



**RESPONSABLE LICENCE 3 :**

**M.**

**Emmanuel JEANDEL**

**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**

## Introduction :

(\*\*amener le sujet\*\*)

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).

(\*\*lien avec Damien\*\*)

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 WiKoPaCo qui est un historique de modification de pages sous Wikipédia.

Puis, elle sera améliorée au fur et à mesure d'utilisation.

(\*\*lien avec André\*\*)

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.

(\*\*Présentation du sujet\*\*)

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

(\*\*annoncer le plan\*\*)

Tout d'abord, je vais présenter la conception, l'environnement de développement, suivi de la création d'une base de données et enfin le développement de l'interface web. (\*\*a complet s'il aura du changement\*\*).

(\*\*REMERCIEMENT\*\*)

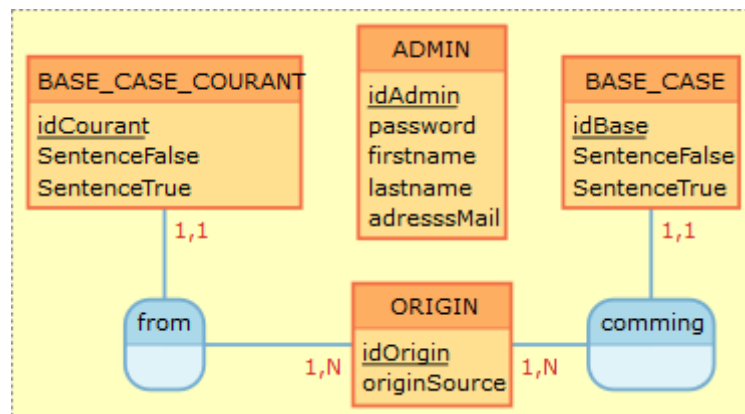
(\*\*en cours \*\*)

(\*\*SOMMAIRE\*\*)

(\*\*en cours\*\*)

(\*\*Création de Data base\*\*)

(\*\* MCD\*\*)



Notre base de donne contient quatre tables :

La table BASE\_CASE est composée une clé primaire idBase, ainsi deux colonnes SentenceFalse et SentenceTrue, ces deux dernières forment alors la couple (problème, solution) a RAPC.

La table ORIGIN (ou provenance) stock les informations de la provenance de la couple (problème, solution), la clé primaire est idOrigin.

La BASE\_CASE\_CURANT est identique que celle de BASE\_CASE, mais elle stock les couple (problème, solution) proposées par les utilisateurs et attendent la validation de l'administrateur.

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

(\*\*Code DONE\*\*)

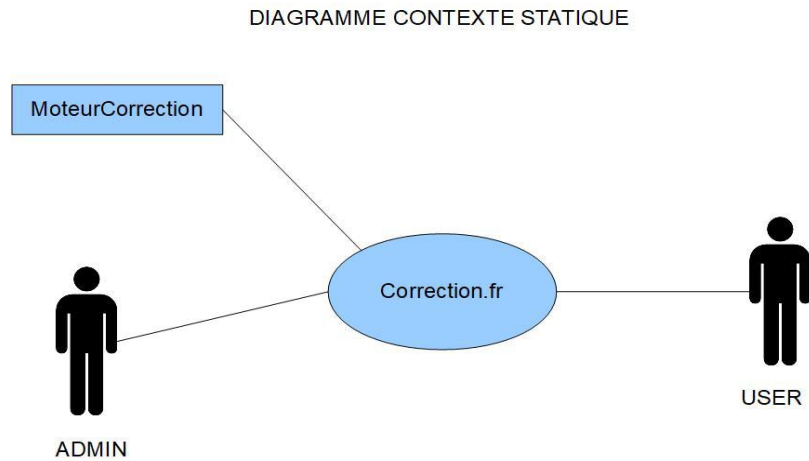
(\*\* problème MYISM ne prend pas en charge les constraints clé étrangère, il faut utiliser innodb, le code de la création est fait + code insertion, \*\*)

(\*\*Conception\*\*)

(\*\*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.

