

Rapport d'analyse - Gestion des projets de recherche d'un laboratoire



*Figure 1- FotoSearch... Search, recherche, projet de
recherche... Ahah, c'est drôle !*

Table des matières

I.	Introduction	3
II.	Modèle conceptuelle de données.....	4
A.	UML.....	4
B.	Explication du modèle et des choix de transformation	5
1.	Le personnel	5
2.	Les institutions	5
3.	La gestion de projet	6
III.	Modèle logique de données.....	7
A.	Le modèle relationnel textuel.....	7
B.	Explication du schéma et de ses choix de transformation	11
1.	Choix des clés	11
2.	Choix des héritages	11
3.	Autres choix	12
IV.	Normalisation.....	13
A.	Fermeture transitive	13
B.	Preuve des formes normales	14
1.	1 NF	14
2.	2 NF	14
3.	3 NF	14
V.	Conclusion	15
VI.	Annexes.....	16
A.	Note de clarification	16
1.	Contexte.....	17
2.	Définition du projet et objectifs	17
3.	Acteurs du projet	18
4.	Produit du projet	18
5.	Environnement du projet.....	19
6.	Contraintes à respecter	19
B.	Organigramme produit	20
1.	Environnements du produit du projet	21
2.	Fonctions.....	21
3.	Classements des fonctions	21
4.	Solutions.....	23
5.	Organigramme produit.....	24

I. Introduction

Le projet de NF17 est divisé en deux parties. La première consiste en l'analyse d'un des sujets proposés. Nous avons à faire l'analyse du sujet n°1 : un laboratoire de recherche veut créer un logiciel de base de données pour tenir à jour l'ensemble de ses projets de recherche. Notre analyse permettra ensuite d'implémenter un site internet répondant aux besoins des différents utilisateurs. Pour cela nous avons d'abord du expliciter les hypothèses du sujet.

Nous avons ainsi ajouté plusieurs contraintes :

- l'employé contact ne fait pas parti des membres du projet ;
- Un financeur a un unique employé contact pour chaque projet ;
- Le comité est un ensemble de personne qui n'a pas d'existence propre ;
- le document correspondant au projet n'a pas à être modélisé mais son contenu oui ;
- Une classe « laboratoire » sera créée afin que les membres d'un projet puissent faire partis de laboratoires différents.

Ces différents éléments sont à prendre en compte dans la définition puis dans la réalisation du projet, afin de satisfaire au mieux les attentes des clients.

