

Projet Base de données

Rapport analyse et conception

Développement d'une application Web

Tutrice : Marie-Eléonore Kessaci

Trinôme : Saliou Diallo / Cosnelle Djoumekoum/ Safia Zendouz

Présentation :

Notre projet en base de données consiste à développer une application web qui consiste à gérer l'ensemble des articles rédigés par un chercheur ou un enseignant-chercheur. Ceux-ci existent déjà dans une bibliographie dblp.

L'enjeu ici est de permettre à l'utilisateur de consulter et de mettre à jour, le cas échéant, ses publications d'une façon accessible et assez complète. Pour ce faire, une telle gestion tient compte de plusieurs critères relatifs à l'article recherché. Chaque article est donc caractérisé par son titre, son auteur, sa date de soumission et celle de publication. Cette distinction de date est due au fait que l'article ne soit pas directement publié dans une source. En effet, une fois que la rédaction de l'article est terminée, celui-ci est soumis et examiné par un comité de lecture qui juge si l'article est accepté, rejeté ou en révision. Si l'article est accepté, il est donc publié dans une des sources considérées. Dans notre cas, seules les conférences, les revues scientifiques, les thèses et les livres sont considérés. Autrement, il est soit rejeté soit soumis en révision. Il faut donc avoir une trace des éventuelles modifications de ces derniers si on suppose que le chercheur essaie de soumettre son article à nouveau.

Ceci dit, le chercheur est celui qui est notifié par la décision du comité de lecture. Autrement dit, c'est lui qui s'occupe de la mise à jour du statut de l'article. C'est la raison pour laquelle il cherche un outil qui lui permet non seulement de consulter ses articles d'une façon « intelligente » mais aussi de les modifier et d'en rajouter des nouveaux.

Par ailleurs, un constat statistique des publications permet indéniablement au chercheur de percevoir ses publications sous un autre angle. Il peut donc avoir une idée sur le

pourcentage de collaboration avec un chercheur X, ou un nombre d'articles publiés dans une revue scientifique Y. Bref, toute information qui lui sera utile.

En somme, l'application Web sera mise en place pour répondre à tous les besoins exprimés par l'utilisateur. Il aura donc la main pour consulter ses articles publiés mais aussi ceux qui sont rejetés ou en révision (contrairement à la bibliographie dblp). Une modification reste tout de même inévitable étant donné que l'utilisateur sera constamment amené à changer le statut de ses articles (accepté-rejeté-en révision) par exemple. Il pourra également saisir les mots clé relatifs à son article ce qui facilitera son accès mais aussi son attribution à un thème précis.

Cahier des charges :

1. Contexte et présentation du projet :

Développement d'une application Web.

Contexte :

Un chercheur consulte généralement ses articles sur une bibliographie assez complète. Ils sont généralement organisés selon la date de publication ou par ordre alphabétique. Ce qui n'est pas assez pratique pour l'utilisateur. D'autant plus que, accéder à ses articles rejetés ou soumis en révision par le comité de lecture s'avère difficile voire impossible. Ce qui le pousse à vouloir une alternative qui lui facilite la tâche.

Objectifs :

Développer une application Web est la solution que nous proposons. Celle-ci permet au chercheur de :

- Consulter l'ensemble de ses articles, même ceux qui n'étaient pas accessibles sur la bibliographie, selon plusieurs critères ;
- Mettre à jour ses articles ;
- Avoir un aperçu statistique vis-à-vis des thèmes abordés, des collaborations effectuées...

2. Besoins et contraintes liés au projet :

Besoins fonctionnels :

- Accès à la base de données contenant les articles du chercheur ;

