



**TUTRICE DE STAGE :**

**Mme.**

**Martine GAUTIER**

**RESPONSABLE DE STAGE :**

**MM.**

**Bruno GUILLAUME**

**Yves LEPAGE**

**Jean LIEBER**

**Emmanuel NAUER**

**Hue-Nam LY**

**Rapport De Stage**

## **Développer une infrastructure Web pour un système de correction de phrases en français à partir de cas**

**Du 9 avril au 6 juin 2018**

**Année Universitaire 2017-2018**



**Licence 3 informatique**

## **SOMMAIRE**

<b>REMERCIEMENTS</b>	<b>3</b>
<b>INTRODUCTION</b>	<b>4</b>
<b>PRESENTATION ENTREPRISE</b>	<b>5</b>
<b>I. BASE DE DONÉES</b>	<b>6</b>
1. Présentation de la base de données	6
2. Génération du script de la création et de l'insertion	7
<b>II. CONCEPTION</b>	
1. Diagramme contexte statique	8
2. Diagramme de cas d'utilisation	8
3. Diagramme d'activité	9
4. Diagramme de séquences	11
<b>III. ENVIRONNEMENT DE DEVELOPPEMENT</b>	
1. MySQL et phpMyAdmin	13
2. Bootstrap	13
3. SublimeText	14
4. Balsamiq	14
5. Wampserver	15
6. GIT	15
<b>IV. INTERFACE WEB</b>	
1. Correction un site multilingue	16
2. La maquette du site	17
3. Interface d'utilisateur	18
4. Interface d'administrateur	19
<b>CONCLUSION</b>	<b>20</b>
<b>ANNEXE</b>	<b>21</b>
<b>SITOGRAFIE</b>	<b>21</b>

## **REMERCIEMENTS**

Je tiens à présenter mes sincères remerciements à :

Messieurs Jean LIEBER, Emmanuelle NAUER, Yves LEPAGE, Bruno GUILLAUME pour m'avoir accepté dans ce stage, ainsi que leurs explications et leurs conseils durant mes deux mois de stage.

Madame Martine GAUTHIER pour ses enseignements pendant mes années universitaires et son soutien pendant mon stage.

Mes collaborateurs : Messieurs André GIANG, Damien LEVY pour leur aide et leurs idées sur le design du site web.

## Introduction :

De nos jours, il existe plusieurs systèmes qui corrigent des fautes d'orthographe ou la grammaire automatiquement. Mais que se passe-t-il si on rentrait une phrase syntaxiquement incorrecte ? Est-ce que ces systèmes nous donnent une bonne ou mauvaise solution ?

Dans notre cas, on cherche à corriger ce genre de problème et fournir à l'utilisateur une correction acceptable en utilisant le raisonnement à partir de cas.

Le raisonnement à partir de cas (RàPC) : consiste donc à s'appuyer sur des couples (problème, solution) où problème est une phrase incorrecte qui est corrigée en solution pour proposer une correction à une nouvelle phrase cible. Par exemple :

*Problème : Je aimer les pommes.*

*Solution : J'aime les pommes.*

La phrase cible sera proposée par les utilisateurs via une interface web. Cela déclenchera le système de remémoration puis d'adaptation et la solution sera affichée sur l'interface web. Par exemple :

*Cible : Je manger des tomates.*

*Solution : Je mange des tomates.*

Si le résultat est accepté par l'utilisateur il forme alors un couple (cible, solution) qui sera insérée dans la base de cas pour être utilisable ultérieurement en tant que nouveau cas.

La base de cas minimale est alimentée de manière manuelle par M. Levy. Ensuite elle sera complétée de manière semi-automatique à partir du corpus WiCoPaCo qui est un historique de modification de pages sous Wikipédia. Puis, elle sera améliorée au fur et à mesure de l'utilisation. La remémoration et l'adaptation sont des algorithmes développés par M. Giang en Python. En s'appuyant sur le RàPC pour corriger un problème et donner une solution.

Dans ce stage, je dois créer une base de données qui stocke les cas et développer une interface web multilingue qui donnera la possibilité aux utilisateurs de rentrer une phrase incorrecte en français ou anglais et affichera la solution aux utilisateurs.

Tout d'abord, je vais présenter la création d'une base de données, la conception, suivi de l'environnement de développement, et enfin le développement de l'interface web Corrector.

## **Présentation de l'entreprise :**

Le stage s'est déroulé au Loria<sup>1</sup>, Laboratoire lorrain de recherche en informatique et ses applications situé à Nancy. Ce laboratoire est une Unité Mixte de Recherche (UMR), il est commun à plusieurs établissements le CNRS, l'université de Lorraine et l'Inria. Créé depuis 1997 son domaine est la recherche fondamentale et appliquée en sciences informatiques.

Cet UMR possède un effectif total de plus de 400 personnes et est composée de 28 équipes répartis en 5 départements, dont 15 sont communes avec Inria. Le Loria est un des plus grands laboratoires de la région lorraine.

Notre stage s'est déroulé au sein du département 4 TALC<sup>2</sup> (traitement automatique des langues et des connaissances) qui comme son nom l'indique est spécialisé dans 3 domaines, la recherche et l'étude :

- Des langues qui vont de l'analyse à sa reconnaissance et d'autres secteurs encore.
- Des connaissances ou plus particulièrement sa représentation et les méthodes d'apprentissage et la classification.
- Des documents la reconnaissance de formes, la reconnaissance de symboles et la modélisation de l'écriture manuscrite.

Nous étions dans l'équipe Orpailleur qui compte 12 membres permanents ainsi que de nombreux ingénieurs, doctorants et étudiants en doctorat.

# I. BASE DE DONNÉES

## 1. Présentation de la base de données

Pour ce stage, le type de base de données n'était pas imposé. J'ai choisi le mySQL, car c'est celui que j'ai utilisé pendant mon année universitaire.

