

## Extension de Moodle :

Tests interactifs

Mémoire présenté par  
**François GRIMART**

en vue de l'obtention du grade de Master  
**sciences informatiques**

Promoteur(s)  
**Yves DEVILLE**

Lecteur(s)  
**Isabelle MOTTE, Frédéric FERVAILLE , Charles PECHEUR**

Année académique 2015-2016



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	Moodle . . . . .	7
2.1.1	Présentation générale . . . . .	7
2.1.2	Tests . . . . .	8
2.1.3	Services web . . . . .	8
2.2	Logiciels de tests interactifs . . . . .	9
2.2.1	Socrative . . . . .	9
2.2.2	Ipal . . . . .	10
2.2.3	Active Quiz . . . . .	10
2.2.4	Comparaison des fonctionnalités . . . . .	11
2.2.5	Wooclap . . . . .	12
<b>3</b>	<b>Cahier des charges</b>	<b>13</b>
3.1	Définition du problème . . . . .	13
3.2	Définition des acteurs : . . . . .	13
3.3	Objectif . . . . .	14
3.4	Contrainte . . . . .	14
3.5	Exigences fonctionnelles . . . . .	14
3.5.1	Fonctionnalités de l'enseignant : . . . . .	14
3.5.2	Fonctionnalités de l'étudiant : . . . . .	16
3.6	Exigences non-fonctionnelles . . . . .	16
<b>4</b>	<b>Développement</b>	<b>18</b>
4.1	Framework . . . . .	18
4.1.1	Choix du framework . . . . .	18
4.1.2	Framework Phalcon . . . . .	20
4.1.3	Design Framework . . . . .	22
4.2	Architecture . . . . .	22
4.2.1	Services web . . . . .	22
4.2.2	Serveur Websocket . . . . .	24

4.2.3	InteractiveQuiz . . . . .	26
4.3	Déploiement . . . . .	30
4.3.1	Moodle . . . . .	30
4.3.2	InteractiveQuiz . . . . .	30
<b>5</b>	<b>Expérimentations et analyses</b>	<b>32</b>
5.1	Résultat . . . . .	32
5.1.1	Fonctionnalité commune : . . . . .	32
5.1.2	Fonctionnalités de l'enseignant : . . . . .	33
5.1.3	Fonctionnalités de l'étudiant : . . . . .	35
5.2	Limitations . . . . .	36
5.3	Comparaison des performances . . . . .	37
<b>6</b>	<b>Conclusion</b>	<b>39</b>
	<b>Bibliographie</b>	<b>40</b>
	<b>Annexes</b>	<b>42</b>
<b>A</b>	<b>Tableaux comparatifs de Socrative, Ipal et ActiveQuiz</b>	<b>42</b>
A.1	Enseignant . . . . .	42
A.2	Etudiants . . . . .	43
<b>B</b>	<b>Mesures des performances d'ActiveQuiz et d'InterActiveQuiz</b>	<b>45</b>
B.1	ActiveQuiz . . . . .	45
B.2	InteractiveQuiz . . . . .	46

## Résumé

L'objectif de ce mémoire est de développer une extension pour la plate-forme d'apprentissage en ligne Moodle qui pourra être utile pour l'UCL.

Moodle permet, entre autre, aux enseignants de créer des tests pour leurs étudiants. Il met pour cela à disposition de nombreux outils comme par exemple le partage des questions, la correction automatique, de nombreux types de questions,...

L'utilisation de tests interactifs durant un cours procure plusieurs avantages. Cela permet à l'enseignant d'avoir un feedback sur le niveau de compréhension des étudiants. Les étudiants peuvent se sentir plus impliqués grâce à l'interaction. Cette utilisation permet aussi un apprentissage plus actif.

Notre extension se présente sous la forme d'une application web. Cette application donne la possibilité aux enseignants de pouvoir utiliser leurs tests, créés sur Moodle, de manière interactive. Ils pourront diffuser leurs tests pendant leurs cours et voir en direct les réponses des étudiants.

Notre application permet donc de combiner à la fois des outils complets, proposés par Moodle, pour la construction des tests avec les avantages que procure le fait d'utiliser ces tests de manière interactive.

# Chapitre 1

## Introduction

Rendre un cours attractif et vivant n'est pas toujours une chose aisée pour un enseignant. La plupart des étudiants viennent aux cours avec un ordinateur ou un smartphone. Cela offre la possibilité d'intégrer facilement ces nouveaux outils à l'enseignement d'aujourd'hui. Une manière d'employer ces outils est de les utiliser pour réaliser des tests interactifs.

Nous définirons un test interactif comme étant un test qui se déroule lors d'un cours et qui permet à l'enseignant d'obtenir les résultats en direct et de les partager instantanément avec ses étudiants.

Nous avons pu constater plusieurs avantages aux tests interactifs. Nous en développerons trois[27].

Le premier est le quality feedback, autrement dit un retour d'information de qualité. Il désigne le retour d'information, selon les réponses données, du groupe classe vers l'enseignant et permet à celui-ci d'avoir un aperçu rapide et global sur le niveau de compréhension que les étudiants ont obtenu. Un bon feedback doit être spécifique à la connaissance en cours d'acquisition et dispensé rapidement après l'évaluation [21]. Il est également important qu'il soit correctif et explicatif (les erreurs sont pointées et des explications doivent être données afin de ne plus les commettre). De plus, le feedback doit permettre à l'étudiant de suivre de près sa propre progression.

L'utilisation des tests interactifs permet aussi un engagement actif des étudiants dans la manipulation et l'application des idées, ce qui nous amène au deuxième avantage : l'interactivité. Le feedback permet une interactivité entre les étudiants et l'enseignant. Or, nous constatons aussi une interactivité entre étudiants puisque l'usage des tests interactifs incite des discussions entre étudiants avant ou après le vote [24].

La transmission directe de contenus ne serait pas le moyen le plus efficace pour enseigner [26]. En effet, l'apprentissage actif est plus utile que l'apprentissage passif [25]. Cette affirmation concerne par exemple la participation des élèves. Il existe pour cela plusieurs moyens d'enseignement qui font appel aux tests interactifs : une question centrale ciblée sur le sujet mène à une réflexion individuelle ou de groupe qui conduit à un vote puis à une discussion en fonction des résultats. Ceux-ci sont projetés sous forme d'un graphique afin que tout le monde en prenne connaissance. L'enseignant décide alors soit d'expliquer une nouvelle fois, soit d'amener une

discussion en petits groupes d'élèves, soit de passer au sujet suivant s'il estime que la matière a bien été comprise.

L'objectif de ce mémoire est de développer une extension pour la plate-forme d'apprentissage en ligne Moodle qui pourra être utile pour l'UCL. Moodle dispose d'outils performant pour permettre aux enseignants de mettre en place des tests standards pour les étudiants. Au vu des avantages que peuvent procurer des tests interactifs, nous avons choisi de développer cette fonctionnalité afin de pouvoir utiliser les tests de Moodle de manière interactive.

Dans le background, nous présenterons brièvement le logiciel Moodle avant d'aborder sa fonctionnalité de tests et la possibilité que permet Moodle d'utiliser des services web. Nous présenterons et comparerons également quelques logiciels permettant l'utilisation de tests interactifs.

Dans le cahier des charges, nous définirons le contexte du problème, énoncerons le résultat attendu avant de décrire l'ensemble des exigences fonctionnelles et non-fonctionnelles.

Dans le chapitre destiné au développement, nous aborderons différents aspects du développement du site Interactive-Quiz. Nous expliquerons pourquoi nous avons choisi le framework phalcon et en quoi consiste-t-il. Nous présenterons aussi l'architecture d'Interactive-Quiz en décrivant ses principales parties avant d'énoncer les prérequis nécessaires à son déploiement.

Enfin, nous analyserons les résultats obtenus suite au développement, aborderons les différences entre ce résultat et le cahier des charges, et effectuerons une comparaison des performances entre notre site le plugin ActiveQuiz pour Moodle.

# Chapitre 2

## Background

Ce chapitre contiendra toutes les informations qui nous ont été nécessaires pour la réalisation du cahier des charges. Nous commencerons d'abord par une brève présentation du logiciel Moodle. Nous détaillerons ensuite sa fonctionnalité de tests. Puis, nous présenterons la possibilité qu'offre Moodle d'utiliser des services web pour permettre des interactions entre celui-ci et d'autres sites. Et enfin nous décrirons quelques logiciels existant permettant l'utilisation de tests interactifs. Nous les avons comparés pour pouvoir définir un ensemble de fonctionnalités nécessaires pour la réalisation de tests interactifs.

### 2.1 Moodle

#### 2.1.1 Présentation générale

Moodle est une plate-forme d'apprentissage en ligne (Learning Management System). Ce site a été créé par Martin Dougiamas en 2002. Il est open source, distribué sous la licence GPL (General Public License) [20]. Il est utilisé par plus de 88 millions d'utilisateurs via plus de 77000 sites enregistrés [9]. Moodle est l'acronyme de Modular Object Oriented Dynamic Learning Environment (Environnement d'apprentissage orienté-objet, dynamique et modulaire).

Moodle permet aux enseignants de créer des cours et d'y intégrer des ressources et des activités. Il donne la possibilité de gérer plusieurs types de ressources : dossier, étiquette, fichier, livre, page, paquetage IMS content et url [13]. Les activités peuvent être sous la forme d'atelier, base de données, chat, consultation, devoir, feedback, forum, glossaire, leçon, outil externe, paquetage SCORM, sondage, test, wiki [3]. Moodle fournit une interface personnalisable. Son interface est composée de blocks qui peuvent être masqués ou réduits. Les fonctionnalités peuvent être activées ou désactivées en fonction des besoins. Il est aussi disponible en plusieurs langues. Il jouit d'une communauté active qui participe à son développement en fournissant un grand nombre de modules complémentaires. Cela permet l'ajout de nombreuses autres fonctionnalités via des plugins [8]. Il dispose d'une application : Moodle Mobile [4] qui fournit une grande partie des fonctionnalités de base de Moodle.

L'UCL a décidé d'effectuer la migration de sa plateforme Icampus vers MoodleUCL en 2015[2].

### 2.1.2 Tests

Moodle permet aux enseignants d'ajouter des activités Tests dans leurs cours pour leur donner la possibilité d'évaluer les connaissances ou le niveau de compréhension de leurs étudiants.

Les tests peuvent être composés de différents types de questions : calculée, calculée simple, composition, appariement, clause à réponse intégrée, choix multiples, réponses courtes, numérique, vrai/faux, calculée à choix multiples, appariement aléatoire à réponses courtes, glisser-déposer sur une image, glisser-déposer sur un texte, marqueurs à glisser-déposer et description [16].

Hormis les questions de composition, les questions de tous types peuvent être corrigées automatiquement. Un rapport pour chaque test de chaque étudiant est produit pour permettre à l'enseignant de visualiser l'ensemble des réponses de l'étudiant.

Les nouvelles questions peuvent être ajoutées à une banque de questions dans le but d'être partagées et réutilisées pour d'autres tests.

L'enseignant peut permettre des tentatives multiples et adapter le système de notation. Un feedback par question peut être affiché après chaque réponse ainsi qu'un feedback global. Les réponses peuvent aussi ne pas être montrées. Il peut fixer la disponibilité d'un test en fonction des groupes ou des utilisateurs et choisir le nombre de questions par page. L'ordre peut être aléatoire pour les questions et les réponses. [15]

### 2.1.3 Services web

D'après le World Wide Web Consortium (W3C), un service web est : *"un logiciel créé pour supporter des interactions interopérables de machine à machine à travers un réseau. Il a une interface décrite dans un format exploitable par des machines (spécifiquement WSDL). D'autres systèmes interagissent avec le service web d'une manière prescrite par sa description en utilisant des messages SOAP, typiquement convertis en utilisant HTTP avec une sérialisation XML en conjonction avec d'autres standards web"* [1].

