



RAPPORT DE PROJET

PROGRAMMATION ORIENTEE OBJET

Projet : Génération de documents

Réalisé par Ilyes Haj Ali et Peter Virlois

Etudiants en Licence Professionnelle TeProw

Année Universitaire 2016/2017



TABLE DES MATIERES

L'architecture	2
Pattern MVC	2
Modele	2
Vue.....	2
Controller.....	2
Répartition des tâches.....	3
Ilyes.....	3
Peter	3
Les fonctionnalités	3
L'interface – Présentation	3
Barre de recherche	3
L'arbre	3
Panneau latéral droit : l'édition.....	4
La génération de documents.....	4
Le design.....	5
Conclusion	5
Table des illustrations.....	6
ANNEXES.....	6
Annexe 1 : Capture d'écran de l'interface.....	6
Annexe 2 : Capture d'écran de l'arborescence	7
Annexe 3 : Capture d'écran de la génération de document	8
Annexe 4 : Capture d'écran des designs de génération de document	8
Annexe 5 : Capture d'écran de l'aperçu du rendu	9

L'ARCHITECTURE

Pour la réalisation de ce projet, nous avons utilisé le pattern MVC afin d'obtenir un découpage de l'application plus clair. Celui-ci permet notamment de faciliter la maintenance, de corriger les bugs ou encore améliorer le projet.

PATTERN MVC

MODELE

Nous avons créé deux modèles : un pour les catégories et un pour les portions de texte (et ceux pour la génération de document).

Les modèles se trouvent dans le dossier Application/Models.

La catégorie possède un `id` de type integer, un `id_parent` de type integer qui fera référence à une autre catégorie (ce qui permettra de créer une arborescence) et enfin un `libelle` de type string, qui sera la clef primaire avec l'`id`.

La portion de texte contiendra 4 attributs :

- Un `id` de type integer
- Un `texte` de type String
- Un `idCatégorie` de type integer, pour relier la portion à une catégorie
- Un attribut `keywords` qui correspond au mot-clé

Chacun des modèles possède les setters et getters correspondant à leurs attributs qui permettront de mettre à jour ou d'afficher les différentes catégories et portions de texte.

De plus, chaque modèle contient plusieurs méthodes permettant de mettre à jour, d'ajouter, supprimer etc... qui seront utiliser dans la vue.

VUE

Dans un premier temps, une première version avec une interface console avait été développée. Celle-ci permettait une gestion des catégories (ajout, suppression, afficher toutes les catégories ou une catégorie en spécifiant le libelle). Ensuite, une interface graphique a été implémentée qui permet une gestion complète des catégories et des portions de texte.

Les différents fichiers concernant la vue se situent dans le dossier Application/View.

CONTROLLER

Le rôle du contrôleur est celui de faire le contrôle entre les informations provenant des vues (interfaces) et leurs effets sur la Base de données. Ce contrôleur est appelé notamment lors ajouts/suppressions/éditions de catégories ou portion et texte et traite les informations de l'utilisateur avant d'interagir avec la base de données. Son rôle est d'assurer l'intégrité de la base de données.

REPARTITION DES TACHES

ILYES

Ilyes s'est principalement occupé de l'interface graphique et ses fonctions, ainsi que la création de la hiérarchie des catégories. Cette partie étant très complète, vous pouvez voir les différentes fonctionnalités réalisées par Ilyes [en cliquant ici](#).

Erreur ! Source du renvoi introuvable.**Erreur ! Source du renvoi introuvable.****Erreur ! Source du renvoi introuvable.**

PETER

Peter s'est principalement occupé de la gestion de la base de données en créant le fichier .db avec la structure de la base de données et les différents jeux de test. De plus, il a géré les modèles Catégorie et Portion ainsi que les principales fonctions pour la gestion (ajout, suppression, vu) d'une catégorie et portion de texte. Le projet avait dans un premier temps commencé en mode console. Le mode console permettait une gestion des catégories et des portions de texte jusqu'à ce que Ilyes implémente une interface graphique.