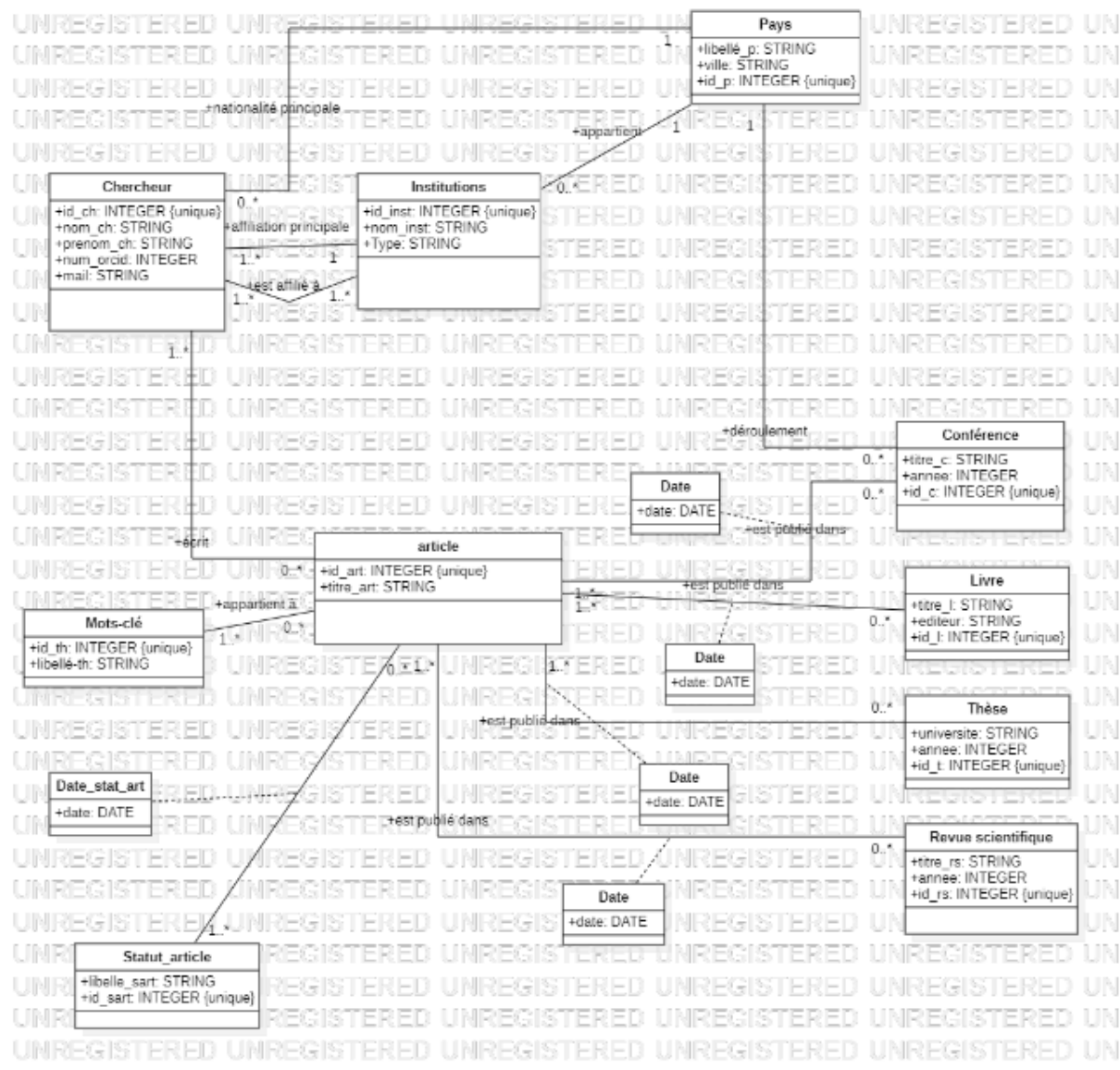
Contraintes :

- Développer l'application avec HTML/PHP et Postgres ;
- Mettre en avant le lien entre la première version d'un article rejeté/soumis en révision et celle qui a été acceptée après la modification du chercheur ;
- Le nombre de révisions maximal possible est de 2 ;
- Les articles peuvent être publiés dans une conférence, une revue scientifique, une thèse, un livre ;
- Permettre l'ajout d'une nouvelle publication ;
- Afficher les publications selon : les mots-clés, le type de la publication, les liens entre elles, l'année et leur état.

3. Résultats attendus :

Une application fonctionnelle répondant aux exigences du client.

Modèle UML :



Les tables Personne, article et institutions sont des tables fondamentales pour la création de notre base de données. Cela revient au fait que chaque article est écrit par un chercheur, et que chaque chercheur est attaché à une université ou un centre de recherche.