Moodle fournit une API (Application Programming Interface) proposant des fonctions utilisables par des services web pour la plupart de ses fonctionnalités de base [19]. Cela permet donc à des programmes tiers d'utiliser ces fonctionnalités.

Un service doit d'abord être créé au sein de Moodle pour lister l'ensemble des fonctions que celui-ci mettra à disposition du programme tiers. Une fois ce service créé, sous la forme d'un plugin, un client pourra faire appel aux fonctions du services en utilisant un protocole de service web. Moodle propose plusieurs protocoles : SOAP, REST et XMLRPC. [18]

Un programme qui utilise des fonctions d'un service web de Moodle devra disposer des identifiants et des mots de passe de ses utilisateurs. Ils seront utilisés pour générer des tokens qui permettent l'identification des utilisateurs par ces fonctions. Ces utilisateurs ont les mêmes droits d'accès aux fonctionnalités de Moodle que ce soit via un service web ou via le site Moodle directement. Par exemple, même via un service web, un étudiant ne pourra pas avoir accès à un cours qui requière une inscription s'il n'y est pas inscrit.

## 2.2 Logiciels de tests interactifs

Vous trouverez dans cette section une présentation de différents services qui permettent la mise en place de tests interactifs. Le principal atout de notre application est la possibilité d'utilisation des tests de Moodle. Nous nous limiterons donc à ne présenter qu'un seul site qui n'a pas cette fonctionnalité, à savoir Socrative<sup>1</sup>. Nous décrirons ensuite les deux plugins, Ipal<sup>2</sup> et Active Quiz<sup>3</sup>, qui ajoutent déjà l'utilisation de tests interactifs à Moodle.

### 2.2.1 Socrative

Socrative est une application proposée par la société MasteryConnect<sup>4</sup>. Elle est disponible sous la forme d'un site web ou d'une application mobile. L'accent est mis sur l'ergonomie et la facilité d'utilisation. L'enseignant doit se créer un compte pour pouvoir en profiter tandis que les étudiants n'auront besoin que d'un numéro de salle pour rejoindre un test. La création d'un test par l'enseignant est très intuitive. Trois types de questions sont disponibles : choix multiples, vrai/faux et questions ouvertes. Des images peuvent être affichées avec les questions. L'enseignant peut choisir s'il passera d'une question à l'autre ou si chaque étudiant pourra avancer à son rythme lors du test et éventuellement retourner à une question précédente. Il pourra aussi décider de faire en sorte que l'ordre des questions ou des réponses soit aléatoire. Un feedback peut être affiché avec la réponse lorsque l'étudiant aura répondu à une question. Les étudiants peuvent avoir la possibilité de commenter chaque question. L'enseignant peut voir en direct le pourcentage d'étudiants qui ont répondu à une question. Les tests peuvent être anonymes ou nominatifs. Dans le deuxième cas, l'enseignant pourra voir les réponses individuelles des étudiants et produire un rapport à la fin du test. Ce rapport comportera le nom de l'étudiant, sa note et l'ensemble de ses réponses. Il pourra être exporté en PDF ou en fichier Excel. Les tests peuvent être partagés entre des enseignants grâce à un numéro d'identification ou à l'aide d'un fichier Excel. Ils peuvent aussi être exporté en PDF.

Outre les tests classiques, un mode "question rapide" est disponible. Il permet à l'enseignant de poser une question oralement à laquelle les étudiants pourront répondre via l'application. Deux autres modes sont mis à disposition. Le premier est le mode "billet de sortie". Il propose des questions typiques à poser à la fin d'un cours pour juger de la compréhension des étudiants. Le deuxième est le mode "course de l'espace" qui met en compétition des étudiants par groupe dont l'objectif est de répondre le plus rapidement possible à un certain nombre de questions. La progression de tous les groupes est affichée chez l'enseignant, sous la forme d'un graphique. Plusieurs limitations peuvent être observées[14]. Seulement cinquante étudiants par test sont garantis pour une expérience optimale. L'utilisation de symboles mathématiques dans les questions ou les réponses est très limitée. Les types de questions sont restreints. La suppression des rapports est impossible. On peut tout au plus les déplacer dans la catégorie archive qui fera office de

- 
1. <http://www.socrative.com/>
  2. [https://moodle.org/plugins/mod\\_activequiz](https://moodle.org/plugins/mod_activequiz)
  3. [https://moodle.org/plugins/mod\\_ipal](https://moodle.org/plugins/mod_ipal)
  4. <https://www.masteryconnect.com/>

corbeille.

### 2.2.2 Ipal

Ipal[23] (In-class Polling for All Learners) est un plugin pour Moodle créé sous la direction de William (Bill) Junkin du Eckerd College<sup>5</sup>. Une fois que le plug-in a été intégré à Moodle, l'enseignant peut créer dans son cours Moodle une activité Ipal. Ensuite, il aura le choix d'ajouter à cette activité des questions de la banque de questions de Moodle, des questions provenant du site <http://www.compadre.org/> qui contient essentiellement des questions concernant le domaine de la physique, ou des nouvelles questions. Ces nouvelles questions peuvent être de trois types : choix multiples, vrai/faux et questions ouvertes. Seulement les questions d'un de ces types peuvent être ajoutées à l'activité à partir de la banque de questions de Moodle. L'enseignant peut choisir si un test sera anonyme et s'il se fera par groupe. Il peut aussi permettre l'utilisation de l'application mobile Ipal pour Android<sup>6</sup> ou IOS<sup>7</sup>.

Lorsque l'enseignant démarre un test interactif, il doit sélectionner une question du test à afficher chez les étudiants. Ceux-ci peuvent alors y répondre. Un graphe montrant la répartition des réponses des étudiants ou un tableau montrant les réponses de chaque étudiant peut être affiché chez l'enseignant dans une nouvelle fenêtre du navigateur.

On peut reprocher à Ipal de ne pas avoir intégré tous les types de questions disponibles dans Moodle. Ses principaux points faibles sont son manque d'ergonomie du côté de l'enseignant et, du côté des étudiants, le design très rudimentaire. L'impossibilité de sauvegarder les réponses des étudiants est un autre défaut.

### 2.2.3 Active Quiz

Active quiz[22] est un plugin pour Moodle créé par John Hoopes pour l'université de Wisconsin-Madison<sup>8</sup>. Ce plugin a été écrit à partir du plugin real-time quiz<sup>9</sup>. Il est donc une évolution de ce dernier. L'enseignant peut créer grâce à lui une activité Active quiz pour son cours sur Moodle. Il pourra y ajouter des questions provenant de la banque de questions Moodle ou des nouvelles questions. Ce plug-in supporte tous les types de questions disponibles par défaut pour les tests de Moodle [16]. Le défilement des questions peut être manuel ou automatique. L'enseignant peut définir la durée des questions et un temps d'attente entre celles-ci. Il pourra aussi profiter du système de notation des tests de Moodle. Il peut décider de laisser le test accessible aux étudiants avec éventuellement leurs réponses et un feedback spécifique à chaque question ou un feedback général. Les tests peuvent être réalisés de manière anonyme et par groupe.

Lorsque le test est en cours, l'enseignant peut afficher un graphe montrant la répartition des réponses en direct. Cependant, le graphe ne peut être affiché seul à l'écran. Toute l'interface de

---

5. <http://www.eckerd.edu/>

6. <https://play.google.com/store/apps/details?id=com.ipalandroid>

7. <https://itunes.apple.com/us/app/ipal/id873943119?mt=8>

8. <http://www.wisc.edu/>

9. [https://moodle.org/plugins/mod\\_realtimequiz](https://moodle.org/plugins/mod_realtimequiz)

l'enseignant sera toujours visible. Pour participer à un test, les étudiants doivent juste rejoindre, via Moodle, une activité Active quiz qui aura été démarrée par l'enseignant et répondre à la question affichée.

#### 2.2.4 Comparaison des fonctionnalités

Vous trouverez en annexe A des tableaux qui reprennent l'ensemble des fonctionnalités des trois logiciels testés. Ceux-ci nous permettent de faire plusieurs observations :

On remarque que Socrative propose plusieurs fonctionnalités intéressantes qui ne sont pas disponibles chez les plugins. La première est le sondage rapide. Cette fonctionnalité étend l'utilisation des quiz interactifs à des questions qui n'auraient pas été préparées à l'avance par l'enseignant. Elle permet aussi d'improviser un sondage ou de recueillir simplement les avis des étudiants. Elle donne donc la possibilité d'augmenter l'interactivité entre l'enseignant et les étudiants sans préparation préalable. Une autre fonctionnalité intéressante présente seulement chez Socrative est la connexion via un numéro de salle. Cela permet de simplifier et d'accélérer la connexion des étudiants à un test contrairement aux plugins qui requièrent de se connecter à Moodle et, ensuite, une navigation à travers celui-ci jusqu'à atteindre l'activité souhaitée. Un temps précieux lors du cours est donc sauvé. La connexion via l'application Ipal est simplifiée grâce à l'utilisation d'un code, ce qui la rend semblable et aussi efficace que la connexion via un numéro de classe de Socrative. On peut aussi noter que le style graphique beaucoup moins austère que celui de Moodle, le rend bien plus convivial à utiliser.

Active quiz se démarque de Socrative par l'utilisation de quatre fonctionnalités importantes de Moodle. La première est la possibilité d'utiliser la banque de questions de Moodle. Cela permet à l'enseignant de réutiliser des questions, déjà faites par lui-même ou partagées avec d'autres enseignants, ce qui facilite la conception des tests. La deuxième fonctionnalité est l'accès à un grand choix de type de questions. Cela donne à l'enseignant plus de liberté dans la manière d'interroger ses étudiants. La troisième fonctionnalité est la possibilité d'utiliser le système de notation. Socrative peut fournir une note à un test via les rapports mais ceux-ci ne sont pas directement accessibles aux étudiants. Et enfin, la quatrième fonctionnalité est de permettre l'accès à la tentative de l'étudiant. Ils peuvent donc retrouver après le test, les questions et leurs réponses, si l'enseignant en donne l'autorisation.

Du côté d'Ipal, son seul atout par rapport à Active quiz est son application mobile. Il permet l'utilisation de la banque de questions de Moodle mais ne dispose pas des autres fonctionnalités intéressantes.

Active quiz semble le plus complet en terme de fonctionnalités. Cependant, comme il est directement intégré dans Moodle, il oblige, pour chaque étudiant, une connexion au site et le chargement de nombreux éléments inutiles au test interactif. Cela peut donc générer un trafic non nécessaire qui peut être préjudiciable dans, par exemple, un grand auditoire comportant un réseau wifi limité. Cette remarque est aussi valable pour Ipal s'il est utilisé via le site.

### 2.2.5 Wooclap

Wooclap<sup>10</sup> est l'application qui a été choisie dans le courant de l'année par l'UCL pour permettre d'effectuer des tests interactifs. Comme Socrative, il dispose d'une interface claire et intuitive qui permet une création rapide de tests ainsi qu'une utilisation très simple pour les étudiants. On peut observer cependant que certaines fonctionnalités sont manquantes parmi lesquelles : la possibilité d'introduire des fichiers audio ou vidéo dans les questions, les tests nominatifs, les feedbacks et l'affichage du nombre de personnes connectées au test. Il propose néanmoins la possibilité d'afficher un mur de question pendant les tests et permet l'utilisation de SMS pour transmettre les réponses. Les fonctionnalités de Wooclap n'ont pas été prises en compte pour l'élaboration du cahier des charges.

---

10. <https://www.wooclap.com/>

# Chapitre 3

## Cahier des charges

Ce chapitre décrira le cahier des charges établi pour la réalisation de notre site de tests interactifs. Nous commencerons par définir le contexte de notre problème. Nous énoncerons ensuite le résultat attendu. Nous continuerons par la description de l'ensemble des exigences fonctionnelles. Pour finir, nous déterminerons les exigences non-fonctionnelles.

