



**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 07/07/2015

****

Présenté par :

**Le stagiaire Développeur**

**Hilmi BOUALLEGUE**

Encadré Par :

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

**M. Réda TOUNSI, Maître de stage**

Année 2014-2015

**Sommaire**

Contenu

[1. Introduction : 7](#_Toc422225663)

[2. Contexte du stage: 8](#_Toc422225664)

[2.1. Présentation établissement : 8](#_Toc422225665)

[2.1.1. Historique de L’EID Méditerranée : 8](#_Toc422225666)

[2.1.2. Activité de L’EID Méditerranée : 9](#_Toc422225667)

[2.1.3. Structure de L’EID Méditerranée : 10](#_Toc422225668)

[2.2. Présentation du projet : 10](#_Toc422225669)

[2.2.1. Recherche & Développement : 10](#_Toc422225670)

[2.2.2. L’échange de données : 10](#_Toc422225671)

[2.2.3. Organisation et Planification : 11](#_Toc422225672)

[2.2.4. Choix de l’outil et contraintes : 11](#_Toc422225673)

[3. Réalisation de l’outil : 11](#_Toc422225674)

[3.1. Conception : 12](#_Toc422225675)

[3.1.1. Outils méthodologiques : 12](#_Toc422225676)

[3.1.1.1. Rédaction du document d’exigences : 12](#_Toc422225677)

[3.1.1.2. Diagramme de cas d’utilisation : 13](#_Toc422225678)

[3.1.1.3. Diagramme de séquences : 14](#_Toc422225679)

[3.1.1.4. Le cahier de recette des tests : 15](#_Toc422225680)

[3.1.2. Définition des besoins spécifiques : 16](#_Toc422225681)

[3.1.2.1. Administration & gestion des utilisateurs : 16](#_Toc422225682)

[3.1.2.2. Navigation et consultation du DT-Wiki : 18](#_Toc422225683)

[3.1.2.3. Ajout de contenu dans le système (DT-Wiki) : 21](#_Toc422225684)

[3.1.2.4. Indexation des articles : 24](#_Toc422225685)

[3.2. Développement : 27](#_Toc422225686)

[3.2.1. Environnement de développement : 27](#_Toc422225687)

[3.2.1.1. Media Wiki version 1.24.2: 27](#_Toc422225688)

[3.2.1.2. Technologies additionnels : 30](#_Toc422225689)

[3.2.2. Développement des besoins spécifiques: 31](#_Toc422225690)

[3.2.2.1. L’Installation du Framework Media Wiki : 31](#_Toc422225691)

[3.2.2.2. Développement de la gestion des utilisateurs : 32](#_Toc422225692)

[3.2.2.3. Développement du l’architecture de système : 33](#_Toc422225693)

[3.2.2.4. Développement du Cycle de vie de l’article : 35](#_Toc422225694)

[3.2.2.5. Développement du système d’indexation des articles : 38](#_Toc422225695)

[3.3. Test et déploiement : 39](#_Toc422225696)

[3.3.1. Tests de développement : 40](#_Toc422225697)

[3.3.1.1. Tests unitaires : 40](#_Toc422225698)

[3.3.1.2. Tests de composants : 40](#_Toc422225699)

[3.3.1.3. Tests système : 40](#_Toc422225700)

[3.3.2. Tests de version : 40](#_Toc422225701)

[3.3.3. Tests utilisateur : 40](#_Toc422225702)

[3.4. Formation et perspective : 40](#_Toc422225703)

[4. Conclusion : 41](#_Toc422225704)

[5. Annexe et bibliographies : 41](#_Toc422225705)

[5.1. Bibliographies : 41](#_Toc422225706)

[5.2. Webographies : 41](#_Toc422225707)

**Liste des figures**

[Figure 1: Schéma de la méthode "agile" (2) 12](#_Toc421631984)

[Figure 2 **Erreur ! Signet non défini.**](#_Toc421631985)

**Liste des tableaux**

[Tableau 1 : Extrait du cahier de texte fourni en Annexe 2 15](#_Toc421632008)

[Tableau 2 : 17](#_Toc421632009)

**Remerciement**

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 même à ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable.

Aussi, je remercie M. TOUNSI Reda, mon maître de stage qui m’a formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie. Encore je remercie Madame CROS Myriam et Madame BARRAS Nathalie pour leurs conseille et aide dans la réalisation du stage.

Encore, je remercie l’ensemble des agents de l’EID méditerrané pour les conseils et la bienvenue qu’ils ont pu me prodiguer au cours de ces mois du stage.

Enfin, je remercie les enseignants de master web intelligence à l’université Jean Monet, Saint Etienne pour leurs conseils, leurs cours professionnelles et leurs les outils pédagogiques et techniques fournisse durant les mois des cours, qui ont déjà m’aider beaucoup dans la réalisation de ce stage.

# Introduction :

Du 07/04/2015 au 07/08/2015, j’ai effectué un stage au sein de L’EID Méditerranée (Entente Interdépartementale pour la Démoustication du littoral Méditerranéen) située à Montpellier.

L’EID Méditerranée 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, deux laboratoires, la direction des moyens, le service de l’information géographique et un parc de véhicules et matériels.

Au sein de la direction technique où j’ai fait mon stage, il y a une forte activité de recherche et de développement qui génère une quantité énorme des données dans des secteurs différents, ça permet de générer des problèmes de diffusion et d’échange les connaissances dans un environnement hétérogène. Tel que pour diffuser une petite information il faut faire des grandes réunions sinon l’information prendre beaucoup de temps pour le diffuser.

Dans le but de valorisation et de diffusion les connaissances, le service SI () propose le choix d’un système de type « WIKI », pour cela la direction technique recherche un stagiaire pour mettre en œuvre un logiciel participatif de type « wiki » afin de développer une base de connaissances techniques de la démoustication.

Au cours de ce stage à la direction technique de l’EID Méditerranée, j’ai réalisé des tâches de conception (documents du système, conception de la structure du système, conception de la structure des utilisateurs et leurs droits d’accès), de réalisation et de développement du système (Création de la structure globale du système (le squelette du système), création de la structure globale des utilisateurs (les droits d’accès), création et développement des fonctionnalités du système.

Et ces taches sont bien détaillées dans la partie « Réalisation de l’outil ». Tel que ce document est divisé en 3 grandes parties :

* **Contexte du stage.**
* **Réalisation de l’outil.**
* **Conclusion.**

# Contexte du stage:

## Présentation établissement :



Figure logo EID Méditerranée

### Historique de L’EID Méditerranée :

L’EID Méditerrané est un établissement interdépartemental créé en 1958, afin de la démoustication du littoral méditerranéen

***« Il y a 30-40 ans, le littoral méditerranéen français n'était pas celui que tout le monde connaît aujourd'hui. Les villes, les campagnes et les plages risquaient de passer à côté de sérieuses opportunités de développement économique et touristique. Car elles étaient envahies par des insectes insupportables : les moustiques, qui rendaient ce pays magnifique invivable !***

***C'est pourquoi en 1958, les Conseils généraux des départements de l'Hérault, du Gard et des Bouches-du-Rhône évaluant l'avenir exceptionnel qui s'ouvrait à leurs départements, ont décidé de créer l'EID Méditerranée, l'Entente Interdépartementale pour la démoustication du littoral méditerranéen (EID-Med).***

***En 1963, l'Aude et les Pyrénées-Orientales les rejoignent. C'est cette année-là que l'Etat a créé la Mission interministérielle d'aménagement touristique du Languedoc-Roussillon, dit "Mission Racine". Fixant la démoustication comme préalable incontournable à toute démarche d'aménagement, l'Etat a contribué au financement de l'EID Méditerranée, aux côtés des collectivités territoriales, jusqu'en 1982, date de dissolution de la Mission interministérielle. »1.***

***« De 1959 à 1963, le financement de l'EID Méditerranée est ensuite assuré grâce à une participation majoritaire de la Mission Racine (Etat), qui fluctue autour de 85%, et des Départements, qui se situe aux environs de 15% jusqu'en 1974. »2.***

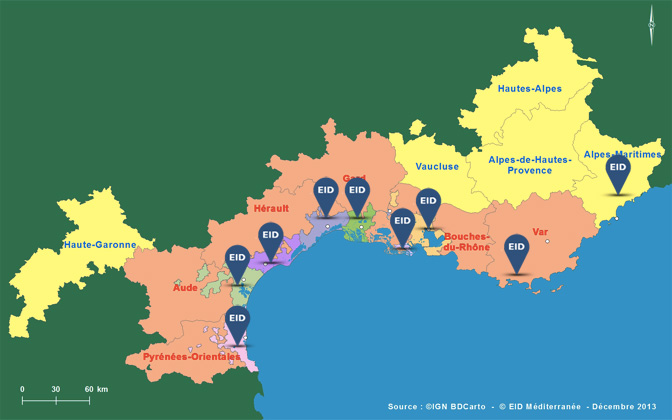


Figure :La zone d’action de l'EID Méditerranée

### Activité de L’EID Méditerranée :

Les principales missions de l’EID Méditerranée sont :

* Le contrôle des nuisances causé par les moustiques (traité les larves des moustiques).
* La préservation des zones humides littorales (conservation des littorales).

***« L’EID-Med a pour mission centrale le contrôle de la population des espèces nuisances de moustiques proliférant dans les zones humides marginales des étangs et lagunes du littoral. Ce contrôle consiste, non pas à éradiquer l’ensemble des insectes piqueurs, mais à maintenir la gêne due aux moustiques à un seuil jugé tolérable…***

***L’EID Méditerranée participe également à des actions importantes de conservation et de mise en valeur écologique des milieux naturels (restauration des cordons dunaires et suivi des systèmes littoraux)…***

***L’EID Méditerranée neutralise chaque année l’apparition des larves de moustiques sur plus de 90 000 hectares de zones marécageuses, 2 000 kilomètres de fossés et 40 000 gîtes larvaires urbains »3.***

### Structure de L’EID Méditerranée :

***« La direction générale de l’EID-Med, installée à Montpellier, regroupe la direction administrative, la direction technique, deux laboratoires, dont un sécurisé et cofinancé au titre du contrat état / Région Languedoc-Roussillon (CPER), la direction des moyens, le service de l’information géographique et un parc de véhicules et matériels. »4.***

## Présentation du projet :

### Recherche & Développement :

La direction technique de l’EID méditerrané, contient une forte activité de recherche et de développement qui produit une grande quantité des données dans des secteurs différents

***« L’EID Méditerranée dispose de laboratoires dédiés à la recherche et au développement 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. »6***

### **L’échange de données :**

La grande quantité des données produire par la direction technique permet de créer le phénomène de la malédiction de la dimensionnalité.

Par exemple, pour diffuser un nouveau résultat, la direction technique besoin de faire plusieurs réunions.

Ensuite, en cas de recherche des informations, il y a des difficultés pour trouver l’information.

Et encore, le retard de la diffusion de l’information par exemple des gens de l’EID Méditerranée connu une information après plusieurs mois.

### Organisation et Planification :

### Choix de l’outil et contraintes :

Le service SI (Système d’information) propose une solution Wiki, afin de valorisation et de diffusion des connaissances en interne de la direction technique, selon des contraintes :

* La gestion des utilisateurs est plus fine.
* La facilite de navigation du système (Structure apriori).
* Création des catégories apriori pour indexer les articles et faciliter les trouver.
* Faciliter la création et l’édition des articles.

Pour cela, la direction technique cherché un stagiaire informaticien pour développer et mettre en œuvre un logiciel participatif de type « wiki ».

Enfin, j’ai choisi ce stage pour mieux pratiquer les technologies web dans un projet réel et professionnel. Et encore l’intégration dans une société pour découvrir le monde du travail.

# 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 réalisé 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.»3.***



Figure : 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.

## Conception :

### 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.

#### 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)»3.***

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.).

