

Projet NF17 :
Réalisation
d'un journal électronique :
Le Fil Electronique



Sommaire

I. Note de clarification

I.1. Contexte du projet :

I.2. Définition du projet :

I.3. Acteurs du projet :

I.4. Produit du projet :

I.5. Risques du projet :

I.6. Contraintes du projet :

II. Organigramme produit

III. Réalisation de l'UML

III.1. UML

III.2. Hypothèses réalisées

IV. Réalisation du MLD et Normalisation



I. Note de clarification

Réalisation d'un journal électronique : le Fil Electronique

1. Contexte du projet

L'objectif du projet est de réaliser un système informatique permettant la gestion d'un journal étudiant en ligne.

Actuellement, à l'UTC il existe un journal étudiant sous forme de papier, ce qui laisse beaucoup moins de souplesse quant à la gestion de celui-ci. En effet, à l'heure actuelle, il est complexe pour les étudiants de commenter des articles, d'en rechercher etc...

Le projet concernant la création d'un Fil Électronique permettra la création d'articles, leur catégorisation, leur recherche, la gestion du suivi éditorial, et l'apposition de commentaires.

2. Définition du projet

2.1. Problématique

Le "Fil-électronique" est un projet permettant l'informatisation d'un journal étudiant. Ce projet permettra, comme pour la version papier, de créer des articles et de les soumettre à un comité éditorial. La plus-value de cette version informatique doit résider en la catégorisation des articles (et donc leur recherche), une meilleure gestion du suivi éditorial (soumission et validation des articles), ainsi que la possibilité aux utilisateurs/lecteurs de commenter ou noter un article.

2.2. Objectifs visés

- Modélisation conceptuelle d'une base de donnée
- Implémentation de la base de donnée sur les serveurs de l'UTC
- Mise en place d'une interface web
- Délai : 16 semaines

2.3. Date de début du projet

Le projet démarre le mercredi 25 février 2015.



2.4. Date de fin du projet

Les livrables du projet doivent être rendu au plus tard le 17 juin 2015.

2.5. Budget du projet

Le projet "Le Fil-électronique" s'inscrit dans l'enseignement de l'unité de valeur NF17 de l'Université de Technologie de Compiègne. Par conséquent, il est réalisé gratuitement par les étudiants.

3. Acteurs du projet

3.1. Maîtrise d'ouvrage

Monsieur Benjamin Lussier, chargé de TD de l'UV NF17.

3.2. Maîtrise d'oeuvre

- Chef de projet : Valentin Feld
- Responsable modélisation : Kevin Chidiac
- Responsable technique : Shuhan Lin
- Responsable qualité : Alexis Wanderoild
- Secrétaire : Marie Daguin

3.2. Maîtrise d'oeuvre

Compétences	Service	Responsable
compétences de gestion de projet, bonnes capacités relationnelles, connaissances techniques dans les domaines concernés	Chef de Projet	Valentin Feld
Rigueur, adaptabilité	Responsable modélisation	Kevin Chidiac
compétences techniques, capacité à prendre des décisions, esprit d'initiative	Responsable technique	Shuhan Lin



Être à l'écoute, disponible et s'adapter, Être très rigoureux(se) et précis(e), Analyser et synthétiser de nombreuses informations, Gérer des priorités	Responsable qualité	Alexis Wanderoild
Utilisation d'outils bureautiques (traitement de texte, tableur,...), Modalités d'accueil, Techniques de prise de notes, Techniques d'écriture rapide	Secrétaire	Marie Daguin

4. Produit du projet

- Rapport d'analyse
- Journal électronique

5. Risques

- Perte de données
- Interface peu claire
- Sécurité des informations des utilisateurs
- Mauvaise gestion dans la modération des commentaires
- Redondance des articles

6. Contraintes

6.1. Délais

- Rendu intermédiaire : le 16 mars 2015
- Livraison du rapport d'analyse : le 8 avril 2015
- Échange des sujets : le 22 avril 2015
- Livraison du site et démonstration : le 17 juin 2015

6.2. Performances

Optimisation des requêtes sur la base de donnée, normalisation des relations, facilité de maintenance et de mises à jour.