### 3.1 Définition du problème

L'utilisation de tests interactifs pendant un cours permet à l'enseignant d'évaluer directement le niveau de compréhension des étudiants. L'enseignant peut donc adapter ses explications ou réexpliquer certains points de son cours qui poseraient des problèmes. Cela renforce l'implication des étudiants et donc éventuellement leur motivation.

Plusieurs logiciels existent déjà pour permettre la réalisation de tests interactifs. Nous avons précédemment présenté Socrative qui offre un environnement simple et convivial.

L'UCL utilise depuis 2015 le site Moodle comme plateforme d'apprentissage en ligne. Ce site apporte des outils complets (rapports, corrections, types de questions variés, ...) pour réaliser et corriger des tests classiques. Cependant, il ne propose pas de tests interactifs hormis sous la forme de plugin.

Deux plugins permettent déjà la mise en place de tests interactifs avec Moodle : Ipal et Active Quiz. Ils sont caractérisés par un manque d'ergonomie dans leur utilisation à cause de leurs interfaces graphiques. De plus, le fait qu'ils soient intégrés dans Moodle ne procure pas la facilité d'utilisation qu'on peut retrouver chez Socrative.

### 3.2 Définition des acteurs :

On retrouvera deux acteurs différents qui utiliseront notre site :

1. L'enseignant : personne qui fait passer un test interactif.
2. L'étudiant : personne qui répond au test interactif d'un enseignant.

### 3.3 Objectif

L'objectif de ce mémoire est la création d'un site simple et convivial permettant, à des enseignants, de réaliser des tests interactifs, pour des étudiants, à partir de tests créés via un site Moodle.

### 3.4 Contrainte

Dans un souci de faciliter la maintenance et le déploiement, il nous a été suggéré de développer le site en utilisant le langage de programmation PHP, comme celui utilisé pour Moodle.

### 3.5 Exigences fonctionnelles

Voici une description de l'ensemble des fonctionnalités dont notre site devrait disposer. Cet ensemble a été établi à partir des observations venant de la comparaison des logiciels effectuée dans le chapitre précédent.

#### 3.5.1 Fonctionnalités de l'enseignant :

##### **Connexion :**

L'enseignant entre son identifiant Moodle et son mot de passe dans les champs correspondants et clique sur le bouton "connexion". Le nom et le mot de passe sont vérifiés sur Moodle ainsi que le rôle de l'utilisateur. La liste de ses cours est récupérée depuis Moodle. Elle est ensuite affichée.

##### **Choisir un cours :**

L'enseignant sélectionne un cours parmi la liste de ses cours. Il a ensuite le choix entre un bouton "Sondage rapide" et un bouton "Test".

##### **Sélectionner un test :**

L'enseignant clique sur le bouton "Test" après avoir sélectionné un cours. La liste des tests de ce cours est récupérée depuis Moodle. Elle est affichée. L'enseignant sélectionne un test parmi la liste de tests disponibles pour le cours sélectionné. La description d'un test peut être affichée. Un bouton "Démarrer le test" est affiché.

##### **Démarrer un test :**

L'enseignant clique sur "Démarrer le test". Un message est affiché demandant s'il désire que le test soit anonyme. L'enseignant répond au message en cliquant sur le bouton "oui" ou sur le bouton "non". La première question s'affiche sur la page de l'enseignant et des étudiants. La barre de navigation s'affiche sur la page de l'enseignant. Elle est composée de :

- un bouton "précédent"

- un bouton "suivant"
- un bouton "stop"
- un bouton "Feedback"
- un bouton "Réponses"
- du nombre d'étudiants ayant répondu à la question sur le nombre d'étudiants participant au test.

#### **Passer à la question suivante :**

L'enseignant clique sur le bouton "suivant". La question suivante s'affiche sur la page de l'enseignant et des étudiants. Le bouton n'est pas utilisable pour la dernière question du test.

#### **Passer à la question précédente :**

L'enseignant clique sur le bouton "précédent". La question précédente s'affiche sur la page de l'enseignant et des étudiants. Le bouton n'est pas utilisable pour la première question du test.

#### **Afficher le feedback :**

L'enseignant clique sur le bouton "Feedback". Le feedback de la question s'affiche sur la page de l'enseignant. Le bouton n'est pas utilisable si aucun feedback n'est disponible.

#### **Afficher/Masquer la répartition des réponses et la bonne réponse :**

L'enseignant clique sur le bouton "Réponses". La réponse à la question et un graphe montrant la répartition des réponses des étudiants s'affichent sur la page de l'enseignant.

#### **Arrêter le test :**

L'enseignant clique sur le bouton "stop". Un message signalant que le test est terminé est affiché sur la page de l'étudiant. Un message est affiché sur la page de l'enseignant lui proposant de sauvegarder les résultats du test sur Moodle. Un bouton "Sauvegarder" est affiché.

#### **Sauvegarder les résultats d'un test :**

Lorsqu'un test est terminé, l'enseignant peut choisir de sauvegarder les résultats de ce test sur Moodle en cliquant sur le bouton "Sauvegarder".

#### **Créer/modifier une question/un test :**

Fonctionnalités disponibles via Moodle.

### **Effectuer un sondage rapide :**

L'enseignant clique sur le bouton "Tests" après avoir sélectionné un cours. Il doit ensuite choisir le type de sondage rapide parmi "vrai/faux", "choix multiples" et "réponses ouvertes" pour le démarrer. Le sondage rapide commence et les commandes propres au type sélectionné sont affichées chez les étudiants. Plusieurs éléments sont affichés chez l'enseignant :

- Le nombre d'étudiants ayant répondu à la question sur le nombre d'étudiants participant au sondage
- les boutons de choix de type pour effectuer un nouveau sondage
- les réponses des étudiants sous forme de graphe ou par étudiant en fonction du type de sondage.

### **3.5.2 Fonctionnalités de l'étudiant :**

#### **Connexion :**

L'étudiant entre le code du cours dans le champ correspondant et clique sur rejoindre le test. Si le test n'est pas anonyme, l'étudiant entre son identifiant Moodle et son mot de passe dans les champs correspondants. Si aucun test n'est en cours, un message invitant l'étudiant à patienter est affiché. Si un test est en cours, une question est affichée.

#### **Répondre à une question :**

L'étudiant effectue l'action correspondant au type de la question pour y répondre.

#### **Retrouver une tentative :**

L'étudiant aura accès à une review de son test sur Moodle si celui-ci n'était pas anonyme.

## **3.6 Exigences non-fonctionnelles**

### **Ergonomie :**

Socrative propose une excellente ergonomie que ce soit pour l'enseignant ou pour les étudiants. Pour justifier l'utilisation de notre site, en plus des fonctionnalités ajoutées, on doit aussi pouvoir compter sur une bonne intuitivité.

### **Design :**

Le design du site a aussi son importance pour que celui-ci soit agréable à utiliser. Il doit servir l'ergonomie tout en restant léger.

**Légèreté :**

L'utilisation des plugins pour Moodle dans le but de réaliser des tests interactifs implique le chargement d'éléments de Moodle qui ne sont pas nécessaires. Le site doit donc permettre d'avoir accès à Moodle sans passer directement par celui-ci afin de ne pas consommer des ressources inutilement.

# Chapitre 4

## Développement

Nous décrivons dans ce chapitre plusieurs aspects du développement de notre site InteractiveQuiz. Nous commencerons par justifier notre choix de framework et nous détaillerons celui que nous avons sélectionné. Nous présenterons ensuite l'architecture de notre site en décrivant ses principales parties. Nous finirons ce chapitre en expliquant les prérequis nécessaires pour le déploiement d'InteractiveQuiz.

### 4.1 Framework

L'utilisation d'un framework permet de démarrer sur une bonne base. Il fournit une structure pour le code et force donc à respecter certaines normes de développement et une méthodologie. Cela facilite la compréhension du code et la maintenance. En fonction des frameworks, cela permet aussi de ne pas devoir implémenter certaines fonctionnalités de base ainsi que des mécaniques plus techniques. On peut donc se concentrer uniquement sur le développement des fonctionnalités propres à notre site.

#### 4.1.1 Choix du framework

Plusieurs critères ont été retenus pour le choix du framework PHP que nous avons utilisé :

##### **Documentation :**

Une documentation claire et complète facilite la prise en main. De plus, elle permet de rendre la maintenance plus aisée grâce à une meilleure compréhension du framework. La communauté active autour du framework est aussi importante. Lors du développement, elle permet de trouver rapidement une réponse en cas de problème en plus d'ajouter un complément d'informations à la documentation ainsi qu'un autre point de vue. La majorité des principaux frameworks dispose d'une bonne documentation et d'une communauté bien développée.

## Complexité :

La complexité n'est pas toujours facilement perceptible. La plupart des framework mettent à disposition des tutoriels, plus ou moins complets, qui décrivent la mise en œuvre des fonctionnalités qu'ils proposent. A partir de là, il n'est pas trop compliqué de démarrer un nouveau projet. Cependant, cela demande un certain investissement en temps et ne permet pas toujours de pouvoir juger de la réelle complexité du framework.

## Performances :

Le nombre de requêtes par seconde, le pic de consommation de mémoire sur le serveur et le nombre de fichiers utilisés sont des données qui peuvent refléter les performances d'un framework. Nous avons collecté ces données pour les principaux frameworks php grâce à un script de benchmarking<sup>1</sup>. Les mesures ont été prises en effectuant des requêtes sur une page "Hello world". Cela va donc mesurer les coûts minimum de fonctionnement de ces frameworks.

Environnement du benchmark :

- Ubuntu 14.4
- Apache 2.4.12
- PHP 5.6.11

Résultats :