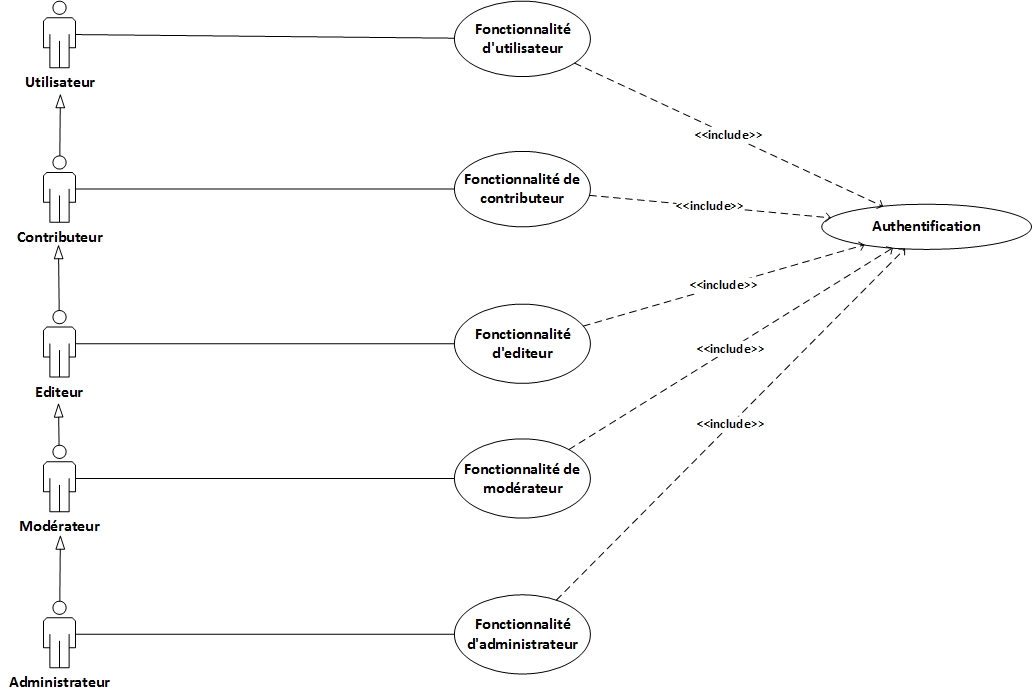
#### Diagramme de cas d’utilisation :

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. (3) »***

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 :



#### 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. »4***

Ce document, nous a permis de mieux visualiser la communication entre les composants du système et l’utilisateur selon un ordre chronologique par exemple le diagramme de séquences XXX:

#### 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 | **Erreur de connexion**  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 | **Erreur de connexion**  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  : Extrait du cahier de texte fourni en Annexe 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.

#### Administration & gestion des utilisateurs :

##### 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 (Cf. 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 créé 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.

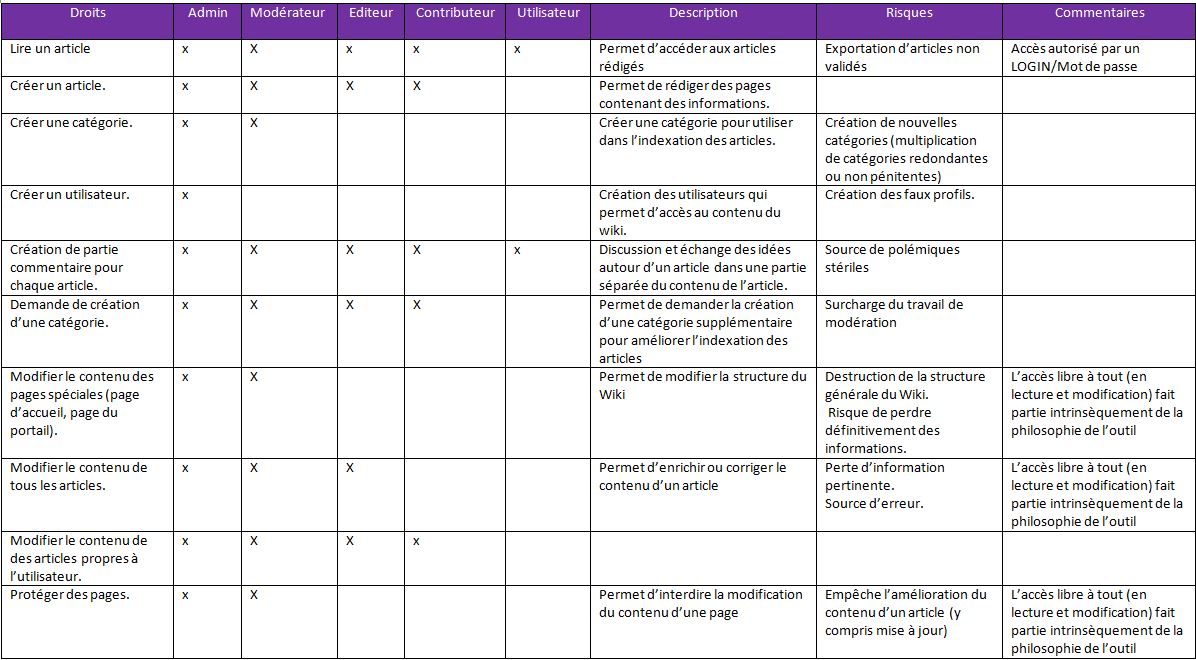
* ***Contributeur :***

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.

Ce document, nous a permis de mieux et leurs rôles par exemple :

Tableau  : Extrait du Droits d’utilisateur-EID-MED Wiki fourni en Annexe 5

##### 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 choisi 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 diffèrent à l’utilisateur en utilisant l’option « Gestion des droits d’utilisateur ».