II. Modèle conceptuelle de données

A. UML

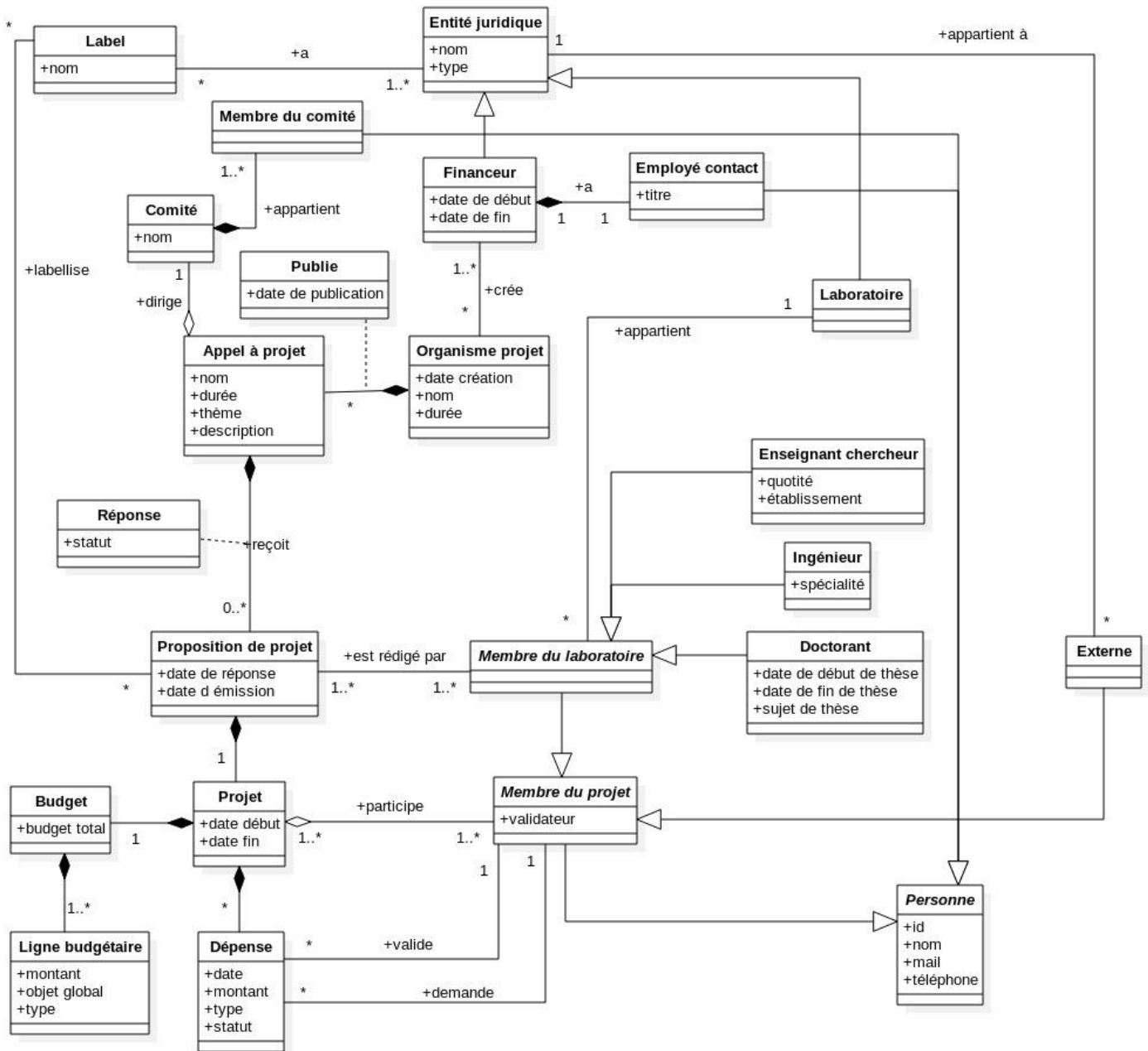


Figure 2 - Modèle UML

B. EXPLICATION DU MODELE ET DES CHOIX DE TRANSFORMATION

1. Le personnel

Nous avons créé une classe abstraite personne afin de pouvoir représenter toutes les personnes ayant une relation avec un projet. Nous avons procédé par héritage multiple afin de pouvoir représenter les personnes intervenant réellement. **Tous nos héritages concernant le personnel sont exclusifs.** Après le premier héritage, nous avons trois catégories de personne. Les membres du projet, l'employé contact et les membres du comité.

L'employé contact est en relation avec un financeur. Il possède un titre. Nous avons mis une composition car le jour ou le financeur disparaît l'employé n'a plus de raison d'être présent.

La classe membres du projet est une classe abstraite regroupant deux types d'employés. Ils peuvent soit être employé d'un laboratoire soit externes c'est à dire des membres de toutes autres entités juridiques.

Les membres du laboratoire se divisent enfin en trois sous catégories les doctorants, les ingénieurs et les chercheurs. Les ingénieurs possèdent une spécialité, les doctorants se différencient par la date de début et de fin de leur thèse et le sujet de leur thèse, enfin **les chercheurs travaille pour un établissement et a un nombre d'heure de recherche (quotité).**

Enfin seuls les membres du laboratoire peuvent rédiger une ou plusieurs propositions de projet.

2. Les institutions

Nous avons créé une classe entité juridique composée d'un nom (nom d'une ville, d'une région) et d'un type correspondant. Nous n'avons pas fait d'énumération car nous considérons que la liste donnée dans l'énoncé n'est pas exhaustive. Cette classe n'est pas abstraite car une entité peut exister sans jamais prendre part à un projet.