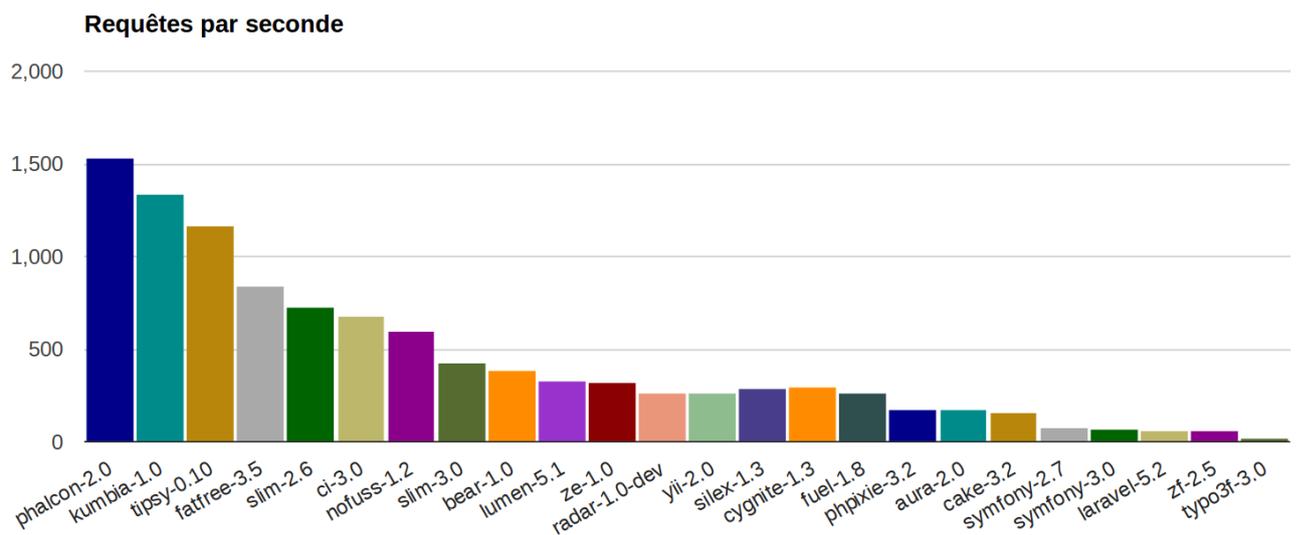


FIGURE 4.1 – Requêtes par seconde

1. <https://github.com/kenjis/php-framework-benchmark>

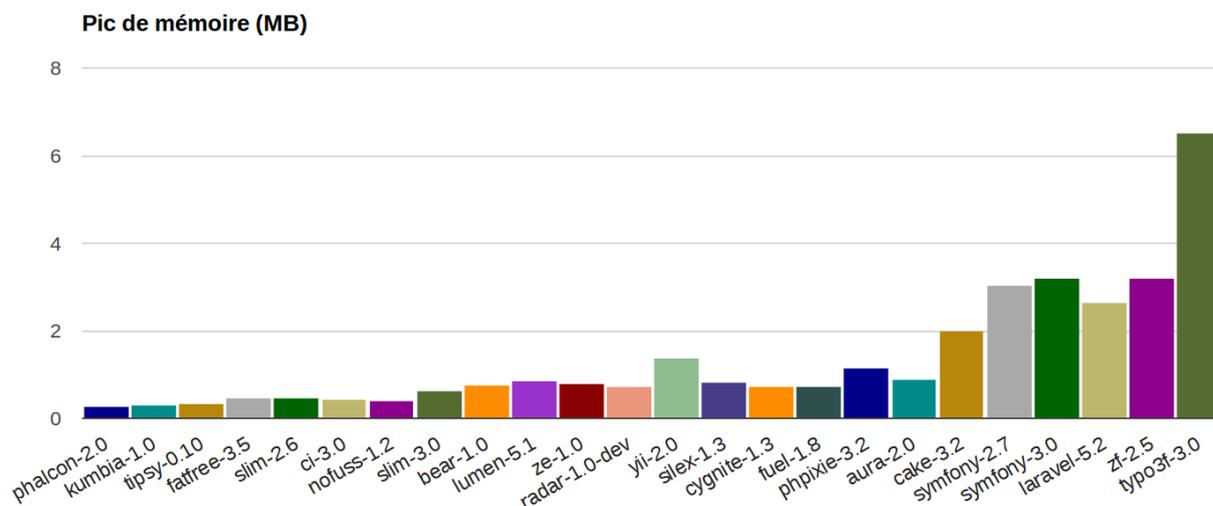


FIGURE 4.2 – Pic de mémoire (MB)

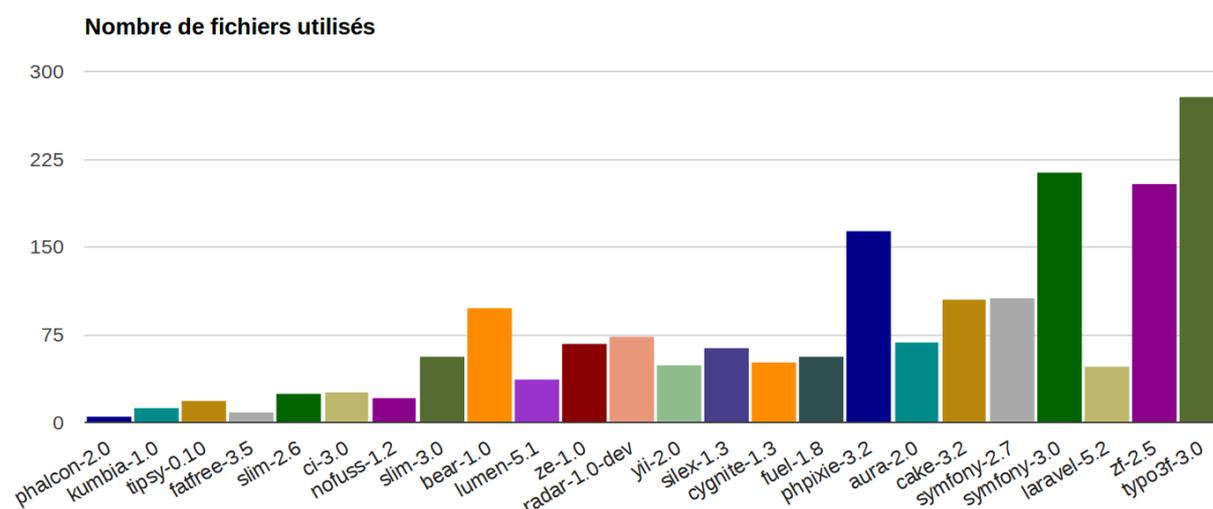


FIGURE 4.3 – Nombre de fichiers utilisés

Le framework phalcon [10] est ressorti premier du benchmark pour les trois données mesurées. Puisqu'il dispose d'une bonne documentation et qu'il ne semble pas particulièrement complexe, il fait un bon candidat dans le choix du framework PHP à utiliser pour notre site.

#### 4.1.2 Framework Phalcon

Phalcon est un framework PHP open source optimisé pour la performance [11]. Il a été implémenté comme une extension en C de PHP ce qui évite de devoir interpréter ses fichiers car ils sont déjà compilés. Il propose une structure MVC (Model-View-Controller) mais celle-ci n'est pas stricte. Il dispose des principales fonctionnalités recherchées dans un framework et utiles pour notre site. Par exemple :

## Fichier de configuration :

Les paramètres généraux sont stockés dans un fichier de configuration. Phalcon propose deux adaptateurs de fichiers différents pour permettre de lire le fichier de configuration de manière orientée objet. [12]. Le premier stocke les paramètres sous la forme de tableaux dans un fichier PHP. Le deuxième permet l'utilisation d'un fichier INI. Nous avons choisi la deuxième méthode car elle fournit un fichier de configuration plus clair. Pour récupérer, par exemple, le nom de la base de données stocké dans le fichier config.ini sous la forme :

```
[database]
dbname    = test_db
```

Nous devons exécuter le code suivant :

```
<?php
use Phalcon\Config\Adapter\Ini as ConfigIni;
$config = new ConfigIni("path/config.ini");
$name = $config->database->username;
```

## Gestion du routage :

Phalcon utilise un routage qui respecte le pattern suivant : "/controller/action/params". Par exemple, la route "/teachercoursechoice/selectCourse/1/C1" exécutera la fonction

```
selectCourseAction($courseId, $courseCode)
```

du contrôleur "TeachercoursechoiceController".

## Injection de dépendances :

Phalcon fournit un système d'injection de dépendances[6] qui permet d'avoir accès à des services à plusieurs endroits de l'application sans avoir à les recréer. L'injection de dépendances se fait dans le fichier "/app/config/service.php". Par exemple en inscrivant ce service de cette manière :

```
$di->set('moodleWebservices', function () {
    return new MoodleWebservices();
});
```

Nous pourrions avoir accès à l'instance de MoodleWebservices dans les contrôleurs de cette façon :

```
$this->moodleWebservices
```

## Plugins :

Phalcon permet l'utilisation d'un plugin "SecurityPlugin" qui va gérer l'accès aux différentes pages et fonctionnalités du site en fonction du rôle défini pour l'utilisateur. Un autre plugin utile

est le "NotFoundPlugin" qui permet d'afficher une page avec un message correspondant à l'erreur survenue. Ces plugins se trouvent dans le dossier "/app/plugins".

### 4.1.3 Design Framework

Nous utiliserons aussi le design framework Bootstrap [5]. Celui-ci met à disposition de nombreux outils écrits en CSS ou javascript qui permettent d'obtenir un design soigné et homogène. Il permet aussi d'avoir un responsive design qui va adapter l'affichage de la page en fonction de la taille de l'écran. Il rend donc l'utilisation du site sur appareil mobile plus aisée.

## 4.2 Architecture

Notre site est composé de trois parties. La première partie se rapporte aux services web qui sont utilisés afin de communiquer avec le site Moodle associé à notre site. La deuxième partie se compose du serveur websocket qui permet de répercuter les actions effectuées par l'enseignant sur la page des étudiants et vice-versa grâce à l'envoi de messages entre eux. Et enfin, la troisième partie comprend le site en lui-même qui propose l'interface utilisateur pour l'enseignant et les étudiants et qui permet d'interagir avec les deux premières parties citées.

### 4.2.1 Services web

La partie service web regroupe le code utilisé pour faire appel aux fonctions des services web mis à disposition par le site Moodle. Ces fonctions permettent donc d'interagir avec ce site. Nous les utiliserons pour effectuer différentes actions :

#### **Identification :**

L'utilisation des fonctions des services web nécessite un token afin d'identifier l'utilisateur. Ce token peut être obtenu en utilisant le script "/login/token.php" disponible parmi les fichiers du site Moodle associé. Ce script nécessite comme entrées le nom de l'utilisateur, son mot de passe et le nom du service pour lequel le token sera généré.

Ce script n'est pas disponible pour les administrateurs pour des raisons de sécurité. Il faudra donc passer par l'interface du site Moodle pour créer les tokens pour ces utilisateurs.

#### **Transmission de la liste des cours :**

Pour l'enseignant, nous avons besoin de pouvoir récupérer la liste des cours auxquels il est inscrit et pour lesquels il a le statut d'enseignant. Cette liste sera affichée à l'enseignant afin qu'il puisse sélectionner le cours pour lequel il voudra faire passer un test.

Pour l'étudiant, il nous faut aussi récupérer sa liste de cours. Lorsqu'il tentera de rejoindre un test, cela permettra de déterminer s'il est bien inscrit au cours auquel le test appartient.

Fonctions utilisées :

- `core_webservice_get_site_info` : permet de récupérer l'identifiant de l'utilisateur.
- `core_enrol_get_users_courses` : permet de récupérer la liste des cours de l'utilisateur.
- `core_user_get_course_user_profiles` : permet de vérifier le rôle de l'utilisateur (dans le cas de l'enseignant).

### **Transmission de la liste des tests :**

Une fois que l'enseignant a sélectionné un de ses cours, la liste des tests de ce cours est nécessaire. Elle permettra de lui afficher les différents tests disponibles.

Fonction utilisée :

- `mod_quiz_get_quizzes_by_courses` : permet de récupérer la liste des tests pour un cours de l'utilisateur.

### **Transmission d'un test :**

Lorsque la liste des tests de l'enseignant lui est affichée, il en sélectionne un. Il faut ensuite récupérer ce test pour pouvoir afficher ses questions. Pour ce faire, Moodle impose de démarrer une tentative de résolution du test en utilisant l'identifiant du test. Cela permettra de récupérer un identifiant de tentative nécessaire pour pouvoir obtenir les données du test.

Fonctions utilisées :

- `mod_quiz_start_attempt` : permet de récupérer l'identifiant d'une tentative.
- `mod_quiz_get_attempt_data` : permet de récupérer le contenu d'une page du quiz.

Une fois que l'enseignant a transmis l'identifiant du test sélectionné et le numéro de la page actuellement affichée, les étudiants peuvent démarrer une tentative en utilisant la première fonction. Ils peuvent ensuite récupérer les données du test à afficher pour eux à l'aide de la deuxième fonction en utilisant le numéro de page venant de l'enseignant .

### **Intégrer les résultats d'un test :**

A chaque fois qu'un étudiant répond à une question, sa réponse est envoyée au site Moodle afin qu'elle puisse être sauvegardée.

Fonction utilisée :

- `mod_quiz_process_attempt` : permet de sauvegarder des réponses de la tentative .

### Clôturer un test :

Lorsque l'enseignant décide de terminer le test, les tentatives de tous les étudiants sont clôturées.

Fonctions utilisées :

- `mod_quiz_process_attempt` : (avec le paramètre `finishattempt = 1`) permet de clôturer la tentative.