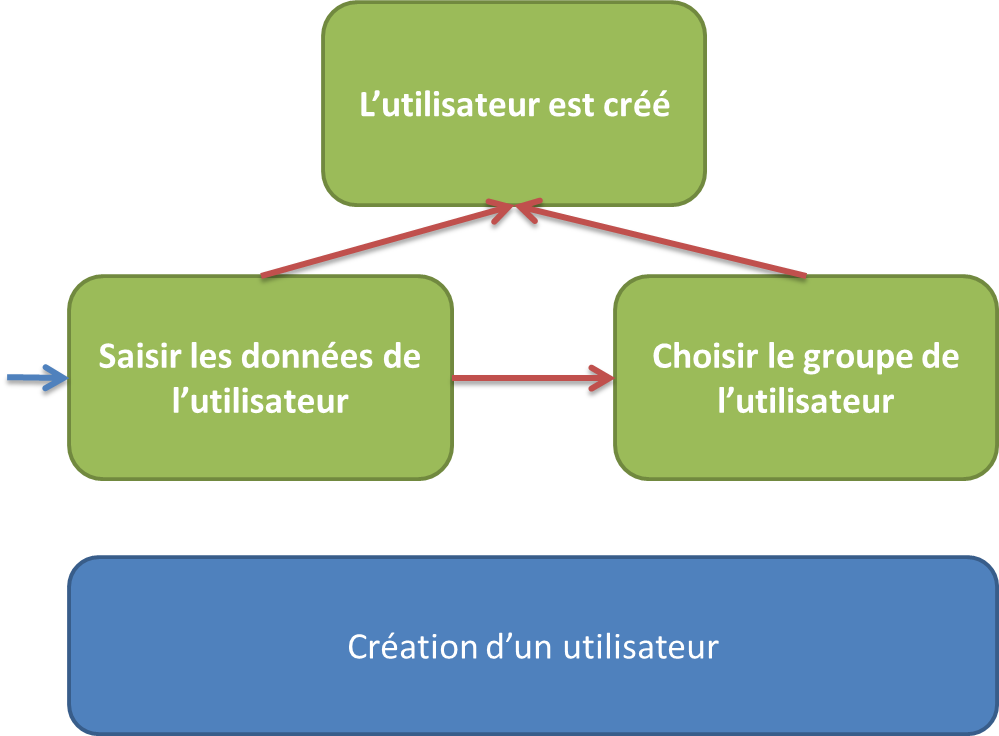


Figure : Architecture de création d'un utilisateur

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

##### 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 :

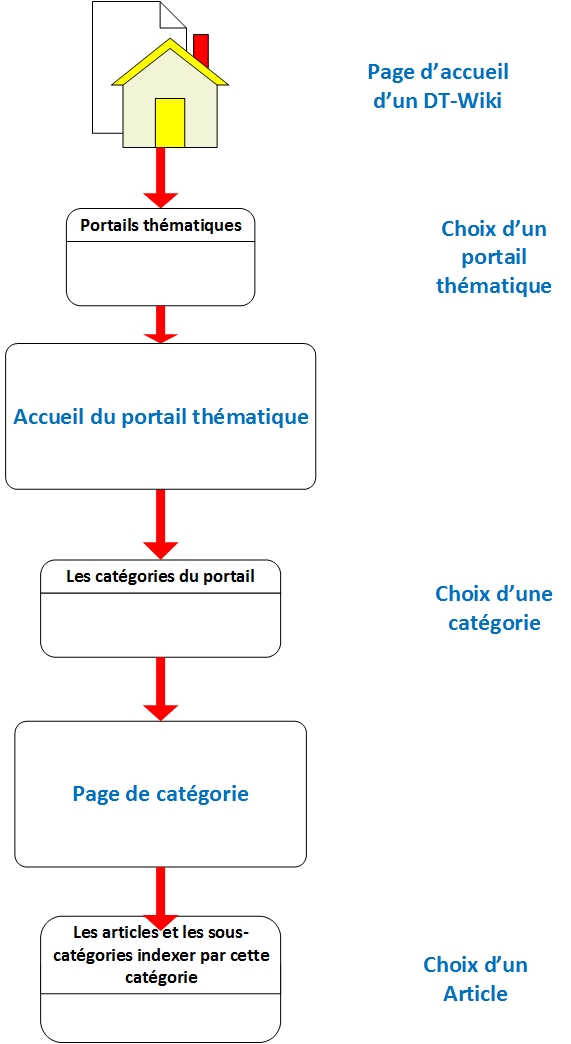


Figure : Architecture de consultation du DT-Wiki

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

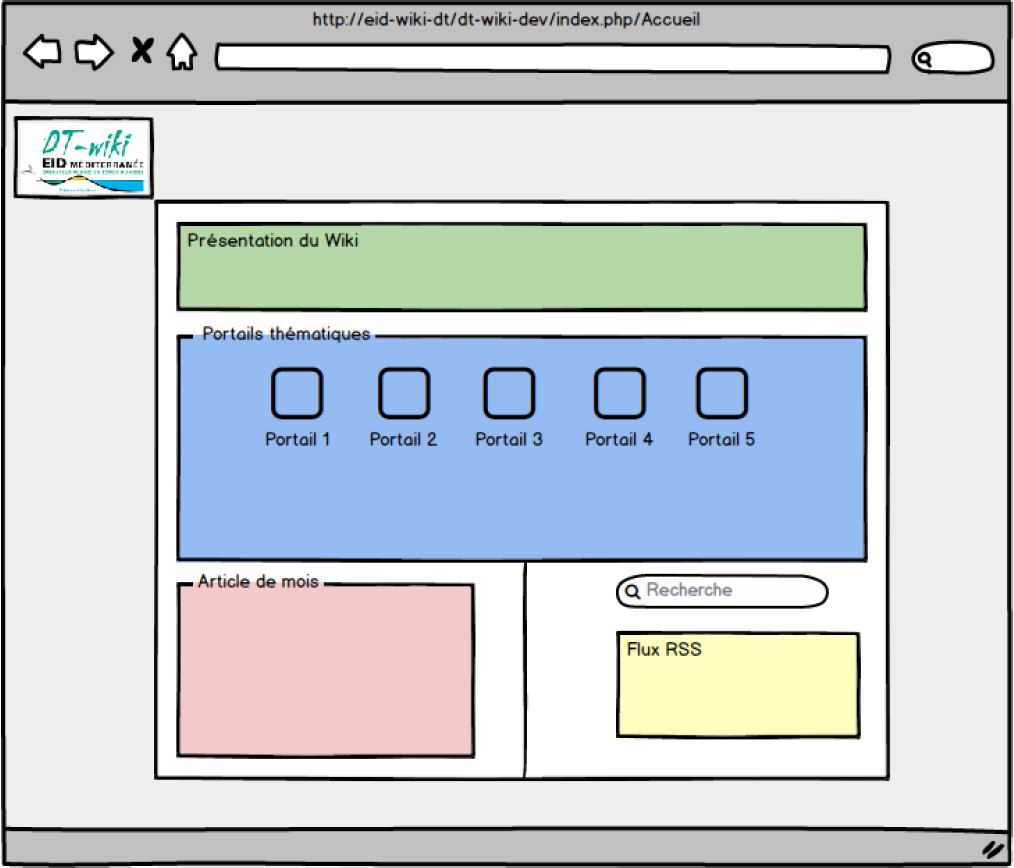


Figure : Extraction de maquette fournie en Annexe 6 Maquette de page d’accueil

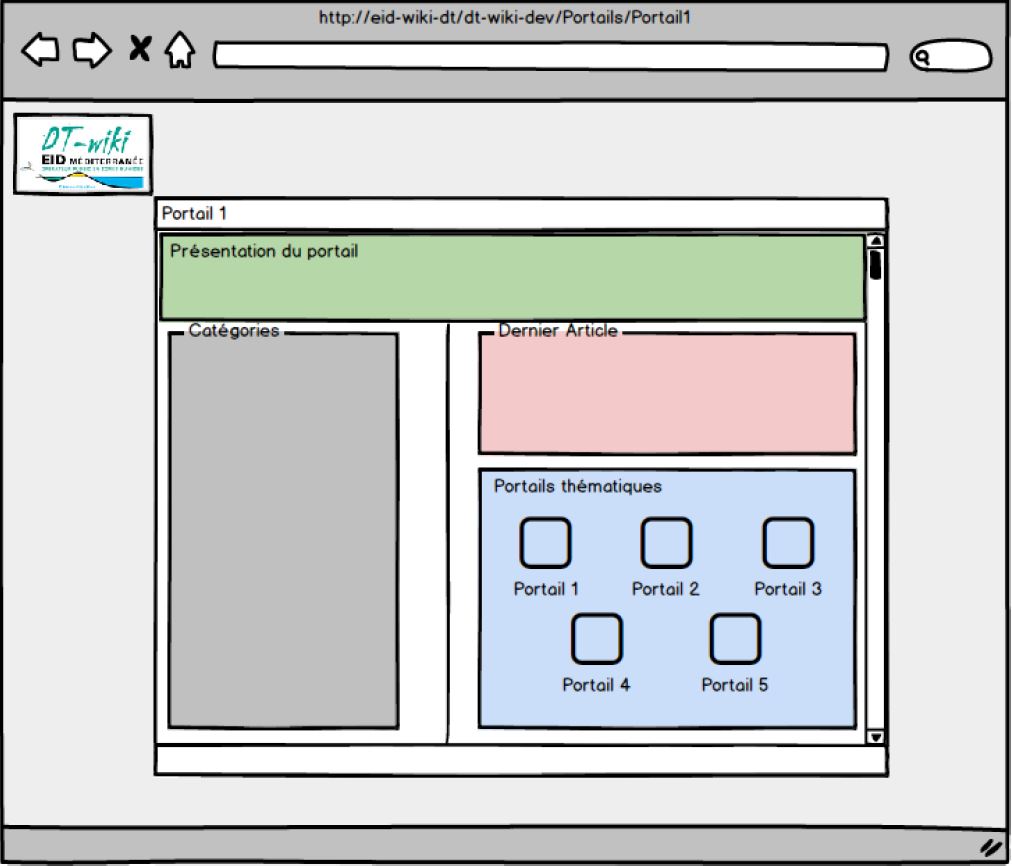


Figure : Maquette u portail thématique

##### 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 nommé « Dernier article ».

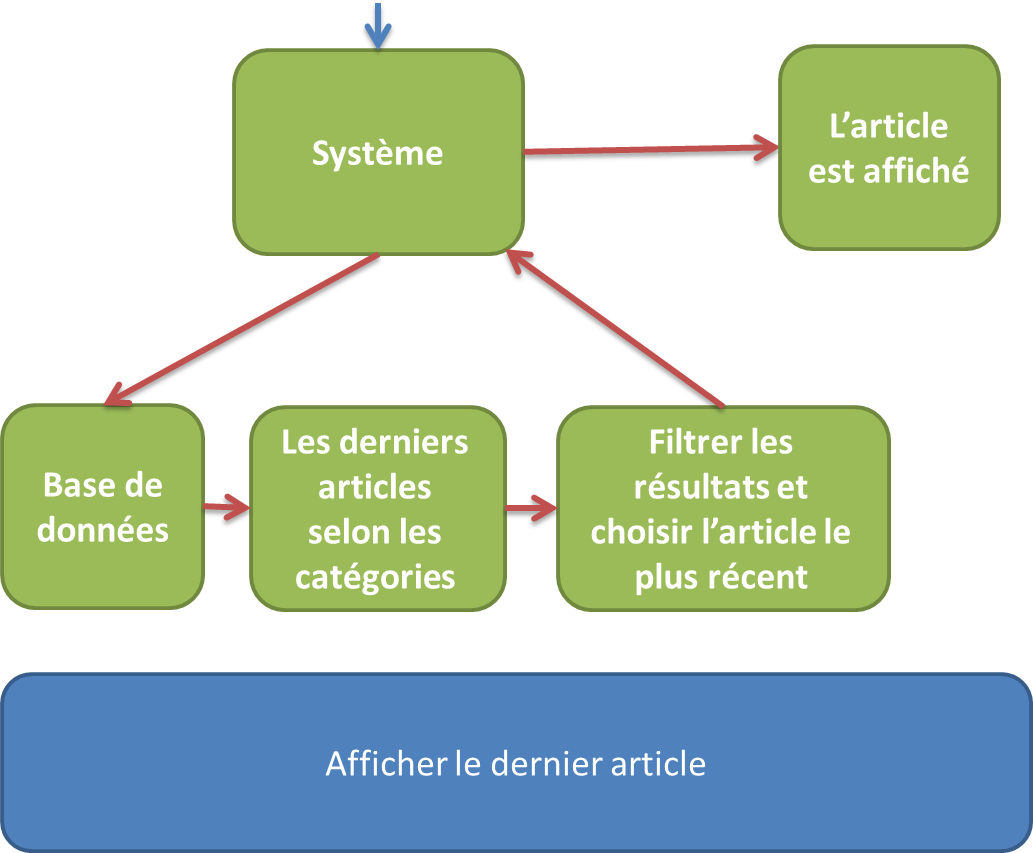


Figure : Architecture d'afficher le dernier article

##### Rechercher un article :

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

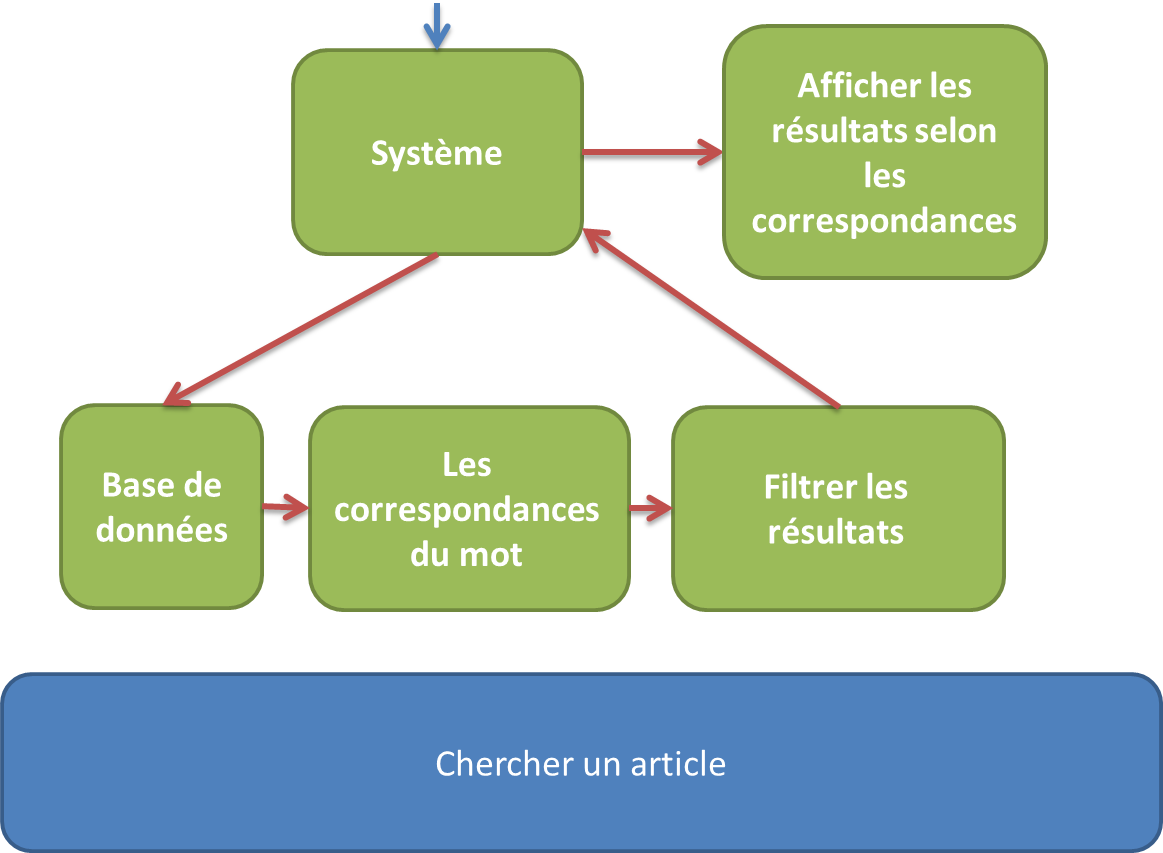


Figure : L'architecture de chercher un article

#### 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.

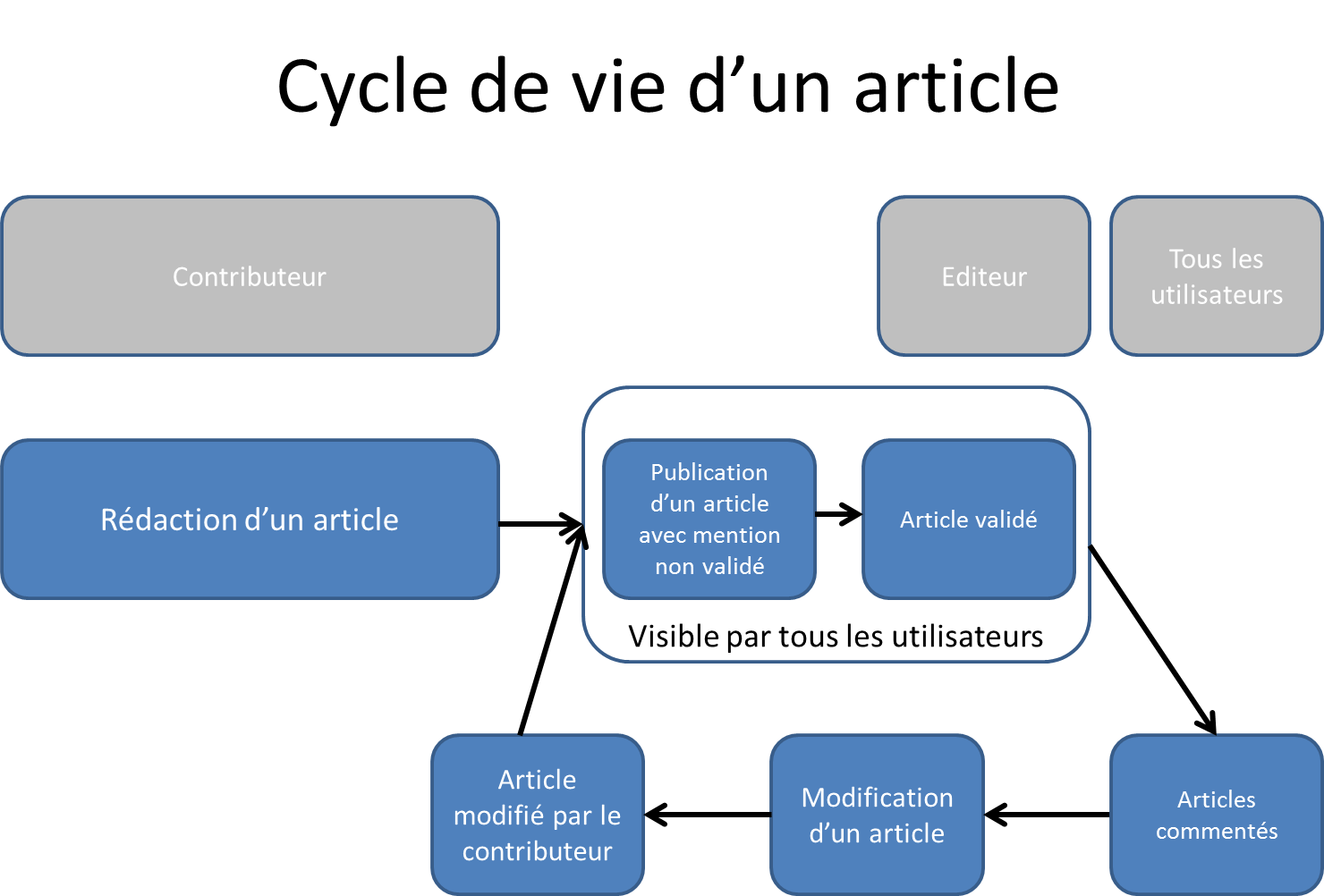


Figure : Cycle de vie d'un article

##### 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 apparait, elle permet de choisir le modèle d’article à créer.

Après la validation du choix du modèle, une autre interface graphique apparait 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 publier mais pas encore validé par un éditeur. Le système envoi un email au contributeur qui a rédigé l’article pour indiquer que l’article est créé avec succès.

