Documentation technique

L'objectif de cette documentation est de préciser dans les grandes lignes l'architecture technique de chaque module de l'application.

Structure de la solution

L'application respecte l'architecture MVVM. Elle est ainsi découpée en trois couches, représentées par des dossiers contenus dans la solution.

- « View » : contient les vues représentant l'interface visuelle utilisateur.
- « Model » : contient les DAL gérant les opérations et les interactions avec la base de données.
- « ViewModel » : contient les contextes associés aux vues, permettant un lien entre les vues et les modèles.

La solution est également composée d'un dossier « Entity » contenant toutes les classes POCO nécessaires à la gestion de l'application.

Des convertisseurs sont également disponibles dans la racine de la solution.

L'application est découpée selon six modules :

- Connexion
- Gestion des tâches de production
- Gestion des tâches annexes
- Saisie de temps
- Exportation des données
- Synthèse des versions

Module « Connexion »

Réalisé

A l'ouverture de l'application la liste de toutes les personnes est récupérée pour alimenter la combobox. Une fois une personne choisie, l'enregistrement de son code (unique) est fait dans les paramètres de l'application (portée utilisateur). Ensuite on effectue une requête avec comme paramètre le code de la personne pour récupérer toutes les personnes de son équipe et elle-même.

Si la requête renvoie plusieurs personnes alors la personne connectée est responsable d'une équipe et le boolean « Manager » qui est enregistré dans les paramètres de l'application passe à true (false par défaut). On débloque alors certaines fonctionnalités dans l'application suivant ce paramètre.

Module « Saisie de temps »

Réalisé

L'interface associée à ce module permet à toute personne connectée de consulter les tâches de production. En temps normal elle devrait pouvoir saisir les temps de travail suivant la date choisie.

Les taches de production s'affichent dans une gridview à gauche et les taches annexes dans une gridview à droite. Ces listes sont remplies grâce à la base de données et aux paramètres Logiciel et Version disponibles dans des Combobox en haut de l'interface.

Dans un expand latéral à droite se trouvent les informations des taches de production et annexe sélectionnées, liées aux gridview respectives. Les champs des heures et celui de la durée restante sont saisissables mais non implémentés.

Les données sont récupérées en totalité, depuis la base, et traitées avec du linq en interne. De plus l'affichage est directement lié aux radios boutons de filtrage.

Optimisation

Finir les impératifs non terminés :

- Saisie et enregistrement des temps de travail en base de données (création des nouveaux travaux ou update des travaux existants modifiés)
- Affichage de la description dans le champ correspondant

Module « Gestion des tâches annexes »

Réalisé

L'interface associée à ce module permet au manager d'une équipe de créer ou supprimer des tâches annexes de chaque personne de son équipe, lui compris. La liste des activités annexes est fixe et chaque personne ne peut avoir qu'une seule tâche par activité annexe.

Concernant l'accessibilité des fonctions de la fenêtre, les éléments de gestion des tâches annexes sont uniquement disponibles si l'utilisateur courant a un statut de manager. Dans le cas contraire, ces éléments sont grisés et l'utilisateur n'accède finalement qu'aux informations le concernant. De plus, seul le manager peut accéder à l'ensemble des membres de l'équipe. Pour les autres utilisateurs, seules leurs informations sont accessibles.

Lors du lancement du module, la liste des membres de l'équipe concernée avec leurs tâches annexes est chargée via une requête. Pour chaque membre de l'équipe, la liste des tâches annexes est complétée pour faire apparaître la totalité des activités annexes disponibles (la liste des activités est chargée par requêtage). L'affectation ou non d'une activité à une personne est alors gérée à l'aide d'un booléen.

Les informations obtenues sont affichées dans une ListBox et une ListView, selon un mode maître-détail. La liste des personnes est affichée dans la ListBox (maître) et le détail de son élément courant est affiché dans une ListView.

Des CheckBox permettent d'afficher le statut d'assignation des activités annexes et un bouton d'enregistrement permet de lancer une méthode implémentant l'ajout et la suppression des tâches annexes dans la base de données. Après appui sur la touche d'enregistrement, la liste des tâches actuelle est comparée à celle de départ, permettant de détecter les changements effectués par le manager. Pour chaque tâche annexe dont l'assignation associée a changé de valeur, une requête d'ajout ou de suppression de cette tâche est lancée.

Optimisation

Confort graphique :

- [Gestion des tâches annexes] --> Lors de la modification d'une tâche d'une personne, il faudrait verrouiller la liste des employés. Il faudrait alors cliquer sur le bouton enregistrer pour pouvoir sélectionner un autre employé (ce cas se présente seulement si l'utilisateur de l'application a le statut de manager).
- [Gestion des tâches annexes] --> Il reste à gérer le rafraîchissement du champ description suite à une suppression.

Gestion de la synthèse des versions

Elle se fait à l'aide d'une méthode qui récupère tous les logiciels avec leurs versions et modules.

Optimisation

Colorier les listes en fonctions des résultats

Effectuer le calcul de l'écart directement en base pour gérer plus facilement le type retourné

Gestion des tâches

Les entités tâches sont définies en 2 formats (tache,tache Prod et tacheAnnexe) et tacheApercu.

TacheApercu afficher à plat toutes les tâches avec travail existantes pour permettre la visualisation des tâches qui seront exportées. Cette entité permet aussi de gérer la sauvegarde des tâches pour l'enregistrement en base de données.

Particularité: La saisie des tâches de production. Elle se fait en en 2 temps:

- Une partie visuelle où l'utilisateur saisit toutes ses tâches et les voit en totalité avant d'enregistrer

- Une partie behind où l'enregistrement se fait en masse

La suppression des tâches se fait une par une directement en base. On va du principe qu'il n'y aura pas une grande quantité de tâches à supprimer

Les types tables permettant l'insertion de masse en BDD sont fournis dans les livrables

La gestion des tâches de production se fait avec les mêmes requêtes utilisées pour l'export.

La méthode d'export a été définie avec un XMLWriter pour une meilleure lisibilité. Il suffit d'ouvrir le fichier Xml avec Excel pour traiter les données.

Chargement des données dans les fenêtres

A chaque ouverture de fenêtre, la liste des données concernées est chargée de la base en fonction de la personne connectée.

Les requêtes sélectionnent le maximum d'information à l'ouverture de la fenêtre. Tout le reste du traitement se fait par des requêtes Linq en général en fonction de la version et du logiciel sélectionnés.

Pour une meilleure optimisation des performances, on réutilise les listes chargées. C'est la raison pour laquelle elles sont chargées dans le VMMain afin d'être accessibles pour toutes les fenêtres.

Gestion des fenêtres

Des UC sont réutilisés dans d'autres pour une meilleure optimisation. Il s'agit de celui concernant la liste des logiciels et versions qui apparait dans la majorité des fenêtres.

Les styles sont en majorité ajoutés au début de chaque fenêtre mais une partie est aussi définie dans le fichier App.xaml.