Le diagramme de classes des classes Métier

- Développement
 - L'algorithmie
 - Les bases du langage java et les conventions de nommage
 - Les concepts de la Programmation Orientée Objet
 - La documentation en Javadoc et les commentaires techniques
 - Les collections Java
 - Les Regex
 - La gestion des exceptions
 - La création des logs
 - Les tests unitaires avec Junit5
 - L'utilisation de Git et d'un dépôt distant de genre GitLab, GitHub

Cahier des charges

Gestion de clients-prospects pour une entreprise

Une entreprise 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
 - 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.
Un label devra lui indiquer s'il gère des clients ou des prospects et s'il a choisi une modification ou une suppression
 - L'utilisateur pourra quitter l'application ou revenir au choix de clients/prospects à tout moment
- Pour la création, suppression, modification, une nouvelle page s'affichera :
 - Pour la création, tous les champs seront vides hormis l'identifiant
 - 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 boîte de dialogue demandera la 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 dossier dédié
- La gestion des données se fera à l'aide de Collections Java de type ArrayList
- Un fichier log sera fourni dans un dossier dédié

Spécifications détaillées

1. Classe Societe et classe Adresse

- Créez une classe Societe abstraite avec les variables d'instance suivantes :
 - Identifiant : entier
 - Raison sociale : chaîne de caractères
 - Adresse : objet de la classe Adresse
 - Téléphone : chaîne de caractères
 - Adresse mail : chaîne de caractères
 - Commentaires : chaîne de caractères
- Créez une classe Adresse avec les variables d'instance suivantes :
 - Numéro de rue : chaîne de caractères
 - Nom de rue : chaîne de caractères
 - Code postal : chaîne de caractères
 - Ville : chaîne de caractères

Contraintes :

- L'identifiant sera affecté et incrémenté automatiquement en commençant par 1
- L'identifiant est différent pour chaque classe fille : il commencera donc à 1 pour chaque classe fille
- Il ne pourra pas être modifié par l'utilisateur. Il sera visible en cas d'affichage (liste), de création, de modification ou de suppression pour information
- La raison sociale devra être saisie
- La raison sociale devra être unique. Il ne pourra pas y avoir 2 sociétés qui aient le même nom, que ce soit un client ou un prospect
- Tous les champs de l'adresse devront être renseignés
- Le code postal devra être renseigné et vérifié par un Regex (il devra comporter 5 chiffres)
- Le téléphone devra être renseigné et être vérifié par un Regex
- L'adresse mail devra être renseignée et être vérifiée par un Regex
- Les commentaires ne seront pas obligatoires

2. Créez une classe Client héritant de la classe Societe avec comme variables d'instance supplémentaires :

- Le chiffre d'affaires du client : long
- 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

3. Créez une classe Prospect héritant de la classe Societe avec comme variables d'instance supplémentaires :

- La date de prospection : format date (LocalDate)
- Prospect intéressé : chaîne de caractères

Contraintes :

- La date doit être renseignée et avoir le format « jj/mm/aaaa ». Utilisez un pattern
- Les valeurs affichées de « Prospect intéressé » seront « oui » ou « non ». Gérez une énumération et une liste déroulante dans le formulaire de saisie/modification

4. Les collections des objets devront être gérées dans deux classes à part, une pour les clients, une pour les prospects. Il ne devra y avoir qu'une seule liste déroulante dans la page accueil : La liste déroulante de la page d'accueil n'affichera que les clients ou les prospects suivant le choix de l'utilisateur.

Rq : Les collections peuvent être initialisées avec quelques objets au démarrage de l'application pour faciliter les tests

5. L'application aura 3 interfaces utilisateurs (3 vues) :

- Une vue pour la page accueil
- Il ne devra y avoir qu'une seule vue pour la création, modification et suppression des clients et des prospects. Celle-ci devra donc s'adapter aux choix de l'utilisateur
- Il ne devra y avoir qu'une seule vue pour afficher la liste des clients ou des prospects. Celle-ci devra donc s'adapter au choix de l'utilisateur

6. En option :

- Gérer la persistance des données avec un fichier
- Permettre à l'utilisateur de sélectionner un client ou un prospect dans la page d'affichage, lui afficher la page formulaire pour pouvoir le modifier ou le supprimer ou prendre en compte les modifications faites sur la page

Travail demandé

➤ Conception

- La liste des fonctionnalités, le maquettage, le diagramme de classes des classes Métier me seront déposés **sur Mètis pour validation**

Le maquettage de l'application sera réalisé avec un utilitaire de votre choix

Le diagramme de classes Métier, comportant les variables et les méthodes pour chaque classe, sera effectué avec un utilitaire de votre choix.

➤ Développement

- Le développement ne commencera qu'après validation de la partie conception
- Un lien et une permission sur le dépôt distant me seront fournis avant le lundi 16 décembre 9h30
- Le code sera **déposé sur Mètis avant le lundi 16 décembre 9h30.**