

Travail de session

PROGRAMMATION D'INTERFACE GRAPHIQUE

FABRICE DÉHOULÉ

Table des matières

I.	Équipes du travail	1
II.	Présentation du travail	1
III.	Conception du programme	1
1.	Démarrage du programme.....	1
2.	Gestion des employés	1
3.	Gestion des clients de l'entreprise et de leurs projets.....	2
4.	Affichage des projets.....	3
5.	Compte administrateur	3
IV.	Exigences du programme (critères de correction)	4
1.	Interface	4
2.	Validation des données	4
3.	Exportation des données.....	4
4.	Sécurité du programme.....	4
5.	Organisation du code	4
6.	Robustesse et cohérence du programme	5
7.	Contrôle de version	5
V.	Remise du travail.....	5

I. Équipes du travail

Travail à réaliser en équipe de **2 personnes au plus**

Date de remise : **19 Décembre 2023 à 17h00**

Les travaux remis en retard ne seront pas acceptés

II. Présentation du travail

Vous devez concevoir un logiciel connecté à une base de données MySQL. Le logiciel aura pour but d'aider une entreprise à gérer des projets de ses clients.


Grâce à votre logiciel, l'entreprise sera en mesure de :

- Gérer ses employés
- Gérer ses clients
- Gérer l'ensemble des projets de ses clients

Une personne connectée avec les informations de connexion adéquates est seule en mesure de faire des ajouts, modifications ou suppressions.

III. Conception du programme

1. Démarrage du programme

Lors de l'ouverture du programme, on doit afficher les projets en cours. 

2. Gestion des employés

Un employé a les informations suivantes :

- | | |
|---|--|
| <ul style="list-style-type: none">• Un matricule<ul style="list-style-type: none">○ Unique○ Combinaison des 2 premières lettres du nom, de l'année de naissance, un nombre aléatoire entre 10 et 99○ Exemple : DO-1978-25 pour John Doe né en 1978• Un nom• Un prénom | <ul style="list-style-type: none">• Une date de naissance• Un email• Une adresse• Une date d'embauche• Un taux horaire<ul style="list-style-type: none">○ Au moins 15\$/heure• Une photo d'identité<ul style="list-style-type: none">○ Lien vers une image• Un statut<ul style="list-style-type: none">○ Journalier ou permanent |
|---|--|

Le logiciel doit permettre de :

- Afficher la liste des employés ✗
- Ajouter un employé ✗
- Modifier les informations de l'employé (sauf son matricule, sa date d'embauche et sa date de naissance) ✗
- Changer le statut de l'employé (de journalier à permanent après 3 ans d'ancienneté). ✗

3. Gestion des clients de l'entreprise et de leurs projets ✗

L'entreprise a des clients qui peuvent être des personnes ou d'autres compagnies. Un client a les informations suivantes :

- Un identifiant
 - Unique
 - Une valeur aléatoire allant de 100 à 999
- Un nom
- Une adresse
- Un numéro de téléphone
- Un email

Les clients vont faire la demande de projets. Le programme doit donc permettre de créer des projets pour un client. Un projet a les informations suivantes :

- Un numéro de projet
 - Unique
 - Numéro de client suivi d'une valeur allant de 01 à 99 et de l'année de début
 - Exemple : 255-10-2023
- Un titre
- Une date de début
- Une description (un texte)
- Un budget
- Un nombre d'employés requis (max 5)
- Un total des salaires à payer
- Un client (identifiant du client)
- Un statut (terminé ou en cours)



Tout nouveau projet est créé avec le statut en cours automatiquement (valeur par défaut).

De plus, on doit pouvoir assigner des employés à un projet. Ainsi, pour un projet, on devra conserver :

- Le nombre d'heures travaillées par chacun des employés assignés au projet ✗
- Le salaire à payer pour chacun d'entre eux ✗
 - Taux horaire * nombre d'heures travaillées
 - Vous devez créer une fonction stockée permettant de calculer le salaire de chaque employé en fonction de son taux horaire et le nombre d'heures travaillés sur chaque projet.

ATTENTION : on ne peut pas assigner un employé à un projet s'il est déjà sur un projet qui n'est pas encore marqué comme terminé

Le programme doit aussi :


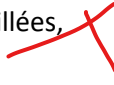

- Afficher la liste des clients 
- Afficher la liste des projets 

4. Affichage des projets

Une page de votre programme doit afficher la liste des projets. Dans la liste des projets, on doit avoir des cartes qui indiquent le numéro du projet, le titre, le nom du client, la date de début et le budget.

