



**TUTRICE DE STAGE :**

**Mme.**

**Martine GAUTHIER**

**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) Correcteur 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>

## **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 grammaire automatique. 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 à partir du raisonnement à partir de cas.

Le raisonnement à partir de cas (RAPC) : (\*\*pour expliquer le RAPC est ce que je peux utiliser l'explication du PDF du sujet ? \*\*) consiste donc à s'appuyer sur des couples (problème, solution) ou problème est une phrase incorrecte qui est corrigée en solution pour proposer une correction à une nouvelle phrase cible. Par exemple :

*Problème : Je aimer des pommes.*

*Solution : J'aime des 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 utiliser plus tard en tant que (problème, solution).

La base de cas minimal 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 d'utilisation. Le remémoration et l'adaptation sont des algorithmes développés par M. Giang en Python. En s'appuyant sur le RAPC pour corriger un problème et donner une solution optimale.

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 de la création d'une base de données, la conception, suivi l'environnement de développement, et enfin le développement de l'interface web.

## **Présentation Entreprise :**

Le LORIA est le laboratoire lorrain de recherche en informatique et ses applications. Il est situé à Vandœuvre-lès-Nancy et Villers-lès-Nancy, près de Nancy, sur le campus de l'université Henri-Poincaré (Nancy-I).

Le Loria : Laboratoire lorrain de Recherche en Informatique et ses Applications est une réunion de CNRS, l'Université de Lorraine et Inria.

Le Loria a pour mission la recherche fondamentale et appliquée en sciences informatiques et ce, depuis sa création, en 1997.

Le Loria est membre de la Fédération Charles Hermite qui regroupe les trois principaux laboratoires de recherche en mathématiques et STIC (science et technologies de l'information et de la communication) de Lorraine. Le laboratoire fait partie du pôle scientifique AM2I (Automatique, Mathématiques, Informatique et leurs interactions) de l'Université de Lorraine.

Nos travaux scientifiques sont menés au sein de 28 équipes structurées en 5 départements, dont 15 sont communes avec Inria, représentant un total de plus de 400 personnes. Le Loria est un des plus grands laboratoires de la région lorraine. (*Source : Loria.fr*)

## I) BASE DE DONÉES

### 1) Présentation de la base de données

Pour ce stage, le type de base de données n'est pas imposé. Donc je choisis le MySQL, car c'est le type que j'utilise pendant mon année universitaire.

Le seul inconvénient ici c'est que cette base devrait adapter aussi au travail de Mr. Levy. Ainsi, Après plusieurs réunions, on a décidé que cette base est de la forme ci-dessous (fig.1).

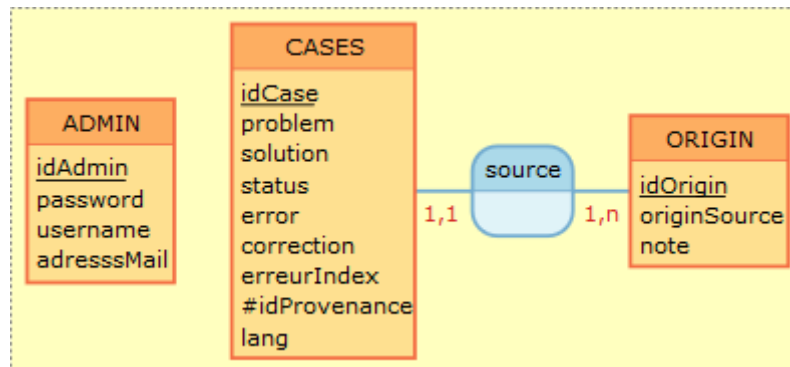


Fig.1 : Base de données.

Notre base de données contient trois tables :

La table CASES est composée d'une clé primaire idCase, ainsi de deux colonnes « problem » et « solution », ces deux dernières forment alors la couple (problème, solution) de RAPC. La colonne « status » est de forme « bon, mauvais, en attente » pour indiquer si la solution est correcte rapport au problème. Les deux colonnes suivantes sont « erreur » et « correction » elles indiquent le(s) mot(s) qui a été corrigé. « erreurIndex » donne la position du mot erroné dans la phrase. « lang » pour l'instant notre projet ne comporte seulement du français et d'anglais mais elle peut évoluer dans le futur.

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

La table ADMIN contient les informations nécessaires de l'administrateur.

