

RAPPORT DE PROJET SMA

Module de Structuration de contenus : XCSM

Membres du Groupe :

- KOUAYEP Paterne
- NOUPELAH Vladimir
- NYASSA Pascal
- TANWI NKIAMBOH
- TCHUITSE Madina

Sous la supervision de : Dr BATCHAKUI

TABLE DES MATIERES

DESCRIPTION.....	2
EXIGENCES FONCTIONNELLES.....	2
DESCRIPTION DES CAS D'UTILISATION.....	3
Les acteurs	3
Les cas d'utilisation	3
Le diagramme de cas d'utilisation	6
DIAGRAMME DE CLASSES	8
DIAGRAMME DE SEQUENCE	9
La consultation d'un cours déjà importé et préalablement structuré.....	9
Le passage d'un cours Word	11
Le passage d'un cours PDF	13
La structuration.....	14
La notification	17
PARTIES IMPLEMENTEES.....	18
L'interface graphique	18
La conversion	20
La structuration.....	20
OUTILS UTILISES	21
IMPLEMENTATION DES AGENTS	22
ENVIRONNEMENT	24
INSTALLATION ET UTILISATION	24
Lancement de laravel.....	24
Création de la base de données MySQL	24
Lancement de l'application dans un navigateur	24
Importation d'un nouveau cours	26
Configurations pour les notifications.....	28
Création du plug-in XCSM pour Moodle	28
SOURCES	30

DESCRIPTION

XCSM est un module de structuration des contenus développé dans le but :

- ✎ D'organiser les documents en leur offrant une structure pédagogique de 5 niveaux (cours, partie, paragraphe, chapitre, notion)
- ✎ De réduire les risques de surcharge cognitive
- ✎ De faciliter la publication des contenus en ligne
- ✎ De garantir la réutilisation des notions lors de la composition de nouveaux cours
- ✎ De réduire considérablement les flux sur les bandes passantes lors des formations à distance

Depuis son développement, l'application XCSM est mise à jour de génération en génération. Il s'agit donc pour nous de savoir comment améliorer l'IHM de l'application XCSM afin de la rendre plus accessible et plus conviviale et la navigation plus fluide.

EXIGENCES FONCTIONNELLES

Le module de structuration offre les fonctionnalités suivantes :

- ✎ Charger un cours au format docx ;
- ✎ Le parser ou le structurer ;
- ✎ Naviguer à travers les différents niveaux d'un cours ;
- ✎ Consulter la liste de tous les cours disponibles ;
- ✎ Procéder à la notification ;
- ✎ Accéder à Moodle depuis l'application ;

Les exigences fonctionnelles attendus qui nous ont été soumises sont les suivantes :

- ✎ L'application doit pouvoir signaler à l'utilisateur qu'un document non structuré ne peut être pris en charge au cas où il introduirait un tel document ;
- ✎ Envisager la possibilité de lire d'autres formats en entrée tels que PDF ;
- ✎ Lors de la navigation dans un cours afficher le fil d'Ariane (chemin) pour éviter la surcharge cognitive ;
- ✎ Inclure la possibilité d'effectuer des suggestions et de les transmettre à un enseignant par email ;

- ☞ Recenser le nombre de fois que les étudiants s'attardent sur une notion pour juger de leurs compréhensions en ce qui concerne ladite notion ;
- ☞ Proposer des solutions et instructions pour améliorer un document mal structuré ;

DESCRIPTION DES CAS D'UTILISATION

Les acteurs

Les acteurs en présence sont :

- ☞ Les utilisateurs
- ☞ Le système

Les cas d'utilisation

- ☞ Charger un cours

Cas d'utilisation	Charger un cours
Acteur	Utilisateur
Description	Un utilisateur peut charger ou introduire un nouveau cours dans le module

- ☞ Parser le cours en chapitres, paragraphes et notions

Cas d'utilisation	Parser les cours
Acteur	Système
Description	Quand un cours est importé, le système procède automatiquement à sa structuration

- ☞ Naviguer à travers les différents niveaux d'un cours

Cas d'utilisation	Naviguer à travers les différents niveaux d'un cours
Acteur	Utilisateur
Description	L'utilisateur, une fois le cours structuré, doit pouvoir atteindre sans difficulté la hiérarchie dudit cours à partir d'un menu

- ☞ Consulter la liste de tous les cours disponibles

Cas d'utilisation	Consulter la liste de tous les cours disponibles
Acteur	Utilisateur

Description	Une fois le cours introduit, il peut être lu ou consulté par les autres utilisateurs
--------------------	--

☞ Notifier par email et/ou sms

Cas d'utilisation	Notifier par email ou sms
Acteur	Système
Description	Le système doit permettre de pouvoir notifier d'autres utilisateurs de la présence d'un cours structuré

☞ Accéder à la plateforme Moodle

Cas d'utilisation	Accéder à la plateforme Moodle
Acteur	Système
Description	Le système doit permettre de pouvoir accéder à la plateforme Moodle depuis l'application et pouvoir revenir

☞ Parser un cours au format PDF

Cas d'utilisation	Parser un cours PDF
Acteur	Système
Description	Le système doit pouvoir structurer un cours au format PDF

☞ Fournir une rétroaction pour des document mal structuré ;

Cas d'utilisation	Fournir une rétroaction pour des documents mal structuré
Acteur	Système
Description	Le système doit pouvoir signaler à l'utilisateur qu'un document non structuré ne peut être pris en charge au cas où il introduirait un tel document ;

☞ Afficher le fil d'Ariane ;

Cas d'utilisation	Afficher le fil d'Ariane
Acteur	Système
Description	Lors de la navigation dans un cours ; le système doit afficher le fil d'Ariane (chemin) pour éviter la surcharge

	cognitive ;
--	-------------

☞ Effectuer des suggestions ;

Cas d'utilisation	Effectuer de suggestions
Acteur	Utilisateur
Description	Un utilisateur doit avoir la possibilité d'effectuer des suggestions et de les transmettre à un autre utilisateur par email ;

☞ Transmettre des suggestions par email a un utilisateur ;

Cas d'utilisation	Transmettre des suggestions par email a un utilisateur
Acteur	Système
Description	Un utilisateur doit avoir la possibilité d'effectuer des suggestions et de les transmettre à un autre utilisateur par email ;

☞ Recenser le nombre de consultation d'une notion;

Cas d'utilisation	Recenser le nombre de consultation d'une notion
Acteur	Système
Description	Le système doit recenser le nombre de fois que les étudiants s'attardent sur une notion ;

☞ Juger la compréhension d'une notion;

Cas d'utilisation	Recenser le nombre de consultation d'une notion
Acteur	Système
Description	Après avoir recenser le nombre de fois que les étudiants s'attardent sur une notion, le système doit pouvoir juger de leurs compréhensions en ce qui concerne ladite notion .