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

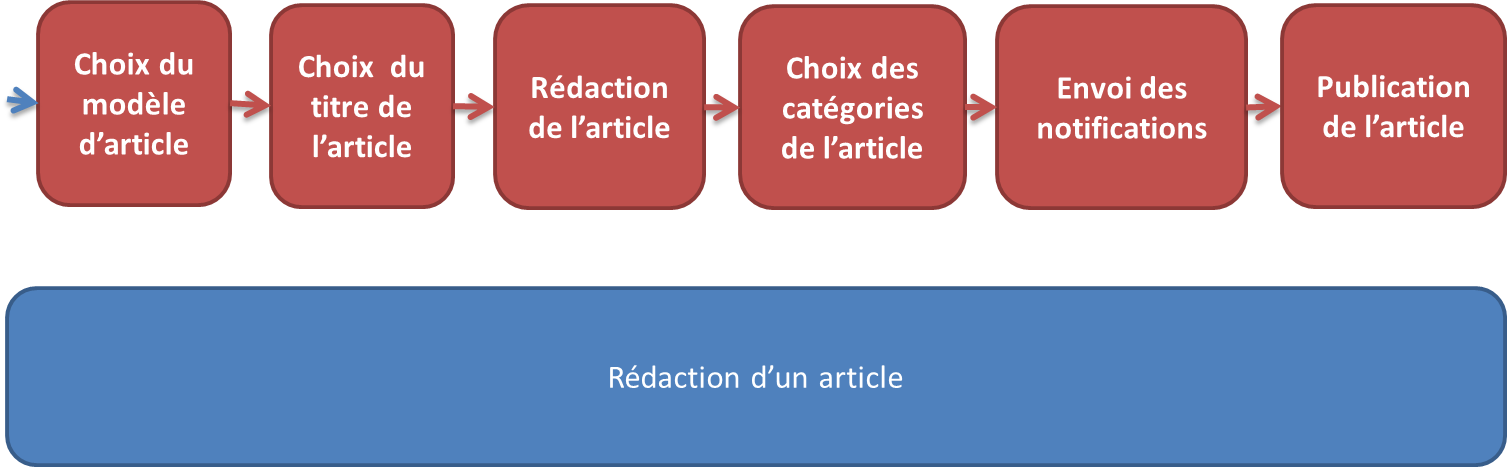


Figure : Architecture de création d'un article

##### 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.

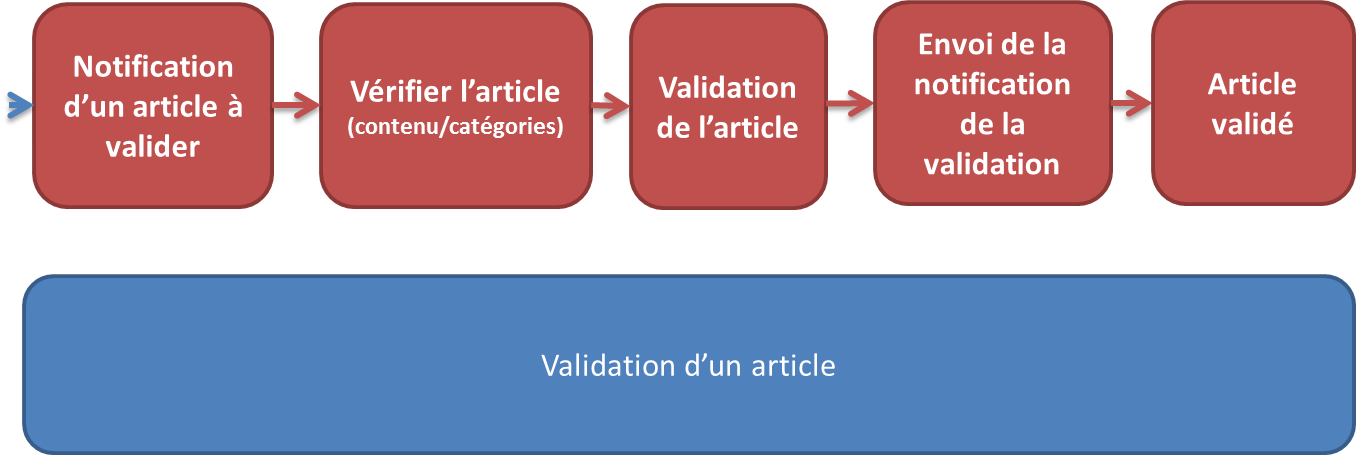


Figure : Architecture de validation d'un article

##### 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.

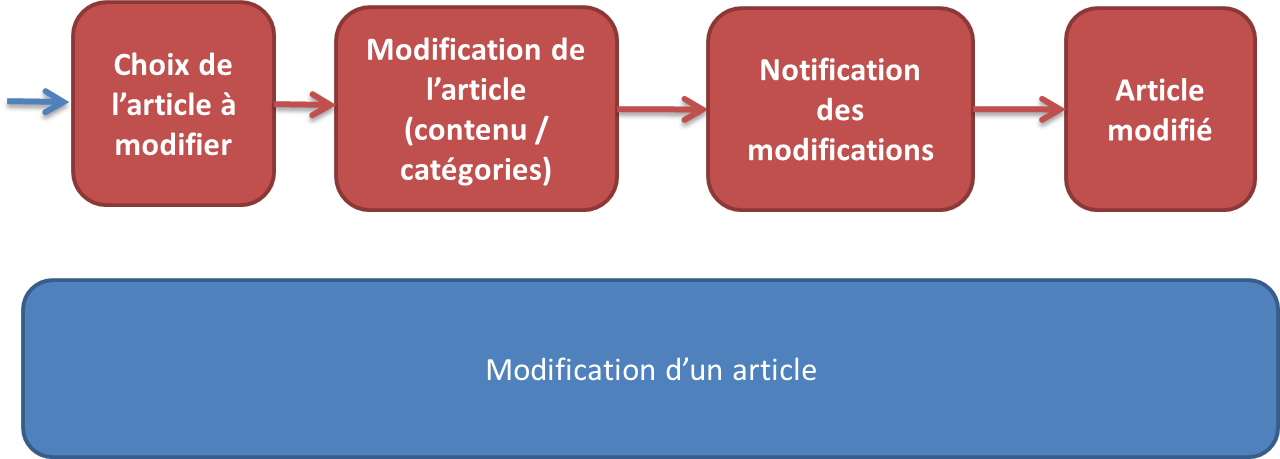


Figure : Architecture de modification d'un article

##### 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.

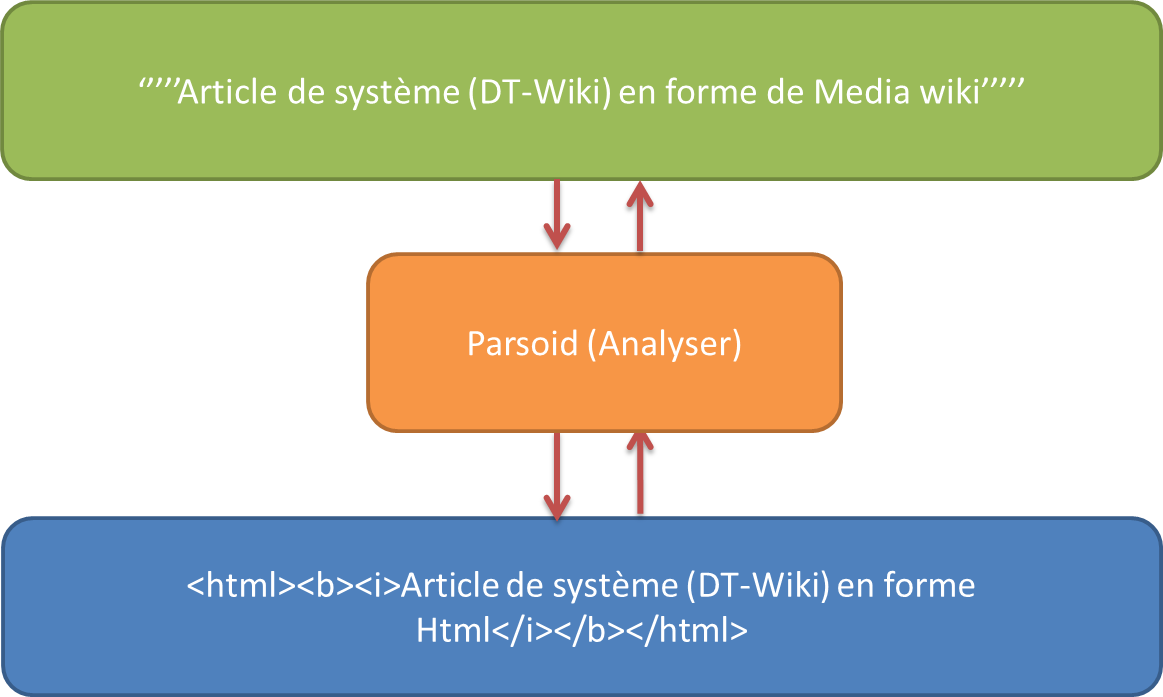


Figure : Architecture d'analyseur du système (DT-Wiki)

#### Indexation des articles :

##### 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 choisi 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).

