|  |
| --- |
| INF4173 – Projet Synthèse |
| Rapport de progrès |
| Développement d’un site interactif « French as Second Language » |



|  |
| --- |
| François Charette Nguyen 28/02/2010 |

Table des matières

[1. Introduction 1](#_Toc255150792)

[2. Progrès dans la recherche d’outils et d’information 1](#_Toc255150793)

[Technologies de « base » 1](#_Toc255150794)

[Base de données vs fichier XML 1](#_Toc255150795)

[Recherche d’outils existants 2](#_Toc255150796)

[3. Progrès dans le développement d’une solution 3](#_Toc255150797)

[Format des questionnaires en XML 3](#_Toc255150798)

[Interface étudiant 4](#_Toc255150799)

[Interface professeur 5](#_Toc255150800)

[4. Travail restant 6](#_Toc255150801)

# Introduction

Tout d’abord, voici un rappel de la description du projet :

Le projet consiste à élaborer un prototype d’un module de questionnaires pour un site d’apprentissage du français en tant que langue seconde. Le site sera sous la forme d’une plate-forme sociale et interactive sous le nom de « French as Second Language » (FSL). Le but du site est de permettre à des membres anglophones d’apprendre le français en les mettant en contact avec du contenu authentique provenant de la culture francophone. Le projet sera coordonné et supervisé par M. Michal Iglewski avec la collaboration de M. Alain Charbonneau.

Le document suivant servira donc de documenter le progrès qui a été fait jusqu'à maintenant sur le projet. Il sera premièrement question de décrire les technologies et outils retenus pour le développement du module pour les questionnaires de FSL. Ensuite, les parties commencées et/ou complétés du projet seront décrites. Finalement, un aperçu du travail restant sera présenté pour conclure ce rapport de progrès.

# Progrès dans la recherche d’outils et d’information

Au début du projet, il a fallu réunir de l’information sur diverses technologies et évaluer les avantages de chacune. Pour ce faire, j’ai puisé dans mes connaissances acquises lors du cours de technologies internet et programmation web avancée ainsi que dans un site social ayant une catégorie programmation ([www.reddit.com/r/programming](http://www.reddit.com/r/programming)). Ce dernier m’a amené à plusieurs blogues et sites de programmation.

## Technologies de « base »

J’ai dès le début choisi d’utiliser les technologies XHTML, CSS, PHP, Javascript et Flash simplement car je le connaissais déjà à divers degré et qu’elles sont des standards répandus dans le développement web pour du contenu généré dynamiquement (PHP/Javascript) et vidéo (Flash).

## Base de données vs fichier XML

J’ai hésité pour le choix de la structure de données que j’utiliserais. Plus précisément, j’hésitais entre l’utilisation d’une base de données relationnelle (MySQL) et l’utilisation de fichiers XML pour sauvegarder l’information des questionnaires qui seraient traiter par le site FSL. Suite à une discussion avec M. Iglewski, nous en sommes venus à la conclusion que XML serait plus approprié pour les besoins du projet.

Une des raisons principales est qu’avec XML nous pouvons utiliser XSLT pour faciliter la transformation des données brutes en une page web entièrement fonctionnelle et attrayante pour les utilisateurs.

De plus, l’approche d’un fichier par questionnaire facilite la compréhension de la hiérarchie utilisée pour ordonner l’information nécessaire à chaque questionnaire (titre, url du vidéo, questions, choix de réponse, etc.).

## Recherche d’outils existants

**YouTube**

L’un des outils qui m’a apparu le plus évident dès le début du projet était YouTube. En effet, le site permettait d’héberger des vidéos sans avoir à s’inquiéter de la bande passante. De plus, YouTube fournit une API complète pour faciliter l’interaction entre le lecteur vidéo et javascript. Par exemple, on peut spécifier d’aller à un certain intervalle de temps en faisant appel à un script javascript via un bouton input html standard.

Voir: <http://code.google.com/apis/youtube/getting_started.html>

**jQuery**:

Voici la description officielle de jQuery telle que sur la page principale de leur site (<http://jquery.com/>):

*jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to change the way that you write JavaScript.*

En d’autres mots, jQuery étend les capacités de javascript et facilite son utilisation pour diverses fonctionnalités (surtout l’ajout d’autres plugins). Pour les besoins du projet, jQuery est utilis. Pour pouvoir avoir les fonctionnalités de jQueryUI. Ce plugin est utilisé pour enrichir l’interface graphique avec des onglets et des « fenêtres » style pop-up sans faire appel à une page externe (la fenêtre est en fait un élément flottant).

