Table des matières

Remerciements 2

Dédicace 3

Liste des figures 3

Liste des Tableaux 4

I- Introduction générale 5

I-1 Contexte 5

I-2 Apprentissage par les scénarios 6

a- Définition 6

b- Etude d'outils pour la création de scénarios d'apprentissage 7

II- Analyse des besoins 10

II-1 Méthodologie 10

II-2 Etude des besoins 13

a- Introduction 13

b- Approche 13

c- Exigences fonctionnelles 14

d- Exigences non fonctionnelles 17

II-3 Contraintes et choix technologique 18

a- Contraintes 18

b- Choix technologique 19

III- Architecture logicielle 25

III-1 Architecture globale 25

III-2 Architecture détaillée 29

a- studypress-sbl 29

b- sbl-creator 33

c- sbl-player 34
IV- Réalisation 38
IV-1 Démarche suivie 38
a- Système de gestion de versions 38
b- Environnement de développement 39
c- Tests 40
IV-2 Résultats 41
a- Back office 41
b- Front office 45
V- Conclusion et perspective 47
V-1 Conclusion 47
V-2 Amélioration future 47
a- Fonctionnalités 47
b- Processus de développement 49
Bibliographie 51
Glossaire 53

Liste des figures

Figure I.1 : Exemple de déroulement d'un scénario 7
Figure I.2 : Interface de création de scénario dans Twine 8
Figure II.1 : Cycle de vie d'une itération 11
Figure II.2 : Exemple d'un tableau Kanban dans Trello 12
Figure II.3 : Diagramme de cas d'utilisation pour le système StudyPress SBL 14
Figure II.4 : Diagramme de séquence pour la création des scénarios 16
Figure II.5 : Architecture d'une application Herbert 20
Figure II.6 : Anatomie d'un module dans AngularJS 22
Figure III.1 : Architecture globale de l'extension StudyPress 26
Figure III.2 : Modèle de données 27
Figure III.3 : Architecture héxagonal du projet studypress-sbl 30
Figure III.4 : Architecture de studypress-sbl sous forme de couches 31
Figure III.5 : Diagramme de classe utilisé dans les hexagones 32
Figure III.6 : Architecture utilisée dans le projet sbl-creator 34
Figure III.7 : Récupération de l'URL du scénario 35
Figure III.8 : Diapositive représentant une scène 35
Figure III.9 : Diapositives représentant le dialogue dans une scène 36
Figure III.10 : Diapositives après imbrication 36
Figure III.11 : Représentation d'un défi dans la présentation 36
Figure IV.1 : Schéma d'utilisation de Git 38
Figure IV.2 : Création de l'environnement de développement 39
Figure IV.3 : Processus de développement TDD 40
Figure IV.4 : Liste des scénarios 42
Figure IV.5 : Création d'un nouveau scénario 42
Figure IV.6 : Edition d'un scénario 43
Figure IV.7 : Page d'édition d'une scène 44
Figure IV.8 : Scénario vu par l'apprenant 45
Figure IV.9 : Branchement vu par l'apprenant 46
Figure V.1 : Visualisation d'un scénario sous forme d'un graphe 48
Figure V.2 : Processus Lean Startup 49

Liste des Tableaux

Tableau I.1 : Tableau comparatif entre les outils de création de scénarios..... 9
Tableau III.1 : Dictionnaire de données..... 27
Tableau III.2 : API utilisée pour l'accès aux resources du serveur 29

I- Introduction générale

I-1 Contexte

Dans un monde qui ne cesse d'évoluer, les entreprises doivent s'adapter pour rester compétitives. La formation des employés est un moyen pour y remédier. Cependant, les évolutions sont rapides, et les formations classiques (c'est à dire des formations dispensées en classe par un formateur, structurée par des objectifs d'apprentissages, une durée, un contenu et un planning) restent insuffisantes pour suivre. Une des méthodes au quelle les entreprises s'intéressent ces dernières années est le e-learning.

Aujourd'hui, le e-learning est devenu un marché plus important que jamais. Avec un investissement record des entreprises de 6,54 milliards de dollars, une hausse spectaculaire comparée aux années précédentes, 2,42 milliard de dollars en 2014 et 1,64 milliards de dollars en 2013 [1].

Le e-learning est une forme d'apprentissage s'effectuant sur des supports multimédia [2]. Ce terme est généralement utilisé pour définir l'utilisation du réseau et des TIC (Technologie de l'Information et de la Communication) pour dispenser une formation.

Ce travail rentre dans le cadre de l'évolution de l'outil *StudyPress*. Il consiste à ajouter la possibilité de créer des leçons construites autour de scénarios. Un scénario représente souvent une situation difficile. Les apprenants doivent prendre des décisions et c'est selon leurs décisions que le scénario évolue. Ce mode d'apprentissage permet de développer des aptitudes en matière de résolution de problèmes, utile notamment dans les domaines critiques et où le travail consiste à prendre des décisions.

Le projet *StudyPress* consiste à créer une plateforme d'apprentissage accessible au grand public. L'idée est de faciliter la création et le déploiement de contenus e-learning tout en ayant une large base d'utilisateurs. Pour ce faire, le développement d'une extension WordPress a été choisi comme solution. WordPress est un système de gestion de contenu, souvent utilisé comme moteur de blog. Il est utilisé par plus de 25% des sites web et représente près de 60% de part de marché dans les systèmes de gestion de contenu, ce qui en fait le système de gestion de contenu le plus utilisé sur le web [3].

La première version créée en 2013, l'extension, connue sous le nom *StudyPress* et développée par BENSMAINE Yasser et BOUACHA Oussama, permettait la création de cours, des présentations et de quiz interactifs. En 2014, BELHABIB Abdelkader et MATAHRI Abdelmadjid ont repris le travail et ont intégré un système de recommandations. L'année suivante, BENDELLA Meryem et SAIDI Salim ont réécrit l'extension en se basant sur une nouvelle architecture tout en y ajoutant de nombreuses fonctionnalités, dont la gestion des droits d'accès des formateurs, le payement en ligne des cours, l'intégration aux réseaux sociaux, etc. Depuis 2015, l'extension réalise une moyenne de 70 téléchargements par semaine et compte en total plus de 9100 téléchargements [4].

I-2 Apprentissage par les scénarios

a- Définition

L'apprentissage basé sur les scénarios, ou scenario-based learning (SBL) en anglais, est une approche d'apprentissage actif qui s'inscrit dans l'apprentissage cognitif. L'approche utilise les scénarios interactifs de telle sorte que les apprenants décident du déroulement du scénario qui généralement est basé sur des problèmes complexes. Dans ce procédé, les apprenants doivent appliquer leur connaissance, pensée critique et leurs talents pour résoudre les problèmes dans un contexte réaliste et sûr. Les scénarios SBL sont souvent non linéaires, les apprenants peuvent être défiés, et c'est selon leur réponse que le déroulement du scénario peut changer [5]. En d'autres termes, le choix de l'apprenant provoque une nouvelle situation qui fait suite à la précédente [6].

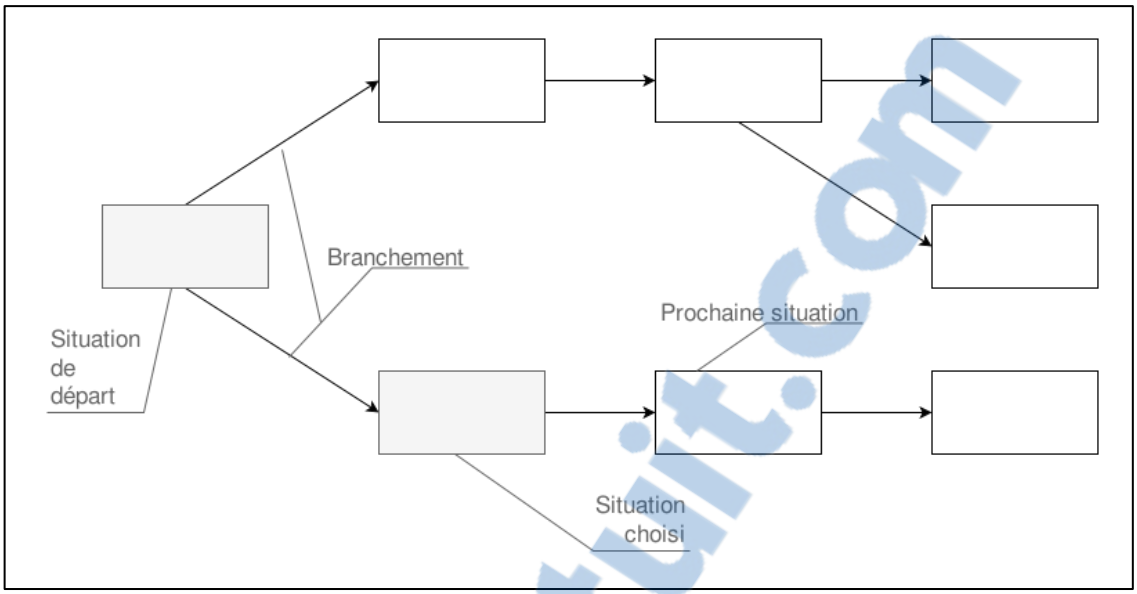

Figure I.1 : Exemple de déroulement d'un scénario

L'apprentissage basé sur les scénarios a pour but d'accélérer l'expertise de l'apprenant qui joue le rôle d'un acteur répondant à un travail réaliste [7]. Il permet de développer des compétences cognitives dans un domaine spécifique. Il améliore la perception, la mémoire, la compréhension, les états émotionnels, la prise de décision, et le raisonnement de l'apprenant.

b- Etude d'outils pour la création de scénarios d'apprentissage

Introduction

Sur le marché, il existe une multitude d'outil e-learning, mais peu sont spécialisés dans la création de leçons basées sur les scénarios. Dans cette étude comparative, deux logiciels E-learning des plus populaires: *Adobe Captivate* et *Articulate Storyline*, ainsi qu'un logiciel spécialisé dans la création de scénarios : *Twine*.

Outils

Twine

Twine[1] est un outil pour la création de documents hypertexte. Il permet notamment de créer des scénarios et de les publier sous format HTML. Il est possible d'enrichir le scénario avec du CSS, des variables, une logique conditionnelle, des images, et du

[1] https://twinery.org/

Rapport-Gratuit.com

JavaScript. L'interface de l'outil permet de visualiser les branchements sous forme de graphe. Le branchement se fait à travers des liens hypertexte.

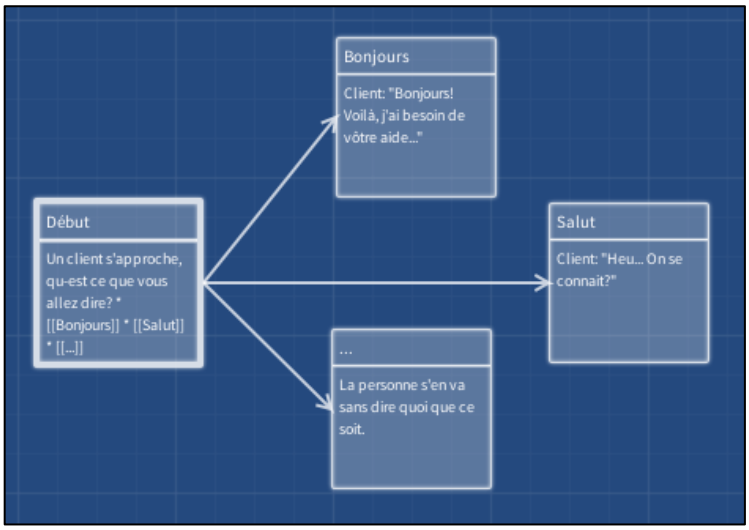

Figure I.2 : Interface de création de scénario dans Twine

Chaque nœud dans le graphe, nommé passage, représente une situation dans le scénario. Un passage contient du texte suivant une syntaxe lui permettant la mise en page. Après exportation du scénario, le passage apparaît sous forme d'une page web contenant le texte transformé. Ainsi, le choix de l'apprenant se fait en ouvrant un lien contenu dans le texte ce qui redirige l'apprenant vers une autre page web, et donc un autre passage.

Adobe Captivate

Adobe Captivate[2] est un logiciel de création de contenus e-learning comme des présentations, des quiz, des démonstrations, mais aussi des scénarios. Cependant, le logiciel n'inclut pas d'outil spécialisé dans la création de scénarios. Cette dernière se fait à travers la création d'une présentation contenant des diapositives pour le dialogue ou la narration, et des quiz dont le comportement est altéré pour le branchement. Cependant, le fait de modifier le comportement de chaque quiz rend l'utilisation du logiciel difficile et peu pratique pour la création de scénarios. Adobe Captivate peut exporter le contenu créé sous différents formats de fichier, notamment SWF (Flash) et HTML.

[2] http://www.adobe.com/fr/products/captivate.html

Articulate Storyline

Storyline[3] est un logiciel créé par Articulate. Il est permet de créer des présentations interactives, des vidéographies et des simulations. Les présentations peuvent incorporer du texte, de l'audio et des vidéos. Comme Adobe Caprivate, Articulate Storyline ne dispose pas d'outil spécialisé pour la création de scénarios. Les scénarios sont créés à partir de présentations et de quiz.

Discussion

L'apprentissage par scénarios est une pratique peu répondu. Les logiciels e-learning n'ont généralement pas d'outil spécialisé pour cette approche d'apprentissage. Il est à noter que ces logiciels ne visent pas le même publique et ont différents objectifs. Adobe Captivate et Articulate Storyline sont des logiciels pour un usage général. Ils ne se focalisent pas que sur la création de contenus basés sur scénario contrairement à Twine.

	Twine	**Adobe Captivate**	**Articulate Storyline**
Support SBL	Native	En altérant le comportement des présentations	En altéré le comportement des présentations
Personnalisation	Difficile	Facile	Facile
Ressources pré-disponibles (images, sons, …)	Non	Oui	Oui
Format	HTML	HTML, SWF	HTML, SWF, Articulate Mobile Player
Licence	Opensource (MIT)	Propriétaire	Propriétaire
Prix	Gratuit	1 099 $	1 398 $

Tableau I.1 : Tableau comparatif entre les outils de création de scénarios

[3] https://fr.articulate.com/products/storyline-why.php

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

II- Analyse des besoins

II-1 Méthodologie

Afin d'accomplir les objectifs visés dans ce mémoire, une méthode de gestion de développement de projet a été instauré. La méthode s'inspire des méthodes agiles et *Scrum*. Elle permet d'avoir des objectifs précis et motivants. Elle offre aussi une visibilité, adaptation et inspection des objectifs.

Comme pour *Scrum*, la méthode appliquée utilise une approche itérative permettant de découper le développement en plusieurs itérations appelées *Sprint*. Chaque Sprint est utilisé pour réaliser un objectif. Un Sprint est constitué notamment d'une planification (Sprint Planning), des activités de développement, des mêlées quotidiennes (*Daily Scrum*) et de la rétrospective du Sprint (*Sprint Rétrospective*) [8].

Au début de chaque Sprint, l'équipe choisit un objectif. Cet objectif est définit par un ensemble de tâches, aussi appelé *Backlog*. Il est créé lors de la réunion de planification du Sprint. S'ensuit alors une mêlée. Contrairement à Scrum, les mêlées ne sont pas quotidiennes et se font tous les trois jours. Une mêlée est un événement au cours duquel l'équipe se réunit, synchronise ses activités et crée un plan pour les prochaines 72 heures. C'est en inspectant le travail effectué depuis la dernière mêlée que l'équipe décrit les prochaines fonctionnalités à réaliser. Dans ce projet, les réunions sont faites par visioconférence grâce au logiciel Skype. A la fin de chaque Sprint, une revue et rétrospective de Sprint est tenue sous forme de réunion. Elle a pour but d'inspecter le travail fait, et d'améliorer et optimiser le déroulement des prochains Sprints.

