

Sommaire

I – Partie Etudiant	page 02
II – Partie Administrateur	page 02
III - Choix techniques :	page 05
IV - Problèmes rencontrés	page 05
V - Créer un test en PDF	page 06
1 – Exigences	page 06
2 – Réalisation.....	page 06
3 - Problèmes rencontrés	page 08
4 – Fonctionnement de la classe	page 10
Annexes	page 11
Adobe Document Server for Reader Extensions®.....	page 11
Diagramme de classe.....	page 12

Dossier de Conception et de Codage

Dans ce dossier nous allons rapidement décrire, pour chaque partie, les exigences du Client, les choix que nous avons du faire, les problèmes rencontrés et leurs solutions.

Nous décrivons également l'architecture du site, ce qui peut être utile si des modifications se révèlent nécessaires.

I – Partie Etudiant

Les différentes fonctions disponibles dans cette partie sont les suivantes :

- Passer un test d'Auto-positionnement
- Passer un test (examen)
- Consulter ses archives (tests déjà passés)

Auto-positionnement :

Les tests d'auto-positionnement permettent une évaluation précise des connaissances, par une suite de questions. Le candidat doit lui-même déterminer s'il sait faire ou non certaines actions.

Fichiers concernés : les fichiers du répertoire « qcmposi »

Passer test (examen) :

Une fonctionnalité demandée était le fait de pouvoir passer un examen en PHP lors de la certification. Le site web sélectionne les questions dans la base selon l'étudiant et son numéro de test. L'étudiant répond aux questions, puis il obtient une correction juste après l'envoi de ses réponses qui sont sauvegardées dans la base de données.

Fichiers concernés : qcm.php, question.php, resultats.php

Archives :

Un étudiant peut accéder aux corrections de ses examens déjà effectués.

Fichiers concernés : archives.php, question.php

II – Partie Administrateur

Cette partie est accessible via un lien placé dans la partie Etudiant.

Les différentes fonctions disponibles dans cette partie sont les suivantes :

- Créer un test (php ou pdf)
- Gestion des questions
- Export des candidats en CSV
- Import LDAP de candidats
- Notes des étudiants (Consulter)
- Insérer une question pour les tests de positionnement
- Données sur les candidats
- Statistiques par référentiels
- Statistiques par questions
- Archives

- Autres

Créer un test en php

L'administrateur peut créer un QCM en PHP en choisissant divers paramètres tels que : l'intitulé du QCM, le nombre de question, le niveau de difficulté minimum et maximum, le nombre de points par réponse bonne/mauvaise/non répondue, la date du test, le nom du créateur, et le pourcentage de questions par référentiels.

Le script va ensuite créer le QCM en piochant au hasard des questions en fonction des paramètres que l'administrateur a choisi. Pour gérer les paramètres de choix de questions, on utilise une table temporaire. Le script remplit les différentes tables ChoixQuestion, SessionTest, Test.

Fichiers concernés : creaqcmphp2.php creation_qcm.php creaqcmphp.php resultats.php

Créer un test en pdf

Comme cette partie est relativement complexe, elle a été développée en une quatrième partie. Voir donc plus loin la partie V

Fichiers concernés : resultats.php, QCM.php

Gestion des questions

Cette partie permet de visionner, ajouter, modifier, ou supprimer des questions

Exports des candidats en CSV

Export des candidats du format MySQL vers un fichier csv. Cette fonctionnalité permet à l'administrateur d'exporter tous les étudiants contenus dans la base de donnée vers un fichier au format csv qui pourra être importé dans la base de donnée nationale.

La table Candidat dans notre base MySQL se présente sous la forme :

IdLogin – nom – prenom – mdp – ine - promotion

Le script parcourt donc la table et la met sous un format csv, les différents champs sont donc séparés par des ";".

Fichiers concernés : export_csv.php export_csv2.php

Import LDAP de candidats

Le script nous permet de récupérer sur un annuaire LDAP les candidats et leurs promotions respectives. L'administrateur remplit un champ qui correspond à l'université dont il veut récupérer les étudiants. Ensuite le script parcourt toutes les promotions, et pour chaque promotion il liste les étudiants et les ajoute dans la base de donnée.

Fichiers concernés : import_ldap.php

Notes des étudiants :

L'administrateur doit pouvoir récupérer les notes de tous les étudiants ayant effectués un même test.