Date : 07 /03 / 2015	Version : 1
Signature du chef de projet :	Signature du maître d'ouvrage :

Note de clarification réalisée par Valentin Feld, Kevin Chidiac, Shuhan Lin, Alexis Wanderoild et Marie Daguin.



II. Organigramme produit

Voir Annexe 1

III. Réalisation de l'UML

III.1. UML

Voir Annexe 2

III.2. Hypothèses réalisées

Hypothèses :

- Le droit des lecteurs est partagé par tous les autres acteurs
- L'adresse mail fournie par l'utilisateur doit être unique
- Un mot clé peut être associé à plusieurs articles.
- Tous les articles doivent être indexé (posséder au moins un mot clé)
- Toute sous rubrique est une rubrique et peut contenir des articles et des sous rubriques
- Un acteur peut chevaucher les droits des différents type d'acteur. Par exemple un modérateur peut également être un éditeur.
- L'auteur d'un article ne peut pas être son éditeur
- Les commentaires des lecteurs sont également soumis à historique
- Article.numéro > 0



Contraintes :

Acteur	Action	Contrainte
Auteur	Récupérer Article	Article.supprimé = 1
Editeur	Publication Article	Article.statut= 'validé'
Auteur	Modifier Article	Article.statut='en redaction'
Auteur	Voir article	Article.statut='en redaction'
Lecteur	Toutes les actions sur Article	Article.publié=1
Lecteur	Lire commentaire	Article.masqué=0

Toutes les dates des historiques sont non nulles.



IV. Réalisation du MLD et Normalisation

Personne (#login :string ,mail :string, nom :string, prenom :string) WITH mail KEY

login → nom

login → prenom

login → mail

mail → login

“login” et “mail” sont des clés candidates. Nous choisissons alors “login” en clé primaire.

La relation est en 3NF: les attribus sont atomiques et il y a au moins une clé (1NF), aucune partie de clé determine des attributs non clé car les clés comportent une unique attribut (2NF) et un attribut non clé ne détermine pas d'autres attributs non clé (3NF). Ceci est vérifié pour les deux clés (primaire et candidate).

Pour traiter l'héritage de la classe Personne, nous avons choisi la représentation par référence. En effet, Personne est une classe abstraite mais l'héritage n'est pas exclusif (d'où le rejet de la méthode par les classes filles), de plus l'héritage n'est pas complet ou presque complet ce qui réduit l'efficacité de la représentation par la classe mère.

Administrateur (#loginAdmin=>Personne(login))

Auteur (#loginAu =>Personne(login))

Editeur (#loginEdit=>Personne(login))

Lecteur (#loginLec =>Personne(login))

Modérateur (#loginMode=>Personne(login))

Toutes ces classes possèdent un unique attribut : la clé primaire, il n'y a pas de DF.

Ces relations sont donc en 3NF.

ComiteEditorial (#nom_groupe :string)

Compose (#editeur=>Editeur(login), #comité=>ComiteEditorial(nom_groupe))

Toutes ces relations sont toutes clés, il n'y a pas de DF.

Ainsi, ces deux relations sont en 3NF.



Rubrique(#nom :string, createur=> Editeur(loginEdit), dateCreationRub)
WITH dateCreationRub NOT NULL, WITH createur NOT NULL

Ici, seul “nom” est clé candidate, c’est donc notre clé primaire.

La table Rubrique est en 3NF.

Ici, nous allons nous traiter l’héritage de la classe Bloc. Nous avons choisi la représentation par les classes filles, Bloc étant une classe abstraite dont l’héritage est exclusif.

Image(#titre :string, #numeroArt=>Article(numéro), image:image) WITH (numeroArt, image) KEY

Texte(#titre :string, #numeroArt=>Article(numéro), texte:string) WITH (numeroArt, texte) KEY

Pour ces 2 relations, nous avons 2 clés candidates. Nous choisissons le couple (titre, numeroArt) comme clé primaire.

Aucun attribut non clé ne détermine un autre attribut non clé, ces 2 relations sont donc en 3NF

Statut (#statut:enum_statut, createur => Editeur(loginEdit), dateCreationStatut)
WITH dateCreationStatut NOT NULL, WITH createur NOT NULL

Ici, seul “statut” est clé candidate, c’est donc notre clé primaire.