Voir: <http://jquery.com/>

**CKEditor**:

CKEditor est un module en javascript permettant d'incorporer un éditeur WYSIWYG (What You See Is What You Get) HTML pour faciliter l'édition des questionnaires pour un utilisateur ne connaissant pas HTML. Le module sert donc à fournir un éditeur similaire à Word via un fureteur.

Voir: <http://ckeditor.com/>

**Swfobject**:

Swfobject est une librairie javascript permettant de contrôler les objets SWF (Shockwave Flash) Dans ce cas-ci, l’objet SWF est le lecteur YouTube. C’est la librairie recommandée officiellement par YouTube pour interagir avec leur lecteur.

Voir: <http://code.google.com/p/swfobject/>

# Progrès dans le développement d’une solution

## Format des questionnaires en XML

Créer une structure de données pouvant contenir toutes les informations nécessaires pour les questionnaires a été la tâche centrale pour pouvoir faire les autres (créer une interface d’utilisateur pour les étudiants et pour les professeurs).

Voici un fichier type d’un questionnaire à une seule question :

<?xml version="1.0" encoding="ISO-8859-1"?>

<Questionnaire>

 <Titre>Infoman</Titre>

 <YouTubeLink>http://www.youtube.com/watch?v=60syuCvRvyw</YouTubeLink>

 <Auteur>Alain Charbonneau</Auteur>

 <Question>

 <TypeQuestion>short</TypeQuestion>

 <Enonce>

 En quelle année était la dernière fois que le Prince Charles a visité Montréal?

 </Enonce>

 <StartTime>55</StartTime>

 <EndTime>100</EndTime>

 <Ponderation>10</Ponderation>

 <Reponses>

 <Reponse pts="2">76</Reponse>

 <Reponse pts="2">1976</Reponse>

 </Reponses>

 </Question>

</Questionnaire>

On remarque donc qu’un questionnaire a un titre, un lien vers la vidéo, un auteur et ensuite une série de questions. Ces questions ont un type, un énoncé, un temps de début et de fin (dans la vidéo), une pondération et des choix de réponses ayant chacun une certaine valeur en termes de points.

## Interface étudiant

L’interface de l’étudiant est partiellement complétée. Voici ce qui fonctionne jusqu’à maintenant. Voir section 4 pour ce qui reste à faire.



Figure - Interface de l'étudiant

* L’interface de l’étudiant permet à l’utilisateur de choisir un questionnaire et d’y répondre.
* L’interface a donc un visionneur vidéo à la gauche et le questionnaire à la droite. Cela permet de répondre aux questions tout en écoutant la vidéo.
* Les questions sont séparées par des onglets se déroulant. Cela permet d’éviter à avoir à utiliser la barre de défilement pour alterner entre la vidéo et les questions.
* Chaque question a un bouton/lien qui permet de positionner la vidéo au temps spécifié par l’auteur du questionnaire. Cela permet d’éviter à chercher la réponse à une question dans la vidéo en entier.
* La pondération de chaque question est donnée entre parenthèses
* Il y a trois types de questions :
	+ Cases à cocher (une ou plusieurs réponses parmi les choix)
	+ Radio (une seule réponse parmi les choix)
	+ Courte (l’utilisateur entre une réponse à la main)

## Interface professeur

L’interface du professeur est partiellement complétée. Voici ce qui fonctionne jusqu’à maintenant. Voir section 4 pour ce qui reste à faire.



Figure - Interface du professeur

* L’interface du professeur permet à l’utilisateur de créer des questionnaires en spécifiant une adresse d’un vidéo YouTube, un titre et une description du questionnaire.
* L’interface permet d’ajouter des questions parmi les trois types précédemment présentés (Cases à cocher, radio et réponse courte).
* L’énoncé peut être modifié facilement grâce à l’interface graphique conviviale de CKEditor.
* Le questionnaire peut être sauvegardé en fichier XML de façon transparente à l’utilisateur.
* Le professeur peut téléverser des fichiers annexes.

# Travail restant

Comme vous pouvez le constater, l’apparence générale du module est avancée et la direction du projet est bien comprise et définie. Il reste néanmoins plusieurs détails à compléter. Voici une liste de ceux-ci avec une brève description pour chacun :

* Améliorer l’interface pour la gestion de plusieurs questionnaires. En ce moment, l’accent est mis sur l’utilisation d’un seul questionnaire pour des raisons techniques (il est plus facile de mettre l’accent sur un questionnaire que sur plusieurs à la fois).
* Intégrer les fonctionnalités reliées aux fichiers annexés aux questionnaires (permettre de faire jouer un mp3 qu’un professeur a téléversé par exemple).
* Avoir un module de correction du questionnaire lorsqu’un étudiant soumet ses réponses
* Améliorer la documentation et les commentaires dans le code source.