

DICTIONNAIRE INTELLIGENT EN LIGN

DOSSIER DE CONCEPTION

ERKANG ZHANG, YISHU HAN & CHENGXIAN LIN

12/06/2016
V1.0

SOMMAIRE

1.RAPPEL DU CAHIER DES CHARGE.....	2
2. ENCHAINEMENT DES FONCTIONNALITÉS.....	3
3. ANALYSE ET CONCEPTION.....	4
3.1 DIAGRAMMES DE BASE DE DONEES.....	4
3.2 MODELISATION ET DECOMPOSITION.....	5
3.2.1 Éléments Préliminaires	5
3.2.2 Exercices.....	6
3.2.3 Dictionnaire.....	6
4. DESCRIPTION DES FICHIERS.....	7
4.1 INVENTAIRE DES FICHIERS DE PROGRAMME.....	7
4.2 INVENTAIRE DES FICHIERS DE DONNEES.....	7
5. LIVRAISON.....	8
6.BILAN DE PROJET.....	8
7.ANNEXES.....	8

1.RAPPEL DU CAHIER DES CHARGE

L'objectif de ce projet est de réaliser une application web permettant de consulter des mots français et de proposer des parcours d'apprentissage personnalisés au niveau de lexique aux utilisateurs.

L'utilisateur devra pouvoir:

- Créer un compte personnel
- Consulter des mots français
- Consulter son parcours de progression
- Ajouter des mots dans sa liste lexique
- Consulter sa liste de mot
- Faire des exercices au niveau de lexique
- Faire un test pour évaluer son niveau de français
- Lire des articles en français par thème

L'application devra par ailleurs:

- Afficher des explications d'un mot
- Produire des exercices automatiquement
- Produire un test d'évolution automatiquement
- Evoluer le niveau de l'utilisateur selon ses réponses
- Sélectionner tous les jours des articles par thème
- Permettre de créer un espace personnel, y compris liste de lexique, parcours de progression...

2. ENCHAÎNEMENT DES FONCTIONNALITÉS

Partie Dictionnaire:

l'utilisateur épelle le mot dans la barre de recherche

- si le mot n'existe pas

 - retourne une erreur

- sinon, afficher l'explication de ce mot et proposer un test vocabulaire

- si l'utilisateur ajout ce mot dan sa liste de lexique

 - si l'utilisateur est connecté,

 - retourne d'un message de réussite

 - sinon, demander de connecter

 - si posséder un compte déjà

 - si connecter réussi

 - retourne au page l'explication

 - sinon

 - réessayer ou oublier le mot de passe

 - sinon, inscription comme un nouvel utilisateur

Partie Parcours d'apprentissage

l'utilisateur connecte

- si l'utilisateur est connecté

 - retourne d'un message de réussite

 - si l'utilisateur a choisi des thèmes intéressés

 - afficher des articles par thèmes

 - sinon, demande de choisir des thèmes intéressés

- sinon, demander de connecter

 - si posséder un compte déjà

 - si connecter réussi

 - retourne au page l'explication

 - sinon

 - réessayer ou retrouver le mot de passe(cas oublier le mot de passe)

 - sinon(pas de compte)