Concernant la table **Chercheur** :

- Nous introduisons les attributs « nom_ch, prenom_ch, » car c'est ce qui définit une personne en premier. Un chercheur est également caractérisé par un numéro orcid (code alphanumérique imposé par les grands éditeurs) d'où l'attribut « num_orcid ». L'« id_ch » quant à lui est indispensable pour caractériser chaque personne de façon unique pour éviter toute sorte de confusion. Il s'agit évidemment de la clé primaire de la table.

Concernant la table **Article** :

- Nous introduisons les attributs « id_art » et « titre_art » car c'est ce qui caractérise le mieux un article.

Concernant la table **Institution** :

- Pareil que les tables précédentes, l'identifiant sert toujours de clé primaire. Le type quant à lui désigne « université » ou « centre de recherche ».

Concernant la table **Livre** :

- Un livre est caractérisé par son titre et son éditeur. Il est également caractérisé par son numéro de série ou encore sa date de parution, mais cela nous importe peu.

Concernant la table **Conférence** :

- Une conférence est définie aussi par son titre et l'année de son déroulement. C'est ce qui servira essentiellement dans notre base de données.

Concernant la table **Revue scientifique** :

- Nous introduisons les attributs qui nous seront utiles notamment le titre et l'année de publication.

Concernant la table **Thèse** :

- Pareil, l'année de sa réalisation et l'université à laquelle elle se rattache sont principalement les données qui nous intéressent.

Concernant la table **Mots-clé** :

- Là aussi on introduit le « libellé_th » comme attribut ainsi que l'identifiant (clé primaire) si jamais on en a besoin lors des requêtes SQL concernant les rapports statistiques.

Concernant la table **statut_article** :

- Bien entendu, nous avons la possibilité d'introduire le statut de l'article comme attribut dans la table « Article », sauf que pour des raisons de pratique, nous avons décidé de lui créer une table à part. Cela revient au fait que le libellé lié à cette table sera constamment consulté et modifié puisque les articles du chercheur peuvent passer du statut « en révision » à « accepté ».

Concernant les tables **Date** :

- Ces tables ont pour but de « sauvegarder » les dates de publication de l'article dans une revue scientifique, livre, conférence et thèse.

Concernant la table **Pays** :

- On introduit l'attribut « ville » pour apporter un peu plus de détails.

Les cardinalités :

- **« +est affilié a »** : Chaque chercheur ou enseignant chercheur est attaché à une ou plusieurs universités ou centres de recherche. Par contre, une institution peut ne pas avoir de chercheurs (ou EC) ou en avoir plusieurs.
- **« affiliation principale »** : Chaque chercheur a une affiliation principale à une et une seule université.
- **« nationalité principale »** : Comme son nom l'indique, un chercheur peut avoir plusieurs nationalité, mais ne peut en avoir qu'une seule qui est principale.
- **« déroulement »** : Chaque conférence se déroule dans un pays précis. Un pays quant à lui peut accueillir plusieurs conférences.
- **« +écrit »** : Chaque chercheur (ou EC) peut éventuellement écrire un article, mais cela n'exclut pas le fait qu'un jeune chercheur n'ait pas encore rédigé son article. Inversement, chaque article admet au moins un auteur.
- **« +appartient à »** : Chaque thème englobe un ou plusieurs articles. Mais selon la base de données considérée, il se peut qu'un thème ne soit traité par aucun article. Par contre, un article appartiendra forcément à au moins un thème.
- **« +est publié dans »** : Chaque article, en fonction de la décision du comité de lecture, est publié dans une source. Ceci dit, un article rejeté n'est publié nulle part. Or chaque source contient au moins un article. Autrement, elle ne sera pas considérée comme étant une source.

Schéma relationnel :

Le modèle relationnel est déduit à partir du schéma UML, qui représente le modèle conceptuel. Il contient au moins autant de classes que le modèle conceptuel. Dans notre système, plusieurs associations vont se traduire en une classe supplémentaire.