Le diagramme de cas d'utilisation

XCSM : Module de Structuration des Contenus

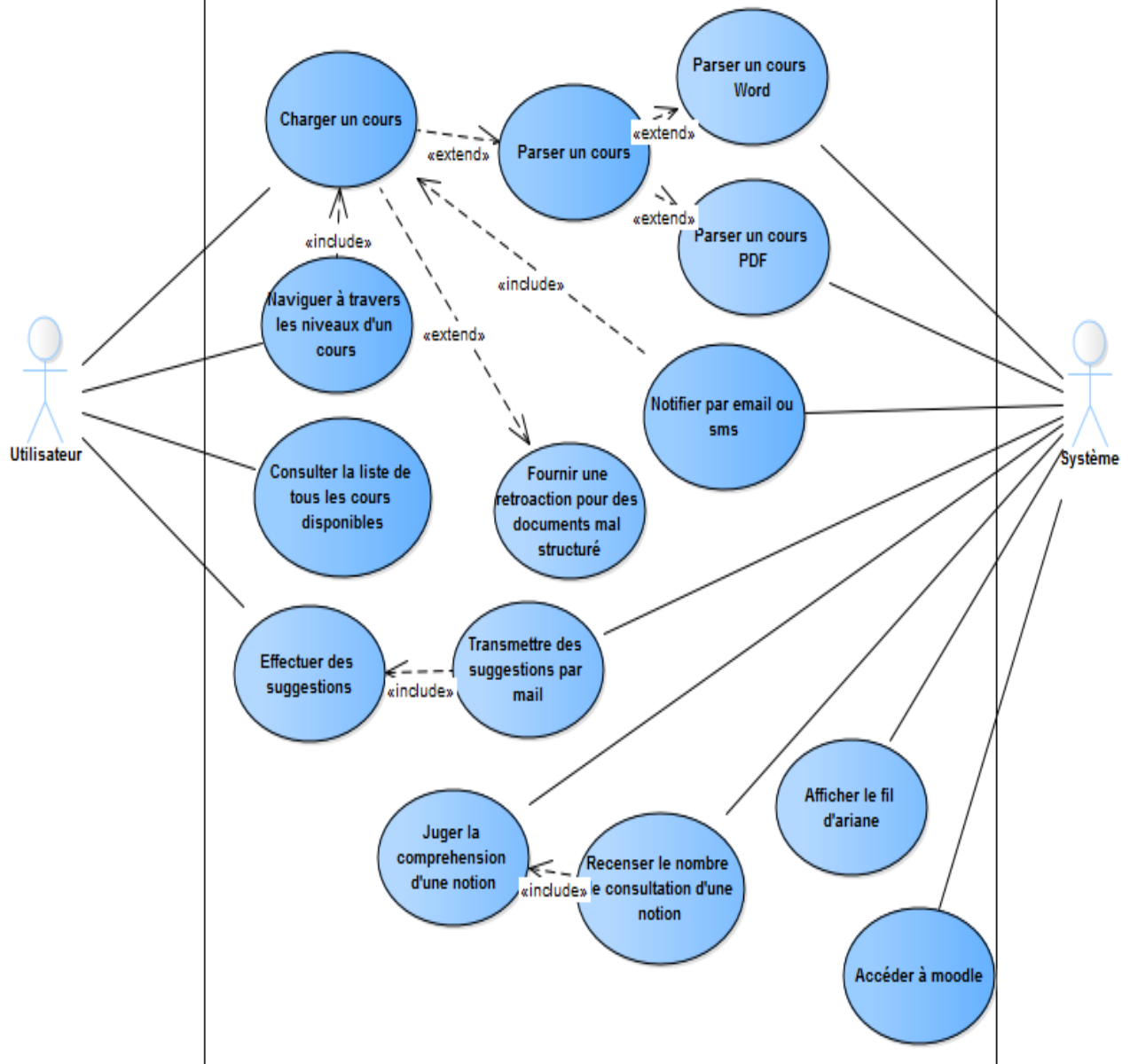


DIAGRAMME DE CLASSES

Les différentes classes sont groupées en fonction des modules :

- ☞ **La vue** : l'accueil, la page de cours (le cours actuellement utilisé), la page qui liste tous les cours disponibles, la page de notifications
- ☞ **Le parseur** : parser (classe principale), Docx4J et WordProcessingMLPackage qui sont des bibliothèques de traitement de documents Word, FO Setting, PDF_to_fo_Parser qui est la bibliothèque de traitement de documents PDF.
- ☞ **La structuration** : les cours, les parties, les chapitres, les paragraphes, les notions
- ☞ **La notification** : les étudiants, les notifications, les suggestions.

Le diagramme de classe est donc le suivant :

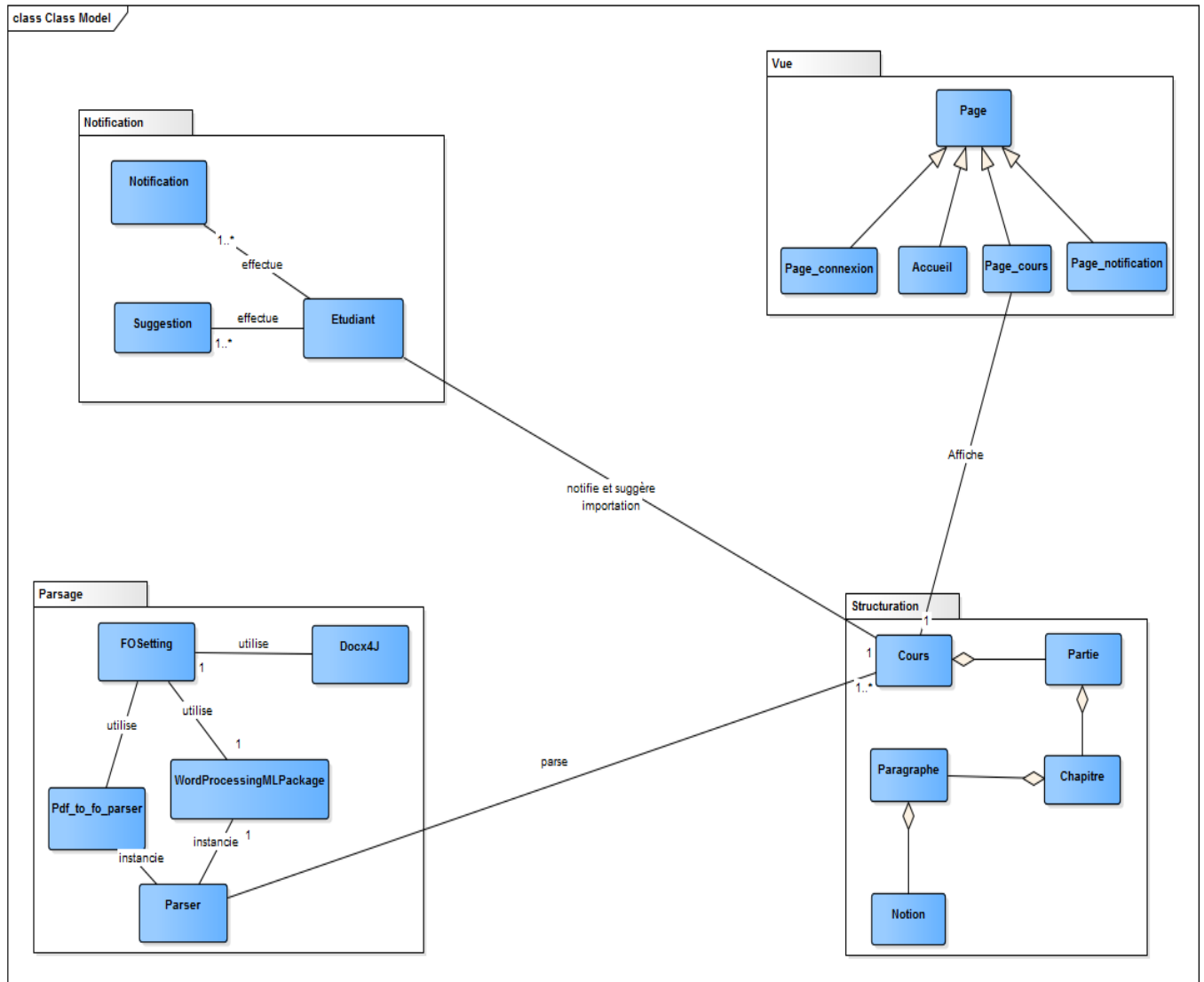
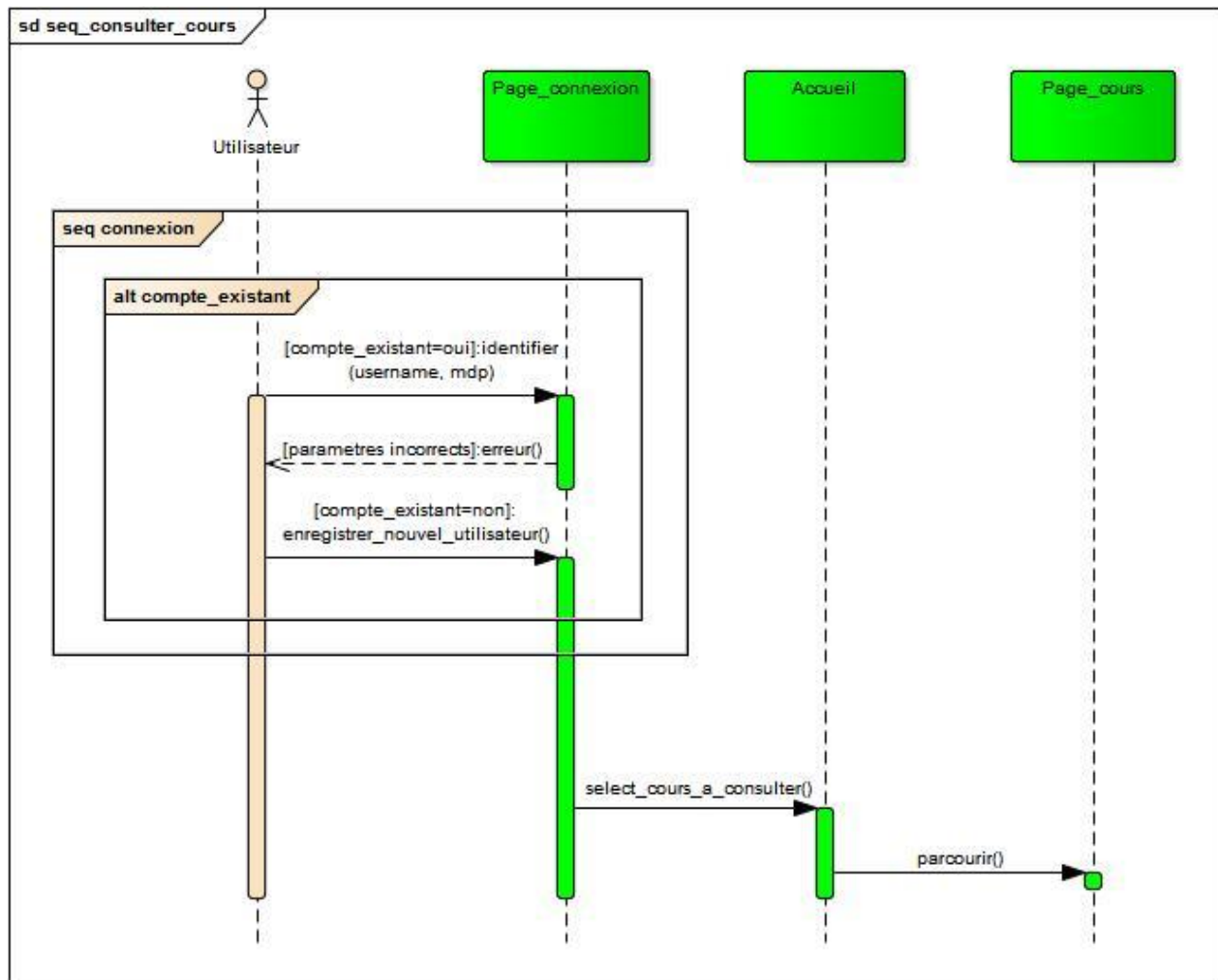