### 4.2.2 Serveur Websocket

Lorsqu'un enseignant commence un test, un serveur websocket est démarré pour ce test. L'implémentation de ce serveur est une adaptation du code de PHP Push WebSocket<sup>2</sup>. Ce serveur stockera l'identifiant du test, le numéro de la page affichée chez l'enseignant et le nombre de réponses des étudiants pour cette page. Le rôle de ce serveur est de pouvoir établir un canal de communication direct entre la page de l'enseignant et les pages des étudiants.

Le port utilisé par le serveur websocket est défini lors du démarrage du premier test d'un cours. Le numéro de port est donc associé au (code du) cours. Les ports disponibles sont à définir lors de la configuration du site 4.3.2. Le nombre de port disponibles ne doit pas être inférieur au nombre de cours différents susceptibles d'utiliser le site.

Plusieurs sortes de messages sont échangées entre le serveur websocket, l'enseignant et les étudiants :

#### Connexion :

Lors de la connexion de l'enseignant, celui-ci envoie au serveur le message :

```
{"teacher_connected", quizId, page}
```

Le serveur va alors diffuser l'identifiant du test et le numéro de page à tous les étudiants déjà connectés. Il enverra ensuite à l'enseignant le nombre d'étudiants connectés.

Lorsqu'un étudiant se connecte, il va envoyer au serveur le message :

```
{"student_connected"}
```

Si l'enseignant est déjà connecté, le serveur transmettra le nombre d'étudiants connectés à celui-ci. Le serveur enverra aussi dans ce cas l'identifiant du test et le numéro de page à l'étudiant qui vient de se connecter.

#### Nombre d'étudiants :

Le nombre d'étudiants connectés au test est transmis à l'enseignant par le serveur via le message :

```
{"nb_students", nb_students}
```

2. <https://github.com/srchea/PHP-Push-WebSocket>

Ce message lui est envoyé lors de sa connexion et lorsqu'un nouvel étudiant se connecte au serveur.

Le nombre d'étudiant est aussi envoyé à l'enseignant lorsqu'un étudiant se déconnecte grâce au message :

```
{"student_disconnected", student_id, nb_students_connected}
```

L'enseignant reçoit en plus par ce message l'identifiant de l'étudiant qui vient de se déconnecter afin de ne plus tenir compte des éventuelles réponses de l'étudiant pour la page affichée.

### Identifiant du test :

Un étudiant peut connaître l'identifiant du test lorsque le serveur lui envoie le message :

```
{"quiz", quizId}
```

Ce message lui est envoyé lorsqu'il se connecte et que l'enseignant est déjà connecté au serveur ou lors de la connexion de l'enseignant si celui-ci n'était pas encore connecté.

### Numéro de page :

L'étudiant est informé du numéro de la page du test actuellement affichée chez l'enseignant lorsqu'il reçoit le message :

```
{"page", page}
```

Ce message lui est envoyé lorsqu'il se connecte et que l'enseignant est déjà connecté, lorsque l'enseignant se connecte et que l'étudiant est déjà connecté ou lorsque l'enseignant change de page.

### Changement de page de l'enseignant :

Lorsque l'enseignant change de page, il envoie le message :

```
{"new_page", page}
```

Le serveur diffusera ensuite ce nouveau numéro de page à tous les étudiants

### Réponse d'un étudiant :

Un étudiant va envoyer sa réponse au serveur à l'aide du message :

```
{"response", response}
```

Le serveur enverra alors le message à l'enseignant :

```
{"response", student_id, n_answers, response}
```

Ce message permet à l'enseignant de connaître la réponse de l'étudiant ainsi que le nombre d'étudiants ayant déjà répondu à la question.

## Déconnexion :

Quand l'enseignant se déconnecte du serveur, il envoie le message :

```
{"quit"}
```

Le serveur va ensuite déconnecter tous les étudiants avant de s'arrêter.

Lorsque c'est un étudiant qui se déconnecte, celui-ci envoie aussi au serveur le message :

```
{"quit"}
```

Le serveur prévient l'enseignant qu'un étudiant s'est déconnecté et que le nombre d'étudiants connectés a changé.

### 4.2.3 InteractiveQuiz

Grâce aux services web, nous pouvons récupérer les tests du site Moodle associé et lui communiquer les réponses des étudiants. Le serveur websocket permet d'établir une communication entre la page de l'enseignant et celles des étudiants. Il nous reste donc à décrire comment l'enseignant et les étudiants peuvent utiliser tout ça afin de réaliser des tests interactifs.

Nous présenterons le fonctionnement de notre site à travers un scénario qui montre les interactions qu'il peut y avoir entre les acteurs et les différentes parties.

Le scénario débute après la sélection d'un test par l'enseignant. Le test va alors démarrer. Ensuite, un étudiant se connecte au test et répond à la question affichée. Après, l'enseignant passe à la question suivante. Et enfin, l'enseignant va terminer le test avant que l'étudiant n'ait pu répondre à la seconde question affichée.

Pour plus de lisibilité, nous avons divisé le scénario en quatre parties. Chaque partie sera illustrée par un diagramme de séquence.

## 1. Démarrage du test :

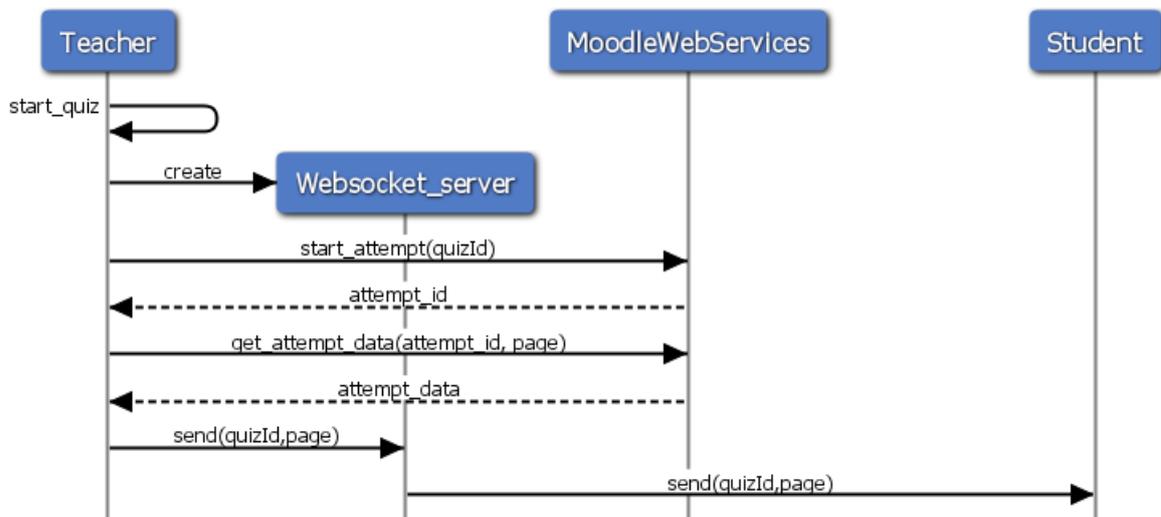


FIGURE 4.4 – Démarrage d'un test et réponse d'un étudiant

Le scénario commence juste après que l'enseignant ait sélectionné le test qu'il veut faire passer. L'identifiant du test (*quizId*) est donc déjà connu.

Tout d'abord, un serveur websocket est créé. Ensuite, l'enseignant va faire appel à la fonction *start\_attempt(quizId)* du MoodleWebServices pour démarrer une tentative pour récupérer l'*attempt\_id*, l'identifiant de la tentative. Cet identifiant permettra à l'enseignant d'utiliser la fonction *get\_attempt\_data(attempt\_id, page)* du MoodleWebServices afin d'obtenir les données de la première page du test. Le paramètre *page* correspond au numéro de page du test. Comme on débute le test, celui-ci vaudra zéro.

Après avoir récupéré ces données, la première question du test pourra être affichée sur la page de l'enseignant. L'identifiant du test et le numéro de la page affichée seront envoyés vers le serveur websocket afin qu'ils puissent être diffusés à tous les étudiants qui y seraient déjà connectés.

## 2. Connexion et réponse d'un étudiant :

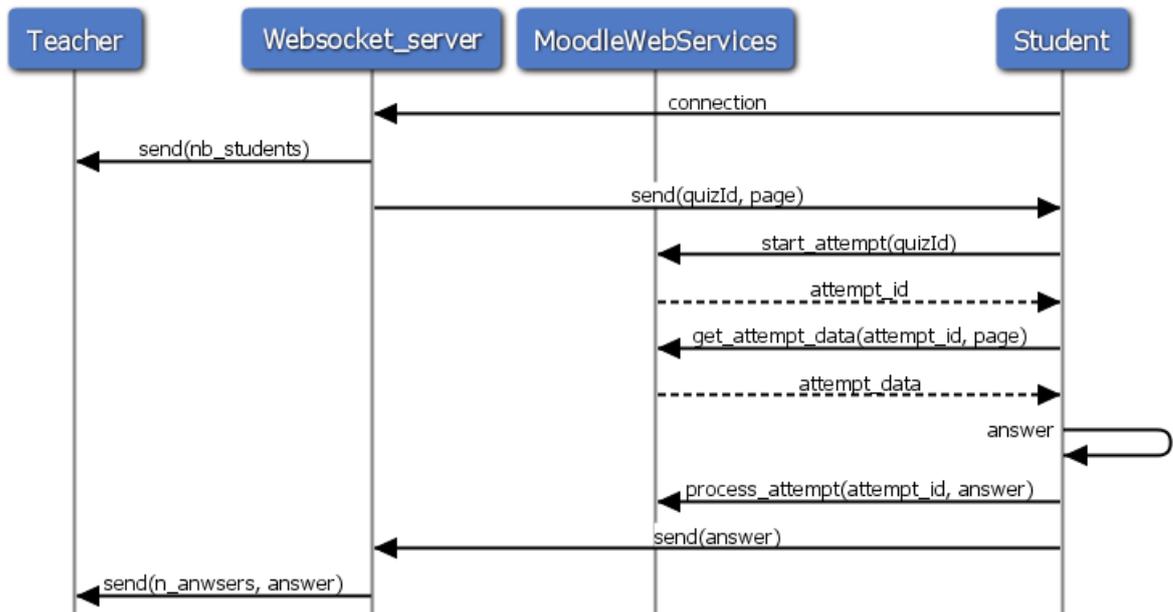


FIGURE 4.5 – Connexion et réponse d'un étudiant

Après avoir introduit le code du cours, l'étudiant se connecte au serveur websocket de ce cours afin de pouvoir répondre au test. Lorsque sa connexion est effectuée, le nouveau nombre d'étudiants connectés est envoyé à l'enseignant. Le serveur websocket va ensuite transmettre à l'étudiant l'identifiant du test en cours ainsi que numéro de la page actuellement affichée chez l'enseignant.

L'étudiant pourra démarrer sa tentative en utilisant la fonction *start\_attempt(quizId)* et affichera la question, comme pour l'enseignant, grâce aux données obtenues avec la fonction *get\_attempt\_data(attempt\_id,page)*.

Lorsque l'étudiant aura répondu à la question et décidera de l'envoyer, la fonction *process\_attempt(attempt\_id,answer)* du MoodleWebServices sera utilisée pour sauvegarder sa réponse sur le site Moodle. Cette réponse sera aussi transmise au serveur websocket pour qu'elle puisse être connue de l'enseignant. Le nombre de réponses total des étudiants est aussi transmis à l'enseignant par la même occasion. L'enseignant pourra alors afficher les réponses des étudiants ainsi que leur répartition sous forme de graphes.

