P\_GestProj2 – Module ICT 306



Daniel Baltensperger, Sam Nadir Naït Abbou, Lucas Da Silva, Yvan Cochet

ETML, Lausanne

Rue Sebeillon 12, 1004 Lausanne

10 semaines (du 14 mars au 27 mai 2016)

Chef de projet : Gilbert Gruaz

Table des matières

[1 Spécifications 3](#_Toc449960120)

[1.1 Titre 3](#_Toc449960121)

[1.2 Description 3](#_Toc449960122)

[1.3 Matériel et logiciels à disposition 3](#_Toc449960123)

[1.4 Prérequis 3](#_Toc449960124)

[1.5 Cahier des charges 4](#_Toc449960125)

[1.5.1 Fonctionnalités requises (du point de vue de l’utilisateur) 4](#_Toc449960126)

[1.5.2 Contraintes 4](#_Toc449960127)

[1.5.3 Travail à réaliser par l'apprenti 4](#_Toc449960128)

[1.5.4 Méthodes de validation des solutions 4](#_Toc449960129)

[1.6 Les points suivants seront évalués 5](#_Toc449960130)

[1.7 Validation et conditions de réussite 5](#_Toc449960131)

[2 Analyse 5](#_Toc449960132)

[2.1 Opportunités 5](#_Toc449960133)

[2.2 Document d’analyse et conception 6](#_Toc449960134)

[2.3 Modèles de Données 6](#_Toc449960135)

[2.3.1 Modèles de Conceptuel de Donnée (MCD) 6](#_Toc449960136)

[2.3.2 Modèle Logique de Données (MLD) 7](#_Toc449960137)

[2.3.3 Modèle Physique de Données (MPD) 7](#_Toc449960138)

[2.4 Conception des tests 7](#_Toc449960139)

[2.5 Planification détaillée 7](#_Toc449960140)

[3 Réalisation 7](#_Toc449960141)

[3.1 Dossier de Réalisation 7](#_Toc449960142)

[3.2 Modifications 7](#_Toc449960143)

[4 Tests 8](#_Toc449960144)

[4.1 Dossier des tests 8](#_Toc449960145)

[5 Conclusion 8](#_Toc449960146)

[5.1 Bilan des fonctionnalités demandées 8](#_Toc449960147)

[5.2 Bilan de la planification 8](#_Toc449960148)

[5.3 Bilan personnel 8](#_Toc449960149)

[6 Divers 8](#_Toc449960150)

[6.1 Journal de travail 8](#_Toc449960151)

[6.2 Bibliographie 8](#_Toc449960152)

[6.3 Webographie 8](#_Toc449960153)

[7 Annexes 8](#_Toc449960154)

# Spécifications

## Titre

Application pour la gestion d'un catalogue de formations, et des formations.

## Description

Ce projet a pour but de remplacer le système actuel de gestion de formations, réalisé à l'aide de classeurs et feuilles Excel, par une application modulaire réalisée en PHP, MySQL et JavaScript.

Il convient tout d'abord de remplir les fichiers Excel avec des valeurs fictives, qu'il s'agira ensuite de transférer dans la BD.

Le public cible est constitué de personnels de bureau, capables d'utiliser un ordinateur et des applications simples et standards. Ces personnes ne sont pas capables de corriger les incohérences de données (un formateur apparait 2 fois avec des noms différents car il y a un espace dans le nom composé dans un cas, et pas dans l'autre, par exemple).

Le dispositif permettra d'effectuer les saisies des informations, de les modifier, de les supprimer, et de les imprimer sous forme de listes, permettant ainsi de reproduire la gestion faite jusque-là avec des classeurs Excel.

Si le temps le permet, les fonctionnalités suivantes pourront encore être implémentées :

* Une fonctionnalité de statistiques permettant aux gestionnaires d'exploiter les informations provenant des enquêtes de satisfaction.
* Une fonctionnalité de récupération des données des formations pour l'année suivante
* Une fonctionnalité permettant un archivage des formations, des participants, des formateurs, etc.

## Matériel et logiciels à disposition

1 ordinateur PC avec Windows 7 standard de l'ETML

Une architecture WAMP, comme EasyPhp ou Xampp

Des classeurs Excel sans données permettant de comprendre la situation actuelle.

## Prérequis

Avoir suivi les modules I-CH.