Toute entité juridique peut donner 0 ou plusieurs labels à une proposition de projet. Un laboratoire est une entité juridique mais **il ne peut pas attribuer des labels à des propositions de projet écrites par ses employés.**

Nous avons ensuite fait un héritage pour lequel nous avons séparé les laboratoires des autres financeurs. Cela nous permet de mettre en avant le lien spécial qu'un laboratoire a avec les propositions de projet.

Nous avons ensuite les propriétés suivantes pour le financeur :

- Un financeur a un employé contact référent ;
- Un financeur crée 0 ou plusieurs organismes projet.

Par ailleurs, un organisme projet est en charge de la publication des appels à projet. Un laboratoire est une entité juridique qui peut être un financeur pour un autre laboratoire ou qui peut être une entité au sein de laquelle on se positionne quand on fait le site de gestion.

Enfin, les comités projets sont composés de plusieurs membres. Ils peuvent diriger plusieurs appels à projet. Nous avons choisi l'agrégation pour lier ces deux classes car si un appel à projet disparaît le comité continue d'exister et peut gérer d'autres appels à projet.

3. La gestion de projet

La gestion d'un projet débute avec la publication d'un appel à projet par un organisme. Nous avons mis une composition entre appel à projet et organisme car l'appel à projet ne peut pas exister si l'organisme projet n'existe plus.

Un appel à projet est dirigé par un comité et il reçoit des propositions de projet. Nous avons fait une composition entre proposition de projet et appel à projet car on ne peut plus effectuer de proposition après une date limite et qu'une proposition est relative à un appel à projet existant sur le site d'offre. Chaque proposition de projet est soit validée, soit refusée, soit en attente de validation.

Un projet est créé après validation d'une proposition de projet. **Il hérite de son budget.** Un budget est propre à chaque projet. Il est composé de lignes budgétaires. Les lignes budgétaires permettent de détailler le budget et l'utilisation de l'argent. Ce sont des lignes directrices mais les dépenses réelles peuvent différer. Une dépense est demandée par un membre du projet et est validée par un autre membre du projet. Elle ne peut être effectuée que dans le cadre d'un projet précis.

III. Modèle logique de données

A. LE MODELE RELATIONNEL TEXTUEL

```
OrganismeProjet (#nom : string,  
                 date de création : Date,  
                 durée : int);  
  
AppelàProjet (#nom : string,  
               #organisme => OrganismeProjet.nom,  
               #publication : Date,  
               duree : int,  
               thème : string,  
               description : string,  
               comite =>Comite.nom) ;  
  
Propositiondeprojet (#numero : int,  
                      #appel => AppelàProjet.nom : string,  
                      #organisme => AppelàProjet.organisme,  
                      #publicationappel => Appel à Projet.publication,  
                      date de réponse : Date,  
                      date d'émission : Date,  
                      statut : bool) ;  
  
Projet (#numero => Propositiondeprojet.numero,  
         #appel => Propositiondeprojet.nom : string,  
         #organisme => Propositiondeprojet.organisme,  
         #publicationappel => Propositiondeprojet.publicationappel,  
         date début : Date,  
         date fin : Date) ;  
  
Budget (#numero => Projet.numero,  
         #appel => Projet.nom : string,  
         #organisme => Projet.organisme,  
         #publicationappel => Projet.publicationappel,  
         budget_total : float) ;  
  
LignesBudgétaires (#numero => Budget.numero,  
                    #appel => Budget.nom : string,  
                    #organisme => Budget.organisme,  
                    #publicationappel => Budget.publicationappel,
```

```
#objet : string,  
montant : float,  
type : Type);
```

```
Dépense (#numero => Projet.numero,  
#appel => Projet.nom : string,  
#organisme => Projet.organisme,  
#publicationappel => Projet.publicationappel,  
date : Date,  
montant : float,  
type : Type,  
status : int,  
demandeurext => Externe.id,  
demandeurlab => Membreslabo.id,  
valideurext => Externe.id,  
valideurlab => Membreslabo.id);
```