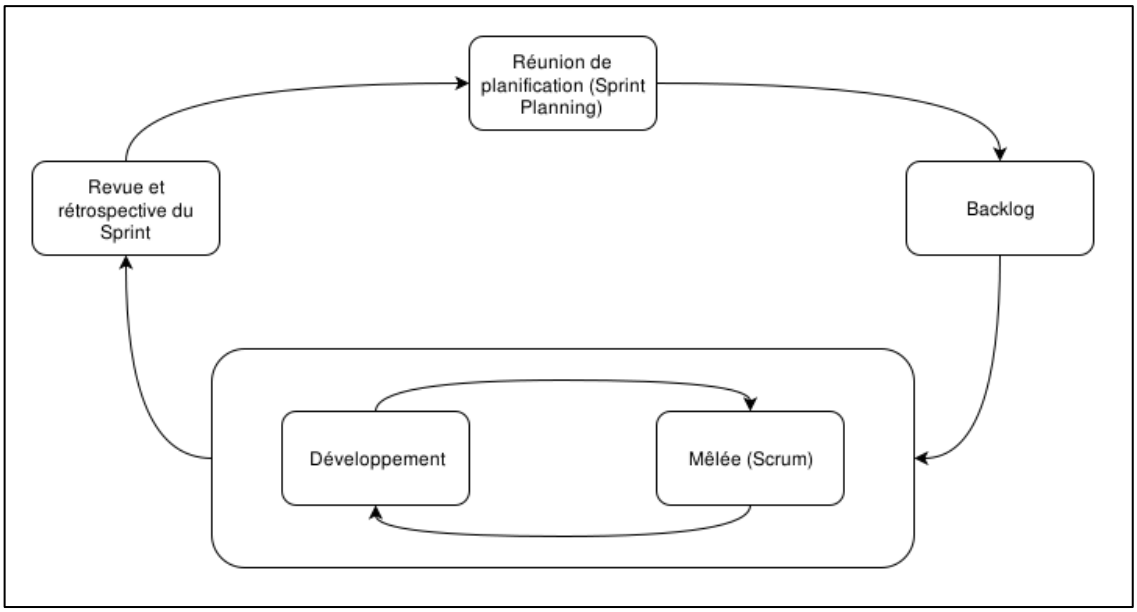

Figure II.1 : Cycle de vie d'une itération

Pour la gestion, suivi et planification des taches, Trello[4] est un outil en ligne permettant notamment de gérer le Backlog. Similaire à un tableau *Kanban*, l'interface de l'outil permet de visualiser les tâches sous formes de post-it rangées dans des colonnes. Les tâches peuvent être déplacées d'une colonne à l'autre et peuvent aussi être réorganisées. Dans ce projet, il est question de quatre colonnes :

- **A faire** : colonne qui regroupe les prochaines tâches à réaliser. Chaque tâche créée doit être placée dans cette colonne.
- **En cours** : les tâches situées dans cette colonne sont en développement ou en cours de réalisation. Ces tâches sont déplacées de la colonne *« A faire »* vers la colonne *« En cours »*.
- **En test** : Après réalisation d'une tâche, cette dernière est déplacée vers la colonne *« En test »* où l'équipe teste et vérifie l'accomplissement de la tâche. Si la tâche est incomplète, l'équipe la déplace vers la colonne en cours. Sinon, si la tâche est bien accomplie, elle sera déplacée vers la colonne *« Fait »*.
- **Fait** : Cette colonne regroupe les tâches réalisées et vérifiées. Elle permet de lister ce qui a été fait durant le Sprint. Elle est utile pour l'inspection, notamment lors de la réunion pour la revue et rétrospective du Sprint.

[4] https://trello.com/

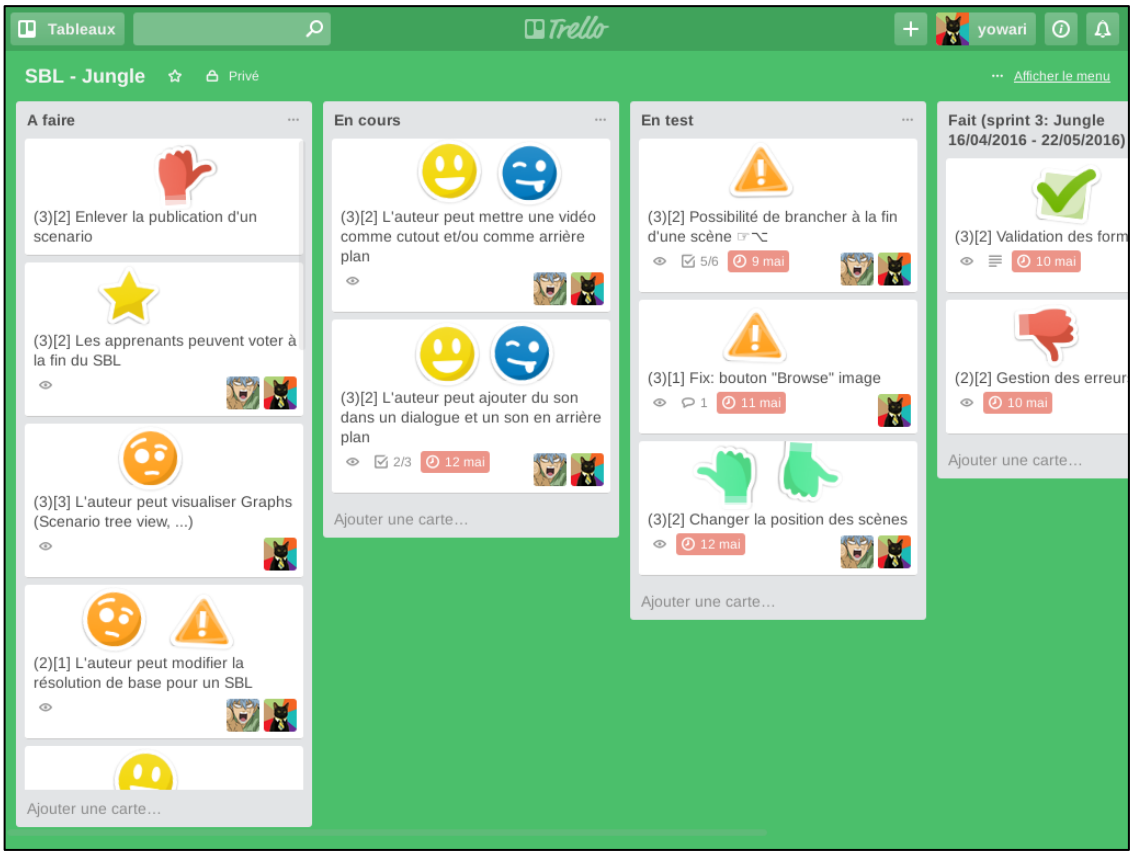

Figure II.2 : Exemple d'un tableau Kanban dans Trello

Une phrase désignant une tâche ou une fonctionnalité à réaliser est généralement préfixée par des indices d'importance et de difficulté. L'indice d'importance, noté entre parenthèses, est compris entre 1 et 3 (1 : peu important, 2 : important, 3 : très important). Quant à l'indice de difficulté, il est noté entre crochets et est aussi compris entre 1 et 3 (1 : facile, 2 : moyen, 3 : difficile). Ces indices permettent de prioriser les taches pour ainsi avoir une meilleur organisation.

Durant ce projet, trois Sprints ont eu lieu :

- **Sprint 1 (du 26/01/2016 au 08/02/2016) – nom de code Green Hill -** : L'objet de ce Sprint est de faire un état de l'art des outils utilisés pour l'apprentissage basé sur scénario. Ainsi que l'étude et la maintenance du code source de l'extension StudyPress (l'extension développée par BENDELLA Meryem et SAIDI Salim). Aussi, durant ce sprint, il a été question de choisir la méthodologie à adopter et configurer les outils de développement, de tests, etc.

- **Sprint 2 (du 08/02/2016 au 16/04/2016) – nom de code Bridge -** : L'objectif du Sprint est de créer une première version de l'application. L'idée est de créer les composants de base de l'application.
- **Sprint 3 (du 16/04/2016 au 09/06/2016) – nom de code Jungle -** : Ce Sprint a pour objectif d'ajouter des fonctionnalités à l'application comme la gestion des branchements.

II-2 Etude des besoins

a- Introduction

Comme la méthode utilisée pour la gestion du projet est une méthode agile, le recueille des exigences s'est fait à fur et à mesure. Ainsi, l'objectif de ce chapitre est de résumer les fonctionnalités de la solution proposée pour la réalisation de cours basé sur un scénario SBL, ainsi que les contraintes sur lesquels elle doit fonctionner.

Les outils étudiés pour l'état de l'art ne se spécialisent pas dans la création de scénarios ou ne vise pas le même public que celui du projet *StudyPress*. De ce fait, la solution proposée est une solution originale pour la création de scénarios.

b- Approche

Grâce aux articles parus sur le web, les objectifs de l'apprentissage basé sur un scénario peuvent être résumés en trois points :

- Exposer aux apprenants les informations sur la situation sur laquelle se déroule le scenario ;
- Interactivité avec les scenarios: Les apprenants peuvent avoir un impact sur le déroulement du scenario ;
- Réalisme dans le scenario de tel sorte que l'apprenant soit immergé.

A travers ces objectifs, la solution proposée décrit le scénario de tel sorte que :

- Le contexte du scenario est décrit à travers un dialogue entre des personnages ;
- Un scenario se compose d'une suite de scènes ;
- Chaque scène comprend un dialogue ;

- A la fin d'une scène, l'apprenant peut être défié avec la possibilité de choisir sa réponse à partir d'une liste de réponses possibles ;
- Le choix de la réponse dans un défi a un impact sur le déroulement du scénario.

c- Exigences fonctionnelles

L'étude des besoins a permis d'extraire les fonctionnalités de la solution suivant le diagramme de cas d'utilisation:

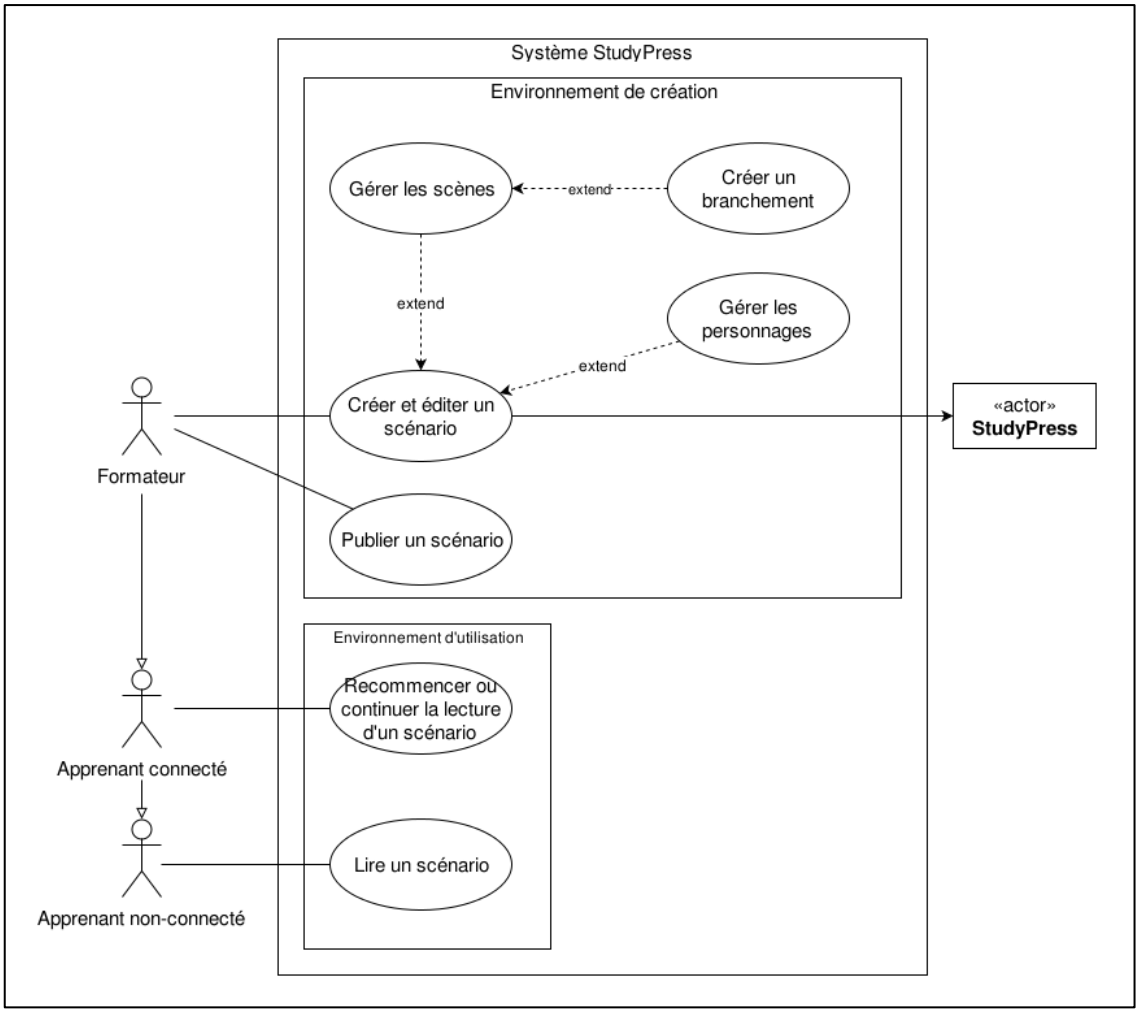

Figure II.3 : Diagramme de cas d'utilisation pour le système StudyPress SBL

Le système à réaliser est appelé *StudyPress SBL*. Ce système interagit avec quatre acteurs. A savoir que le formateur, l'apprenant connecté et l'apprenant non-connecté accèdent au système à travers le web et sont identifiés par le système d'authentification de WordPress. Chacun de ces acteurs a un privilège et certains, donc, accède à plus de fonctionnalités que d'autres :

- **Apprenant non-connecté** : personne accédant au système sans être identifié dans WordPress.
- **Apprenant connecté** : personne accédant au système, authentifiée dans WordPress et ayant le privilège d'un simple utilisateur.
- **Formateur** : personne accédant au système, authentifiée dans WordPress et ayant le privilège de rédacteur.
- **StudyPress** : cet acteur représente l'extension *WordPress StudyPress* réalisée par BENDELLA Meryem et SAIDI Salim. Ce système gère notamment les cours qui sont définis comme des conteneurs de leçons.

Les cas d'utilisation sont groupés par deux catégories :

- **Environnement de création** : catégorie dans laquelle les fonctionnalités ont pour but de créer ou modifier un scénario. Cet environnement peut aussi être appelé *back office*.
- **Environnement d'utilisation** : cette catégorie contient toutes les fonctionnalités relatives à la visualisation et l'utilisation des scénarios. Elle peut aussi être appelée *front office*.

Chaque cas d'utilisation décrit l'intention de l'acteur lors de son utilisation du système. De ce fait, le système doit proposer les fonctionnalités adéquates pour satisfaire ses utilisateurs.

Créer et éditer un scénario

Seuls les formateurs étant assignés au cours ont le droit de créer et éditer un scénario. En effet, le système *StudyPress* gère les cours, et permet aux administrateurs d'assigner aux formateurs des cours [9]. Un scénario est défini comme une leçon dans ce système. Ainsi, lors de la création d'un scénario, le formateur doit impérativement assigner le cours correspondant, ce qui permet à *StudyPress* d'ajouter une nouvelle leçon au cours.