DIAGRAMME DE SEQUENCE

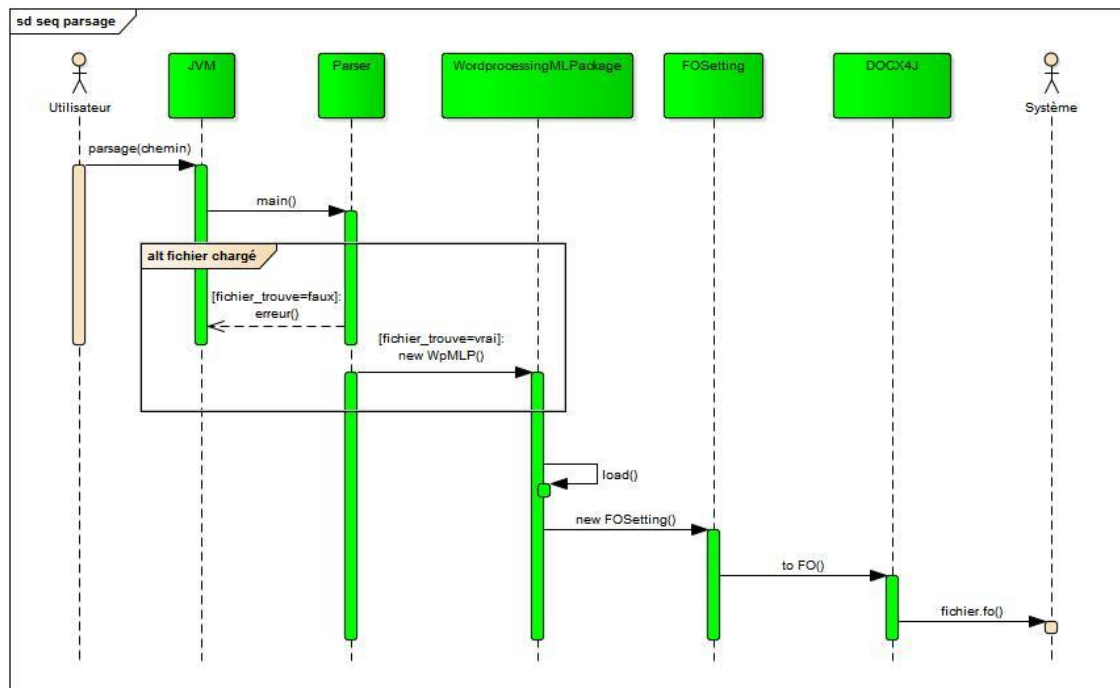
Les principales séquences dans l'utilisation du module sont :

La consultation d'un cours déjà importé et préalablement structuré



L'importation d'un nouveau d'un nouveau cours, ce qui passe par les processus de parsing et de structuration.

Le passage d'un cours Word



Le processus de conversion d'un fichier du format docx vers le format xlsfo se fait suivant le processus suivant :

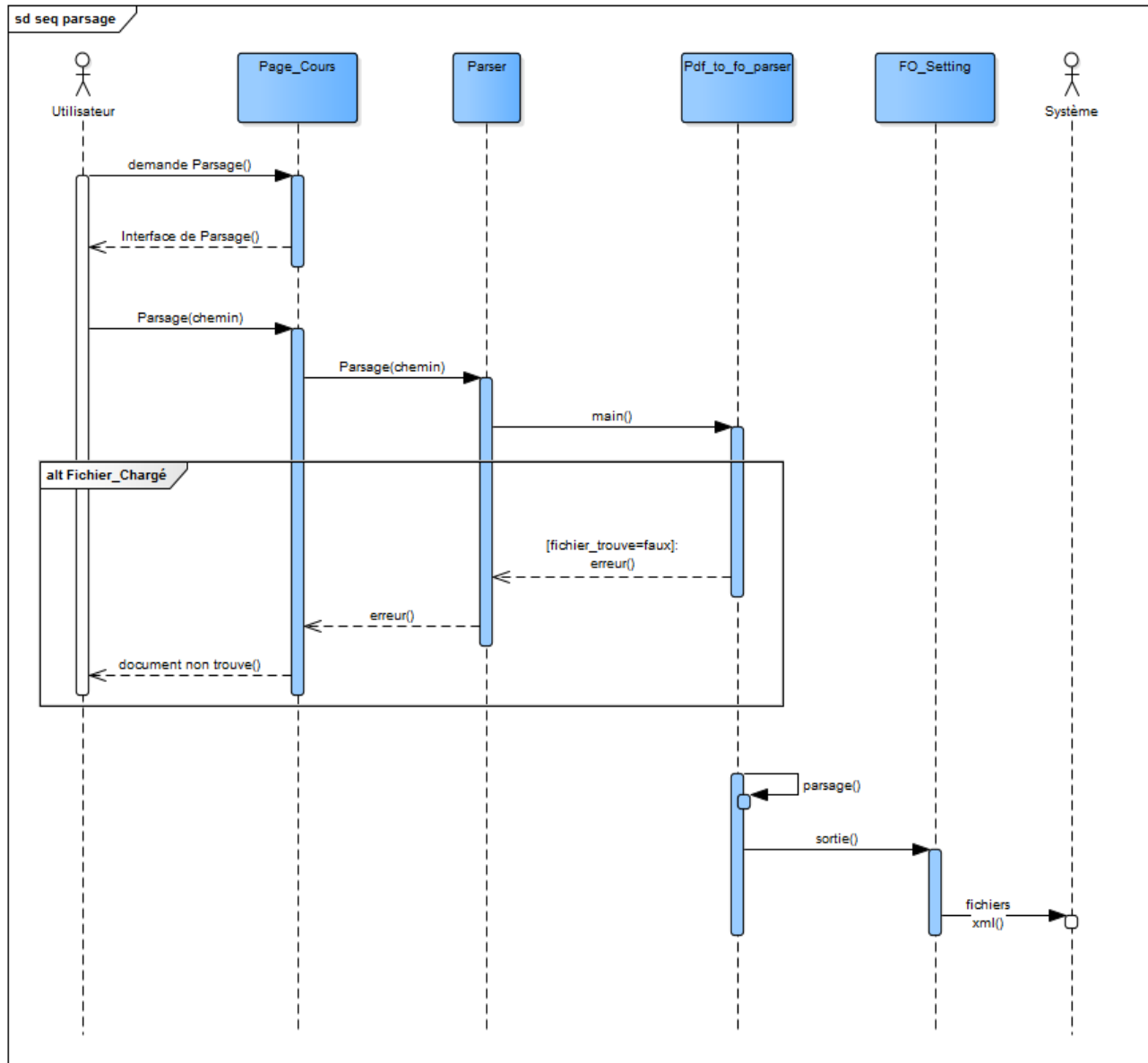
- ☞ L'utilisateur appelle le programme de passage (fichier jar) en lui passant en paramètres le chemin vers le fichier source à convertir (fichier au format docx) :