INSTITUTIONS (id_inst : integer, nom_inst : varchar (50), type : varchar (50), #refpays (pays) : integer)

CHERCHEUR (id_ch : integer, nom_ch : varchar (50), prenom_ch : varchar (50), num_orcid : integer, mail : varchar (50), #refinstitutions (institutions) : integer, #refpays (pays) : integer)

AFFILIATION (#refchercheur (chercheur) : integer, #refinstitutions (institutions) : integer)

PAYS (id_p : integer, libelle_p : varchar (30), ville : varchar (50))

ARTICLE (id_art : integer, titre_art : varchar (100))

ECRIT (#refchercheur (chercheur) : integer, #refarticle (article) : integer)

MOTS_CLES (id_th : integer, libelle_th : varchar (30))

APPARTIENT (#refmots_cles (mots_cles) : integer, #refarticle (article) : integer)

CONFERENCE (id_c : integer, titre_c : varchar (100), annee_c : integer, #refpays (pays) : integer)

PUBLIE_1 (#refarticle (article) : integer, #refconference (conference) : integer, date : date, statut_article : varchar (50))

LIVRE (id_l : integer, titre_l : varchar (100), editeur : varchar (50))

PUBLIE_2 (#refarticle (article) : integer, #reflivre (livre) : integer, date : date, statut_article : varchar (50))

THESE (id_t : integer, universite : varchar (100), annee : integer)

PUBLIE_3 (#refarticle (article) : integer, #refthese (these) : integer, date : date, statut_article : varchar (50))

REVUE_SCIENTIFIQUE (id_rs : integer, titre_rs : varchar (100), annee_rs : integer)

PUBLIE_4 (#refarticle (article) : integer, #refrevue_scientifique (revue_scientifique) : integer, date : date, statut_article : varchar (50))

Requêtes SQL pour la création de base :

A l'aide du modèle relationnel, nous pouvons créer les tables de notre projet sous postgres. Le type << SERIAL>> sous le SGBD postgres permet d'auto-incrémenter la clé primaire dans la base de données, cela permet d'optimiser la rentrée des données.

CREATE TABLE PAYS (

id_p SERIAL NOT NULL,

libelle_p VARCHAR (30) NOT NULL ,

ville VARCHAR (50),

PRIMARY KEY (id_p)

);

CREATE TABLE INSTITUTIONS (

id_inst SERIAL NOT NULL,

nom_inst VARCHAR(50) NOT NULL,

type_inst VARCHAR(50) ,

i_pays SERIAL CONSTRAINT i_pays REFERENCES pays (id_p),

PRIMARY KEY (id_inst)

);

CREATE TABLE CHERCHEUR (

id_ch SERIAL NOT NULL,

nom_ch VARCHAR(50) NOT NULL,

prenom_ch VARCHAR(50) NOT NULL,

num_orcid INTEGER UNIQUE,

mail VARCHAR (50) UNIQUE,

c_pays SERIAL CONSTRAINT c_pays REFERENCES pays (id_p),

```
PRIMARY KEY (id_ch)
);
```

```
CREATE TABLE AFFILIATION (
    af_chercheur SERIAL CONSTRAINT af_chercheur REFERENCES chercheur(id_ch) ,
    af_institutions SERIAL CONSTRAINT af_institutions REFERENCES institutions(id_inst),
    PRIMARY KEY (af_chercheur, af_institutions)
);
```

```
CREATE TABLE ARTICLE (
    id_art SERIAL NOT NULL,
    titre_art VARCHAR (100) NOT NULL ,
    PRIMARY KEY (id_art)
);
```

```
CREATE TABLE ECRIT (
    e_chercheur SERIAL CONSTRAINT e_chercheur REFERENCES chercheur(id_ch) ,
    e_article SERIAL CONSTRAINT e_article REFERENCES article(id_art) ,
    PRIMARY KEY (e_chercheur, e_article)
);
```

```
CREATE TABLE MOTS_CLES (
    id_th SERIAL NOT NULL,
    libelle_th VARCHAR (30) NOT NULL,
    PRIMARY KEY (id_th)
);
```

CREATE TABLE APPARTIENT (

a_mots_cles SERIAL CONSTRAINT a_mots_cles REFERENCES mots_cles (id_th) ,

a_article SERIAL CONSTRAINT a_article REFERENCES article(id_art) ,

PRIMARY KEY (a_mots_cles, a_article)

);

CREATE TABLE CONFERENCE (

id_c SERIAL NOT NULL,

titre_c VARCHAR (100) NOT NULL,

annee_c INTEGER ,

c_pays SERIAL CONSTRAINT c_pays REFERENCES pays (id_p),

PRIMARY KEY (id_c)

);

