



# Rapport de stage

Création d'un wiki pour développer la base de connaissances techniques de l'EID Méditerranée

Stage en entreprise

Du 07/04/2015 au 10/08/2015



Présenté par :

**Hilmi BOUALLEGUE, développeur stagiaire**

Encadré Par :

**Dr. Amaury HABRARD, Responsable de Master WI**

**M. Réda TOUNSI, Maître de stage**

Année 2014-2015

# Sommaire

## Contenu

1.	Introduction :	6
2.	Contexte du stage:	7
2.1.	Présentation de l'établissement :	7
2.1.1.	Historique de L'EID Méditerranée :	7
2.1.2.	Activité de L'EID Méditerranée :	8
2.1.3.	Structure de L'EID Méditerranée :	8
2.2.	Présentation du projet :	9
2.2.1.	Recherche & Développement :	9
2.2.2.	La diffusion et le partage de données :	9
2.2.3.	Organisation et planification :	10
2.2.4.	Choix de l'outil et contraintes :	10
3.	Réalisation de l'outil :	11
3.1.	Conception :	12
3.1.1.	Outils méthodologiques :	12
3.1.1.1.	Rédaction du document d'exigences :	12
3.1.1.2.	Diagramme de cas d'utilisation :	13
3.1.1.3.	Diagramme de séquences :	14
3.1.1.4.	Le cahier de recette des tests :	14
3.1.2.	Définition des besoins spécifiques :	15
3.1.2.1.	Administration & gestion des utilisateurs :	15
3.1.2.2.	Navigation et consultation du DT-Wiki :	17
3.1.2.3.	Ajout de contenu dans le système (DT-Wiki) :	20
3.1.2.4.	Indexation des articles :	24
3.2.	Développement :	26
3.2.1.	Environnement & outils de développement :	26
3.2.2.	Développement des besoins spécifiques:	26
3.2.2.1.	L'Installation du Framework Media Wiki :	26
3.2.2.2.	Développement de la gestion des utilisateurs :	27
3.2.2.3.	Développement de l'architecture du système :	29
3.2.2.4.	Développement du Cycle de vie de l'article :	30
3.2.2.5.	Développement du système d'indexation des articles :	33

3.3.	Test et déploiement : .....	34
3.3.1.	Tests de développement : .....	35
3.3.1.1.	Tests unitaires : .....	35
3.3.1.2.	Tests de composants : .....	35
3.3.1.3.	Tests système : .....	35
3.3.2.	Tests de version : .....	35
3.3.3.	Tests utilisateur : .....	35
3.4.	Formation et perspective : .....	35
4.	Conclusion : .....	36
5.	Annexes et bibliographie : .....	38
5.1.	Bibliographie : .....	38
5.2.	Webographie : .....	38
5.3.	Liste des annexes : .....	38

# Liste des figures

Figure 1 : Logo EID Méditerranée.....	7
Figure 2: La zone d'action de l'EID Méditerranée .....	7
Figure 3 : Organigramme général de l'EID Méditerranée. ....	8
Figure 4 : Schéma de la méthode "agile" (2).....	11
Figure 5 : Diagramme de cas d'utilisation globale .....	13
Figure 6 : Architecture de création d'un utilisateur .....	17
Figure 7 : Architecture de consultation du DT-Wiki.....	18
Figure 8 : Extraction de maquette fournie en Annexe 6 Maquette de page d'accueil .....	19
Figure 9 : Maquette du portail thématique (annexe 6) .....	19
Figure 10 : Architecture pour afficher le dernier article .....	20
Figure 11 : Architecture de recherche d'un article .....	20
Figure 12 : Cycle de vie d'un article.....	21
Figure 13 : Architecture de création d'un article .....	22
Figure 14 : Architecture de validation d'un article.....	22
Figure 15 : Architecture de modification d'un article .....	23
Figure 16 : Architecture d'analyseur du système (DT-Wiki).....	23
Figure 17 : Structure d'indexation des articles.....	24
Figure 18 : Architecture de création d'une catégorie .....	25
Figure 19 : Architecture de création d'une sous-catégorie.....	25
Figure 20 : Architecture de demande de création d'une catégorie .....	26
Figure 21 : Interface d'accès.....	27
Figure 22 : Page de création d'un utilisateur .....	28
Figure 23 : Changer le mot de passe .....	28
Figure 24 : Page d'accueil du DT-Wiki-Dev.....	29
Figure 25 : Exemple de recherche d'un article.....	30
Figure 26 : Exemple d'un article créé .....	31
Figure 27 : Exemple de validation d'un article par l'éditeur .....	32
Figure 28 : Page de création d'une catégorie ou d'une sous-catégorie.....	33
Figure 29 : Page de demande de création d'une catégorie.....	34

# Liste des tableaux

Tableau 1 : Extrait du cahier de recette fourni en Annexe 2 .....	15
Tableau 2 : Extrait des Droits utilisateur-EID-MED Wiki fourni en Annexe 5.....	16

# Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de stage par des remerciements à ceux qui m'ont beaucoup appris au cours de ce stage et à ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable.

Aussi, je remercie Monsieur Réda TOUNSI, mon maître de stage qui m'a formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie. Je remercie Madame Myriam CROS pour ses conseils et son expertise très utiles pour orienter mon travail. Je remercie aussi, Madame Nathalie BARRAS Nathalie pour ses conseils et son aide dans la réalisation du stage.

Je remercie également l'ensemble des agents de l'EID Méditerranée pour leur accueil et les conseils qu'ils ont pu me prodiguer au cours de ces mois du stage.

Enfin, je remercie les enseignants du Master *Web Intelligence* à l'Université Jean Monnet, à Saint-Etienne, pour la qualité de leurs conseils, des outils pédagogiques et techniques fournis. Ces enseignements m'ont déjà été beaucoup utiles dans la réalisation de ce stage.

# 1. Introduction :

L'EID Méditerranée (Entente Interdépartementale pour la Démoustication du littoral Méditerranéen) dont le siège se situe à Montpellier, établissement public de 160 agents, œuvre depuis plus de 50 ans à la conservation et la gestion des zones humides méditerranéennes en contrôlant les nuisances liées aux moustiques.

La Direction générale de l'EID Méditerranée, regroupe la Direction administrative, la Direction technique (avec deux laboratoires spécialisés), et la Direction de la Logistique regroupant notamment le service Système d'Information (SI) et celui du parc de véhicules et matériels.

Au sein de la Direction technique (DT) où j'ai fait mon stage (intégré au Pôle Etudes et prospective), il y a une forte activité de Recherche et Développement qui génère une quantité énorme des données dans des secteurs différents (entomologie, botanique, biocides...). Cette quantité importante de données génère des problèmes de diffusion et d'échange des connaissances au sein de la Direction.

Pour répondre au besoin exprimé par la DT (valoriser et faciliter la diffusion des connaissances produites), le service SI propose une solution collaborative type « WIKI » qui nécessite la recherche d'un développeur stagiaire chargé de l'élaboration de l'application en respectant quelques contraintes : rester dans le périmètre de la Direction technique, respecter la culture de la fonction publique territoriale et de la Charte de l'EID Méd.

J'ai donc réalisé des tâches de conception (documents du système, conception de la structure du système, conception de la structure des utilisateurs et de leurs droits d'accès), de réalisation et de développement du système (création de la structure globale du système ou squelette du système), de création de la structure globale des utilisateurs (les droits d'accès), et enfin de création et développement des fonctionnalités du système.

Ce rapport de stage présente la mission que j'ai effectuée pendant ces quatre mois.

## 2. Contexte du stage:

### 2.1. Présentation de l'établissement :



Figure 1 : Logo EID Méditerranée

#### 2.1.1. Historique de L'EID Méditerranée :

L'EID Méditerranée est un établissement interdépartemental créé en 1958, afin de réaliser la démoustication du littoral méditerranéen. En 1963, les conseils généraux de l'Aude et des Pyrénées-Orientales rejoignent les Conseils généraux des départements de l'Hérault, du Gard et des Bouches-du-Rhône. Cette même année, l'Etat crée la Mission interministérielle d'aménagement touristique du Languedoc-Roussillon, dit "Mission Racine", qui fixe la démoustication comme préalable incontournable à toute démarche d'aménagement du littoral. Aujourd'hui, l'établissement intervient sur toute la frange littorale méditerranéenne française (hors Corse), comme le montre la Figure 2.



Figure 2: La zone d'action de l'EID Méditerranée



## 2.1.2. Activité de L'EID Méditerranée :

Les principales missions de l'EID Méditerranée sont :

- Le contrôle des nuisances causées par les moustiques des milieux naturels et urbains. Ce contrôle a pour objectif non pas d'éradiquer l'ensemble des insectes piqueurs, mais à maintenir la gêne due aux moustiques à un seuil jugé tolérable par la population.
- La préservation des zones humides littorales (conservation du littoral). Elle participe à des actions importantes de conservation et de mise en valeur écologique des milieux naturels (restauration des cordons dunaires et suivi des systèmes littoraux)...

## 2.1.3. Structure de L'EID Méditerranée :

L'EID Méditerranée a un siège à Montpellier qui regroupe les activités de Recherche et Développement (DT), de gestion administrative et logistique en appui à la mission centrale (contrôle de la nuisance) qui est confiée aux agences opérationnelles. Ces 9 agences opérationnelles sont réparties, sur l'ensemble de la zone d'action et sont localisées au plus près des sites qui génèrent les moustiques.