Contraintes:

- demandeurext, demandeurlab un des deux est non nul ;
- status compris entre 0 et 2 : 0 en attente 1 refusé ou 2 validé ;
- valideurext != demandeurext ;
- valideurlab != demandeurlab.

```
EmployeContact (#id : int,  
#financier => financier.nom,  
nom : string,  
mail : string KEY,  
telephone : int,  
titre : string) avec nom non nul ;
```

```
MembreComite (#id : int,  
#comite => comite.nom,  
nom : string,  
mail : string KEY,  
telephone : int) avec nom non nul ;
```

```
Externe (#id : int  
nom : string,
```



```
mail: string KEY,
telephone : int,
validateur : bool,
entité => Entité.id) avec nom, entité non nul ;

MembreLabo (#id : int,
    nom : string,
    mail : string KEY,
    telephone: int,
    dateDeb:
    date, DateFin : date,
    quotite : float,
    etablissement : string,
    specialite : string,
    validateur : bool,
    type {I,D,C},
    sujet : string,
    labo => laboratoire.id) avec nom, type non NULL ;

vIngenieur = Projection (Restriction(membreLabo, type = I), id, nom, mail,
telephone, validateur, labo, specialite) ;

vDoctorant = Projection (Restriction(membreLabo, type = D), id, nom, mail,
telephone, validateur, labo, sujet, DateDeb, DateFin) ;

vChercheur = Projection (Restriction(membreLabo, type = C), id, nom, mail,
telephone, labo, validateur, quotite, etablissement);

EntiteJuridique (#id : int,
    nom : string NOT NUL,
    type : string NOT NUL);

Comite (#nom : string) ;

Financeur (#id => entiteJuridique.id,
    dateDebut : date,
    dateFin: date) ;

Laboratoire (#id=>entiteJuridique.id) ;
```

```
vfinanceur = Jointure(entiteJuridique, financeur, entiteJuridique.nom =
financeur.nom);

vlaboratoire = Jointure(entiteJuridique, laboratoire, entiteJuridique.nom =
laboratoire.nom) ;

Redaction (#id Membrelabo =>MembreLabo.id,
            #appel => AppelàProjet.nom : string,
            #organisme => AppelàProjet.organisme,
            #publicationappel => Appel à Projet.publication,
            #numero : int) ;

Participantdulabo(#id => membreLabo.id,
                  #numero => Propositiondeprojet.numero,
                  #appel => Propositiondeprojet.nom : string,
                  #organisme => Propositiondeprojet.organisme,
                  #publicationappel =>
                  Propositiondeprojet.publicationappel) ;

Participantexterne(#id => Externe.id,
                   #numero => Propositiondeprojet.numero,
                   #appel => Propositiondeprojet.nom : string,
                   #organisme =>Propositiondeprojet.organisme,
                   #publicationappel =>
                   Propositiondeprojet.publicationappel) ;

Labellise(#nomdulabel => Label.nom,
           #proprojetnum => Propositionprojet.num,
           #proprojetappel=> Propositionprojet.nom
           #proprojetorgan => Propositionprojet.organisme,
           #proprojetappel => Propositionprojet.publicationappel) ;

DonneLabel(#nomdulabel => Label.nom,
            #entite => Entite.id) ;
```

B. EXPLICATION DU SCHEMA ET DE SES CHOIX DE TRANSFORMATION

1. Choix des clés

Dans la relation Proposition de projet nous avons utilisé une clé artificielle puisqu'on a considéré qu'on peut soumettre plusieurs propositions pour le même appel à projet le même jour. Nous en avons également utilisé une pour la relation entité car plusieurs entités peuvent avoir le même nom (par exemple deux villes) et pour personne les autres attributs ne composant pas une clé optimale selon nous (nom, mail..).