 - inscription comme un nouvel utilisateur

 - l'utilisateur doit obligatoirement compléter le premier test

 - attente les réponses

 - selon les réponses

 - si les réponses sont complètes

 - évoluer le niveau

 - sinon

 - affiche d'un avertissement et attente les réponses complètes

Partie Test

attente de choix

- si le choix est correcte

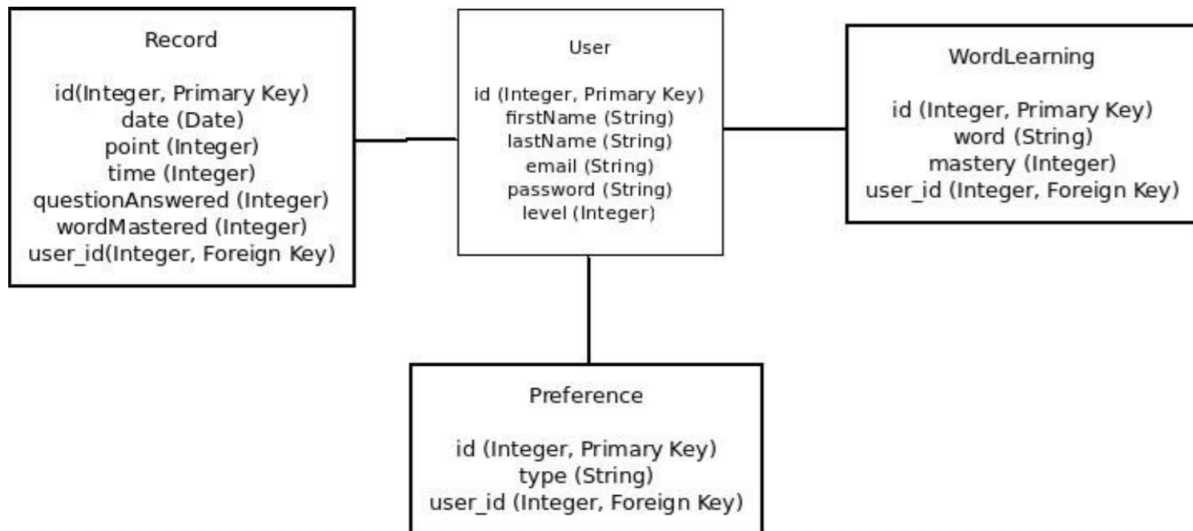
 - affiche « bravo »

- sinon attente de choix

3. ANALYSE ET CONCEPTION

3.1 DIAGRAMMES DE BASE DE DONEES

Dans la suite on représentera la structure de base de données, il s'agit d'un espace d'utilisateur personnalisé comprenant 4 entités, ce qui permet de créer un parcours d'apprentissages et consulter des statistiques d'utilisateur.



User: Les informations de l'utilisateur, y compris son identifiant, nom et prénom, adresse mail, mot de passe et niveau de langue.

WordLearning: L'information qui montre si l'utilisateur connaît bien un mot.

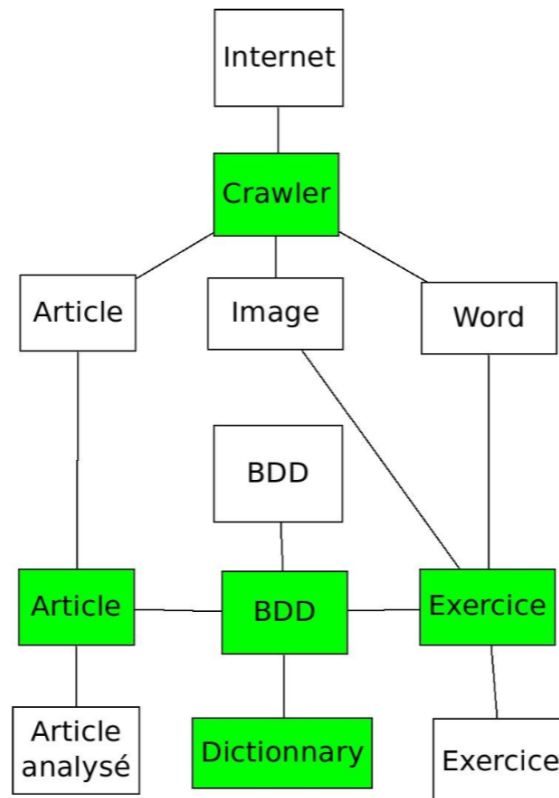
Record: Des statistiques de l'utilisateur sur l'apprentissage de mot

Preference: L'information qui montre des thèmes que l'utilisateur s'intéresse

3.2 MODELISATION ET DECOMPOSITION

On applique Flash comme le framework web, comme il est plus flexible et moins lourd que Django. La plupart d'information est venue de wikidictionnaire, à l'aide d'un outil python très pratique Crawler, qui nous permet de récupérer des informations sur un mot de wikidictionnaire. (pour notre projet non lucratif et privé).

Le schéma présente ci-dessous:



3.2.1 Éléments Préliminaires

Il y a 3 sous-module:

- CrawlArticle
Entrée : String indiquant le thème
Sortie : Objet Article (titre, résumé, contenu)
Site Cible : 20 minutes
Articles libres à accéder
Pages HTML très structurées
- CrawlImage
Entrée : String mot
Sortie : String URL de l'image correspondant au mot
Site cible : <http://lemoteur.orange.fr>
- CrawlWord
Entrée : String mot

Sortie : Objet Word contenant tous les informations du mot (définitions, phrases, synonyme, antonyme)