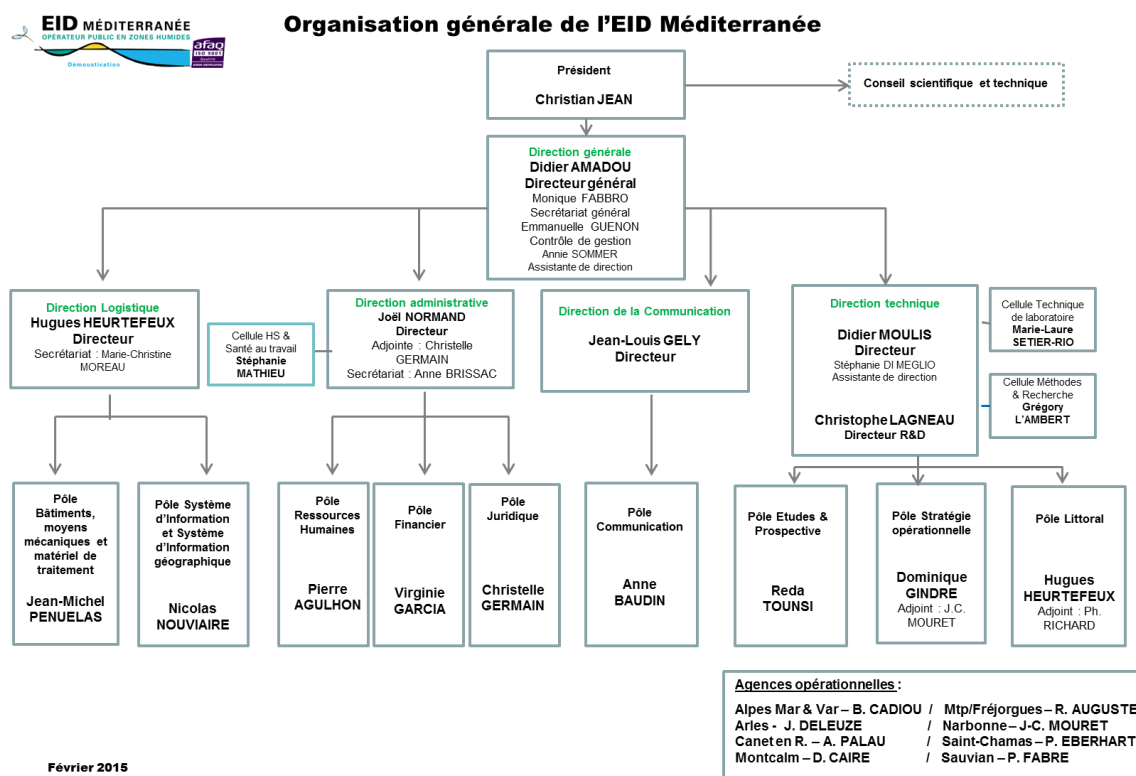


Figure 3 : Organigramme général de l'EID Méditerranée.

## 2.2. Présentation du projet :

### 2.2.1. Recherche & Développement :

La Direction technique de l'EID Méditerranée, contient une forte activité de Recherche et de Développement qui produit une grande quantité des données dans des secteurs différents

Cette Direction dispose de deux laboratoires permettant d'assurer le support scientifique nécessaire à la mise en œuvre d'une stratégie de lutte intégrée respectueuse de l'environnement. Les objectifs et les compétences de la Direction technique sont les suivants :

- ***Développer les connaissances sur la taxonomie, la biologie et l'écologie des espèces cibles et non cibles et sur les milieux propices à leur développement, notamment par la mise en place de réseaux de veille entomologique basés sur l'utilisation de différentes techniques de piégeage ;***
- ***Optimiser globalement les techniques de contrôle et l'ensemble des méthodes et outils nécessaires à l'accomplissement des opérations de démoustication et de LAV en proposant des alternatives, et notamment des biocides plus efficaces et les moins préjudiciables pour la santé humaine et l'environnement ;***
- ***Développer et mettre en œuvre des méthodes et des procédures d'évaluation permanente de l'efficacité des opérations en termes de réduction de la nuisance afin de les améliorer, d'en réduire les effets éventuels et, dans la mesure du possible, le coût ;***
- ***Développer et mettre en œuvre des méthodes et des procédures d'évaluation permanente des effets non intentionnels de ces opérations sur l'environnement et de la biodiversité des milieux démoustiqués afin de mieux les préserver. »<sup>1</sup>***

### 2.2.2. La diffusion et le partage de données :

La difficulté liée à la diffusion et au partage des grandes quantités des données produites par la Direction technique est comparable au phénomène de la malédiction de la dimensionnalité.

Les réunions de présentations organisées au sein de la Direction ne sont pas suffisantes et les outils de recherche documentaires présents pas performants : certains résultats de projets sont connus par l'équipe voisine deux ans après leur production.

### 2.2.3.Organisation et planification :

L'équipe du projet contient 7 personnes :

- Un chef de projet
- Un chargé de projet SI.
- Un chargé d'étude « valorisation des connaissances ».
- Un responsable applications métiers et réseaux.
- Deux futurs utilisateurs.
- Un développeur stagiaire.

L'équipe du projet faisait une réunion par semaine, afin de suivre l'avancement du projet et pour valider les étapes importantes avec le Directeur technique.

### 2.2.4.Choix de l'outil et contraintes :

Le service SI (Système d'information) propose une solution Wiki, afin de valoriser et de diffuser les connaissances en interne, selon des contraintes :

- ✓ La gestion des utilisateurs est plus fine.
- ✓ La facilité de navigation du système (Structure apriori).
- ✓ Création des catégories a priori pour indexer les articles et faciliter la recherche.
- ✓ Faciliter la création et l'édition des articles.

Le service SI n'a pas de compétence particulière en développement. Pour cela, la direction technique a cherché un stagiaire capable de développer et mettre en œuvre en totalité un logiciel participatif de type « wiki ».

Cette mission contient beaucoup d'aspects qui m'ont attirés : l'occasion de mieux pratiquer les technologies web dans un projet réel et professionnel (avec une large autonomie pour le développement) et de découvrir la gestion de projet dans un environnement de travail.

### 3. Réalisation de l'outil :

Le développement du système (DT-Wiki) est basé sur les méthodes « Agile » pour assurer la satisfaction des besoins définis avec l'équipe projet.

Ces méthodes sont plus souples que l'approche traditionnelle qui consiste pour le futur utilisateur à détailler et valider les besoins au début du projet. En effet celle-ci peut poser un problème de déphasage entre les besoins et l'application réalisée ou retarder le projet en cas de changements des spécifications en cours de projet.

L'approche « Agile » permet de réduire ces problèmes par l'implication de l'équipe projet du début jusqu'à la livraison de l'application, par l'adoption d'un processus de développement :

- Itératif
- Incrémental
- Adaptatif.

« *Principes sous-jacents au manifeste de méthode «Agile» :*

*Nous suivons ces principes:*

- *Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.*
- *Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.*
- *Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.»<sup>2</sup>.*



Figure 4 : Schéma de la méthode "agile" (2)

Le système (DT-Wiki) est développé suivant des cycles (itérations, sprints) pour s'adapter aux besoins (évolution des besoins, modification de documentation, modification de la conception, ajout des nouvelles fonctionnalités, test et validation) ainsi on a sélectionné une portion des exigences à réaliser dans une portion de temps.

Ensuite, à la fin de chacune de ces itérations (sprints) le produit est montré et évalué par l'équipe projet avant les prochaines itérations.

### 3.1. Conception :

#### 3.1.1. Outils méthodologiques :

J'ai utilisé 4 types d'outils méthodologiques afin de concevoir le système :

- Document d'exigences.
- Diagramme des cas d'utilisations.
- Diagramme de séquences.
- Cahier de recette des tests.

##### 3.1.1.1. Rédaction du document d'exigences :

Le document d'exigences rédigé (Cf. Annexe 1 : Document d'exigence) suit la norme standard IEEE-1998.

**« Ce guide propose un cadre pour l'élaboration d'un ensemble d'exigences, afin de satisfaire à un besoin exprimé. Dans ce document, cet ensemble d'exigences est dénommé Spécification d'exigences de système » (Guide de l'IEEE pour la Spécification d'Exigences de Système)».**

Ce document, nous a permis de mieux comprendre les besoins de l'utilisateur, de lister et représenter les besoins fonctionnels (création d'article, création d'une catégorie, validation d'un article, création d'un compte utilisateur...etc.). Il nous a permis aussi de représenter les besoins non fonctionnels du système (pas d'accès sans autorisation, utilisation du réseau local seulement...etc.).

Il m'a permis aussi de mieux définir les contraintes de développement du système, telles que les contraintes de conception (langage de programmation PHP 5, utilisation de modelé MVC, la base des données de type PostgreSQL, le système sera de type client/serveur...etc.).

### 3.1.1.2. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation (Cf. Annexe 2 : Diagramme de cas d'utilisation-EID-MED Wiki) est un diagramme utilisé pour traduire visuellement les exigences listées dans le document d'exigences.

«

- *Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.*
- *Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés.*
- *Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.*
- *Ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.*
- *Ils permettent de classer les acteurs et structurer les objectifs du système. »<sup>3</sup>*

Ce document nous a permis de mieux visualiser les fonctionnalités et leurs interactions avec les utilisateurs voir par exemple le diagramme de cas d'utilisation globale :

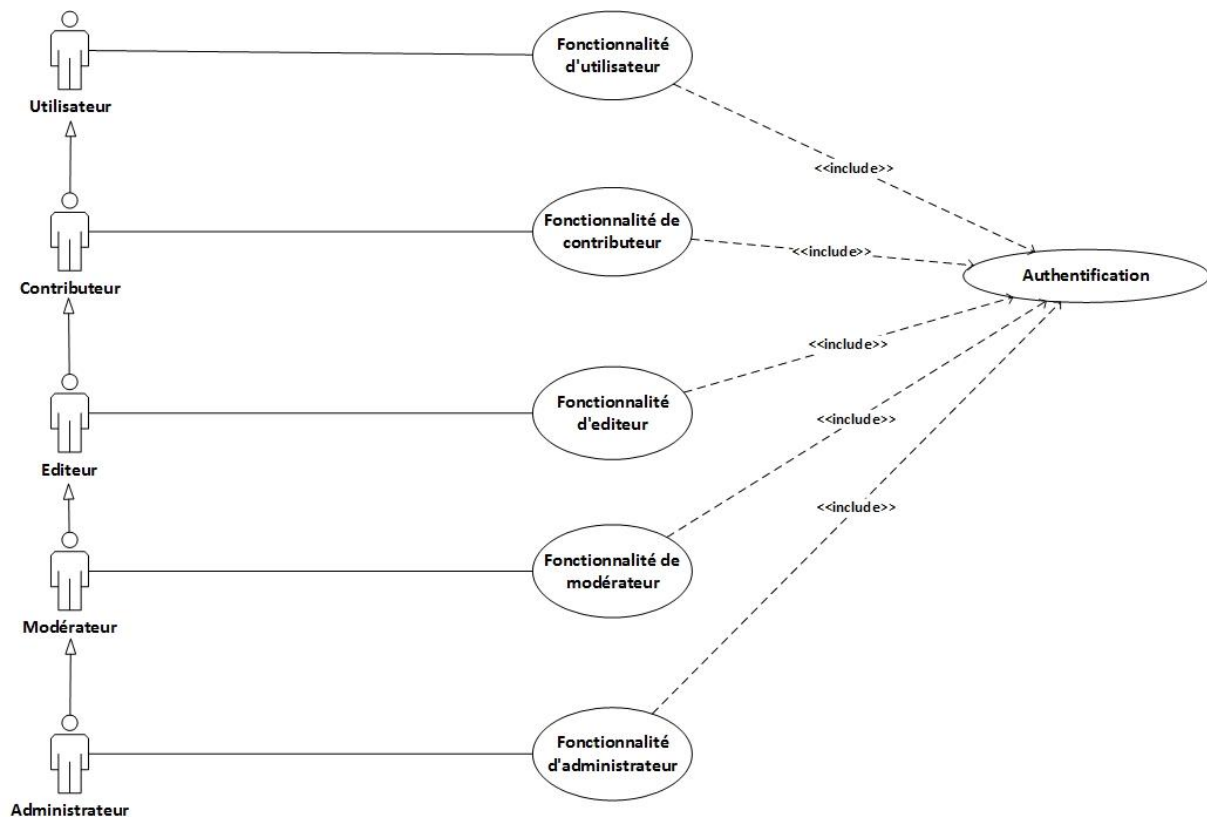


Figure 5 : Diagramme de cas d'utilisation globale

### 3.1.1.3. Diagramme de séquences :

Le diagramme de séquences (Cf. Annexe 3 : Diagramme de séquences-EID-MED Wiki) est un diagramme utilisé pour représenter graphiquement un scénario d'utilisation du système et les messages échangés entre les utilisateurs selon un ordre chronologique.