Dans la relation Lignes budgétaires, on a considéré que la ligne budgétaire était définie par son objet puisque dans un budget chaque objet n'est référencé qu'une seule fois (ex : une ligne matériel, une ligne photocopie...).

Nous avons souvent utilisé des clés primaires composées de clés étrangères. Il s'agit du cas des compositions comme par exemple pour l'appel à projet, le projet, le budget...

2. Choix des héritages

Pour l'héritage de financeur et laboratoire, nous avons considéré la classe mère comme étant non abstraite, il existe des entités juridiques qui ne sont ni des laboratoires, ni des financeurs. L'héritage n'est pas complet puisque les classes filles ont des associations propres (par exemple laboratoire avec membre du laboratoire), l'héritage n'est pas non plus exclusif puisque les laboratoires peuvent être des financeurs et vis versa. En l'absence d'une meilleure solution nous avons donc choisi un héritage par référence.

Pour l'héritage membre du projet, membre du comité, l'héritage est exclusif puisque nous avons considéré que les membres du projet ne sont pas membre du comité. De plus la classe mère personne est abstraite, nous avons donc favorisé un héritage par les classes filles il nécessite l'ajout des attributs de la relation personne aux relations membre du projet et membre du comité. La relation personne ne sera donc pas présente dans notre modèle relationnel.

Pour l'héritage membre du laboratoire et externe, l'héritage est exclusif puisque les membres du projet sont soit externe soit membre du laboratoire et pas les deux

en même temps, La classe mère membre du projet est abstraite c'est pourquoi nous avons décidé d'utiliser un héritage par les classes filles, il nécessite l'ajout des attributs de la relation membre du projet aux relations membre du laboratoire et externe. La relation membre du projet ne sera donc pas présente dans notre modèle relationnel.

Pour l'héritage doctorants, enseignant chercheur et ingénieur, l'héritage étant complet, nous avons décidé d'utiliser un héritage par la classe mère. On a donc ajouté à la classe mère un attribut type qui prend sa valeur parmi ces trois catégories la classe mère étant abstraite (un membre du laboratoire est forcément un enseignant chercheur, un doctorant, ou un ingénieur) ce type ne doit pas être nul. Nous avons également créé **trois vues afin de pouvoir visualiser les membres de chaque classe ainsi que ces attributs propres.**

3. Autres choix

On a préféré mettre un attribut date de publication dans la classe Appel à projet plutôt qu'une classe d'association Publication jugée redondante. De plus on a considéré le cas où un Organisme Projet redemande un appel à projet ayant le même nom (pas de réponse durée expirée ou approfondissement du sujet). **Par soucis d'homogénéité dans notre relationnel nous avons utilisé la même méthode pour le statut de la Proposition de Projet.**

Lors de la transformation de la relation d'héritage vers membres du projet, nous avons perdu l'information contenue dans notre association vers le projet. Nous avons donc créé deux tables Participantdulabo et Participantexterne.

Nous avons ajouté une contrainte sur la table dépense afin d'éviter qu'une dépense soit validée par son demandeur. Nous avons également créé dans cette même table deux attributs demandeurs et validateurs afin de ne pas perdre l'information contenue par la classe abstraite de notre UML membre du projet. Ainsi nous pouvons représenter les membres du laboratoire ainsi que les externes ce qui a nécessité l'ajout de deux clés étrangères vers les tables ParticipantLabo et ParticipantExterne.

Il sera également nécessaire de créer une énumération pour le Type de la dépense (fonctionnement ou matériel).

