

UNIVERSITÉ DE FRANCHE-COMTÉ

L3 CMI Informatique

STAGE DE LICENCE 3

Développement de fonctionnalités pour une application web de gestion d'évènements

BENZ-INFO SOLUTIONS



Auteur : BORDE Corentin

Maître de stage : LAPOINTE Benoit

Enseignant tuteur : HERRMANN Bénédicte

Année universitaire 2018-2019

Remerciements

Je remercie la société Benz-Info Solutions de m'avoir accueilli pendant 3 mois et plus particulièrement Monsieur LAPOINTE Benoit, directeur de la société, pour l'aide et les différents conseils donnés lors du développement. Je remercie également Madame HERRMANN Bénédicte, enseignant chercheur et tuteur universitaire, pour les différents conseils donnés pour l'écriture du rapport de stage et le temps passé à le relire.

Table des matières

Introduction	1
1 Contexte et objectifs du stage	2
1.1 Présentation de Benz-Info Solutions	2
1.2 Présentation de la mission : développement de fonctionnalités dans l'application web de gestion HSEQ	3
1.3 Présentation d' <i>Intelex</i> et des méthodes de travail	3
2 Gestion de versions de réglementations dans le module <i>Laws and Regulations</i>	6
2.1 Présentation du module <i>Laws and Regulations</i> et des attentes	6
2.2 Phase d'analyse du module <i>Laws and Regulations</i>	6
2.3 Développement de la fonctionnalité	10
3 Modifications à apporter dans le module <i>Training Management</i>	17
3.1 Présentation du module <i>Training Management</i>	17
3.2 Dispenser des cours	19
3.3 Modifier la date d'expiration d'une qualification	20
3.4 Suppression des qualifications vides	22
4 Présentation succincte des autres tâches	23
5 Bilan	24
5.1 Bilan des missions réalisées	24
5.2 Bilans professionnels et personnels	25
Conclusion	26

Table des figures

1	Diagramme de classe de <i>Laws and Regulations</i>	7
2	Diagramme d'activité représentant le <i>Workflow</i> de <i>Regulation</i> <i>Revision Evaluation Form</i>	9
3	<i>Action Handler</i> de <i>Law or Regulation Revision</i>	12
4	Code source simplifié d'une vue	14
5	Capture d'écran de la page pour le formulaire d'évaluation . .	16
6	Diagramme de classe <i>Training Management</i>	18
7	<i>Workflow</i> de <i>Manual Training Record</i>	20
8	<i>Action Handler</i> de <i>AL - Qualification Status</i>	23

Introduction

Dans le cadre des études, les étudiants en Licence 3 Informatique doivent effectuer un stage de trois mois dans un service informatique d'une entreprise. Ce stage a pour but de mettre en pratique les connaissances acquises durant la formation dans un cadre professionnel, mais aussi d'en acquérir de nouvelles.

Mon stage s'est déroulé dans une entreprise nommée Benz-Info Solutions, située à Montréal au Canada. Cette entreprise est spécialisée dans le développement d'applications web destinées à faire de la gestion documentaire.

Le but du stage sera de développer des nouvelles fonctionnalités et de maintenir des fonctionnalités sur une application web existante qui est en partie utilisée par un client. Cette application permet à une entreprise de gérer des événements ainsi que des documents.

Dans un premier temps, nous présenterons l'entreprise, les différents outils utilisés et l'application sur laquelle nous travaillerons. Nous présenterons ensuite le travail effectué, nous décrirons deux tâches particulières. Nous terminerons par un bilan du travail réalisé ainsi qu'un bilan professionnel et personnel.

1 Contexte et objectifs du stage

1.1 Présentation de Benz-Info Solutions

L'entreprise Benz-Info Solutions est une petite entreprise située à Montréal (Canada) et créée en 2004 par LAPOINTE Benoit qui en est toujours le gérant. Elle compte un employé.

L'entreprise est spécialisée dans la conception d'applications web pour la gestion documentaire.

Elle propose des solutions pour d'autres entreprises qui peuvent compter plusieurs filiales et plusieurs sites localisés dans le monde entier. De plus, les solutions proposées doivent être multilingues.

Les solutions proposées permettent :

- la gestion documentaire des différents documents qu'une entreprise peut être amenée à utiliser ;
- la gestion de la conformité réglementaire : gestion des obligations en matière de santé, sécurité, environnement et qualité d'une entreprise ;
- la génération et gestion des fiches techniques de produits fabriqués ou utilisés par une entreprise ;
- la gestion du temps améliorée en automatisant le processus de répartition des heures travaillées par catégorie en fonction des départements, des conventions collectives et des différentes tâches des employés.

Le client principal de l'entreprise est *Air Liquide Group*, c'est une multinationale qui possède plusieurs secteurs d'activités dans plusieurs pays du monde.

C'est d'ailleurs sur l'application développée pour *Air Liquide Group* que porte le sujet de mon stage.

Principaux projets Les applications web développées sont destinées à faciliter la gestion FDS (Fiche de Données de Sécurité) et HSEQ (Health Safety Environment and Quality).

La gestion FDS permet de gérer des fiches de sécurité sur les produits chimiques utilisés par des entreprises.

La gestion HSEQ permet de gérer tout types d'événements qu'il peut arriver dans la vie d'une entreprise (notamment autour de la santé, la sécurité, l'environnement et la qualité).

1.2 Présentation de la mission : développement de fonctionnalités dans l'application web de gestion HSEQ

Le stage consistera à maintenir et ajouter des fonctionnalités dans l'application de gestion HSEQ pour *Air Liquide Group*. Nous aurons des missions plus ou moins longues à réaliser sur cette application. Ces missions seront des demandes du client.

Nous ne décrirons pas toutes les missions réalisées, mais nous en présenterons quelques unes qui ont nécessité de la réflexion et un temps de travail plus long.

La première mission que nous présenterons est l'ajout d'une fonctionnalité dans le module *Laws and Regulations*. Nous devons gérer le versionnage de réglementations. Pour chaque version d'une réglementation, nous devons créer des formulaires d'évaluation et d'actions que l'entreprise doit réaliser avant de pouvoir publier la version actuelle de la réglementation.

La deuxième mission que nous présenterons est de réaliser quelques modifications dans le module *Training Management*. Nous devons pouvoir dispenser certains cours pour des employés, et supprimer des formations qui peuvent être vides.

1.3 Présentation d'*InteleX* et des méthodes de travail

InteleX Pour développer l'application de gestion HSEQ, Benz-Info Solutions a choisi d'utiliser un outil nommé *InteleX* [1]. Cet outil est un CMS (Content Management System - système de gestion de contenu) qui permet de concevoir et de mettre à jour dynamiquement une application web.

InteleX repose sur la programmation orientée objet. Le principe est de créer et de lier des objets. Chaque objet possède :

- des champs ;
- un ou plusieurs gestionnaires d'action, appelés « *Action Handler* », ils peuvent être comparés aux fonctions d'une classe ;
- un flux de travail, appelé « *Workflow* », qui permet de gérer les différents états que peut avoir un objet ;

- des vues qui permettent d’afficher l’objet à l’écran.