La principale difficulté est que cette base doit être adaptée aux travaux de M. Giang et de M. Levy. Ainsi, Après plusieurs réunions, on a décidé que cette base est de la forme présente en figure 1. Le MCD a été créé par un site web mocodo<sup>3</sup>.

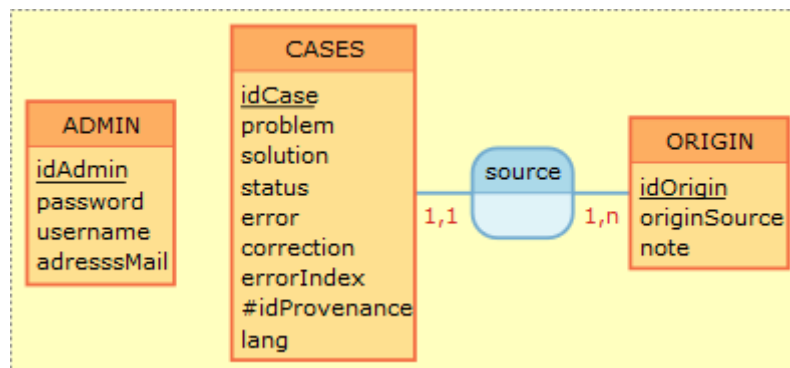


Fig.1 : Base de données du système Corrector.

Notre base de données contient trois relations :

La relation CASES est composée une clé primaire idCase, ainsi que deux colonnes « problem » et « solution », ces deux dernières forment alors le couple (problème, solution) constituant un cas. La colonne « status » est de forme « correct, incorrect, en attente » pour indiquer si la solution est correcte. Les deux colonnes suivantes sont « erreur et correction » elles indiquent quel(s) le(s) mot(s) a(ont) été corrigé(s). « errorIndex » donne la position du mots erroné dans la phrase. « lang » pour l'instant notre projet ne comporte que du français et de l'anglais mais cela peut évoluer dans le futur.

La relation ORIGIN (ou provenance) stocke les informations de la provenance du couple (problème, solution), c'est-à-dire de quelle source Mr Levy a trouvé ce genre de cas. La clé primaire est « idOrigin » et « originSource » est le nom de la source (Par exemple : WiCoPaCo un site de correction et modification de la page wikipedia).

La relation ADMIN contient les informations nécessaires à l'administrateur.

## 2. Génération du script de création et d'insertion

### a. Création

Le code de création d'une base est très standard (voir figure 2) :

```
CREATE TABLE IF NOT EXISTS CASES(  
    idCase int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    problem varchar(200) CHARACTER SET utf8,  
    solution varchar(200) CHARACTER SET utf8,  
    status varchar(200) CHARACTER SET utf8,  
    error varchar(200) CHARACTER SET utf8,  
    correction varchar(200) CHARACTER SET utf8,  
    errorIndex varchar(200) CHARACTER SET utf8,  
    idProvenance int NOT NULL ,  
    lang varchar(10) CHARACTER SET utf8,  
  
    CONSTRAINT fk__provenance FOREIGN KEY (idProvenance) REFERENCES ORIGIN(idOrigin)) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Fig.2 Code de création de la relation CASES.

Le seul problème que j'ai rencontré ici est la partie « ENGINE=InnoDB ». Il faut préciser le type ENGINE si non le système va prendre celui par défaut : MYISAM. Cette dernière ne prend pas en charge les clés étrangères.

### b. Insertion

Les données du projet sont collectées par Mr Levy et stockées dans un fichier.csv. A partir de ce fichier on a deux choix :

- On peut l'importer directement sur phpMyAdmin avec le système import fichier que fournit le site.
- On développe un script php pour importer les données et les insérer dans la base.

J'utilise la deuxième méthode. Je développe deux fonctions : une qui stock les informations du fichier.csv dans un « array », détaille du code dans la figure 3. Et la deuxième fonction insère les contenus de cette « array » dans la base de données (voir figure 4).

```
function importCSV($filename){  
    $row = 1;  
    $array;  
    if (($handle = fopen($filename, "r")) !== FALSE) {  
        while (($data = fgetcsv($handle, 1000, "\n")) !== FALSE) {  
            $num = count($data);  
            for ($c=0; $c < $num; $c++) {  
                $line = $data[$c];  
                $col1 = explode("\t", $line);  
                $col = str_replace('\'', '_', $col1);  
                $arrayLine = array(  
                    "problem" => ' '. $col[0]. ' ',  
                    "solution" => ' '. $col[1]. ' ',  
                    "status" => ' '. $col[2]. ' ',  
                    "error" => ' '. $col[3]. ' ',  
                    "correction" => ' '. $col[4]. ' ',  
                    "errorIndex" => ' '. $col[5]. ' ',  
                    "idProvenance" => 1,  
                    "lang" => ' '. $col[7]. ' ',  
                );  
                $array[] = $arrayLine;  
            }  
            $row++;  
        }  
        fclose($handle);  
    }  
    return $array;  
}
```

Fig3. importCSVFile

