Projet Winforms

# Présentation

Rappel du contexte : un éditeur de logiciels scientifiques souhaite mettre en place un système de gestion de tâches pour mieux suivre et planifier sa production. Comme ce n’est pas sa spécialité, il souhaite vous sous-traiter la mise en place du système, nommé « Job Overview ».

La présente partie concerne l’accès aux données stockées en base, et la création de l’interface visuelle permettant de consulter et modifier les données. L’éditeur souhaite que l’application fonctionne en mode MDI, avec une base de données SQL Server.

# Eléments fournis

|  |  |  |  |
| --- | --- | --- | --- |
| **Backup de la base** | **MPD** | **Données de tâches de production et temps de travail** | **Solution Visual Studio qui implémente le modèle MDI** |
| Fichier donné en début de projet |  |  |  |

# Hypothèses et contraintes

## Techniques

**Structure de la solution**

On séparera la couche d’accès aux données en différents fichiers cs afin de faciliter la maintenance du code. Chaque fichier traitera d’un ensemble cohérent de concepts (ex : DALLogiciel.cs contiendra ce qui concerne un logiciel, ses versions et ses modules).

Les fichiers à l’intérieur de la solution devront être correctement nommés et organisés pour qu’on s’y retrouve facilement.

**Connexion à la base de données**

La chaîne de connexion à la base de données doit être modifiable sans avoir à recompiler l’application.

## Fonctionnelles

**Utilisateur de l’application**

Comme l’application ne proposera pas de fenêtre de login, on supposera que la personne qui utilise l’application est le chef de projet de l’équipe «Dev Bio humaine ». Les fenêtres de l’application n’auront pas à afficher de listes déroulantes de sélection de filière ou d’équipe.

**Visuel et ergonomie**

L’éditeur a insisté pour que le visuel et l’ergonomie de l’application soient soignés, afin que son utilisation au quotidien soit agréable. Il compte sur votre expérience pour définir l’ergonomie la plus adaptée à chaque fenêtre, dans le respect des contraintes imposées.

# Développements à réaliser

Voici les fenêtres que l’éditeur souhaite avoir dans l’application :

## Gestion des versions de logiciels

Cette fenêtre affichera une liste déroulante de sélection d’un logiciel, et affichera les modules et versions du logiciel sélectionné. Elle affichera également le numéro de la dernière release de chaque version.

NB/ les modules seront affichés à plat, sans les relations hiérarchiques entre eux

Il sera possible d’ajouter de nouvelles versions, et d’en supprimer si elles n’ont aucune tâche de production associée.

## Gestion des taches de production

**Fenêtre MDI fille**

Cette fenêtre permettra au chef de projet de consulter les tâches des personnes de son équipe, afin de suivre l’avancement du projet. Pour une personne et une version de logiciel données, il souhaite avoir le détail des tâches de production classées par numéro, et pouvoir filtrer les tâches terminées ou non terminées (choix par défaut)

Les tâches seront affichées en tableau, mais la description de la tâche courante (celle qui est sélectionnée) sera affichée dans une zone de texte à part, pour plus de lisibilité.

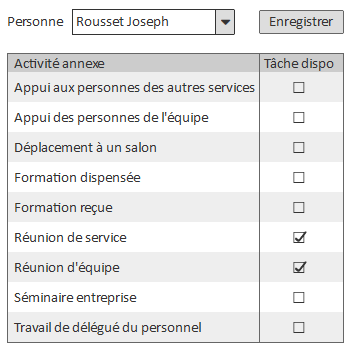
**Fenêtre Modale de saisie**

La fenêtre MDI fille donnera accès à la création d’une nouvelle tâche via une fenêtre modale, pour la personne et la version sélectionnées. Comme les managers créent souvent plusieurs tâches d’affilée, l’éditeur souhaite que la saisie soit rapide, et que l’enregistrement en base soit optimisé pour que l’application reste toujours réactive.

## Gestion des tâches annexes

Chaque personne ne peut avoir qu’une seule tâche par activité annexe

La fenêtre ci-dessous présente l’ensemble des activités annexes. Lorsqu’on coche une case, cela crée une tâche pour l’activité correspondante et pour la personne sélectionnée. Lorsqu’on décoche une case, cela supprime la tâche correspondante, sauf si du temps de travail a déjà été saisi sur cette tâche. Dans ce cas, un message d’avertissement doit s’afficher à l’enregistrement.



## Vérification des saisies de temps

L’application doit proposer une fenêtre permettant au manager de vérifier si les personnes de son équipe ont bien saisi tous leurs temps sur un mois qu’il aura sélectionné. Chaque personne doit avoir saisi 8h pour chaque jour du mois, hormis les jours de week-end. Le manager soit pouvoir savoir sur quels jours la saisie est incomplète.

NB/ On supposera qu’il n’est pas possible pour une personne de saisir plus de 8h de travail par jour. Les congés ne sont pas gérés par le modèle de données.

## Echanges de données

L’application JobOverview doit être capable d’importer des tâches de production et travaux associés, que l’éditeur gérait auparavant sous Excel, et qu’il a exportées au format xml dans le fichier fourni plus haut.

L’application devra également être capable d’exporter la liste complète des tâches de l’équipe, sous le même format.

Les opérations d’importation / exportation de données seront lancées directement depuis des sous-menus du menu « Echanges » de l’application. Un message de confirmation devra s’afficher à la fin de l’opération.

## Tests unitaires

Coder les tests unitaires qui vous semblent pertinents pour tester vos méthodes de DAL.

# Livraison

Vous devez mettre sur GitHub, dans un référentiel nommé ProjetWinforms :

* L’ensemble des sources : solution Visual Studio complète, scripts SQL éventuels
* L’exécutable et le fichier de configuration associé, afin de pouvoir changer la chaîne de connexion à la base de données
* Une sauvegarde de la base de données
* Une documentation utilisateur succincte pour le guider dans l’utilisation de chaque fenêtre de l’application
* Une documentation technique à destination d’un développeur, et qui explique dans les grandes lignes l’architecture technique de chaque fenêtre et méthodes de DAL associées.