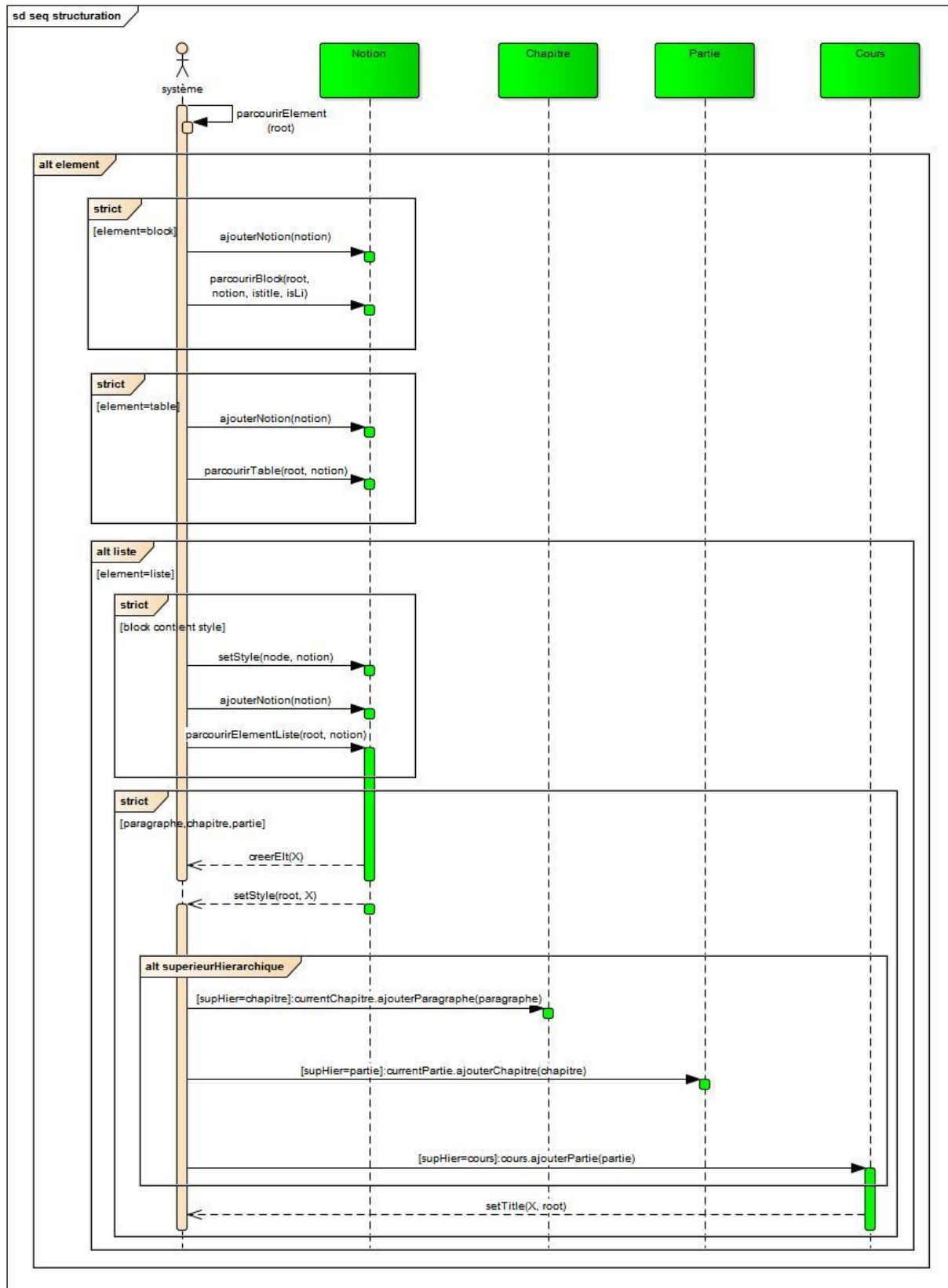
Java -jar Docx_To_Fo_Parser.jar <fichier source au format docx>

- ☞ La machine virtuelle Java appelle la méthode main de la classe **Parser**
- ☞ La méthode main appelée lit et vérifie le paramètre de la méthode et renvoie un message d'erreur si elle n'arrive pas à charger le fichier.
- ☞ Si le fichier est bien présent et accessible en lecture, la méthode main crée et instancie un objet de la classe **WordprocessingMLPackage**. Le fichier docx (son flux qui a été lu) est chargé dans l'objet de la classe **WordprocessingMLPackage** grâce à la méthode **load ()** de cette classe.
- ☞ Le main crée ensuite un objet de la classe FOSettings et l'instancie avec des valeurs par défaut accessibles par une méthode statique de la classe **DOCX4J (createFOSettings ())** ;

- ☞ D'autres paramètres sont ajoutés à l'objet de la classe FOSettings créé tels que le nom du fichier de sortie, son extension et bien d'autres.
- ☞ La méthode main crée ensuite un flux de sortie correspondant au fichier au format xslfo qui sera produit en sortie par le parseur.
- ☞ Le main appelle ensuite la méthode statique **toFO ()** de la classe **DOCX4J** en lui passant en paramètres le flux de sortie et l'objet de la classe FOSettings contenant les paramètres de conversion.
- ☞ Cette méthode réalise la conversion en affichant les différentes étapes et les éventuelles erreurs en console. A la fin du processus un fichier contenant le même nom que le fichier source augmentée d'une « **.fo** » à la fin est produit dans le même répertoire que le fichier source.
- ☞ Ce fichier est formaté en xslfo