```
function insertData($filename){  
    $array = $this->importCSV($filename);  
    $num = 0;  
    $data = "";  
    $myliste;  
    foreach ($array as $value){  
        $myliste[] = $value;  
    }  
    while($num < sizeof($myliste)){  
        $data .= '('. $num. ', '. $myliste[$num]["problem"]. ', '.  
            $myliste[$num]["solution"]. ', '.  
            $myliste[$num]["status"]. ', '.  
            $myliste[$num]["error"]. ', '.  
            $myliste[$num]["correction"]. ', '.  
            $myliste[$num]["errorIndex"]. ', '.  
            $myliste[$num]["idProvenance"]. ', '.  
            $myliste[$num]["lang"]. ')';  
        $num++;  
    }  
    $res = str_replace(' ', ' ', $data);  
    $valuesInfo = str_replace('_', '\\_', $res);  
    $valuesInfo .= ' ';  
    // $this->pdo->exec('INSERT INTO ORIGIN(idOrigin,originSource,note) VALUES(1,"test","testNotes")');  
    $this->pdo->exec('INSERT INTO CASES( idCase, problem, solution, status, error, correction,  
        errorIndex, idProvenance, lang) values '. $valuesInfo );  
}
```

Fig.4 insertData

## II. LA CONCEPTION

### 1. Diagramme contexte statique

Le diagramme de contexte permet de visionner les interfaces entre le système sous enquêtes et le système des entités externes avec lesquels il est en interaction représenté par la figure 6. Dans notre cas, des entités externes sont des utilisateurs (USER) et le système sous enquêtes est le moteur de correction.

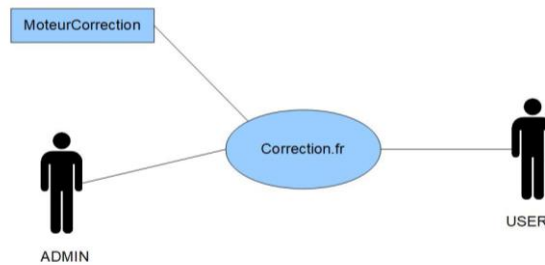


Fig.6 Diagramme de contexte statique.

### 2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les fonctionnalités nécessaires pour les utilisateurs (voir figure 7). Après les études des besoins des utilisateurs, on a trouvé quatre types de cas :

- Demande de correction (*request correction*).
- Proposer une correction (*propose correction*).
- Valider une correction (*validate a correction*).
- Modifier une base de cas (*Edit case base*).

Les deux derniers types sont réservés aux utilisateurs ayant un droit d'administrateur.

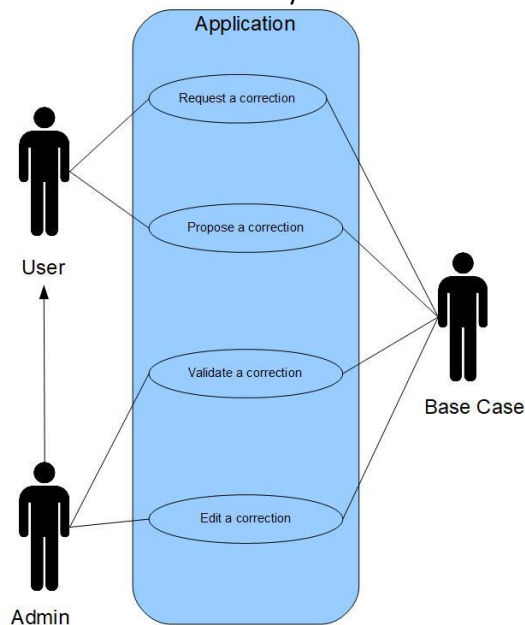


Fig.7 Diagramme de cas d'utilisation.