## 2) Génération du script de la création et de l'insertion

### a) Création

Le code de création d'une base est très standard (Fig.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 création une table

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 : MYISM. 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 :

- 1) On peut l'importer directement sur phpMyAdmin avec le système import fichier que fournit le site.
- 2) 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 » (Fig.3). Et la deuxième fonction insère les contenus de cette « array » dans la base de données (Fig.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('\t', '_', $col1);
                $arrayLine = array(
                    "problem" => '\t'.$col[0].'\t',
                    "solution" => '\t'.$col[1].'\t',
                    "status" => '\t'.$col[2].'\t',
                    "error" => '\t'.$col[3].'\t',
                    "correction" => '\t'.$col[4].'\t',
                    "errorIndex" => '\t'.$col[5].'\t',
                    "idProvenance" => 1,
                    "lang" => '\t'.$col[7].'\t',
                );
                $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 (Fig.6).

Dans notre cas, des entités externes sont des utilisateurs (USER) et le système sous enquêtes est le moteur de correction.

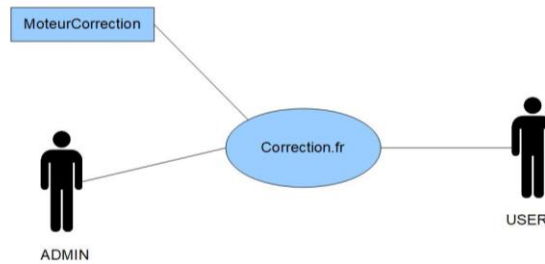


Fig.6 Diagramme contexte statique

### 2) Diagramme Cas Utilisation

Le diagramme de cas d'utilisation représente les fonctionnalités nécessaires pour les utilisateurs. Après les études aux 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 base case*).

Les deux derniers types sont réservés aux utilisateurs avec le droit administrateur.

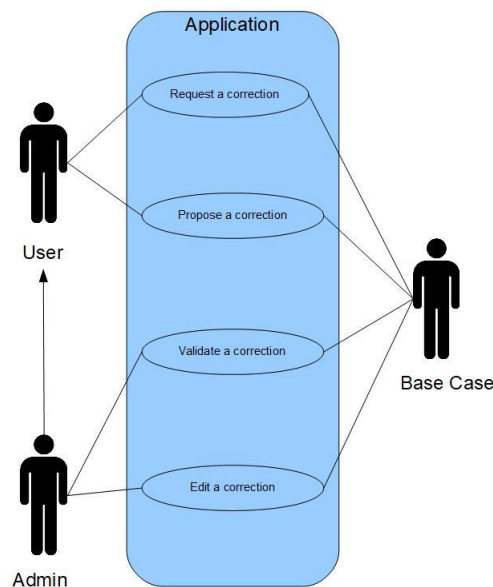


Fig.7 Diagramme cas utilisation



### 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 modéliser 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, elle affiche le résultat à l'utilisateur.

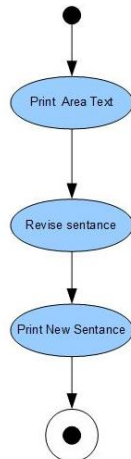


Fig.8 Demande une correction

#### b) Proposer une correction

Le système a le même comportement que "Demander une correction", 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.

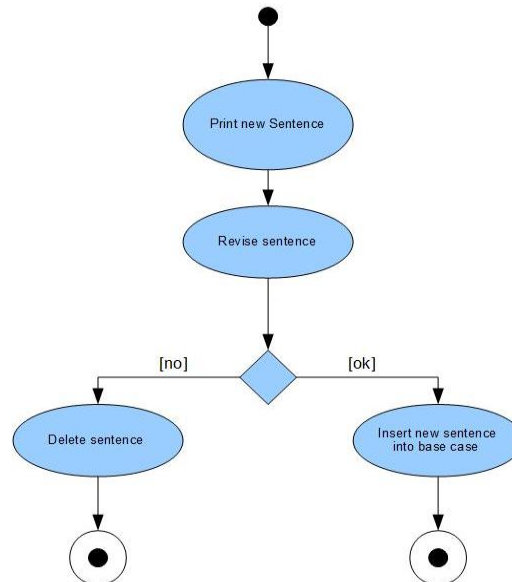


Fig.10 Validation une phrase par Admin

d) Modifications une phrase

Cette activité est identique que celle précédemment. Mais au lieu d'insérer dans la base courante, le système va modifier la phrase que l'administrateur pense que c'est incorrect.