Peter a également créé un dépôt gitlab afin de pouvoir travailler à distance et voir l'évolution du projet. Il a également utilisé gitlab afin d'établir une liste des tâches à faire et à vérifier, principalement pour reporter les différents bugs et amélioration qu'Ilyes devait corriger/implémenter dans l'interface graphique.

Ce rapport a été réalisé par Peter, qui regroupe les principales informations du projet, de la création à la conclusion du projet.

LES FONCTIONNALITES

L'INTERFACE – PRESENTATION

La première interface est l'élément central de l'application. En effet, tous les ajouts éditions et suppression de contenu (catégories/portions) se font depuis cette interface. Attention cependant à ne pas prendre la racine « Accueil » sans quoi le bouton de génération de document sera désactivé.

BARRE DE RECHERCHE

La barre de recherche permet d'effectuer une recherche par mots-clés, ainsi les résultats de recherche correspondent aux portions de texte dont les mots clés correspondent à la recherche de l'utilisateur.

L'ARBRE

L'arbre est la représentation graphique du contenu de la base de données. Les portions prennent une apparence de dossiers et les portions sont représentées par ce logo. Il est possible d'y ajouter autant d'éléments qu'on le souhaite.

PANNEAU LATERAL DROIT : L'ÉDITION

Le panneau latéral droit est le panneau d'édition du contenu. C'est ici que nous pouvons éditer un contenu en modifiant la portions de texte ou en ajoutant/supprimant des mots-clés.

Les mots-clés permettent de retrouver une portion de texte lors d'une recherche, il est donc important de bien les choisir. Il est possible d'en ajouter plusieurs simultanément en les séparant par une virgule « exemple,de,mots-clés,multiples » rajoute donc 4 mots-clés.

Le bouton de sauvegarde permet l'enregistrement des modifications apportées à la portion de texte, les mots-clés non cochés ne sont pas sauvegardés.

Le bouton de suppression permet la suppression des mots-clés sélectionnés.

LA GÉNÉRATION DE DOCUMENTS

La génération de documents se fait de la manière suivante : sur l'arbre de la page principale, vous devez sélectionner les éléments que vous souhaitez faire figurer dans votre document. Les catégories sélectionnées seront vos chapitres et les portions de votre contenu.

L'utilisateur arrive alors sur la fenêtre de génération de documents, celle-ci demande trois informations :

- Le nom du fichier
- Le format de sortie (Texte brut / HTML)
- Le thème appliqué

L'utilisateur peut également choisir la profondeur de sous-chapitres grâce à la liste déroulante au-dessus de l'arbre. Par défaut, l'arbre affiche 4 niveaux de sous chapitres (même si la sélection en amont en avait plus ce qui dépasse n'est pas affiché).

Il peut également réorganiser son document grâce aux flèches de navigation en dessous de l'arbre.

A tout moment, l'utilisateur a un aperçu des modifications qu'il effectue sur son fichier dans le panneau central. Cet affichage est constamment synchronisé avec les modifications de l'utilisateur :

- Si l'utilisateur choisit le format Texte, l'aperçu sera en texte brut.
- Si l'utilisateur choisit le format HTML, l'aperçu sera le rendu du fichier en HTML avec le CSS sélectionné.
- Si l'utilisateur change de CSS, l'aperçu montre le CSS sélectionné.
- Si l'utilisateur change l'ordre des catégories/portions, le changement dans l'aperçu est aussi immédiat.

De plus, l'interface de génération possède une liste déroulante dans laquelle on peut spécifier le niveau max de sous-catégorie que l'utilisateur souhaite afficher. Par défaut, cette valeur est initialisée à 4 niveaux de profondeur.

LE DESIGN

Afin d'obtenir une interface graphique agréable à l'œil et facile d'utilisation/compréhension, nous avons opté pour l'utilisation d'une charte graphique basé sur des jeux d'icônes et de couleurs.