### 3. Diagramme Activité

Le diagramme d'activité représente le déroulement des actions, il est utilisé pour détailler les spécifications d'un cas d'utilisation et le déclenchement d'événements en fonction des états du système et la modélisation de ses comportements. Ainsi après, le diagramme de cas d'utilisation, on possède alors quatre diagrammes d'activité.

#### a. Demander une correction

Le système affiche une zone de texte où un utilisateur peut entrer la phrase incorrecte. Ensuite il récupère la phrase et déclenche le moteur de correction (Revise Sentence), puis, il affiche le résultat à l'utilisateur (présenter dans la figure 8).

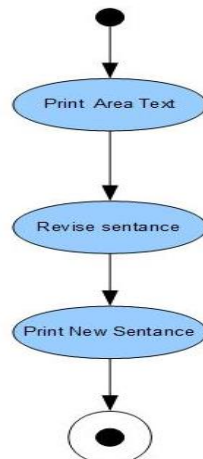


Fig.8 Demander une correction.

#### b. Proposer une correction

Le système a le même comportement que "Demander une correction " (voir figure 9), mais au lieu de déclencher le moteur de correction, il sauvegarde la phrase proposée par utilisateur dans la base courante.

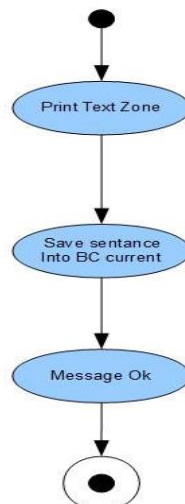


Fig.9 Proposer une correction.

### c. Valider la phrase

Cette activité est réservée pour les administrateurs. Le système affiche la phrase proposée par un utilisateur, puis elle déclenche le moteur de corrections et affiche le résultat. Il attend alors une confirmation de l'administrateur sur ce résultat pour savoir s'il doit stocker la phrase proposée dans la base de données ou l'enlever de la base de cas courant (figure 10).

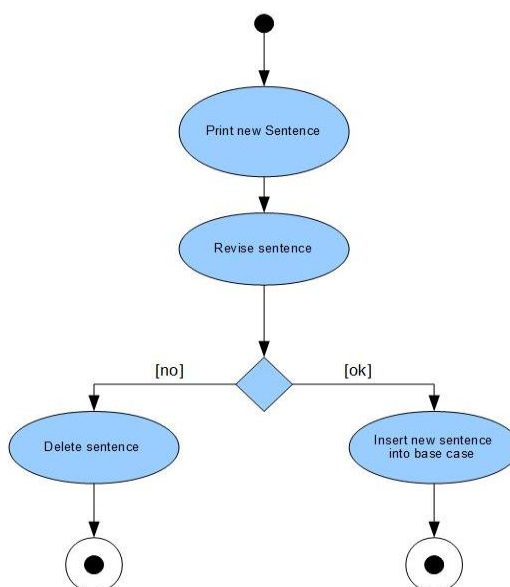


Fig.10 Validation une phrase par un administrateur.

### d. Modifications une phrase

Cette activité est identique à la précédente. Sauf qu'au lieu d'insérer dans la base courante, le système va modifier la phrase que l'administrateur pense être incorrecte (figure 11).

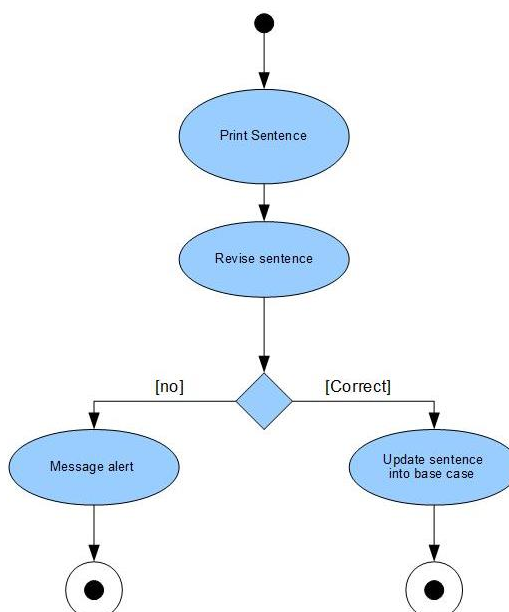


Fig.11 Modifications une phrase par administrateur.

#### 4. Diagramme de séquence

Ce type de diagramme permet de décrire les différents scénarios d'utilisation du système (figure 12). Il représente des interactions entre les acteurs et le système dans l'ordre chronologique. Pour générer ce genre de diagramme je me sers du site [websequencediagrams.com](http://websequencediagrams.com)<sup>4</sup>.

##### a. Demander une correction

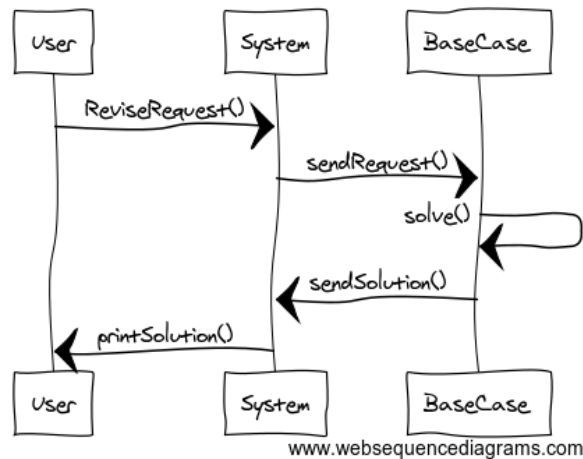


Fig.12 Demander une correction.

L'utilisateur demande une correction au système, le système envoie la requête au moteur de correction, qui va résoudre le problème et donne une solution au système, le système va afficher le résultat aux utilisateurs.

