

06 janvier au 07 février 2020

NOTE DE SYNTHÈSE

Raphael Trabelsi



BTS SIO SLAM
LILMOD & LELAMED

Sommaire

CONTEXTE	2
PRESENTATION DE L'ASSOCIATION	2
PRESENTATION DES PROJETS	2
PRESENTATION DE L'EQUIPE	3
EXPRESSION DES BESOINS	4
DESCRIPTION DE L'EXISTANT	4
LISTE DES BESOINS	4
CONTRAINTES A RESPECTER	5
DEFINITION DES RESSOURCES MATERIELLES ET LOGICIELLES	5
ANALYSE	6
CAS D'UTILISATIONS	6
SPECIFICATION DES IHM	9
MODELE DE DONNEES	12
REALISATION	13
CHOIX TECHNIQUES	13
CONCEPTION DU SITE WEB	13
PLANNING DE REALISATION	16
EXEMPLES DE CODAGES DU SITE WEB	17
CONCLUSION	23

Contexte

○ Présentation de l'association

Lilmod & Lelamed est une association à but non lucratif.

Elle possède un site internet destiné à l'apprentissage de l'hébreu, de la culture hébraïque à travers l'histoire au travers de sa bibliothèque de jeux, d'exemples et de leçons. L'apprentissage de l'hébreu (lecture, sens des mots, grammaire) constitue le premier niveau d'apprentissage. L'application permet d'apprendre en jouant, ce qui facilite la découverte des mots et de leurs sens, en proposant des exercices et un mode de révision s'appuyant sur de nombreux exemples et de nombreuses illustrations. L'application est destinée à un public francophone, bien que l'interface supporte en plus du français, l'anglais. Elle est orientée pour un usage récréatif et un public, en premier lieu d'enfants. Elle propose également des contenus pour les plus grands.

Le site a été créé en 2015 et compte aujourd'hui plus de 1000 connexions uniques par jours, tout type de supports confondus.

○ Présentation des projets

L'association possède une application web mettant à disposition de l'utilisateur de nombreuses fonctionnalités.

En hébreu, chaque lettre possède une valeur numérique qui lui est propre. Dans le cadre de l'apprentissage de cette langue, l'utilisateur peut être amené à rechercher la valeur numérique d'une lettre ou d'un mot. Cette valeur permet dans certains cas d'avoir plus d'informations sur la nature du mot. Il existe plusieurs façons de trouver cette valeur, c'est pourquoi, un document listant plusieurs méthodologies de calcul avec des exemples, nous a été fourni.

Il nous a donc été demandé de rajouter un module permettant à l'utilisateur d'entrer une lettre ou un mot afin d'en obtenir la valeur numérique. Etant donné qu'il existe de nombreuses manières de calculer cette valeur, nous ne devons en afficher que six.

Un deuxième module doit également être développé, permettant à l'utilisateur d'afficher l'ensemble des psaumes (poèmes liturgiques écrit par le Roi David). Cet affichage peut se faire sous forme de liste ou par thème. Le module doit également proposer de réaliser des

« chaînes ». Il s'agit d'une manière de terminer la lecture du livre des psaumes en répartissant chaque chapitre entre plusieurs personnes.

○ Présentation de l'équipe

L'application est développée par des bénévoles, des membres de l'association et des étudiants en informatique.

❖ Les fondateurs

Laurent Hadjadj : Président et développeur du site

Daniel Mazouz : Secrétaire général

Claude Hadjadj : Comptable

❖ Notre équipe

Il nous a été donné de créer deux modules pour le site. Nous nous sommes donc réparti la tâche en 2 équipes :

Module Guématria

Nathan Hayoun et moi-même.

Module Psaumes

Guila Bismuth, Shanna Hagege et Yariv Sarfati

Expression des besoins

○ Description de l'existant

L'association possède actuellement un site développé à l'aide du Framework Symfony 3, JavaScript, jQuery et Angular, permettant à l'utilisateur d'apprendre la culture hébraïque. Celui-ci contient une section « Boîte à outils » permettant d'approfondir certaines connaissances ou de faciliter la maîtrise des cours présents sur le site. De nombreux modules sont ainsi disponibles au sein de cette section. Le travail qui nous a été demandé consiste donc à créer un module de calcul de valeur numérique en partant d'un document nous présentant différents modes de calcul.