Le choix des couleurs a donc été le bleu, le blanc et le gris, qui a également été repris pour la réalisation de ce rapport. Voici quelques exemples d'icônes pouvant être retrouvées dans l'application :



Figure 1 : Suppression



Figure 2 : Sauvegarde



Figure 4 : Mot clé



Figure 3 : Ajout

CONCLUSION

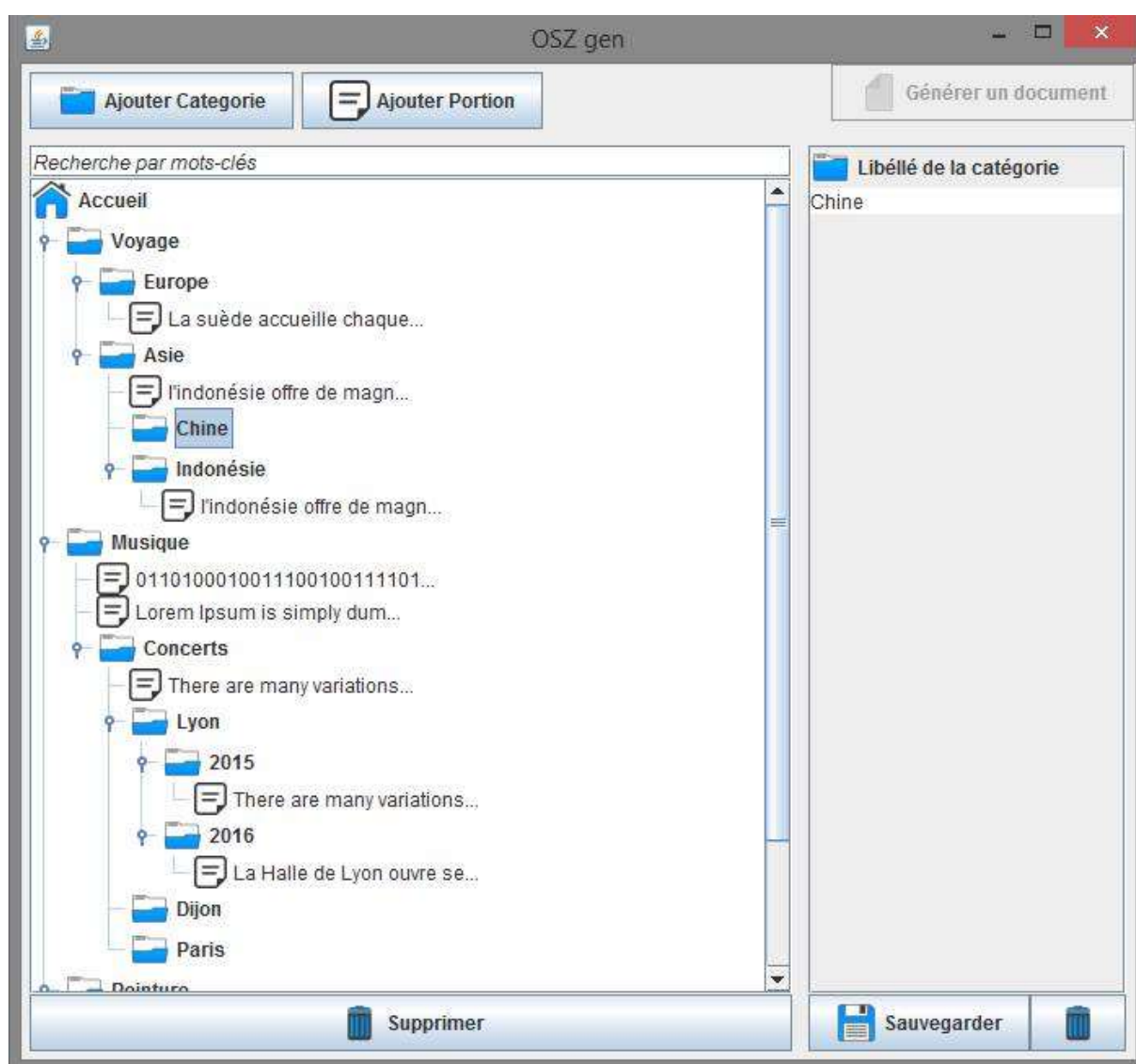
Ce projet nous a permis d'apprendre à mettre en place une organisation de travail rapidement et efficacement. Aussi, cela nous a permis d'apprendre à mettre en place une interface graphique et de conforter nos connaissances en Java. Pour ce qui est de la gestion du projet, l'utilisation continue de Gitlab nous appris comment bien gérer les différentes versions du projet et de pouvoir communiquer sans avoir de conflit dans le développement du projet.