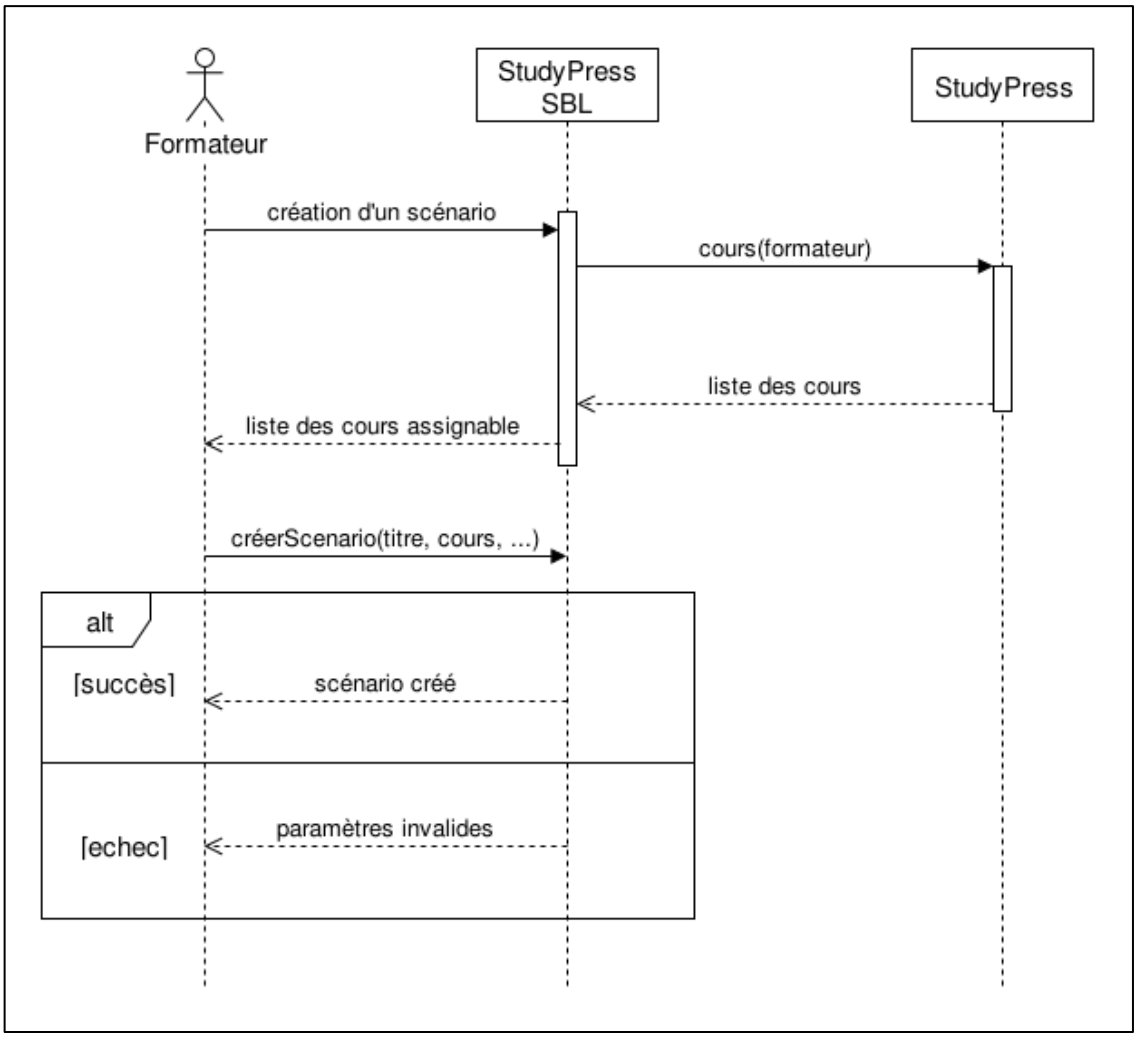

Figure II.4 : Diagramme de séquence pour la création des scénarios

Gérer les scènes

Un scénario est séparé en scènes. Gérer un scénario peut revenir à gérer les scènes de ce scénario. Chaque scène dans un scenario comporte un dialogue. Donc, la gestion se fait par l'édition du dialogue et des informations relatives à la scène (titre, description, etc.).

Créer un branchement

Le branchement dans un scénario se fait à la fin d'une scène. Le formateur gère un branchement en ajoutant et en modifiant un défi et les multiples choix de réponses à la scène.

Gérer les personnages

Le dialogue dans une scène se fait entre des personnages. La gestion des personnages est faite par le formateur en créant ou modifiant un personnage, tout en définissant son nom, avatar, etc.

Publier un scénario

Les scénarios peuvent être publiés dans des articles *WordPress*. L'article doit contenir un lecteur de scénario pour que les apprenants puissent le lire.

Lire un scénario

La lecture du scénario peut se fait par les apprenants. Elle consiste à voir le scénario sous forme de présentation dans un article WordPress.

Recommencer ou continuer la lecture d'un scénario

Un apprenant connecté peut lire un scénario comme le fait un visiteur non-connecté. Cependant, l'état de la lecture du scénario est sauvegardé automatiquement. Ainsi, à la prochaine lecture du scénario, l'apprenant peut continuer la lecture là où il s'est arrêté la dernière fois.

d- Exigences non fonctionnelles

En plus des exigences fonctionnelles, le composant doit répondre aux critères suivants :

- **L'ergonomie** : le produit final doit être facile à utiliser. L'interface se doit être intuitive étant donné que l'utilisateur final ne doit pas forcément être initié à l'informatique.
- **L'intégration** : le composant à réaliser doit s'intégrer dans l'extension *StudyPress*. C'est pourquoi, l'implémentation doit rester modulaire.
- **La rapidité de traitement** : Le composant doit être performant et léger. Il ne doit pas alourdir le site *WordPress* sachant que d'autres extensions peuvent être installées à côté de *StudyPress*.

Rapport-Gratuit.com

II-3 Contraintes et choix technologique

a- Contraintes

WordPress est système de gestion de contenu écrit en *PHP* et reposant sur une base de données *MySQL*. Il est possible de modifier et d'ajouter des fonctionnalités à travers des extensions.

Les extensions sont des modules tierces qui se greffe sur le système *WordPress* et apporte des améliorations diverses et variées, selon le but de l'extension. Elles permettent aux utilisateurs d'ajouter des fonctionnalités sans passer par un développeur. Techniquement, c'est un programme écrit en *PHP* qui permet d'ajouter des fonctionnalités et services à un site *WordPress* et d'être intégré de façon transparente en utilisant des points d'accès et des fonctions fournis par l'interface de programmation applicative (ou *API* en anglais pour Application Programming Interface).

StudyPress est une extension *WordPress* qui permet de créer et gérer des formations. En d'autres termes, *StudyPress* est un système de gestion de cours mais aussi un outil auteur. Il permet aux apprenants d'accéder à des cours mis à disposition par les enseignants. Il est possible aussi aux enseignants de créer du contenu e-learning sans avoir des compétences informatiques. L'extension utilise la base de donnée *MySQL*, utilisée par *WordPress*, pour stoker les données relatives aux cours et aux leçons [9].

Comme le système à développer, *StudyPress SBL*, est une extension au système *StudyPress*, il y a certaines contraintes auxquelles le composant doit évoluer. Le composant utilise *PHP* comme langage de programmation dans le développement de l'extension, mais aussi, *HTML*, *CSS*, *JavaScript* pour la partie concernant l'interface utilisateur.

PHP (PHP Hypertext Preprocessor) est un langage de programmation utilisé pour la produire des pages web dynamiques. Il est exécuté sur un serveur web. Il existe différentes versions pour le langage. Le code écrit dans ce projet nécessite *PHP 5.4* ou une version ultérieure.

HTML (Hypertext Markup Langage) est langage de balisage qui permet d'écrire des documents hypertexte qui représente les pages web. Il permet de mettre en forme le contenu des pages web, des formulaires de saisie, d'inclure des images, du son, etc. Il

est utilisé conjointement dans ce projet avec le langage de programmation *JavaScript* pour l'interactivité et des feuilles de style en cascade, *CSS* (Cascading Style Sheet), pour la mise en page du contenu. Pour une meilleure expérience utilisateur, le projet nécessite un navigateur web compatible les standards *HTML 5*, *CSS 3* et *ES 5* (*ECMAScript 5* pour le langage *JavaScript*).

b- Choix technologique

Pour une meilleure gestion de développement et une facilité pour la programmation, des outils et technologies sont choisis. Ils permettent de créer les fondations ainsi que les grandes lignes du composant à développer, facilitant ainsi la modélisation de l'architecture.

Herbert

Quand il s'agit de la création d'un plugin *WordPress*, les développeurs sont libres de comment structurer leur code et organiser leurs fichiers. Pour un petit projet cela peut se voir comme un avantage, mais une fois le projet grandi, le développement devient difficile et peu flexible.

Chaque année *Stack Overflow* publie son enquête sur les plateformes les plus redouté par les développeurs. En 2016, *WordPress* a été classé 2ème comme étant la plateforme la plus crainte des développeurs [10].

En janvier 2015, *Jason ANGNEW*, le directeur technique de *Big Bit Creative*, l'un des douze partenaires *WordPress*, a annoncé *Herbert*. *Herbert*[5] est un framework permettant de développer des extensions *WordPress*. Il est basé sur le pattern Modèle-Vue-Contrôleur (MVC), ce qui offre une meilleure flexibilité et maintenabilité des plugins.

Herbert fournit des fonctionnalités comme le routage des requêtes, le mapping objet-relationnel, un système de migration de base de données, etc. Ces fonctionnalités sont disponibles grâce à des bibliothèques se trouvant dans d'autres framework. Par exemple, pour le mapping objet-relationnel et l'abstraction des données, *Herbert* utilise *Eloquent* qui est une bibliothèque du framework *Laravel*[6]. Aussi, pour la séparation

[5] http://getherbert.com/
[6] https://laravel.com/

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

entre la vue et les contrôleurs, le framework utilise *Twig* qui est une bibliothèque de *Symfony*[7] comme moteur de templates.

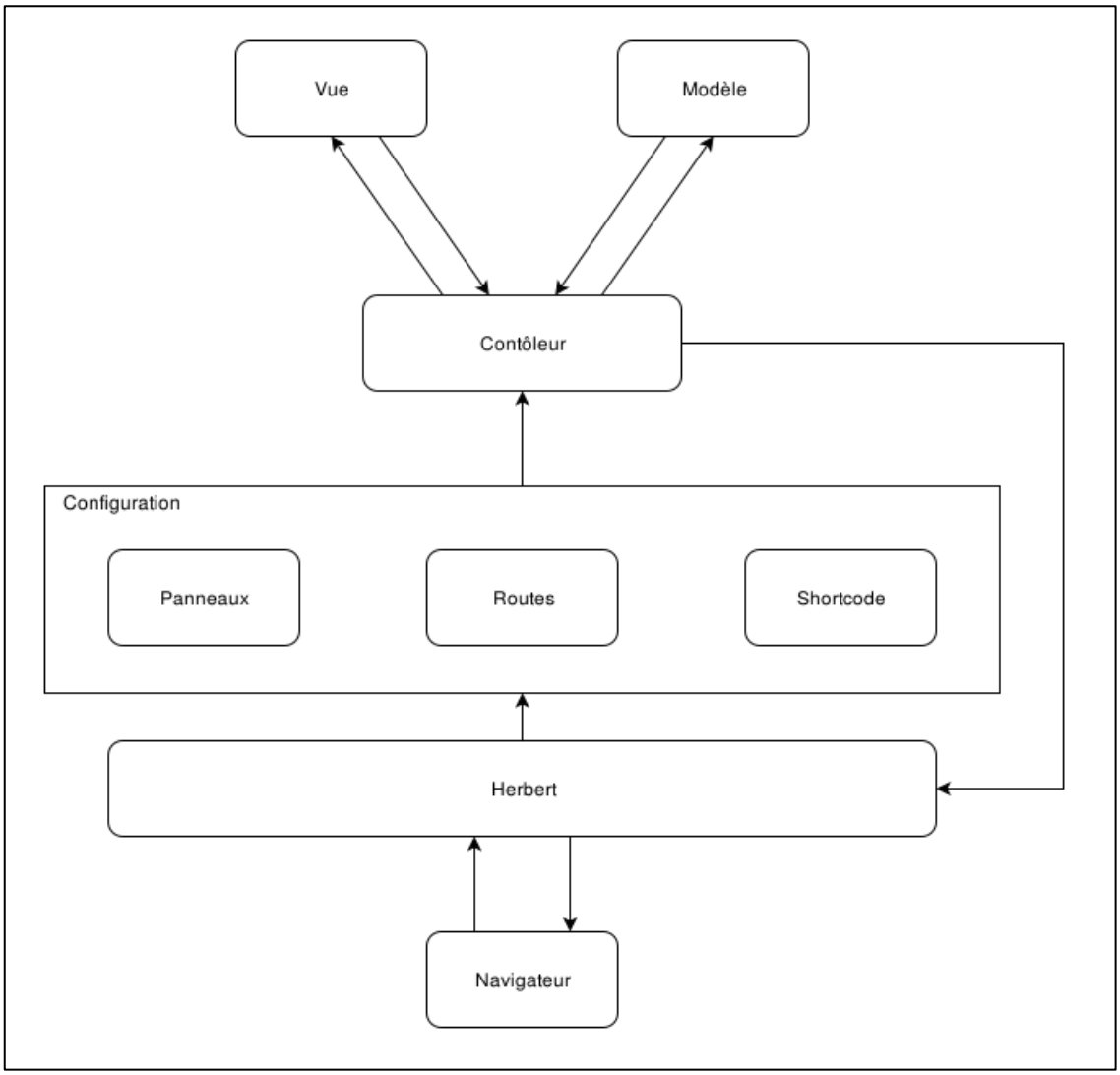

Figure II.5 : Architecture d'une application Herbert

Herbert sépare le code en 4 parties :

- **Modèle** : Code concernant les données issues de la base de données. Ces données sont accessibles grâce à la bibliothèque *Eloquent*. Cette dernière utilise le paterne Active Record pour la définition des objets-relationnel.
- **Vue** : C'est la partie qui concerne le document HTML, JSON ou XML à retourner comme réponse aux requêtes. Ces documents, grâce à *Twig*, peuvent avoir une syntaxe étendue pour accéder à plus de fonctionnalités.

[7] https://laravel.com/

- **Contrôleur** : Les contrôleurs ont pour rôle de traiter les requêtes. Ils prennent en charge la gestion et la synchronisation du modèle et la vue.
- **Configuration** : Cette partie concerne la configuration et l'association des fonctionnalités à *WordPress*. Elle permet, par exemple, d'ajouter des menus dans le panneau d'administration du site *WordPress* (*panneaux*), ou la configuration du routage des requêtes (*Routes*) pour le traitement de réponses. Aussi, cette partie permet de créer des *shortcodes*. Les shortcodes sont des balises de texte qui peuvent être placés dans un article ou une page du site *WordPress*. Ces balises sont automatiquement interprétées par WordPress et permettent d'ajouter de nouvelles fonctionnalités.

AngularJS

AngularJS[8] est un framework *JavaScript*. Il est exécuté sur le navigateur web. Il permet de simplifier la création d'application web moderne, à page unique, en utilisant *AJAX*. Il est notamment utilisé dans les applications type CRUD (Create Read Update Delete). Les fonctionnalités de ces applications se résument sur l'accès, la création, la modification et la suppression de données.

La particularité d'*AngularJS* est d'avoir un système de template se basant sur le langage *HTML*. Ces templates sont actualisés sans avoir à recharger la page. Il est même possible d'étendre le langage *HTML* par de nouvelles balises et attributs. Aussi, le framework intègre le support de l'injection des dépendances. Cette dernière permet de découper l'application en petits fragments ce qui peut faciliter la maintenance.

AngularJS utilise le patron de conception MVC et sépare la présentation, les données, et les composants métiers. Les templates *HTML* correspondent à la vue et les contrôleurs sont définis à travers des fonctions *JavaScript*. Le modèle est représenté par un simple objet JavaScript.

L'injection de dépendances dans *AngularJS* permet de créer dynamiquement (injecter) des objets.

Un objet dans *AngularJS* peut être créé de différentes manières. Il est ainsi possible d'assigner directement une valeur dans un objet (Value), ou de créer une fonction qui et

[8] https://angularjs.org/

appelé qu'une fois pour la création de l'objet (Service), ou de créer une fonction qui renvoie un objet et qui est appelée à chaque fois que le code en a besoin (Factory), ou définir l'objet comme une constante (Constant).

En plus de créer des objets dynamiquement, il est possible d'étendre le langage *HTML* avec des directives (Directive) et de modifier l'affichage des données avec filtres (Filter). Les pages sont routées (Routes), c'est à dire qu'*AngularJS* permet l'accès directe à une page à travers une *URL* (Uniform Resource Locator).