##### b. Proposer une correction

L'utilisateur propose une phrase, le système l'envoie à la base de cas, qui le stocke dans une relation CASES avec le statut « en attente » (figure 13).

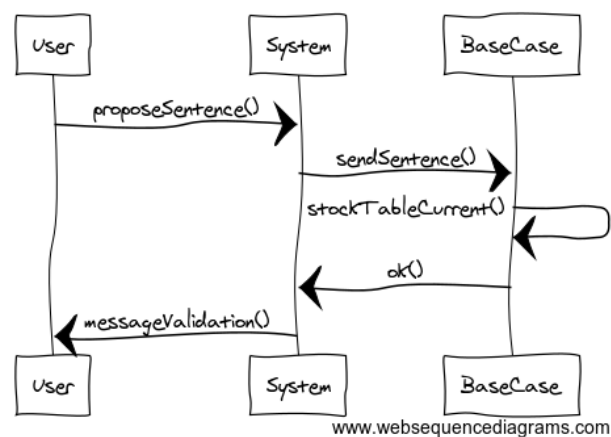


Fig.13 Proposer une Correction.

### c. Validation une phrase

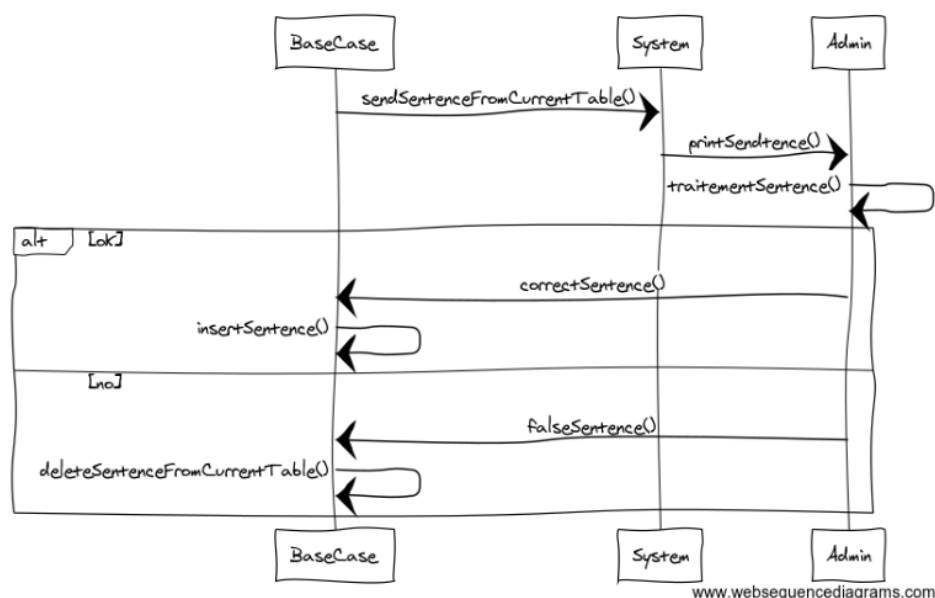


Fig.14 Validation une phrase.

Le système dans la relation CASES le cas avec statut « faux ou en attente », et l’affiche aux administrateurs. Ce dernier valide la phrase, si c’est correct, elle sera insérée dans la base de cas, sinon elle sera enlevée de la relation CASES (figure 14).

### d. Modifications une phrase

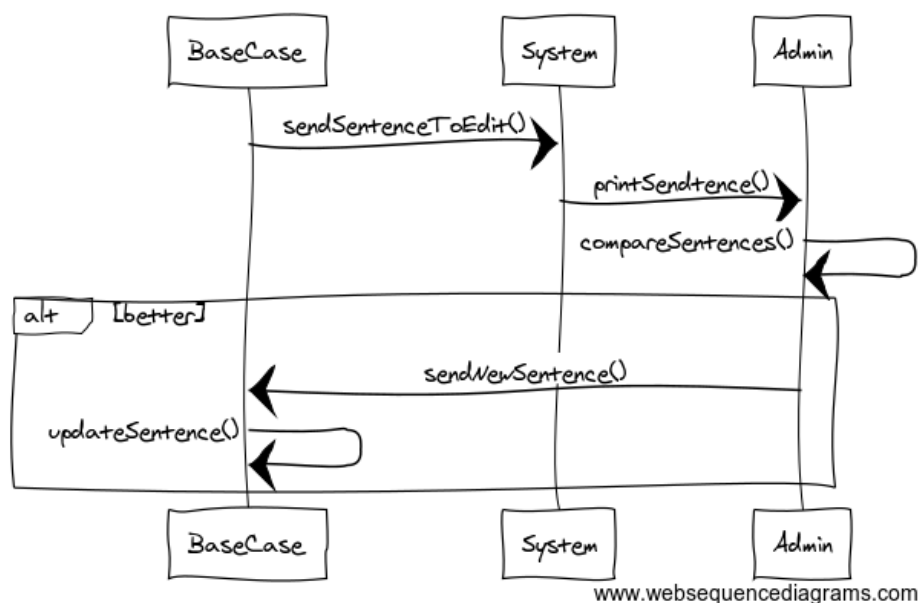


Fig.15 Modifications une phrase

Le système affiche la phrase à modifier, l’administrateur compare la nouvelle phrase par rapport à l’ancienne. Si elle est meilleure, la nouvelle phrase remplacera alors l’ancienne dans la base de cas (figure 15).

### III. ENVIRONNEMENT DE DEVELOPPEMENT

#### 1. MySQL<sup>5</sup> et phpMyAdmin

J'ai choisi MySQL comme gestionnaire de base de données pour notre site web. PhpMyAdmin (figure 16) est une application web pour la gestion de MySQL. Comme vous pouvez le constater, ces trois tables portent le même nom que dans le MCD (figure 1). Ici je choisis « Interclassement (dans figure 18) » est l'utf8. Cette convention a été imposée par les encadrants.

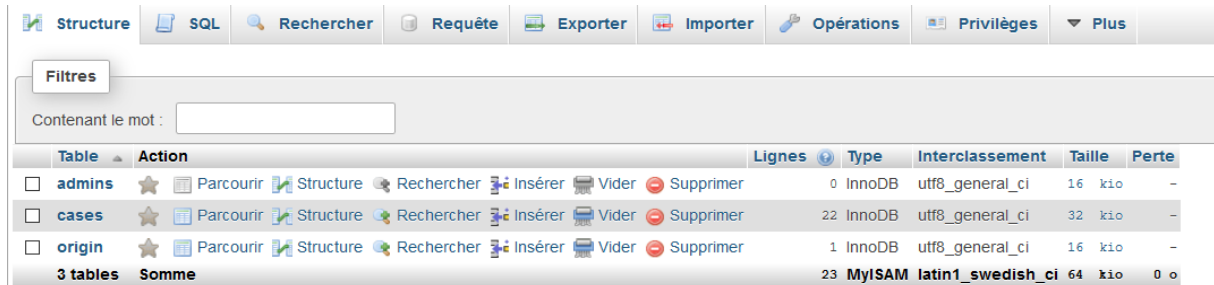
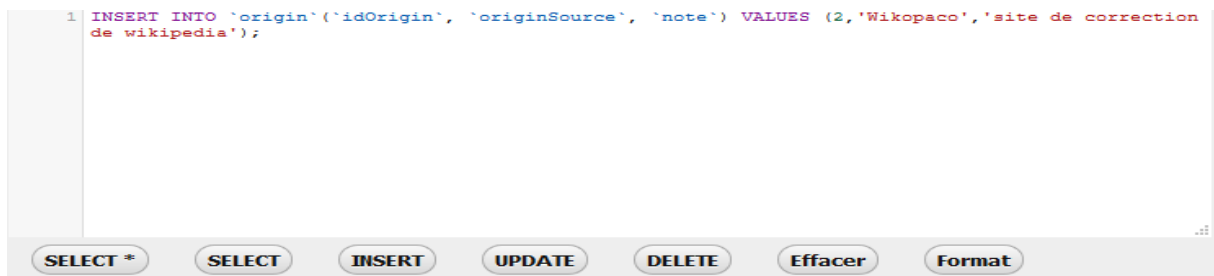


Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> admins	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	16 kio	-
<input type="checkbox"/> cases	Parcourir Structure Rechercher Insérer Vider Supprimer	22	InnoDB	utf8_general_ci	32 kio	-
<input type="checkbox"/> origin	Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8_general_ci	16 kio	-
<b>3 tables</b>	<b>Somme</b>	<b>23</b>	<b>MyISAM</b>	<b>latin1_swedish_ci</b>	<b>64 kio</b>	<b>0 o</b>

Fig.16 : interface phpMyAdmin.

L'interface est simple à utiliser. On peut rentrer directement les requêtes SQL via les formulaires proposées par le site (fig17). J'utilise ces formulaires souvent lors de la création de la base de données et les tests d'insertion.