IV. Normalisation

A. FERMETURE TRANSITIVE

De ce que l'on peut déduire du sujet, de notre modèle conceptuel et de notre modèle relationnel, nous avons les dépendances fonctionnelles suivantes :

- Pour l'organisme projet, nous avons les dépendances suivantes :

nom -> date de création, durée

- Pour l'appel à projet, nous avons les dépendances suivantes :

(nom, publication, organisme) -> durée, thème, description, comité

- Pour la proposition de projet, nous avons les dépendances suivantes :

(numero, appel, publicationappel, organisme) -> date de réponse, date d'émission, statut

- Pour le projet, nous avons les dépendances suivantes :

(numero, appel, publicationappel, organisme) -> date de réponse, date d'émission

- Pour le budget, nous avons les dépendances suivantes :

(numero, appel, publicationappel, organisme) -> budget total

- Pour les lignes budgétaires, nous avons les dépendances suivantes :

(numéro, appel, organisme, publicationappel, objet) -> montant, type

- Pour une dépense, nous avons les dépendances suivantes :

(numéro, appel, organisme, publicationappel) -> date, montant, type, statut, demandeurExt, demandeurLab, validateurExt, validateurLab

- Pour un employé contact, nous avons les dépendances suivantes :

(id, financeur) -> nom, mail, telephone, titre

(mail, financeur) -> id, nom, telephone, titre

- Pour un membre du comité, nous avons les dépendances suivantes :

(id, comité) -> nom, mail, telephone

(mail, comité) -> id, nom, telephone

- Pour un externe, nous avons les dépendances suivantes :

id-> nom, mail, telephone, validateur, entité

mail -> id, nom, telephone, validateur, entité

- Pour un membre du laboratoire, nous avons les dépendances suivantes :

*id -> nom, mail, telephone, dateDeb, dateFin, quotité,
établissement, spécialité, type, sujet validateur,
laboratoire*

*mail -> id, nom, telephone, dateDeb, dateFin, quotité,
établissement, spécialité, type, sujet validateur,
laboratoire*

- Pour une entité juridique, nous avons les dépendances suivantes :

id - > nom, type

- Pour un financeur, nous avons les dépendances suivantes :

id -> dateDébut, dateFin

B. PREUVE DES FORMES NORMALES

1. 1 NF

Le schéma relationnel est en 1ère forme normale car toutes les relations possèdent au moins une clé et ont tous leurs attributs atomiques.

2. 2 NF

Le schéma relationnel est en 2nd forme normale car toutes les relations sont en 1NF et tous leurs attributs qui ne sont pas dans une clé ne dépendent pas d'une partie seulement d'une clé, mais de la clé entière.

3. 3 NF

Le schéma relationnel est en 3ème forme normale car toutes les relations sont en 2NF et tous leurs attributs qui ne sont pas dans une clé ne dépendent pas d'autres attributs n'appartenant pas à une clé.

V. Conclusion

Nous avons volontairement essayé de généraliser au plus grand nombre de cas possibles notre modèle de façon à ce qu'il soit facilement adaptable pour convenir à d'autres projets de ce genre. Cela a rendu le travail de compréhension et de conceptualisation bien plus prise de tête, mais a été très enrichissant.

Cependant, lors de la phase de réalisation, il va falloir se limiter à certaines fonctionnalités car implémenter tout le système serait extrêmement chronophage¹.

Enfin, nous espérons que ce document répondra à son but premier : permettre à l'équipe réalisatrice de développer une solution fonctionnelle et ainsi simplifier le travail de développement.

¹ <http://www.potiondevie.fr/wp-content/uploads/2013/10/mind-maps-chronophage.jpg>

VI. Annexes

A. NOTE DE CLARIFICATION

Référence du document	NotedeClarification.doc	version	2
Auteurs création		Date de création	26/02/2015
Auteur modification	Cohen-Tanugi Dylan Michel-Dansac Pauline Raingeval Lucas Weill Solene	Date de modification	31/03/2015

1. CONTEXTE

2. DEFINITION DU PROJET ET OBJECTIFS

- A) DEFINITION DU PROJET
- B) OBJECTIFS VISES
- C) DEBUT DU PROJET
- D) FIN DU PROJET
- E) BUDGET DU PROJET

3. ACTEURS DU PROJET

- a) MAITRE D'OUVRAGE
- b) MAITRE D'ŒUVRE
- c) COMPETENCES NECESSAIRES

4. PRODUIT DU PROJET

5. ENVIRONNEMENT DU PROJET

6. CONTRAINTES A RESPECTER

- A) CONTRAINTES DE DELAIS
- B) CONTRAINTES DE COUTS

1. Contexte

Le laboratoire de recherche Heudiasyc, instigateur du projet, est une unité mixte de recherche entre l'Université de Technologie de Compiègne (UTC) et le CNRS. Son directeur-adjoint : Yves Grandvalet souhaite bénéficier d'une base de données couplée d'un outil web facilitant la gestion de l'ensemble des projets qui sont soumis au laboratoire.