Fichiers concernés : notes.php, statistiques.php

Insérer une question pour les tests de positionnement

Les QCM de positionnement permettent aux étudiants de se tester facilement, et connaître leur niveau dans différents domaines. L'étudiant choisit donc le domaine dans lequel il veut se tester, puis coche les choses qu'il connaît, et on lui donne une adresse web d'aide pour les éléments qu'il ne connaît pas. Cette fonction permet l'ajout de nouvelles questions

Fichiers concernés : insquestposi.php insposi2.php

Données sur les candidats :

L'administrateur doit pouvoir regarder la liste des candidats. Il peut aussi modifier les données des candidats ou les supprimer.

Fichiers concernés : stats_Candidats.php, Statistiques.php

Statistiques par référentiels :

L'administrateur peut consulter les statistiques (taux de réussite, taux d'échec, taux d'abstention...) générales pour chaque référentiel.

Fichiers concernés : stats_référentiel.php, Statistiques.php

Statistiques par questions :

L'administrateur peut consulter les statistiques (taux de réussite, taux d'échec, taux d'abstention...) générales pour chacune des questions.

Fichiers concernés : stats_questions.php, Statistiques.php

Archives :

L'administrateur peut voir la correction de n'importe quel candidat en sélectionnant l'identifiant unique.

Fichiers concernés : archives.php, question.php

Autres :

Identification :

Les différentes parties du site sont accessibles si et seulement si on est identifié en tant qu'utilisateur ou en tant qu'administrateur.

Fichiers concernés : verification.php

Interface

L'interface générale du site devait être homogène et ergonomique. L'utilisation de feuilles de style permet une modification facile de l'interface.

Fichiers concernés : style.css

Base de données

L'application a besoin d'une base de données afin de sauvegarder les différentes questions, corrections, statistiques...

Fichiers concernés : bdd_table.sql, bdd_tuples.sql

III - Choix techniques :

Tout d'abord nous avons choisi de séparer la partie admin de la partie utilisateur en la mettant dans un répertoire « admin », afin de pouvoir réutiliser et adapter à une autre interface les différentes fonctions du code. Nous avons utilisé des classes php pour implémenter la plupart des fonctions qui seront ainsi facilement réutilisables.

Pour l'interface le choix d'un style css semblait le plus judicieux étant donné que cela permet de modifier à volonté et simplement l'interface générale du site. La partie administrateur a également été séparée dans l'interface, puis nous avons ajouté un lien demandé par le client pour pouvoir facilement changer de partie (partie utilisateur partie administrateur et vice versa).

Pour l'identification, le choix le plus pratique était l'utilisation des cookies. En effet il y avait deux solutions pour garder en mémoire des informations concernant l'utilisateur. Les variables de session et les cookies. Le choix du cookie est venu spontanément tant par la simplicité d'utilisation et la sûreté (car seul le serveur peut modifier le contenu).

IV - Problèmes rencontrés

Pour l'ergonomie du site, le problème majeur fut de trouver un moyen de créer différentes interactions (comme la création d'un popup de confirmation), le langage habituellement utilisé étant javascript, on s'est donc aperçu de plusieurs problèmes liés à son utilisation, le problème étant que le php s'exécute côté serveur et javascript côté client. De plus le javascript n'est pas du tout fiable et sécurisé. Donc pour pallier à ce problème nous avons décidé de limiter son utilisation à des parties sécurisées par un cookie.

La connexion et le parcours de l'annuaire LDAP nous ont également causés beaucoup de problèmes, car aucun de nous ne connaissait LDAP avant le projet. Après beaucoup d'efforts, nous avons réussi à utiliser LDAP.

La création du fichier candidats.csv ne marche pas correctement à l'IUT en raison de la configuration et des droits associés à PHP sur le serveur web de www-ens. Le script marche cependant parfaitement sur n'importe quel autre serveur.

V - Créer un test en PDF

1 - Exigences

Une des fonctionnalités demandées était la création de qcm en pdf, statiques, et dynamiques. (Les qcm dynamiques sont réalisés sous la forme d'un document pdf dans lequel est inséré du code Javascript.). Cette classe devait donc être capable de générer des documents pdf, contenant éventuellement du javascript.