```
1 INSERT INTO `origin` (`idOrigin`, `originSource`, `note`) VALUES (2, 'Wikopaco', 'site de correction de wikipedia');
```

SELECT \* SELECT INSERT UPDATE DELETE Effacer Format

Fig.17 : insertion d'une ligne dans la relation Origin.

#### 2. Bootstrap<sup>6</sup>

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est la première fois que, j'utilise ce framework. Mais il est simple à utiliser et intuitif. En plus, il y a des tutoriaux sur l'Internet pour apprendre à l'utiliser. Ainsi après une dizaine heures, je commence à maîtriser cette framework.

### 3. SublimeText<sup>7</sup>

SublimeText est un éditeur de texte code en C++ et Python. C'est une extension pour VIM. Il possède une interface de développement standard. Le côté droit (Fig18.A) est une arborescence de projet. La partie central (Fig18.B) est la zone de cote.

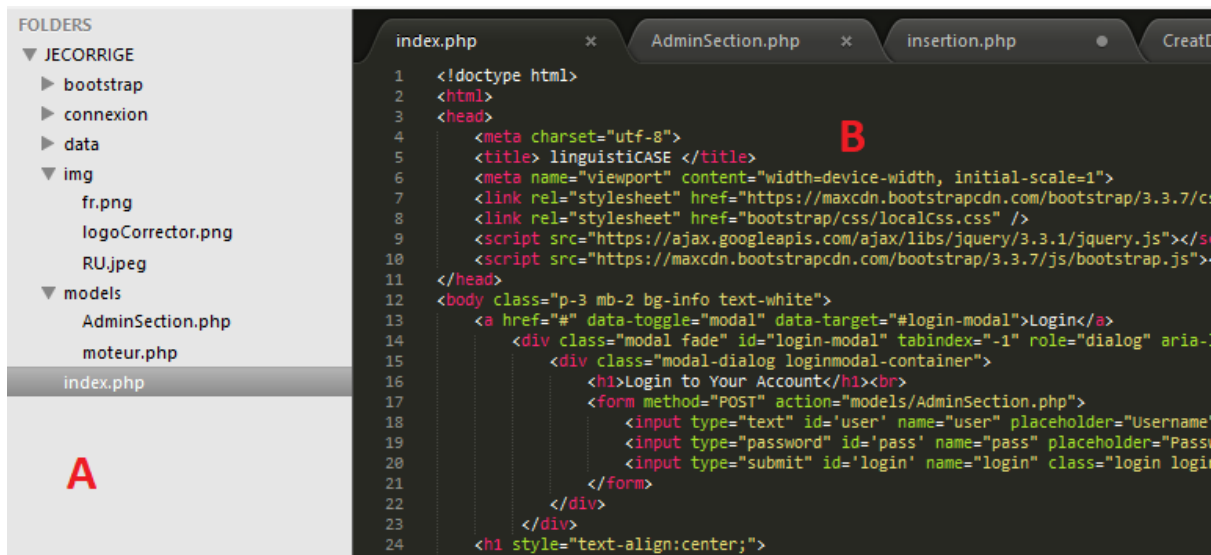


Fig18 : interface de SublimeText.

Ce choix n'est pas imposé pour le stage. Au début, j'ai choisi phpStorm<sup>8</sup> (c'est un environnement de développement, pour le développement PHP). Mais j'ai un problème avec l'installation et la performance du PC, j'ai décidé de prendre SublimeText pour ce projet. Certes, SublimeText est moins puissante que phpStorm car c'est un éditeur de texte et l'autre est un IDE. Mais SublimeText dispose assez de fonctionnalité pour mes besoins.

### 4. Balsamiq<sup>9</sup>

Balsamiq est un logiciel payant pour faire les maquettes d'interface utilisateur. Je l'ai déjà utilisé pour un de mes projets universitaires donc, je connais et maîtrise certaines de ses fonctionnalités. A la demande du responsable, j'ai installé une version essai pour mes besoins, c'est très suffisant pour les éléments de notre interface web.

## 5. Wampserver<sup>10</sup>

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas un logiciel, mais un environnement qui contient trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

J'utilise souvent cette interface souvent pour accéder au site phpMyAdmin et localhost.

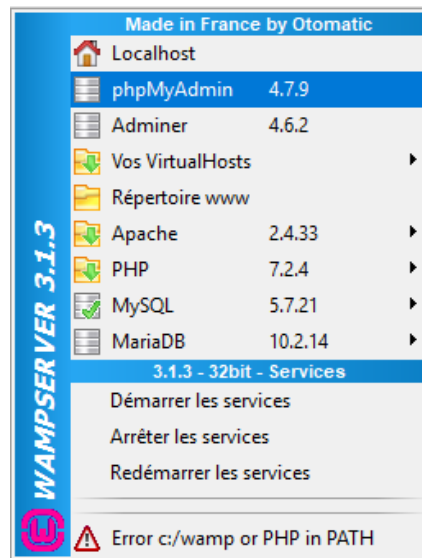


Fig19. Wampserver

Parmi les options disponibles(Fig19) je n'utilise que :

Localhost : pour accéder au site web en cours de développement.

PhpMyAdmin : pour afficher l'interface web de la base de données.

## 6. GIT

Comme expliqué dans l'introduction, je suis en collaboration avec deux autres personnes pour ce projet. Ainsi pour mettre en commun notre code, on a décidé de tous utiliser un logiciel de gestion de version GIT, le dépôt est sur le site de GitHub<sup>11</sup>. On a choisi ce logiciel, parce qu'il est libre, et qu'on le connaît bien.

## IV. INTERFACE WEB

### 1. Corrector un site multilingue

#### a. Historique

Ce projet est un système de correction de phrases. Ainsi, on a dû trouver un nom et un logo pour le site web, le premier nom a été linguistiCase, mais après une certaine réunion, on a constaté que le nom était trop vague, il nous fallait un nom plus précis, on a donc trouvé « Corrector » et le nom n'est pas encore utilisé (Fig20). Pourquoi Corrector ? Cette idée vient de notre responsable Mr Lieber, pendant une réunion, on n'avait pas d'idée, donc il a proposé Corrector (Figure 20).



Fig20 : Logo de Corrector.

#### b. Site multilingue

Ce projet est à la base un sujet de recherche changé en sujet de stage. Ainsi, pour les explications, c'est mieux d'être en anglais. Donc, le responsable préfère que le site marche en français et en anglais. Pour représenter les langues, on a décidé de les représenter avec des drapeaux (Fig21).

Ainsi, l'interface du site change de langue si on clique sur le drapeau.



Fig21 : Site multilingue.

