Une image contenant Police, conception, illustration

Description générée automatiquement

Coline BAUDRIER POULIQUEN – DIA2 – 28/02/2024

Compte rendu – Annuaire d’entreprise

Réalisation d’une API et d’un client lourd en Java

# Hogwarts Company – Cahier des charges

## Présentation du projet

Dans le cadre de cette évaluation, vous allez créer une application d'annuaire d'entreprise. L'application va être architecturées de cette façon :

* Création d'une base de données
* Manipulation des données à travers une API (Java ou C#)
* Interface utilisateur (client lourd)

## Contexte

Développeur informatique dans une industrie agroalimentaire sur le territoire française, l'entreprise en croissance est aujourd'hui composée avec 5 sites différents et plus de 1000 salariés :

1. Paris : siège administratif
2. Nantes : site de production
3. Toulouse : site de production
4. Nice : site de production
5. Lille : site de production

Il est nécessaire de fournir une application permettant de retrouver des informations sur chaque collaborateur :

* Nom
* Prénom
* Fixe
* Portable
* E-mail
* Service de travail
* Site

## Rôles dans l’application

* **Visiteur** : rechercher et afficher des informations
  + Doit pouvoir afficher la fiche d'un autre salarié
  + Doit pouvoir faire une recherche :
    - Par nom, prénom : saisie de X lettres pour afficher les salariés avec une correspondance
    - Par site avec une liste complète des sites
    - Par service avec la liste des services
* **Administrateur** : ajouter, modifier, supprimer des informations
  + Doit pouvoir avoir les mêmes droits que le visiteur
  + Doit pouvoir faire un CRUD :
    - **Site** : champ ville uniquement
    - **Service** : comptabilité, production, accueil, informatique
    - **Salarié** : nom, prénom, téléphone fixe, portable, e-mail, service et site de travail

L'accès à l'espace administrateur se fait avec une combinaison de touches et un mot de passe.

# Mise en place

Ce projet a été réalisé par une seule personne avec trois chantiers :

* Base de données cohérente
* API
* Client lourd

## Choix techniques

Pour réaliser ce projet, différents choix techniques ont été réalisés :

* **Suivi de projet** : GITLAB avec des issues et des jalons
* **Versionning du code** : GITLAB avec un groupe de projet et un projet différent pour l'API et le client lourd
* **Base de données** : MariaDB, gestion avec DataGrip
* **API** : Java, code avec IntelliJ
* **Client lourd** : Java et JavaFX, code avec IntelliJ et Scene Builder
* **Cahier de test** : Rédigé sous Excel, avec une version .md
* **Compte rendu de projet** :
  + Version écrite : Word et un .md
  + Version orale : Prize

## Liste des tâches

### API Rest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | DAO | CRUD | Business | Controller | Autres |
| Worksite | Worksite | Worksite | Worksite | Worksite | Requête SQL sécurisées |
| Service | Service | Service | Service | Service |  |
| Employee | Employee | Employee | Employee | Employee |  |
| Address | Address | Address | Address | Address |  |
|  |  |  |  |  |  |

### Client lourd

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Général | Worksite | Service | Employee |
| Connexion | X |  |  |  |
| Accueil | X |  |  |  |
| Liste |  | X | X | X |
| Ajout |  | X | X | X |
| Modification |  | X | X | X |
| Suppression |  | X | X | X |
| Recherche | X | X | X | X |

## Base de données

La base de données est composée de quatre tables : vous pouvez trouver les fichiers concernés dans */resources/sql*.

* **Worksite** : contient toutes les données liées aux sites de travail, détient une adresse
* **Service** : contient toutes les données liées aux services, rattachés à un Worksite, détient une adresse
* **Employee** : contient toutes les données liées aux employés, rattachés à un Service
* **Address** : contient toutes les adresses définies

## API Rest

Pour une maintenabilité du code, vous pouvez trouver tous les commentaires sur les fichiers qui concernent les **Worksites**, le but étant de ne pas alourdir le code complètement. L'API est gérée avec 4 packages et une application :

* **Business** : couche qui fait le lien entre la base de données et l'API
* **Controller** :
  + Le fichier "**Controller**" : détermine les end-points
  + Les fichiers "**Model**" : détermine les aspects des class par rapport à ce que nous avons déterminé dans la base de données
* **DAO** : couche qui réalise toutes les requêtes (HTTP et SQL) qui vont permettre l'utilisation des données et la discussion avec la BDD
* **Utils** : contient les mappers, ils permettent de gérer la discussion entre les class DTO (BDD) et les class Java, afin de factoriser le code lors de la création des DAO

## Client lourd

Pour une maintenabilité du code, vous pouvez trouver tous les commentaires sur les fichiers qui concernent les **Worksite**, le but étant de ne pas alourdir le code complètement. Le client lourd est géré avec plusieurs packages :