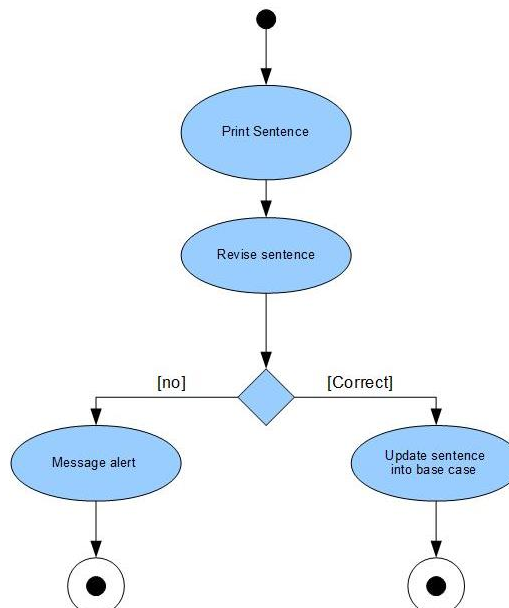


Fig.11 Modifications une phrase par Admin

#### 4) Diagramme séquence

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

##### a) Demander une correction

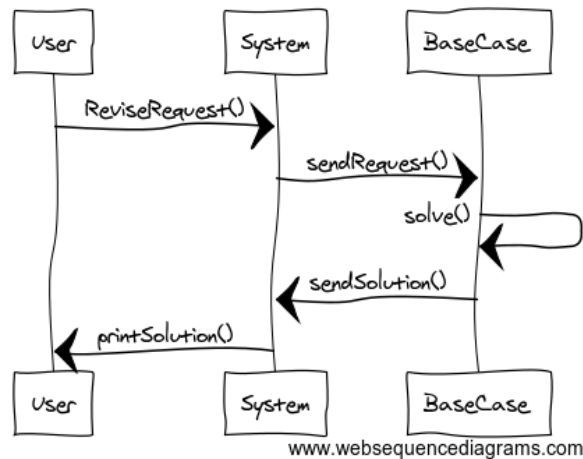
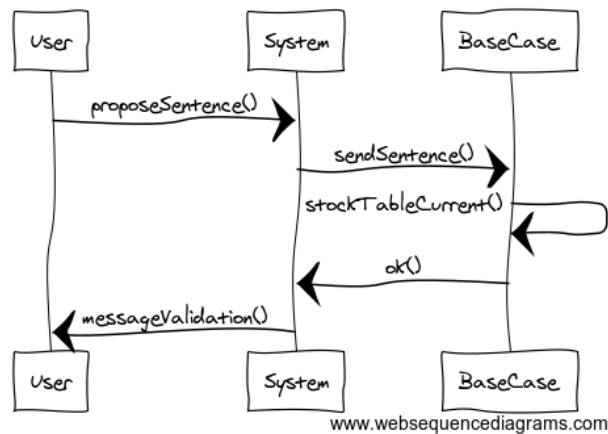


Fig.12 Demander une correction

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 stock dans une table courant.