### 3. Question suivante :

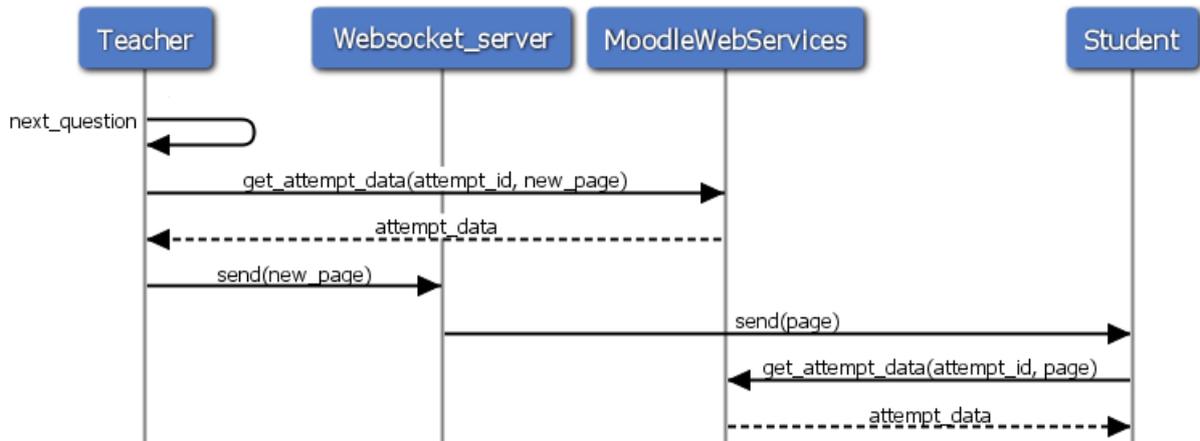


FIGURE 4.6 – Question suivante

Lors du test, quand l'enseignant décide passer à la question suivante, il obtient les données de la page suivante en utilisant toujours la fonction *get\_attempt\_data(attempt\_id, new\_page)*. Il enverra ensuite le nouveau numéro de page au serveur websocket qui le diffusera aux étudiants. Ceux-ci pourront alors afficher la nouvelle question comme vu précédemment.

### 4. Fin du test :

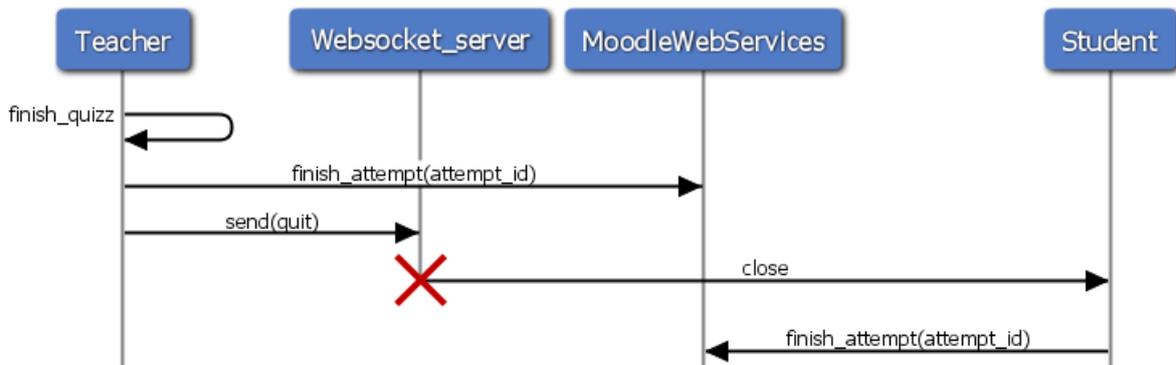


FIGURE 4.7 – Fin du test

L'enseignant peut à tout moment terminer le test. Il clôturera sa propre tentative grâce à la fonction *finish\_attempt(attempt\_id)* du MoodleWebServices. Il enverra ensuite le message "quit" au serveur websocket qui déconnectera tous les étudiants du serveur websocket avant de déconnecter l'enseignant et de s'arrêter. Lorsqu'un étudiant est déconnecté du serveur websocket, il clôturera en utilisant aussi la fonction *finish\_attempt(attempt\_id)*.

## 4.3 Déploiement

Nous décrirons dans cette section les différentes manipulations requises pour permettre le déploiement de notre site InteractiQuiz.

### 4.3.1 Moodle

Tout d'abord il faut disposer d'un site Moodle version 3.1 ou plus. Les services web utilisés par InteractiveQuiz ont été, en grande partie, développés pour l'application Moodle Mobile. Celle-ci ne gérait pas les tests dans les versions précédant la version 3.1.

Ensuite il faut paramétrer son site Moodle pour qu'il puisse accepter l'utilisation des services web. Cela peut se faire en trois étapes : [17]

1. Activer les services web(Site administration/Advanced features)
2. Activer l'utilisation du protocole REST (Site administration/Plugins/Web services/Manage protocols)
3. Autoriser la création de token pour les utilisateurs authentifiés (Site administration/Users/Permissions/Define roles/Authenticated user)

Si le site Moodle permet l'utilisation de Moodle Mobile, les deux premières étapes ne sont pas nécessaires car ces paramétrages auront déjà été effectués.

Enfin, il faut intégrer le service InteractiveQuiz à son site Moodle. Ce service est inclus dans un plugin local. Pour l'installer, un simple drag&drop du dossier du plugin suffit via l'onglet "Site administration/Plugins/Install plugins".

### 4.3.2 InteractiveQuiz

Le site InteractiveQuiz nécessite l'installation du framework Phalcon 2.0.10 sur le serveur. Celui-ci devra disposer de PHP 5.4 et de GCC (Linux/Solaris) ou de Xcode (Mac). [7]

Le fichier de configuration "/app/config/config.ini" doit aussi être complété. Plusieurs champs sont à remplir :

#### [database]

- host = hôte pour la base de données.
- username = nom d'utilisateur pour la base de données.
- password = mot de passe pour l'utilisateur de la base de données.
- dbname = nom de la base de données.

#### [websocket]

- startPort = le plus petit numéro de port disponible pour le serveur websocket

- `endPort` = le plus grand numéro de port disponible pour le serveur websocket
- `restrictedPorts` = la liste des ports interdits qui se trouveraient entre `startPort` et `endPort`.  
Le nombre de numéros de ports disponibles ne doit pas être inférieur au nombre de cours différents pour lesquels le site serait susceptible d'être utilisé.
- `address` = l'adresse utilisée pour le serveur websocket

**[moodle]**

- `domainname` = l'adresse du site Moodle associé

Pour la base de données, il faut en créer une à partir des informations données ci-dessus et ensuite exécuter le script `courses.sql` du dossier `app/db` afin de la construire.

# Chapitre 5

## Expérimentations et analyses

Nous commencerons par présenter dans ce chapitre les résultats obtenus à la suite de l'étape de développement. Nous détaillerons ensuite les divergences entre le résultat et le cahier des charges. Pour finir, nous effectuerons une comparaison des performances entre InteractiveQuiz et le plugin ActiveQuiz pour Moodle.

### 5.1 Résultat

#### 5.1.1 Fonctionnalité commune :

##### Connexion :

Un enseignant ou un étudiant peut se connecter à InteractiveQuiz en utilisant son identifiant et son mot de passe du site Moodle associé. Lors de cette connexion, un rôle est attribué à l'utilisateur et celui-ci aura accès seulement aux fonctionnalités qui lui sont propres.

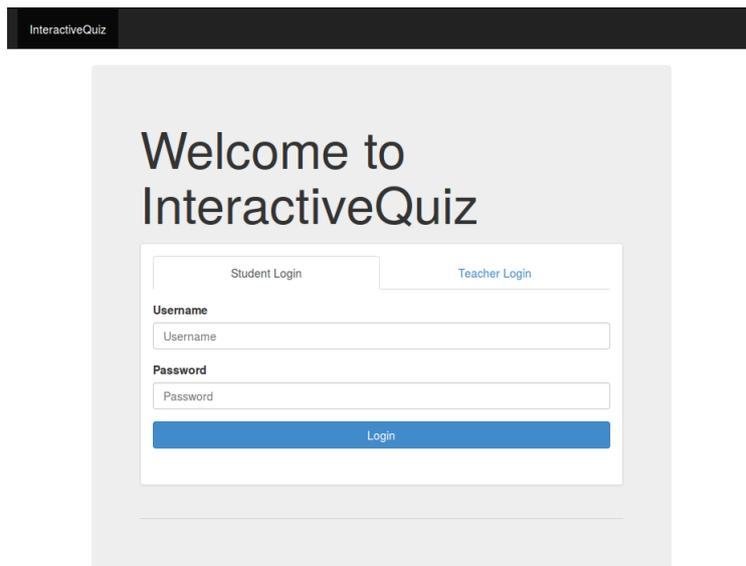


FIGURE 5.1 – Connexion

## 5.1.2 Fonctionnalités de l'enseignant :

### Choisir un cours :

Si l'enseignant a introduit un mot de passe correct, l'ensemble des cours de Moodle pour lesquels il a le rôle d'enseignant est affiché.

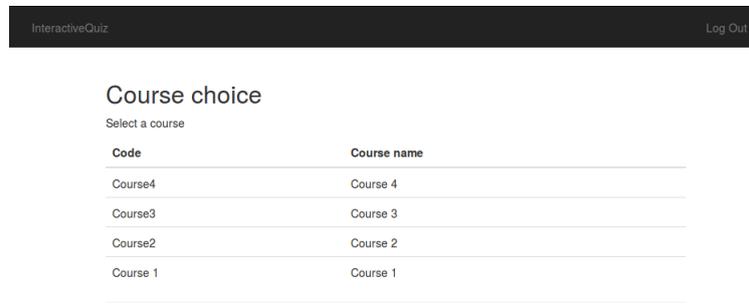


FIGURE 5.2 – Choix du cours

### Sélectionner un test :

Lorsque l'enseignant sélectionne un de ses cours, une liste contenant les tests de ce cours est affichée.

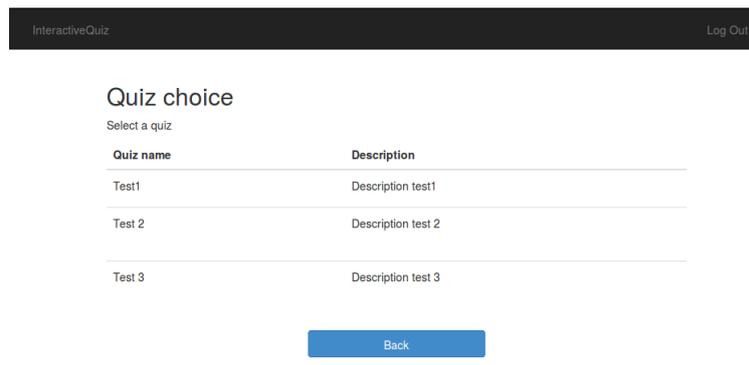


FIGURE 5.3 – Choix du test

### Démarrer un test :

L'enseignant peut alors choisir le test à démarrer. Il pourra naviguer à travers les questions du test en utilisant les boutons contenant les flèches. Il pourra à tout moment mettre un terme au test en utilisant le bouton avec le carré. L'ensemble des types de questions de Moodle 2.9 est supporté hormis celles devant disposer de la fonctionnalité drag&drop.

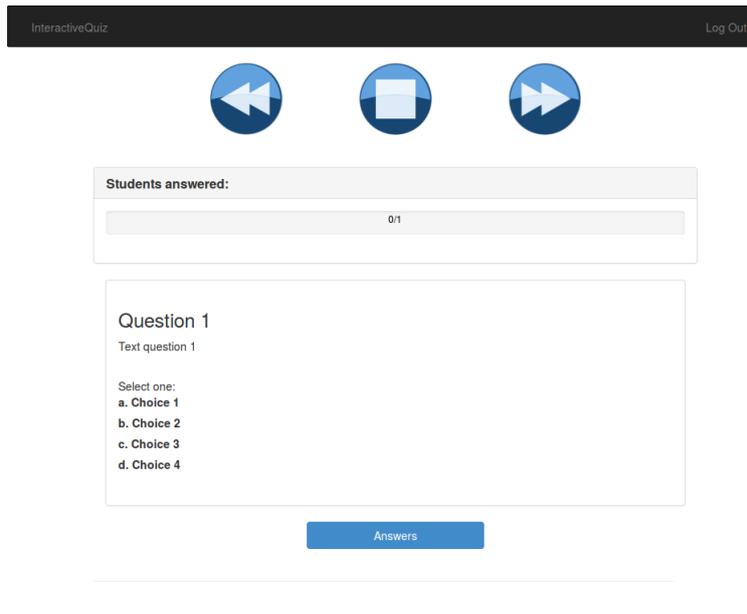


FIGURE 5.4 – Test

### Afficher/Masquer la répartition des réponses :

L'enseignant peut afficher et masquer la répartition de réponses des étudiants à l'aide du bouton "Answers". Celle-ci évoluera en direct, au fur et à mesure que les étudiants envoient leurs réponses.