TABLE DES ILLUSTRATIONS

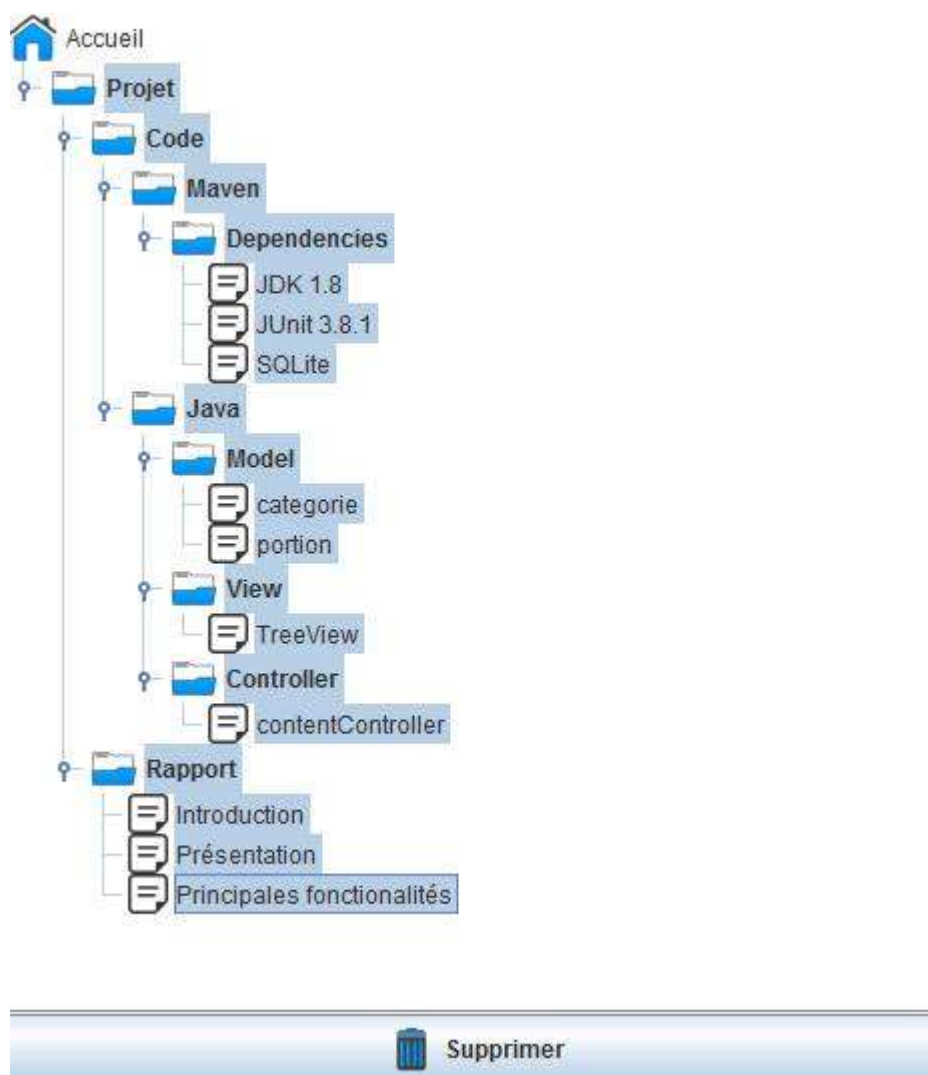
Figure 1 : Suppression	5
Figure 2 : Sauvegarde	5
Figure 4 : Ajout	5
Figure 3 : Mot clé.....	5