Fig.13 Proposer une Correction

### c) Validation une phrase

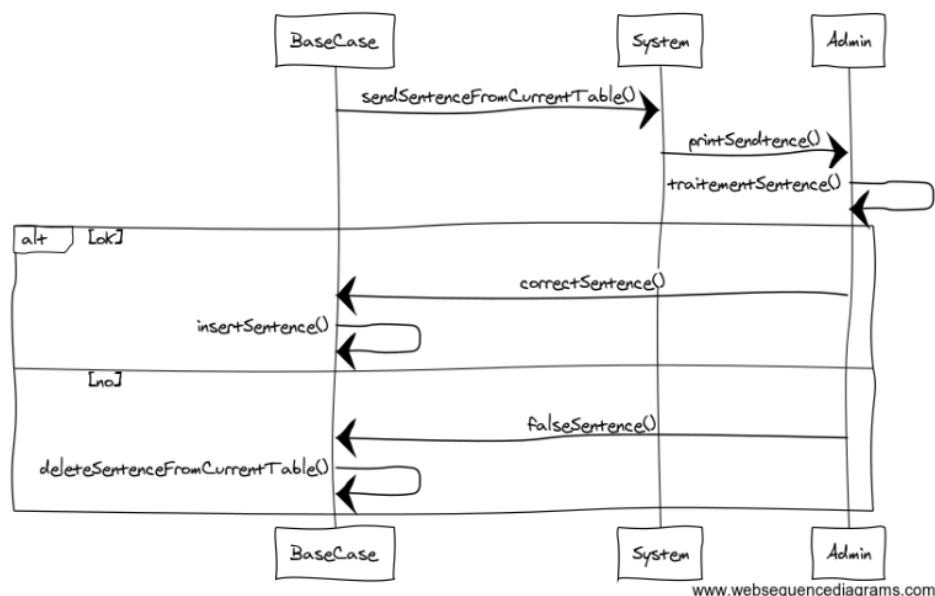


Fig.14 Validation une phrase

Le système récupère la phrase stockée dans la table courante, 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 table courante.

### d) Modifications une phrase

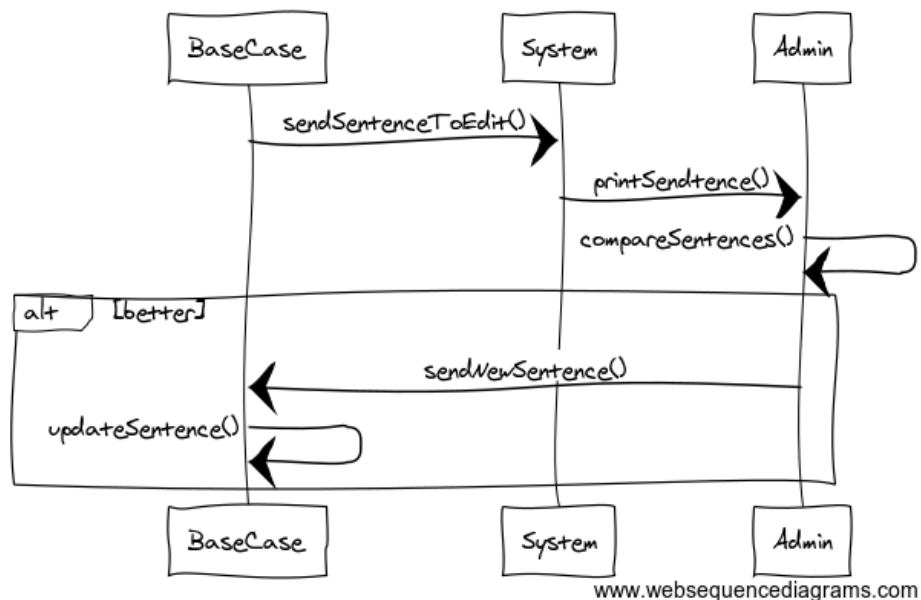


Fig.15 Modifications une phrase

Le système affiche la phrase a modifié, l'administrateur compare la nouvelle phrase par rapport à l'ancienne. Si elle est meilleur, la nouvelle phrase remplacera alors l'ancienne dans la base de case.

### III) ENVIRONNEMENT DE DEVELOPPEMENT

#### 1) MySQL et phpMyAdmin

Je choisis MySQL comme le gestionnaire de base de données pour notre site web. Car comme j'ai déjà utilisé pendant mes années universitaires, je me sens plus à l'aise avec ce gestionnaire. Et le phpMyAdmin (fig. 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 (Fig.1). Ici je choisis « Interclassement » est l'utf8. Car, la convention est imposée par le chef de stage.

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	-
3 tables	Somme	23	MyISAM	latin1_swedish_ci	64 Kio	0

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 lorsque de la création de la base de données et les testes d'insertion.

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

Fig.17 : insertion une line dans la table Origin

#### 2) Bootstrap

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 j'utilise cette framework. Mais il est simple et intuitive à utiliser. En plus, il y a des tutoriaux sur internet pour apprendre à l'utiliser, ainsi après une dizaine d'heures, je commence à maîtriser cette framework.

### 3) SublimeText

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.