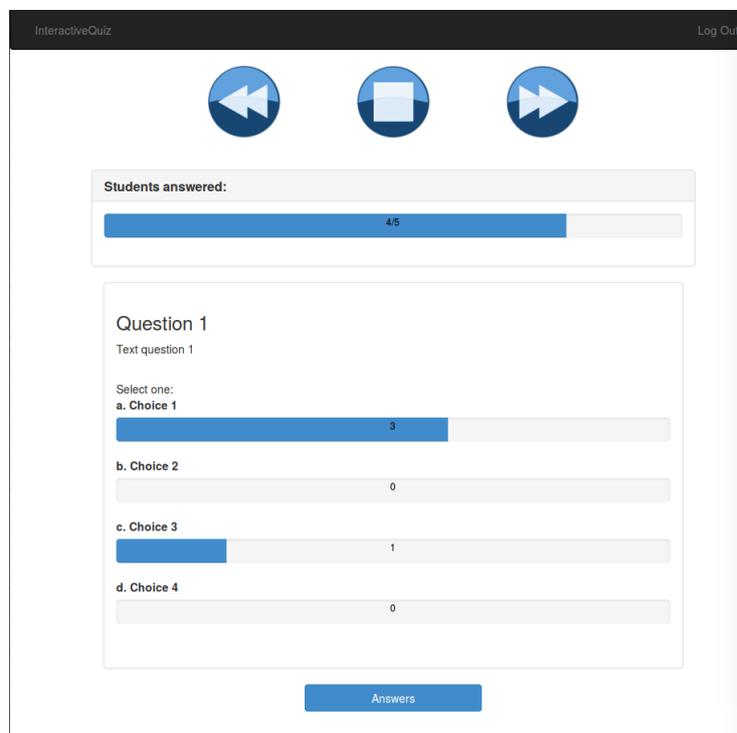


FIGURE 5.5 – Réponses de étudiants

### 5.1.3 Fonctionnalités de l'étudiant :

#### Rejoindre un test

Après s'être connecté, l'étudiant peut rejoindre un test en introduisant le code du cours auquel le test appartient. Il n'aura accès au test que si il est bien inscrit à ce cours sur le site Moodle associé.



FIGURE 5.6 – Rejoindre un test

#### Répondre à une question :

L'étudiant voit sur son écran la même question que celle affichée par l'enseignant. Il peut alors transmettre sa réponse en utilisant le bouton "Send". Elle sera envoyée à l'enseignant et sauvegardée sur le site Moodle associé. Lorsque l'enseignant change de question, la question de l'étudiant changera également.

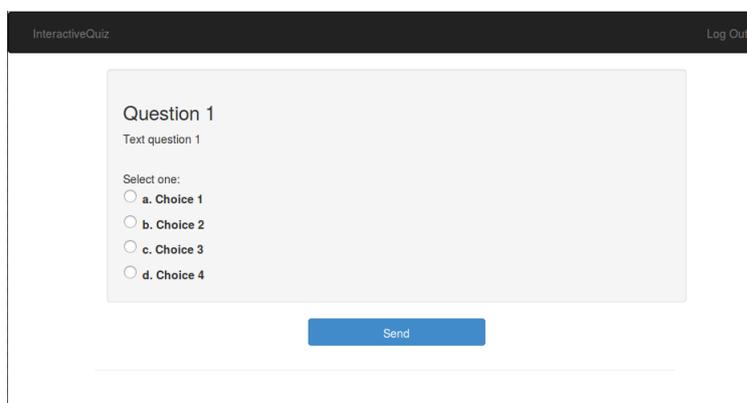


FIGURE 5.7 – Question de l'étudiant

#### Retrouver une tentative :

L'ensemble des réponses de l'étudiant pourra être retrouvé sur le site Moodle associé.

## Attente de l'enseignant

Lorsqu'un étudiant tente de se connecter à un test qui n'a pas encore été démarré par l'enseignant, celui-ci est mis en attente et sera connecté automatiquement au test lorsqu'il sera démarré.

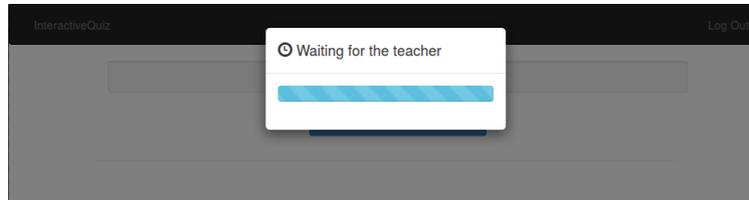


FIGURE 5.8 – Attente

## 5.2 Limitations

L'essentiel des fonctionnalités a été mis en place afin de pouvoir réaliser des tests interactifs à partir de tests de Moodle. Cependant, toutes les fonctionnalités du cahier des charges n'ont pas pu être réalisées par manque de temps.

Les services web de Moodle concernant les tests ont été développés dans le courant de l'année et ont été disponibles en partie en version alpha à partir du mois de janvier. C'est seulement fin mai qu'ils ont été intégrés à Moodle 3.1 en version stable. Au vu de la masse de travail que cela représente, le développement de ces services web a été une aubaine. Cependant, le revers de la médaille est que la maigre documentation et quelques bugs ont fortement ralenti ou postposé le développement du site InteractiveQuiz.

Voici les divergences par rapport au cahier des charges initial :

### Sondage rapide :

Cette fonctionnalité aurait demandé une interface supplémentaire pour pouvoir générer différentes sortes de questions. Il aurait aussi fallu adapter la transmission d'information entre l'enseignant et les étudiants pour signaler ce type de question et les transmettre ou éventuellement les faire générer chez les étudiants. Vu qu'il y a moyen de générer ces genres de question via un test Moodle, cette fonctionnalité ne nous a pas semblée indispensable.

### Proposition de sauvegarde des résultats :

Lorsqu'un étudiant répond à une question et transmet sa réponse à l'enseignant, celle-ci est automatiquement sauvegardée sur le site Moodle associé. Permettre à l'enseignant de choisir si les résultats du test effectué doivent être sauvegardés ou non posait un problème. Il aurait fallu stocker l'ensemble des réponses des étudiants jusqu'à la fin du test et ensuite envoyer toutes les réponses d'un seul coup. Il nous semblait plus judicieux de sauvegarder automatiquement au fur et mesure les réponses des étudiants quitte à supprimer simplement les tentatives depuis

le site Moodle un fois que le test serait terminé. Une autre solution aurait été de proposer à l'enseignant d'autoriser la sauvegarde des réponses au démarrage du test. Dans cette situation, en cas de réponse négative de l'enseignant, une tentative sans réponse serait de toute manière présente sur le site Moodle pour les étudiants puisque ceux-ci doivent en démarrer une pour avoir accès aux questions du test.

#### **Tests anonymes :**

En partant de l'idée que le code du cours permet de rejoindre un test, les tests anonymes auraient été accessibles à n'importe quels utilisateurs sans restriction. L'enseignant aurait dû être capable de transmettre les questions et pas seulement un numéro de page. Les identifiants des questions étant propres à chaque utilisateur et tentative, cette solution ne pouvait pas être envisagée si on voulait pouvoir sauvegarder les tentatives des étudiants sur le site Moodle. La possibilité d'effectuer cette sauvegarde nous a semblé plus prioritaire que de permettre à n'importe quel utilisateur de pouvoir rejoindre un test.

#### **Feedback/réponses :**

Seuls les tests avec un feedback immédiat permettent d'avoir accès aux feedbacks et aux réponses aux questions avant la fin d'un test. En utilisant les services web disponibles, aucune solution n'a été trouvée pour avoir accès à ces informations.

#### **Paramétrage des tests :**

Certaines restrictions sont à émettre à propos du paramétrage des tests sur Moodle pour qu'ils puissent être utilisés dans de bonnes conditions avec InteractiveQuiz.

- L'option "Shuffle within questions" dans la section "Question behaviour" doit être désactivée pour garantir que l'ordre des questions des étudiants est bien le même que pour l'enseignant.
- Le test doit être disponible pour les étudiants sur le site Moodle lorsque l'on veut l'utiliser sur InteractiveQuiz. Les services web ne confèrent pas de droit d'accès supplémentaires aux fonctionnalités et aux ressources pour les étudiants que sur le site Moodle.

### **5.3 Comparaison des performances**

InteractiveQuiz se démarque de Socrative car il permet l'utilisation des tests réalisés sur un site Moodle. Il propose aussi une fonctionnalité importante par rapport au plugin Ipal qui est la sauvegarde des tentatives des étudiants. Nous allons maintenant voir en quoi notre site se distingue du plugin ActiveQuiz.

Une des exigences non-fonctionnelles énoncée dans le cahier des charges était la légèreté. Nous allons donc comparer les performances entre ActiveQuiz et InteractiveQuiz. Pour ce faire, nous exécuterons deux scénarios :

1. Un enseignant démarre un test interactif.
2. Un étudiant rejoint un test interactif.

Lors de ces scénarios nous collecterons des données à l'aide de l'extension Firebug<sup>1</sup> de Firefox afin de pouvoir effectuer une comparaison. Ces données sont :

1. Le nombre de requêtes effectuées.
2. Le poids des réponses reçues.
3. Le temps total pris pour charger toutes les requêtes.

### Résultats :

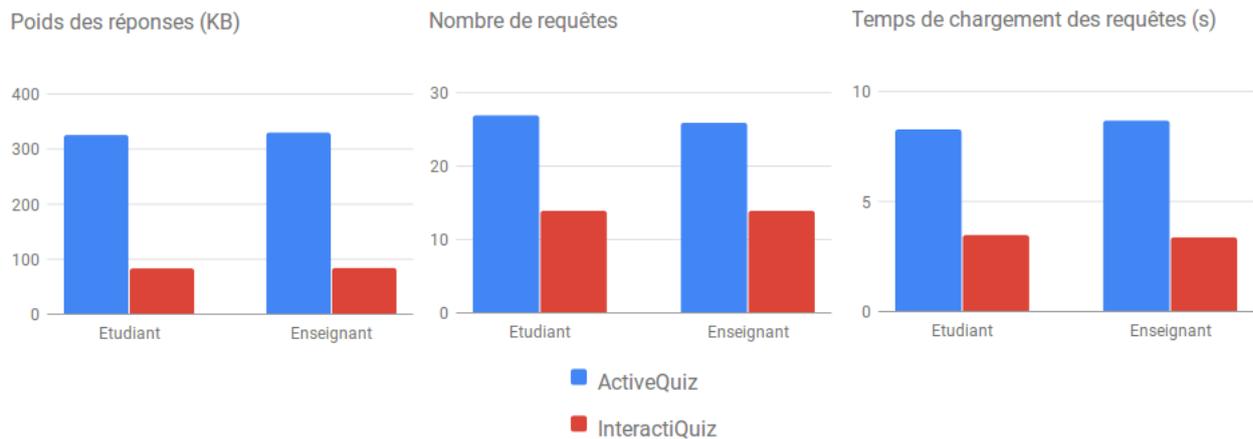


FIGURE 5.9 – Comparaison des performances entre ActiveQuiz et InteractiveQuiz

Vous trouverez les résultats détaillés dans l'annexeB. On observe que le nombre de requêtes pour InteractiveQuiz est presque deux fois moins élevé pour les deux scénarios. Cela implique un temps de chargement des requêtes ainsi que le poids total des requêtes sont bien inférieurs. La performance d'InteractiveQuiz est donc un atout par rapport à ActiveQuiz.