Pour ainsi séparer et encapsuler le code, *AngularJS* permet de regrouper le code dans des modules. Ces modules peuvent dépendre les uns des autres suivant les objets à injecter, les contrôleurs, filtres, directives, etc. Ainsi, il est facile de structure, maintenir et tester l'application.

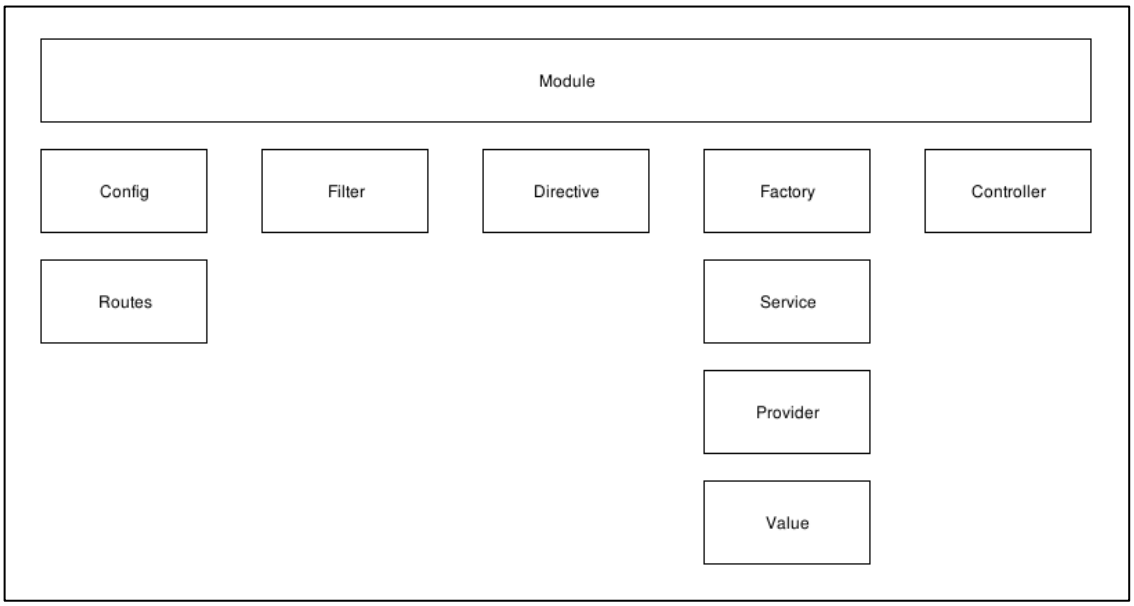

Figure II.6 : Anatomie d'un module dans AngularJS

Bootstrap

Bootstrap[9] est un framework regroupant des outils pour faciliter la création d'interface utilisateur. Il contient des modèles *HTML* et *CSS* pour la typographie, les formulaires, les boutons, outils de navigation, etc. Il est possible grâce au framework de concevoir des sites web adaptatifs et offrir ainsi aux pages web une adaptation

[9] http://getbootstrap.com/

dynamique aux formats de supports utilisés (moniteur d'ordinateur, smartphone, tablette, etc.).

Une variante de Bootstrap est utilisée dans ce projet, *bootstrap-sass*[10]. Cette variante utilise *Sass* (Syntactically Awesome Stylesheet) au lieu de *CSS*. Sass est un langage de feuille style interprété en *CSS*. Il permet d'étendre le langage en ajoutant des fonctionnalités généralement disponibles dans les langages de programmation plus traditionnel. Ainsi, l'intégration de *Bootstrap* dans *WordPress* est facilement réalisable.

Reveal.js

Reveal.js[11] est un framework pour créer des présentations *HTML*. Les diapositives sont créées en écrivant un simple code *HTML*, et c'est grâce aux scripts *JavaScript* et les feuilles de style *CSS* que la mise en page est effectuée. Les diapositives peuvent être imbriquées, utiles pour organiser les éléments communs dans un même ensemble. Il n'est pas nécessaire d'héberger les présentations sur un serveur web, il est possible de les lire directement sur le navigateur web. *Reveal.js* dispose d'une API permettant d'étendre ses fonctionnalités via des extensions.

Grunt et Robo

Grunt[12] est un lanceur de tâches (ou task runner en anglais) conçu pour le développement frontend (développement côté client dans une application client/serveur). Il est, d'ailleurs, lui-même écrit en JavaScript et utilise Node.js. *Grunt* fonctionne grâce à des extensions, chacune dédiée à une tache spécifique, comme :

- la concaténation et la minimification des scripts ;
- la compilation de fichiers Sass ;
- le lancement et exécution de tests ;
- la vérification et la validation de syntaxes.

[10] https://github.com/twbs/bootstrap-sass
[11] http://lab.hakim.se/reveal-js/
[12] http://gruntjs.com/

Comme *Grunt*, *Robo*[13] est aussi un lanceur de tâches. Cependant, *Robo* est conçu pour le développement backend (développement côté serveur dans une application client/serveur). Il est écrit en *PHP* et permet comme *Grunt* d'automatiser les tâches.

[13] http://robo.li/

III- Architecture logicielle

III-1 Architecture globale

L'interface de WordPress est divisée en deux parties :

- **back office** : c'est la zone où les réglages sont faits et où le contenu est écrit, par exemple : la gestion des images, la gestion des nouveaux commentaires, la modification du thème du site, etc.
- **front office** : c'est la partie visible du site. Tout le contenu créé dans le back office est affiché dans le front office, par exemple : des pages, des articles, des widgets, etc.

Le composant à développer contient une interface pour la création des scénarios et une autre pour afficher à l'utilisateur ces scénarios sous forme d'un article contenant une présentation interactive. Donc, le composant doit ajouter des fonctionnalités au back office et au front office de *WordPress*.

Le composant s'intègre à l'extension *StudyPress*. *WordPress* ne limite pas les extensions d'utiliser que certaines fonctionnalités du langage et ne les contraint pas à avoir un architecture bien défini. L'architecture choisie pour le développement de ce composant est une architecture 3-tiers.

L'architecture 3-tiers, ou l'architecture à trois niveaux, est une architecture client-serveur dans laquelle l'application est modélisée comme un empilement de trois couches logicielles [11] :

- **couche présentation** : correspond à l'affichage et l'interface de dialogue avec l'utilisateur à travers le poste de travail.
- **couche métier** : correspond à la gestion et la mise en œuvre de l'ensemble des règles de gestion et la logique applicative.
- **couche accès aux données** : correspond à la persistance et la conservation des données.

Dans l'architecture du composant *StudyPress SBL*, la couche accès aux données est gérée par le système de gestion de base de données *MySQL*, et les données sont sauvegardées dans la base de données de *WordPress*. La couche métier correspond au

code *PHP* à ajouter à l'extension *StudyPress*. Enfin, la couche présentation correspond aux interfaces disponibles dans le back office et le front office. De cette manière, il est possible d'avoir un code PHP unique pour deux interfaces différentes.

Pour une meilleur gestion et maîtrise, le composant est découpé en 3 sous-projets :

- **studypress-sbl** : projet contenant le code PHP à intégrer dans l'extension *StudyPress*.
- **sbl-creator** : projet correspondant à l'interface de création de scénarios dans le back office.
- **sbl-player** : projet correspondant au lecteur de scénarios, utilisé dans le front office.

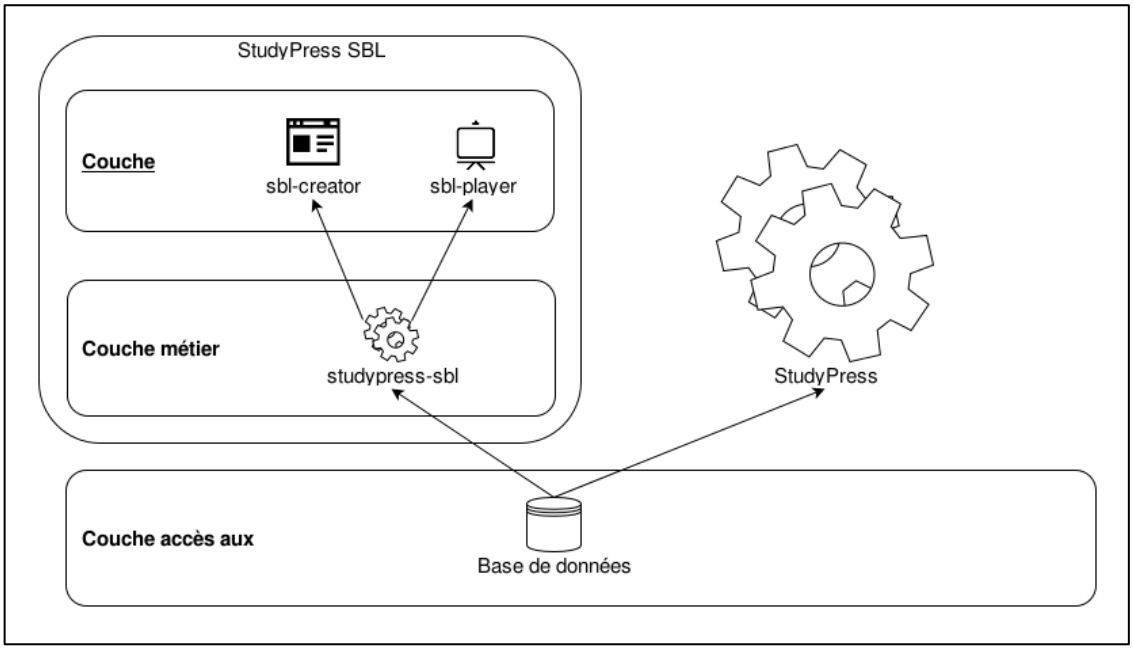

Figure III.1 : Architecture globale de l'extension StudyPress

Le modèle de données utilisé dans le composant *StudyPress SBL* est relativement simple. Il est illustré dans le diagramme suivant :

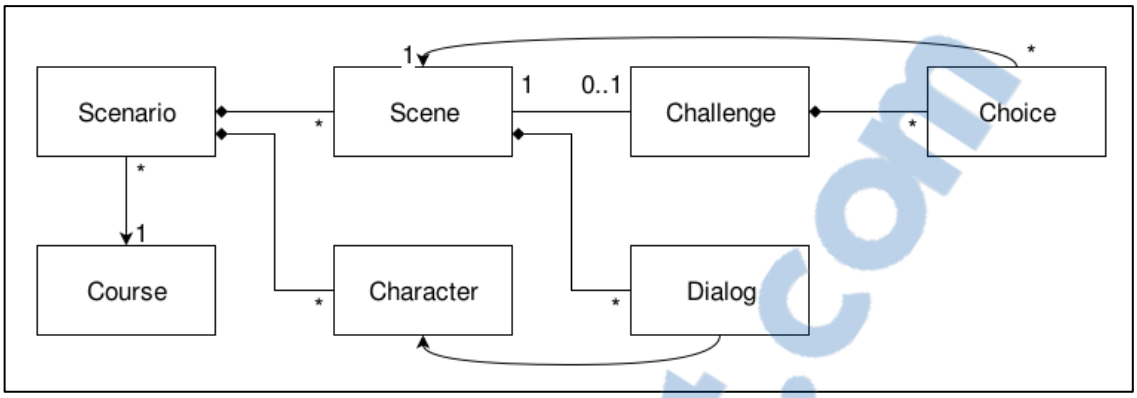

Figure III.2 : Modèle de données

Le modèle défini 7 entités de données :

Ensemble	Description
`Course`	L'entité correspondant aux cours disponibles dans *StudyPress*.
`Scenario`	Contient les informations relatives au scénario d'apprentissage (titre, description, etc.).
`Character`	Représente l'ensemble des personnages disponibles dans le scénario.
`Scene`	Ensemble de scènes dans un scénario. La scène représente une situation dans un scénario d'apprentissage.
`Dialog`	Chaque scène contient un dialogue entre des personnages. Cet ensemble regroupe les messages et les images de chaque personnage.
`Challenge`	Cet ensemble représente l'ensemble des défis dans un scénario.
`Choice`	Les choix de réponses dans les défis sont regroupés dans cet ensemble.

Tableau III.1 : Dictionnaire de données

La couche présentation accède aux données en communiquant avec la couche métier. Cette communication se fait à travers le web en respectant le style *REST* (Representational State Transfer).

REST est un style d'architecture qui est constitué d'un ensemble de composants, connecteurs, et de données dans un système hypermédia distribué [12]. Le but étant de réduire le trafic, et de rendre le système de communication performant, simple, modifiable, visible et portable. Pour ce faire, *StudyPress SBL* utilise le même protocole que celui utilisé dans le *World Wide Web*, *HTTP* (Hypertext Transfer Protocol) avec les

Rapport-Gratuit.com

mêmes méthodes (GET, POST, PUT, DELETE, etc.) qui sont des commandes que le navigateur utilise pour la réception et l'envoie de données.

L'interface de communication avec les ressources (scenario, scene, character, etc.) sont identifiées par des *URI* (Uniform Resource Identifier).

Le tableau suivant résume l'interface de communication entre la couche présentation et la couche métier. A savoir que les *URI* sont relatives et que les segments dans l'URI préfixé par « `$` » représentent des segments variable, c'est à dire qu'ils sont remplacés par n'importe quel valeur, comme il est le cas ici d'identifiant de ressource :

Méthode	**URI**	**Description**
GET	/scenarios	Liste des scénarios disponibles dans le site
GET	/scenarios/$scenario_id	Reçoit les détails d'un scénario selon son ID
POST	/scenarios	Créer un nouveau scénario avec les données envoyées
PUT	/scenarios/$scenario_id	Met à jour les informations du scénario
DELETE	/scenarios/$scenario_id	Supprime le scénario
GET	/scenarios/$scenarios_id/scenes	Liste des scènes concernant le scénario selon son ID
GET	/scenarios/$scenario_id/scenes/$scene_id	Les détails d'une scène selon l'ID du scénario et l'ID de la scène
POST	/scenarios/$scenario_id/scenes	Créer une nouvelle scène et l'ajouter au scénario identifié par son ID
PUT	/scenarios/$scenario_id/scenes/$scene_id	Met à jour les informations de la scène
DELETE	/scenarios/$scenario_id/scenes/$scene_id	Supprime la scène
GET	/scenarios/$scenario_id/scenes/$scenes_id/dialog	Liste des messages dans le dialogue d'une scène
POST	/scenarios/$scenario_id/scenes/$scene_id/dialog	Créer un nouveau message au dialogue
PUT	/scenarios/$scenario_id/scenes/$scene_id/dialog/$dialog_id	Met à jours les informations concernant le message dans un dialogue
DELETE	/scenarios/$scenario_id/sce	Supprime le message dans un dialogue

	nes/$scenes_id/dialog/$dialog_id	
GET	/scenarios/$scenario_id/characters	Liste des personnages disponibles dans un scénario selon l'ID du scénario
GET	/scenarios/$scenario_id/characters/$character_id	Reçoit les détails d'un personnage selon l'ID du scénario et l'ID du personnage
POST	/scenarios/$scenario_id/characters/$character_id	Créer un nouveau personnage au scénario identifié par l'ID
PUT	/scenarios/$scenario_id/characters/$character_id	Met à jour les informations concernant le personnage selon son ID et l'ID du scénario
DELETE	/scenarios/$scenario_id/characters/$character_id	Supprime un personnage du scénario
POST	/scenarios/$scenarios_id/scenes/$scene_id/branch/$branch_id/decisions	Ajoute un nouveau choix de réponse au défi
PUT	/scenarios/$scenarios_id/scenes/$scene_id/branch/$branch_id/decisions/$decision_id	Met à jour les informations concernant le choix de réponse
DELETE	/scenarios/$scenarios_id/scenes/$scene_id/branch/$branch_id/decisions/$decision_id	Supprime un choix de réponse d'un défi selon l'ID du scénario, l'ID de la scène et l'ID du défis

Tableau III.2 : API utilisée pour l'accès aux resources du serveur

Le format d'échange choisi pour la communication est *JSON* (JavaScript Object Notation). Ce format permet de représenter les données sous forme de paires nom/valeur. La syntaxe est dérivée de la notation des objets du langage *JavaScript*.

