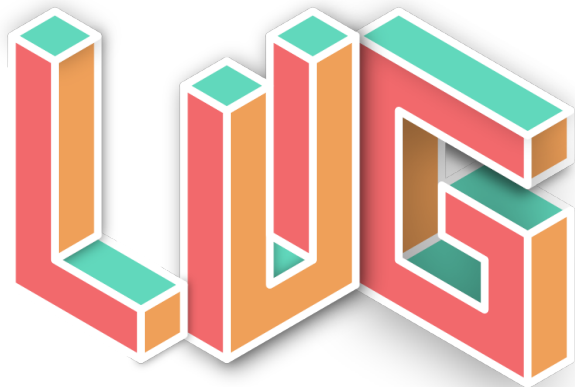


Document de synthèse

PLD SMART



HEX AFTER WORK

Mai 2019

Hexanôme 4412 :

Arthur BELLEMIN

Etienne DELAHAYE

Lynn GHANDOUR

Loïc CASTELLON

Grégoire GENTIL

Hector MISSIAEN



1. Description du projet	2
2. Architecture générale	2
2.1. Architecture applicative	2
2.2. Modèle de données	3
2.3. Technologies utilisées	4
3. Fonctionnalités	5
3.1. Application “Professeur”	5
3.1.1. Authentification	5
3.1.1.1. S’inscrire	5
3.1.1.2. Se connecter	5
3.1.2. Gérer ses classes	5
3.1.2.1. Créer une classe	5
3.1.2.2. Modifier la classe (ajouter des élèves)	6
3.1.2.3. Supprimer des classes	6
3.1.2.4. Assigner un parcours à une classe	6
3.1.2.5. Visualiser ses classes	6
3.1.3. Créer un parcours	6
3.1.3.1. Créer un tout nouveau parcours	6
3.1.3.2. Créer un POI	6
3.1.3.3. Créer un quizz pour valider une étape	7
3.1.3.4. Visualiser ses parcours	7
3.1.3.5. Visualiser tous les parcours publiés	7
3.1.3.6. Créer un parcours à partir d’un parcours existant	7
3.1.4. Suivre ses élèves	7
3.2. Application “Élève”	7
3.2.1. Authentification	7
3.2.2. Suivre un parcours	7
3.2.2.1. Voir ses parcours	7
3.2.2.2. Visualiser un parcours	8
3.2.2.3. Valider une étape	8

1. Description du projet

Aujourd'hui, les cours enseignés au lycée et au collège restent très théoriques, les élèves n'apprennent qu'à travers leurs manuels scolaires. Or, Lyon offre un grand patrimoine culturel, et il suffirait que les élèves sortent la tête de leurs livres pour découvrir leur ville et profiter de ce qu'elle a à leur offrir.

Avec Lug, les professeurs pourront donner des devoirs différents : proposer des parcours dans Lyon et ses alentours afin d'approfondir ou de découvrir le programme de façon originale. Les élèves pourront les réaliser seuls ou en groupe et devront valider chaque étape du parcours à l'aide de l'application Lug.

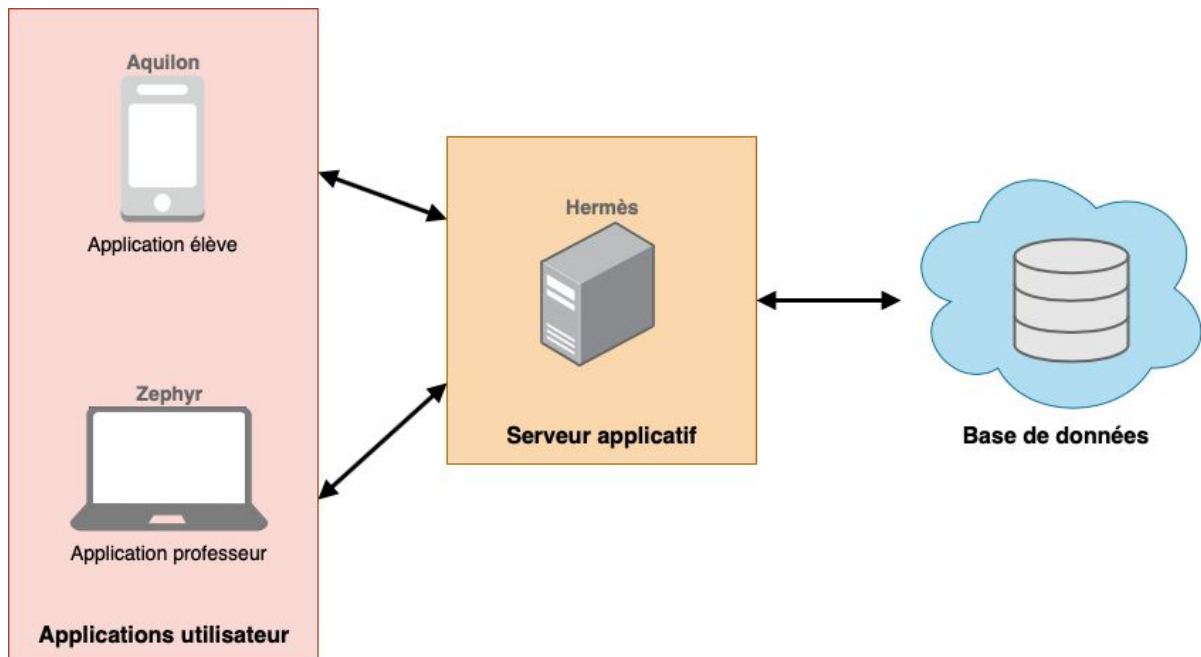
Ainsi, la ville de Lyon pourra être employée comme lieu d'apprentissage dans le but d'intéresser davantage les élèves au programme scolaire et de susciter leur curiosité.