2. Définition du projet et objectifs

a) Définition du projet

Le laboratoire de recherche Heudiasyc veut créer un logiciel de base de données pour tenir à jour l'ensemble de ses projets de recherche. Ce projet répond à quatre besoins :

- Gérer les informations liées aux projets et propositions de projet du laboratoire ;
- Gérer les informations des organismes projets, des appels à projets et des financeurs ;
- Gérer les informations des membres des projets, internes ou externes ;
- Permettre de réaliser des recherches d'informations sur les projets (les appels d'offre en cours non répondus, etc.) et des études statistiques (mois de l'année où on fait le plus de dépenses de projet au laboratoire).

b) Objectifs visés

Les objectifs visés sont les suivants :

- Simplifier la gestion des projets;
- Informatiser cette gestion.

c) Début du projet

Le projet débute le 26 Février 2015.

d) Fin du projet

Le projet se termine le 18 Juin 2015.

e) Budget du projet

Nous ne disposons d'aucun budget.

3. Acteurs du projet

a) Maître d'ouvrage

La personne en qualité de maître d'ouvrage est :

- Taha Arbaoui.

b) Maître d'œuvre

Les personnes en qualité de maître d'œuvre sont :

- Dylan Cohen-Tannugi,
- Lucas Raingeval,
- Solène Weill,
- Pauline Michel-Dansac.

c) Compétences nécessaires

Compétence	Service	Responsable	Personne pressentie	Remarque
esprit de synthèse		chef de projet	Solène Weill	
	service qualité	responsable qualité	Lucas Raingeval	
		responsable modélisation	Pauline Michel-Dansac	
	service informatique	responsable technique	Dylan Cohen Tanugy	

4. Produit du projet

a) Une base de données

Elle devra permettre de gérer l'ensemble des projets en satisfaisant les besoins suivants :

- Gérer l'ensemble des organismes projets
- Gérer l'ensemble des participants d'un projet
- Modifier les participants d'un projet
- Permettre de réaliser divers études statistiques (budget moyen d'un projet, etc.)

b) Interface Homme Machine

L'interface Homme Machine devra permettre au personnel du laboratoire Heudysac et au demandeur de projet d'utiliser la base de données de la façon la plus claire possible.

Elle devra :

- Être accessible depuis n'importe quel lieu à l'aide d'une identification selon un nom d'utilisateur et un mot de passe;
- Permettre la consultation et la modification des données selon le type d'utilisateurs;
- Posséder une partie côté financeur et organismes projets, et une autre pour les membres des laboratoires :
 - La partie accessible par les organismes projets leur permet de consulter les différents appels et les différentes offres que leurs appels ont suscité et dès lors accepter l'une des propositions;
 - La partie accessible par les membres des laboratoires leur donne la possibilité de voir les différents projets auxquels ils participent, de valider les dépenses s'ils en ont le droit et de proposer des projets.

5. Environnement du projet

Le service de gestion des projets ne pouvant fermer pendant l'implémentation du système, il faut que cela se fasse de manière transparente.

6. Contraintes à respecter

a) Contraintes de délais

Les délais à respecter sont les suivants :

- Début du projet : 26 Février 2015
- Rendu intermédiaire : 19 Mars 2015
- Rendu du rapport d'analyse : 09 Avril 2015
- Fin du projet : 18 Juin 2015

b) Contraintes de coût

Le budget étant nul, il n'y a aucune contrainte de coût à signaler.