Pour développer ce site, j'utilise PHP<sup>12</sup> et Javascript<sup>13</sup>, langages que j'ai déjà utilisé une ou deux fois pendant mes projets universitaires, j'ai aussi suivi des tutoriels sur l'Internet<sup>14</sup> pour le débogage et pour chercher les informations pour mes besoins. Je n'ai pas rencontré de difficultés particulières pendant le développement. Le seul problème que j'ai eu étant au niveau du design du site.



## 2. La maquette du site

Avant de commencer à développer le site web, je dois d'abord faire la conception (PARTIE II) et une maquette à présenter à mes responsables. Ainsi j'ai décidé d'utiliser balsamiq pour faire la maquette du site (Fig.22). Après les analyses des besoins d'utilisateur. Il suffit d'une zone de texte (Fig.22A), un bouton pour valider (Fig.22B), une zone de réponse du serveur (Fig.22C), des boutons sur les retours d'utilisateurs (Fig.22.D).

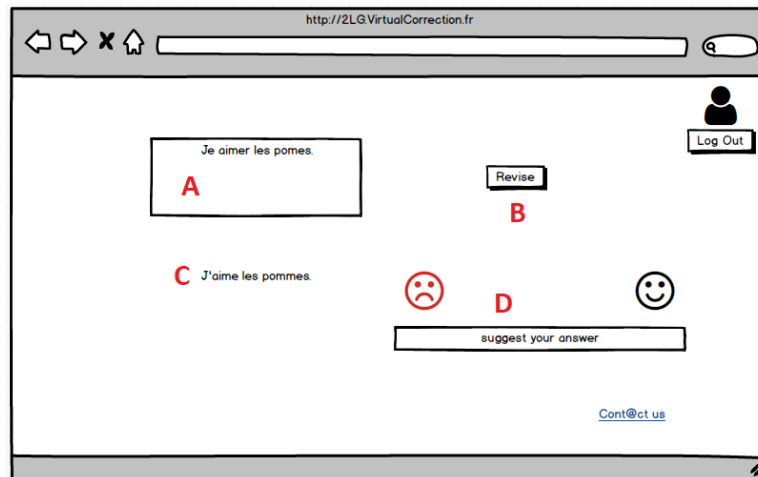


Fig22 : Une première maquette du site

Cette première version n'a été jugée améliorable suite à une réunion avec mes encadrants : les éléments doivent être alignés, la section « identification » réserve pour l'administrateur est trop en valeur. Il doit être plus petit et changer en LoginAdmin. De plus, les noms des boutons doivent être modifiés au fur et à mesure du projet. Si je me rappelle bien, j'ai changé le nom des boutons dizaine de fois.

Pour la partie d'identification de l'administrateur (Fig23), j'ai pensé à faire un popup standard qu'on rencontre souvent sur les sites internet, avec une identification, un mot de passe et un bouton « ok ».

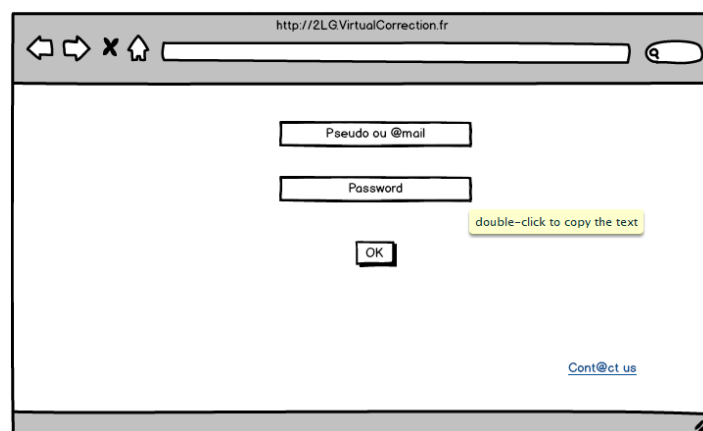


Fig23 : connexion à la section de l'administrateur.

### 3. Interface utilisateur

La première version du site (Fig24) n'utilise pas Bootstrap. De plus lors de la présentation devant mes responsables, ils m'ont fait remarquer qu'il y a trop de changement de page. C'est-à-dire si je clique sur « Français » ou « English » cela redirige vers une autre page au lieu d'actualiser la page (Fig.25). En parallèle, le style de texte du site web n'était pas esthétique. Ainsi, après la réunion, Mr Nauer m'a conseillé d'utiliser Bootstrap, et au lieu de faire des redirections de page, il préfère que le déroulement se fasse en une seule page avec des balises qui affichent au fur et mesure (Fig26).



Fig24 accueil

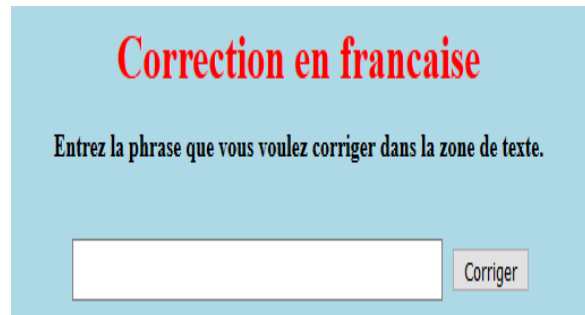


Fig25 Page française

Lorsque j'utilise bootstrap, surtout avec des couleurs, j'ai appris que chaque couleur choisie par bootstrap correspond à une signification. Le bleu clair est pour l'information, le bleu standard pour valider, le vert est pour le succès, et le rouge c'est pour un « warning ».

L'utilisation de Corrector est très intuitive : l'utilisateur entre une phrase dans le champ « Entrez la phrase à corriger », il clique sur « valider », le site affiche la réponse avec deux boutons juste en dessous, ces deux boutons sont « Correct et Incorrect ».

Si l'utilisateur clique sur « Correct », le site demande si l'utilisateur veut insérer sa solution dans la base de données pour améliorer la performance du site.

Si l'utilisateur clique sur « Incorrect » le site affiche alors un autre formulaire pour demander à l'utilisateur de rentrer sa correction. Et « OK » est pour insérer dans la base de données le problème de l'utilisateur et sa correction.

Toutes les insertions des utilisateurs sont stockées dans la table CASE mais avec un statut « EN ATTENTE ».



Fig 26 : Corrector

## 4. Interface administrateur

L'interface administrateur a moins d'éléments que l'interface utilisateur. Après avoir identifié l'administrateur, le site redirige vers l'interface réservé uniquement à l'administrateur (pas d'inscription).