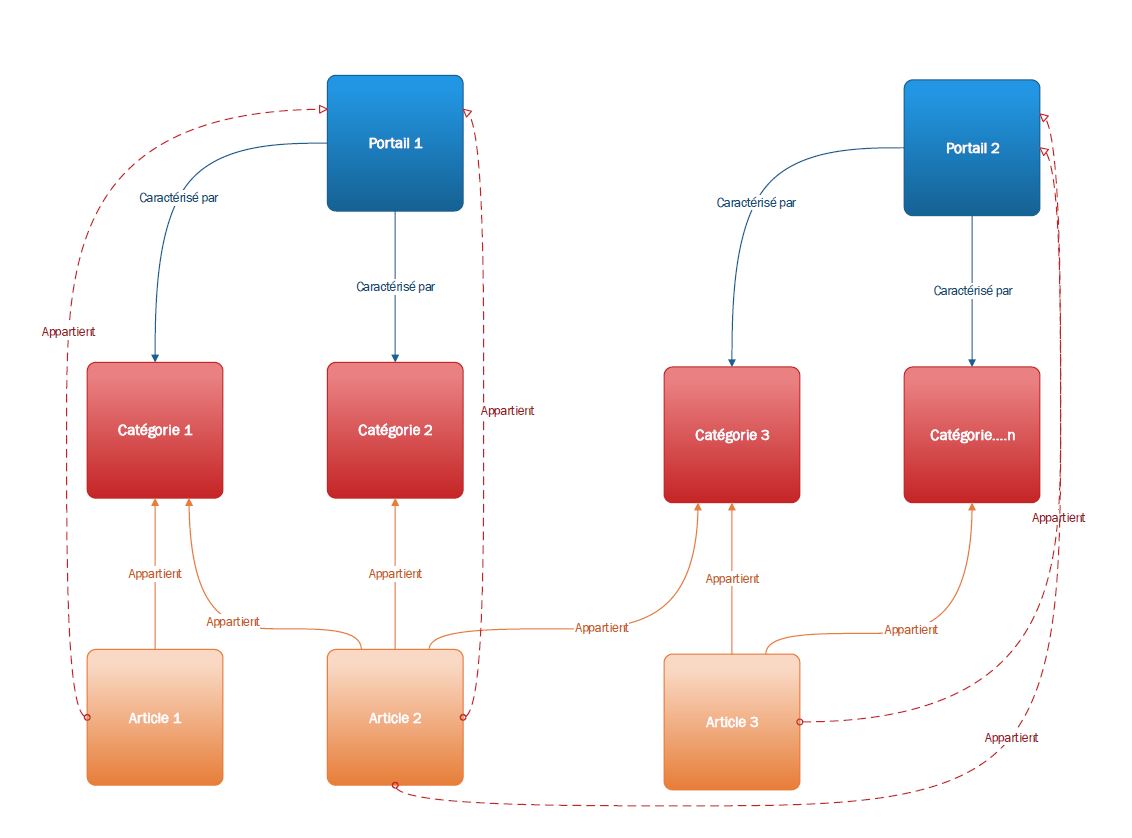


Figure  : Structure d'indexation des articles

##### 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.

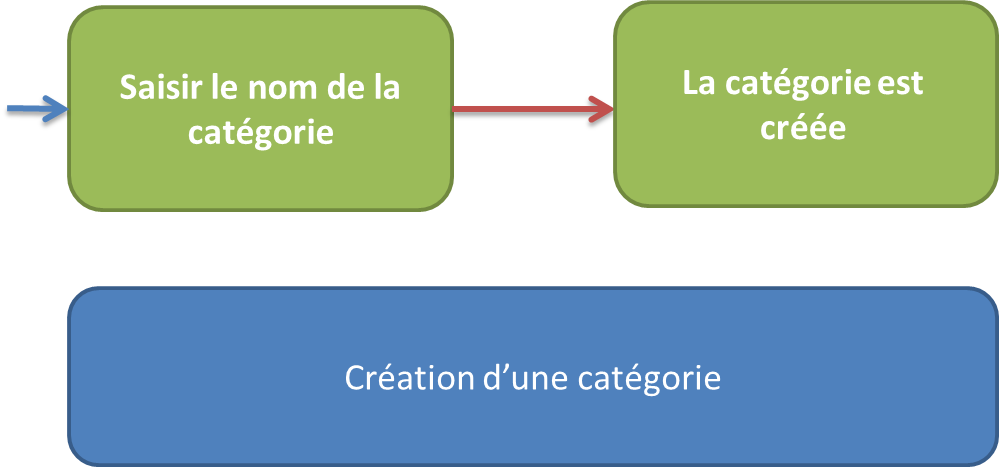


Figure : Architecture de création d'une catégorie

##### 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.

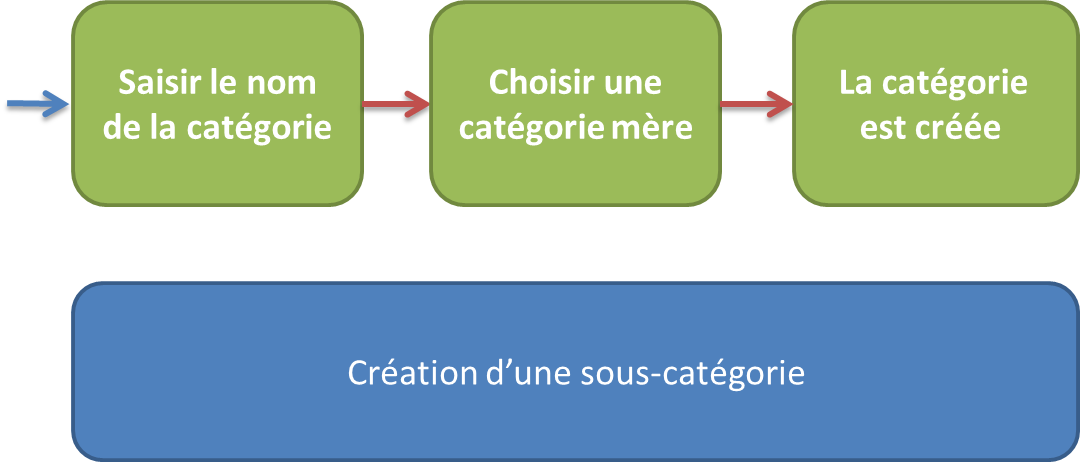


Figure : Architecture de création d'une sous-catégorie

##### 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.

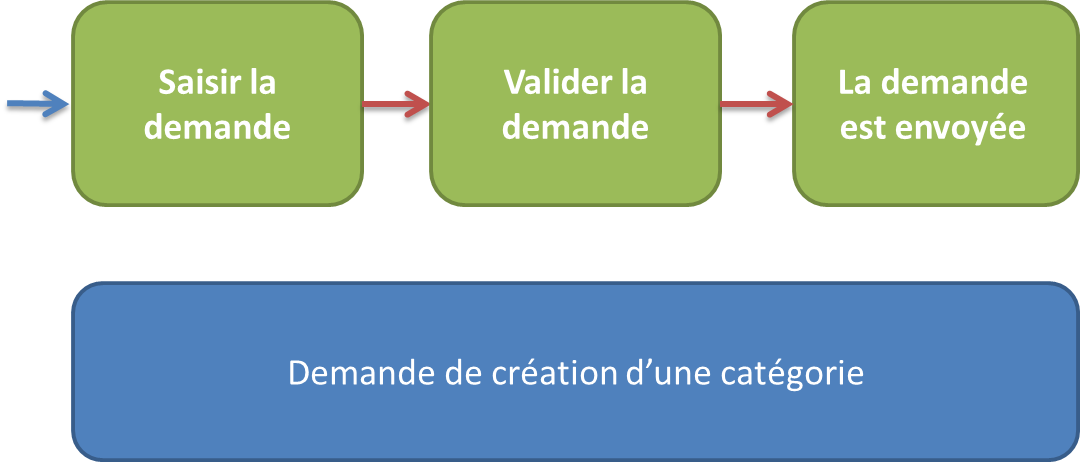


Figure : Architecture de demande de création d'une catégorie

## Développement :

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

### Environnement de développement :

#### Media Wiki version 1.24.2:

Media Wiki est un Framework utilisé afin de faciliter le développement des wiki, en bref le dernier fournit une architecture de base simple et ouvert d’un wiki.

***«*** ***Media Wiki est un ensemble***[***wiki***](http://fr.wikipedia.org/wiki/wiki)***à base de***[***logiciels libres***](http://fr.wikipedia.org/wiki/logiciel_libre)[***Open source***](http://fr.wikipedia.org/wiki/Open_source)***, développé à l’origine pour***[***Wikipédia***](http://fr.wikipedia.org/wiki/Wikip%C3%A9dia)***et aujourd’hui utilisé par plusieurs autres projets de l’association à but non lucratif***[**Wikimédia Fondation**](http://wikimediafoundation.org/wiki/Accueil)***ainsi que par***[***beaucoup d’autres wikis***](http://www.mediawiki.org/wiki/Special:MyLanguage/Sites_using_MediaWiki)***. »5.***

Cet outil est choisi par le service SI (Système d’information) comme un logiciel de base (Framework) de développement du système DT-Wiki.

Je valide ce choix car après une recherche, je trouve que le logiciel media wiki est le logiciel le plus célèbre et adaptatif à nos besoins à cause de grande communauté, plusieurs documentations en différentes langues, possibilité d’ajout des extensions, et enfin il fournit quelque outils par défaut comme :

* L’éditeur Wiki qui utilise un langage de script spécial pour mise en forme le contenu d’une page.
* La possibilité de n’importe qu’elle utilisateur d’ajouter ou de modifier les pages c’est-à-dire tout le système est ouvert pour tous les types des utilisateurs encore les utilisateurs ne sont pas inscrit.

Malgré ça, les outils fournie par default dans le logiciel media wiki a pour rôle de faciliter la mise en œuvre du wiki. Mais dans notre cas on a rencontré des difficultés, à cause les besoins du client sont contre la philosophie de wiki et encore l’utilisateur du notre système ne sont pas des informaticiens donc on ne peut pas utilisés des langages des balises pour mettre leurs contenu. Ensuite les extensions existe ne sont pas capable de résoudre notre problèmes car il y a beaucoup des extensions contient des bugs et ne fonctionne pas, par ailleurs les besoins du client sont spéciales est contre la philosophie du wiki par exemple restriction d’accès aux pages selon le profil d’utilisateur, des utilisateurs spéciales reçu des notifications par email lorsque la création d’un article, seulement les éditeurs permet de validé les articles, seulement les administrateurs permet d’ajouter des utilisateurs au système, le système n’est pas accessible que par des membres ont des login et mot de passe. Donc le système n’est pas un wiki exacte mais est un autre système hérite de notion du wiki.

Pour résoudre ces problèmes, je commence par lire les documentations de media wiki pour connaitre l’architecture globale du logiciel puis je commence à modifier l’architecture de logiciel c’est-à-dire modifier le schéma de la base de données (ajouter et modifier des relation), en suite ajouter et modifier des blocs de codes afin que le système (DT-Wiki) satisfaire les besoins du client, ainsi j’utilise des technologies et des langages de programmation web associés (adapté) avec le logiciel media wiki sont les suivants :

* **Modèle MVC :**

***« L’architecture MVC******(*modèle*,*vue *et*contrôleur) *est un concept très puissant qui intervient dans la réalisation d’une application. Son principal intérêt est la séparation******des******données (*modèle*), de l’affichage (*vue*) et des actions (*contrôleur*). »15.***

Je conserve ce patron de conception, car les fichiers de base du Framework media wiki sont regroupés et séparés selon l’architecture MVC ce que facilite le développement.

* **PHP 5 :**

PHP est un langage script utilisé dans le développement web pour dynamisé le contenu des pages web.

***« PHP (officiellement, ce sigle est un acronyme récursif pour*PHP: Hypertext Preprocessor*) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web.7.***

Ce langage est utilisé car tous les fichiers des modèles et des contrôleurs de Framework media wiki sont écrits avec le langage PHP 5.

* **HTML :**

HTML (HyperText Markup Language): utilisé pour représenter des pages web avec l’utilisation des balises spéciales autour des textes.

***« HTML is a markup language for describing web documents (web pages). »8***

Ce langage est utilisé car toutes les vues du Framework media wiki sont écrites avec le langage HTML.

* **CSS :**

Le CSS utilisé pour la mise en forme des styles pour les pages écrites en HTML.

***« Les feuilles de styles (en anglais "*Cascading Style Sheets*", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web.»9.***

Ce langage est utilisé car tous les fichiers de mise en forme des vues de contrôleur de Framework media wiki sont écrits en CSS.

Ensuite, pour stocker les données du système (DT-Wiki), il faut l’intégrer avec un SGBD, dans notre cas :

* **Base de données PostgreSQL :**

PostgreSQL est un système de gestion de base de données open source et gratuit.

***« PostgreSQL est le système de gestion de base de données Open source le plus sophistiqué du marché. Développé activement depuis plus de quinze ans par une communauté internationale »6.***

Cet outil est choisi par le service SI (Système d’information) comme un logiciel de base de données pour le développement du système DT-Wiki.

Enfin, pour mettre en œuvre le système, il faut l’intégrer et l’héberger sur un serveur comme dans notre cas le développement se fait directement sur le serveur :

* **Serveur Web XAMPP :**

***« XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s’agit d’une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d’utilisation, réputée pour son installation simple et rapide.»14.***

Cet outil est choisi par le service SI (Système d’information) comme serveur web du système DT-Wiki.

* **Serveur physique :**

Le serveur est installé avec le système d’exploitation Microsoft Windows Server R8 pour héberger l’application web sur le réseau local du l’EID méditerranée.

Cet outil est choisi par le service SI (Système d’information) pour mettre le système DT-Wiki en œuvre.

Le choix de tous ces outils me parait pertinent et ne pose pas des problèmes car j’ai l’expérience de travailler avec eux.

#### Technologies additionnels :

Lors du développement du système (DT-Wiki), j’ai rencontré des problèmes avec les technologies précédentes car ces technologies ont des limites, par exemple le PHP est exécuté côté serveur mais il ne peut pas vérifier les données entrées par l’utilisateur avant l’envoi des données au serveur.

Afin de corriger ces problèmes, j’ai eu besoin d’utiliser d’autres technologies que j’ai choisi selon les problèmes rencontrés.

Ces autres technologies et langages qui sont les suivantes :

* **JavaScript :**

Pour dynamiser les pages localement et contrôler les données entrées par l’utilisateur, j’ai choisis d’utiliser le « JavaScript ».

Le JavaScript est un langage script pour dynamiser les pages web et qu’il est exécuté côté navigateur du client.

***« JavaScript est un langage de script orienté objet principalement utilisé dans les pages***[***HTML***](http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-html-480/)***. A l'opposé des langages***[***serveurs***](http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-serveur-1950/)***(qui s'exécutent sur le site), JavaScript est exécuté sur l'***[***ordinateur***](http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/informatique-ordinateur-586/)***de l'internaute par le***[***navigateur***](http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-navigateur-3986/)***lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions.»10.***

* **AJAX :**

J’ai choisi de travailler avec la technologie « AJAX » afin de synchroniser les requêtes entre les pages web, par exemple exécuter des codes PHP trouvé côté serveur à partir de code JavaScript trouvé côté client.

AJAX est une technologie utilisée pour synchroniser les données entre le serveur et le client.

***« Asynchronous JavaScript And XML (AJAX) : il désigne un nouveau type de conception de pages Web permettant l'actualisation de certaines données d'une page sans procéder au rechargement total de cette page. Cette méthode de conception repose sur la combinaison de technologies déjà existantes : HTML/CSS, JavaScript/DOM, XML et les requêtes HTTP. »11.***

* **JQuery :**

Afin de faciliter l’interaction entre le code JavaScript et la technologie Ajax, j’ai choisis d’utiliser la bibliothèque JavaScript « JQuery ».

JQuery est une bibliothèque de code JavaScript qui permet de faciliter l’utilisation des codes JavaScript.