La communication entre la couche métier la couche accès aux données se fait à travers des appels de fonctions disponibles dans l'API *WordPress*.

III-2 Architecture détaillée

a- studypress-sbl

Le projet *studypress-sbl* comprend le code *PHP* à intégrer à l'extension *StudyPress*. Il joue le rôle du tiers métier dans l'architecture 3-tiers.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Le projet suit une architecture *hexagonale* [13]. L'architecture hexagonale repose sur l'idée d'isoler la logique métier des détails techniques de son implémentation. Le métier ou le modèle est situé dans un hexagone, tandis que les détails techniques ou infrastructure sont situés en dehors de l'hexagone. Aussi, le code métier ne dépend de rien. En d'autres termes, c'est l'infrastructure qui doit dépendre du métier. Pour ce faire, le modèle expose des interfaces (APIs) pour communiquer avec d'autres métiers ou services.

studypress-sbl utilise deux hexagones, *Scénario* et *Cours* :

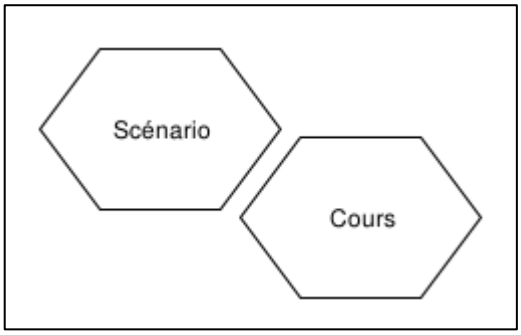

Figure III.3 : Architecture héxagonal du projet studypress-sbl

L'hexagone *Scénario* comprend les modèles et la logique métier concernant les scénarios d'apprentissage. L'hexagone *Cours* contient que les modèles *Course* et *Activity*. Ces modèles sont issues de l'extension *StudyPress* et dans l'hexagone il n'est question que de mapping de données. La logique métier est traitée dans l'extension *StudyPress*. Ces modèles sont importants pour l'intégration du composant étant donné qu'un scénario est une nouvelle activité dans un cours [9].

Cette manière de séparer les modèles permet une testabilité accrue du métier et permet aussi un changement technique sans risque d'introduire une régression fonctionnelle.

studypress-sbl utilise le framework de développement *Herbert* en appliquant le paradigme de programmation orienté objet. Différentes couches constituent le composant :

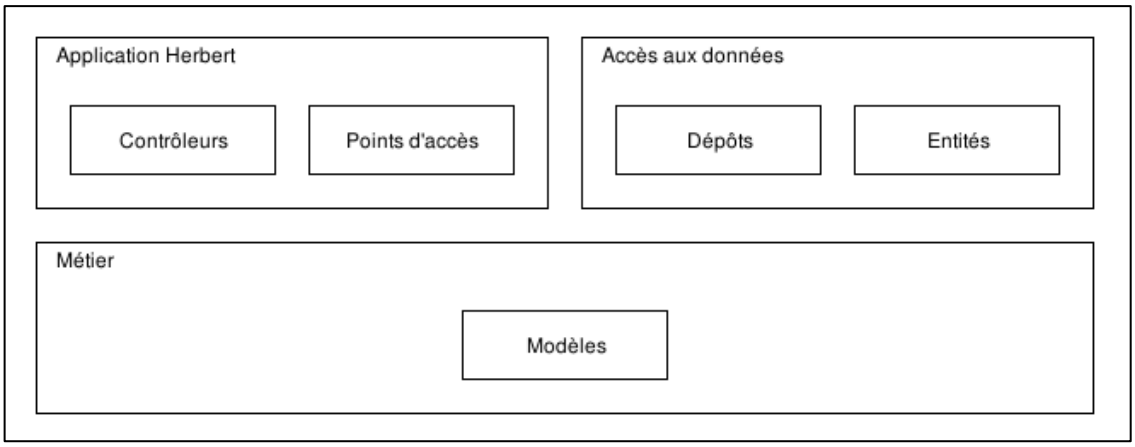

Figure III.4 : Architecture de studypress-sbl sous forme de couches

Les dépendances vont vers le bas, c'est à dire que la couche application *Herbert* dépend de la couche métier, mais la couche métier ne dépend d'aucune autre couche. La couche application *Herbert* contient les classes contrôleurs pour la gestion des requêtes et les classes points d'accès pour l'API REST. La couche accès aux données contient les classes entités pour faire la correspondance entre les éléments d'un ensemble de données et les classes dépôts pour la persistance des données. Quant à la couche métier, elle contient les différents classes pour le traitement de la logique métier.

Par ailleurs, le modèle est implémenté en utilisant les spécificités de la programmation orienté objet comme l'héritage mais aussi celles du *PHP* comme les variables non typés.

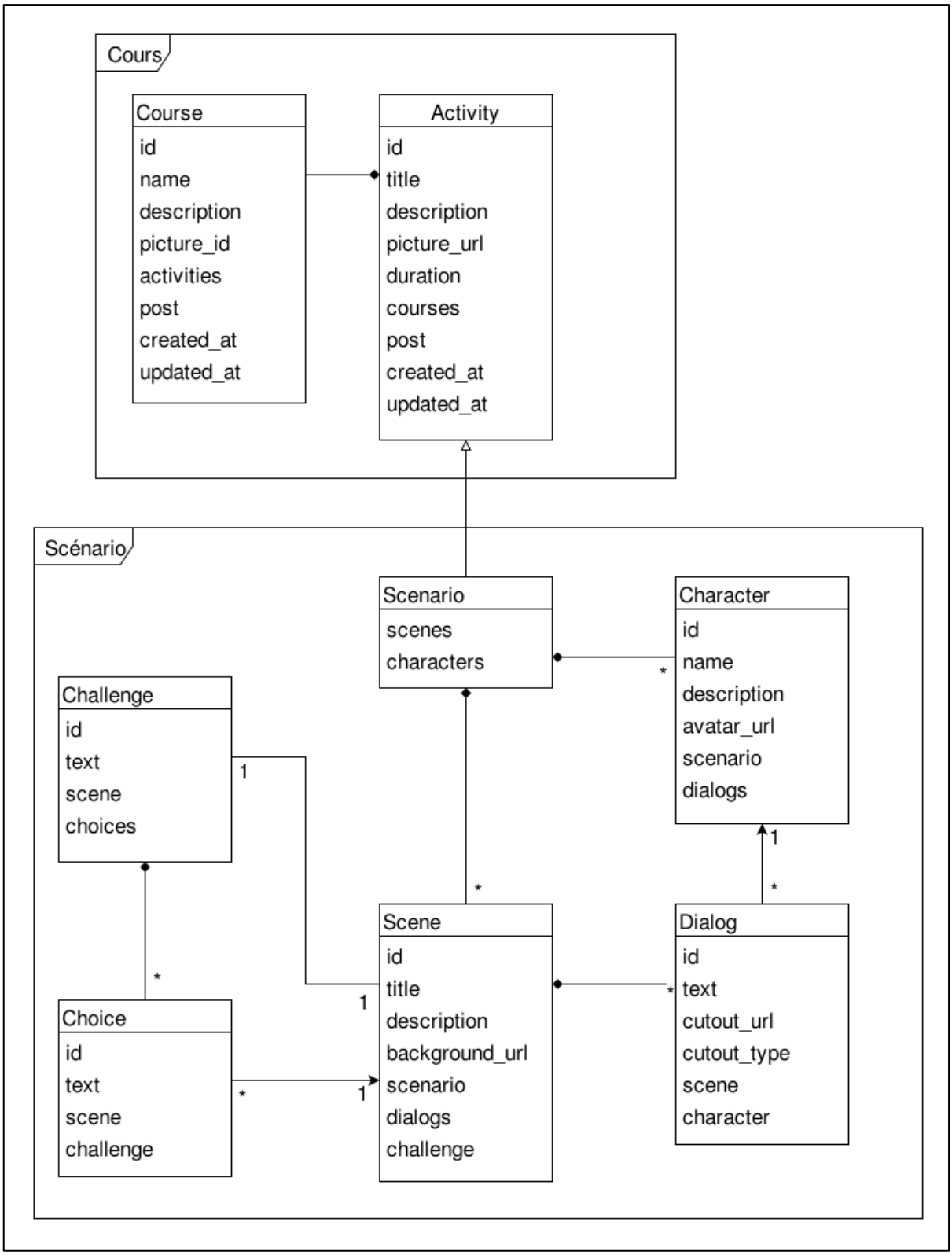

Figure III.5 : Diagramme de classe utilisé dans les hexagones

b- sbl-creator

Ce projet se résume à la réalisation d'une application web avec *AngularJS*. Le but de l'application est d'aider les formateurs à gérer les scénarios. Elle permet entre autres de:

- récupérer, afficher et modifier les données issues du serveur ;
- valider les formulaires en temps réel ;
- avoir une navigation complexe comme une interface avec des éléments modaux.

L'architecture utilisée divise les modules en trois parties :

- **Application** : Cette partie contient un seul module : `App`. C'est le module supérieur à tous les autres modules. Il dépend des modules requis pour le bon fonctionnement de l'application notamment ceux implémentant les fonctionnalités de base de l'application et ceux concernant les fonctionnalités transversales.
- **Fonctionnalités** : concerne les modules implémentant les fonctionnalités de base de l'application. Ce projet consiste à créer un outil pour la création de scénarios, donc la fonctionnalité de base est la gestion de scénarios. Les dépendances des modules dans cette partie sont gérées suivant la navigation dans l'interface de l'application. Ainsi, le module `scenarios` dépend par exemple du module `scenes`, car dans l'application, c'est à partir d'un scénario qu'une liste des scènes est accessible. En générale, ces modules contiennent les contrôleurs des templates *HTML* ainsi que les configurations pour la navigation.
- **Infrastructure** : Les modules dans cette partie concernent l'infrastructure de l'application. Cette partie est divisé à son tour an trois :
    - <u>services</u> : Cette partie contient les modules qui concernent les préoccupations transversales telles que la gestion des notifications (`notifications`) et la détection des erreurs HTTP (`httpExceptionInterceptor`).
    - <u>resources</u> : Les modules dans cette partie permettent la communication avec le serveur. c'est à travers ces modules que l'application peut accéder et modifier les données.

- directives : Les scripts contenant les directives sont groupés dans cette partie. Ils sont divisés en modules dont chacun répond à un but. Le module `directives.modal` simplifie la création d'élément modal. Le module `directives.browse` permet, quant à lui, d'abstraire l'accès aux médias disponibles dans WordPress.

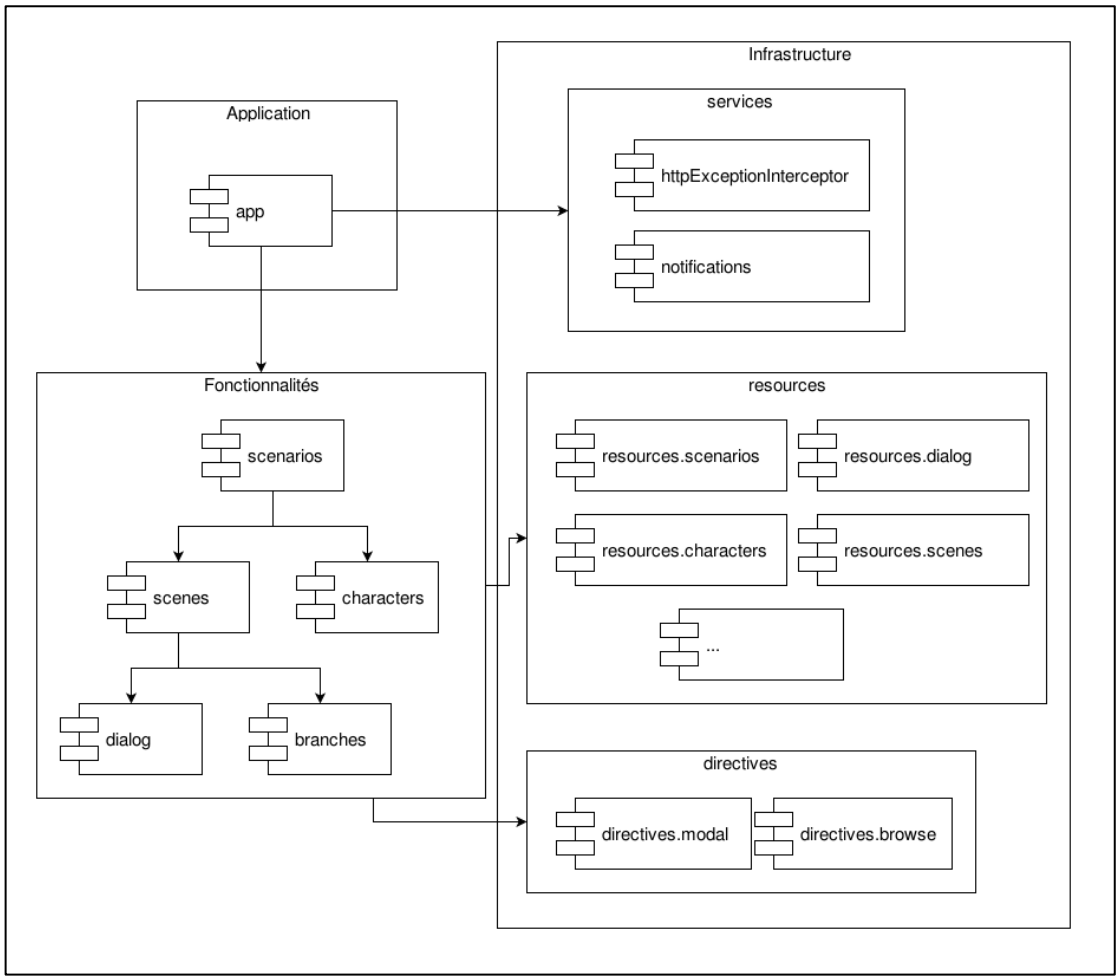

Figure III.6 : Architecture utilisée dans le projet sbl-creator

c- sbl-player

Ce dernier projet est un dérivé (fork) du projet *Reveal.js*. Il permet d'afficher les scénarios aux apprenants sous forme de présentations interactives. Il est, en quelque sorte, un lecteur de scénarios.

Comme *Reveal.js* intègre un système d'extensions, il n'a pas été nécessaire de modifier les scripts de ce dernier. En revanche, pour accéder et afficher un scénario, une extension est mise au point.

Lors du chargement de la présentation, l'extension récupère les données relatives au scénario (titre, scènes, dialogues, etc.) et construit la présentation en ajoutant des éléments *HTML* au *DOM* (Document Object Model) du lecteur. La récupération des données se fait en utilisant *AJAX*, et l'URL du scénario est passé comme paramètre GET au lecteur.

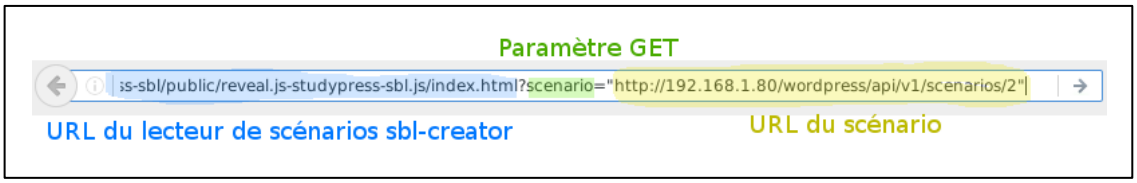

Figure III.7 : Récupération de l'URL du scénario

Après la récupération du scénario, c'est à l'aide d'un template en utilisant la bibliothèque *Handlebars*[14], que les éléments de la présentation sont construits. Étant donné que *Reveal.js* supporte les diapositives imbriquées, dans la présentation obtenue, chaque scène est représentée par une diapositive et le dialogue de la scène est représenté aussi par des diapositives, cependant, ces dernières sont imbriquées dans celle de la scène.