Chaque objet est affiché sous la forme d’un formulaire au moyen de ses vues. Un objet peut avoir plusieurs vues différentes suivant son état. Par exemple, à sa création, l’utilisateur peut remplir les différents champs, alors qu’à sa fermeture, les champs doivent être bloqués.

Il est possible de modifier ces vues en utilisant le langage XML. Il est également possible d’ajouter des scripts JavaScript dans une balise XML de la vue.

Les *Action Handler* permettent de faire des calculs, de donner ou modifier des valeurs aux champs de son propre objet ou de tout les objets liés. Ils peuvent être exécutés à différents moments de la vie d’un objet (à la création, sur un changement d’état ou sur une suppression par exemple).

Un *Workflow* permet de modifier l’état d’un objet au cours de son existence. Il contient différents états (« Stage »), et on peut ajouter toute sorte de condition pour passer d’un stage à un autre. Il est aussi possible de faire des calculs et de modifier des valeurs des champs des objets.

Les *Action Handler* et les *Workflow* se développent avec une interface graphique qui propose un langage algorithmique très simple à base de conditions, d’affectation de champs et l’utilisation de quelques fonctions tableurs.

Pour réaliser le développement et les tests sans interférer sur le fonctionnement des modules existants et utilisés par le client, *Intelex* propose trois environnements (cela peut être perçu comme trois sites web) pour une même application :

- développement : c’est dans cet environnement que nous développons les fonctionnalités ou les modifications à réaliser. Nous sommes les seuls à y avoir accès ;
- test : cet environnement permet aux personnes responsables chez le client de faire tous les tests nécessaires et de nous annoncer si il y a des ajustements à faire ;
- production : c’est dans cet environnement que l’application est utilisée.

Lorsque le développement d’un module est terminé, on va créer un *package* qui sera ensuite installé dans l’environnement test. Une fois qu’il sera validé par le client, on va pouvoir l’installer dans l’environnement de production.

Méthodes de travail Benz-Info Solutions fonctionne sur une méthode agile. En effet, le développement des applications se fait par petites livraisons rapides et incrémentales qui suivent les demandes des clients qui arrivent pour la plupart par email et par téléphone.

Il y a aussi une phase d'analyse en développant des diagrammes de classe et d'activité sur *DrawIo*.

Dès qu'une tâche est terminée, elle est ajoutée sur un tableur partagé entre les employés.

Durant le stage, différentes missions nous ont été confiées. Ces missions étaient pour la plupart de la maintenance, et d'autres des ajouts de fonctionnalités. Nous décidons de présenter par la suite deux tâches.

2 Gestion de versions de réglementations dans le module *Laws and Regulations*

2.1 Présentation du module *Laws and Regulations* et des attentes

La mission la plus imposante était d'ajouter une fonctionnalité dans le module *Laws and Regulations*. Ce module permet à *Air Liquide Group* de gérer les différentes lois et réglementations suivant la location et le secteur d'activité.

Actuellement, ce module permet de créer et de compléter des réglementations, qui contiennent différentes informations.

Nôtre tâche sera de gérer le versionnage des réglementations. Pour chaque version d'une réglementation, le but sera de créer automatiquement un ou plusieurs formulaires d'évaluation. Ces formulaires d'évaluations seront créés en fonctions d'une combinaison de valeur.

Un formulaire d'évaluation aura plusieurs états.

Dans chacun de ces formulaires, il sera possible d'ajouter des actions que l'entreprise devra réaliser.

Tant que toutes les actions ne sont pas complétées, l'état du formulaire d'évaluation ne pourra pas être complété.

Lorsque le formulaire d'évaluation est complété, il sera possible de créer une nouvelle révision pour la réglementation.

2.2 Phase d'analyse du module *Laws and Regulations*

Nous avons commencé le travail par une phase d'analyse.

La première tâche, était de réaliser le diagramme de classe des objets existant et des futurs objets pour le module *Laws and Regulations*, qui est représenté par la figure 1.

Nous choisissons de ne représenter que les classes importantes, celle que nous avons créées et modifiées et seulement les principaux champs qui permettent de comprendre le fonctionnement du module.

Les objets *Law or Regulation Revision*, *Regulation Global Action Plan* et *Global Action Plan Abstract* existaient déjà dans la version précédente du

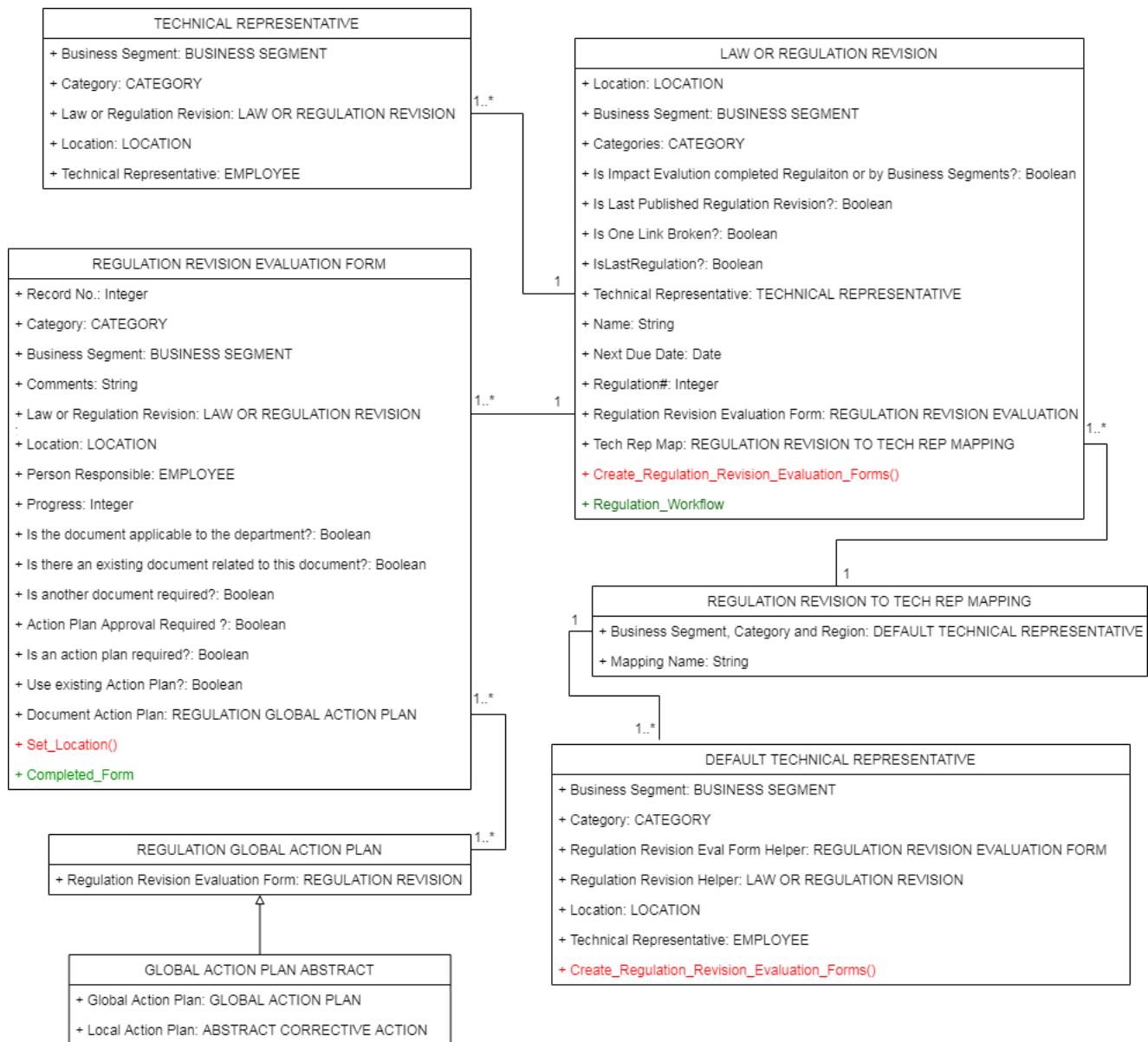


FIGURE 1 – Diagramme de classe de *Laws and Regulations*

module.