***« JQuery******est un Framework******JavaScript******sous licence libre qui permet de faciliter des fonctionnalités communes de***[***JavaScript***](http://glossaire.infowebmaster.fr/javascript/)***. »13.***

### Développement des besoins spécifiques:

Après la conception des fonctionnalités, le choix de l’environnement du travail et les outils nécessaires pour le développement, je commence à développer et réaliser le système DT-Wiki :

#### 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.



Figure : Media wiki après l'installation

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 :

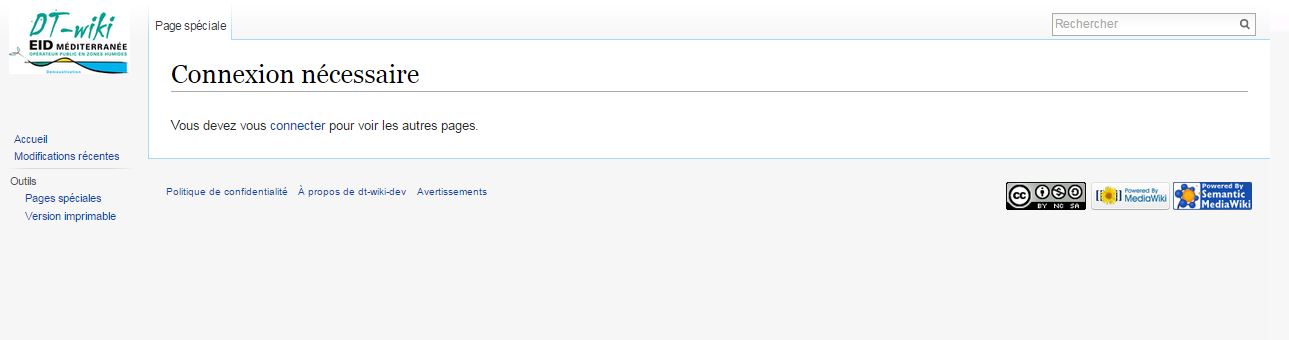


Figure : Empêcher l'accès

* 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.

#### Développement de la gestion des utilisateurs :

##### La structure des droits des utilisateurs :

Dans le logiciel media wiki, il n’y a pas une notion des types des utilisateurs. Pour cela j’ai créé les groupes des utilisateurs (Administrateur, Modérateur, Editeur, Contributeur, Utilisateur) par l’association à chaque groupe les fonctionnalités qui a le droit de manipuler par exemple : mettre seulement pour les administrateurs le droit de créer des utilisateurs.

Et l’ajout d’une colonne dans la base de données pour sauvegarder le rôle de l’utilisateur.

##### Création d’un utilisateur :

J’ai commencé par la vérification de système d’inscription par défaut en media wiki : n’importe qui permet de l’inscrire et mettre leurs informations de façon facultative.

Mais, j’ai trouvé que ce système n’est pas satisfaire le besoin de l’équipe du projet : seule 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 à base de l’ancien système.

Une illustration de cette fonction est donnée dans la figure 21 :

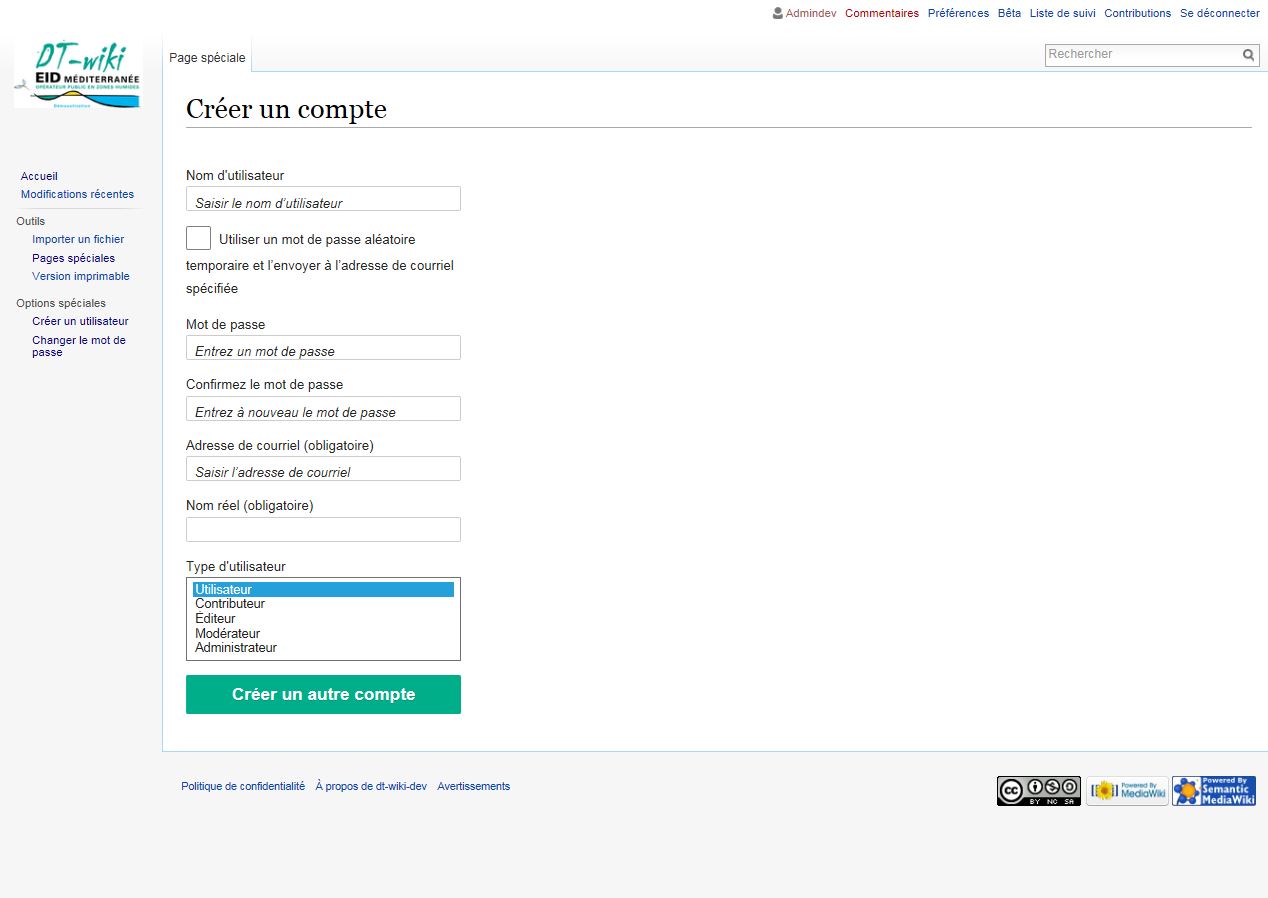
****

Figure : La page de création d'un utilisateur

##### Changer le mot de passe :

J’ai commencé par créer une page 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 information de l’utilisateur puis manipule un script PHP pour vérifier que l’ancien mot de passe est correcte puis changer le avec le nouveau mot de passe.

J’ai fini cette fonction et une illustration est donnée dans la figure 21 :

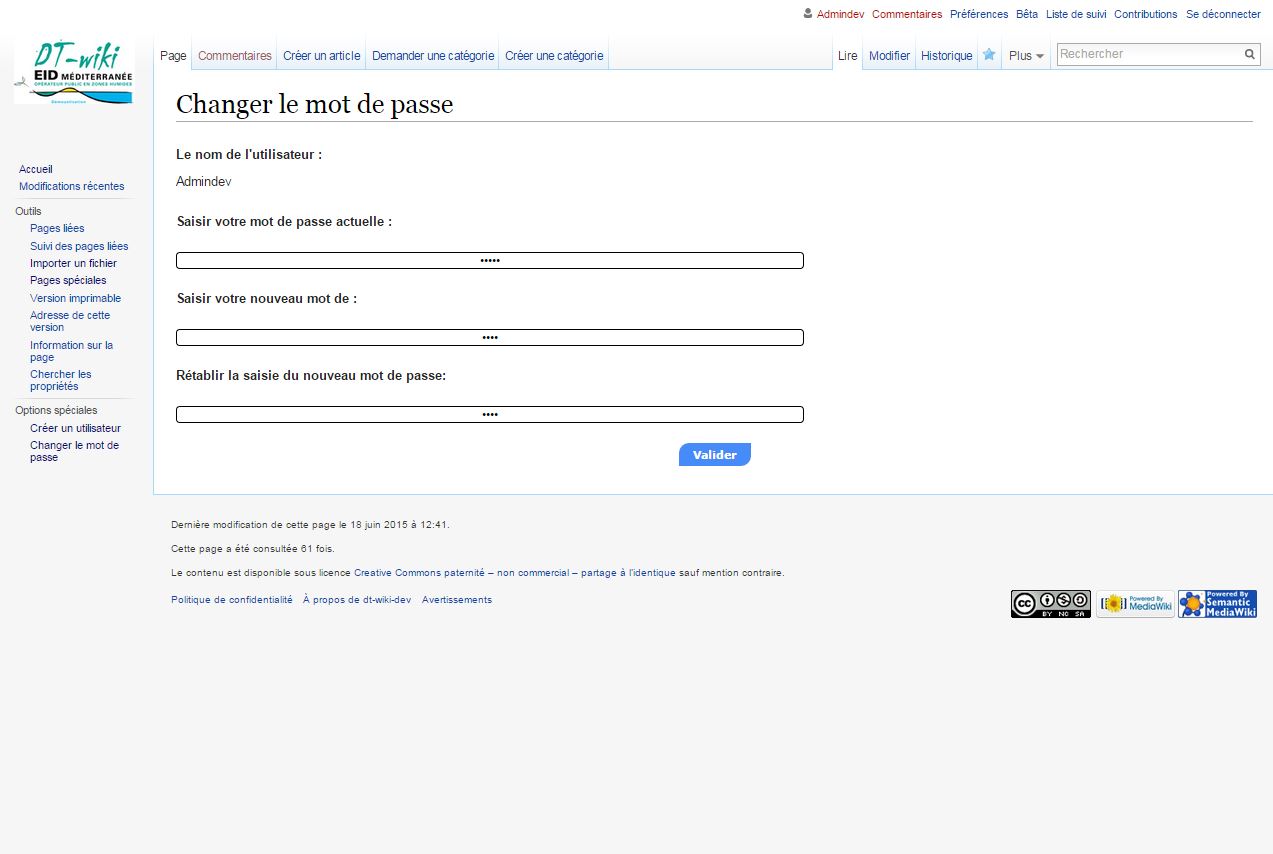


Figure : Changer le mot de passe

#### Développement du l’architecture de système :

##### 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 n’est pas satisfaire la création des pages avec de mise en forme avancé. Pour cela j’ai changé le cœur de media wiki pour je peux utiliser l’Html et le CSS dans la mise en forme des pages.

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

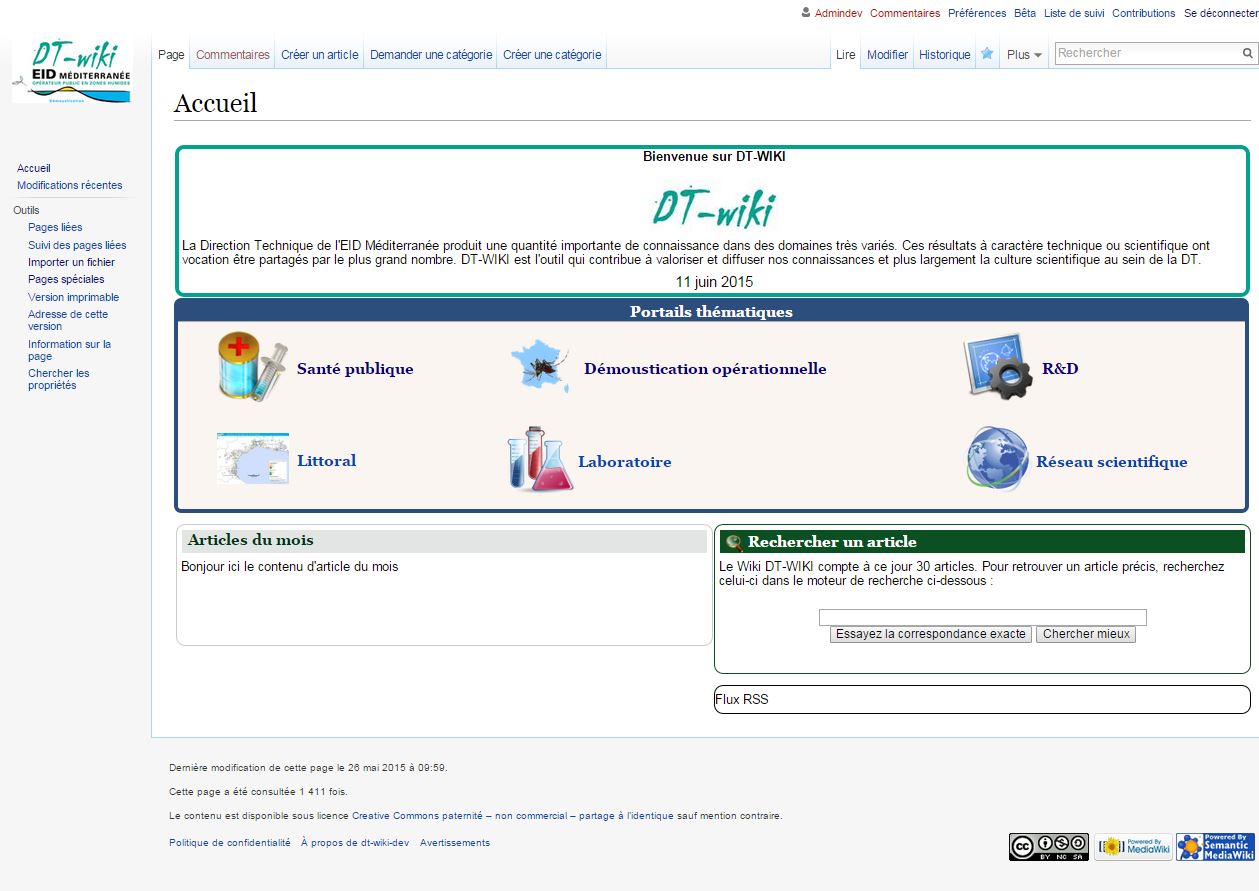


Figure : Page d'accueil u DT-Wiki-Dev

##### 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 « Hoocks » : un code PHP est injecté dans un autre fichier par l’appel d’un Hoock se trouve dans le fichier) et encore le contrôle de flux binaire c’est-à-dire dans l’affichage injecte 256 bits du texte originale d’un article.