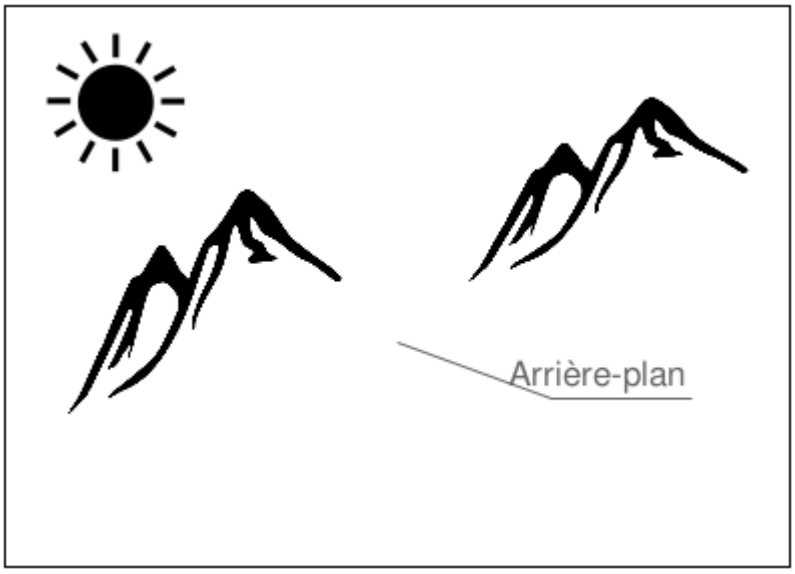

Figure III.8 : Diapositive représentant une scène

[14] http://handlebarsjs.com/

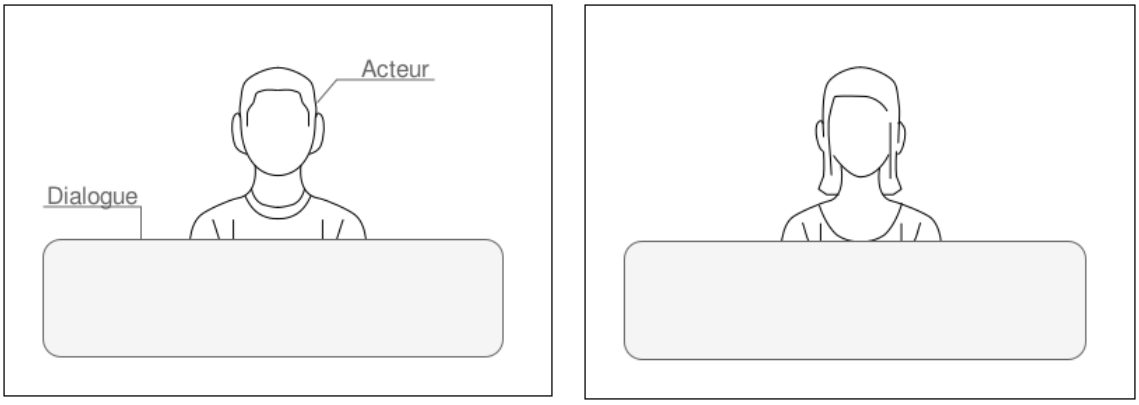

Figure III.9 : Diapositives représentant le dialogue dans une scène

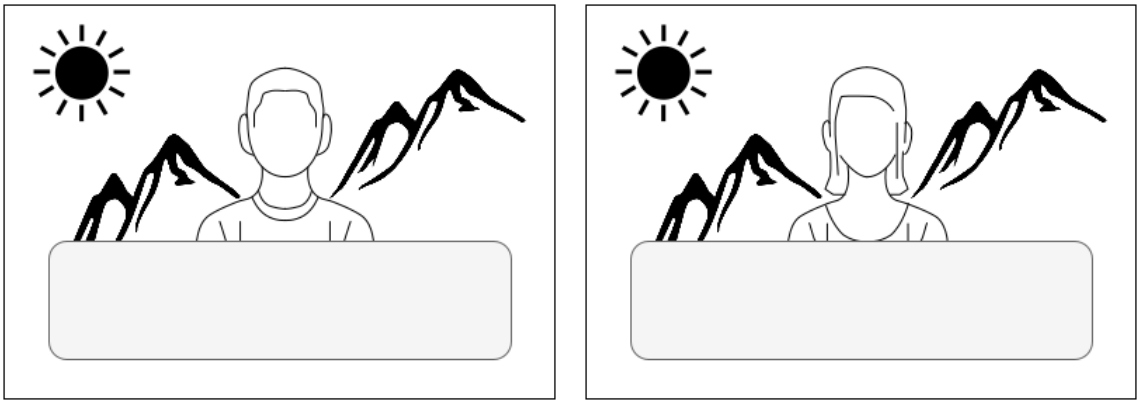

Figure III.10 : Diapositives après imbrication

Le défi dans la présentation est représenté par une diapositive. Cette diapositive est imbriquée et située à la fin de la scène, c'est à dire après les diapositives du dialogue. Les choix sont représentés par des boutons. Lors d'un clic sur le bouton, la présentation fait un saut vers la diapositive correspondant au choix.

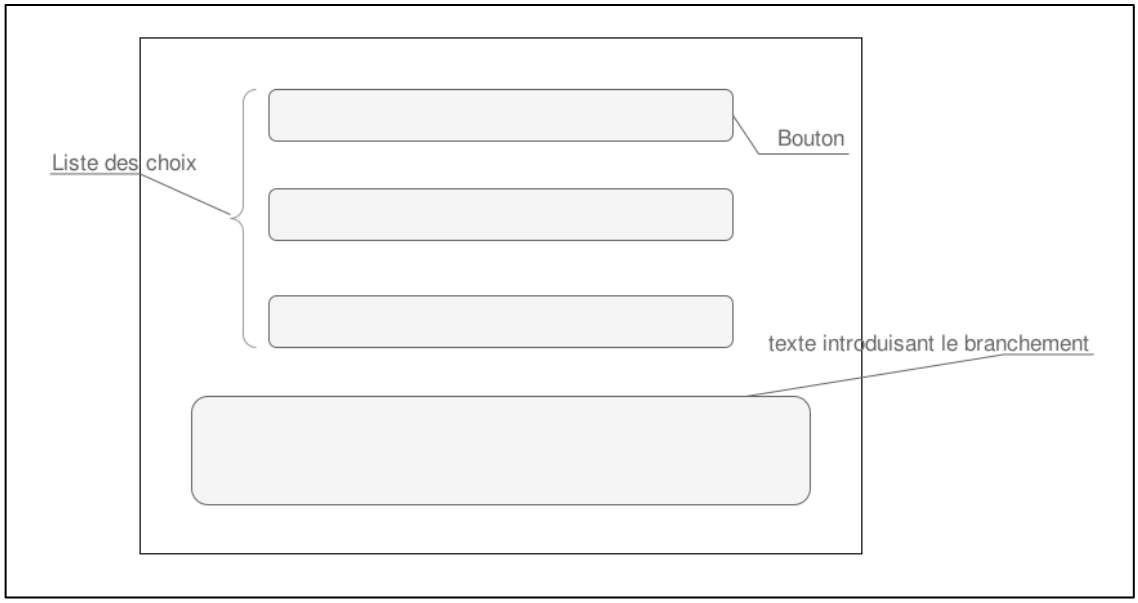

Figure III.11 : Représentation d'un défi dans la présentation

Reveal.js permet de changer le thème d'affichage des présentations. Ces thèmes sont écrits avec le langage *Sass*. Comme de nouveaux éléments non mis en page sont créés dans la présentation (texte pour le dialogue, liste des choix dans le branchement, etc.), un thème est créé pour modifier l'affichage de ces éléments et de les positionner.

Bootstrap permet d'embarquer des éléments adaptatifs dans une page web. Ainsi, en utilisant un élément *HTML* `iframe` dont la valeur de l'attribut `src` pointe vers le lecteur de présentation sbl-player (`index.html`) et intégrant les paramètres GET pour le scénario, la présentation est adaptatif et isolé de la page (les feuilles de style utilisées dans la page web n'interfère pas avec le lecteur). De cette manière, une présentation est embarquée dans un article *WordPress* de manière sûr et sans bugs.

Rapport-Gratuit.com

IV- Réalisation

IV-1 Démarche suivie

a- Système de gestion de versions

Dans le cas d'un développement en équipe, l'utilisation d'un système de gestion de versions est nécessaire. *Git*[15] est un logiciel qui assure un contrôle de versions distribué. Cet outil permet à chacun de travailler à son rythme tout en offrant un moyen aux développeurs d'échanger leurs travaux. Dans ce projet, *Git* est utilisé comme logiciel de gestion de versions avec un serveur d'hébergement de projets qui sert de point de rencontre entre les développeurs permettant ainsi une simplicité et une facilité d'utilisation.

GitHub[16] a été utilisé comme serveur d'hébergement de projets. C'est un service web proposant plusieurs fonctionnalités comme le suivi des projets, utile pour que les développeurs soient notifiés et donc rester synchronisés.

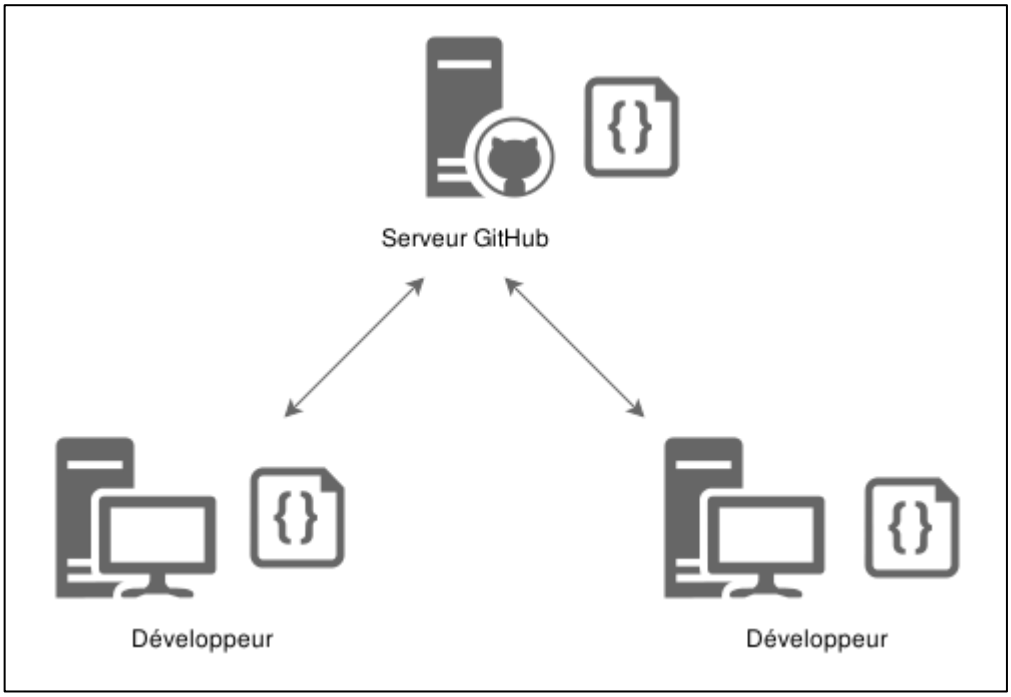

Figure IV.1 : Schéma d'utilisation de Git

[15] https://git-scm.com/
[16] https://github.com/

b- Environnement de développement

Au cours du développement, l'environnement doit ressembler le plus possible à l'environnement de production. De cette façon, la possibilité pour les utilisateurs de rencontrer des bugs est réduite. *Vagrant*[17] est utilisé pour permettre, via la virtualisation, d'avoir un environnement de développement similaire à celui des utilisateurs. Plus précisément, Vagrant est un outil permettant de créer des machines virtuelles afin d'obtenir l'environnement souhaité. Pour ce faire, l'outil fournit un environnement configurable, portable et reproductible, construit grâce aux logiciel de virtualisation comme *VirtualBox*, *VMWare*, etc., un fichier de configuration (allocation de la mémoire, nombre de cœur pour le processeur, etc.), et un système de provisionning (scripts *Shell*, *Chef* ou *Puppet*) qui permet d'automatiser l'installation et la configuration des logiciels sur la machine virtuelle.

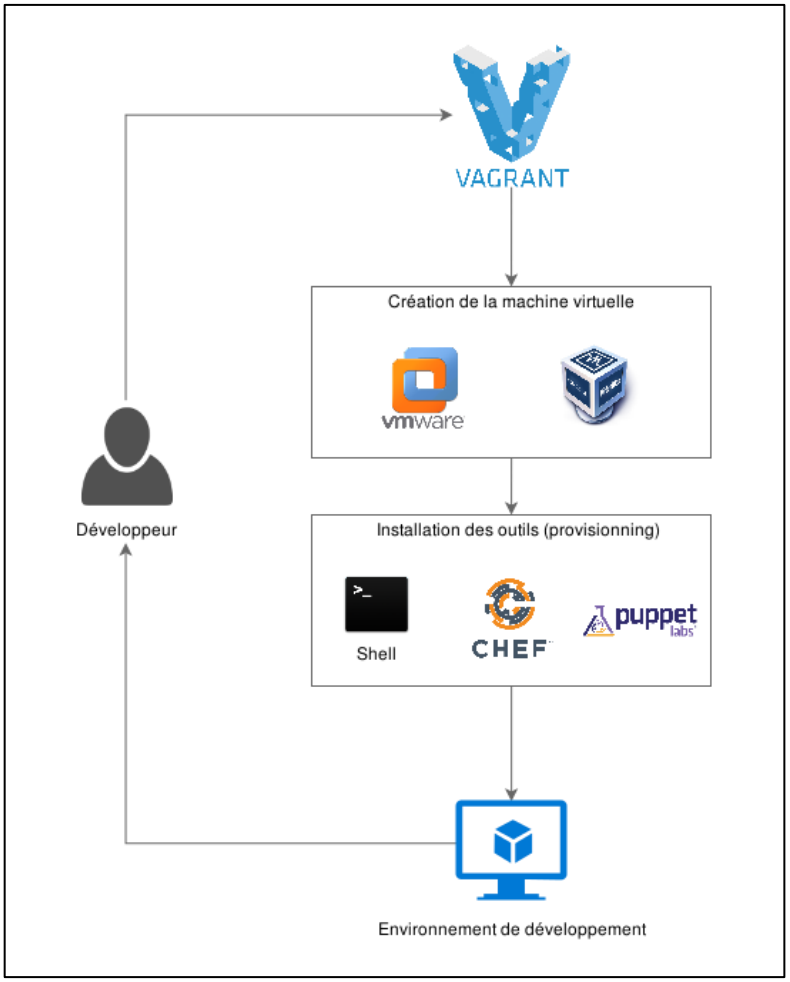

Figure IV.2 : Création de l'environnement de développement

[17] https://www.vagrantup.com/

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

c- Tests

Les tests sont un moyen pour trouver les erreurs rapidement. Ils permettent de faciliter la maintenance en détectant d'éventuelles régressions. Les tests sont aussi utilisés pour comparer la réalisation et la spécification, et donc ils peuvent être utilisés comme documentation.

Les tests permettent de vérifier le fonctionnement d'une portion du programme. Le processus durant dans le développement suit la pratique du Test-Driven Developpement (TDD). Cette pratique consiste à écrire les tests décrivant la spécification avant d'écrire le code de l'implémentant pour ainsi mieux répondre aux exigences de l'application. Elle est divisée en trois étapes :

1) **Rouge** : Tout d'abord, écrire le code pour le test. Évidemment, le test échoue étant donné que le code à tester n'est pas encore écrit, d'où le nom de l'étape « Rouge », représentant l'échec du test.
2) **Vert** : Dans cette étape, le développeur écrit le code à tester, suffisamment pour passer le test. Le nom « Vert » représente, cette fois-ci, le succès du test.
3) **Réusinage de code** : La dernière étape consiste à retravailler le code source précédemment écrit sans y ajouter des fonctionnalités. Le but est d'améliorer la lisibilité et donc la maintenance du code.