Dans nos cas, des entités externes sont des utilisateurs et le système sous enquêtes est le moteur de correction.



(\*\*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 :

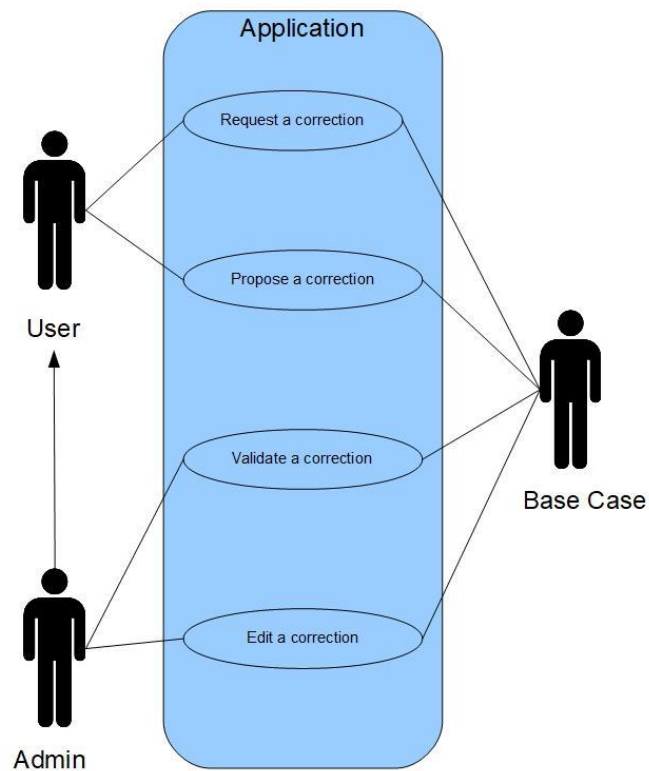
Demande de correction (request correction).

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.



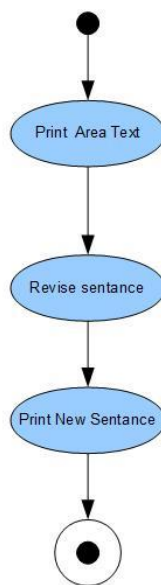
### (\*\*Diagramme Activité \*\*)

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'évènement 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.

#### (\*\*request 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.

Request a correction



#### (\*\* propose correction\*\*)

Le système a le même comportement que "request correction ", mais au lieu de déclencher le moteur de correction, il sauvegarde la phrase proposée par utilisateur dans la base courante.

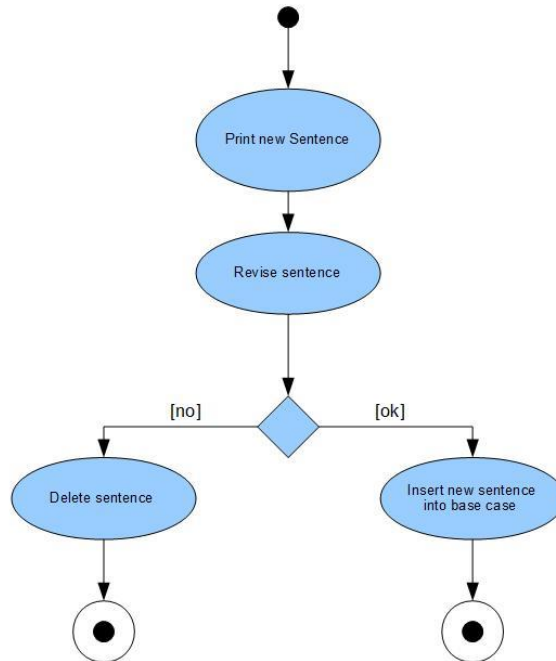
Propose a correction



(\*\*validate a sentence \*\*)

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.