Malgré quelque problème rencontré, j’ai fini cette fonctionnalité et une illustration de cette fonction est donnée dans la figure 23 :

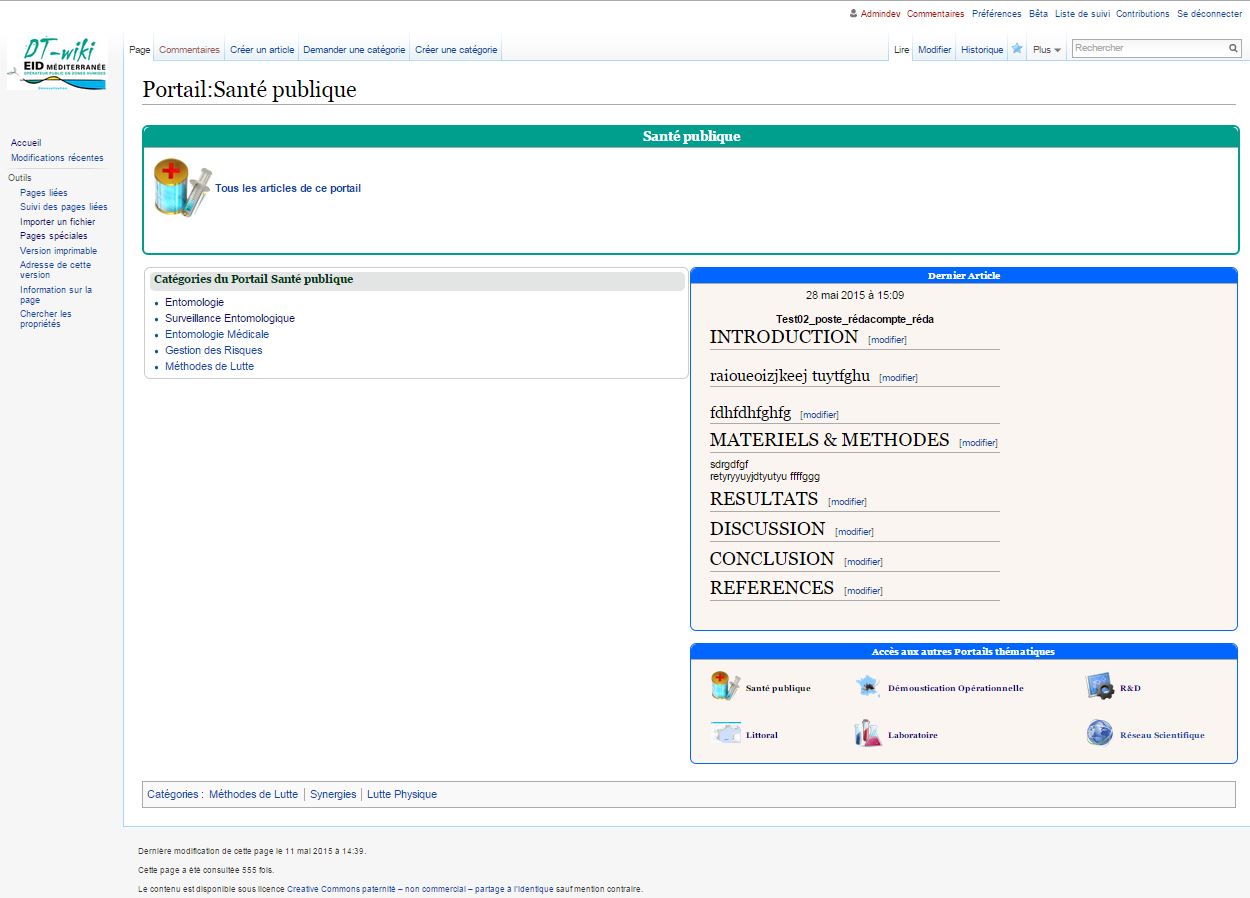


Figure : Exemple de dernier article

##### 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 qui je n’ai pas intégralement développé. 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.

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



Figure : Exemple de chercher un article

#### Développement du Cycle de vie de l’article :

##### Création d’un article :

Après la conception de cette fonctionnalité, j’ai commencé par développer intégralement cette fonctionnalité car cette fonctionnalité est spéciale c’est-à-dire n’existe pas une solution en media wiki.

J’ai réalisé une extension pour ajoute un tab en haut de la page contient « Créer un article » avec l’utilisation de notion de Hooks.

Cette tab a pour rôle de faciliter à l’utilisateur de trouver 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 contient tous les modèles d’articles possibles afin d’obliger l’utilisateur à choisir un modèle.

Et encore 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 des autres pages contiennent le formulaire de saisie le nom de l’article associé à chaque modèle.

Selon le modèle d’article et leur formulaire de saisie du nom le système génère une page dans l’espace du nom article et affiche l’éditeur de texte avec l’injection automatique de modèle d’article avec la notion de preload en media wiki.

Après que l’utilisateur rédige son article et cliquer sur le bouton « enregistrer ».

Le système affiche un pop-up listé toutes les catégories du système pour obliger l’utilisateur à choisir un ou plusieurs catégories afin d’indexer cet article c’est-à-dire 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’Ajax pour manipuler un code PHP coté serveur pour envoyer des emails de notification aux éditeurs.

Malgré quelque problème rencontré, j’ai fini cette fonctionnalité et une illustration de cette fonction est donnée dans la figure 25

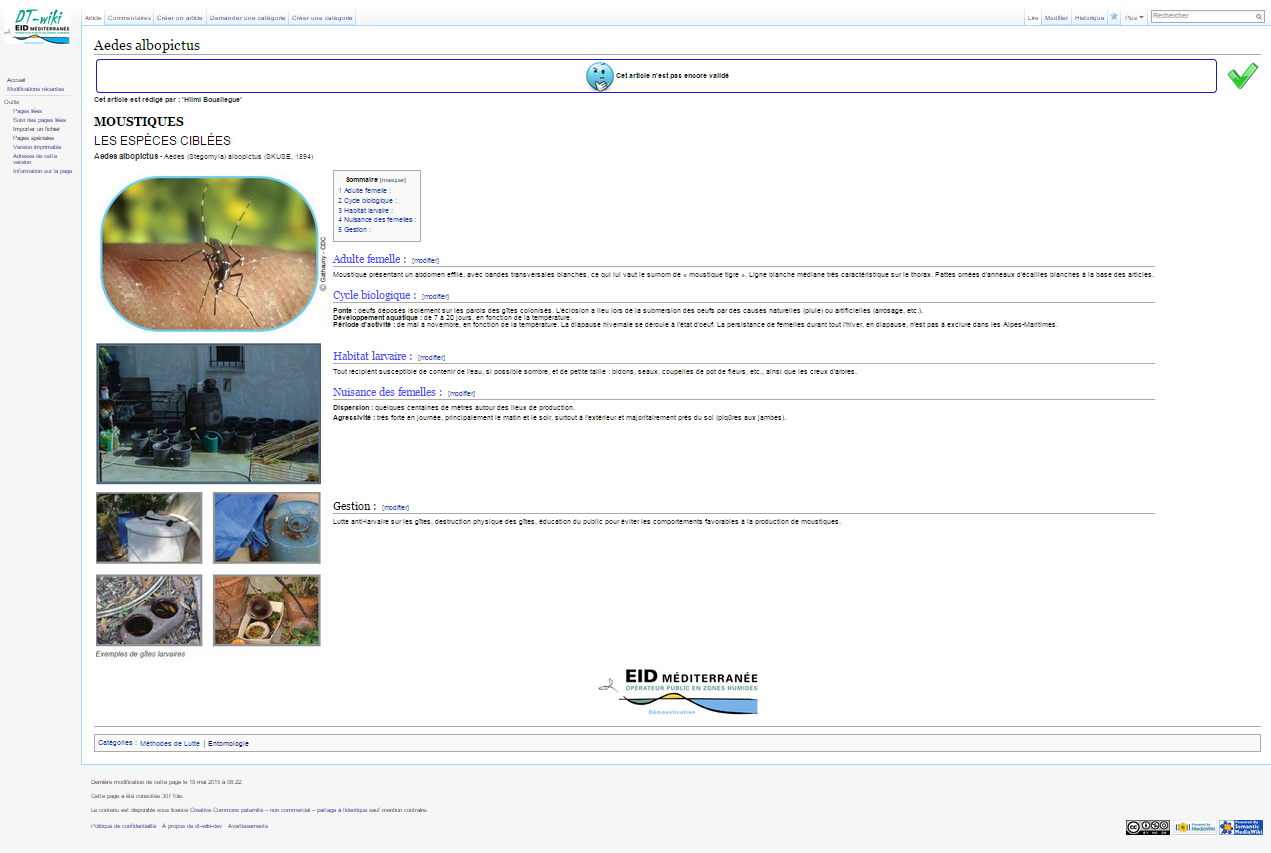


Figure : Exemple d'un article créé

##### Validation d’un article :

J’ai commencé à développer cette fonctionnalité intégralement car cette fonctionnalité est spéciale.

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 cliqué 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 26 :