CREATE TABLE PUBLIE_1 (

Date DATE,

Statut_article VARCHAR (50),

p1_article SERIAL CONSTRAINT p1_article REFERENCES article (id_art) ,

p_conference SERIAL CONSTRAINT p_conference REFERENCES conference(id_c) ,

PRIMARY KEY (p1_article, p_conference)

);

CREATE TABLE LIVRE (

id_l SERIAL NOT NULL,

titre_l VARCHAR (100) NOT NULL,

editeur VARCHAR (50),

PRIMARY KEY (id_l)

);

CREATE TABLE PUBLIE_2 (

Date DATE,

Statut_article VARCHAR (50),

p2_article SERIAL CONSTRAINT p2_article REFERENCES article (id_art) ,

p_livre SERIAL CONSTRAINT p_livre REFERENCES livre (id_l) ,

PRIMARY KEY (p2_article, p_livre)

);

CREATE TABLE THESE (

id_t SERIAL NOT NULL,

universite VARCHAR (100) NOT NULL,

annee INTEGER,

PRIMARY KEY (id_t)

);

CREATE TABLE PUBLIE_3 (

Date DATE,

Statut_article VARCHAR (50),

p3_article SERIAL CONSTRAINT p3_article REFERENCES article (id_art) ,

p_these SERIAL CONSTRAINT p_these REFERENCES these (id_t) ,

PRIMARY KEY (p3_article, p_these)

);

CREATE TABLE REVUE_SCIENTIFIQUE (

id_rs SERIAL NOT NULL,

titre_rs VARCHAR (100) NOT NULL,

annee_rs INTEGER,

PRIMARY KEY (id_rs)

);

CREATE TABLE PUBLIE_4 (

Date DATE,

Statut_article VARCHAR (50),

p4_article SERIAL CONSTRAINT p4_article REFERENCES article (id_art) ,

```
p_revue_scientifique SERIAL CONSTRAINT p_revue_scientifique REFERENCES
revue_scientifique (id_rs) ,

PRIMARY KEY (p4_article, p_revue_scientifique)

);
```

Répartition des tâches :

Etape 1 : Analyse / Conception

Saliou Diallo :

- Réalisation du code source.sql (Insertion des valeurs dans la base de données) ;

Cosnelle Djoumekoum :

- Réunion avec le client ;
- Réalisation du modèle relationnel (avec les justifications);
- Réalisation des requêtes SQL ;
- Ecriture des programmes php/html d'affichage ;
- Test et correction des programmes php ;
- Planning des réunions/répartition des tâches .

Safia Zendouz :

- Réunion avec le client ;
- Réalisation du schéma UML sur StarUML (avec les justifications) ;
- Rédaction de la présentation du projet + cahier des charges ;
- Ecriture des programmes php/html d'affichage ;
- Rédaction du rapport ;
- Planning des réunions/répartition des tâches .

Etape 2 : Mise en place de l'application WEB

La répartition des tâches se fera en fonction de l'élaboration des sous-programmes en PHP et de l'élaboration des interfaces d'interaction avec l'utilisateur en langage html. Toutefois, nous n'avons pas encore planifié qui s'occupera de quoi étant donné que nos capacités en PHP et en HTML sont relativement similaires.

Planning Horaire :

Nous organisons une réunion de 2h chaque Samedi et une réunion d'1h chaque Dimanche sur DISCORD, en plus des heures de travail en Projet BD figurant sur l'emploi du temps.

Documentation technique

http://serveur-etu.polytech-lille.fr/~cdjoumek/menu_projetBD_DDZ.html

Guide d'utilisation :

En cliquant sur le lien au-dessus, vous serez dirigé vers la page suivante :

20/12/2020 menu_projetBD_DDZ.html
http://serveur-etu.polytech-lille.fr/~cdjoumek/menu_projetBD_DDZ.html

Affichage des publications en fonction des types de publications

choisissez le type de la publication

Envoyer Annuler

Affichage des publications en fonction de leur état

choisissez l'état de la publication

Envoyer Annuler

Affichage des publications en fonction des mots-clés

choisissez un mot_clé

Envoyer Annuler

affichage des publications en fonction de leur année de publication

saisissez l'année de publication

file:///C:/Users/Sahar/Downloads/menu_projetBD_DDZ.html

1/2

file:///C:/Users/Sahar/Downloads/menu_projetBD_DDZ.html

2/2

20/12/2020 menu_projetBD_DDZ.html

Envoyer Annuler

modification de l'etat de la publication

saisissez le titre de l'article

saisissez le nouveau etat de l'article

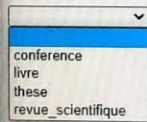
Envoyer Annuler

Les publications suivant leur type :

Cette page vous permet de choisir le mode d'affichage de vos articles en fonction du critère que vous voulez. Par exemple, si vous souhaitez consulter un article publié dans une revue scientifique, il vous suffit de cliquer sur la bannière et de sélectionner « revue scientifique ». Vous serez ensuite redirigé vers une page contenant tous les articles publiés dans une revue scientifique.