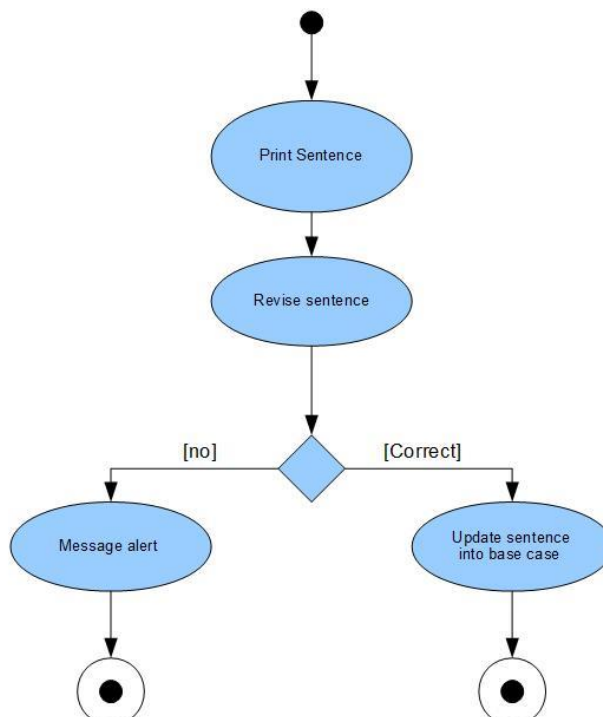
#### Validation sentence



(\*\* modifications a sentence\*\*)

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.

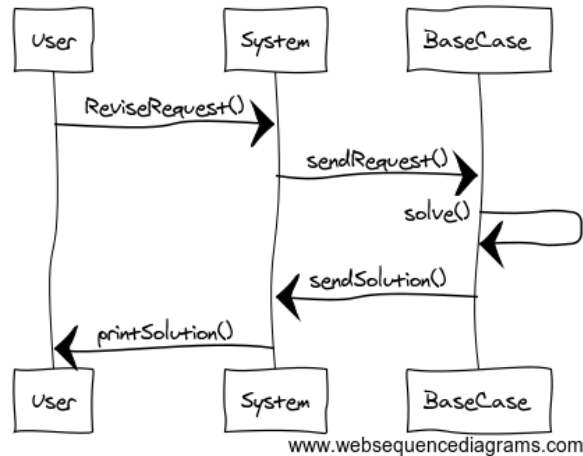
#### Edit Sentence



(\*\*Diagramme séquence\*\*)

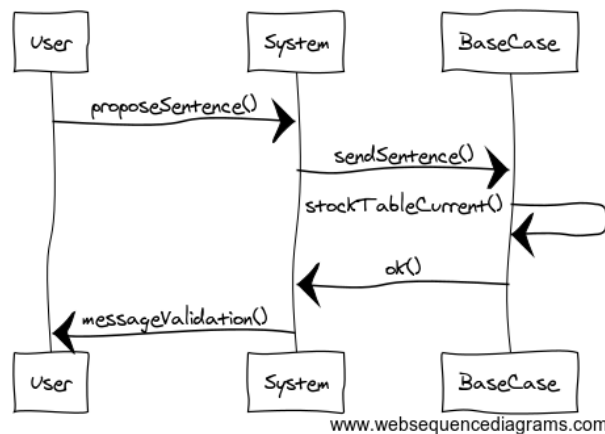
Ce type de diagramme permet de décrire les différents scénarios d'utilisation du système. Il représente des interactions entre les acteurs et le système dans l'ordre chronologie.

(\*\*request 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.

