Projet JobOverview – partie 1

# Présentation

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 le développement de cette application, qui sera nommée « Job Overview ».

L’éditeur développe plusieurs logiciels, organisés en projets indépendants. Chaque logiciel est développé progressivement en plusieurs versions.

## Métiers et activités

Un projet rassemble des personnes de différents métiers, et chaque métier a un ensemble d’activités associées, résumées par le tableau ci-dessous :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Métier  Activité | ANA  Analyste | CDP  Chef de projet | DEV  Développeur | DES  Designer | TES  Testeur |
| DBE - Définition des besoins | ✓ |  |  |  |  |
| ARF - Architecture fonctionnelle | ✓ | ✓ |  |  |  |
| ANF - Analyse fonctionnelle | ✓ | ✓ | ✓ | ✓ |  |
| DES - Design |  |  |  | ✓ |  |
| INF - Infographie |  |  |  | ✓ |  |
| ART - Architecture technique |  | ✓ | ✓ |  |  |
| ANT - Analyse technique |  |  | ✓ |  |  |
| DEV - Développement |  |  | ✓ |  |  |
| RPT - Rédaction de plan de test |  |  |  |  | ✓ |
| TES - Test |  | ✓ | ✓ |  | ✓ |
| GDP - Gestion de projet |  | ✓ |  |  |  |

Il y a aussi des activités annexes liées à la vie dans l’entreprise, et communes à tous les métiers. Exemples : aide des collègues de son équipe, appui auprès d’autre services, participations à diverses réunions, travail de délégué du personnel, déplacement à des évènements (salons, séminaires) …etc. Cette liste n’est pas figée. Elle est saisissable à partir de l’application.

## Tâches

Chaque personne peut avoir des tâches de 2 natures :

* Des tâches de production, qui se rapportent à une activité liée à son métier. On souhaite suivre l’avancement de ces tâches, c’est-à-dire la date de début, la durée de travail prévue, la durée de travail réalisée, et la durée de travail restante. La durée restante est ajustable au cours du temps, car il se peut que la tâche prenne plus ou moins de temps que prévu. Une tâche est considérée comme terminée lorsque sa durée de travail restante est nulle.
* Des tâches annexes, liées aux activités annexes. Ces tâches ne font l’objet d’aucune planification, mais on veut pouvoir connaître le cumul de temps passé sur chacune d’elles tous les mois.

## Jeu de données

On se propose de suivre la production du logiciel Genomica, qui a commencé en 2016. Voici ses versions, et les personnes qui le développent :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Versions** | | | |  | **Personnes** | | | |
| **Numéro** | **Millésime** | **Date début** | **Date publi** | **Code** | **Prénom** | **Nom** | **Métier** |
| 1.00 | 2017 | 02/01/16 | 08/01/17 | GL | Geneviève | LECLERCQ | ANA |
| 2.00 | 2018 | 28/12/16 | \_ | AF | Angèle | FERRAND | ANA |
|  |  |  |  | BN | Balthazar | NORMAND | CDP |
|  |  |  |  | RF | Raymond | FISHER | DEV |
|  |  |  |  | LB | Lucien | BUTLER | DEV |
|  |  |  |  | RB | Roseline | BEAUMONT | DEV |
|  |  |  |  | MW | Marguerite | WEBER | DES |
|  |  |  |  | HK | Hilaire | KLEIN | TES |
|  |  |  |  | NP | Nino | PALMER | TES |

Les tâches associées à ces versions et personnes, sont décrites par le fichier ci-dessous. Il ne contient que des tâches de production, les colonnes sont séparées par des tabulations, et les durées sont exprimée en jours. Il ne sera pas nécessaire de gérer d’éventuelles erreurs de contenu du fichier.



# A faire

## Modélisation du système

* Modéliser l’application Job Overview en UML, avec le(s) diagramme(s) qui vous paraîtront les plus pertinents.
* Implémenter ce modèle en C#
* Créer une classe nommée « DAL » (Data Access Layer), qui crée ou charge l’ensemble des données décrites précédemment. L’application travaille sur les données d’un seul logiciel à la fois. Ces données seront exposées au travers de 4 propriétés représentant :
  + Le logiciel et ses versions
  + La liste des métiers et activités associées
  + La liste des personnes avec leur métier
  + La liste des tâches

## Saisie des activités annexes

Au démarrage, l’application doit permettre de saisir la liste des activités annexes (code et libellé). Chaque code d’activité doit être unique.

Lorsque la saisie est terminée, vider l’écran et afficher la liste complète des activités annexes, triées par code.

NB/ Il n’est pas demandé de sauvegarder cette liste dans un fichier.

## Suivi de la production

Créer une classe Results avec des méthodes qui renvoient :

* Les durées de travail réalisée et restante d’une personne sur une version
* Le nombre de jours et le pourcentage d’avance ou de retard sur une version (différence entre durées de travail prévues et réalisées totales)
* Les durées totales de travail réalisées sur la production d’une version, pour chaque activité.

A titre de vérification :

* Sur la version 2.00, Geneviève Leclercq a réalisé 58 jours de travail, et il lui reste 21 jours de planifiés. Marguerite Weber a réalisé 72 jours de travail, et il lui reste 11 jours de planifiés.
* Sur la version 1.00, la durée de travail réalisée a dépassé de 3j la durée prévue, ce qui représente un pourcentage proche de 0.
* Quelques durées de travail réalisées sur la version 1.00 :
  + Analyse fonctionnelle : 295 j
  + Architecture technique : 95 j
  + Développement : 259 j
  + Test : 248 j

Afficher les résultats de quelques appels aux méthodes de la classe Results, après l’affichage des activités annexes.

## Livraison de l’application

L’ensemble des sources, diagrammes et exécutable final devra être disponible sur GitHub, dans un référentiel nommé ProjetConsole. L’ensemble doit être suffisamment commenté et documenté pour permettre une compréhension rapide du code.