## Cahier des charges

### Fonctionnalités requises (du point de vue de l’utilisateur)

Une institution possède un département de formations internes et externes. La gestion se faisait, jusqu'à maintenant, avec des classeurs Excel et des feuilles de calculs, qui permettaient de …

* Gérer la liste des formations (titre, description, date, durée, formateurs, prix, etc.)
* Gérer la liste des formateurs (nom, prénom, adresse, email, tél, titres obtenus, etc.)
* Gérer la liste des participants-étudiants (nom, prénom, adresse, etc.)
* Gérer les inscriptions aux formations (participants, formations, dates, etc.)
* Gérer les résultats des enquêtes de satisfaction pour chacune des formations

De nombreuses erreurs de saisies ont démontré que ce système, simple à utiliser, avait atteint ses limites, et qu'il convient maintenant de lancer un projet informatique qui va permettre de fédérer ces données dans un seul conteneur, et de les gérer avec une application dédiée.

### Contraintes

Utiliser la méthode scrum et un logiciel de visionnage de type GIT.

### Travail à réaliser par l'apprenti

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

* Une planification initiale
* Un rapport de projet
* Un journal de travail
* Un manuel de mise en œuvre des composants de l'application sur une nouvelle machine (architecture WAMP utilisée, importation de la BD dans l'architecture WAMP, etc.)
* Un fichier d'archives contenant la BD exportée, les codes développés

### Méthodes de validation des solutions

Validation par le Product Owner

## Les points suivants seront évalués

* Le rapport
* Les planifications (initiale et détaillée)
* Le journal de travail
* Le code et les commentaires
* Un manuel de mise en œuvre des composants de l'application sur une nouvelle machine (architecture WAMP utilisée, importation de la BD dans l'architecture WAMP, etc.)
* Un fichier d'archives contenant la BD exportée, les codes développés

## Validation et conditions de réussite

* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré

# Analyse

## Opportunités

Ce paragraphe énumère la liste des difficultés potentielles de tout ordre :

Liste des compétences à acquérir ou approfondir

Liste du matériel à exploiter

Recherche d’informations particulières

Gestion du travail en équipe & collaboration

Ainsi que les solutions possibles

Si les spécifications de départ ne laissent pas de doutes sur la manière de réaliser un projet, ce chapitre ne fera que renvoyer le lecteur aux spécifications.

## Document d’analyse et conception

## Modèles de Données

### Modèles de Conceptuel de Donnée (MCD)

Nous utilisons le logiciel ***AnalyseSI*** pour la création du MCD.

Mais nous nous sommes rendu compte que celui-ci était très mal optimisé pour notre utilisation. C’est pourquoi nous nous sommes par la suite dirigés vers un logiciel dit « basique » de dessin : Microsoft Visio 2013.

### Modèle Logique de Données (MLD)

### Modèle Physique de Données (MPD)

## Conception des tests

Ce paragraphe permet de spécifier la stratégie de test qui sera menée au point 4.1

Qui, quand, avec quelles données, dans quel ordre, etc.

## Planification détaillée

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

# Réalisation

## Dossier de Réalisation

Cette partie permet de reproduire ou reprendre le projet par un tiers.

Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :

Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)

Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)

Code source commenté des éléments logiciels développés.

Modèle physique d’une base de données.

Arborescences des documents produits.

Il faut décrire le parcours de réalisation et justifier les choix.

## Modifications

Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.

Date, raison, description, etc.

# Tests

## Dossier des tests

On dresse le bilan des tests effectués (qui, quand, avec quelles données…) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats obtenus et les actions à entreprendre en conséquence (et une estimation de leur durée).

Si des tests prévus dans la stratégie n'ont pas pu être effectués :

raison, décisions, etc.

Liste des bugs répertoriés avec la date de découverte et leur état:

Corrigé, date de correction, corrigé par, etc.

# Conclusion

## Bilan des fonctionnalités demandées

Il s’agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.

Si ce n’est pas le cas, estimé en « % » ou en « temps supplémentaire » le travail qu’il reste à accomplir pour terminer le tout.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différences entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu’une activité de recherches a été entreprise, il convient d’énumérer ce qui a été trouvé, avec les références.

## Bibliographie

Références des livres, revues et publications utilisés durant le projet.

## Webographie

Références des sites Internet consultés durant le projet.

# Annexes

Listing du code source (partiel ou, plus rarement complet)

Guide(s) d’utilisation et/ou guide de l’administrateur

Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).

Extraits de catalogue, documentation de fabricant, etc.