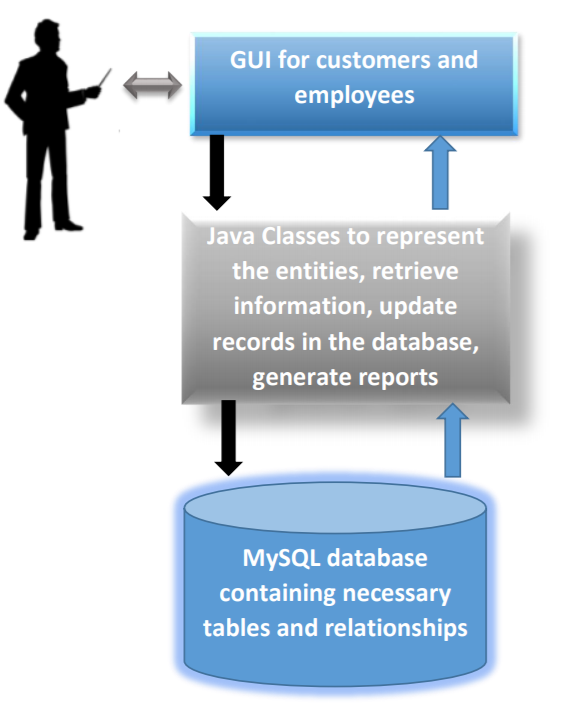
**Objectif :**

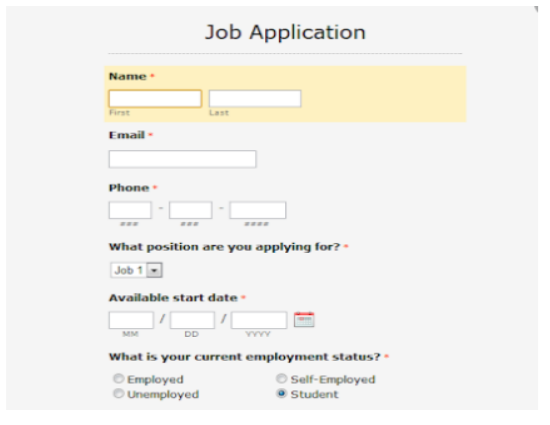
Le but du projet est de permettre à l'emploi recherché de postuler à l'emploi et de l'accompagner les entreprises pour atteindre le bon candidat. L'application prendra également en charge les responsables de l'agence de recrutement pour gérer les dossiers des demandeurs d'emploi et les employeurs.



**Description du programme :**

Dans ce programme, vous écrirez un ensemble de classes de soutien pour les agences de recrutement.

Voici un exemple d'interface graphique.



L'application doit permettre aux demandeurs d'emploi de parcourir tous les emplois de catégorie différente. Les demandeurs d'emploi pourront postuler pour l'emploi qui convient à leur profil.

L'employeur pourra mettre les nouveaux emplois sur l'application au besoin et pourra également passer en revue les candidats qui se sont inscrits ainsi que ceux qui ont postulés pour les emplois. Il y aura deux types d'employeurs : Nouveaux employeurs et employeurs membres. Les nouveaux employeurs devront afficher les emplois sans rabais offert. L'employeur membre peut être de type micro ou non micro. Les employeurs membres pourront afficher les emplois avec une réduction offerte sur les frais basés sur le type d'adhésion.

Les responsables de l'agence de recrutement pourront ajouter de nouveaux emplois, supprimer les anciens emplois et sera également en mesure d'analyser les emplois les plus populaires.

L'application implique principalement des détails sur les emplois disponibles, leurs fonctionnalités et ainsi que les dossiers générés et conservés pour les demandeurs d'emploi et les employeurs.

L'application doit être développée pour trois types d'utilisateurs :

1. Chercheurs d'emploi - Parcourez les emplois, postulez aux emplois, consultez la mise à jour sur demande, calcul de facture avec / sans remises, navigation disponibilité etc.

2. Employeurs - Mettez à jour les emplois actuellement disponibles, ajoutez de nouveaux emplois, vérifiez candidats

3. Agence de recrutement : ajouter / supprimer des emplois, évaluer la popularité des emplois.

Vous devez concevoir et développer la base de données pour cette application en même temps avec les classes Java nécessaires pour implémenter l'application.

**Exigences de mise en œuvre**

• Les classes, méthodes et attributs nécessaires doivent être conçus en utilisant UML notation de diagramme. Toutes les classes, méthodes et attributs doivent être expliqué dans votre documentation. Veuillez discuter de la conception avec moi avant vous commencez la mise en œuvre.

• Vous devez être en mesure d'identifier et d'introduire l'héritage et l'agrégation relation le cas échéant

• Les écrans GUI nécessaires doivent être ajoutés pour une exécution réussie.

• Les enregistrements doivent être conservés dans la base de données. Votre code Java est attendu pour lire et écrire dans plusieurs tables selon les besoins.

• Chaque table doit contenir au moins 6 enregistrements.