Le parsing d'un cours PDF



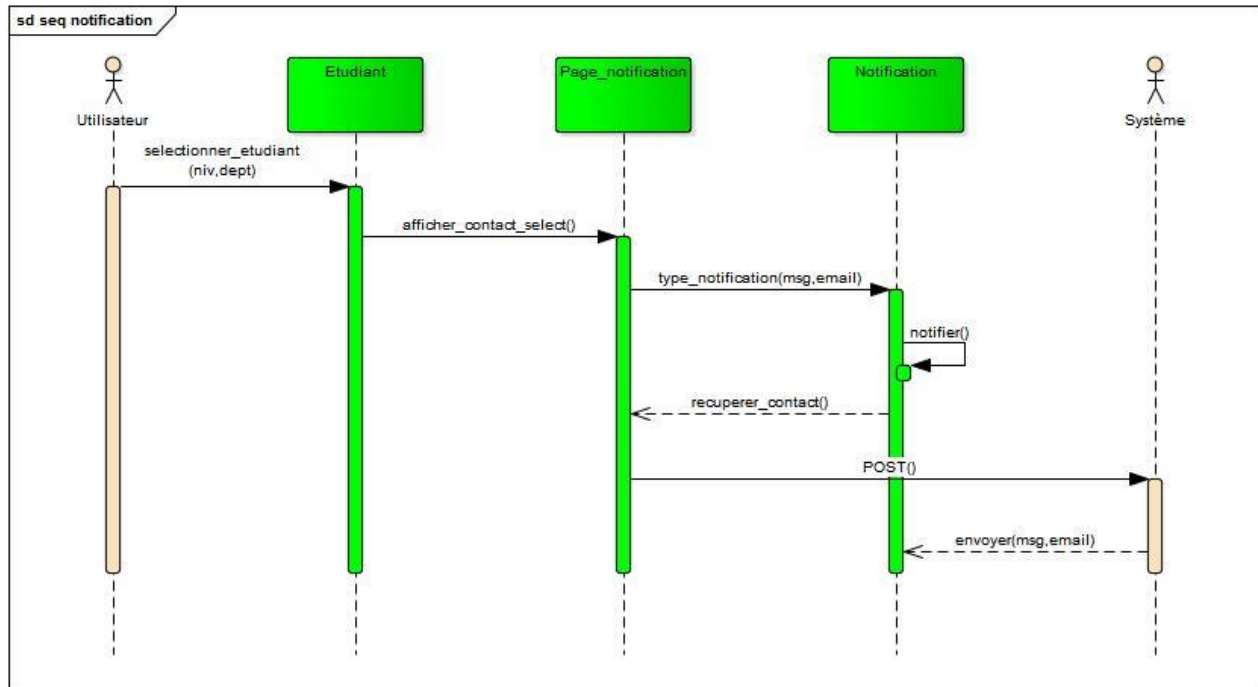


- ☞ On parcourt tous les nœuds qui ne sont pas des métadonnées: `parcoursElement(Node root)`
- ☞ On écrit ensuite les documents de descriptions

Description de la fonction `parcoursElement(Node root)`

- S'il s'agit d'un `fo:block` je créer une Notion
 - On ajoute la notion: `ajouteNotion(Notion notion)`
 - On parcourt l'élément: `parcoursBlock(Node root, Notion notion, boolean isTitle, boolean isLi)`
- S'il s'agit d'un `fo:table` je créer une Notion
 - On ajoute la notion: `ajouteNotion(Notion notion)`
 - On parcourt la table: `parcoursTable(Node root, Notion notion)`
- S'il s'agit d'un élément de liste (un `fo:list-block`)
 - Si son `fo:list-item-label` a un `fo:block` de `font-family symbol`, je créer une notion
 - On copie le style: `setStyle(Node node, Element notion)`
 - On ajoute la notion: `ajouterNotion(Notion notion)`
 - On récupère le contenu de l'élément de liste: `parcoursElementListe(Node root, Notion notion)`
 - Si non, il s'agit soit d'un paragraphe, un chapitre ou une partie
notons "X" qui sera un paragraphe, un chapitre ou une partie
 - On crée un élément X,
 - on met son style : `setStyle(Node root, X x)`
 - On l'ajoute à son supérieur hiérarchique (chapitre, partie ou cours):
`cours.ajouterPartie(Partie partie),`
`currentChapitre.ajouterParagraphe(Paragraphe paragraphe)` ou
`currentPartie.ajouterChapitre(Chapitre chapitre).`
 - On met son titre: `setTitle(X x, Node root)`

La notification



PARTIES IMPLEMENTEES

L'interface graphique

L'ihm de l'application a été totalement redéfinie, et repensée. Un nouveau logo a également été pensé et créé, et inséré à l'interface. De façon générale, l'application a 4 pages principales à savoir :

- ☞ La page de connexion : c'est elle qui permet de recueillir les informations d'identification afin d'accorder ou non l'accès à l'application

Connexion au module XCSM

Connectez-vous pour accéder aux cours

jojodjizanne@gmail.com

.....

☐ Se souvenir de moi

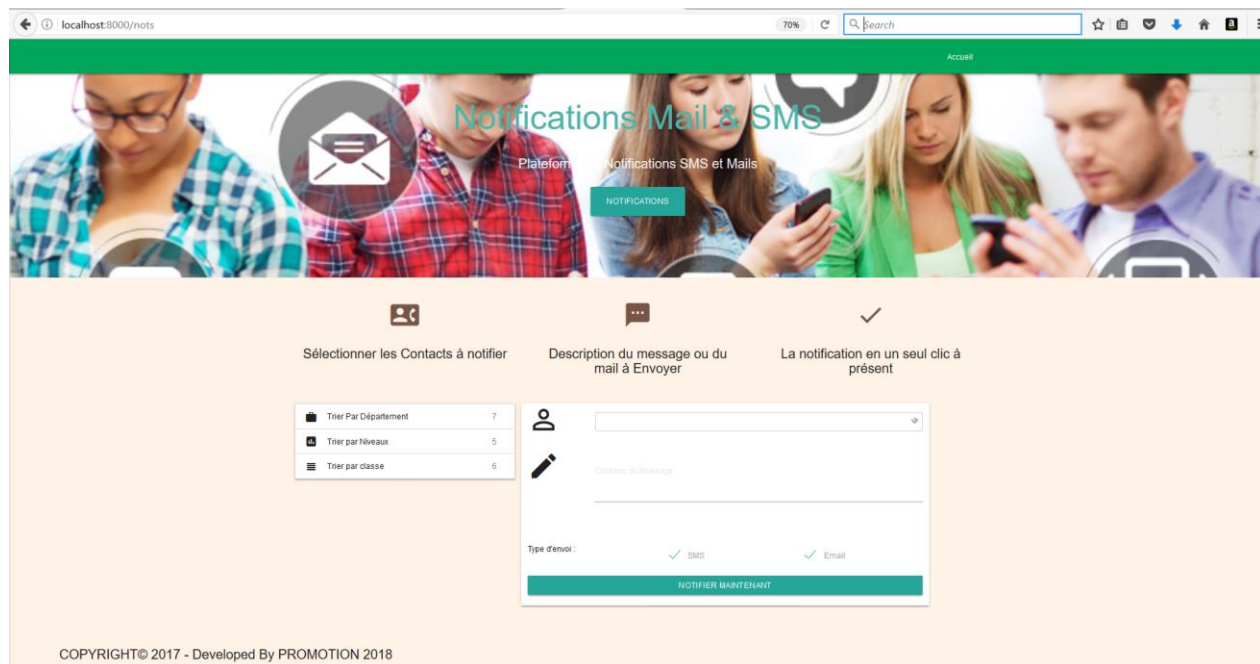
Connexion

[Enregistrer un nouvel utilisateur](#)

- ☞ L'accueil : après une identification réussie on arrive sur la page d'accueil qui présente les différents menus et la hiérarchie

- ☞ La page de cours : c'est la page qui permet de lire et de parcourir un document importé et structuré

- ☞ La page de notification : c'est à partir de cette page qu'on effectue les notifications. Il s'agira de choisir les personnes à notifier, de saisir le message à envoyer et d'effectuer l'envoi soit par message soit par mail.



La conversion

Il s'agit du processus de « passage ». La conversion ici consiste à partir d'un document au format PDF pour un autre au format fo, contenant exactement les même informations que le document de départ. Ce nouveau format obtenu est celui exploitable par le module de structuration.




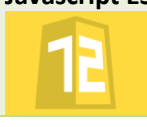



La structuration

Ce module permet de lire le fichier xsfio produit par le module de structuration afin d'en ressortir la hiérarchie sous la forme :

- 🔗 Partie
- 🔗 Chapitre
- 🔗 Paragraphe
- 🔗 Notion

Ainsi le document pourra donc être parcouru suivant ces grandes parties, ce qui facilite la lecture et l'apprentissage.