L'objet *Law or Regulation Revision* permet de créer des régulations. L'objet *Regulation Revision Evaluation Form* permettra de créer les formulaires d'évaluation. Il sera lié à l'objet *Law or Regulation Revision*. Pour créer et lier des actions à notre formulaire d'évaluation, nous utiliserons les objets *Regulation Global Action Plan* et *Global Action Plan Abstract*. Ces objets sont utilisés dans d'autres modules. Pour créer automatiquement les formulaires d'évaluation en fonction d'une combinaison de valeurs et de les lier à une personne, nous aurons besoin d'un objet *Default Technical Representative* qui aura comme champ la combinaison de valeur attendue.

Nous avons modélisé les *Action Handler* en rouge et les *Workflow* en vert.

Pour créer les formulaires, nous ferons appel aux deux *Action Handler* nommés *Create Regulation Revision Evaluation Form()*. Nous modifierons le *Workflow* de *Law or Regulation Revision* (*Regulation Workflow*) et nous en créerons un pour *Regulation Revision Evaluation Form* qui sera *Completed Form*.

Nous avons réalisé le diagramme d'activité du *Workflow* de l'objet *Law or Regulation Revision* ainsi que celui pour le *Workflow* du futur objet *Regulation Revision Evaluation Form* que nous choisissons de représenter à la figure 2.

- Cet objet contient 5 états qui sont représentés par des rectangles verts :
- *Under Evaluation* : c'est l'état de création de l'objet ;
 - *Action Plan Creation* : permet à l'utilisateur de créer des actions globales ou locales ;
 - *Action Plan Approval* : lors de la création de la régulation, la personne responsable a décidé si il y avait besoin d'une approbation. Si oui, alors c'est à ce stage que cette personne pourra approuver ou non les actions créées, sinon ce stage est passé. Si les actions ne sont pas approuvées, l'objet retourne au stage précédent ;
 - *Action Plan Execution* : c'est pendant cette étape que l'entreprise réalise et complète les différentes actions ;
 - *Completed*.

Les rectangles jaunes représentent l'action que l'utilisateur doit effectuer

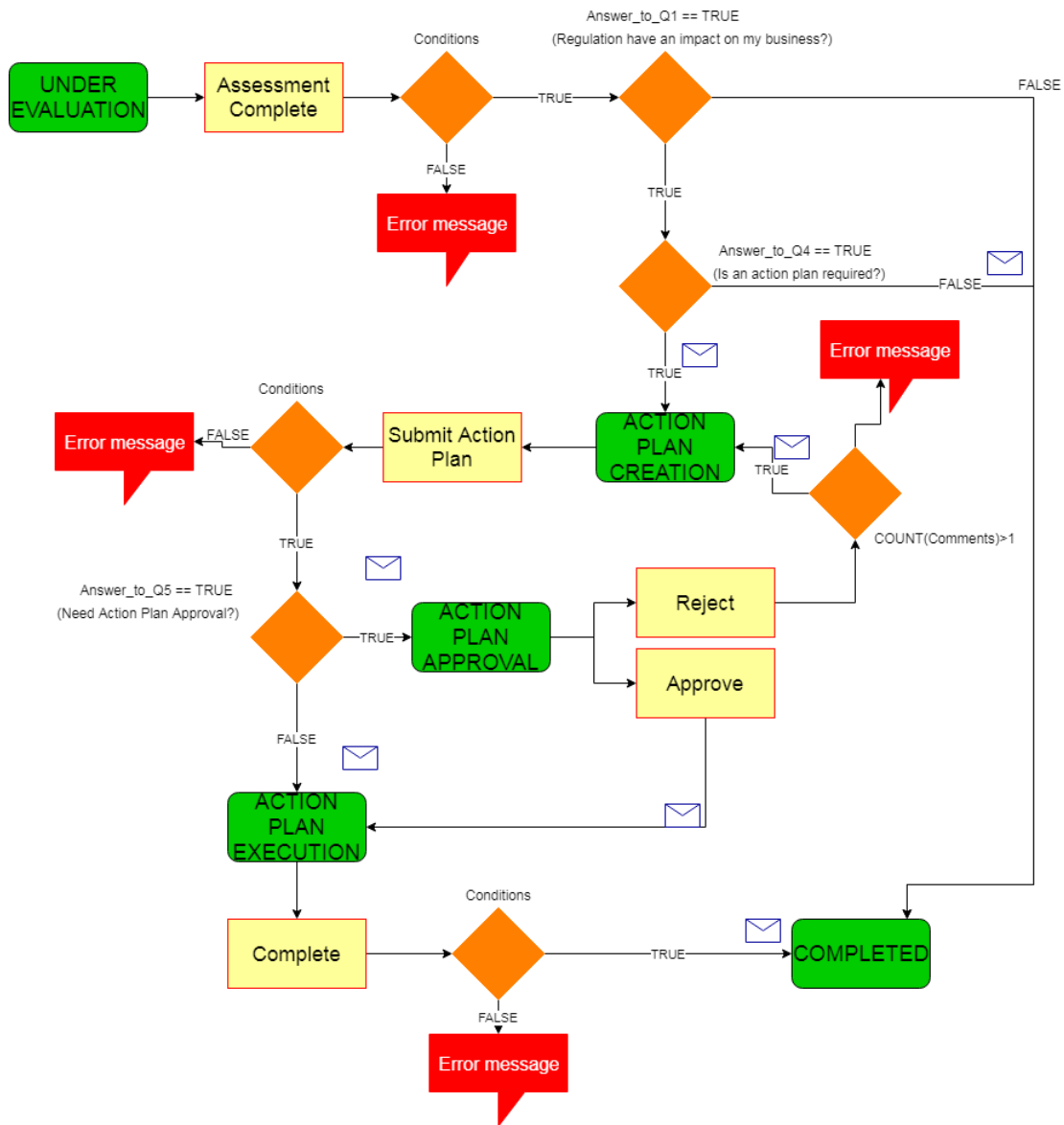


FIGURE 2 – Diagramme d'activité représentant le *Workflow* de *Regulation Revision Evaluation Form*

pour passer au stage suivant et les losanges oranges montrent les différentes conditions. Dans cet objet, il y a beaucoup de conditions, c'est pour cela que

nous décidons d'en montrer quelques unes.