Affichage des publications en fonction des types de publications

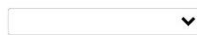
choisissez le type de la publication



A screenshot of a web form. At the top, it says 'choisissez le type de la publication'. Below this is a dropdown menu with a downward arrow. The dropdown is open, showing four options: 'conference', 'livre', 'these', and 'revue_scientifique'. The 'conference' option is highlighted in blue.

Affichage des publications en fonction des types de publications

choisissez le type de la publication



A screenshot of a web form. It shows a dropdown menu with a downward arrow. Below the dropdown are two buttons: 'Envoyer' and 'Annuler'.

Envoyer Annuler

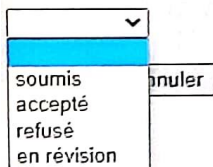
Vous serez ensuite redirigé vers une page contenant tous les articles publiés dans une revue scientifique.

Les publications selon leur état :

Si vous souhaitez consulter un article en fonction de son état, soit soumis, accepté, en révision ou encore refusé. Vous pouvez sélectionner l'état qui vous intéresse et cliquer ensuite sur « envoyer ». Vous serez redirigé vers une page contenant tous les articles sous l'état sélectionné.

Affichage des publications en fonction de leur état

choisissez l'état de la publication



A screenshot of a web form. It shows a dropdown menu with a downward arrow. The dropdown is open, showing four options: 'soumis', 'accepté', 'refusé', and 'en révision'. The 'soumis' option is highlighted in blue. To the right of the dropdown is a button labeled 'Annuler'.



Affichage des publications en fonction de leur état

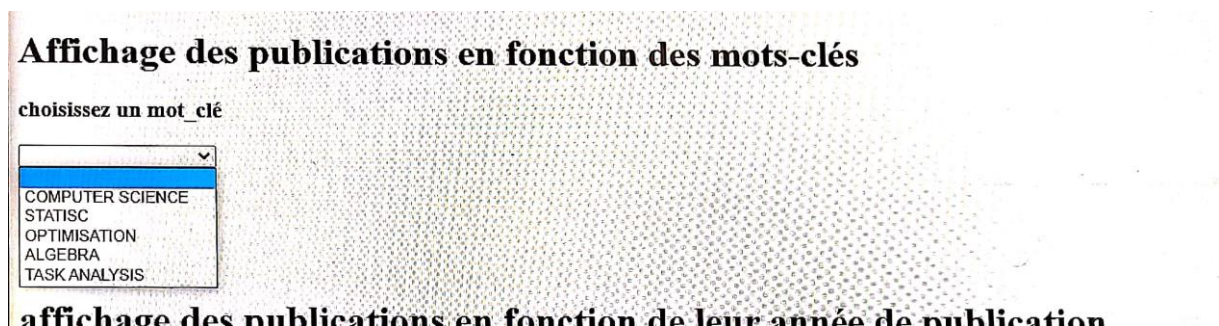
choisissez l'état de la publication

Envoyer

Annuler

Les publications selon des mot-clés :

Si vous désirez consulter les articles contenant un mot-clé précis, il vous suffit d'en choisir parmi ceux proposés dans la liste déroulante, et de cliquer sur envoyer. Ensuite, une page s'affiche et contient tous les articles relatifs au mot sélectionné.



