

GESPRO

Edition 2014

PROJET MACSI1 : Réalisation d'une
application de gestion de projet

RUCKEBUSCH Arnaud - NORBAL Xavier -
JOULOT Philippe - NAHA Myriam

Encadrant : Yehia Taher

Table des matières

1	Introduction	2
2	Approche du sujet et conception du modèle	3
2.1	Les outils utilisés	3
2.2	Méthode de travail	3
2.3	Modèle de classe	3
3	Transformation du modèle	5
3.1	Méthode de transformation	5
3.2	Modèle relationnel	5
3.3	Éléments 3FN et cohérence	5
4	Conception	7
4.1	Implémentation de la base de données	7
4.2	Développement de l'application	10
4.3	Difficultés rencontrées	11
5	Présentation de l'application	12
6	Bibliographie	13
6.1	Logiciels	13
6.2	Recherche d'informations	13
7	Conclusion	14

1 Introduction

Dans le cadre du module de méthodes d'analyse et de conception de systèmes d'informations nous avons été amenés à réaliser un projet visant à réaliser une application permettant la gestion de projets avec toutes ses contraintes. En effet, la bonne gestion de projet est un problème récurrent, ce projet a pour but de nous apprendre un processus de gestion de projet étant très grandement utilisé dans différents domaines et plus particulièrement en informatique. Le but final est de réaliser une application permettant la gestion de projet suivant le processus décrit.

2 Approche du sujet et conception du modèle

2.1 Les outils utilisés

Pour ce projet et pour nous simplifier le travail de groupe nous avons utilisés GIT (outils de gestion de version très répandu), notepad++ pour effectuer toute la partie rédaction du code, easyphp/wamp afin de tester notre application en mode local, starUML pour la réalisation du diagramme de classe, ainsi que TexWorks pour la rédaction de ce rapport.

2.2 Méthode de travail

Nous avons donc travaillé sur un répertoire git accessible à l'adresse suivante :

<https://github.com/xnorbal/MACSI1>

Via ce lien vous pourrez voir les différents moments auxquels nous avons travaillé, les différents commit que nous avons fait ainsi que nos différentes contributions.

2.3 Modèle de classe

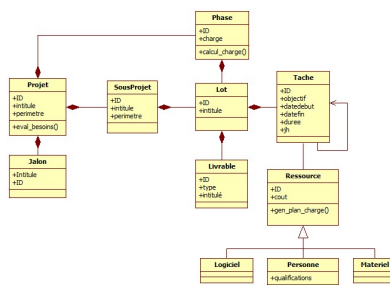


FIGURE 1 – Diagramme de classe

Pour ce diagramme de classe nous avons choisis de ne pas créer de classe tableau de bord, car nous pensons que cela correspond plus à une méthode qui va générer ce tableau de bord en fonction des autres informations du diagramme.

Un projet est composé de sous projets qui sont eux-mêmes composés de lots qui sont eux-mêmes composés de phases. Un projet est également composé de

phases qui est associé à 1 ou plusieurs livrables qui sont générés par les jalons. Les tâches dépendent d'elles-mêmes et ont chacune un taux d'affectation de ressources qui peuvent être de types humains, logistiques ou bien physiques.

3 Transformation du modèle

3.1 Méthode de transformation

Pour transformer le diagramme de classe précédent en modèle relationnel nous avons simplement appliqué les principes du cours de macsil en commençant par les spécialisations, s'en suit les relations et nous finissons par les agrégations pour arriver au modèle relationnel ci-dessous.

3.2 Modèle relationnel

Suite à la lecture du sujet et à différent rendez-vous avec M Taher nous avons établis le modèle relationnel suivant :

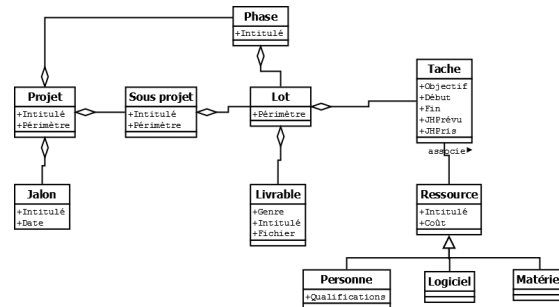


FIGURE 2 – Modèle relationnel

3.3 Eléments 3FN et cohérence

Afin de diminuer les anomalies et les redondances nous avons modifié le modèle relationnel de façon à obtenir une 3ème forme normale et faire en sorte à ce que les attributs non-clé ne soient plus dépendants d'autres attributs ne faisant pas partie de la clé.