«

- **Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.**
- **Les diagrammes de séquences peuvent servir à illustrer un cas d'utilisation.**
- **L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule "de haut en bas" de cet axe.**
- **La disposition des objets sur l'axe horizontal n'a pas de conséquence pour la sémantique du diagramme. »<sup>4</sup>**

Ce document, nous a permis de mieux visualiser la communication entre les composants du système et l'utilisateur selon un ordre chronologique.

### 3.1.1.4. Le cahier de recette des tests :

Le jeu de test (Cf. Annexe 4 : Jeu de test-EID-MED Wiki) est un document utilisé pour tester les fonctionnalités (rédiger des tests des fonctionnalités du système) afin de vérifier que le système marche correctement, et découvrir s'il y a des défauts avant de mettre en service.

Ce document nous a permis de mieux contrôler les résultats du test afin de trouver les erreurs et les anomalies, ci-dessous l'exemple des tests de la fonctionnalité « Ouvrir une session » :

- T11 : Cliquer sur l'option « Se connecter ».
- T12 : Saisir le login.
- T13 : Saisir le mot de passe.
- T14 : S'authentifier en cliquant sur le bouton « se connecter ».

Nom du Test	Exigence associée	Action	Condition	Résultat attendu	Résultat observé	Conséquences
T11	3.1.2	Cliquer sur l'option « Se connecter ».	Connexion internet établie et un message d'erreur est affiché demande de connexion pour accéder au système.	Formulaire de login est affiché		

T12	3.1.2	Saisir « »	Formulaire de saisie affiché	<b>Erreur de connexion</b> Vous n'avez pas saisi un nom d'utilisateur valide.		
T12	3.1.2	Saisir « trara »	Formulaire de saisie affiché	Il n'y a pas de contributeur avec le nom « trara ». Veuillez vérifier l'orthographe.		
T13	3.1.2	Saisir « »	Formulaire de saisie affiché	<b>Erreur de connexion</b> Vous n'avez pas entré de mot de passe. Veuillez essayer à nouveau.		
T14	3.1.2	Cliquer sur le bouton « se connecter »	Les champs sont remplis avec des données correctes	Accès à la page d'accueil		
T14	3.1.2	Cliquer sur le bouton « se connecter »	Les champs ne sont pas remplis correctement.	Il n'y a pas de contributeur avec le nom « .... ». Veuillez vérifier l'orthographe.		
T15	3.1.1.4	Clique sur un bouton autre que validation	Les champs ne sont pas remplis correctement.	Rien		Ce n'est pas accepté

Tableau 1 : Extrait du cahier de recette fourni en Annexe 2

### 3.1.2. Définition des besoins spécifiques :

Avec l'équipe projet j'ai défini les besoins spécifiques et j'ai proposé des idées pour satisfaire ces besoins, ceux-ci sont regroupés en 4 domaines :

- Administration & gestion des utilisateurs.
- Navigation et consultation du DT-Wiki.
- Ajout de contenu dans le DT-Wiki.
- Indexation des articles.

#### 3.1.2.1. Administration & gestion des utilisateurs :

##### 3.1.2.1.1. Gestion des droits d'utilisateurs :

J'ai commencé par la conception de la structure des droits d'utilisateurs, afin de mieux comprendre les types des utilisateurs du système (DT-Wiki) et leurs fonctionnalités.

Les droits d'utilisateurs dont un extrait est présenté en Tableau 2 (Version complète donnée en Annexe 5 : Droits d'utilisateur-EID-MED Wiki) sont listés dans un document utilisé pour mieux visualiser la structure des utilisateurs avec leurs fonctionnalités (avec leurs activités). Dans cette itération de développement, les rôles de notre système (DT-Wiki) sont :

- **Administrateur :**

L'administrateur a accès à toutes les fonctionnalités du wiki, il crée les utilisateurs et définit leurs droits d'accès.



- **Modérateur :**

Le modérateur anime le système, rappelle les règles, crée les catégories, protège ou supprime des pages ou des articles, remonte à la hiérarchie et à l'administrateur les décisions à prendre pour améliorer le fonctionnement du wiki.

- **Editeur :**

L'éditeur vérifie les articles rédigés : corrections si besoin, validation ou transfert pour validation vers un autre éditeur plus qualifié.

- **Contributeur :**

Le contributeur crée des articles et peut demander l'ajout de nouvelles catégories.

- **Utilisateur :**

L'utilisateur peut consulter les articles et faire des commentaires sur le contenu des articles.

Droits	Admin	Modérateur	Editeur	Contributeur	Utilisateur	Description	Risques	Commentaires
Lire un article	x	X	x	x	x	Permet d'accéder aux articles rédigés	Exportation d'articles non validés	Accès autorisé par un LOGIN/Mot de passe
Créer un article.	x	X	X	X		Permet de rédiger des pages contenant des informations.		
Créer une catégorie.	x	X				Créer une catégorie pour utiliser dans l'indexation des articles.	Création de nouvelles catégories (multiplication de catégories redondantes ou non pénitentes)	
Créer un utilisateur.	x					Création des utilisateurs qui permet d'accès au contenu du wiki.	Création des faux profils.	
Création de partie commentaire pour chaque article.	x	X	X	X	x	Discussion et échange des idées autour d'un article dans une partie séparée du contenu de l'article.	Source de polémiques stériles	
Demande de création d'une catégorie.	x	X	X	X		Permet de demander la création d'une catégorie supplémentaire pour améliorer l'indexation des articles	Surcharge du travail de modération	
Modifier le contenu des pages spéciales (page d'accueil, page du portail).	x	X				Permet de modifier la structure du Wiki	Destruction de la structure générale du Wiki. Risque de perdre définitivement des informations.	L'accès libre à tout (en lecture et modification) fait partie intrinsèquement de la philosophie de l'outil
Modifier le contenu de tous les articles.	x	X	X			Permet d'enrichir ou corriger le contenu d'un article	Perte d'information pertinente. Source d'erreur.	L'accès libre à tout (en lecture et modification) fait partie intrinsèquement de la philosophie de l'outil
Modifier le contenu de des articles propres à l'utilisateur.	x	X	X	x				
Protéger des pages.	x	X				Permet d'interdire la modification du contenu d'une page	Empêche l'amélioration du contenu d'un article (y compris mise à jour)	L'accès libre à tout (en lecture et modification) fait partie intrinsèquement de la philosophie de l'outil

Tableau 2 : Extrait des Droits utilisateur-EID-MED Wiki fourni en Annexe 5

#### 3.1.2.1.2. L'architecture de création d'un utilisateur :

J'ai conçu une architecture pour faciliter la création des utilisateurs et la définition de leurs droits d'accès par l'administrateur.

L'administrateur choisit l'option « Créer un utilisateur ». Ensuite, il remplit les informations du nouvel utilisateur (login, mot de passe, email, nom réel), puis il valide.

Une interface graphique informe l'administrateur que l'utilisateur est créé avec succès, par défaut l'utilisateur a un rôle « Utilisateur ».

Cependant, l'administrateur peut attribuer un rôle différent à l'utilisateur en utilisant l'option « Gestion des droits d'utilisateur ».

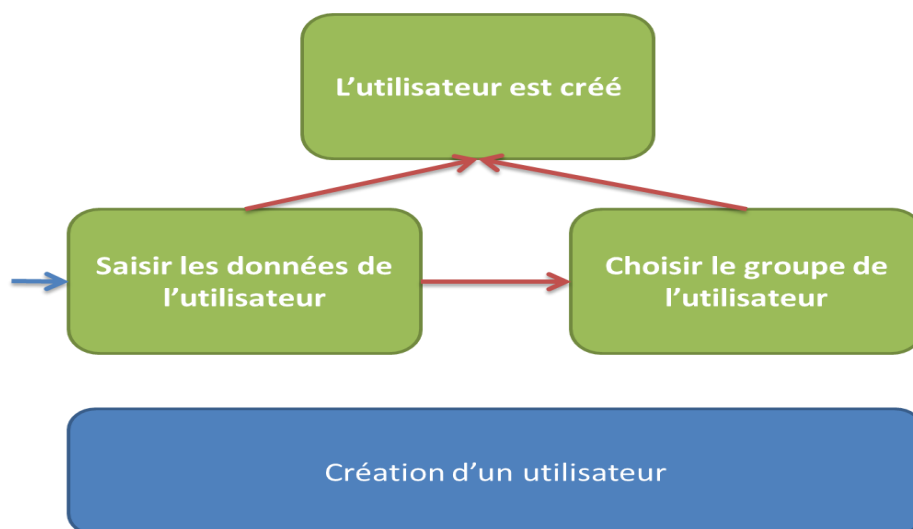


Figure 6 : Architecture de création d'un utilisateur

#### 3.1.2.2. Navigation et consultation du DT-Wiki :

##### 3.1.2.2.1. L'architecture du système (DT-Wiki) :

J'ai conçu l'architecture du wiki pour faciliter la navigation des utilisateurs et l'accès aux articles.

La navigation est basée sur des thématiques d'étude :

- Santé publique.
- Démoustication opérationnelle.
- Recherche & Développement.
- Littoral.
- Laboratoire.
- Réseau scientifique.