L'interface administrateur contient seulement deux éléments (Fig27).

« Restaurer la base de données » : elle a pour le but de remettre l'état de la base de données du site à l'état initial, c'est-à-dire à partir d'un fichier CSV que Mr Levy a créé. Cette partie est en cours de discussion avec les responsables, si on restaure, toutes les insertions des utilisateurs seront effacées.

« Valider les cas utilisateur » : si l'administrateur clique sur afficher, le site affiche dans un tableau les données de la table CASE avec statut « en attente » avec un bouton « OK ». Le bouton OK permet de changer le statut « en attente » au statut « true » qui signifie que la phrase est validée et c'est un bon couple (problème, solution) dans la table CASES. Cette partie est en cours de réflexion, dois-je créer un bouton supplémentaire pour refuser une proposition, ainsi, la proposition sera supprimée de la table CASES.

Bonjour Admin1

Restaurer la base de données Reset

Valider les cas d'utilisateur Afficher

Probleme	Solution	Langue	
je bois pas	je ne bois pas	fr	<span style="background-color: #4682b4; color: white; padding: 2px 5px;">OK</span>

Fig27 : interface Administrateur.

Ici on ne peut pas s'inscrire car les administrateurs sont les responsables du projet. Cette partie est en cours de développement.

## CONCLUSION :

Niveau professionnel : Comme c'est un sujet de recherche. A part, il faut respecter les consignes de l'encadrant. Je suis libre sur mes décisions, c'est-à-dire, au niveau du langage de code, le design du site web, le type de base de données choisie. Pendant les réunions, l'encadrant donne un objectif à atteindre et à faire pour la prochaine réunion. On fait une réunion toutes les deux semaines avec l'encadrant. Des fois, on fait aussi des visioconférences avec Mr LEPAGE au japon.

Niveau formation : Les connaissances que j'ai acquise pendant mes années universitaires m'ont bien servi. Par exemple : la conception, les logiciels utilisés, la façon de coder.

Niveau personnel : Corrector est le premier site web que j'ai développé tout seul. Je suis très content quand les encadrants sont satisfaits de mon travail. Au début, pour mettre en place c'est difficile. Soit, je n'ai pas d'idée sur la façon de comment développer une fonction, soit il y a trop et il faut choisir le meilleur. A partir de ce projet, maintenant, si j'utilise une phrase négative, je me rappelle qu'il y a « ne ou n' ». De plus, si je disposais plus de temps, je pourrais ajouter plus de fonctionnalités pour Corrector.

## Annexe :

Les codes source sont disponibles sur le dépôt Github<sup>15</sup>.

## Sitographie<sup>16</sup> :

---

<sup>1</sup> <http://www.loria.fr/fr/presentation/>

<sup>2</sup> <http://www.loria.fr/fr/la-recherche/departements/4-traitement-des-langues-et-des-connaissances/>

<sup>3</sup> <http://mocodo.wingi.net/>

<sup>4</sup> <https://www.websequencediagrams.com/>

<sup>5</sup> <https://www.mysql.com/fr/>

<sup>6</sup> <https://www.w3schools.com/bootstrap/>

<sup>7</sup> <https://www.sublimetext.com/>

<sup>8</sup> <https://www.jetbrains.com/phpstorm/>

<sup>9</sup> <https://balsamiq.com/>

<sup>10</sup> <http://www.wampserver.com/>

<sup>11</sup> <https://github.com/>

<sup>12</sup> <https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql>

<sup>13</sup> <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript>

<sup>14</sup> <https://stackoverflow.com/> , <https://openclassrooms.com/forum/> et <https://fr.wikipedia.org/wiki/Wiki>

<sup>15</sup> <https://github.com/Sle3pingForest/StageL3/tree/master/WEB>

<sup>16</sup> <https://fr.wiktionary.org/wiki/sitographie>