ANNEXES

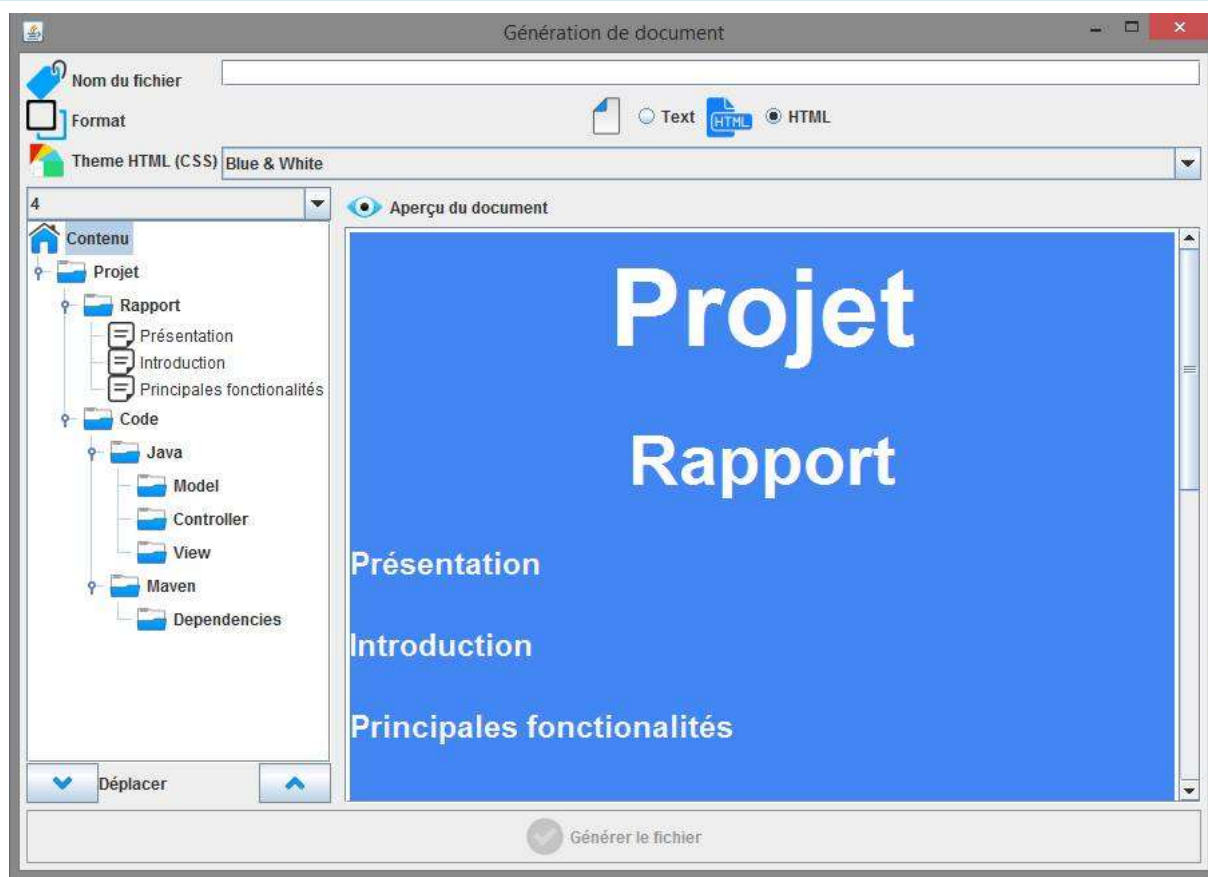
ANNEXE 1 : CAPTURE D'ECRAN DE L'INTERFACE