Chaque thématique se présente sous forme d'un portail :

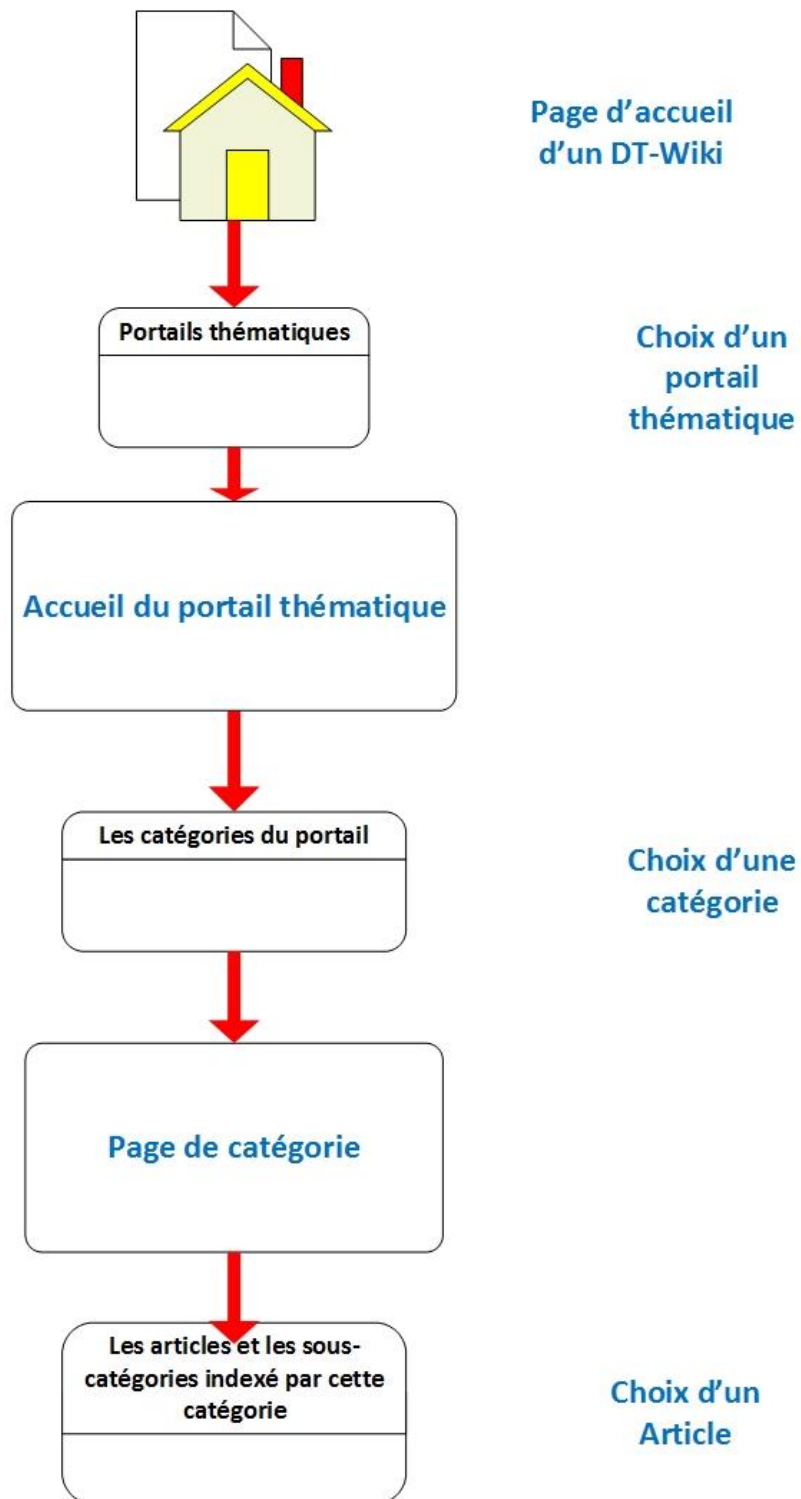


Figure 7 : Architecture de consultation du DT-Wiki

Les maquettes du système (Cf. Annexe 6 : Maquettes-EID-MED Wiki) sont lister dans ce documents.

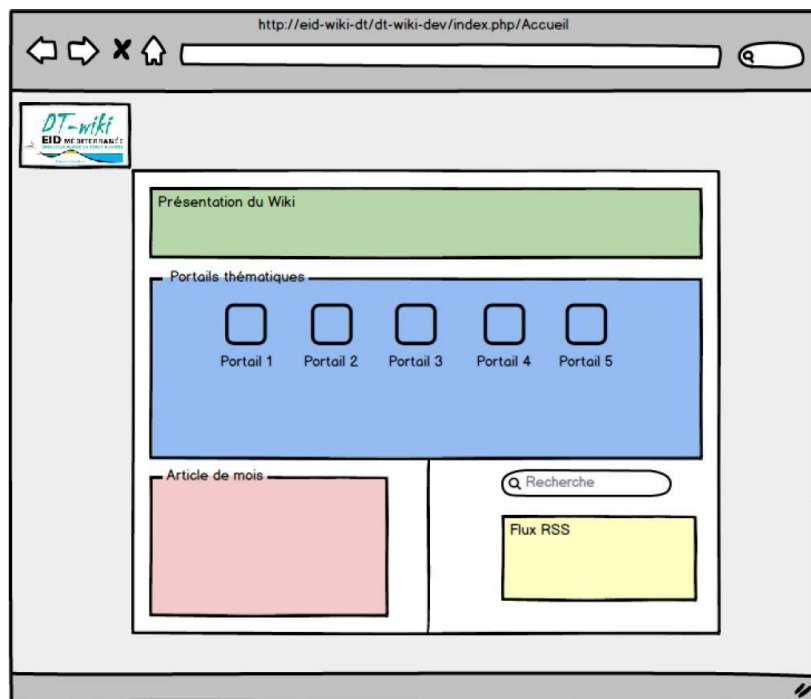


Figure 8 : Extraction de maquette fournie en Annexe 6 Maquette de page d'accueil

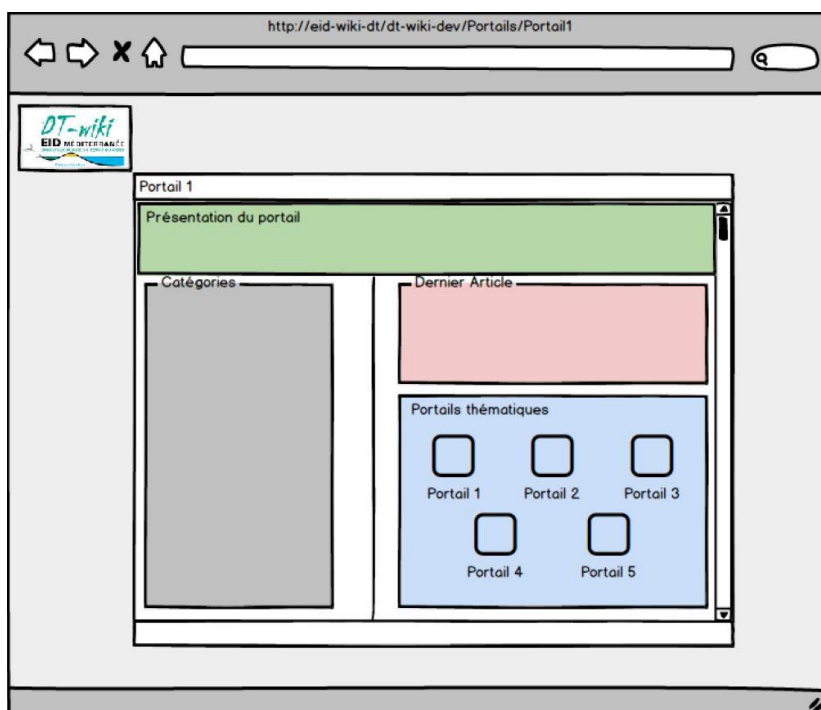


Figure 9 : Maquette du portail thématique (annexe 6)

### 3.1.2.2.2. L'architecture d'afficher le dernier article :

J'ai conçu pour cette fonctionnalité une architecture simple, afin d'afficher le dernier article d'un portail thématique dans la page d'accueil de ce portail.

Le système parcourt la base de données pour vérifier les derniers articles des catégories associées au portail thématique, puis sélectionne l'article le plus récent.

L'article est affiché dans une partie nommée « Dernier article ».

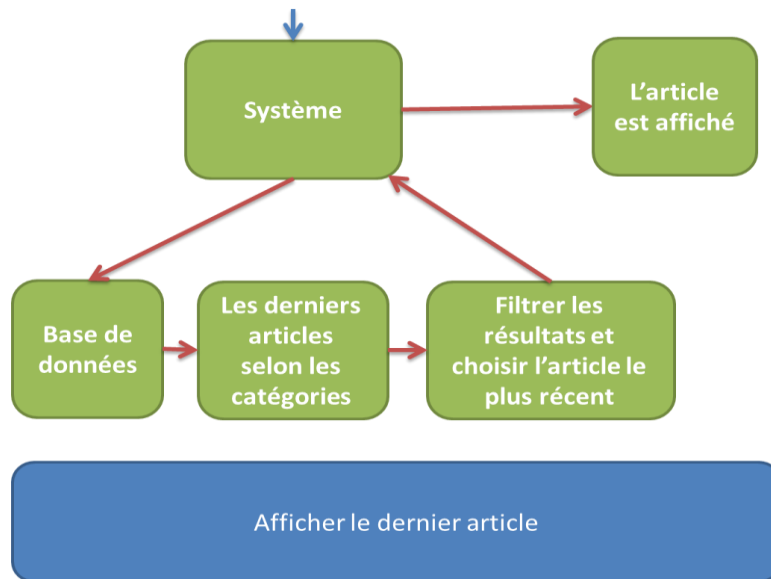


Figure 10 : Architecture pour afficher le dernier article

#### 3.1.2.2.3. Rechercher un article :

J'ai adapté la solution déjà existante dans le Framework media wiki pour la recherche des pages, afin que les articles apparaissent en priorité par rapport aux autres types des pages.

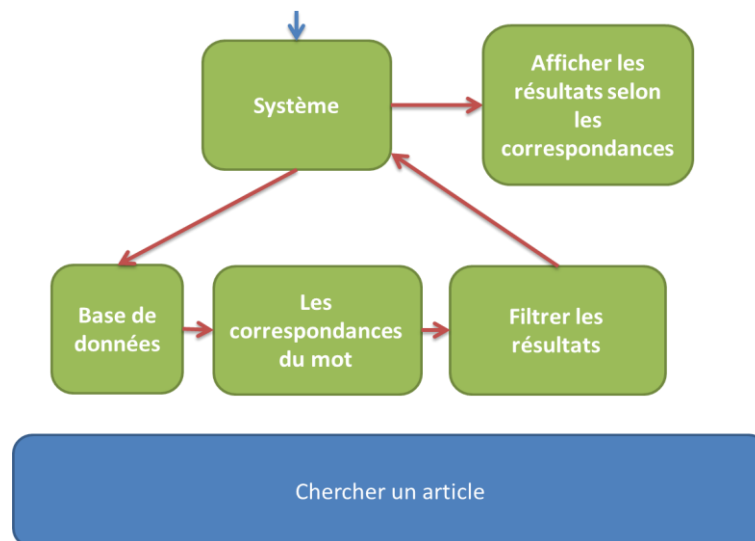


Figure 11 : Architecture de recherche d'un article

#### 3.1.2.3. Ajout de contenu dans le système (DT-Wiki) :

Avec l'équipe projet, nous avons défini le cycle de vie d'un article, j'ai ensuite conçu les fonctionnalités adaptées à ce cycle de vie.