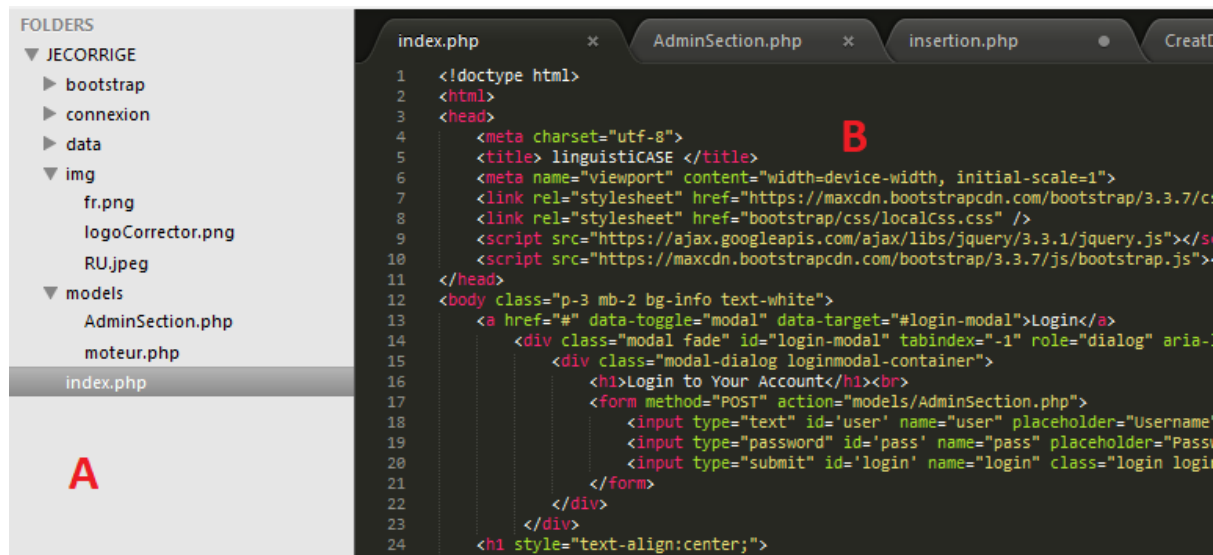


Fig18 : SublimeText

Ce choix n'est pas imposé pour le stage. Au début, j'ai choisi le phpStorm (c'est une interface développement environnement : IDE, pour le développement PHP). Mais j'ai un problème avec l'installation et la performance du CPU, j'ai décidé de prendre SublimeText pour ce projet. Certes, SublimeText est moins puissance que phpStorm car une c'est un éditeur de texte et l'autre est un IDE. Mais SublimeText dispose assez de fonctionnalité pour mes besoin (Exemple : complétion automatique...).

### 4) Balsamiq

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

## 5) Wampserver

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 contient trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

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

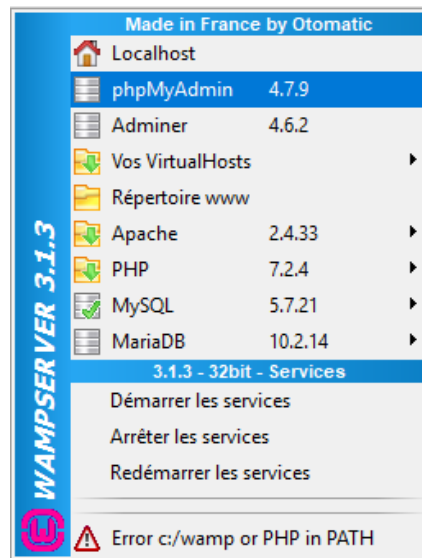


Fig19. Wampserver

Parmi les options disponibles(Fig19) j'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 j'ai déjà expliqué dans l'introduction, je suis en collaborative avec deux autres personnes pour ce projet. Ainsi pour mettre en commun notre partie, on a décidé de tous utiliser un logiciel de gestion de version GIT. Pourquoi on a choisi ce logiciel, car il est libre, et comme on utilise très souvent pendant notre projet universitaire, on est plus à l'aise.

## IV) INTERFACE WEB

### 1) Corrector un site multilingue

#### a) Historique

Ce projet est un système de correction de phrases. Ainsi, on doit trouver un nom et un logo pour le site web, le premier nom on a trouvé est lingustiCase, mais après une certaine réunion, on a constaté que le nom est trop vague, il nous faut 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 le corrector.