Les fonctionnalités exigées étaient les suivantes (certaines ont été rajoutées en cours de route) :

- Création de qcm statique : les questions sont choisies aléatoirement dans la base de donnée, tout en respectant un niveau de difficulté, et un nombre de questions par référentiel qui seront définis par le créateur du qcm. Le créateur a également la possibilité de choisir le titre de son qcm.
- Création de qcm dynamique : les qcm dynamiques reprennent les fonctionnalités du qcm statique, mais permettent également :
 - Une auto-correction du qcm lorsque l'on clique sur un bouton.
 - La possibilité d'ajouter un mot de passe pour autoriser la correction.
 - L'affichage du temps écoulé, avec possibilité de définir un temps maximum. Une fois ce temps maximum atteint, l'utilisateur ne peut plus modifier ses réponses.
 - Une fois le document corrigé, l'affichage d'un lien vers une ressource Internet où l'on peut trouver plus d'informations.

2 – Réalisation

La première question à se poser était la création de document pdf à partir de php. Il s'avère que plusieurs librairies ont été créées dans ce but, et nous avons donc cherché quelle serait la meilleure solution.

Il existe de nombreuses librairies permettant la création de pdf. Les librairies les plus connues étant :

- FPDF
- PDFlib
- ClibPDF

FPDF – Les points forts

- Gratuit, proposé en open source.
- FPDF ne nécessite aucune extension/librairie et est compatible PHP3 et PHP4
- Le projet est bien suivi par ses créateurs, et de nombreuses évolutions sont attendues :

- Rajout de polices,
- Liens,
- Compression des pages,
- Figures géométriques,
- Possibilité de choisir les bords des cellules à tracer (plutôt que tout ou rien),

- Signets,
- Formats autres que l'A4.

- La classe est écrite en php et est donc facilement modifiable et adaptable.
- Simple à utiliser, un simple "include" suffit et un tutorial très bien fait est à disposition.
- C'est en français.

FPDF – Les points faibles

- La vitesse de génération du document est plus faible qu'avec PDFlib.
- Le document généré est plus volumineux (en attendant la possibilité de compression).
- Limité à 4 polices pour le moment.

Cependant, l'écart de performance reste très raisonnable et convient dans la plupart des cas, à moins que les documents ne soient particulièrement complexes ou lourds.

PDFlib – Les points forts

Fonctionnalités sont les plus complètes du domaine :

- Taille de page et systèmes de coordonnées
- Couleurs
- Polices embarquées ou non
- Texte justifié, aligné, etc.
- Images (embarquement, échelle, déformation)
- Dessins (courbe, droite, cercle, arc de cercle)
- Modèles
- etc.

De plus, PDFlib est très stable, très rapide, et supporte la compression des documents pdf créés.

PDFlib – Les points faibles

- Difficile à installer et mettre en place.
- et surtout...non gratuite.

ClibPDF – Les points forts

- Gratuit,
- permet de générer des pages très complexes.

ClibPDF – Les points faibles

- écrit en C
- Difficile à installer
- Pas de support/site français.

Conclusion

- La PDFlib est la plus stable et la plus rapide, mais un peu difficile à mettre en place.
 - La ClibPDF est la plus poussée, mais est écrite uniquement en C, ce qui limite son adaptabilité et complique son utilisation.
 - La FPDF est la moins avancée mais est largement suffisante pour un usage normal. Son extrême simplicité à mettre en place et à utiliser est son atout majeur.
- De plus, la FPDF est vouée à évoluer au fil du temps.