# Cycle de vie d'un article

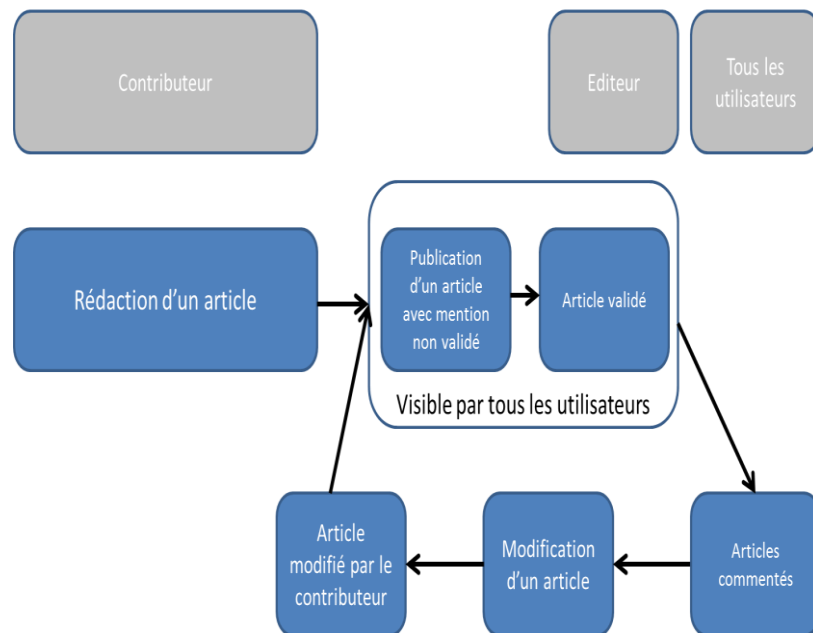


Figure 12 : Cycle de vie d'un article

## 3.1.2.3.1. Création d'un article :

J'ai essayé de concevoir une architecture pour guider et faciliter la création d'un article en séparant les étapes de création pour minimiser les fautes de saisie et le nombre des informations dans la même interface graphique.

L'utilisateur clique sur le bouton « Créer un article », une interface graphique apparaît, elle permet de choisir le modèle d'article à créer.

Après la validation du choix du modèle, une autre interface graphique apparaît pour saisir le nom de l'article.

Après la validation du nom de l'article, l'utilisateur accède à l'interface graphique qui contient l'éditeur de texte.

Ensuite, l'utilisateur peut écrire son article, il enregistre son article, puis accède à une autre interface graphique afin de choisir les catégories d'indexation de cet article.

Enfin, l'utilisateur valide la création de l'article.

Après la création d'un article, le système ajoute par défaut une étiquette en haut de la page pour informer que l'article est publié mais pas encore validé par un éditeur. Le système envoie un email au contributeur qui a rédigé l'article pour indiquer que l'article est créé avec succès.

En parallèle, le système envoie un email à l'éditeur afin d'informer qu'il y a un nouvel article créé à valider.

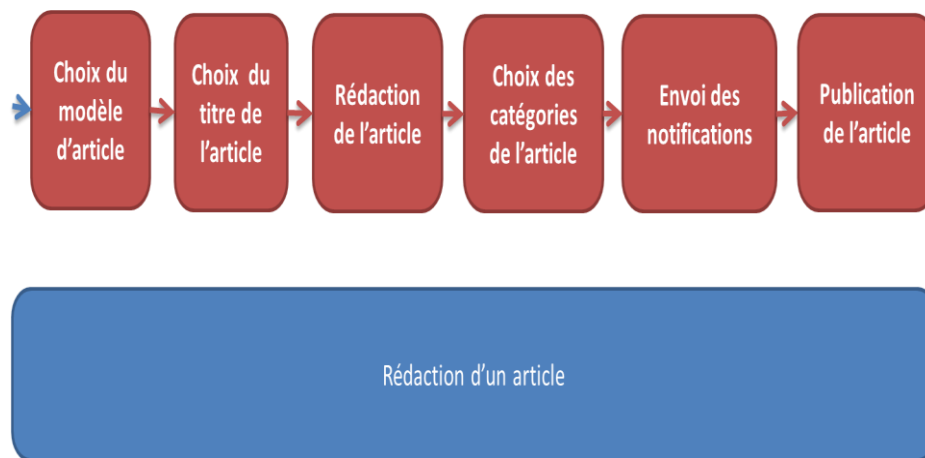


Figure 13 : Architecture de création d'un article

#### 3.1.2.3.2. L'architecture de validation d'un article :

J'ai essayé de concevoir une architecture simple pour la validation d'un article.

L'éditeur reçoit un email qui l'informe qu'un nouvel article est créé. Un lien dans l'email permet à l'éditeur d'accéder à cet article.

Dans la page de l'article, l'éditeur peut consulter le contenu et faire des modifications si nécessaire. Pour valider l'article, l'éditeur clique sur un bouton dans l'étiquette en haut de page.

Enfin, le système vérifie le rôle de l'utilisateur, si c'est un éditeur, l'article est validé. Un email est envoyé au contributeur pour l'informer que son article est validé par un éditeur.

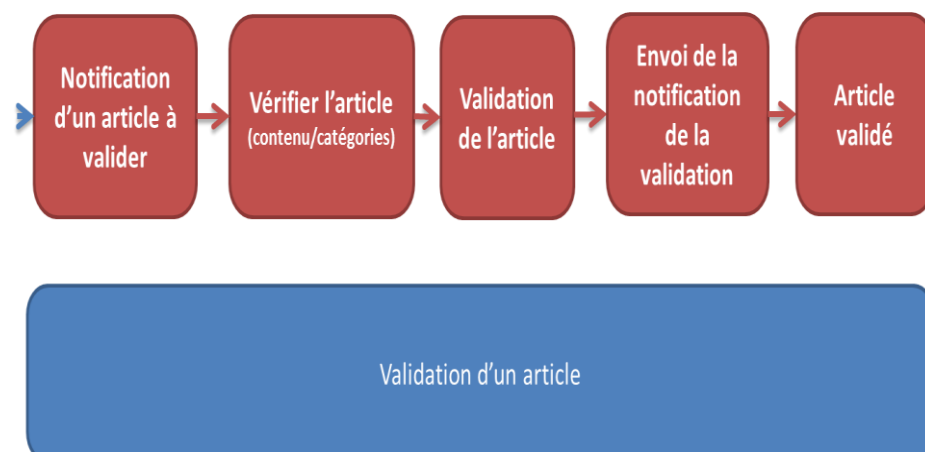


Figure 14 : Architecture de validation d'un article

#### 3.1.2.3.3. L'architecture de modification d'un article :

J'ai conçu une architecture simple pour la modification d'un article.

Le contributeur a accès à ses articles, il peut faire des modifications du contenu et des catégories même après validation.

Après toute modification, l'article doit être revalidé par un éditeur.

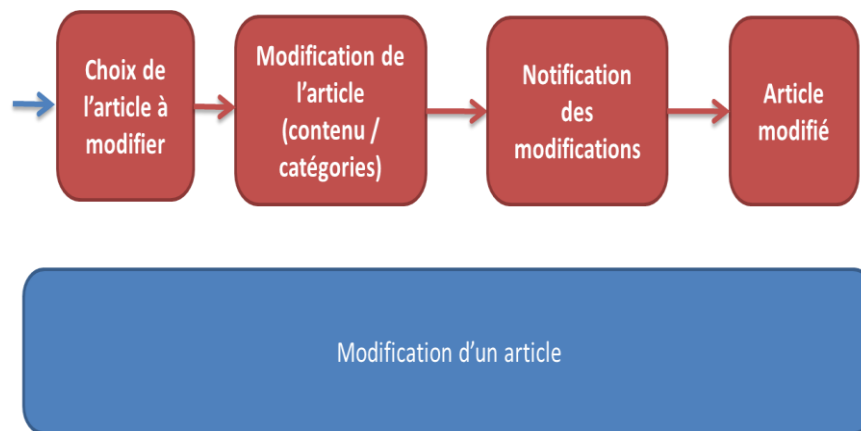


Figure 15 : Architecture de modification d'un article

#### 3.1.2.3.4. Editeur du texte :

Le Framework media wiki utilise un langage de balise spéciale pour formater les textes.

Mais dans notre cas, les futurs utilisateurs du DT-Wiki ne sont pas des informaticiens, ils ne peuvent pas utiliser facilement un langage de balise pour rédiger et mettre en forme leurs articles.

Il faut donc faciliter l'édition des articles. Pour cela j'ai conçu un éditeur de texte qui utilise l'Html, Un Parsoid (Analyseur) permet de convertir les articles du Html vers le format de media wiki.

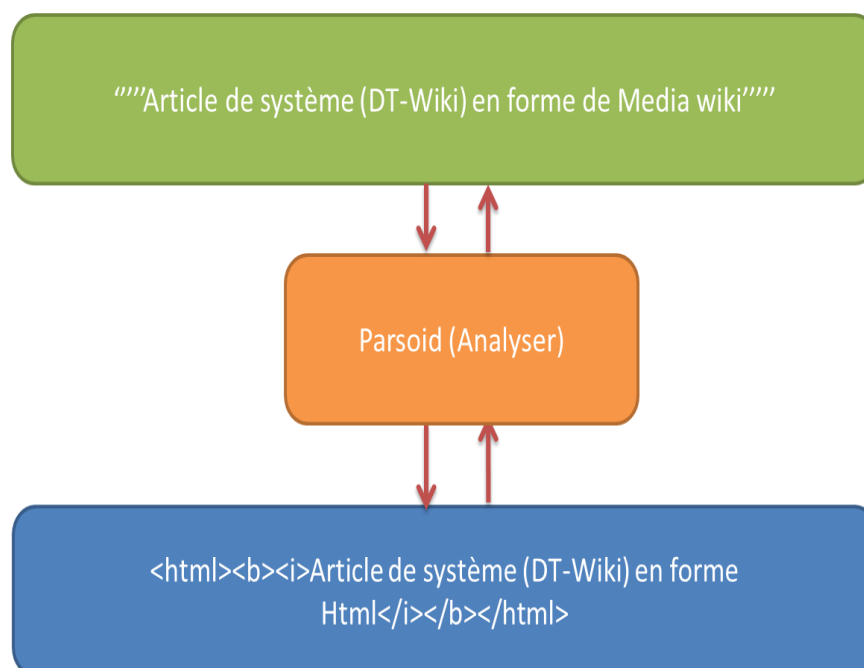


Figure 16 : Architecture d'analyseur du système (DT-Wiki)