Après la création automatique des formulaires, la personne responsable devra répondre à des questions avant de passer à l'état suivant. Les réponses à ces questions sont importantes car suivant les réponses, il est possible de ne pas passer par certains stages. Nous avons représenté les réponses à ces questions dans certaines conditions.

Lorsqu'une condition n'est pas remplie, un message d'erreur est affiché et l'objet reste au même état. L'enveloppe montre les différents envois d'email à la personne responsable.

2.3 Développement de la fonctionnalité

Une fois la phase d'analyse terminée, nous avons pu commencer le développement.

Création des objets et des champs Nous avons commencé par créer l'objet *Regulation Revision Evaluation Form* ainsi que la création de son *Workflow*, nommé *Completed Form*, qui gère ses différents états.

Ensuite, il faut le lier à l'objet *Law or Regulation Revision* en lui ajoutant un champ avec un lien 1-N car il peut avoir plusieurs formulaires d'évaluation pour une révision. Nous avons modifié le *Workflow* de cet objet pour lui ajouter un état intermédiaire d'évaluation. C'est dans cet état que les utilisateurs pourront compléter les formulaires d'évaluation. Il ne sera pas possible de compléter la régulation tant que tous les formulaires d'évaluation ne sont pas complétés.

Nous avons aussi ajouté un champ question de type booléen « *Is impact evaluation completed by regulation or by business segment ?* » avec le choix entre deux valeurs. L'utilisateur répond à cette question lors de la création d'une révision. La réponse à cette question aura un impact sur la création des formulaires d'évaluation. Si la réponse est « *Business Segment* », les formulaires d'évaluations seront créés en fonction de la combinaison de valeur *Business Segment*, *Category*, *Location* et *Employee*. Si la réponse est « *Regulation* », alors on va créer un seul formulaire d'évaluation qui ne prend en compte seulement la personne responsable de la régulation.

Les formulaires d'évaluation doivent être créés automatiquement et liés aux personnes assignées qui devront les compléter. Pour pouvoir créer ces formulaires en fonction des combinaisons de valeur, il fallait créer l'objet *Default Technical Representative*, qui contiendra un champ pour ces 4 valeurs. La personne responsable du module des *Laws and Regulations* pour une location pourra instancier cet objet et créer des enregistrements avec des combinaisons de valeurs différentes.

Création d'*Action Handler* Pour envoyer les formulaires créés aux personnes assignées, il faudra utiliser des *Action Handler* et il nous faudra créer des champs qui nous aideront à garder l'*Id* de certains objets, le nom de ces champs contiendra le mot « Helper ». Le champ *Regulation Revision Helper* de l'objet *Default Technical Representative* permettra de stocker l'*Id* de la régulation.

Nous allons découper la création de ces formulaires en plusieurs étapes. La première étape se déroule dans le *Action Handler* de *Law or Regulation Revision*. Cet *Action Handler* va tout d'abord vérifier la réponse à la question « *Is impact evaluation completed by regulation or by business segment?* ». Si la réponse est « *Regulation* », on crée le formulaire avec une fonction *GETRELATED*. Cette fonction permet de créer un enregistrement que l'on peut ajouter à un objet. Son premier argument est le nom de l'objet suivi par au moins un couple de valeur, qui sont le nom du champ dans l'objet et la valeur à lui attribuer. Si la réponse est « *Business Segment* », on vérifie d'abord qu'il y ait au moins un *Business Segment* et une *Category* attachée à la régulation. On va pouvoir envoyer l'*Id* de la révision de la régulation au champ *Regulation Revision Helper* de *Default Technical Representative* et continuer les traitements dans cet objet.

La figure 3 est une capture d'écran de cet *Action Handler*.

Dans l'objet *Default Technical Representative*, il y a un autre *Action Handler* qui s'exécute sur la mise à jour de l'objet, lorsque la valeur du champ *Regulation Revision Helper* est changée. On va utiliser une fonction *GETDATA* qui va permettre de trouver les différentes combinaisons de valeurs qui correspondent avec les *Business Segment* et *Category* attachées. On va donc pouvoir créer les formulaires d'évaluation avec la fonction *GETRELATED*. On va remplir les différents champs de l'objet *Regulation Revision Evaluation*