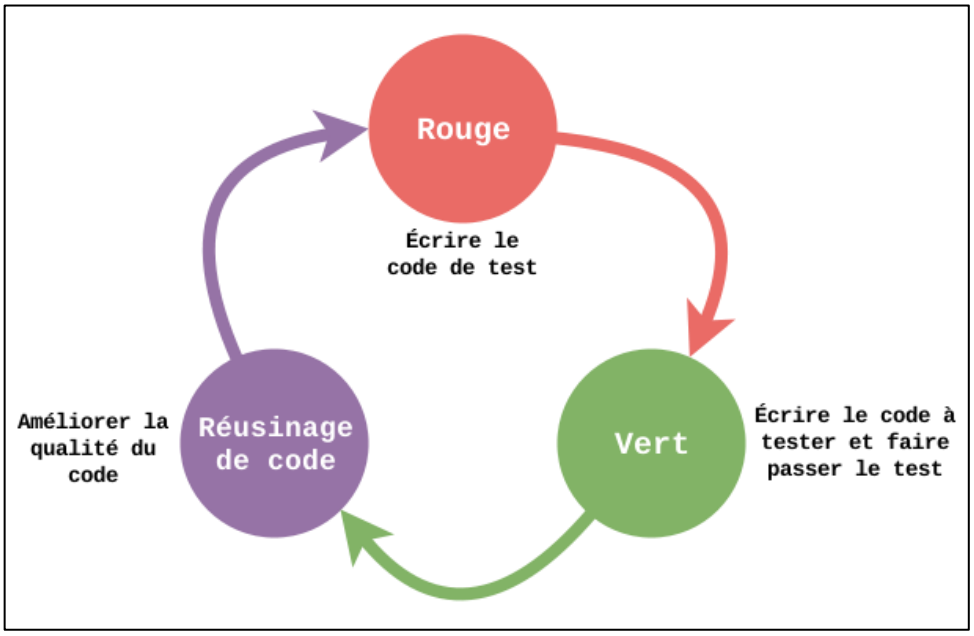

Figure IV.3 : Processus de développement TDD

Pour le projet studypress-sbl, ce sont les outils *php-spec*[18] et *Behat*[19] qui exécute les tests. Quant au projet sbl-creator, les tests sont exécutés grâce à l'outil *Karma*[20]. Ce dernier lance un serveur web qui charge l'application et exécute les tests qui sont écrit grâce au framework *Jasmine*[21]. Enfin, le projet sbl-player n'utilise pas de tests automatiques, notamment pour sa simplicité.

Le Behaviour-Driven Developpement (BDD) est une méthode agile qui met en avant le langage naturel et le langage du domaine dans les tests. Cela permet de se concentrer sur les raisons pour lesquelles le code doit être créé (exigences), plutôt que les détails techniques [14].

Différents niveau de BDD existent suivant le type de tests :

- **BDD spec-level** : Les tests sont écrits suivant la spécification de l'application. Ils correspondent aux tests unitaires. Comme la partie critique du projet sbl-creator est la réception et transmission des données au serveur, ce type de test est bien adapté.
- **BDD story-level** : Les tests sont écrit suivant les attentent des utilisateurs (besoins). Ils correspondent aux tests d'acceptations. Comme, par exemple, dans le projet studypress-sbl la communication avec le serveur est jugée la plus critique, donc les tests sur ce niveau sont chois.

IV-2 Résultats

a- Back office

Comme précisé précédemment, l'interface de WordPress est divisée en deux parties : front office et back office. Le composant développé utilise la partie back office comme interface de création et de configuration des scénarios.

Le composant ajoute un sous-menu nommé « Scénarios » dans le menu de l'extension *StudyPress*. En cliquant sur ce sous-menu, l'application sbl-creator se charge dans le navigateur et affiche la liste des scénarios.

---

[18] www.phpspec.net/
[19] http://docs.behat.org
[20] https://karma-runner.github.io/
[21] http://jasmine.github.io/

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

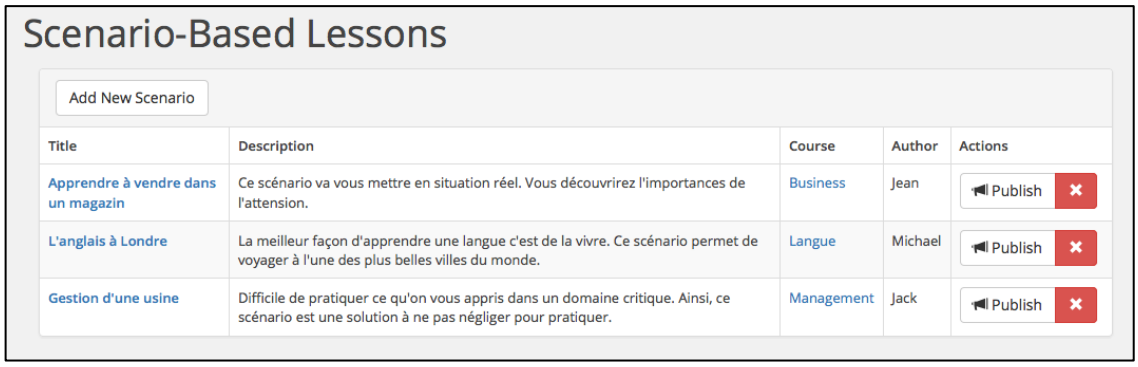

Figure IV.4 : Liste des scénarios

La liste contient de nombreuses informations, comme le titre du scénario, la description, etc. à droite se trouves des boutons permettant de publier le scénario dans un article du site *WordPress* et la suppression d'un scénario. En haut de la liste se trouve le bouton pour la création d'un nouveau scénario. En cliquant sur ce dernier, un élément modal s'affiche contenant un formulaire de saisie et des boutons pour l'enregistrement.

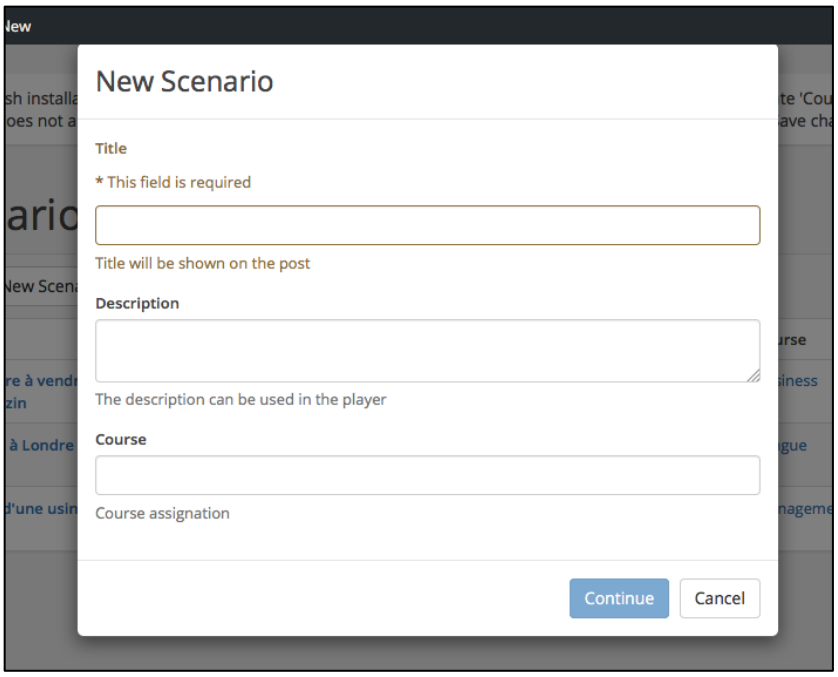

Figure IV.5 : Création d'un nouveau scénario

Les titre d'un scénario dans liste est un lien renvoyant vers la page d'édition du scénario.

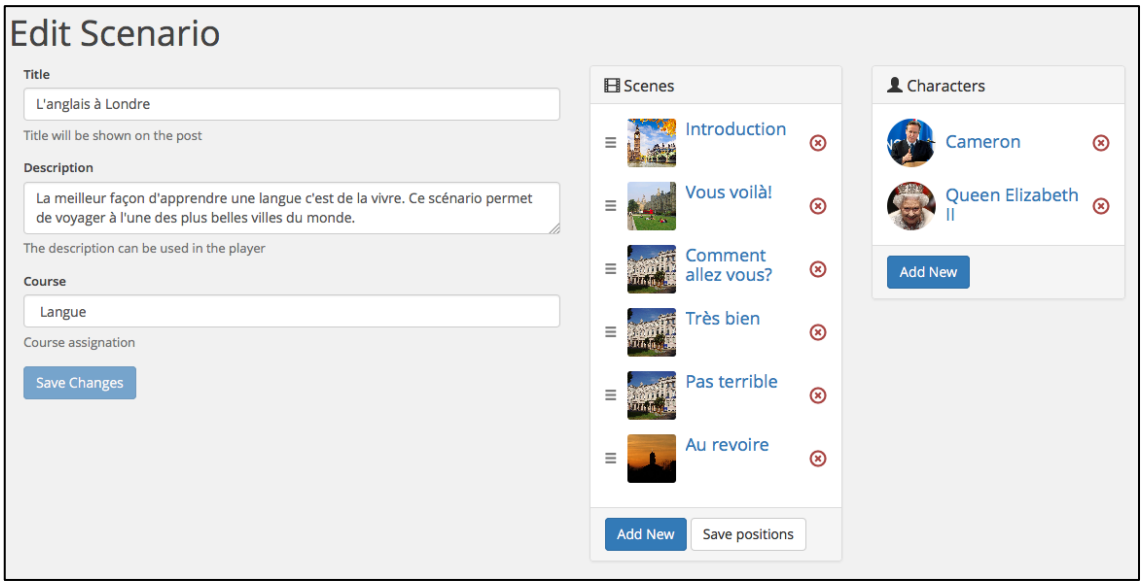

Figure IV.6 : Edition d'un scénario

Dans l'édition d'un scénario, en plus du formulaire préalablement saisi lors de la création, de listes sont affichées. La première contient la liste des scènes contenues dans le scénario. Cette liste affiche le titre et l'arrière-plan de chaque scène. Il est possible de repositionner la scène en faisant du glisser-déposer à partir de l'icône représentant trois barres. Quant à la deuxième liste, elle contient les personnages disponibles dans le scénario. Les personnages sont identifiés par leurs noms et avatars.

Comme pour les scénarios, lors de la création d'une scène, la création d'un personnage ou l'édition d'un personnage, un élément modal s'affiche contenant un formulaire de saisie.

L'accès à l'édition d'une scène se fait par un clic sur le titre de la scène dans la liste correspondante. La page de l'édition de scène s'ouvre alors. Elle regroupe plusieurs informations et configuration relatives à la scène comme le dialogue, le branchement, etc.

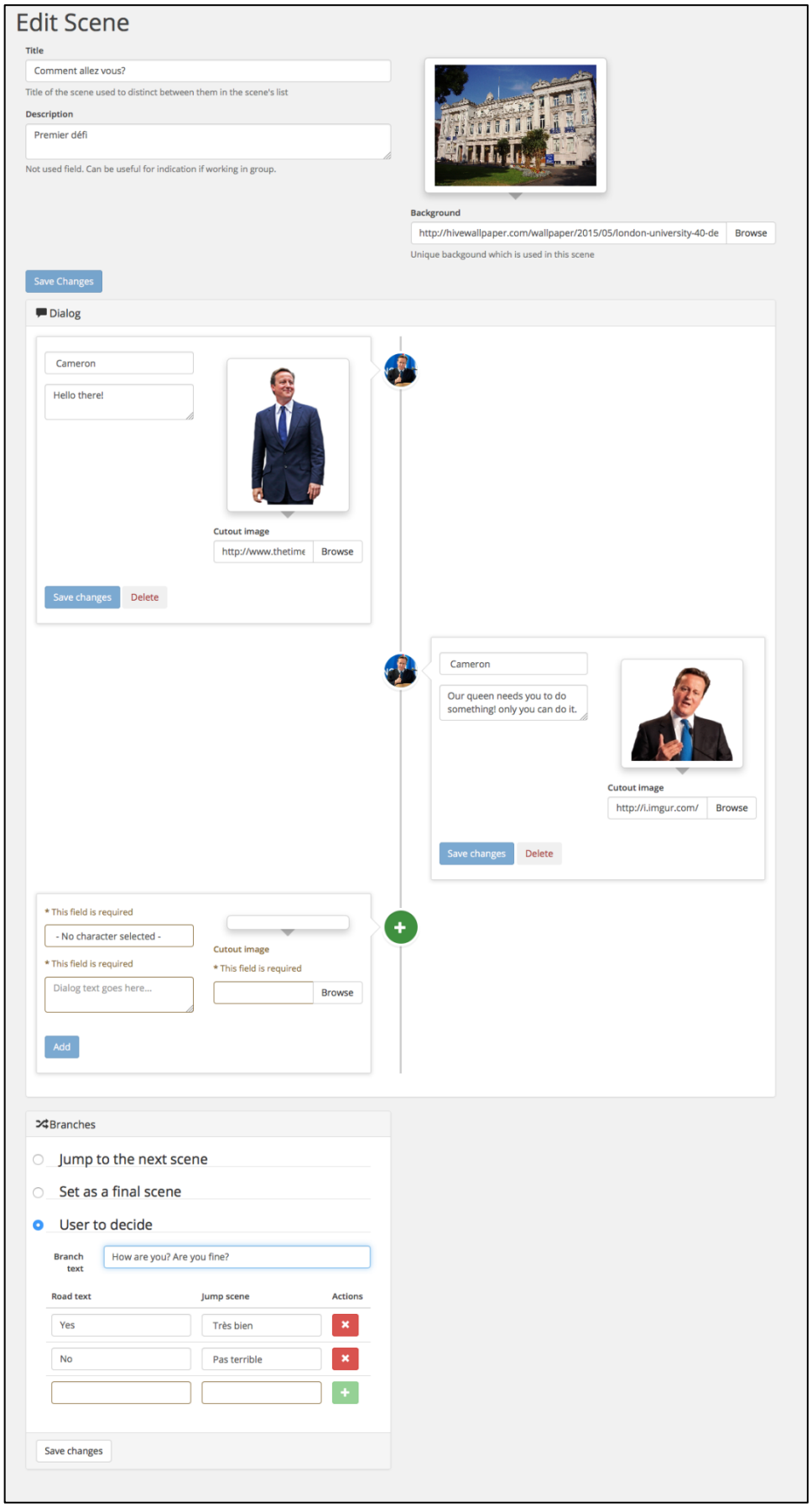

Figure IV.7 : Page d'édition d'une scène

La disposition des éléments et formulaires de la page fait en sorte de diviser la page verticalement en trois parties. La première partie regroupe les informations de la scène comme le titre, la description, l'arrière-plan. La deuxième partie concerne le dialogue dans la scène. Le dialogue est affiché sous forme de Timeline permettant une clarté et une meilleure vision globale. La dernière partie concerne le comportement du scénario à la fin de la scène. Elle permet notamment d'ajouter un branchement, ou de faire en sorte que la scène est une scène finale, ou de continuer la lecture scénario en allant vers la scène suivante après la fin de celle-ci.

b- Front office

La publication de scénario permet de créer un nouvel article dans *WordPress*. Cet article a comme titre le titre du scénario. Le contenu de l'article est les l'application créée dans le projet sbl-player. Cette application permet de récupérer le scénario et de l'afficher sous forme de présentation.