3 – Problèmes rencontrés

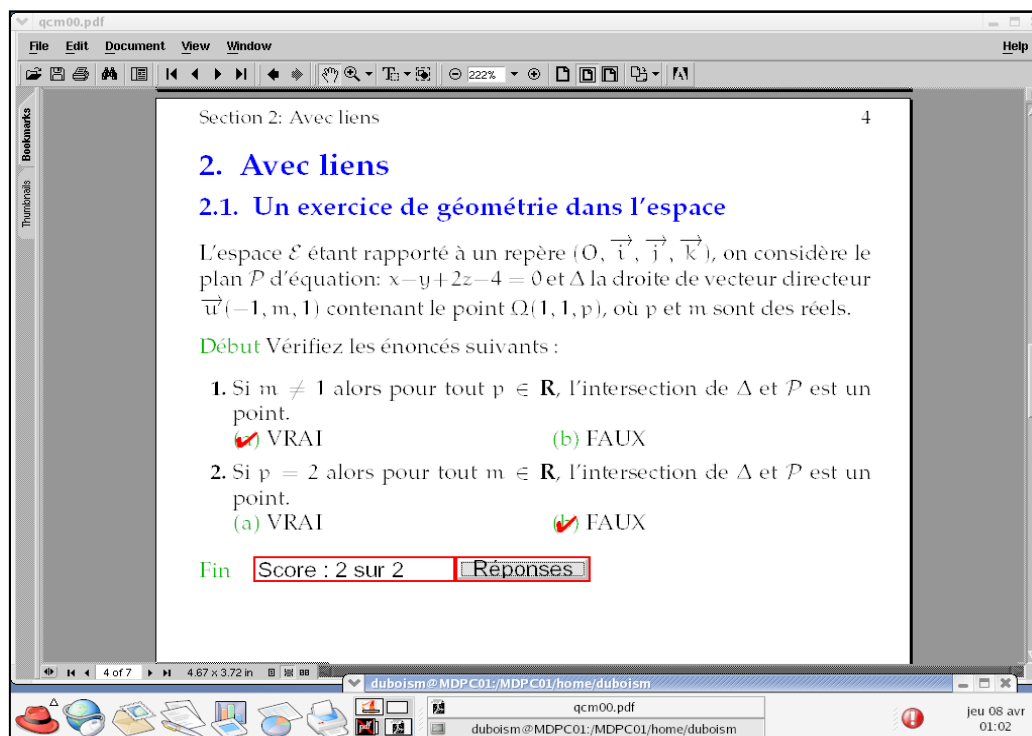
Le codage en lui-même n'a pas posé de problème, puisque le site officiel de FPDF est bien fait, et contient quelques exemples fort utiles. Il reste cependant des améliorations possibles, comme par exemple faire s'afficher des *** au lieu du texte dans le champ du mot de passe.

Problème de compatibilité

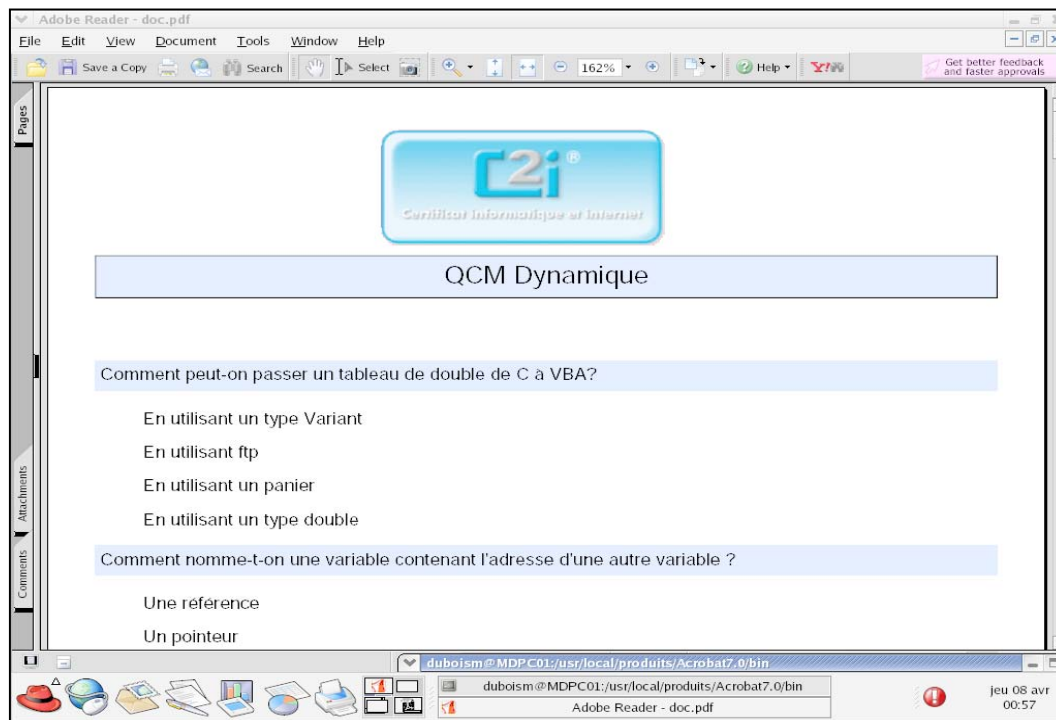
Le vrai problème a malheureusement été découvert à la fin de la période de codage, ce qui m'a laissé peu de temps pour trouver une solution. Merci à M. Dubois qui a grandement aidé Constant sur ce point.