* **Côté applicatif :**
  + **DAO** : gère les end-points de l'API en les appelant avec la bonne méthode pour retourner le résultat attendu
  + **model** : gère toutes les classes utilisées, pour correspondre à celle de l'API afin de ramener et d'envoyer les bonnes données au format JSON
  + **controllers** : tous les controllers se trouvent à la racine du projet, on trouve un controller par type d'action que l'utilisateur peut réaliser
    - ClassController : affiche la liste des données comprises dans la class
    - AddClassController : permet d'affiche la fenêtre d'ajout
    - DetailClassController : au double clic sur une ligne du tableau récapitulatif, ouvre une fenêtre de détails qui permet de réaliser les actions suivantes : modification et suppression (dans le cas où vous avez les droits nécessaires)
    - HeaderController : menu qui reste sur chaque page, permet de se déplacer dans les différents menus
    - SearchController : page de recherche, le visiteur peut cliquer dessus pour accéder à la page et lancer une recherche qui lui donnera des résultats dans toutes les tables
    - ConnexionController : page de connexion qui permet une connexion en tant que visiteur ou administrateur avec une combinaison de touches et un mot de passe
    - HomeController : page d'accueil, chaque personne arrive dessus après la connexion, cela évite que l'utilisateur soit surchargé d'informations au premier coup d'oeil, rend l'expérience plus fluide et agréable
* **Côté front :**
  + **assets** : contient les images et les icons utilsées
  + **application** : tous les fichiers FXML qui gèrent l'apparence

# Mise en production

La mise en production a pu avoir lieu sur tous les ordinateurs de l'entreprise avec le téléchargement d'un exécutable et la transmission d'une documentation. Le service informatique aura reçu tout le .md ainsi qu'une copie du projet non modifiable pour les personnes qui n'ont pas la main sur le projet. Les personnes souhaitant avoir accès au code dans le cadre d'un recrutement doivent contacter Olympe Maxime qui est en charge du projet afin qu'elle puisse donner les accès ou au moins valider la demande.

Les utilisateurs quant à eux auront reçu le mode d'emploi au format PDF avec le contact des personnes en charge du projet afin de faire remonter des demandes d'amélioration.

## Mode d’emploi

### Partie pour les développeurs

* Lancer votre système de gestion de base de données (WampServer)
* Démarrer votre API en passant par le code de phonebook : **PhonebookApplication**
* Démarrer votre client lourd en passant par le code de software : **Main**

Dans le cas où vous rencontrez un bug, vous pouvez dans un premier temps vous référer au cahier de test afin de vérifier si c'est un cas de figure déjà trouvé et qu'il est donc en cours de résolution ou si n'a pas été décelé par nos équipes. Vous pouvez alors contacter Olympe Maxime ou l'un des développeurs en charge, si vous n'êtes pas à même de réaliser les modifications dans le code.

### Partie utilisateur

**Dans le cadre d'une connexion en tant que visiteur, vous avez juste à :**

* Cliquer sur le bouton de connexion à l'ouverture du logiciel
* Vous pouvez vous balader dans les différents menus qui se trouvent dans la barre du haut, ils vous sont tous accessibles
* Lorsque vous arrivez sur la liste des sites de travail/services/employés, vous pouvez cliquer sur l'une des lignes pour ouvrir les détails
* Pour effectuer une recherche, taper le terme recherché dans la barre et cliquer sur le bouton loupe ou sur entrée : les résultats trouvés s'afficheront dans chacun des onglets concernés.
* Attention pour la recherche : elle se fait sur le nom du site, le nom du service, le nom et le prénom de l'employé
* Pour vous déconnecter, vous pouvez soit cliquer sur la croix du programme, soit sur le bouton de déconnexion en haut à droite (icône d'un ordinateur avec une flèche sortante)
* Vous pourrez voir des boutons comme "Ajout", "Modifier", "Supprimer" mais ils vous seront inaccessibles dans ce mode de connexion
* Partie administrateur

**Dans le cadre d'une connexion en tant qu'administrateur, vous devez :**

* Faire la combinaison de touches CTRL + TAB et taper le mot de passe
* Si la connexion se fait, vous avez alors accès aux mêmes interfaces que les visiteurs
* Les boutons vous sont cependant accessibles et vous pouvez réaliser les actions d'ajout, modification et suppression
* Il n'y a aucune restriction de connexion à plusieurs administrateurs : vous pouvez réaliser toutes les actions en même temps que les autres, le système d'ID étant géré par la base de données, vous ne pouvez influer dessus et les conflits peuvent se résoudre comme cela dans un premier temps

Dans le cas où vous êtes confrontés à un bug, un mauvais retour, une mauvaise mise à jour de la base de données, nous vous demandons de contacter Olympe Maxime ou l'un des membres de son équipe. Une adresse électronique de retours sera mise en place dans quelques temps, nous vous informerons en temps voulu.

# Projets pour la version 2

Ici nous vous proposons une première version du projet, plusieurs mises à jour sont amenées à apparaître :

* **Connexion en fonction de vos identifiants :** permettra de garder des logs de connexion et des modifications qui ont été réalisées
* **Mise en place de favoris sur la page d'accueil :** possibilité de mettre des services et sites de travail en favoris et accessibles directement sur la page d'accueil
* **Possibilité de laisser des demandes :** possibilité d'attacher une demande de modification, d'ajout, dans le cas d'un visiteur sans droit administrateur
* **Refonte UI en fonction des retours :** une enquête sera menée auprès des utilisateurs afin de récupérer leurs premières impressions