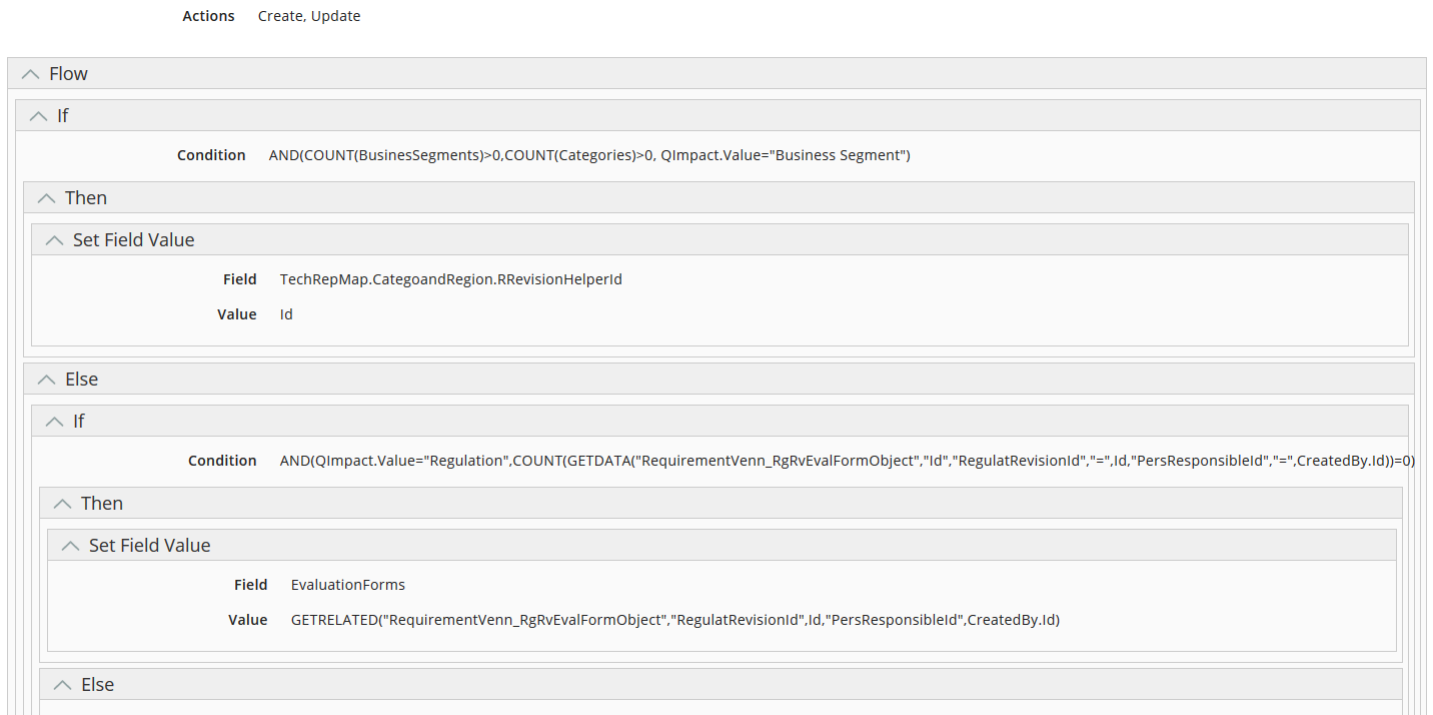


FIGURE 3 – *Action Handler de Law or Regulation Revision*

Form dont le champ *Law and Regulation Revision* avec l'*Id* que nous avons stocké dans le champ *Regulation Revision Helper* de l'objet *Default Technical Representative*. Cela va permettre aux formulaires d'évaluation d'être liés à la régulation. L'algorithme 1 représente cet *Action Handler* de façon simplifiée.


```

if Regulation Revision Helper <> NULL then
  if Au moins une combinaison de valeur existe then
    REvaluationForm =
      GETRELATED("RequirementVenn_RgRvEvalFormObject",
        "LawAndRegulationRevisionId", RegulationRevisionHelperId,
        "BusinessSegmentId", BusinessSegmentId,
        "PersResponsibleId", TechRepresId,
        "CategoryId", CategoryId);
  else
  end
else
end
RRevisionHelper = NULL;

```

Algorithm 1: *Action Handler de Default Technical Representative*

Modification des vues Nous avons aussi modifié la vue de l'objet *Regulation Revision Evaluation Form*. La modification des vues se fait en langage XML. Malgré qu'il y ait plusieurs états pour cet objet, il y a une seule vue. Il est donc possible d'afficher l'objet en fonction de son état grâce à une balise. La figure 4 montre une portion de code source de la vue de l'objet *Regulation Revision Evaluation Form*.

On peut voir les différentes balises utilisées. La balise **StageContainer** permet d'afficher son contenu seulement lorsque l'état de l'objet correspond à la valeur de l'attribut **For**. La balise **DetailSection** permet d'afficher une section qui contiendra les champs que l'utilisateur pourra remplir, chaque champ devra être mis dans une balise **Field** qui contient un attribut **For** pour spécifier le nom du champ. Il est possible d'ajouter d'autres attributs qui permettent de spécifier si le champ est obligatoire ou non par exemple. Les **GridSection** permettent d'afficher une grille d'objets qui sont liés à l'objet actuel. La balise **Menu** contient des balises **GridActionLink** qui seront des boutons d'actions. Dans notre exemple, ce seront les boutons pour ajouter ou supprimer un enregistrement. Les différents champs de l'objet sont affichés dans les colonnes de la grille.

```

1 <ix:StageContainer UseDefaultDuringCreate="true" For="Draft" ID="StageContainer_Create">
2   <ix:DetailSection Title="Details">
3     <ix:FieldArea>
4       <ix:Field For="Title" IsMandatory="true" Span="true"/>
5       <ix:Field For="BusinessSegment" IsMandatory="true" Span="true" editas="entityselector"/>
6       <ix:Field For="Category" IsMandatory="true" Span="true"/>
7     </ix:FieldArea>
8   </ix:DetailSection>
9   <ix:GridSection For="DocumActionPlan" Title="Global Action Plans">
10     <Menu>
11       <ix:GridDeleteActionLink Override="true" Visible="true" runat="server"/>
12       <ix:GridCreateActionLink Override="true" Visible="true" runat="server" OpenIn="popup"/>
13     </Menu>
14     <Columns>
15       <ix:Column DisplayAs="String" For="ActionNo" LinkTo="Default"/>
16       <ix:Column DisplayAs="String" For="CPARActionReq"/>
17     </Columns>
18   </ix:GridSection>
19 </ix:StageContainer>

```

FIGURE 4 – Code source simplifié d'une vue

Le dernier travail était de créer des templates pour l'envoi d'email à la personne responsable à chaque changement d'état de l'objet *Regulation Revision Evaluation Form*.

Nous avons ensuite procédé à des tests, en créant des régulations différentes, puis en passant toutes les étapes une à une. Lors de ces tests, nous vérifions que les objets soient bien créés, que tout se déroule comme on le souhaite, et nous portons aussi une attention sur les vues pour vérifier si elles sont correctes.

La figure 5 est une capture d'écran de la page lorsque le formulaire se trouve à l'état *Action Plan Execution*.

Des rectangles ont été placés pour cacher des informations qui doivent rester confidentielles.

On peut voir l'affichage des différents états de l'objet en haut de la page. Les sections *Details* et *Applicability Questions* sont générées avec les balises *DetailSection* tandis que la section *Global Action Plan* est générée avec la balise *GridSection*.

Sur l'évaluation qui est faite ici, on voit qu'il y a une *Global Action* qui a été

attachée et qui est complétée à 50%, il n'est donc pas possible de passer au stage suivant.

Complete

Add Comment

Edit

Exit

⚙️

🖨️

📄

Regulation Revision Evaluation Form

Workflow Stage: Action Plan Execution

Workflow Status: Action Plan Execution

Person Responsible:

Not Started

Under Evaluation

Action Plan Creation

Action Plan Approval

Action Plan Execution

Completed

Details

Related Regulation Revision - This is a Test

Type

State / Provincial

State / Province

Quebec

Business Segment

Category

Person Responsible

Applicability Questions

Do regulation have an impact on my business ?

Yes

Is there an existing document related to this document?

No

Is another document required?

No

Is an action plan required?

Yes

Use existing Action Plan?

False

Document Action Plan

Global Action Plan: 12049

Action Plan Approval Required ?

No

Action Plan Owner

Global Action Plans

Action No.	Description of Action	Assigned To	Due Date	Progress
<div></div>	<div></div>	<div></div>	<div></div>	<div></div>
<div></div> 12049	Global Action	<div></div>	Thursday, May 23, 2019	<div>50.00</div>

Viewing 1 - 1 of 1 Records

Items Displayed 20

?

FIGURE 5 – Capture d’écran de la page pour le formulaire d’évaluation

3 Modifications à apporter dans le module *Training Management*

3.1 Présentation du module *Training Management*

Nous allons présenter une deuxième mission. Cette mission va consister à faire quelques modifications dans le module *Training Management*.

Ce module permet de répertorier les formations (obligatoires ou non) que les employés ont à suivre durant leur carrière. La personne responsable des formations va créer des tâches, « *Job tasks* », et leur assigner des employés et des cours.

Chaque employé doit compléter des qualifications, il y en a une par tâche. Chaque qualification doit être complétée avant une date d'expiration de maximum 3 ans. Les employés doivent suivre les cours qui leurs sont attribués. Les résultats sont entrés par la personne responsable de l'employé.

Nous décidons de montrer le diagramme de classe simplifié du module *Training Management* pour améliorer la compréhension, la figure 6 le représente.

L'objet *Job Task*, permet de créer des tâches et d'y assigner des employés en utilisant l'objet *Workgroup*. Chaque employé a des qualifications à remplir. Une qualification, représentée par l'objet *AL - Qualification Status* est liée à l'objet *Workgroup*.

Pour compléter une qualification, l'employé devra suivre des cours représentés par l'objet *Course*.

Le résultat d'un cours est entré par la personne responsable de l'employé en utilisant l'objet *Manual Training Record*.