Affichage des publications en fonction des mots-clés

choisissez un mot_clé

Envoyer

Annuler



Les publications selon leur année de publication

Consulter les articles publiés dans une certaine année est également possible. Pour ce faire, notre site propose à l'utilisateur de saisir l'année qui l'intéresse et de soumettre sa demande

en cliquant sur le bouton envoyer. Il sera dirigé vers une page contenant tous les articles publiés au cours de l'année sélectionnée.

affichage des publications en fonction de leur année de publication

saisissez l'année de publication

Envoyer

Annuler



affichage des publications en fonction de leur année de publication

saisissez l'année de publication

2020

Envoyer Annuler



affichage des publications en fonction de leur année de publication

saisissez l'année de publication

2020

Envoyer Annuler



Modification de l'état d'une publication :

Si jamais l'utilisateur souhaite mettre à jour l'état de son article, par exemple le faire passer de l'état « soumis » à l'état « accepté », il peut, grâce à l'interface que nous proposons, saisir le titre de l'article concerné et son nouvel état. La demande est soumise quand l'utilisateur clique sur le bouton envoyer. Il sera dirigé vers une page qui l'informe de la réussite de la modification.

modification de l'etat de la publication

saisissez le titre de l'article



saisissez le nouveau etat de l'article

modification de l'etat de la publication

saisissez le titre de l'article

saisissez le nouveau etat de l'article

NB : Il faut que l'orthographe de la saisie soit conforme avec les états proposés dans l'affichage selon l'état.

Organisation des fichiers :

L'élaboration de notre site fait intervenir un fichier html contenant l'intégralité de nos programmes notamment les programmes d'affichage et de modification d'un ou plusieurs articles. Chaque programme est donc associé à son propre fichier php. L'organisation peut se résumer dans ce schéma :

menu_projetBD_DDZ.html

Affichage selon le type

pubType.php

Affichage selon l'état

pubEtat.php

Affichage selon les mot-clés

pubMotCle.php

Affichage selon l'année

PubAnnee.php

Modification de l'état

PubModif.php

Requêtes difficiles :

Historique d'un article :

Afficher l'historique d'un article à travers une requête SQL a été pour nous d'une difficulté assez importante. L'objectif étant d'afficher à l'utilisateur qu'un article A a été soumis à une date t1 dans une conférence X, qu'il a été ensuite refusé (ou soumis en révision) à une date t2. Sauf qu'à une date t3, il a été modifié et soumis encore une fois dans une conférence Y qui a fini par l'accepter à une date t4. Nous avons posé la requête suivante :

```
"SELECT titre_art , titre_rs , universite , titre_c , titre_l, libelle_sart , PUBLIE1.date ,  
PUBLIE2.date , PUBLIE3.date , PUBLIE4.date FROM Article , POSSEDE ,  
REVUE_SCIENTIFIQUE , THESE , LIVRE , CONFERENCE , STATUT_ARTICLE , PUBLIE1 ,  
PUBLIE2 , PUBLIE3 , PUBLIE4 WHERE titre_rs IS NOT NULL AND universite IS NOT NULL  
AND titre_c IS NOT NULL AND titre_l IS NOT NULL AND Article.titre_art= « .$.titre » ORDER  
BY PUBLIE1.date ASC , PUBLIE2.date ASC , PUBLIE3.date ASC , PUBLIE4.date ASC ;"
```

Tout d'abord, nous sélectionnons les données non nulles qui nous intéressent : Le titre de l'article, le nom de la revue scientifique, l'université (si l'article est publié dans une thèse), le titre du livre et l'état de l'article. L'objectif étant d'afficher quelque chose de similaire à :

« L'article "L'IA" a été soumis à la conférence Technologie-Brésil le 12/01/2019. Il a été rejeté le 02/01/2019.

L'article "L'IA" a été soumis à la conférence des sciences de Genève le 10/03/2019. Il a été accepté le 17/04/2019. »

Vu que les données recherchées sont éparpillées dans plusieurs tables différentes, on se devait de faire un nombre important de jointure naturelle pour collecter les informations qui nous intéressent. Ensuite, on instaure la condition « *Article.titre_art= « .\$.titre »* » pour ne considérer que l'article saisi par l'utilisateur.

Affichage selon un mot-clé :

Nous avons opté pour la requête SQL suivante :

```
SELECT titre_art, titre_c, annee_c, nom_ch FROM article, appartient, mots_cles,  
publie_1, conference, chercheur, ecrit WHERE appartient.a_article = article.id_art AND  
appartient.a_mots_cles = mots_cles.id_th AND publie_1.p1_conference = conference.id_c AND  
publie_1.p1_article = article.id_art AND ecrit.e_article = article.id_art AND ecrit.e_chercheur =  
chercheur.id_ch AND libelle_th = '$Motscles' ;
```

Celle-ci ne prend en compte que les mot-clés figurants dans la table mots_cles. La conception d'un programme qui affiche les articles en fonction de n'importe quel mot figurant dans celui-ci nous a été assez délicat. C'est la raison pour laquelle nous avons préféré simplifier la requête SQL en enrichissant la table mots_cles par de nouveaux mots au fur et à mesure de l'ajout d'un nouvel article.