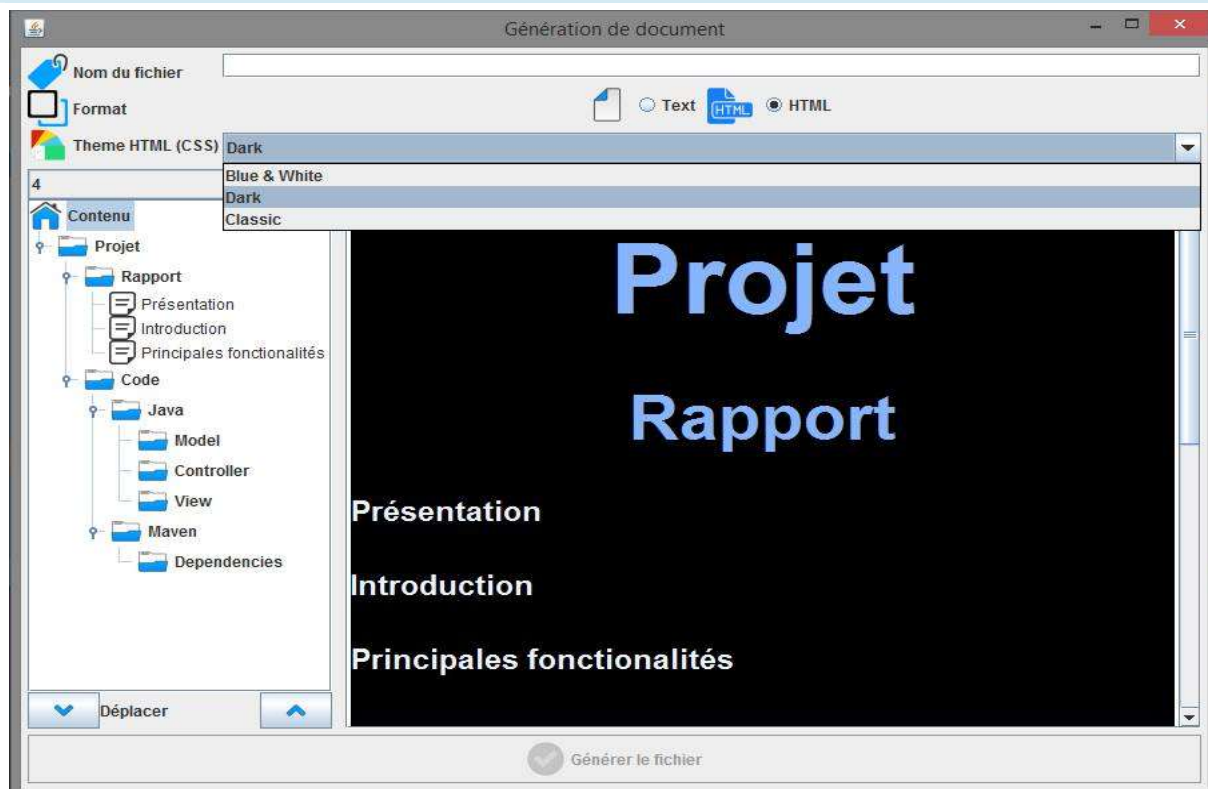
ANNEXE 2 : CAPTURE D'ECRAN DE L'ARBORESCENCE



ANNEXE 3 : CAPTURE D'ECRAN DE LA GENERATION DE DOCUMENT



ANNEXE 4 : CAPTURE D'ECRAN DES DESIGNS DE GENERATION DE DOCUMENT



ANNEXE 5 : CAPTURE D'ECRAN DE L'APERÇU DU RENDU