Nous avons représenté le *Workflow* de *Manual Training Record* en vert, et les *Action Handler* que nous modifierons en rouge.

Nous aurons 3 modifications à apporter dans ce module :

- la possibilité de dispenser des cours à des employés ;
- modifier la date d'expiration d'une qualification ;
- supprimer des qualifications vides.

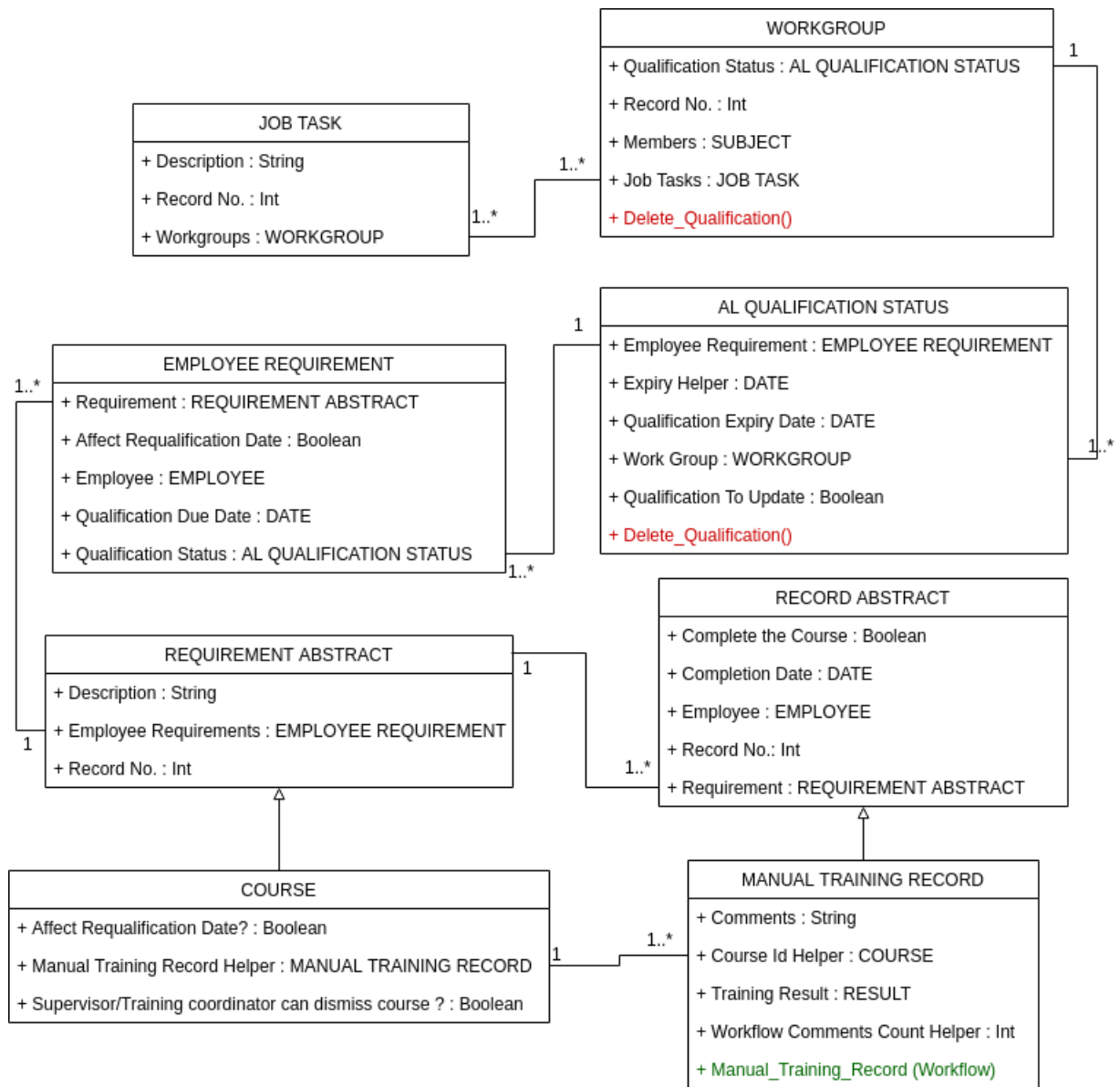


FIGURE 6 – Diagramme de classe *Training Management*

3.2 Dispenser des cours

Attentes La première modification est de permettre à une personne responsable de dispenser un cours à un employé. Tous les cours ne pourront pas être dispensés, c’est pour cela qu’il faudra également ajouter un paramètre à la création du cours pour que la personne responsable des formations puisse dire si le cours peut être dispensé.

Développement La première chose à faire était de s’approprier et comprendre le fonctionnement du module, ainsi que d’assimiler le diagramme de classe.

Ce changement va se répercuter sur deux objets : *Course* et *Manual Training Record*. L’objet *Manual Training Record* permet d’ajouter le résultat d’un cours pour chaque employé.

Un champ a été ajouté dans l’objet *Course* : *Supervisor/Training coordinator can dissmis course ?*. Ce champ est de type booléen, l’utilisateur va répondre à cette question lors de la création du cours. C’est au moment de remplir les résultats que la personne responsable va pouvoir annuler le cours pour l’employé si la réponse à cette question est vraie. Nous avons donc ajouté un champ *Course Id Helper* dans *Manual Training Record* qui va permettre de stocker l’*Id* du cours associé. Nous avons ensuite modifié le *Workflow* pour ajouter un bouton qui va permettre d’annuler le cours. Lors de l’annulation d’un cours, il doit être obligatoire d’ajouter un commentaire, c’est pourquoi nous avons ajouté le champ *Comments* dans *Manual Training Record*.

La figure 7 montre le diagramme d’activité du workflow de *Manual Training Record*.

Les rectangles verts représentent les différents états de l’objet, les rectangles jaunes représentent les différentes actions que l’utilisateur doit effectuer pour passer à l’état suivant.

On remarque que l’état COMPLETED n’est pas tout à fait le même suivant l’action qui a été réalisée. En fait, l’état de l’objet est le même, mais c’est le status du *Workflow* qui est différent. Cela permet à la personne responsable de voir plus facilement si un cours a été complété ou dispensé.