○ Liste des besoins

- Réalisation d'un module de calcul de valeur numérique d'une lettre ou d'un mot, accessible via un formulaire dynamique.
- Donner la possibilité à l'utilisateur d'afficher simultanément différentes méthodes de calcul.
- Donner la possibilité d'afficher les détails de chaque calcul.
- Ajout d'un bouton permettant de garder constamment les détails des calculs affichés.
- Affichage d'un bouton sur chaque méthode de calcul permettant d'en savoir plus sur son fonctionnement.
- Retirer les voyelles de toutes les lettres saisies (en hébreu, chaque lettre peut posséder des voyelles représentées par des signes autour d'elle, permettant d'indiquer comment la lire).
- Ignorer les caractères non-hébraïques et les espaces lors de la saisie.
- Afficher un clavier virtuel hébraïque pour les utilisateurs utilisant un pc (puisque sur mobile, l'ajout d'un clavier est plus simple à mettre en place).
- Ajout d'une fonction « impression » permettant de n'imprimer que les résultats du calcul.
- Ajout d'un bouton « Sources » expliquant le fonctionnement du module.
- Rendre la fonctionnalité responsive (adaptative) sur tout type d'appareil.
- Réaliser une API afin de pouvoir utiliser la fonction de calcul sur l'ensemble du site.

○ Contraintes à respecter

Ressource : Afin de faciliter le développement du module, les fichiers fournis ne sont que ceux ayant un lien avec notre projet.

Fonctionnelle : L'expérience utilisateur se doit d'être la plus fluide possible.








Technique : L'interface devra être développée en respectant les conventions de Symfony 3. Afin de rendre l'expérience utilisateur plus agréable, l'utilisation de Ajax et jQuery est essentielle.

Ergonomique : Le site se doit d'afficher un design dit « responsive » afin de s'adapter à tous les formats d'écrans.

○ Définition des ressources matérielles et logicielles

Je dispose des ressources matérielles et logicielles suivantes :

- Mon ordinateur portable personnel fonctionnant sous Windows 10 Pro.
- Serveur : **Debian 9.6, Nginx 1.10.3**
- Serveur d'application : **Php 7, Symfony 3**
- Base de données : **MongoDB 4.2.1, SQLite 3.16**
- Interface de développement : **Visual Studio Code**

SERVEUR	SERVEUR D'APPLICATION	BASE DE DONNEES	INTERFACE DE DÉVELOPPEMENT
 	  Symfony	 mongoDB  SQLite	 Visual Studio Code