#### 3.1.2.4. Indexation des articles :

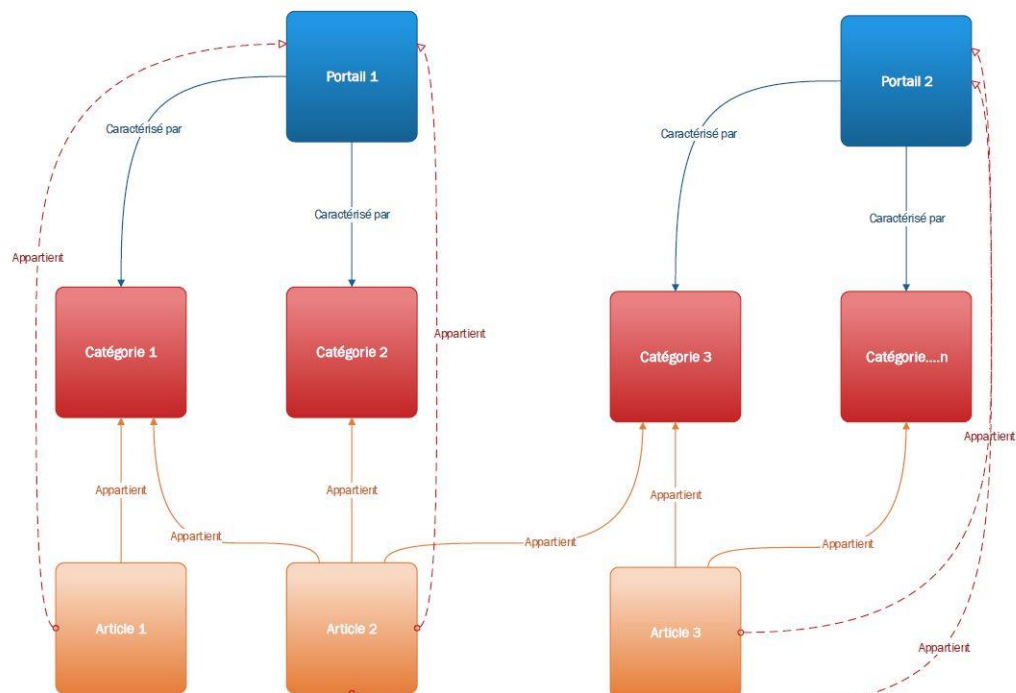
#### 3.1.2.4.1. L'architecture d'indexation des articles :

J'ai conçu la structure d'indexation des articles, pour organiser le classement des articles et faciliter la recherche.

Cette indexation est basée sur des catégories prédéfinies. Certaines sont des catégories « Mères » auxquelles sont rattaché des sous-catégories.

Lors de la création des articles, le contributeur choisit une ou plusieurs catégories dans la liste des catégories existantes (cf. §3.1.2.3.1). Les utilisateurs en fonction de leur rôle peuvent :

- Créer une catégorie ou une sous-catégorie.
- Demander la création d'une catégorie ou sous-catégorie (cf. §3.1.2.1.1).



**Figure 17 : Structure d'indexation des articles**

#### 3.1.2.4.2. L'architecture de création d'une catégorie :

J'ai conçu une architecture pour la création d'une catégorie de cette façon :

L'utilisateur choisit l'option «Créer une catégorie». Puis dans l'interface graphique affichée, l'utilisateur doit saisir le nom de la catégorie puis valider. Le système affiche une interface graphique pour informer que la catégorie est créée avec succès.

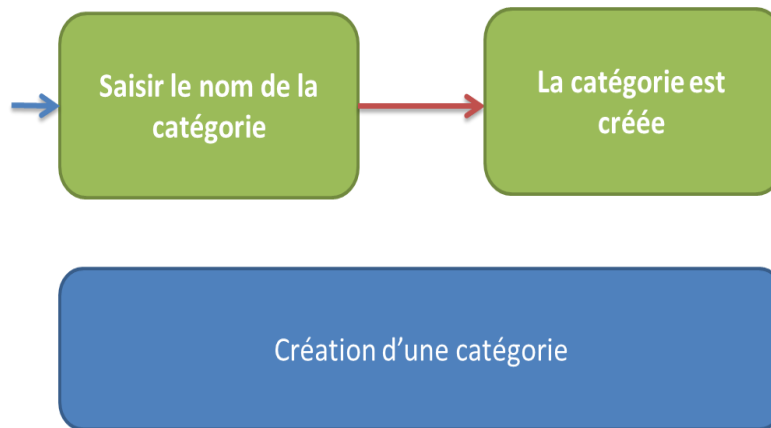


Figure 18 : Architecture de création d'une catégorie

#### 3.1.2.4.3. L'architecture de création d'une sous-catégorie :

L'architecture de création d'une sous-catégorie est la même que pour la création d'une catégorie, sauf que l'utilisateur doit choisir dans l'interface de création une catégorie mère, afin de mettre cette catégorie comme une sous-catégorie.

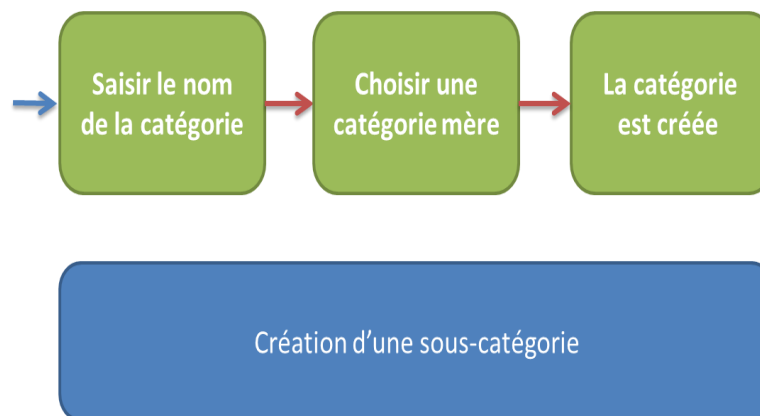


Figure 19 : Architecture de création d'une sous-catégorie

#### 3.1.2.4.4. L'architecture de demande de création d'une catégorie :

J'ai conçu la fonctionnalité « Demande de création d'une catégorie » pour les utilisateurs qui n'ont pas les droits suffisants. L'architecture est la suivante :

L'utilisateur doit choisir l'option « Demande de création d'une catégorie », puis dans l'interface graphique affichée, il faut mettre le nom de la catégorie demandé dans la zone de texte nommé « Demande ». Puis l'utilisateur valide la demande et le système extrait automatiquement les informations de l'utilisateur et les envoie avec la demande vers les utilisateurs autorisés à créer des catégories.

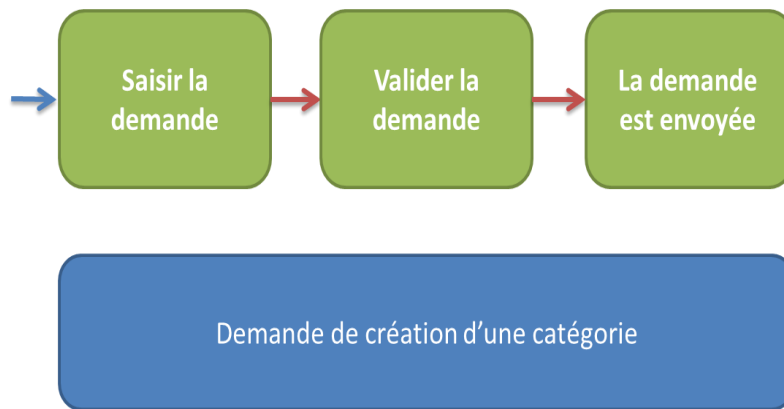


Figure 20 : Architecture de demande de création d'une catégorie

## 3.2. Développement :

Afin de développer le système DT-Wiki, j'ai utilisé l'ensemble des outils informatiques suivants :

### 3.2.1. Environnement & outils de développement :

L'Environnement et les outils de développement sont listés dans le fichier Environnement & outils de développement EID-MED Wiki (Cf. Annexe 6 : Environnement & outils de développement EID-MED Wiki).

### 3.2.2. Développement des besoins spécifiques:

Après la conception des fonctionnalités, le choix de l'environnement du travail et les outils nécessaires pour le développement, j'ai commencé à développer et à réaliser le système DT-Wiki :

#### 3.2.2.1. L'Installation du Framework Media Wiki :

J'ai installé le logiciel media wiki afin de créer une version de développement (DT-Wiki-Dev) et la mettre en œuvre avec une base de données PostgreSQL dans un serveur web.

J'ai fait le paramétrage du système c'est-à-dire les fonctionnalités de la base du système (DT-Wiki) par exemple :

- ❖ Changer les skins (Template) et mettre les logos du système.
- ❖ Empêcher l'accès au système sauf pour les utilisateurs ayant un login et mot de passe :

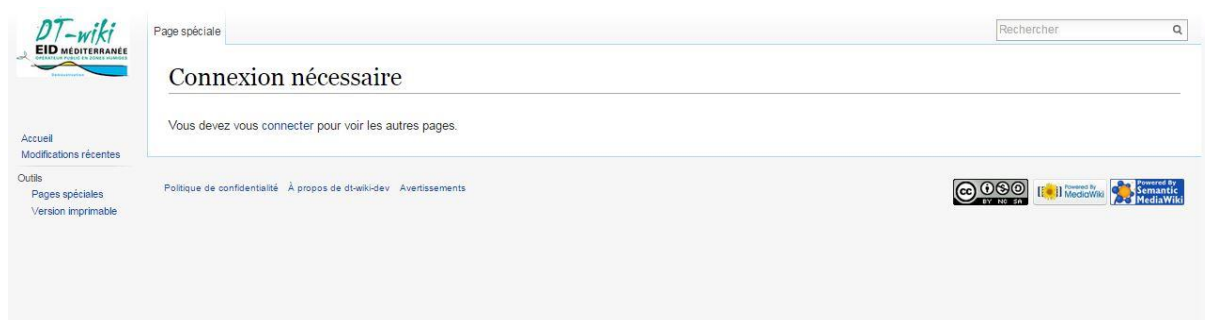


Figure 21 : Interface d'accès

- ❖ Création des espaces des noms, pour la séparation entre les pages c'est-à-dire les regroupements des pages. Par exemple l'espace du nom « Article » comme un espace du nom principale pour la création et la recherche des articles.

### 3.2.2.2. Développement de la gestion des utilisateurs :

#### 3.2.2.2.1. La structure des droits des utilisateurs :

Dans le logiciel Media Wiki, il n'y a pas de notion « types d'utilisateurs ». Pour cela, j'ai créé des groupes d'utilisateurs (Administrateur, Modérateur, Editeur, Contributeur, Utilisateur), à chaque groupe j'ai associé des droits spécifiques par exemple : seuls les administrateurs ont le droit de créer des utilisateurs.