En effet, une fois le qcm dynamique généré, celui-ci n'était lisible que dans Acrobat Writer ® mais ne semblait pas être lisible avec les versions gratuites de Acrobat Reader ®.

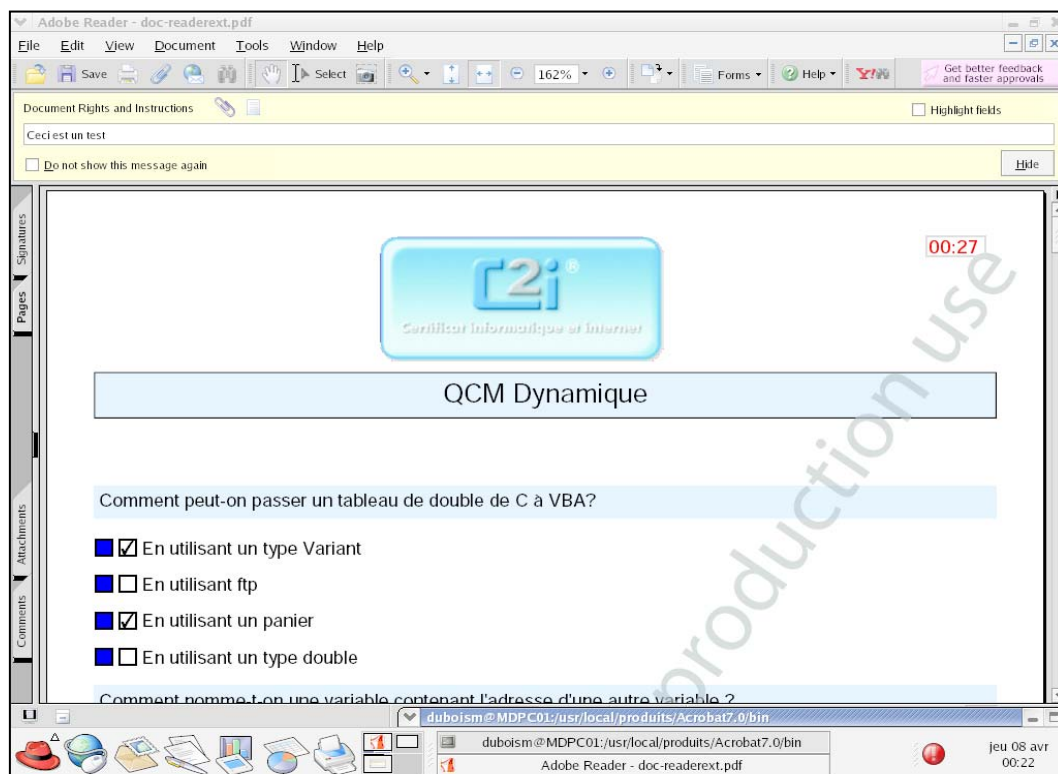
Après une recherche plus approfondie, il s'avère que seule la version 5.1 d'Acrobat Reader ® est capable d'ouvrir les pdf dynamiques. Cela est dû à l'utilisation de champs qui ne seraient pas gérés dans les autres versions.



Les QCM générés à l'aide d'acrotext fonctionnent parfaitement sous les versions 5.010 et 7 d'Acrobat Reader® pour linux mais au prix d'une ergonomie un peu lourde, ce qui prouve bien que ces versions supportent le javascript.



Un document généré avec FPDF ne peut afficher les champs sous linux, et ne fonctionne qu'avec la version writer, ou reader 5.1 pour windows.



Sous Linux, Le document avec droit (modifié avec Adobe Document Server for Reader Extensions®) est reconnu comme tel et les champs sont affichés. Demeurent malheureusement quelques bugs... Sous Windows et avec la version 7, le document marche parfaitement. Il faut cependant noter que la taille du fichier a été multipliée par 10...