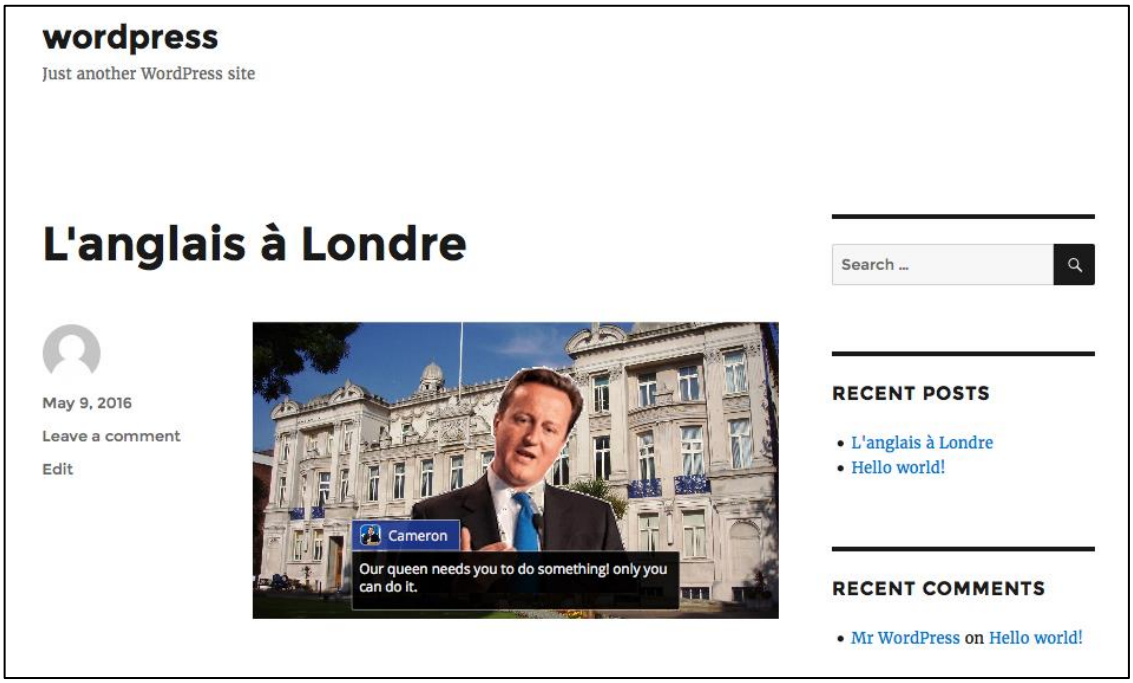

Figure IV.8 : Scénario vu par l'apprenant

Figure IV.9 : Branchement vu par l'apprenant

V- Conclusion et perspective

V-1 Conclusion

Ce travail a permis d'ajouter le support des scénarios dans l'extension *StudyPress*. La navigation dans l'outil de création est intuitive. Il est même possible de créer des scénarios relativement complexes. L'auteur du scénario n'a pas à se soucier des détails comme l'interface du scénario, l'ergonomie, etc., ce qui lui permet ainsi de se concentrer sur le contenu et ainsi laisser l'extension gérer les détails. Le code écrit a lui aussi été travaillé et est organisé dans 128 fichiers dont près de 5300 lignes de code.

Différentes approches et méthodes de développement ont permis d'ajouter de nouvelles fonctionnalités à l'extension *StudyPress* tout en adoptant différentes pratiques. Cependant, l'intégration du composant développé a été complexe. En effet, Les technologies et framework utilisés ne sont pas forcément adaptés pour le développement d'une extension *WordPress*. Certains bugs ont surgis pendant le développement, notamment avec le framework Herbert. Il est à noter que 4 bugs rencontrés sont corrigés et leurs correctifs ont été publiés sur le dépôt officiel du framework.

Enfin, toutes les approches et outils utilisés citées précédemment (Scrum, BDD, Git, etc.) ont pour but d'améliorer le processus de développement mais aussi à éviter les bugs que peut rencontrer l'utilisateur. Ainsi, le développement est sûr. Malgré le temps de s'adapter au maniement de certains outils, le projet StudyPress SBL a atteint ses objectifs.

V-2 Amélioration future

a- Fonctionnalités

L'approche suivie durant le projet est une approche agile. De ce fait, les exigences changes au fur à mesure que le projet se développe. Ainsi, certaines fonctionnalités manques au système pour répondre aux nouvelles exigences. A titre d'exemple, ces exigences sont en cours d'étude :

Rapport-Gratuit.com

Surveiller les progrès des apprenants

Un formateur peut voir des statistiques sur les choix des apprenants connectés et des apprenants non-connectés. Ces statistiques lui permettent de voir notamment le nombre de fois qu'un apprenant a choisi tel réponse dans le scénario ou combien d'apprenants ont commencé à lire le scénario.

Visualiser le graphe des scènes d'un scénario

Comme pour *Twine*, le système doit permettre au formateur de visualiser le scénario sous forme d'un graphe où chaque scène est représentée par un nœud. Ces nœuds sont reliés entre eux selon l'ordonnancement des scènes, et par ailleurs, comme une scène peut avoir un branchement, plusieurs nœuds peuvent être reliés à cette dernière.

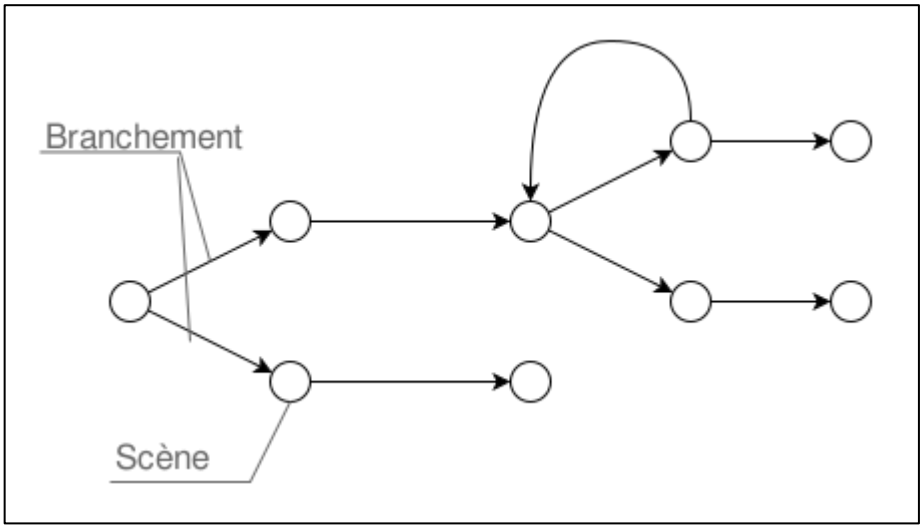

Figure V.1 : Visualisation d'un scénario sous forme d'un graphe

Pour la réalisation d'une telle fonctionnalité, la bibliothèque *D3.js* peut être choisie notamment pour sa flexibilité dans la manipulation des données. Cette flexibilité permet de créer facilement des graphes adaptés en utilisant *HTML*, *SVG* et *CSS*. De plus, la bibliothèque est compatible avec *AngularJS*.

Ajouter du son dans un dialogue

Le formateur peut mettre du son dans une scène et associer chaque message dans un dialogue à un son.

Il est possible dans *Reveal.js* d'ajouter du son grâce à l'extension *Audio Slideshow*. Cette extension permet de lire du son à chaque diapositive.

Mettre une vidéo dans un dialogue

Le formateur doit avoir la possibilité de mettre une vidéo à la place de l'image d'un personnage dans un dialogue. *Reveal.js* permet de mettre des vidéos dans une diapositive.

b- Processus de développement

Il est aussi intéressent à noter que la documentation sur l'apprentissage basé sur un scénario est rare. N'ayant pas d'experts dans le domaine, l'approche de recueille des besoins utilisée est jugée mauvaise pour le développement de tels produits. Ceci peut engendrer le risque de ne pas répondre aux besoins des utilisateurs finaux. A rappeler que le composant développé s'inscrit dans le projet *StudyPress* qui vise le grand public. Une des solutions est d'utiliser les principes *Lean Startup*.

Le *Lean Startup* est une approche qui repose sur l'expérimentation, la vérification et validation d'un concept. Elle permet de réduire le cycle de commercialisation des produits mais surtout à obtenir des retours de la part des utilisateurs et ainsi mesurer la satisfaction des utilisateurs.

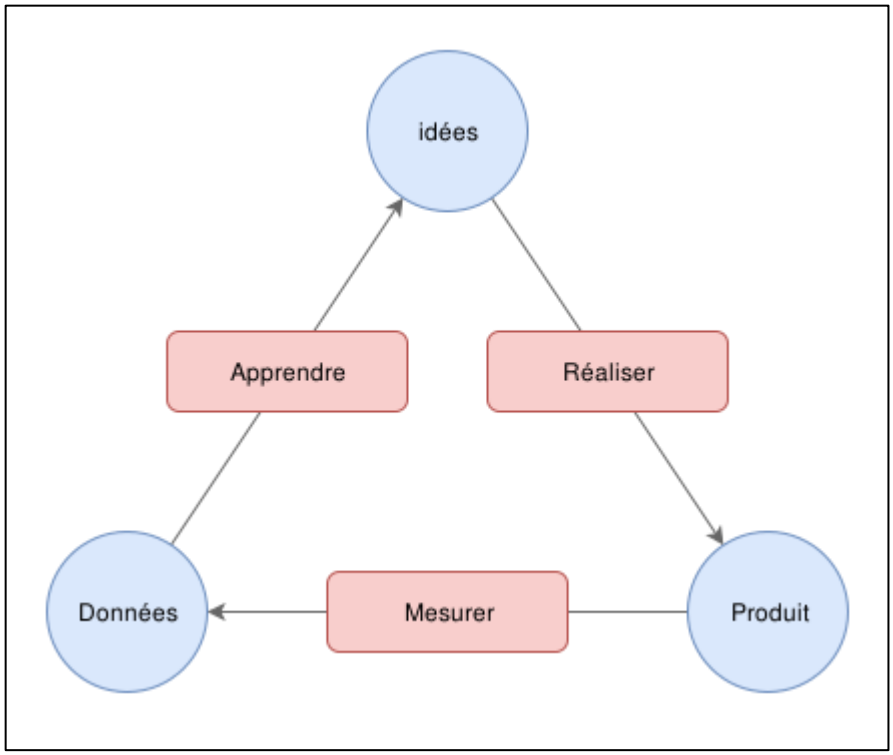

Figure V.2 : Processus Lean Startup

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

La démarche à suivre consiste à trouver des hypothèses (idées) comme solution à un problème, et d'expérimenter ces idées en réalisant et mettant entre les mains des utilisateurs un produit minimum viable. C'est selon des mesures basées sur le retour des utilisateurs que l'hypothèse est validée ou réfutée.

Bibliographie

[1] Sam S. Adkins, 2015 International Learning Technology Investment Patterns, whitepaper, January 2016.

[2] E-learning : Définition simple et facile du dictionnaire, [En ligne]. Disponible: http://www.linternaute.com/dictionnaire/fr/definition/e-learning/ consulté le 12 juin 2016.

[3] Usage Statistics and Market Share of Content Management Systems for Websites, June 2016, [En ligne]. Disponible: https://w3techs.com/technologies/overview/content_management/all consulté le 12 juin 2016.

[4] StudyPress — WordPress Plugins, [En ligne]. Disponible: https://wordpress.org/plugins/studypress/stats/ consulté le 12 juin 2016.

[5] SCENARIO-BASED LEARNING, mars 2016, [En ligne]. Disponible: https://quality4digitallearning.org/wp-content/uploads/2016/03/Scenario-based-learning.pdf consulté le 12 juin 2016.

[6] Beatrice Ghirardini, et al., Méthodologies Pour le développement de cours e-learning: Un guide pour concevoir et élaborer des Cours d'apprentissage numérique, [En ligne]. Disponible: http://www.fao.org/docrep/015/i2516f/i2516f.pdf consulté le 12 juin 2016.

[7] Ruth C. Clark and Richard E. Mayer, Scenario-based e-Learning: Evidence-Based Guidelines for Online Workforce Learning, Livre, ISBN: 978-1118127254, Pfeiffer, 2012.

[8] Ken Schwaber et Jeff Sutherland, Le Guide Scrum - Le guide définitif de Scrum: les règles du jeu, juillet 2013. [En ligne]. Disponible: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-FR.pdf consulté le 12 juin 2016.

[9] Meryem. Bendella et Salim. M. Saidi, Réalisation d'un outil auteur interactif et social sous WordPress, projet de fin d'étude, université de Tlemcen, 2015.

[10] Stack Overflow Developer Survey 2016 Results, 2016. [En ligne]. Disponible: http://stackoverflow.com/research/developer-survey-2016#technology-most-loved-dreaded-and-wanted consulté le 12 juin 2016.

[11] Architecture trois tiers, 11 mai 2016. [En ligne]. Disponible: https://fr.wikipedia.org/wiki/Architecture_trois_tiers consulté le 12 juin 2016.

[12] Representational state transfer, 11 juin 2016. [En ligne]. Disponible: https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=724768726 consulté le 12 juin 2016.

[13] Clément Héliou, Pérennisez votre métier avec l'architecture hexagonale, 16 mars 2016. [En ligne]. Disponible: http://blog.xebia.fr/2016/03/16/perennisez-votre-metier-avec-larchitecture-hexagonale/ consulté le 12 juin 2016.

[14] Behavior driven development, 09 avril 2016. [En ligne]. Disponible: http://fr.wikipedia.org/w/index.php?title=Behavior_driven_development&oldid=125134310 consulté le 12 juin 2016.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Glossaire

Terme	Définition
AJAX	Asynchronous JavaScript and XML. Ensemble de technique utilisés pour que le client web peut modifier la page web sans la recharger.
BDD	Behavior Driven Development. Processus de développement qui se base sur le TDD. La particularité de ce processus est de créer les tests selon les exigences requises dans le cahier des charges.
DOM	Document Object Model. Convension pour la représentation et l'interaction avec des objets dans un document HTML, XHTML et XML.
HTML	HyperText Markup Language. Standard utilisé pour créer des pages web.
REST	Representational State Transfer. Style architectural pour la communication client/serveur. Il permet au client d'accéder aux ressources du serveur de manière performante et simple.
Sass	Syntactically Awesome Stylesheet. Langage de feuille de style qui peut être interprété en CSS.
SBL	Scenario Based-Learning. Méthode d'apprentissage qui consiste à créer des scénarios intéractifs.
SVG	Scalable Vector Graphics. Format de représentation d'image basé sur XML.
SWF	Small Web Format. Extension de fichier Shockwave Flash créé par l'entreprise Macromedia. Le fichier peut contenir des vidéos, des animations vectorielles, du son, etc.
TDD	Test Driven Development. Processus de développement de logiciels. Il consiste à répéter le cycle : écrire le code de test ; écrire le code de l'application ; réusiner le code.
WordPress	Un système de gestion de contenu souvent utilisé comme moteur de blog.

Résumé

L'apprentissage basé sur les scénarios est une méthode d'apprentissage interactive. Elle consiste à fournir aux apprenants un contexte qui simule une situation réelle. A travers ce contexte, l'apprenant réunit les informations et répond aux défis posés. De cette façon, il est possible de vérifier les connaissances acquises et de permettre à la leçon de s'adapter à l'apprenant.

Dans ce travail, une solution est proposée pour intégrer un outil pour la création et la publication de scénarios interactifs dans, l'extension *WordPress*, *StudyPress* tout en respectant les principes du génie logiciel : l'extensibilité, la modularité, la réutilisabilité et la testabilité.

Mots clés : e-learning, scénario, genie logiciel, WordPress.

ملخص

التعليم الالكتروني الموجه بالسيناريوهات, طريقة في التدريب تعتمد على تزويد المتعلم بمجموعة من المعطيات الواقعية. و من ثم جعله يواجه مواقف و تحديات معينة, لاختبار مدى ادراكه لما تعلمه وبناء على القرارات التي يتخذها, يتغير مجرى الدرس, ليتناسب مع القدرة الاستيعابية والادراكية له.

المشروع تمحور حول إيجاد حل يجعل من المفاهيم السابقة الذكر قيد التنفيذ مع الحرص على احترام و تطبيق مبادئ هندسة البرمجيات المتمثلة في الـextensibility modularity reusability testability

كلمات مفتاحية : التعليم الاكتروني, سيناريو, هندسة البرمجيات.

Abstract

Scenario-based learning is an interactive training approach. It involves students in a context where it tries to simulate a real world one. From this context, the student gathers the information and gives feedback by choosing a response to a challenge. This way, it is possible to verify the acquired knowledge and make the lesson adaptable.

In this work, a solution is proposed as an integrated tool for the scenario-based learning to the *WordPress* extension *StudyPress*. The tool respects the software engineering principles: extensibility, modularity, reusability and testability.

Key words: e-learning, scenario, software engineering, WordPress.