Le programme doit permettre que lorsqu'on clique sur un projet de la liste, on arrive sur la page de détails du projet en question.

Sur cette page des détails d'un projet, on doit avoir les fonctionnalités suivantes :

- Afficher toutes les informations du projet 
- Assigner un employé au projet en question
- Afficher la liste des employés avec leur taux horaire, leur nombre d'heures travaillées, leur salaire pour le projet 
- le total des salaires à payer pour tous les employés sur le projet en question 

5. Compte administrateur

Vous devez avoir un compte administrateur dans la base de données.

L'administrateur est le seul autorisé à faire des ajouts/modifications/suppressions. Il faudra donc que le programme offre un mécanisme de connexion au compte administrateur.

Cependant, si l'administrateur n'est pas connecté, on peut toujours parcourir les différentes parties du logiciel (employés, clients, projets).

La création du compte administrateur devra se faire lors de la première utilisation du programme.

Ainsi, on devra entrer le nom d'utilisateur et le mot de passe qui seront stockés dans la base de données. Le mot de passe devra être crypté dans la base de données.

Il doit y avoir également un mécanisme de déconnexion du compte administrateur.

IV. Exigences du programme (critères de correction)

Vous devez développer un logiciel permettant de faire la gestion demandée par la société. Ainsi, toutes les fonctionnalités indiquées devront être implémentées.

1. Interface

Le programme doit avoir une interface intuitive. Les différentes sections du programme doivent être clairement identifiées.

Votre interface devra être visuellement agréable. Vous devez donc mettre l'accent sur la disposition de vos éléments graphiques et l'esthétique en général.

Vous devez utiliser les contrôles adéquats (utiliser une textbox pour entrer un statut n'est pas adéquat par exemple).

Il est de votre responsabilité d'avoir les pages nécessaires afin de répondre aux exigences de l'énoncé.

2. Validation des données

Le programme doit valider les entrées des formulaires. Un **message adéquat** doit être indiqué à l'utilisateur en cas d'erreur. Les messages comme « veuillez remplir les champs » **ne sont pas acceptés**.

Lors d'une opération réussie, on doit avoir un message qui l'indique.

3. Exportation des données

On doit pouvoir exporter les projets dans un fichier CSV. Ce fichier doit contenir les informations d'un projet (on ne tient pas compte des employés du projet).

Les informations d'un même projet devront être dans des colonnes distinctes.

4. Sécurité du programme

Le mot de passe doit être crypté dans la base de données.

On ne peut pas faire d'ajout, de modification ou de suppression si on n'est pas connecté avec les informations du compte administrateur.

Les données entrées dans la base de données doivent être préparées avant d'être insérées (voir notes de cours sur les bases de données).

5. Organisation du code

Votre code devra être bien structuré.

Les noms de classes et des fenêtres/pages/boîtes de dialogue doivent être clairement identifiés et cohérents. Les noms comme Page1, Classe1 vous feront perdre des points.

Les opérations vers la base de données doivent se faire dans un ou plusieurs singletons selon ce que vous jugez nécessaires.

La syntaxe de création d'un singleton doit être respectée. Les classes statiques feront perdre des points.

Les appels vers la base de données, doivent se faire **via des procédures stockées**. Il ne doit donc pas y avoir de code SQL dans votre projet.

6. Robustesse et cohérence du programme

Le programme ne doit pas planter.

Le programme doit être cohérent. Exemples :

- La date de naissance devrait donner au moins 18 ans ou encore pas plus de l'âge de la retraite.
- La date d'embauche ne doit pas être dans le futur.
- Les taux horaires ne doivent pas être négatifs ou trop excessifs ou encore être une valeur non numérique.

7. Contrôle de version

Vous devez avoir votre projet sur le gestionnaire de versions (utilisation de github). Les membres de l'équipe devront y avoir accès ainsi que le professeur.

Le compte est [fabricedehoule](#).

Chacune des personnes de l'équipe devra pousser ses parties.

V. Remise du travail

Vous devez remettre les éléments suivants sur LÉA dans un fichier compressé :

- Votre programme complet.
 - Une seule remise d'un membre de l'équipe est suffisante.
- Un fichier texte avec
 - Votre string de connexion à la base de données
 - Les noms des étudiants de l'équipe et les noms respectifs sur Github.

Attention : il est de votre responsabilité de remettre tous les fichiers de façon adéquate. Des pénalités s'appliqueront si le professeur doit vous contacter pour avoir les bons fichiers.