Fig20 : 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 : siteMultilingue

Pour développer ce site, j'utilise du PHP, du Javascript. Ces deux langages, que j'ai déjà utilisé une ou deux fois pendant mes projets universitaires. Je n'ai pas rencontré de difficultés particulières pendant le développement. Le seul problème que j'ai eu c'est au niveau de 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).

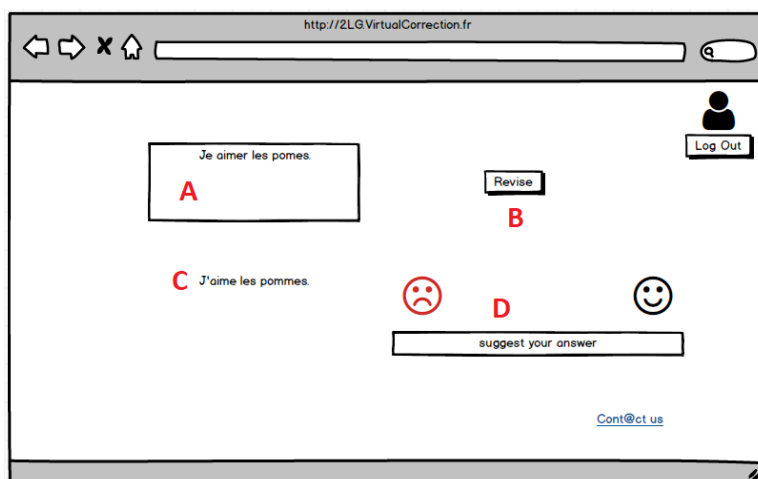


Fig22 : une maquette du site

Cette première version n'est pas optimale après la réunion. Car les éléments doivent être alignée. La section « indetification » n'est réserver que pour l'administrateur est trop en valeur. Il doit être plus petit et changer en LoginAdmin. De plus, les noms des butons doivent être modifie au fur et à mesure du projet. Si je me rappel bien, j'ai changé le nom des butons dizaine de fois.

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

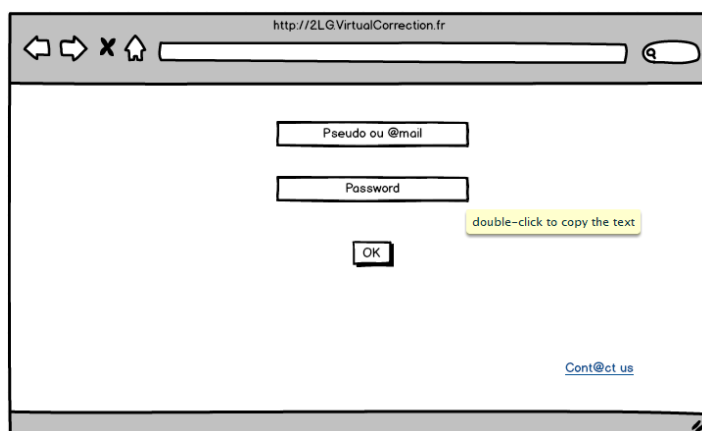


Fig23 : login Admin

### 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 de site web n'est 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

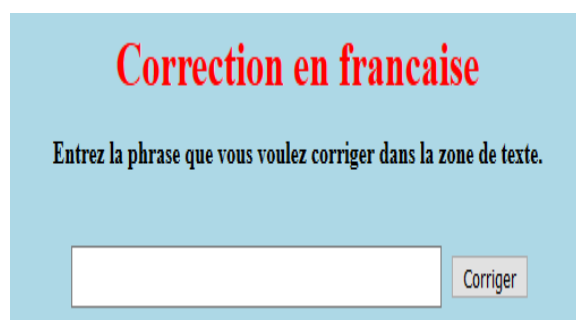


Fig25 Page française

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 « 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 son problème dans la base de données pour améliorer la performance du site.

Si « 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.

Attention 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 d'administrateur a moins d'éléments que l'interface utilisateur. Après avoir identifié, 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 d'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. Pour cette partie, je suis en cours de réflexion, si je dois 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.

## **Bibliographie :**

Présentation entreprise : <http://www.loria.fr/fr/presentation/>

Bug: <https://stackoverflow.com/>

Design: <https://www.w3schools.com/bootstrap/>

Base de données : <https://www.mysql.com/fr/>

Information : <https://fr.wikipedia.org/wiki/Wiki>

Autres : <https://openclassrooms.com/>

phpStorm: <https://www.jetbrains.com/>