Bien entendu, cette requête a été reproduite 4 fois. Chaque fois est relative au type de publication. Autrement dit, la requête au-dessus s'intéresse aux articles publiés dans une conférence. Les autres requêtes dans le fichier php correspondent chacune à « Livre », « Thèse » et « Revue scientifique ».

Modification de l'état d'un article :

Nous sommes arrivés à la requête SQL suivante :

```
UPDATE STATUT_ARTICLE NATURAL JOIN POSSEDE NATURAL JOIN ARTICLE SET  
libelle_sart='Setat' WHERE titre_art= ' $titrea ');
```

'Setat' représente le nouvel état saisi par l'utilisateur '\$titrea' le titre de l'article concerné saisi par l'utilisateur. Nous ignorons ce qui handicape l'exécution d'une telle requête malgré plusieurs essais. C'est ce qui a rendu difficile de concevoir une autre requête qui puisse répondre à la même fonctionnalité attendue.

Bilan général :

Organisation du travail en groupe :

Points positifs : Ce projet nous a permis de comprendre la conception des sites WEB que nous rencontrons assez régulièrement mais aussi la corrélation entre le langage HTML et PHP avec la notion de Base de données. Par ailleurs, La majorité du groupe a su être impliquée et donner le meilleur de soi même malgré les difficultés rencontrées au niveau du php/html et parfois même SQL. Afin de bien comprendre ce qu'attend de nous le client, nous avons pu mener des discussions constructives et des échanges assez réguliers avec notre tutrice Mme Kessaci qui, d'ailleurs, n'a pas hésité à nous aider au niveau de l'organisation.

Points négatifs : Concernant la première étape du projet, nous n'avions pas trouvé beaucoup de difficultés au niveau technique. Ce qui a été le plus délicat par contre, c'était le modèle UML qu'on a dû modifier plusieurs fois puisque celui-ci est le pilier de toutes les réalisations qui suivent. La seconde quant à elle a été plus délicate dans le sens où nous devons écrire des programmes dans un langage encore inconnu, de les tester et d'essayer de les corriger malgré le peu de connaissance que nous avons.

En plus, nous nous retrouvons dans la plupart des cas à travailler en binôme bien que le projet et les tâches associées doivent être équiréparties entre 3 personnes.

Bilans personnels :

Safia Zendouz : « Le projet en soit a été très enrichissant et intéressant étant donné qu'il m'a initié au travail d'ingénieur qui m'attend. Sauf que j'aurai préféré être initiée comme il se doit au langage html/php car dans la majorité des cas je me retrouve dans l'incompréhension de certains programmes. D'ailleurs, la façon dont laquelle j'opère au niveau des écritures des programmes php était souvent mauvaise puisque je m'inspirais des programmes similaires déjà existant ce qui ne convient pas généralement au travail demandé. Je n'avais pas non plus le temps nécessaire pour m'auto-former puisque la deuxième partie du projet a coïncidé avec une semaine chargée de devoirs surveillés. Par ailleurs, je voudrais bien qu'on ait un second projet en BD une fois que tout le monde

soit plus à l'aise avec ces nouvelles notions. Je serai dans ce cas beaucoup plus satisfaite de mon rendu car j'estime qu'il sera forcément bien meilleur. »

Cosnelle Djoumekoum : *« J'ai trouvé le projet très intéressant. En effet, cela m'a permis de mieux comprendre le html et le php. Également, grâce à ce projet, j'ai pu mettre en pratique toutes les notions vues lors du cours de Base de données que j'ai également trouvé très intéressant. »*

REMERCIEMENT

Nous tenons à remercier Mme ME Kessaci de nous avoir encadré et apporté de l'aide quand cela a été nécessaire.

Nous remercions également notre école Polytech-Lille qui nous a permis de travailler dans de bonnes conditions en nous fournissant l'accès à distance aux machines de TP .