3.2.2 Exercices

Pour des différents genres d'exercice, on applique un module dit « ExerciseGenerator » :

ce module nous permet de créer un exercice par un générateur aléatoire

Entrée : String mot

Sortie : Objet Exercice (énoncé, choix, bonne réponse)

et y compris plusieurs sous-modules:

- BlankFillExerciseGenerator: choisir la bonne réponse pour le bon sens
- SynonymExerciseGenerator: choisir le synonyme
- AntonymExerciseGenerator: choisir le antonyme
- ImageExerciseGenerator: choisir le bon image
- LevelTest

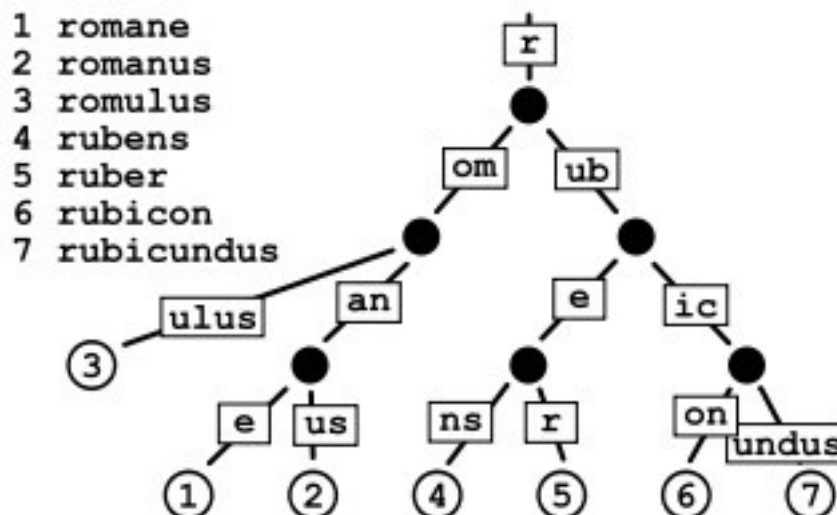
30 niveaux au total

chaque niveau représente une intervalle de fréquence

Chaque intervalle contient le même nombre de mots.

3.2.3 Dictionnaire

Afficher le mot exact n'est pas suffisant, il faut être capable de rechercher le préfixe quand même. Pour trouver le préfixe plus rapide, on applique l'algorithme s'appelant SWAN



4. DESCRIPTION DES FICHIERS

4.1 INVENTAIRE DES FICHIERS DE PROGRAMME

fichier	role
analyseArticle.py	implémenter des méthodes pour analyser les articles
connectBDD.py	offrir une méthode pratique pour connecter la BDD
crawlArticle.py	recupérer des articles d'un site
crawlImage.py	recupérer des images d'un site
crawlWord.py	recupérer des vocabulaires d'un site
dictionary.py	trouver le préfixe, la lemme du mot
exerciceGenerator.py	Générer des exercices
levelTest.py	Générer 30 exercices pour tester le niveau

4.2 <NOM DE CLASSE OU DE MODULE>(<NOM DU FICHIER>)

Ne s'applique pas

4.2 INVENTAIRE DES FICHIERS DE DONNEES

fichier	role
dawgFille.dawg	stocker l'objet DAWG pour trouver les mots ayant le meme préfixe

5. LIVRAISON

Les fonctionnalités prévues seront implémentées en suivant le plus possible les spécifications. La livraison se fera de deux manières.

- L'équipe fournira au client une archive contenant tous les fichiers nécessaires.
- L'équipe mettra en ligne la version la plus récente sur le répertoire Github du projet (voir en annexe)

La livraison en main propre sera constituée:

- Les sources du projet
- D'un fichier README.md

La livraison sur le répertoire Git sera constituée:

- Des sources du projet dans la branche principale
- D'un fichier README.md dans la branche principale, visible depuis la page principale du projet.

6. BILAN DE PROJET

Finalement, nous avons pu réaliser la grande majorité des fonctionnalités dans le projet. Faute de temps, nous n'avons pas forcément pu réaliser tous les fonctionnalités que nous aurions pu désirer. Ce projet nous a permis d'apprendre plein de choses comme Python, Base de donnée, HTML CSS et Javascript. On souhaite que notre projet soit un exemple pour d'autres personnes qui voudraient développer une meilleure application sur le même sujet que notre projet.

7. ANNEXES

Répertoire Git du projet: <https://github.com/xiaoPierre/vocabulary>