Ce rôle est sauvegardé dans la base de données par l'ajout d'une colonne.

#### 3.2.2.2.2. Création d'un utilisateur :

J'ai commencé par la vérification de système d'inscription par défaut dans media wiki : n'importe qui peut s'inscrire et mettre ses informations de façon facultative.

Mais, j'ai trouvé que ce système ne peut pas satisfaire le besoin de l'équipe du projet : seul l'administrateur créer des comptes utilisateurs.

Donc ce n'est pas le cas dans DT-Wiki pour cela j'ai redéveloppé le système de création des comptes utilisateur sur la base de l'ancien système.

Une illustration de cette fonction est donnée dans la figure 22 :

Figure 22 : Page de création d'un utilisateur

### 3.2.2.2.3. Changer le mot de passe :

J'ai commencé par créer une page qui contient un formulaire de saisie l'ancien et le nouveau mot de passe.

Puis, j'ai ajouté cette fonction à la partie gauche (Sidebar) car cette fonction est accessible par n'importe qu'elle utilisateur.

Ensuite, j'ai développé un script JavaScript et jQuery afin de vérifier les informations de l'utilisateur, ce script communique avec un script PHP pour vérifier que l'ancien mot de passe est correct puis le changer avec le nouveau mot de passe.

Cette fonction est opérationnelle. Une illustration est donnée dans la figure 23 :

Figure 23 : Changer le mot de passe

### 3.2.2.3. Développement de l'architecture du système :

#### 3.2.2.3.1. L'architecture de navigation du système :

J'ai essayé de créer et formater les pages principales du système et leurs architectures avec l'utilisation de script wiki spéciale de media wiki.

Mais, le script de media wiki ne permet pas la création des pages avec une mise en forme avancée. Pour cela j'ai changé le cœur de media wiki pour pouvoir utiliser l'Html et le CSS dans la mise en forme des pages.

Une illustration de cette fonction est donnée dans la figure 24 :

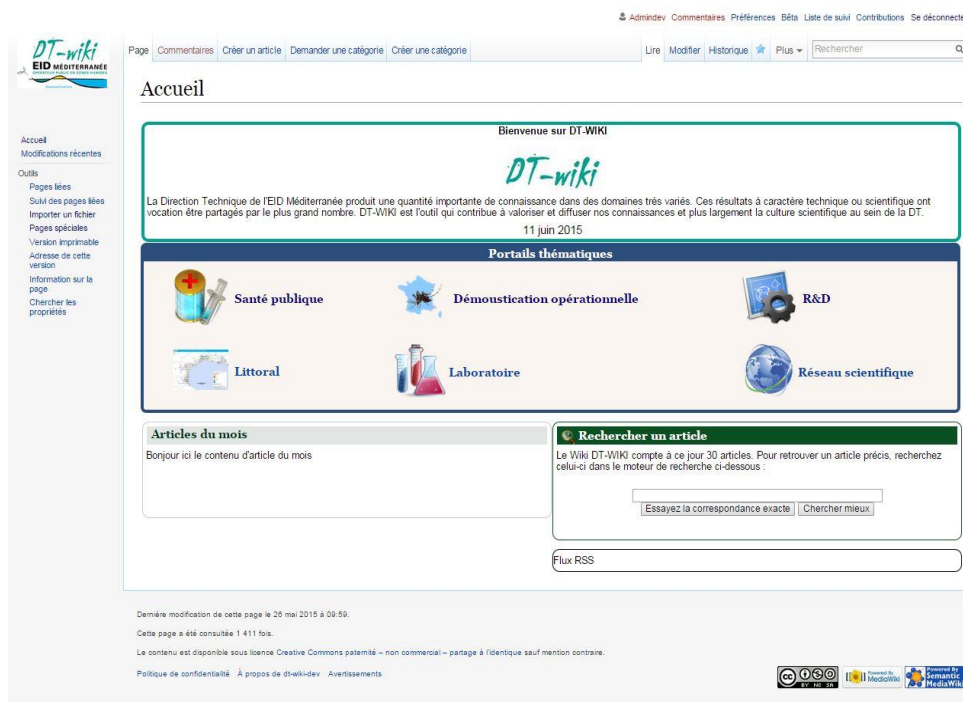


Figure 24 : Page d'accueil du DT-Wiki-Dev

#### 3.2.2.3.2. Afficher le dernier article :

Cette fonctionnalité de « Afficher le dernier article » est absente du Framework media wiki. Je n'ai pas réussi à la trouver dans les extensions. J'ai donc intégralement développé cette fonctionnalité en utilisant la notion de « Hooks » : un code PHP est injecté dans un autre fichier par l'appel d'un Hook.

Encore, j'ai utilisé le contrôle de flux binaire pour n'afficher qu'une partie de l'article (le paramètre d'affichage de 256 bits).

Malgré quelques problèmes rencontrés, cette fonctionnalité est opérationnelle.

### 3.2.2.3.3. Chercher un article :

La fonctionnalité intégrée par défaut dans le Framework media wiki est une recherche sur toutes les pages. Elle n'est pas adaptée à notre cas : le besoin client est de chercher uniquement dans les articles (Titres et contenu).

Cette fonction est la seule que je n'ai pas intégralement développée. J'ai modifié la fonction par défaut en limitant la recherche dans l'espace du nom « Article ».

Ce paramétrage est suffisant pour l'instant (répond au besoin client). Il sera certainement à améliorer.

La figure 26 présent le résultat :

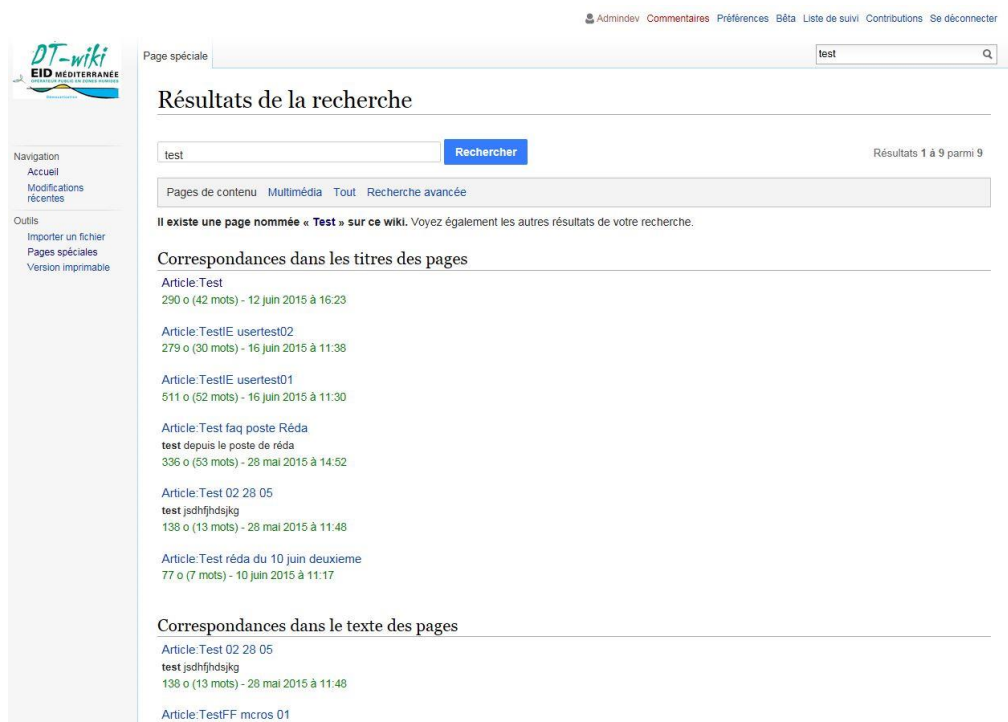


Figure 25 : Exemple de recherche d'un article

### 3.2.2.4. Développement du Cycle de vie de l'article :

#### 3.2.2.4.1. Création d'un article :

Après la conception de cette fonctionnalité, j'ai commencé par développer intégralement cette fonctionnalité car il n'existe pas de solution dans media wiki.

J'ai réalisé une extension pour ajouter un onglet en haut de la page contient « Créer un article » avec l'utilisation de notion de Hooks.

Cet onglet a pour rôle de faciliter l'accès de l'utilisateur à cette fonctionnalité car cette fonctionnalité est accessible par tous les types d'utilisateurs.

De plus, j'ai créé une page avec les scripts HTML, CSS, JavaScript qui contient tous les modèles d'articles possibles afin d'obliger l'utilisateur à choisir un modèle.

Des pages contiennent le script de formatage d'un modèle d'article : script écrit avec le media wiki mark up et la notion de section de media wiki.

Ensuite, j'ai créé d'autres pages qui contiennent le formulaire de saisie du nom de l'article associé à chaque modèle.

Selon le modèle d'article et leur formulaire de saisie le système génère une page dans l'espace du nom « Article : » et affiche l'éditeur de texte avec l'injection automatique du modèle d'article grâce à la notion de *preload* dans media wiki.

Après que l'utilisateur ait rédigé son article et cliqué sur le bouton « enregistrer », le système affiche un pop-up qui liste toutes les catégories du système pour obliger l'utilisateur à choisir une ou plusieurs catégories afin d'indexer cet article (utilisation des boucles de contrôle en JavaScript).

Puis après la validation, le système utilise le JavaScript pour injecter un script media wiki mark up pour indexer l'article, puis convertir le texte de l'article de forme HTML vers la forme media wiki mark up avec un analyseur que j'ai développé, et enfin sauvegarder l'article dans la base de données.

En parallèle, Le système utilise le JavaScript et la Library JQuery afin d'utiliser l'Ajx pour manipuler un code PHP côté serveur pour envoyer des emails de notification aux éditeurs.