En plus, ActiveQuiz envoie toutes les 3 secondes plusieurs requêtes au serveur (trois pour l'enseignant et deux pour l'étudiant). Ces requêtes permettent de signaler les actions entre ces deux types d'utilisateur. Grâce à cela, l'enseignant peut obtenir les réponses des étudiants et ceux-ci peuvent être avertis d'une action (par exemple un passage à la question suivante) de l'enseignant.

L'utilisation d'un serveur websocket pour InteractiveQuiz permet d'éviter d'effectuer de nombreuses requêtes inutiles.

1. <http://getfirebug.com/>

## Chapitre 6

# Conclusion

Tout au long de ce mémoire, nous avons présenté différentes recherches, étapes et analyses qui ont menées à la réalisation d'une extension de Moodle pour permettre d'utiliser ses tests de manière interactive.

Puisqu'on a pu voir que les tests interactifs ont une influence certaine pour améliorer la qualité d'un cours, il nous a semblé opportun de les associer aux fonctionnalités complètes que met à disposition Moodle pour l'élaboration de tests.

Avant l'élaboration du cahier des charges, nous avons dû analyser le logiciel Moodle afin de comprendre de quelle manière nous pourrions lui ajouter des fonctionnalités. La possibilité d'utiliser des services web nous donnait l'opportunité de développer une extension externe et non sous la forme de plugin. La comparaison de logiciels permettant de réaliser des tests interactifs nous a permis de définir quelles seraient les fonctionnalités nécessaires pour notre extension.

Il nous est apparu que nous devions pouvoir proposer une alternative aux plugins déjà existants qui permettent la réalisation de tests interactifs via Moodle. Un design différent pouvait être réalisé sans utiliser des ressources inutiles.

Notre extension a donc pris la forme d'une application web qui communique avec le site Moodle qui lui est associé grâce à des services web. Nous avons choisi d'utiliser un serveur websocket afin d'établir un canal de communication direct entre la page de l'enseignant et les pages des étudiants .

L'application qui résulte de l'étape de développement rencontre la plupart des exigences essentielles établies dans le cahier des charges. On notera tout de même l'absence de la fonctionnalité permettant l'affichage du feedback et de la bonne réponse.

Les fonctionnalités non développées du cahier des charges, comme par exemple les sondages rapides ou les tests anonymes, constituent autant de possibilités d'améliorations. L'ajout d'un mur de commentaires, comme le propose Socrative ou Wooclap, pourrait aussi bien renforcer l'interactivité de notre application.

# Bibliographie

- [1] Web service glossary, 2004. <https://www.w3.org/TR/ws-gloss/>.
- [2] Ucl - les plateformes de cours en ligne, 2015. <http://www.uclouvain.be/326571.html>.
- [3] Activities – moodledocs, 2016. <https://docs.moodle.org/31/en/Activities>.
- [4] App mobile – moodledocs, 2016. [https://docs.moodle.org/3x/fr/App\\_mobile](https://docs.moodle.org/3x/fr/App_mobile).
- [5] Bootstrap, 2016. <http://getbootstrap.com/>.
- [6] Class phalcon di – phalcon 2.0.10 documentation, 2016. [https://docs.phalconphp.com/en/latest/api/Phalcon\\_Di.html](https://docs.phalconphp.com/en/latest/api/Phalcon_Di.html).
- [7] Installation – phalcon 2.0.10 documentation, 2016. <https://docs.phalconphp.com/en/latest/reference/install.html>.
- [8] Moodle plugins directory, 2016. <https://moodle.org/plugins/index.php>.
- [9] Moodle statistics, 2016. <https://moodle.net/stats/>.
- [10] Phalcon – high performance php framework, 2016. <https://phalconphp.com/en/>.
- [11] Phalcon 2.0.10 documentation, 2016. <https://docs.phalconphp.com/en/latest/index.html>.
- [12] Reading configurations – phalcon 2.0.10 documentation, 2016. <https://docs.phalconphp.com/en/latest/reference/config.html>.
- [13] Ressources – moodledocs, 2016. <https://docs.moodle.org/3x/fr/Ressources>.
- [14] Socrative help center | faqs, 2016. <http://help.socrative.com/article/faqs/>.
- [15] Test – moodledocs, 2016. <https://docs.moodle.org/3x/fr/Test>.
- [16] Types de question – moodledocs, 2016. [https://docs.moodle.org/3x/fr/Types\\_de\\_questions](https://docs.moodle.org/3x/fr/Types_de_questions).
- [17] Using web services – moodledocs, 2016. [https://docs.moodle.org/31/en/Using\\_web\\_services](https://docs.moodle.org/31/en/Using_web_services).

- [18] Web service api – moodledocs, 2016. [https://docs.moodle.org/dev/Web\\_services\\_API](https://docs.moodle.org/dev/Web_services_API).
- [19] Web service api functions – moodledocs, 2016. [https://docs.moodle.org/dev/Web\\_service\\_API\\_functions](https://docs.moodle.org/dev/Web_service_API_functions).
- [20] À propos de moodle, 2016. [https://docs.moodle.org/3x/fr/A\\_propos\\_de\\_Moodle](https://docs.moodle.org/3x/fr/A_propos_de_Moodle).
- [21] G. A DeBourgh. Use of classroom "clickers" to promote acquisition of advanced reasoning skills, 2008. 8 (2), 76-87. doi :10.1016/j.nepr.2007.02.002.
- [22] John Hoopes. Moodle plugins directory : Active quiz, 2015. [https://moodle.org/plugins/mod\\_activequiz](https://moodle.org/plugins/mod_activequiz).
- [23] William (Bill) Junkin. Moodle plugins directory : ipal, 2015. [https://moodle.org/plugins/mod\\_ipal](https://moodle.org/plugins/mod_ipal).
- [24] D Laurillard. Rethinking teaching for the knowledge society, 2002. Educause, 37, 16–27.
- [25] P Li. Creating and evaluating a new clicker methodology, 2007. The Ohio State University.
- [26] V Simpson. Using electronic voting systems in lectures, 2002. UCL Internal report (p. 33). UCL. Retrouvé de <http://www.ucl.ac.uk/learningtechnology/examples/ElectronicVotingSystems.pdf>.
- [27] M Thienpont. Recherche sur les boîtiers de votes électroniques : Théories, contenus et méthodes, 2010. Mémoire de master en Communication Scientifique et Technique non publié ,École normale Supérieure de Cachan, Cachan.

## Annexe A

# Tableaux comparatifs de Socrative, Ipal et ActiveQuiz

Voici des tableaux qui reprennent l'ensemble des fonctionnalités qu'offrent Socrative, Ipal et Active quiz pour l'enseignant et pour les étudiants. Les fonctionnalités communes sont colorées en **bleu**, les atouts en **vert** et les manques importants en **rouge**.

### A.1 Enseignant

Fonctionnalités	Socrative	Ipal	Active quiz
Créer un test	X	X	X
Nommer un test	X	X	X
Donner une description au test		X	X
Créer une question	X	X	X
Ajouter une image à une question	X	X	X
Ajouter une vidéo/son à une question		X	X
Choix du type de réponse : Choix multiples	X	X	X
Choix du type de réponse : Vrai/Faux	X	X	X
Choix du type de réponse : Réponses ouvertes	X	X	X
Choix du type de réponse : Autres types de questions par défaut de Moodle			X
Autoriser les commentaires pour les étudiants	X		
Définir une durée maximum pour une question			X
Définir un temps d'attente entre les questions			X
Ajouter une question d'une banque de questions de Moodle		X	X
Ajouter une question du site compadre.org		X	
Modifier l'ordre des questions	X	X	X
Rendre le test anonyme ou nominatif	X	X	X

Fonctionnalités	Socrative	Ipal	Active quiz
Rendre l'ordre des questions aléatoire	X		
Rendre l'ordre des réponses aléatoire	X		
Copier une question	X	X	X
Démarrer un test	X	X	X
Possibilité d'affichage de la répartition de réponses choisies (graphe)	X	X	X
Possibilité d'affichage des réponses des étudiants et leurs noms	X	X	X
Visualisation du nombre d'étudiants ayant répondu à la question sur le nombre d'étudiants connecté	X	X	X
Passer à la question suivante	X	X	X
Afficher un feedback	X		
Sélectionner la question à afficher		X	X
Arrêter un test	X	X	X
Terminer une question	X	X	X
Possibilité de partager les tests	X	X	X
Faire un sondage rapide : Choix multiples	X		
Faire un sondage rapide : Vrai/faux	X		
Faire un sondage rapide : Réponse courte	X		
Faire un billet de sortie	X		
Faire un test en groupe	X	X	X
Choisir le nombre de groupe	X	X	X
Affichage de la progression des groupes	X		
Obtenir un rapport	X		X
Exporter un rapport	X		X

## A.2 Etudiants

Fonctionnalités	Socrative	Ipal	Active quiz
Rejoindre un test via un numéro de salle	X		
Rejoindre un test via Moodle		X	X
Répondre à une question	X	X	X
Passer à la question suivante	X		
Naviguer dans les questions	X		
Soumettre le questionnaire	X		X
Possibilité d'utiliser un clicker (Mobile Apps)	X	X	

<b>Fonctionnalités</b>	<b>Socrative</b>	<b>Ipal</b>	<b>Active quiz</b>
Revoir sa tentative			X
Obtenir sa note			X

## Annexe B

# Mesures des performances d'ActiveQuiz et d'InterActiveQuiz

Vous trouverez ci-dessus l'ensemble des mesures qui ont été collectées afin de réaliser le graphique comparatif des performances entre ActiveQuiz et InteractiveQuiz. Ces mesures ont été collectées en utilisant l'extension Firebug<sup>1</sup> de Firefox lors de l'exécution de toutes les actions nécessaires pour démarrer un test pour l'enseignant et rejoindre un test pour un étudiant.

### B.1 ActiveQuiz

#### Teacher

Action	Nombre de requêtes	Poids des réponses (KB)	Temps <sup>2</sup>
Login	14	291,4	1,1
Choix du cours	4	9,2	2,8
Choix du quiz	2	10,1	1,9
Nom de session	2	11,2	1,4
Démarrage du quiz	4	9,1	1,5
<b>Total</b>	<b>26</b>	<b>331</b>	<b>8,7</b>

#### Student

Action	Nombre de requêtes	Poids des réponses (KB)	Temps <sup>2</sup>
Login	14	291,4	1,1
Choix du cours	4	9,2	2,8
Choix du quiz	2	8,8	1,7

---

1. <http://getfirebug.com/>

2. Temps total pris pour charger toutes les requêtes (s)

Action	Nombre de requêtes	Poids des réponses (KB)	Temps <sup>2</sup>
Choix de la session	2	8,1	1,3
Rejoindre le quiz	5	9,4	1,4
<b>Total</b>	27	326,9	8,3

## B.2 InteractiveQuiz

### Teacher

Action	Nombre de requêtes	Poids des réponses (KB)	Temps <sup>2</sup>
Login	10	80,8	0,5
Choix du cours	1	0,645	1,1
Choix du quiz	1	0,721	0,5
Démarrage du quiz	2	3	1,3
<b>Total</b>	14	85,166	3,4

### Student

Action	Nombre de requêtes	Poids des réponses (KB)	Temps <sup>2</sup>
Login	10	80,8	0,5
Code du cours	1	0,693	1,1
Rejoindre le quiz	3	2,9	1,9
<b>Total</b>	14	84,393	3,5

---

2. Temps total pris pour charger toutes les requêtes (s)