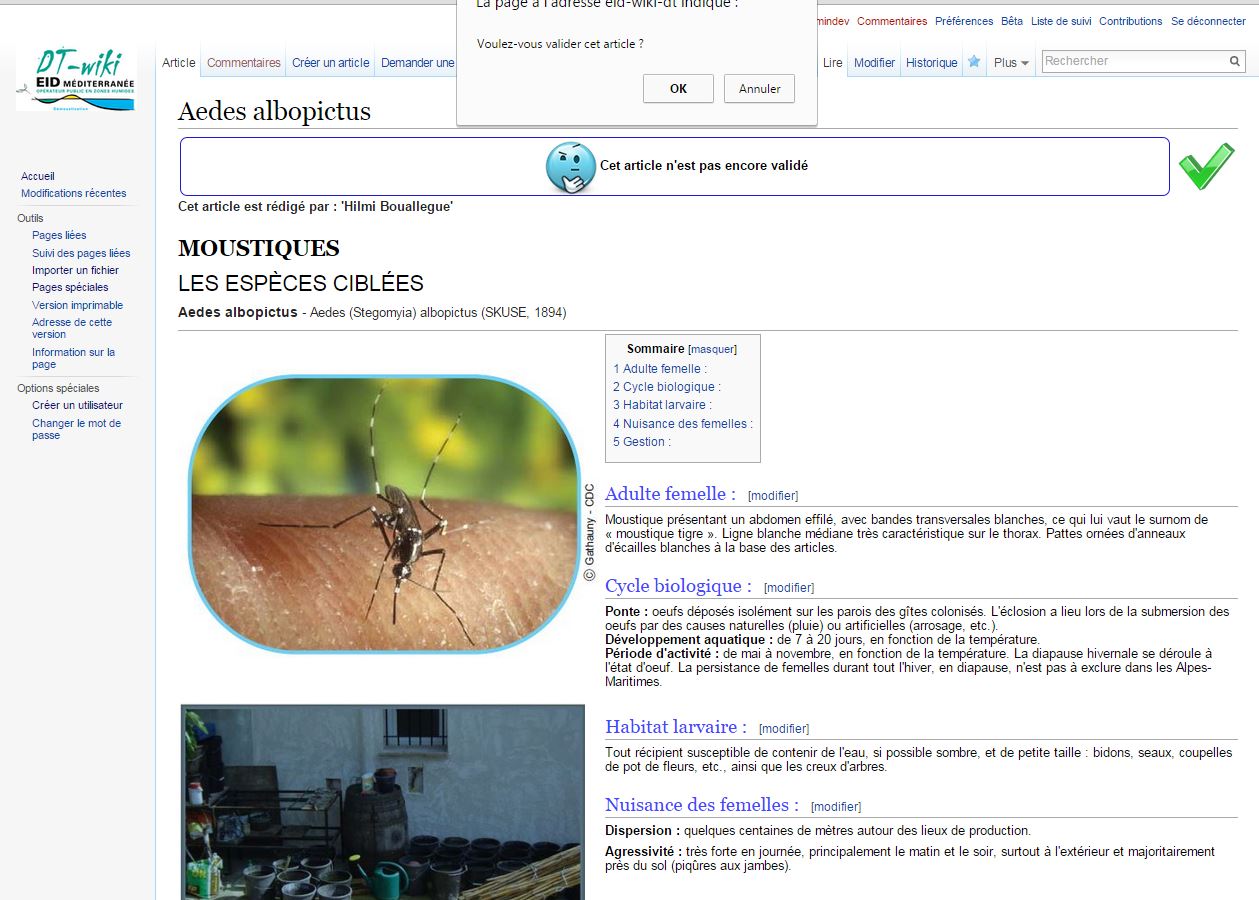


Figure : Exemple de validation d’un article par l'éditeur

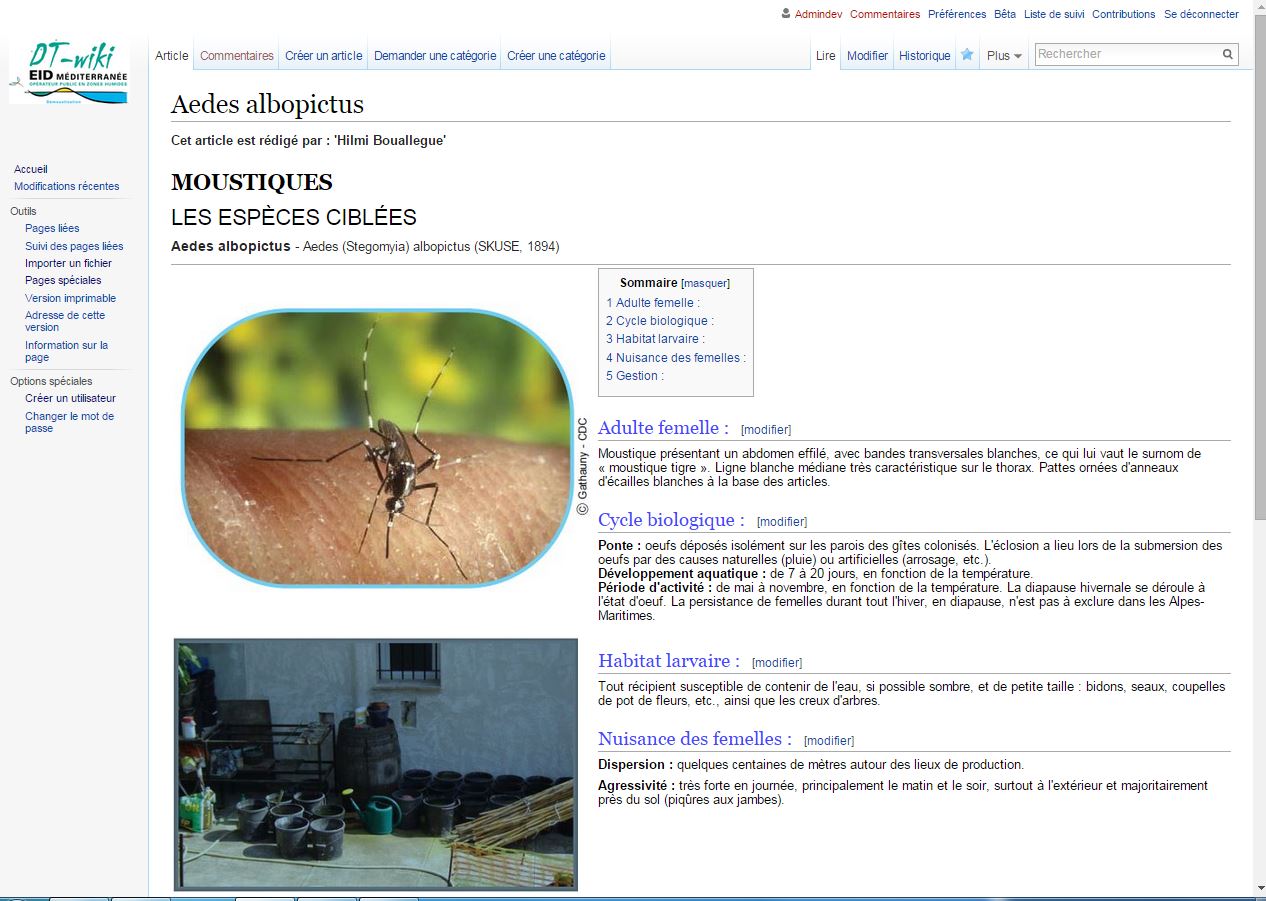


Figure : Exemple d'un article après la validation

##### Modification d’un article :

J’ai développé cette fonctionnalité intégralement car cette fonctionnalité est spéciale.

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

Après que l’utilisateur 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 cliquer sur le bouton « Enregistre ».

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 valider et retransmettre des emails de notification aux éditeurs.

J’ai fini cette fonctionnalité.

##### 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 quelque solution mais ne sont pas efficace et contiennent beaucoup de bugs.

J’ai utilisé un squelette d’un éditeur de texte existe et j’ai modifié quelque bouton 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 quelque problème rencontré, j’ai développé cette fonctionnalité j’ai fini cette fonctionnalité et une illustration de cette fonction est donnée dans la figure 29 :



Figure : Parsoid pour le système (DT-Wiki)

#### Développement du système d’indexation des articles :

##### Création d’une catégorie :

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

J’ai créé une extension pour ajouter une page spécial pour créer des catégories.

J’ai limite l’accès à cette page seulement pour le rôle administrateur et modérateur.

Dans cette page l’utilisateur permet de saisir le nom de catégorie puis valider.

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

Une illustration de cette fonction est donnée dans la figure 30 :

****

Figure : La page de création d'une catégorie et sous-catégorie

##### Demande de création d’une catégorie :

J’ai commencé à développer cette fonctionnalité car c’est une fonction spéciale.

J’ai créé une extension pour ajouter cette page comme une tab en haut de la page pour faciliter l’accès à cette page car cette fonctionnalité est 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 contient le demande et les information du demandeur.

J’ai fini cette fonctionnalité et une illustration est donnée dans la figure 31 :

:

****

Figure : La page de demande de la création d'une catégorie

## Test et déploiement :

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

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.
* d’établir la confiance pour le client et encore pour l’équipe de projet.

Ensuite, ces tests ont été réalisés selon trois approches :

### 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 encore les défauts du système.

#### 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 graphique,…).

#### 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.

#### Tests système :

Après les tests de composants, j’intègre les derniers afin de tester les interactions entre ces composantes et encore vérifier si les composants sont interagisse correctement et transfert les bons données.

### Tests de version :

Le test de version est réalisé par les membres de l’équipe du projet, mais pas les membres de l’équipe de développement afin de tester une version particulière avant qu’elle soit préparée pour les utilisateurs afin de satisfaire les exigences.

### Tests utilisateur :

Le test utilisateur est réalisé par un échantillon d’utilisateurs afin de tester le système dans leurs propres environnements car l’utilisateur est le responsable à la décision concernant l’acceptabilité du système.

## Formation et perspective :

Jusqu’à cette itération de développement, le système n’est pas totalement complet car il y a des autres fonctionnalités pas encore finir par exemple les recherches avancées des articles,..

Ensuite, les futurs utilisateurs du système (DT-Wiki) ne sont pas des informaticiens d’ailleurs les fonctionnalités du système sont des fonctionnalités spéciales (personnalisées), pour cela il faut réaliser une documentation « Manuelle d’utilisations du système DT-Wiki ».

Ensuite, il faut réaliser une documentation pour la configuration du système.

# Conclusion :

Après la réalisation du stage dans l’EID méditerranée, afin de développer le système DT-Wiki pour la valorisation et de la diffusion des connaissances en interne de la direction technique.

J’ai pu m’intéresser l’intégration dans le domaine de travaux professionnels et l’interaction avec des membres du projet qui sont des gens du domaine de l’informatique et des autres hors ce domaine.

Ensuite, malgré les difficultés rencontrées durant le stage :

* Les contraintes du temps(le déphasage entre les calendriers du planning et les calendriers réels).
* La gestion du projet (j’ai travaillé dans tous les phases du projet (un concepteur, développer, testeur).
* Les limites des technologies maitrisées(en cas de nécessité je maitrise un autre outil informatique).
* La nécessité des beaucoup des tests pour minimiser les nombres des bugs.

J’ai réussi de réaliser les taches du développement du système :

* La Conception et la documentation.
* Le développement et le test.

Et j’ai mis une version exécutable, contient les fonctionnalités du base et les nécessaires du système pour tester par les futurs utilisateurs, afin de produire une version finale (la version de la production) du système (DT-Wiki).

Encore, j’ai utilisé beaucoup des outils qui déjà j’ai maitrisé dans la formation du master web intelligence comme le cours de génie logiciel pour la conception et la gestion du projet, et la cour de développement web avancé, le cours de l’interopérabilité,…

Plus largement, ce stage a été l’opportunité pour moi d’appréhender la communication avec les clients, extraire des exigences et les souhaits des clients, comprendre la vision de client, la rédaction des documents professionnelles, l’utilisation des nouveaux technologies et langages de programmations que je les utilise pour la premier fois, apprendre à partir les fautes qui j’ai les fais, l’intégration dans des réunions d’avancement avec l’équipe de projet, apprendre des nouveaux idées à partir des autres membres plus expérimenté.

# Annexe et bibliographies :

## Bibliographies :

4 : Stage\_wiki\_EID\_MED

5 : Guide\_IEEE\_Pour\_la\_Specification

## Webographies :

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

2 : <http://www.eid-med.org/page/financements> (consulté le 03/06/2015).

3 : <http://www.eid-med.org/page/missions> (consulté le 03/06/2015).

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

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

8 : <http://www.journaldunet.com/developpeur/expert/56616/la-methode-agile---optimisation-de-la-relation--client---fournisseur.shtml> (consulté le 05.06.2015).

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

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

11: <http://www.mediawiki.org/wiki/MediaWiki/fr> (consulté le 06/06/2015)

12 : <http://www.dalibo.com/postgresql> (consulté le 06/06/2015).

13 : <https://php.net/manual/fr/intro-whatis.php> (consulté le 06/06/2015).

14 : <http://www.w3schools.com/html/html_intro.asp> (consulté le 06/06/2015).

15: <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-css-4050/> (consulté le 06/06/2015).

16 : <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-javascript-509/> (consulté le 06/06/2015).

17 : <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/high-tech-ajax-3998/> (consulté le 06/06/2015).

18 : <http://glossaire.infowebmaster.fr/xml/> (consulté le 06/06/2015).

19 : <http://glossaire.infowebmaster.fr/jquery/> (consulté le 06/06/2015).

20 : <http://desgeeksetdeslettres.com/programmation-java/xampp-plateforme-pour-heberger-son-propre-site-web> (consulté le 07/06/2015).

21 : <http://www.guillaumevoisin.fr/internet/larchitecture-mvc-dans-le-developpement-dun-site-internet> (consulté le 07/06/2015).

Liste des mots magiques :

Sont les suivant(e)s :

Par exemple :

Tout d’abord,

Ensuite,

De plus,

Ainsi,

En parallèle,

Par ailleurs,

Par conséquent,

Par contre,

Enfin,

Finalement,

En bref,

En résumé,

L’architecture pour créer une partie :

Définition :

Présentation dans le stage : Nous avons réalisé le xxxxxx, suivant la méthode xxxxxx.

Utilisation pendant le stage : Nous avons basé les fonctionnalités sur le schéma. Cela nous a permis de réaliser et de se rendre compte des difficultés, des incohérences, …

Exemple : Droits des utilisateurs (Editeur)

Illustration : impression écran des pages spéciales

Livrable ou annexe : Document d’exigence, Tableau complet des droits utilisateurs et contraintes validé par le directeur techniques

* + Dans la partie de développement :
    - Qui a choisi chaque outil informatique,
    - pourquoi ce choix et un Exemple,
    - difficultés rencontrées et comment tu l’as résolu (recours à un autre type de langage).

Malgré ça l’exécution des instructions avec Ajax dépend de type de navigateur par exemple la méthode d’échange des données en Internet explorer est « ActiveXObject » mais pour les autre Chrome ou Firefox c’est «XMLHttpRequest ».

Pour résoudre ces problèmes, je mettre des blocs des codes pour contrôler la version du navigateur.