2. Architecture générale

Lug est composée d'une application web utilisateur, d'une application mobile utilisateur, d'une base de données et d'un serveur applicatif (back-end).

L'application web est destinée aux professeurs, tandis que l'application mobile est utilisée par les élèves.

2.1. Architecture applicative



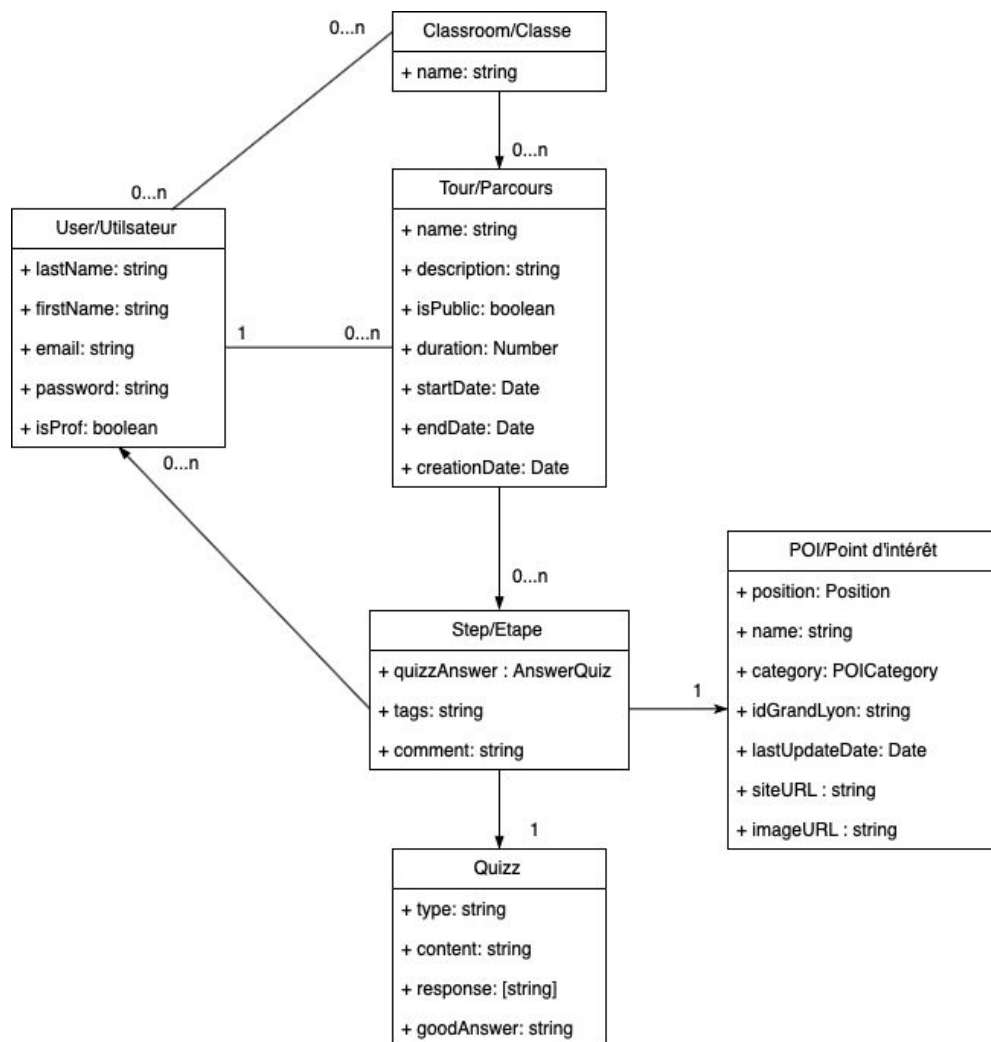


Les deux interfaces des applications utilisateurs communiquent avec le même serveur applicatif. La base de données est externalisée dans le cloud sur des serveurs distants. Nous avons nommé ces trois composants :

- Hermès : le serveur applicatif
- Zephyr : Application web (professeur)
- Aquilon : Application mobile (élève)

2.2. Modèle de données

Nous avons plusieurs objets représentant les données : Utilisateur, Classe, Parcours, Etape, Point d'intérêt et Quizz. Un utilisateur peut être un élève ou un professeur. Nous avons fait ce choix pour simplifier la base de données étant donné que les attributs sont très similaires. Cependant, lorsqu'un utilisateur est un élève, nous avons choisi comme convention "qu'il ne possède pas de parcours" : pour retrouver les parcours qui lui sont accessibles, il faut passer par sa classe.





2.3. Technologies utilisées

Pour réaliser l'application web (front-end web) nous avons utilisé le framework ReactJS, avec Redux pour gérer les états, TypeScript pour étendre les capacités du JavaScript, WebPack pour assembler et compacter le JavaScript en production, Axios pour les requêtes AJAX et la librairie Grommet pour disposer de composants adaptatifs (responsive).

Pour développer l'application mobile (front-end mobile), nous avons choisi de faire du React Native, avec Expo pour faciliter le développement et la transcompilation, Redux pour les états et Axios pour les requête AJAX.

Enfin, pour mettre en place le service applicatif (back-end) nous avons utilisé Node JS pour interpréter le JavaScript, Express pour construire une application web, MongoDB et Mongoose respectivement pour le base de données et l'ORM permettant de manipuler nos modèles.

Nous avons choisi ces technologies car elles sont d'actualités, testées et proposent un langage cohérent de la base de données jusqu'au front-end : le JavaScript. Elles ont également toutes une excellente documentation.