Malgré quelques problèmes rencontrés, j'ai fini cette fonctionnalité et une illustration de cette fonction est donnée dans la figure 27 :

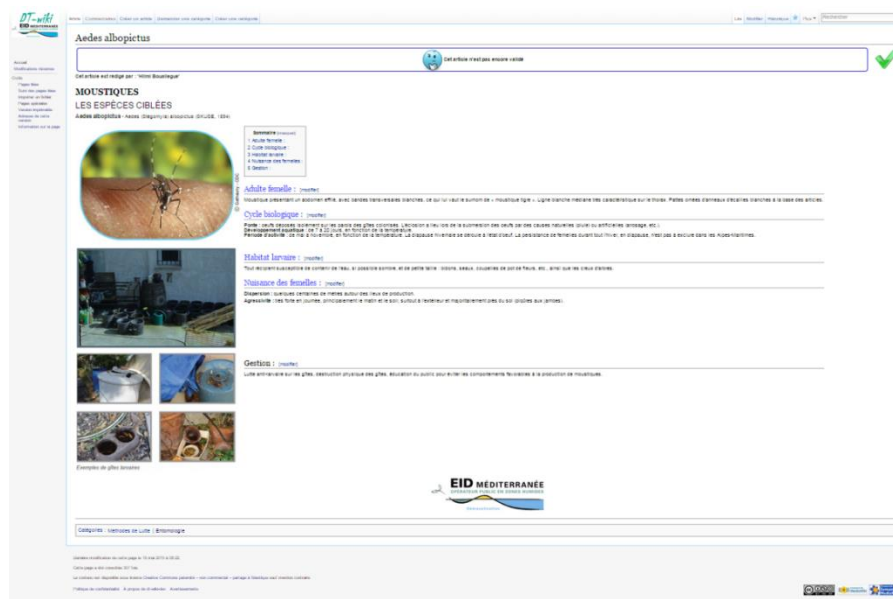


Figure 26 : Exemple d'un article créé



#### 3.2.2.4.2. Validation d'un article :

J'ai commencé à développer cette fonctionnalité intégralement car elle n'existe pas dans media wiki.

J'ai développé un système qui permet de vérifier la session de l'utilisateur (le rôle de l'utilisateur) avec des blocs des codes JavaScript, PHP.

Selon le rôle de l'utilisateur, la fonctionnalité est activée ou désactivée.

Lorsque l'éditeur clique sur le bouton de validation, le système exécute un script JavaScript et jQuery pour manipuler l'Ajax afin de synchroniser avec un script PHP pour envoyer des emails de notification et changer l'état de l'article dans la base de données (j'ai créé des autres tables dans la base de données pour sauvegarder l'état de validation des articles).

J'ai fini cette fonctionnalité et une illustration de cette fonction est donnée dans la figure 28 :



Figure 27 : Exemple de validation d'un article par l'éditeur

#### 3.2.2.4.3. Modification d'un article :

J'ai développé cette fonctionnalité intégralement car elle n'existe pas dans media wiki.

Cette fonctionnalité est basée sur des parties de la fonctionnalité « Créer un article » car :

Après que l'utilisateur ait cliqué sur le bouton « Modifier » en haut de la page, le système affiche l'éditeur de texte avec l'injection d'un article à partir de la base de données.

Puis après avoir fait des modifications, l'utilisateur clique sur le bouton « Enregistrer ».

J'ai développé un script JavaScript et jQuery pour manipuler l'Ajax afin d'utiliser le PHP pour vérifier l'état de l'article dans la base de données, c'est-à-dire une création ou une modification d'un article.

Dans notre cas, c'est modification d'un article donc le système exécute un script PHP pour rétablir l'état de l'article non validé et retransmettre des emails de notification aux éditeurs.

J'ai fini cette fonctionnalité.

#### 3.2.2.4.4. Editeur du texte des articles :

J'ai cherché des extensions pour transformer un texte sous forme wiki mark up vers le format HTML.

J'ai trouvé quelques solutions mais qui ne sont pas efficaces et contiennent beaucoup de bugs.

J'ai utilisé le squelette d'un éditeur de texte existe et j'ai modifié quelques boutons dans l'éditeur.

J'ai créé un parsoid (analyseur) en JavaScript pour transformer le texte écrit dans l'éditeur de format html vers la forme de wiki mark up.

Malgré quelques problèmes rencontrés, j'ai fini cette fonctionnalité.

#### 3.2.2.5. Développement du système d'indexation des articles :

##### 3.2.2.5.1. Création d'une catégorie :

J'ai commencé à développer cette fonctionnalité avec l'utilisation de notion de catégorie de media wiki.

J'ai créé une extension pour ajouter une page spéciale pour créer des catégories.

J'ai limité l'accès à cette page seulement pour le rôle administrateur et modérateur.

Dans cette page l'utilisateur peut saisir le nom de catégorie puis valider.

Après la validation le système enregistre la page dans la base de données avec l'association d'espace de nom « Catégorie : ».

La figure 31 montre le résultat :



Figure 28 : Page de création d'une catégorie ou d'une sous-catégorie

##### 3.2.2.5.2. Demande de création d'une catégorie :

J'ai commencé à développer cette fonctionnalité car elle n'existe pas dans media wiki.

J'ai créé une extension pour ajouter cette page comme un onglet en haut de la page pour faciliter l'accès à cette page car cette fonctionnalité doit accessible par n'importe qu'elle utilisateur.

J'ai créé une page contient un formulaire pour saisir la demande.

De plus, je développé un script JavaScript pour traiter la session de l'utilisateur et extraire leur nom.

Ensuite, ce script utilise jQuery et Ajax pour manipuler un code PHP pour extraire les autres informations de l'utilisateur puis envoyer un email au modérateur qui contient la demande et les informations du demandeur.

J'ai fini cette fonctionnalité et le résultat présent dans la figure 31 :



Figure 29 : Page de demande de création d'une catégorie

### 3.3. Test et déploiement :

Le système (DT-Wiki) est développé selon l'approche «Agile» sous forme de cycles (itérations), chaque itération est finie par une application utilisable d'une portion des exigences du client, ainsi le client est impliqué dans la phase de test pour tester et valider le travail fait et de changer ou ajouter des autres besoins.

Dans la phase de test on a utilisé des types des tests afin de :

- Satisfaire les exigences du client.
- Contrôler et corriger les erreurs avant de mettre en place le système.
- Minimiser le coût de correction.

Ensuite, ces tests ont été réalisés selon trois approches :

### **3.3.1. Tests de développement :**

Les tests de développement du système (DT-Wiki) sont réalisés par le développeur, donc j'ai réalisé ces tests afin de découvrir les erreurs, les bugs, les anomalies et les défauts du système.

#### **3.3.1.1. Tests unitaires :**

Je réalise des tests des unités de programmes afin de tester les blocs des codes développés (les méthodes, les classes, les opérations d'un objet, les boucles, les interfaces graphiques...).

#### **3.3.1.2. Tests de composants :**

J'intègre les unités de code, afin de créer des composants du système (combinaison des objets) pour tester l'accès aux fonctionnalités des objets à partir des autres objets.

#### **3.3.1.3. Tests système :**

Après les tests de composants, j'intègre ces derniers afin de tester les interactions entre ces composants et vérifier si les composants interagissent correctement et transfèrent les bonnes données.

### **3.3.2. Tests de version :**

Le test de version est réalisé par les membres de l'équipe du projet et pas par les membres de l'équipe de développement afin de tester une version particulière avant qu'elle soit préparée pour les utilisateurs.

### **3.3.3. Tests utilisateur :**

Le test utilisateur est réalisé par un échantillon d'utilisateurs afin de tester le système dans leurs propres environnements.

## **3.4. Formation et perspective :**

Jusqu'à cette itération de développement, le système n'est pas totalement complet car il y a des autres fonctionnalités à finir comme par exemple la recherche avancée des articles.

Ensuite, pour les futurs utilisateurs du système (DT-Wiki) il faut réaliser une documentation « Manuelle d'utilisations du système DT-Wiki », puis une documentation pour la configuration du système pour l'administrateur.

## 4. Conclusion :

J'ai intégré l'EID méditerranée, pendant 4 mois, afin de développer le système DT-Wiki pour la valorisation et de la diffusion des connaissances en interne de la direction technique avec le respect des contraintes (la gestion des utilisateurs, la facilite de création et d'édition des articles, la validation des articles,...).

Malgré quelques difficultés rencontrées durant le stage telles que les contraintes de temps (le déphasage entre les calendriers du planning et les calendriers réels), la gestion de projet (j'ai travaillé dans tous les phases du projet (un concepteur, développer, testeur)) et les limites des technologies maîtrisées, nous avons réussi à mettre en place une application fonctionnelle.

L'EID Méditerranée, dispose maintenant d'un outil opérationnel et suffisamment performant pour assurer une bonne diffusion des données et des résultats des différents projets et études. Cet outil est certainement encore à améliorer notamment si on cherche à le déployer dans d'autres services et directions de l'EID Méditerranée.

Lors de ma mission de conception et développement de l'application Wiki-DT, j'ai pu améliorer mes compétences en développement web notamment par l'utilisation des nouvelles technologies et outils informatiques. Cette expérience a été enrichissante pour moi : j'ai travaillé sur la notion de « wiki » (une des raisons qui m'ont fait choisir ce stage), j'ai pu m'intégrer à des équipes de spécialités différentes et apprécier l'interaction avec les membres professionnels du projet, et enfin d'améliorer mes capacité de rédaction des documents techniques avec les exigences du monde professionnel.

Plus largement, cette première expérience professionnelle a été l'opportunité pour moi d'appréhender la communication avec les clients, d'extraire les exigences et les souhaits des clients, de comprendre la vision du client et la rédaction des documents techniques. De plus, cette mission, m'a permis d'utiliser de nouveaux langages de programmation, d'apprendre à partir de mes fautes et de m'enrichir des idées et des discussions des membres de l'équipe projet.

A court terme, il reste à améliorer : la recherche avancée, le manuel de développement (pour l'administrateur), le manuel d'utilisation (pour l'utilisateur) et finaliser la version de production du système DT-Wiki.

A moyen terme, le système est extensible donc on peut imaginer d'ajouter d'autres fonctionnalités (manipulation des cartes géographiques via une API, test d'algorithmes de *data mining* et de *pattern recognition*) et adapter le système pour le rendre compatible avec les smartphones et les tablettes par exemple.

## 5. Annexes et bibliographie :

### 5.1. Bibliographie :

- Stage\_wiki\_EID\_MED
- Guide\_IEEE\_Pour\_la\_Specification

### 5.2. Webographie :

<sup>1</sup> : <http://www.eid-med.org/page/recherche-et-developpement> (consulté le 03/06/2015).

<sup>2</sup> : <http://agilemanifesto.org/iso/fr/principles.html> (consulté le 05.06.2015).

<sup>3</sup> : <http://uml.free.fr/cours/p10.html#concept> (le 05/06/2015).

<sup>4</sup> : <http://uml.free.fr/cours/i-p19.html> (consulté le 05/06/2015).

### 5.3. Liste des annexes :

- Annexe 1 : Diagramme de cas d'utilisation-EID-MED Wiki.
- Annexe 2 : Document des exigences-EID-MED Wiki.
- Annexe 3 : Dossier d'information EID-MED.
- Annexe 4 : Droits d'utilisateur-EID-MED Wiki.
- Annexe 5 : Maquette de Accueil-EID-MED Wiki.
- Annexe 6 : Indexation-EID-MED Wiki.
- Annexe 7 : Environnement & outils de développement EID-MED Wiki.