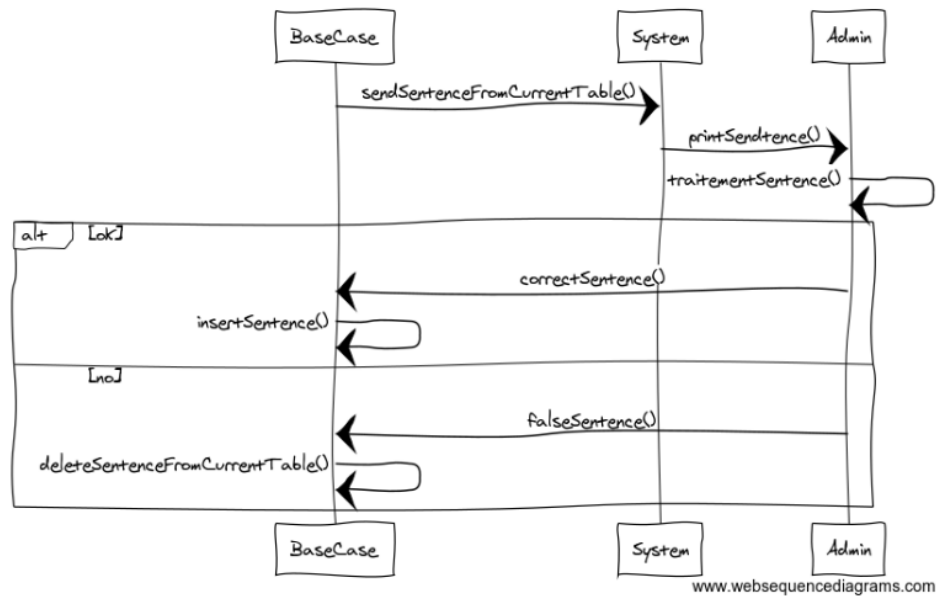
(\*\* propose correction\*\*)



L'utilisateur propose une phrase, le système l'envoie aux bases de cas, qui le stock dans une table courant.

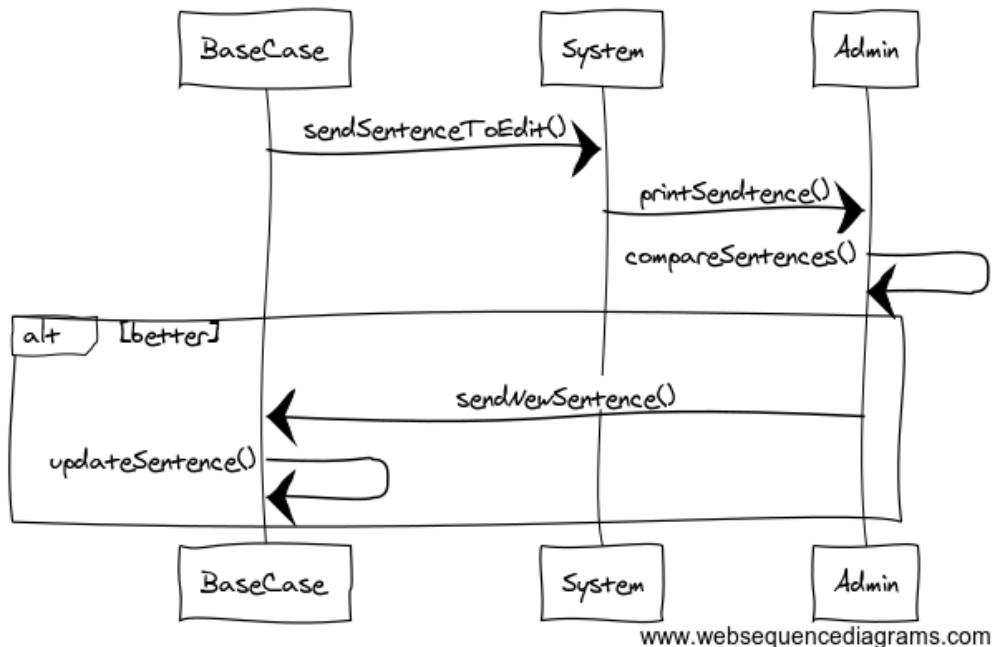


(\*\*validate a sentence \*\*)



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.

(\*\* modifications a sentence\*\*)



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

(\*\*Environnement Développement\*\*)

(\*\*NON\*\*)

(\*\*Interface web\*\*)

(\*\*ATTENTION de preference multi-langage\*\*)

(\*\*création de la maquette via mybalsamiq DONE\*\*)

(\*\*en cours php javascript \*\*)