**Architecture générale du système (modèle MVC)**

Dans cette section, l'architecture générale de gestion de ce planning a été décrit. Ce système compte principalement 5 modules :

• Le module de recherche d'informations : toutes les exigences possibles dans la base de données, selon plusieurs critères de recherche

• Le module de mise à jour des données : toute modification, ajout ou suppression opération dans la base de données

• Le module Reporting : statistiques sous forme de graphiques (camemberts, histogrammes etc.)

• Le module d'accès aux données (DAO) interroge ou met à jour les données dans la base de données et communique avec les 3 modules précédents

• L'interface graphique communique avec les 3 premiers modules pour visualiser le planning

Selon le modèle MVC, votre interface graphique constitue la Vue (uniquement l'affichage) dépendant des actions de l'utilisateur (événement management) au niveau du contrôleur (modules de recherche, de mise à jour et de reporting).

Cela demandera au modèle de récupérer ou de mettre à jour - via le module d'accès aux données (DAO)

- les informations de la base de données, les organiser et les assembler (par exemple, en les stockant dans des collections). Ensuite, le contrôleur demandera au modèle les données, l'analyser, prendre des décisions et renvoyer le texte à la vue.

Il vous est conseillé d'adopter le modèle pour l'élaboration d'un projet cohérent.

Vous pouvez en savoir plus sur le modèle MVC à l'adresse

<https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4678736-comment-fonctionne-une-architecture-mvc>

<https://accu.org/journals/overload/16/88/grenyer_1524/>

<https://www.oracle.com/technical-resources/articles/javase/mvc.html>

**Conseils pour le développement structuré du projet**

Étape 1 : modèle relationnel

Passez en revue toutes les exigences possibles de la base de données et les critères de recherche. Identifier les entités possibles, les attributs de la base de données. Il est important de soigneusement reconnaître le rôle de chaque attribut, puis décider du type de données de l'attribut. Il jouera également une contribution essentielle à la détermination des attributs clés. Documentez les relations entre les entités.

Étape 2 : création de la base de données

En vous basant sur le modèle de relation ci-dessus, créez des tables et des relations en utilisant MySQL. Insérez les enregistrements dans les tables.

Étape 3 : Recherche d'informations

Passez en revue les exigences des utilisateurs pour identifier la gamme possible d'informations que vous besoin de récupérer de la base de données. Plus précisément dans le cas des entreprises organisations, il est important que les employés analysent les ventes. C'est également indispensable pour que les clients analysent leurs achats passés. Par exemple :

• Le nombre d'emplois annoncés par un employeur particulier

• Le nombre de candidatures déposées par le demandeur d'emploi au cours des trois derniers mois

Développer les classes nécessaires pour représenter les entités qui permettront à l'utilisateur pour interroger la base de données.

Étape 4 : accès aux données

Ce module représente la couche d'accès aux données (DAO) dans la base de données. Via un JDBC accès à la base de données, ce module exécute les requêtes chargées de récupérer ou mise à jour des données dans la base de données. Il s'agit d'un type d'objet qui se charge pour récupérer les données de la base de données et qu'un autre type d'objet soit utilisé pour traiter ces données (couche métier).

Étape 5 : GUI et rapports

Une fenêtre de bienvenue permettra à l'utilisateur de se connecter à la base de données en entrant leur EMAIL et MOT DE PASSE. Ces informations, si elles sont stockées dans la table USER, lui donnera des droits d'accès et / et de mise à jour à certaines données de l'horaire.

Votre interface graphique affichera de manière ergonomique, claire et fluide toutes les informations pertinentes. Il vous permettra de naviguer intuitivement d'une page à un autre. Par exemple, une page de votre graphique d'interface peut contenir des menus avec des éléments de menu, ou des onglets si vous préférez.

Ce module permet de générer des statistiques (camemberts, histogrammes, etc.) en utilisant JFreeChart. Vous pouvez trouver les détails dans la section des ressources.

**Livrables**

Le livrable doit être un fichier compressé par équipe contenant les éléments suivants :

1. Une présentation PowerPoint comprenant

a. Titre

b. Nom des membres de l'équipe

c. Sommaire

d. Diagramme de classes (draw.io) ou tout autre outil similaire

e. Conception de base de données

f. Exemples d'écrans GUI

g. Votre propre évaluation du projet

h. Bibliographie

2. Code Java:

Tous les dossiers et fichiers du projet développé sur Eclipse ou Netbeans avec les sources, l'exécutable .jar en mode graphique et Documentation Javadoc avec des commentaires sur les classes et les méthodes.

**Resources**

JDBC: <https://www.jmdoudoux.fr/java/dej/chap-jdbc.htm> (Author: Jean-Michel Doudoux)

My SQL: <http://dev.mysql.com/doc/refman/5.7/en/>

JFreeChart:

The JFreeChart Class Library (Author: David Gilbert) <http://www.jfree.org/jfreechart/api/javadoc/index.html> <http://www.java2s.com/Code/Java/Chart/CatalogChart.htm>

<http://www.jfree.org/forum/>

Wireframe: <https://webdesign.tutsplus.com/articles/a-beginners-guide-to-wireframing--webdesign-7399>