Par exemple :

exemple3FN.sql

```
CREATE TABLE PROJET (
  ID INT AUTO_INCREMENT,
  INTITULE TEXT,
  PERIMETRE TEXT,
  PRIMARY KEY (ID)
);
```

```
CREATE TABLE SOUSPROJET (  
  ID INT AUTO_INCREMENT,  
  PID INT REFERENCES PROJET (ID) ON DELETE CASCADE,  
  INTITULE TEXT,  
  PERIMETRE TEXT,  
  PRIMARY KEY (ID)  
);
```

Nous voyons bien ici que les attributs non clé sont indépendants entre eux.

4 Conception

4.1 Implémentation de la base de données

Le modèle 3FN n'est pas toujours le plus efficace, en effet suivant les jointures que nous faisons régulièrement ou bien les restrictions courantes il peut être utile de dénormaliser afin d'obtenir des tables plus grandes mais permettant une meilleure intégrité des données et ceci sans perte d'information. Nous avons donc choisis la BD suivante :

BD_MACSI1.sql

```
CREATE TABLE RESSOURCEL(
  ID INT AUTO_INCREMENT,
  INTITULE TEXT,
  COUT INT,
  PRIMARY KEY (ID)
);

CREATE TABLE RESSOURCEH(
  ID INT AUTO_INCREMENT,
  INTITULE TEXT,
  COUT INT,
  QUALIFICATIONS TEXT,
  PRIMARY KEY (ID)
);

CREATE TABLE RESSOURCEM(
  ID INT AUTO_INCREMENT,
  INTITULE TEXT,
  COUT INT,
  PRIMARY KEY (ID)
);

CREATE TABLE PROJET(
  ID INT AUTO_INCREMENT,
  INTITULE TEXT,
  PERIMETRE TEXT,
  PRIMARY KEY (ID)
);

CREATE TABLE SOUSPROJET(
  ID INT AUTO_INCREMENT,
  PID INT REFERENCES PROJET(ID) ON DELETE CASCADE,
  INTITULE TEXT,
  PERIMETRE TEXT,
```



```

        PRIMARY KEY (ID)
    );

CREATE TABLE JALON (
    ID INT AUTO_INCREMENT,
    PID INT REFERENCES PROJET (ID) ON DELETE CASCADE,
    INTITULE TEXT,
    SYNCPOINT DATE,
    PRIMARY KEY (ID)
);

CREATE TABLE PHASE (
    ID INT AUTO_INCREMENT,
    PID INT REFERENCES PROJET (ID) ON DELETE CASCADE,
    INTITULE TEXT,
    PRIMARY KEY (ID)
);

CREATE TABLE LOT (
    ID INT AUTO_INCREMENT,
    SPID INT REFERENCES SOUSPROJET (ID) ON DELETE CASCADE,
    PhID INT REFERENCES PHASE (ID) ON DELETE CASCADE,
    PERIMETRE TEXT,
    PRIMARY KEY (ID)
);

CREATE TABLE TACHE (
    ID INT AUTO_INCREMENT,
    LID INT REFERENCES LOT (ID) ON DELETE CASCADE,
    OBJECTIF TEXT,
    DATEDEBUT DATE,
    DATEFIN DATE,
    JH_PREVU INT,
    JH_PRIS INT,
    PRIMARY KEY (ID)
);

CREATE TABLE LIVRABLE (
    ID INT AUTO_INCREMENT,
    LID INT REFERENCES LOT (ID) ON DELETE CASCADE,
    GENRE TEXT,
    INTITULE TEXT,
    FICHER TEXT,
    PRIMARY KEY (ID)
);

CREATE TABLE TACHERL (
    TID INT REFERENCES TACHE (ID) ON DELETE CASCADE,
    RLID INT REFERENCES RESSOURCEL (ID) ON DELETE CASCADE,