OUTILS UTILISES

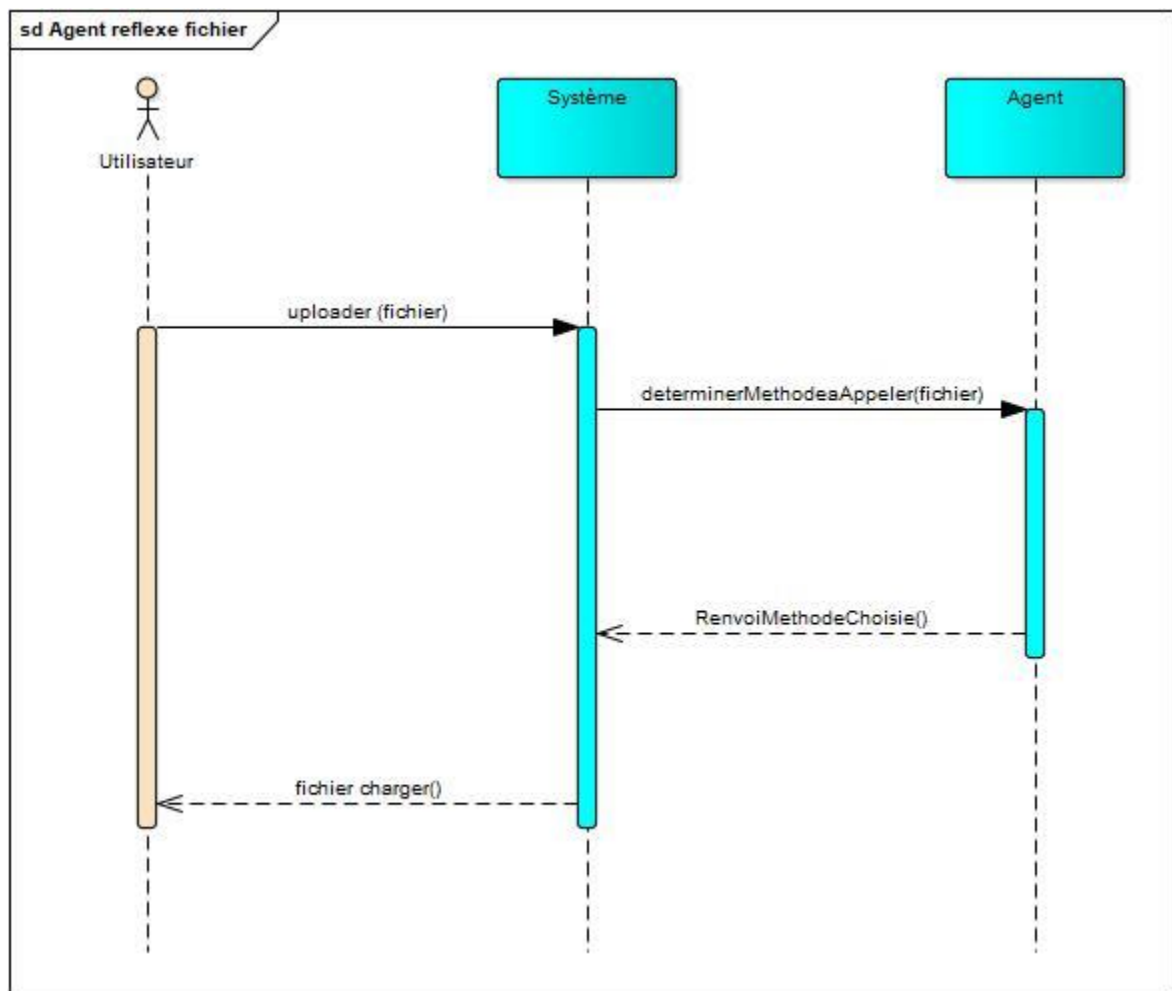
Outil/Technologie	Version	Description	Rôle dans l'implémentation
DOCX4J	3.3.3	Bibliothèque java permettant la manipulation des fichiers au format docx	Utilisé lors du passage pour convertir du format docx au format xslfo
JAVA SE 	8.74	Langage de programmation permettant de développer aussi bien des applications de bureau que sur des serveurs	Langage utilisé pour programmer les processus de conversion et de structuration
Laravel 	5.4	Framework PHP facilitant la création d'applications en PHP	Utilisé pour réaliser les interfaces de présentation, les pages de l'application
PHP 	7	langage de programmation côté serveur	C'est le support du Framework laravel
Javascript ES 	6	langage événementiel permettant la dynamisation des pages côté client	Utilisé pour définir les animations sur les pages
Bootstrap 	3	Framework css facilitant la présentation des pages	Utilisé pour réaliser l'ihm générale de l'application (agencements des couleurs, des blocks de contenus, etc.)
Materialize css 		Framework css facilitant la présentation des pages	Utilisé pour réaliser l'ihm du module de structuration
Moodle 	3.1	plateforme utilisée pour la gestion des contenus à caractère éducatif	Utilisé comme support d'intégration et hébergeur final de l'application
PdfBox	3.1	Bibliothèque java permettant la manipulation de documents au format PDF	Utilisé lors du passage pour convertir du format PDF au format xslfo

IMPLEMENTATION DES AGENTS

Pour faciliter l'apprentissage, nous allons implémenter quelques agents dans notre application. Pour se faire, nous allons implémenter des agents reflexes simples et des agents basés sur les buts.

1- Les agents reflexes simple

Un agent reflexe simple comme son nom l'indique exécute une action en fonction de l'action courante. Ainsi lorsque nous importons un fichier que l'application doit parser, on peut mettre sur pied un agent qui détermine la nature la nature du fichier (son extension). Ainsi, si à l'avenir l'application doit prendre en compte dix (10) catégories de fichier, il ne serait pas nécessaire .d'afficher dix boutons pour chacune des catégories. On entre juste le fichier et l'agent détermine sa nature (son extension) et effectue l'action appropriée. Son comportement peut être décrit par le diagramme de séquence suivant :

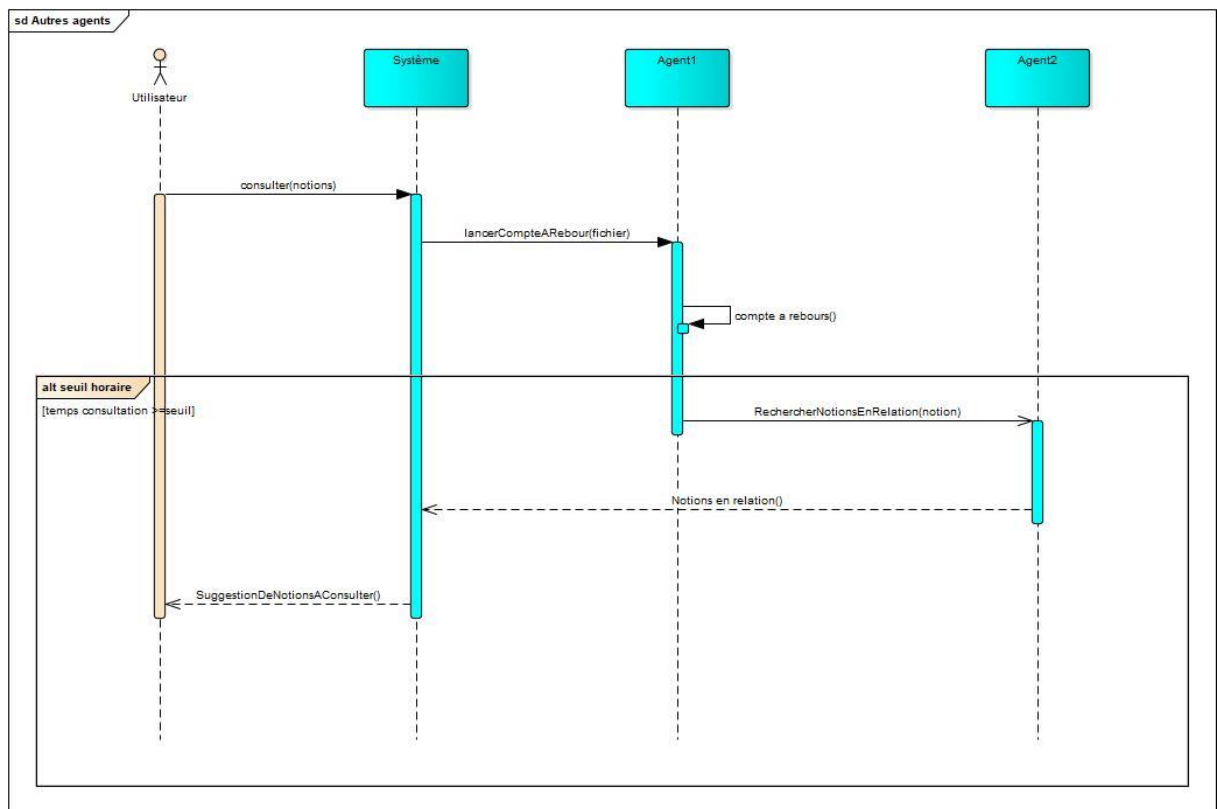


Comme autre agent reflex simple, nous comptons mettre sur pied un agent permettant de faciliter les recherches dans l'application. Ainsi, connaissant juste une notion dans un cours, l'agent nous donne tous les cours, tous les enseignants... en rapport avec cette notion