B. ORGANIGRAMME PRODUIT

Référence du document	OrganigrammeProduit.doc	version	1
Auteurs création		Date de création	26/02/2015
Auteur modification	Cohen-Tanugi Dylan Michel-Dansac Pauline Raingeval Lucas Weill Solene	Date de modification	31/03/2015

1. ENVIRONNEMENTS DU PRODUIT DU PROJET

2. FONCTIONS

3. CLASSEMENTS DES FONCTIONS

- a) CREATION DE PROJETS
- b) PROJETS
- c) STATISTIQUE
- d) RECHERCHE

4. SOLUTIONS

5. ORGANIGRAMME PRODUIT

1. Environnements du produit du projet

Le produit s'adresse aux membres des laboratoires, aux laboratoires et aux membres des organismes projets.

2. Fonctions

Le produit du projet doit permettre aux laboratoires de :

- répondre aux appels à projets par des propositions;
- consulter les différents appels à projets;
- voir les différents projets et leurs détails;
- voir les différentes propositions de projets et leurs détails;
- obtenir des statistiques sur les projets;
- gérer les membres d'un projet.

Le produit du projet doit permettre aux membres d'un projet de :

- consulter les informations relatives aux projets dans lesquels ils sont impliqués;
- valider des dépenses pour ceux disposant des autorisations;
- demander des dépenses.

Le produit du projet doit permettre aux membres des organismes projets de :

- consulter les propositions de projets relatives à leurs appels à projets;
- valider ou refuser les propositions de projets;
- consulter les détails des projets validés;
- faire un appel à projet.

3. Classements des fonctions

a) Création de projets

Le produit du projet doit permettre aux laboratoires de :

- répondre aux appels à projet par des propositions;

Le produit du projet doit permettre aux membres d'un organisme projet de :

- faire un appel à projet;
- valider ou refuser les propositions de projets.

b) Projets

Le produit du projet doit permettre aux membres d'un projet de :

- valider des dépenses pour ceux disposant des autorisations;
- demander des dépenses.

Le produit du projet doit permettre aux laboratoires de :

- gérer les membres d'un projet.

c) Statistique

Le produit du projet doit permettre au laboratoire de :

- obtenir des statistiques sur les projets.

d) Recherche

Le produit du projet doit permettre au laboratoire de :

- voir les différents projets et leurs détails;
- consulter les différents appels à projets;
- voir les différentes propositions de projets et leurs détails;

Le produit du projet doit permettre aux membres d'un projet de :

- consulter les informations relatives aux projets dans lesquels ils sont impliqués;

Le produit du projet doit permettre aux membres d'un organisme projet de :

- consulter les détails des projets validés;
- consulter les propositions de projets relatives à leurs appels à projets;

4. Solutions

Fonction	Solution
Création de projet	Rubrique : création de projet
Appels à projet <ul style="list-style-type: none"> - faire un appel à projet; 	Rubrique : appels à projet <ul style="list-style-type: none"> - création d'un appel - listing des appels
Propositions de projets <ul style="list-style-type: none"> - répondre aux appels à projet par des propositions; - valider ou refuser les propositions de projets. 	Rubrique : Proposition de projets <ul style="list-style-type: none"> - création d'une proposition listing des propositions
Projets	
Projet <ul style="list-style-type: none"> - gérer les projets en cours 	Rubrique : Projet <ul style="list-style-type: none"> - listing des projets - gestion des membres <ul style="list-style-type: none"> - ajout - retrait - modification des droits - dépenses <ul style="list-style-type: none"> - formulaire de demande - historique des dépenses - valider une dépense
Statistiques	
Statistiques <ul style="list-style-type: none"> - visualiser les résultats et statistiques des projets 	Rubrique : Statistiques <ul style="list-style-type: none"> - dépenses de projet par mois - meilleurs laborantins du mois - projets
Recherche	
Recherche	<ul style="list-style-type: none"> - Plan du site - Recherche par mots-clés - Recherche avancée par thème

5. Organigramme produit