```

```

    TXAFFECTATION INT,
    PRIMARY KEY (TID, RLID)
);

CREATE TABLE TACHERH (
    TID INT REFERENCES TACHE (ID) ON DELETE CASCADE,
    RHID INT REFERENCES RESSOURCEH (ID) ON DELETE CASCADE,
    TXAFFECTATION INT,
    PRIMARY KEY (TID, RHID)
);

CREATE TABLE TACHERM (
    TID INT REFERENCES TACHE (ID) ON DELETE CASCADE,
    RMID INT REFERENCES RESSOURCEM (ID) ON DELETE CASCADE,
    TXAFFECTATION INT,
    PRIMARY KEY (TID, RMID)
);

CREATE TABLE CHRONOTACHE (
    TID1 INT REFERENCES TACHE (ID) ON DELETE CASCADE,
    TID2 INT REFERENCES TACHE (ID) ON DELETE CASCADE,
    PRIMARY KEY (TID1, TID2)
);

```

4.2 Développement de l'application

Notre application GESPRO permet à un utilisateur donné de répertorier ses projets avec toutes les informations relatives à ce dernier.

En effet dans un premier temps l'utilisateur crée un compte au niveau de l'application. Par la suite, il a la possibilité de créer un nouveau projet avant d'avoir la main pour insérer les phases et ajouter des sous-projets. Selon les attentes du projet l'utilisateur a la possibilité d'organiser ses sous-projets en différents lots autant de fois que nécessaire, créer des livrables et énumérer les différentes tâches tout en y attribuant les ressources adéquates (logicielles, humaines ou encore matérielles). L'application permet également de modifier ou supprimer tous les éléments cités ci-dessus.

Comme énoncé précédemment nous avons utilisés plusieurs outils pour le développement de notre application. Nous avons également fait le choix de développer cette application en html-css-php principalement car nous préférons ce langage à d'autres et que la gestion de la base de données s'y prêtait bien. Nous avons également utilisé du javascript notamment pour afficher des calendriers lors de la sélection de dates, ainsi que de l'ajax pour avoir des formulaires dynamiques.

Nous nous sommes répartis les différentes parties du projet afin de pouvoir les implémenter de manière parallèles et ainsi profiter au mieux de notre système de gestion de version sans créer trop de conflit lors de nos différents push sur notre dossier Git.

4.3 Difficultés rencontrées

Lors de ce projet les principales difficultés furent l'élaboration du modèle ainsi que la base de donnée, étant des éléments clés pour tout le développement du projet il fallait bien être sûr qu'avant de coder ces modèles étaient correct et que nous n'aurions pas à revenir dessus au cours du projet, sans quoi de nombreuses choses auraient été à modifier. Maîtrisant bien le langage web et il été plus long que difficile d'implémenter notre plateforme.

5 Présentation de l'application

Ci-dessous quelques screenshot de notre application :



FIGURE 3 – Tableau des ressources



FIGURE 4 – Tableau de bord avec diagramme de Gantt

6 Bibliographie

6.1 Logiciels

1. Git
2. Photophiltre
3. EasyPHP / wamp
4. TeXworks
5. Notepad++

6.2 Recherche d'informations

1. Github.com
2. Easyphp.com
3. Php.net
4. Cours de macsi1 du premier semestre
5. Wikibooks section LaTeX

7 Conclusion

Pour conclure, ce projet nous a permis de bien aborder ce processus de gestion de projet en le concrétisant dans la réalisation d'une application. Les différentes étapes du projet nous ont également permis de mettre à profit nos connaissances dans la conception, transformation de modèle et création de base de données ainsi que dans les langages web HTML/CSS/PHP/MYSQL que nous avons utilisés pour la réalisation de l'application.