# View

## Sélection de la chaine de connexion (dans les propriétés de l’appli)

En mode débug : sélection dans une combobox au lancement de l’appli sur toutes les strings commençant par « JobOverviewConnectionString ».

En mode release : sélection du string « JobOverviewConnectionStringDefault ».

Mode utilisation en énumération => évolutif (ajout d’un mode admin par ex)

Fenêtre modale pour création de taches avec champs pré remplis mais modifiable : plus cohérent avec une insertion

Expander pour la saisie de temps : plus cohérent avec une modification

# Model

## UpdateDatabaseWorkTimeOfTaskList

Utilisation d’un merge pour faire l’ajout, la suppression et la modification en une seule requête/connexion. Plus efficace car insertion de masse.

## UpdateDatabaseTaskListOfEmployee

Séparation des requêtes d’ajout et de suppression de taches en masse car celles-ci nécessitent de travailler sur deux tables différentes (jo.Tache, jo.TacheProd) empêchant le merge.

Le choix de faire l’ajout et la suppression dans la même méthode permet de n’utiliser qu’une connexion et de s’assurer que le tout ne sera effectué que si aucune erreur ne se produit (grâce à la transaction)

## ExportListTaskEmployeeToXML

Utilisation d’un XMLSerializer car l’architecture de notre liste d’employés correspond exactement à celle du fichier XML voulu.

# ViewModel

## VMTaskConsultation

### ListTaskToAddOrDelete

Le choix de ne créer qu’une liste pour les insertions, les modifications et les suppressions permet de faire appel à une seule méthode de DAL, il fallait pouvoir distinguer les modifications des suppressions et pour cela la propriété Productivity du temps de travail est passée à -1 dans le cas d’une suppression (une productivité ne peut pas négative dans la logique métier).

### AddWorkTime

Cette méthode permet de créer un nouveau temps de travail et l’affecter au temps de travail courant et de passer en mode  « édition ».

### ModifyWorkTime

Cette méthode permet de passé en mode « modify ».

### DeleteWorkTime

Ajout du temps de travail courant à une nouvelle liste de temps de travail qui sera elle-même ajouter à la liste « ListTaskToAddOrDelete » comme propriété d’une nouvelle tâche.

L’Id de cette nouvelle tâche sera égal à l’Id de la tache courante.

### ValidateWorkTime

Cette méthode permet de faire la vérification de la validité des données rentrées par l’utilisateur (temps de travail avec date unique pour chaque tache et temps maximum de travail total sur une journée de 8h).

Si les données sont valides, on les inserts dans la liste « ListTaskToAddOrDelete » de la même manière qu’avec la suppression et repasse en mode « consultation ».

Si les données ne sont pas valides on reste dans le mode actuel et on affiche un message d’erreur pour signaler la non validité des données.

### CancelWorkTime

Cette méthode permet de repasser en mode « consultation » et affecte le premier temps de travail de la tache courante au temps de travail courant.

### SaveModification

Cette méthode permet d’appeler la méthode de DAL pour insérer la liste « ListTaskToAddOrDelete » dans la base de données.

On réinitialise ensuite la liste « ListTaskToAddOrDelete »