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](#_Toc449965052)

[1.1 Titre 3](#_Toc449965053)

[1.2 Description 3](#_Toc449965054)

[1.3 Matériel et logiciels à disposition 3](#_Toc449965055)

[1.4 Prérequis 3](#_Toc449965056)

[1.5 Cahier des charges 4](#_Toc449965057)

[1.5.1 Fonctionnalités requises (du point de vue de l’utilisateur) 4](#_Toc449965058)

[1.5.2 Contraintes 4](#_Toc449965059)

[1.5.3 Travail à réaliser par l'apprenti 4](#_Toc449965060)

[1.5.4 Méthodes de validation des solutions 4](#_Toc449965061)

[1.6 Les points suivants seront évalués 5](#_Toc449965062)

[1.7 Validation et conditions de réussite 5](#_Toc449965063)

[2 Analyse 5](#_Toc449965064)

[2.1 Modèles de Données 5](#_Toc449965065)

[2.1.1 Modèles de Conceptuel de Donnée (MCD) 5](#_Toc449965066)

[2.1.2 Modèle Logique de Données (MLD). 6](#_Toc449965067)

[2.1.3 Modèle Physique de Données (MPD) 6](#_Toc449965068)

[2.2 Planification détaillée 7](#_Toc449965069)

[3 Réalisation 7](#_Toc449965070)

[3.1 Dossier de Réalisation 7](#_Toc449965071)

[3.2 Modifications 7](#_Toc449965072)

[4 Tests 7](#_Toc449965073)

[4.1 Dossier des tests 7](#_Toc449965074)

[5 Conclusion 7](#_Toc449965075)

[5.1 Bilan des fonctionnalités demandées 7](#_Toc449965076)

[5.2 Bilan de la planification 7](#_Toc449965077)

[5.3 Bilan personnel 7](#_Toc449965078)

[6 Divers 8](#_Toc449965079)

[6.1 Journal de travail 8](#_Toc449965080)

[6.2 Bibliographie 8](#_Toc449965081)

[6.3 Webographie 8](#_Toc449965082)

[7 Annexes 8](#_Toc449965083)

# 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

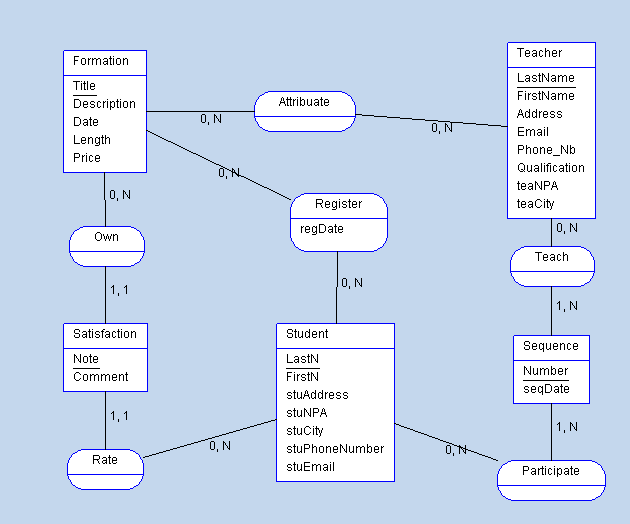
## Validation et conditions de réussite

* 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

## 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, vu qu’il ne nous permettait pas de respecter les normes ETML. C’est pourquoi nous nous sommes par la suite dirigés vers un logiciel dit « basique » de dessin « ***Microsoft Visio 2013*** » pour créer le MLD, puis sur « ***WorkBench*** » pour la création du script SQL.

#### Petite explication :

Par défaut, il faut que le professeur (Teacher), les formations (Formation) et les élèves (Student) doivent figurer dans la base de données.

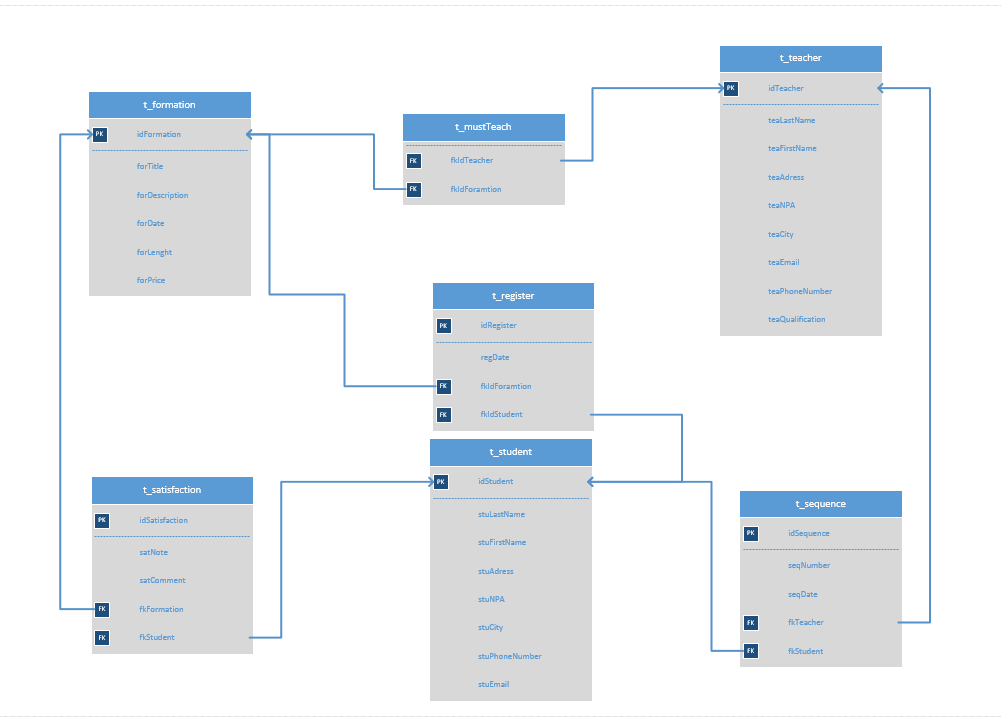
Un professeur peut enseigner zéro ou plusieurs formations. Formations dont zéro ou plusieurs élèves sont inscrits à une date précise (regDate).

Ensuite, une séquence (Senquence) peut être créée uniquement si au moins un élève et un professeur y participe.

Un (seul) élève peu remplir un formulaire de satisfaction pour une seule formation à la fois.

L’adresse a été décomposée en trois propriétés : le code postal, la ville et la rue ainsi que le numéro de la maison. Nous sommes partis du principe qu’un étudiant pouvait suivre plusieurs formations en même temps et qu’il n’était pas limité à une inscription à la fois. Un cours peut avoir plusieurs séquences données.

### Modèle Logique de Données (MLD).



Nous utilisons le logiciel ***Microsoft Visio 2013*** pour la création du MLD.

### Modèle Physique de Données (MPD)

Le MLD a été convertis dans le logiciel ***WorkBench*** pour que, par la suite, celui-ci le convertisse en script SQL de création de la base de données.

## Planification détaillée

Tout le projet est planifié sur le site Web ***Trello***.

Nous avons créé un tableau avec trois colonnes : To Do, In Progress et Done (à faire, en cours et fait). La colonne « To Do » est régulièrement copiée à partir du Product Owner (prof de classe) au fur et à mesure de notre avancement.

Lien vers notre tableau : [https://www.trello.com/](https://trello.com/b/jUyBNHDs/sprint-backlog-bcgn).

# Réalisation

## Dossier de Réalisation

## 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.