Nous pouvons également implémenter un agent qui réfléchit qui enverra automatiquement les notifications par mail à tous les étudiants concernés par le cours lorsqu'une notification est faite au lieu de le faire manuellement comme c'est le cas maintenant.

2- Les agents basés sur les buts

Le but ici peut être de proposer aux apprenants des cours en rapport avec ceux sur lesquels ils mettent un certain nombre de temps. Pour se faire nous pouvons mettre sur pied un SMA constitué d'un agent reflex et d'un agent basé sur les buts. Lorsqu'un étudiant dépasse le temps limite sur une notion, l'agent reflex déclenche l'agent orienté sur le but qui recherchera tous les autres cours qui contiennent cette notion. Ce second agent se doit d'être rationnel. Le comportement de ce SMA peut être décrit par le diagramme de séquence suivant :



ENVIRONNEMENT

Pour pouvoir déployer l'application, l'environnement doit être composé de :

- 🔗 MySQL
- 🔗 Java 1.7 ou plus
- 🔗 Un serveur local (Wamp ou Xamp)
- 🔗 Un IDE PHP (webstorm, phpstorm ou intellijidea)
- 🔗 Laravel 5.4.23 ou plus
- 🔗 Moodle

INSTALLATION ET UTILISATION

Il faut d'abord dézipper le dossier projet dans le www/html de votre serveur local .

Lancement de laravel

En ligne de commande windows ou phpstorm, taper la commande **PHP artisan serve** pour pouvoir démarrer les services laravel en localhost sur le port 8000.

Création de la base de données MySQL

L'application pour fonctionner, a besoin d'une de données nommée **multiauth**, il est donc question de la créer et de migrer ensuite vers celle-ci.

Une fois la bd créée, se rendre dans l'ide et ouvrir le fichier « .env » pour y modifier les paramètres d'accès à MySQL (nom d'utilisateur et mot de passe) et ensuite migrer vers la bd nouvellement créée en faisant : **PHP artisan migrate** (dans le terminal de l'ide).


Lancement de l'application dans un navigateur


Le serveur étant lancé allons dans le navigateur taper: **localhost:8000**, nous avons laissé la route par défaut de laravel.

Ceci nous conduira à la page de connexion suivante :

Connection au module XCSM

Connectez-vous pour accéder aux cours





☐ Se souvenir de moi


Connexion


[Enregistrer un nouvel utilisateur](#)


Il faudra donc, le cas échéant, enregistrer un nouvel utilisateur afin d'accéder aux services. Pour cela il faudra renseigner des informations telles que présentées sur la page de création de compte suivante (nom, adresse mail, numéro de téléphone, département, niveau, classe et mot de passe) :


XCSM Module


Nouvel utilisateur




















Register

[J'ai déjà un compte](#)

Pour avoir les droits d'administrateur (qui vous permettront de pouvoir importer un nouveau dans le module et pouvoir notifier les autres de sa présence), il faudra aller dans la base de données modifier la valeur du paramètre *is_admin* et le faire passer de 0 à 1 : *update users set is_admin=1 where id=1* ;

Une fois connecté, la page d'accueil est la suivante :

XCSM Module Accueil Aller à Moodle Notification Se Deconnecter

recherche

LISTE DES COURS DISPONIBLES

- D
- SMA
- Analyse-de-données
- coursnad
- asa
- ahhoui
- Test
- Test4

Bienvenu sur XCSM module(Extended Content Structured Module)

Le module de structuration des contenus (eXtended Content Structured Module) permet de mettre en ligne des contenus structurés sur une plateforme de formation xMoodle3.0.

Ce module:

- ✱ Organise les contenus en leur offrant une structure pédagogique de 5 niveaux,
- ✱ Réduit les risques de surcharge cognitive en proposant aux apprenants des contenus sur mesure,
- ✱ Facilite la publication des contenus en ligne,
- ✱ Garanti la réutilisation des notions lors de la composition de nouveaux cours,
- ✱ Réduit considérablement les flux sur les bandes passantes lors des formations à distance.

Importer un cours

⚠ Les documents doivent être sous le format docx pour être pris en compte!

COPYRIGHT© 2017 - Developed By PROMOTION 2018 CONTACTEZ-NOUS

Importation d'un nouveau cours

Comme nous l'avons sur l'image précédente présentant la page d'accueil, celle-ci contient un bouton « *importer un cours* » qui nous permettra de structurer et d'introduire un nouveau cours dans le module.

Importation d'un nouveau cours

Nom du cours:

cours_de_sma

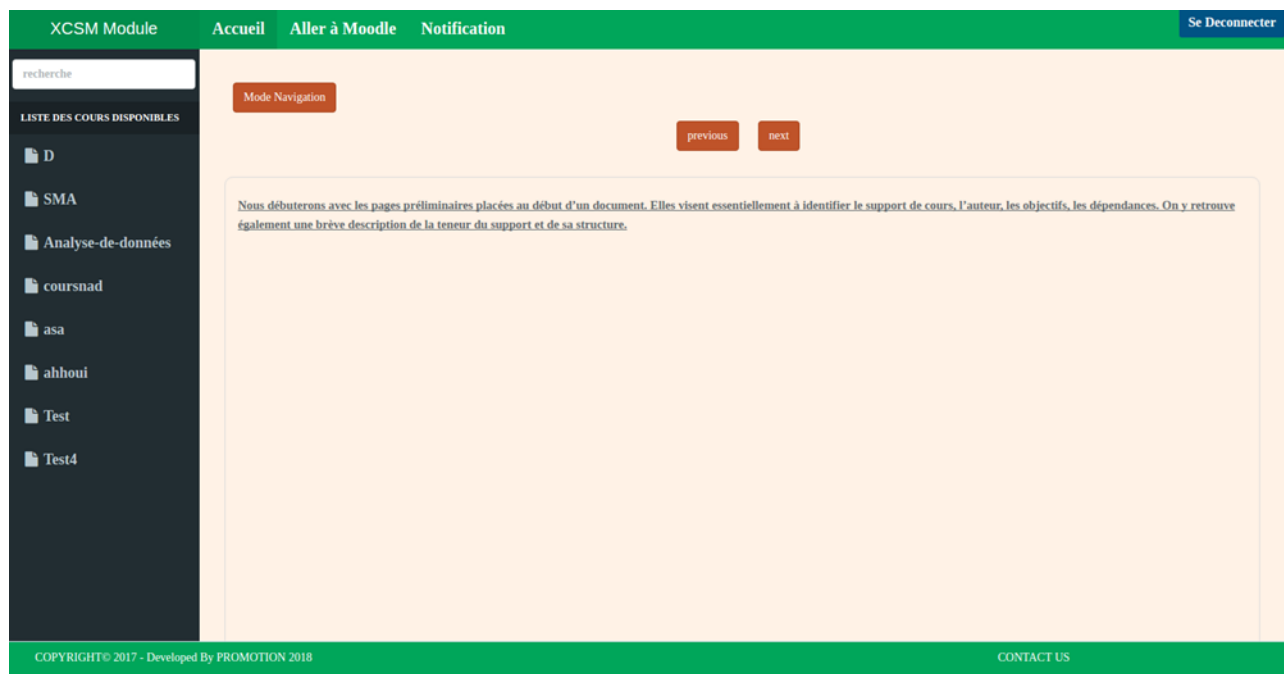
Fichier du cours

Choisissez un fichier Aucun fichier choisi

Enregistrer

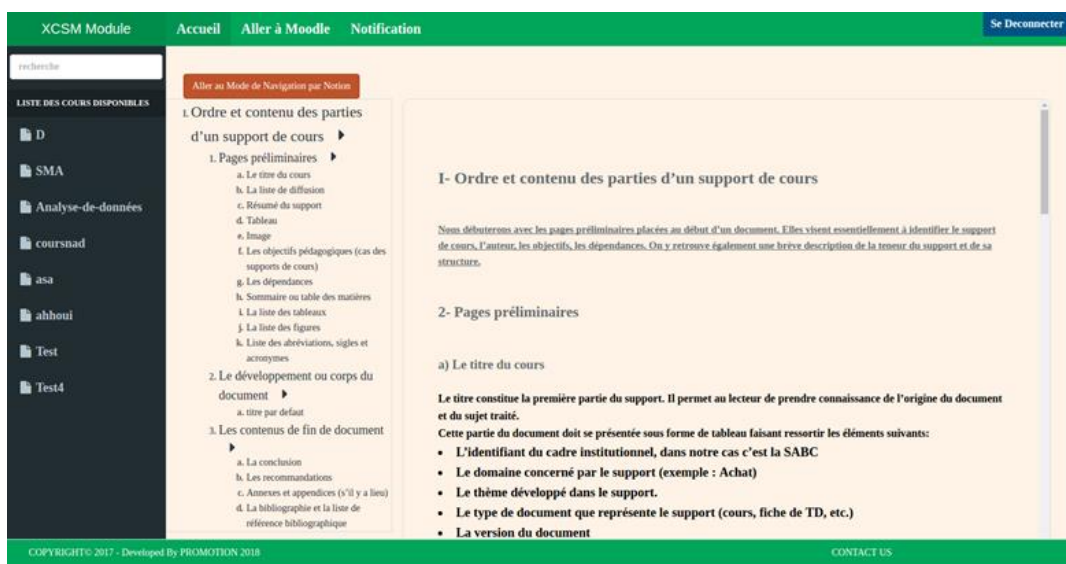
Close

Après avoir cliqué sur ce bouton il faudra donc renseigner le fichier à importer et lui donner un nom. Une fois le cours uploadé, il s'ajoute à la liste de cours dans le volet gauche de la page :



Il y a 2 possibilités de lire le cours : soit en mode navigation (en cliquant sur le bouton **mode navigation**), c'est-à-dire en parcourant la hiérarchie du document, soit en consultant uniquement les notions abordées qui sont présentées sur la page précédente en dessous des bouton **previous** et **next**. Ces boutons permettent de parcourir toutes les notions avec la possibilité à chaque fois d'avancer et de revenir en arrière.

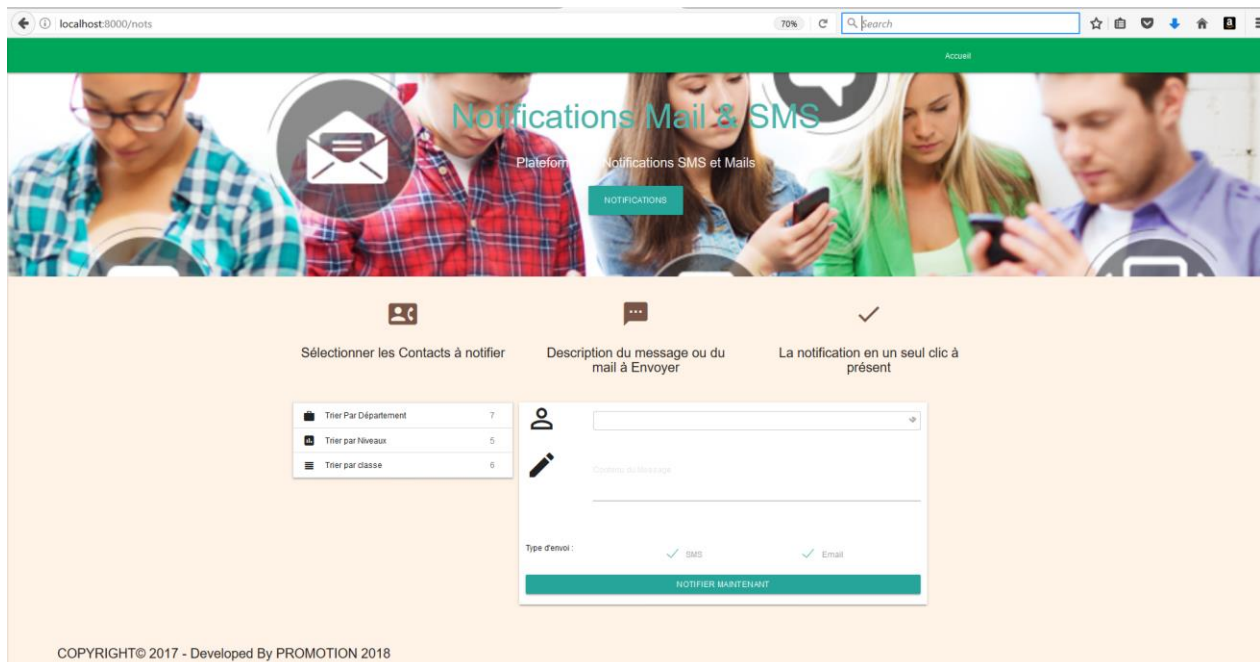
Le mode navigation se présente ainsi :



Configurations pour les notifications

Pour pouvoir effectuer une notification, il faudra au préalable connecter l'ordinateur et le serveur de messagerie sur le même réseau, en utilisant la commande : ***php artisan serve --host ip_address_machine_hôte*** (en se plaçant à la racine du dossier projet).

Ensuite il faudra renseigner, dans le serveur de messagerie, l'adresse ip de la machine hôte sur laquelle l'application est en train d'être déployée (celle qui a été renseignée dans la command précédente). Après quoi, il est donc possible d'effectuer une notification en rédigeant le message à envoyer et en choisissant le type de notification (par message ou mail) puis cliquer sur ***notifier maintenant***.



Création du plug-in XCSM pour Moodle

Nous allons pour cela suivre le lien <https://docs.moodle.org/dev/Blocks>.

Une fois le processus de création du plug-in terminé, nous aurons sur l'interface d'accueil du module XCSM un onglet « ***Aller à moodle*** » **Aller à Moodle** et qui nous redirigera vers Moodle et de même de moodle on peut revenir à XCSM en cliquant sur le lien y relatif.



Admin User

Dashboard

NAVIGATION

Dashboard

- Site home
- Site pages
- My courses
 - m
 - RO
 - sma

ADMINISTRATION

Site administration

NOTIFICATION MAIL

The content of our notificationmail block!
[[notification]]

XCSM

XCSM

Module de structuration de contenu

MODULE XCCM

[Editer le cours](#)
[XCCM](#)

ADD A BLOCK

LEARNING PLANS


COURSE OVERVIEW

Number of courses to display:

Always show all

mm

recherche operationnelle

 You have assignments that need attention

systeme multi-agents

PRIVATE FILES

No files available

[Manage private files...](#)

ONLINE USERS

 Admin User (last 5 minutes)

LATEST BADGES

You have no badges to display

CALENDAR

June 2017						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

UPCOMING EVENTS

 devoir1
recherche operationnelle
Tuesday, 27 June, 12:00 AM

 devoir2
recherche operationnelle
Tuesday, 27 June, 12:00 AM

[Go to calendar...](#)
[New event...](#)

SOURCES

Les sources principales du développement sont essentiellement les documentations présentes sur les sites des différents outils utilisés :

<https://www.oracle.com/>

<https://laravel.com/>

<https://www.docx4java.org/>

<https://moodle.org/>

<https://vuejs.org/>

<materializecss.com/>

En outre nous nous sommes inspirés de la version précédente du module XCSM pour réaliser les améliorations.