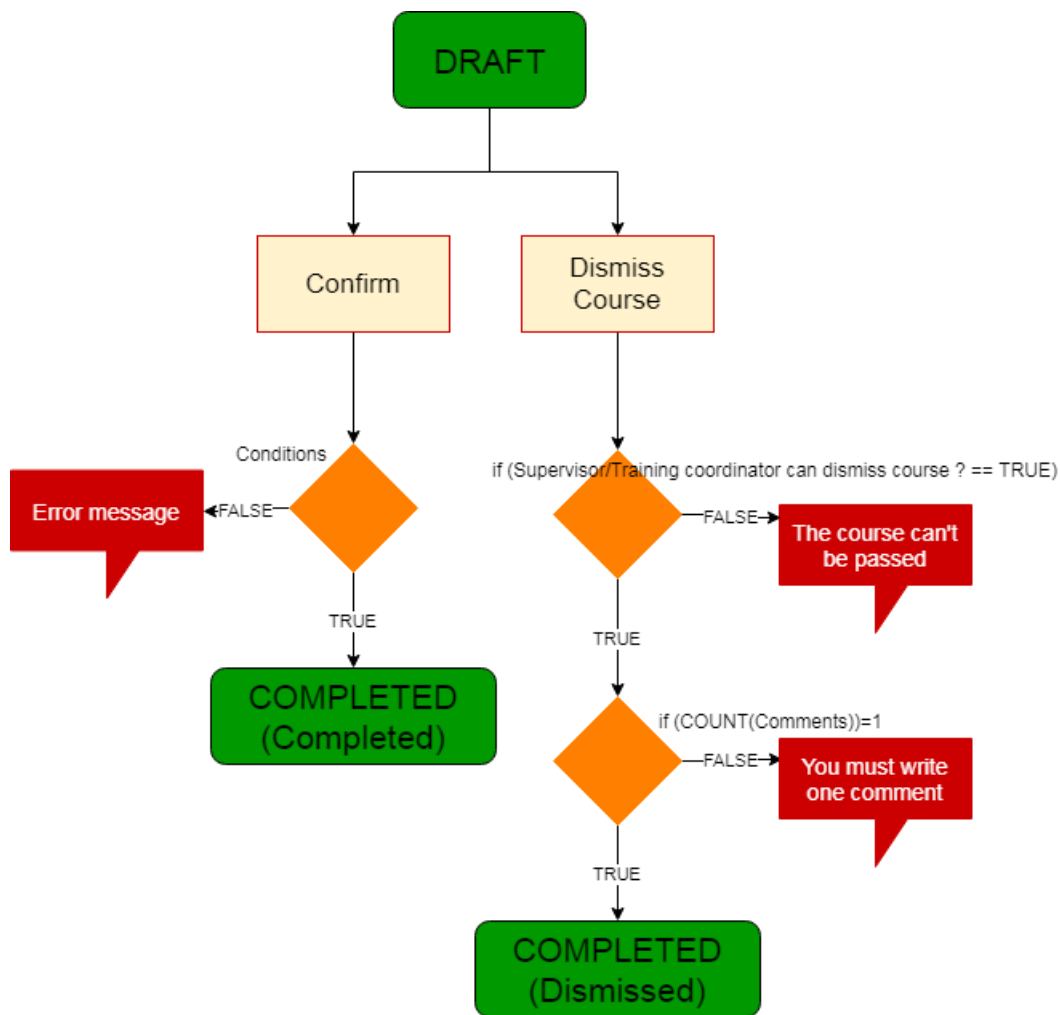


FIGURE 7 – *Workflow de Manual Training Record*

3.3 Modifier la date d’expiration d’une qualification

Attentes La deuxième chose à faire était de pouvoir modifier la durée avant qu’une qualification expire. Cette durée devra être remplie par la personne responsable des formations.

Développement Il existe un objet *Settings* pour ce module qui permet de modifier certains paramètres. Nous avons ajouté un champ dans cet objet

pour permettre de remplir le nombre de jour avant l'expiration de la qualification.

Le champ question *Affect Requalification Date ?* à été ajouté dans l'objet *Course* et permettra à la personne responsable de spécifier si le cours affecte la date d'expiration ou non.

Dans l'objet *AL - Qualification Status*, il existe déjà des champs qui permettent de calculer la date d'expiration de la qualification, notamment *Expiry Helper* et *Qualification Expiry Date*. La date d'expiration peut être de maximum 3 ans. On veut pouvoir modifier ce maximum par le nombre de jour qui serait entré dans les paramètres du module.

Le calcul du champ *Expiry Helper* à donc été changé. L'objet *Settings* n'est pas lié aux objets du module, pour récupérer la valeur du champ, nous allons donc utiliser la fonction **GETSETTING**, qui compte deux arguments : le nom de l'objet et le nom du champ.

L'algorithme 2 montre le calcul du champ *ExpiryHelper*.

```
if ApprovalDate=NULL then
| NULL
else
| DATEADD("d",
| GETSETTING("TrainMgmt_SettingsObject", "qualiexpiration")
| , ApprovalDate))
end
```

Algorithm 2: Calcul du champ *Expiry Helper*

L'algorithme 3 montre le calcul du champ *Qualification Expiry Date* qui utilise le champ *Expiry Helper*.

```
if NextCourseDate < ExpiryHelper then
| NextCourseDate
else
| ExpiryHelper
end
```

Algorithm 3: Calcul du champ *Qualification Expiry Date*

3.4 Suppression des qualifications vides

Attentes La dernière modification à apporter est de supprimer des qualifications qui ne contiendraient aucun cours pour un employé.

Lors de la suppression d'un employé d'un *Job Task*, les cours qui étaient associés à ce *Job Task* vont être détachés de la qualification de l'employé. Si cette qualification ne contient plus aucun cours, alors il faut la supprimer.

Développement Lors de la suppression d'un employé du *Job Task*, c'est l'objet *Work Group* qui se met à jour, c'est donc sur celui-ci que nous allons créer un *Action Handler*.

Nous avons ensuite créé un champ *Qualification To Update* dans *AL - Qualification Status* de type booléen. Le *Action Handler* de *Work Group* va envoyer la valeur **VRAI** dans ce champ lorsqu'il y a une suppression d'un employé. Dans l'objet *AL - Qualification Status*, lorsque la valeur de ce champ est égale à **VRAI**, c'est un autre *Action Handler* qui va pouvoir s'exécuter. La figure 8 montre cet *Action Handler*. Il va vérifier que l'employé détaché du *Job Task* est le même que dans la qualification actuelle, et mettre la valeur du champ *Deleted* de *AL - Qualification Status* à **VRAI**, ce qui marquera l'objet supprimé.

Nous ne voulons pas supprimer complètement la qualification du système, mais l'archiver car nous voulons en garder une trace. Il existe un paramètre *Soft Delete* dans chaque objet qui permet d'archiver l'objet lors de la suppression.