NodeJS est un exécuteur JavaScript conçu pour supporter une forte montée de charge grâce à ses opérations d'entrées/sorties asynchrone. Utilisé par des sites à grosses fréquentations comme IBM, Netflix ou encore LinkedIn, NodeJS a déjà fait ses preuves en termes de performances et de stabilité.

Express est un framework abstrayant les concepts bas niveau de NodeJS (comme la gestion des requêtes HTTP à la main) et facilitant la mise en place d'*endpoint* pour notre API.

React est un framework développé et documenté par Facebook. Il propose une architecture simple permettant de construire des applications web monopage en utilisant des concepts puissant comme l'héritage ou encore un système de gabarits.

React Native, aussi développé par Facebook, et propose une utilisation identique à React remplaçant les composants web par des composants natifs. Ainsi, une balise `<p>` dans React devient dans React Native une balise `<Text>` qui sera traduite en Java ou en Objective-C selon la plateforme cible (Android ou iOS). Le passage de React à React Native est ainsi très facile puisque les concepts sont les mêmes.

Enfin, nous avons choisi Mongo pour ses capacités de répartition des charges et pour le format JSON des documents stockés dans la base. Aussi, c'est la base de données NoSQL la plus populaire et nous avons envie de l'utiliser dans une application réelle.



3. Fonctionnalités

Nous avons deux applications bien distinctes pour le professeur (application web) et pour les élèves (application mobile). Chacune d'entre elle a des fonctionnalités qui lui sont propres.

3.1. Application “Professeur”

3.1.1. Authentification

3.1.1.1. S'inscrire

Un professeur peut se créer un compte sur LUG en renseignant son nom, son prénom, son email et un mot de passe.

Si l'email renseigné est le même que celui d'un autre utilisateur, ou qu'un champ n'est pas rempli, une erreur est renvoyée et l'utilisateur est notifié du problème sur l'interface de l'application.

3.1.1.2. Se connecter

Un professeur peut se connecter à l'application en utilisant son email et son mot de passe renseignés lors de l'inscription. Nous avons choisi l'identification par mail étant donné que nous sommes sûrs qu'il est unique à chaque utilisateur.