La table Statut est en 3NF.

Remarque (#remarque:string, #statut=>Statut)

Remarque est une relation toute clé, elle est donc en 3NF.



Commentaire (#id:int, titre:string, texte:string, enExergue:boolean, masque:boolean, supprime:boolean, createur=>Lecteur(login),article=>Article(numero),dateCreation:date, dateSuppression:date)
WITH (titre,texte) KEY, WITH article NOT NULL, WITH createur NOT NULL,
WITH dateCreation NOT NULL

Commentaire en 3NF

MotsClefs (#keyword:string, editeur => Editeur(loginEdit), dateCreationMot)
WITH dateCreationMot NOT NULL

keyword est la seule clé candidate, c'est donc la clé primaire.

Les attributs non clés ne peuvent pas se déterminer entre eux, MotsClefs est donc en 3NF

Article
(#numero:int,titre:string,supprime:boolean,publie:boolen, honneur:boolean,
comite=>ComiteEditorial(nom_groupe))
WITH statut NOT NULL, WITH comite NOT NULL

De même, Article est aussi en 3NF

Ici, nous traitons l'historique des actions de l'Auteur

Modifier_Statut(#date:Date,#fk_Auteur => Auteur(loginAu),#fk_Article =>
Article(numero),#fk_Statut=>Statut(statut))

Relation toute clé, elle est en 3NF

Il en est de même pour l'historique du modérateur (ci-dessous)



Supprimer_Commentaire(Date:date,#fk_Commentaire=>Commentaire(id)
,#fk_Modérateur => Modérateur(loginMode)) WITH date NOT NULL

Masquer_Commentaire(#Date:date,#fk_Commentaire=>Commentaire(id)
,#fk_Modérateur => Modérateur(loginMode))

Mettre_Exergue_Commentaire(#Date:date,#fk_Commentaire=>Commentaire(id)
,#fk_Modérateur => Modérateur(loginMode))

Ainsi que pour l'éditeur :

Mettre_A_Honneur(#editeur=>Editeur(loginEdit), #article=>Article(id),#date:Date)

Associer_Article_Rubrique(#date: Date, #fk_article = > Article (numero),
#fk_Rubrique => Rubrique(nom), #fk_Editeur => Editeur(loginEdit))

Modifier_Statut(#date: Date, #fk_Editeur => Editeur(loginEdit),
#fk_statut => Statut(statut), #fk_article => Article(numero))

Associer_Article_Article(#date:Date, #fk_article1 = > Article (numero),
#fk_article2= > Article (numero), #fk_Editeur => Editeur(loginEdit))

Indexer_Article(#date: Date, #fk_motcle => Mot Clef(keyword),
#fk_Editeur => Editeur(loginEdit), #fk_article => Article(numero))

Associer_Sous_Rubrique_Rubrique(#date:Date,#fk_Editeur =>
Editeur(loginEdit),
#fk_rubrique => Rubrique(nom), #fk_sousrubrique => Rubrique(nom))

Vues et contraintes :

Contraintes :



- 1) PROJ(Personne,login) IN (UNION (PROJ(Editeur,loginEdit), UNION (PROJ(Administrateur,loginAdmin), UNION (PROJ(Moderateur,loginMode),UNION (PROJ(Auteur,loginAu),PROJ(Lecteur,loginLec)))).
- 2) Proj(Article,numero) IN Proj(Indexer_Article,fk_article)

Vues :

vEditeur=join(Personne,Editeur,login=loginEdit)

vAdministrateur=join(Personne,Administrateur,login=loginAdmin)

vModerateur=join(Personne,Moderateur,login=loginMode)

vAuteur=join(Personne,Auteur,login=loginAu)

vLecteur=join(Personne,Lecteur,login=loginLec)

vArticleNonSupprime=rest(Article,supprime='0')

vArticleHonneur=rest(Article,honneur='1')

vArticlePublie=rest(Article,publie='1')

vCommentaireExergue=rest(Commentaire,enExergue='1')

vCommentaireNonMasque=rest(Commentaire,masque='0')

vCommentaireNonSupprime=rest(Commentaire,supprime='0')