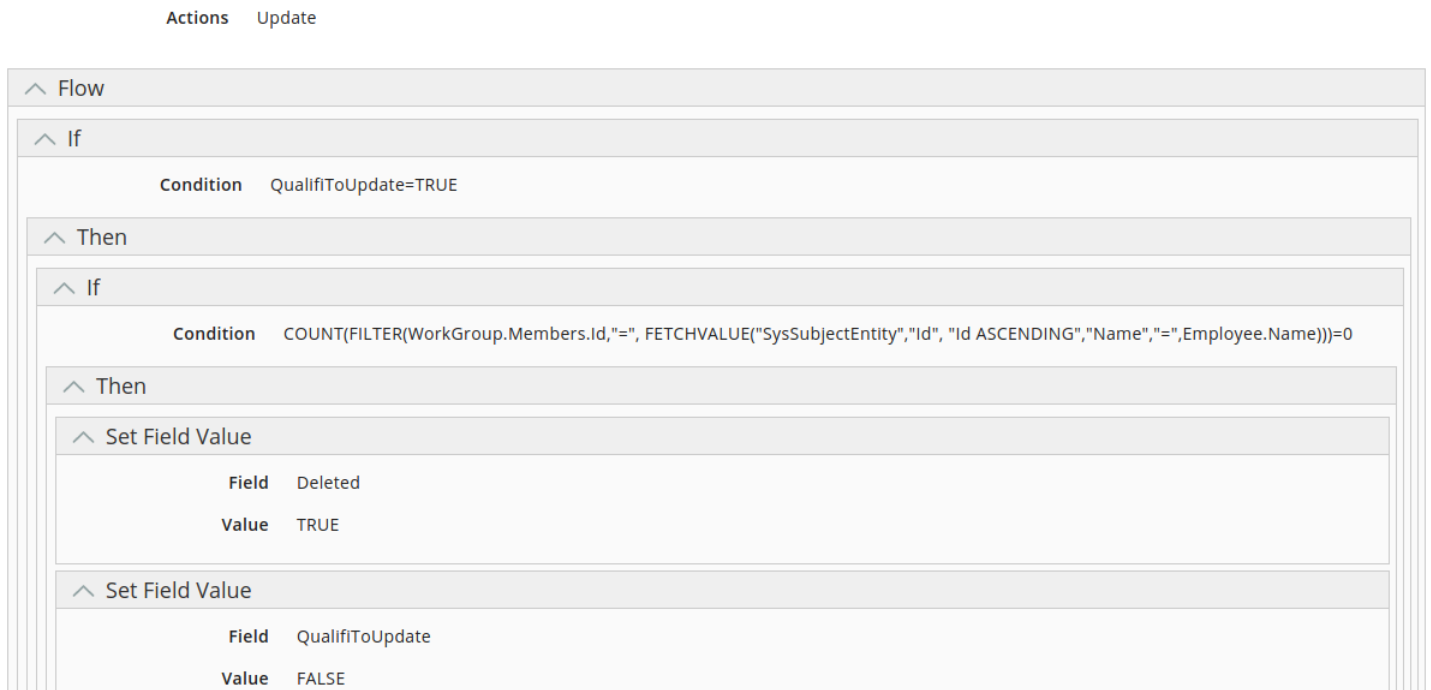


FIGURE 8 – *Action Handler* de *AL - Qualification Status*

4 Présentation succincte des autres tâches

Durant ce stage, nous avons réalisé de nombreuses missions différentes que nous ne développerons pas dans ce rapport. Les missions étaient variées. Nous avons apporté des modifications dans certains objets, ajouter des champs, modifier le calcul de certains champs et d'*Action Handler*. Nous avons également beaucoup travaillé sur le code source des vues. Nous avons aussi fait de l'importation de données (par exemple, de nouveaux pays, de nouveaux employés). Ces informations étaient données par le client dans des feuilles de calcul, type *Excel*. *Intelex* propose un outil d'importation de feuille de calcul.

5 Bilan

5.1 Bilan des missions réalisées

Le but du stage était de réaliser plusieurs tâches de développement et de maintenance de fonctionnalités sur une application qui est en partie utilisée. Nous présentons ici le résultat de ces missions.

Laws and Regulations La première tâche était d'ajouter une fonctionnalité dans le module des *Laws and Regulations*. Il fallait gérer le versionnage de réglementations et de créer des formulaires d'évaluation pour chacune de ces versions.

Le développement de cette fonctionnalité à été fait avec quelques difficultés. En effet, il fallait comprendre le fonctionnement d'*Intalex*, de s'approprier les différentes fonctions utilisées dans les Action Handler et les Workflows ainsi que les différentes balises XML utilisées dans le code source des vues.

La fonctionnalité demandée à été développée entièrement puis nous avons effectué des tests. Une fois les tests réalisés, nous avons installé la fonctionnalité dans l'environnement de test de l'application. Cela permet aux personnes responsables de *Air Liquide Group* de faire des tests, ainsi que de vérifier les détails du module. Cela leur permet de nous annoncer des changements à faire, si il y en a, avant de l'utiliser. À ce jour, le module est toujours dans l'environnement de test, et n'est donc pas encore utilisé.

Training Management La deuxième tâche présentée était d'apporter des modifications dans le module des *Training Management*. Les modifications était de pouvoir dispenser des cours pour des employés, de modifier la date d'expiration d'une qualification et de supprimer les qualifications vides.

Le développement de cette tâche est terminé. La difficulté la plus élevée rencontrée était de s'approprier le fonctionnement du module qui existait déjà et de comprendre les relations entre les différents objets.

Une fois les modifications réalisées et quelques tests effectués, le module à été installé dans l'environnement de test. Après quelques jours, le client a validé le module et nous l'avons donc installé dans l'environnement de production. Ce module est donc actuellement utilisé.

5.2 Bilans professionnels et personnels

Sur le plan professionnel, le stage m'a permis d'acquérir de nouvelles connaissances ainsi que d'utiliser des compétences et des connaissances acquises lors de mes études à l'université. En effet, j'ai pu apprendre à utiliser un nouvel outil que je ne connaissait pas pour développer des applications web. Il fallait que je m'adapte aux conventions de l'outil et de l'entreprise. J'ai pu mettre en pratique mes connaissances en programmation orientée objet et aussi mes compétences en langage XML et en UML pour développer les différents diagrammes.

D'un point de vue personnel, je pense que ce stage m'a apporté une meilleure organisation, et une meilleure adaptation pour l'utilisation de nouveaux outils. De plus, le fait d'être dans un autre pays m'a permis de découvrir une nouvelle culture, de rencontrer de nouvelles personnes et d'être indépendant.

Conclusion

Ce stage m'a permis de découvrir un outil pour générer des sites web de gestion d'évènements. J'ai pu apprendre à l'utiliser. Ce stage m'a également permis de réaliser différentes missions sur l'application, certaines étaient l'ajout de fonctionnalités et d'autres étaient de la maintenance. Ces fonctionnalités ont été développées entièrement. J'ai aussi vu l'organisation d'une petite entreprise de développement web.

Références

- [1] Intelex. Environmental, Health & Safety, and Quality (EHSQ) Management Software Solutions, <https://www.intelex.com/>.
- [2] Benz-Info Solutions. Solutions pour organiser, structurer et contrôler vos documents, <http://benz-info.com/>.

Résumé

Ce rapport est un écrit qui présente le stage en entreprise effectué dans le cadre de mes études en Licence 3 Informatique. Ce stage se déroule dans l'entreprise Benz-Info Solutions située à Montréal. Je présente dans ce rapport l'entreprise, les principaux projets et outils utilisés par celle-ci. Je montre dans ce rapport la réalisations de différentes missions qui m'ont été confiées. Ces missions était d'ajouter des fonctionnalités dans une application web de gestion d'évènements. Je termine ce rapport par un bilan professionnel et personnel de ce que m'a apporté le stage.

Mots-clés

Stage, application web, Intalex, gestion d'évènements, gestion documentaire, objets

Abstract

This report is a written which represent the internship in company as part of my studies in IT bachelor's degree. This internship takes place in the company Benz-Info Solutions located in Montreal. I present in this report the company, the main projects and tools used by it. I show in this report the achievements of different missions that have been entrusted to me. These missions were to add features in an event management web application. I finish this report with a professional and personal assessment of what the internship brought me.

Keywords

Internship, web application, Intalex, event management, document management, objects