Conclusion

Les deux solutions les plus simples pour utiliser les pdf produits (sans tout reprogrammer) sont donc :

1 – Distribuer la version 5.1 d'Acrobat Reader ® avec le pdf. Cette solution est un peu contraignante pour l'utilisateur, qui peut avoir à installer la version 5.1 même s'il possède une version plus récente.

2 – Donner les droits au fichier avec Adobe Document Server for Reader Extensions®. Cette solution demande un peu plus de travail, mais le résultat est plus portable et donc plus intéressant pour l'utilisateur. Sans doute la solution à adopter. Pour plus d'informations sur Adobe Document Server for Reader Extensions®, voir en Annexe.

4 – Fonctionnement de la classe

La Classe permettant la création de qcm en pdf, statique ou dynamique, est dans le fichier QCM.php, dans le dossier creaqcm, dans le dossier admin.

Cette classe est appelée par le fichier resultats.php placé dans le même dossier. Notons que toutes les possibilités de la classe ne sont pas utilisées... On peut normalement spécifier le temps maximum alloué pour la résolution du qcm, l'affichage ou non du compteur, le changement du mot de passe autorisant la correction, ou même la suppression de celui-ci.

Ces options ne sont pas disponibles car nous manquons de temps pour les implémenter, cependant, il est assez facile de les modifier.

La fonction, est appelée de cette manière :

```
function createQcm($needMdp, $mdp, $bddUser, $bddPass, $bddName, $bddPlace,
$download, $showTime, $min, $ref, $nbQuestions, $pts, $dynamique, $titre){
```

et voilà le détail des différents paramètres :

\$needMdp : 0-> pas de mdp ; 1-> mot de passe (pour la correction)

\$mdp : mot de passe pour accéder à la correction (le mdp par défaut est 'test')

\$bddUser : Nom d'utilisateur pour la base de données

\$bddPass : Mot de passe de la base de données

\$bddName : Nom de la base de données

\$bddPlace : Location de la base de données

\$download : 0-> le fichier s'ouvre dans le navigateur ; 1-> Téléchargement fichier

\$showTime : 0-> pas de temps ; 1-> Affichage du temps

\$min : Temps max (en minutes) pour faire le qcm 0-> temps infini

\$ref : Contient les référentiels dans lesquels on doit puiser les questions

\$nbQuestions : Contient le nombre de question à utiliser (par référentiel)

\$pts : Contient dans l'ordre : les points pour une bonne réponse, une mauvaise réponse, une question non répondue

\$dynamique : 0-> pdf statique; 1-> pdf dynamique

\$titre : Titre du QCM

Il est donc facile de modifier les différents paramètres en fonction de vos préférences.

Annexes

**Adobe Document Server for Reader Extensions
Version 6.0**

Adobe® Document Server 5.0/6.0 for Reader Extensions permet d'activer plusieurs fonctions interactives dans les documents PDF existants. Ces fonctions étaient auparavant accessibles uniquement aux utilisateurs de la version intégrale d'Acrobat®. Une fois les fonctions activées dans le document PDF, les utilisateurs qui l'ouvrent avec le gratuiciel Acrobat® Reader® version 5.1 ou ultérieure peuvent accéder à ces fonctions et manipuler le document. Ces fonctions regroupent :

- L'ajout de commentaires;
- L'application de la fonction de signatures numériques;
- L'utilisation de fonctions de formulaire avancées, telles que l'importation et l'exportation de données;
- L'enregistrement du document PDF modifié.

Pour activer les fonctions dans un document PDF, il faut y ajouter des droits d'utilisation. Pour ce faire, on utilise le client Document Server for Reader Extensions, qui est une application Web. Une fois les droits d'utilisation ajoutés, le document PDF est prêt à être distribué à la base d'utilisateurs. Si un utilisateur ouvre un document PDF avec Acrobat Reader version 5.1 ou ultérieure, les fonctions choisies lui sont immédiatement accessibles. Si un utilisateur ouvre un document dans une version d'Acrobat Reader antérieure à la version 5.1 ou dans Acrobat® Approval, un message s'affiche pour informer l'utilisateur qu'une mise à niveau est nécessaire pour pouvoir accéder à toutes les fonctions. Le message affiche aussi un lien vers un URL à partir duquel il est possible de télécharger Acrobat Reader.

Diagramme de classe :