Dans le cas où l'email ou le mot de passe sont incorrects, ou qu'un champ n'est pas renseigné, une erreur s'affiche sur l'interface web.

3.1.2. Gérer ses classes

3.1.2.1. Créer une classe

Un professeur crée une classe en indiquant le nom de cette classe.

Si une classe avec le même nom existe déjà, une erreur est renvoyée.

Dans le cas contraire, il peut alors ajouter des élèves directement à la classe. Un élève ne peut exister qu'une seule fois par classe. Donc si un élève ajouté à une classe, (plus précisément son email) existe déjà, une erreur est affichée. Lorsque l'élève est ajouté à une classe il est créé, s'il n'a pas déjà de compte, et un mot de passe automatiquement généré lui est attribué.



3.1.2.2. Modifier la classe (ajouter des élèves)

L'utilisateur a la possibilité d'ajouter de nouveaux élèves à une classe existante. Là aussi, comme à la création de la classe, si le mail d'un des élèves renseignés existe déjà, une erreur est affichée.

3.1.2.3. Supprimer des classes

Le professeur peut supprimer une classe qu'il a créé.

3.1.2.4. Assigner un parcours à une classe

L'utilisateur peut assigner un parcours qu'il a créé à une de ses classes.

3.1.2.5. Visualiser ses classes

L'utilisateur peut voir la liste de ses classes et leurs informations : son nom, le nombre d'élèves, le nom des élèves, le nombre et le nom de parcours.

3.1.3. Créer un parcours

3.1.3.1. Créer un tout nouveau parcours

Le professeur peut créer un parcours : il devra donner un nom, une description et une date limite pour réaliser le parcours. Si un des champs n'est pas rempli, une erreur s'affiche.

Il peut ensuite choisir les point d'intérêt (POI) constituant les étapes du parcours. Pour chaque étape du parcours il aura la possibilité d'ajouter un commentaire et un quizz que l'élève devra faire pour valider l'étape.

Pour sélectionner les points d'intérêt (POI), le professeur peut les visualiser sur une carte, les voir sur une liste déroulante ou encore les rechercher par nom. Il peut également créer son propre point d'intérêt. Celui-ci sera alors ajouté à la base de données de LUG. Nous avons choisi une dimension collaborative.

Il est envisagé que le professeur ait le choix de rendre son parcours public (visible par les autres professeurs) ou non.

3.1.3.2. Créer un POI

L'utilisateur peut créer un POI en renseignant son nom, sa latitude, sa longitude. Si une des information manque, une erreur est retournée. Il peut également ajouter des informations complémentaires : un site internet, une image, une catégorie.



3.1.3.3. Créer un quizz pour valider une étape

Pour chaque étape du parcours, le professeur peut ajouter un quizz. Il devra indiquer une question et quatre réponses différentes dont la bonne réponse.

3.1.3.4. Visualiser ses parcours

Le professeur peut voir tous les parcours qu'il a créé.

3.1.3.5. Visualiser tous les parcours publiés

De la même façon, l'utilisateur pourra voir les parcours "publics". Il verra uniquement les étapes sur une carte.

3.1.3.6. Créer un parcours à partir d'un parcours existant

En visualisant les parcours publics, un professeur peut copier un parcours existant afin de l'utiliser comme base pour créer son propre parcours.

3.1.4. Suivre ses élèves

- Visualiser le nombre d'élèves ayant validé le parcours

3.2. Application "Élève"

3.2.1. Authentification

Un élève peut se connecter à l'application en utilisant l'email utilisé par le professeur quand il l'a ajouté à une classe et le mot de passe généré automatiquement qui lui a été communiqué en classe par son professeur. Dans le cas où l'email ou le mot de passe sont incorrects, ou qu'un champ n'est pas renseigné, une erreur s'affiche sur l'interface mobile.

3.2.2. Suivre un parcours

3.2.2.1. Voir ses parcours

Dès que l'élève rentre sur l'application, l'élève peut voir les parcours qui ont été attribué à sa classe.

3.2.2.2. Visualiser un parcours

L'élève pourra voir chaque étape constituant le parcours : sa position sur une carte, le nom ainsi qu'une description.



Il peut naviguer d'une étape à l'autre.

3.2.2.3. Valider une étape

En cliquant sur une étape, l'élève peut la valider. La validation se base sur la localisation GPS. Si l'élève est trop loin du point d'intérêt, un message le signalant s'affiche.

Il devra également, sur certaines étapes, valider en répondant à un quizz. Il ne sait pas si la réponse est juste ou fausse après qu'il ait répondu.