Analyse

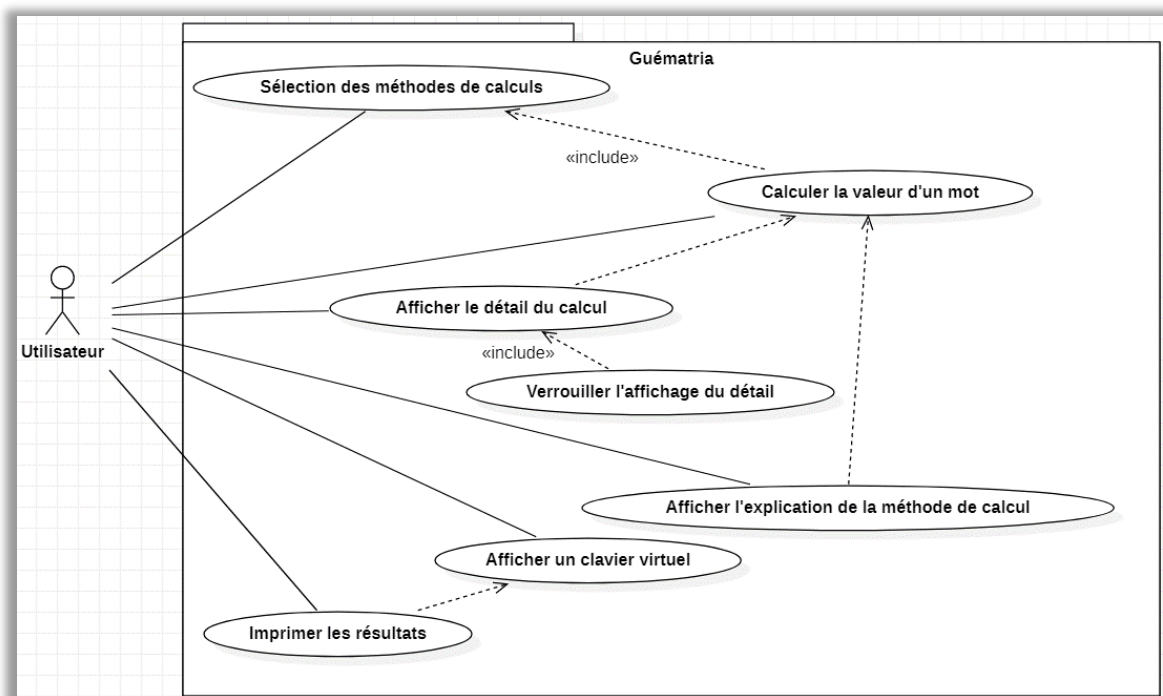
Toutes les fonctionnalités du module Guématria sont regroupées sur une seule et même page afin de faciliter la navigation pour l'utilisateur. Ainsi, le choix de la méthode de calcul s'effectue sur la même page que le résultat du calcul.

Le module peut être utilisé par n'importe quel utilisateur.

Ils peuvent :

- Calculer la valeur d'une lettre ou d'un mot.
- Sélectionner différentes méthodes de calculs.
- Afficher un clavier virtuel en cas de besoin.
- Imprimer les résultats obtenus.

○ Cas d'utilisations



Choisir un mode de calcul

1. Le système affiche un menu présentant les différents modes de calculs disponibles.
2. L'utilisateur sélectionne la/les méthode(s) de calcul désirée(s).
3. Le système affiche les méthodes de calculs sélectionnées lorsque l'utilisateur entre une lettre ou un mot.

Calculer une valeur

Précondition : Au moins une méthode de calcul est sélectionnée.

1. Le système affiche une barre de saisie.
2. L'utilisateur entre la lettre/le mot souhaité.
3. Le système affiche le/les résultat(s) sous la barre.

Exception :

- 3.1 Si l'utilisateur entre un caractère autre qu'en hébreu, celui-ci est retiré.
- 3.2 Le système vérifie si la lettre ou le mot hébreu est ponctué de voyelles afin de les retirer (uniquement les voyelles).

Afficher le détail d'un calcul

Précondition : L'utilisateur a entré une lettre ou un mot dans la barre de saisie.

1. Le système affiche le/les résultat(s) sous la barre.
2. L'utilisateur passe son curseur sur une méthode de calcul.
3. Le système affiche sous le résultat le détail du calcul correspondant.

Verrouiller l'affichage du détail

Précondition : L'utilisateur a entré une lettre ou un mot dans la barre de saisie.

1. Le système affiche le/les résultat(s) sous la barre.
2. L'utilisateur sélectionne l'icône du cadenas (🔒) présent près du résultat.
3. Le système garde l'affichage du détail ouvert.

Exception :

- 3.1 L'utilisateur clique une nouvelle fois sur l'icône du cadenas (🔒).
- 3.2 Le système retire alors le détail du calcul lorsque le curseur de l'utilisateur n'est plus sur le résultat.

Afficher l'explication de la méthode de calcul

Précondition : L'utilisateur a entré une lettre ou un mot dans la barre de saisie.

1. Le système affiche le/les résultat(s) sous la barre.
2. L'utilisateur sélectionne l'icône point d'interrogation (?) présent près du résultat.
3. Le système affiche l'explication de la méthode de calcul sous forme de fenêtre devant le résultat (modal).

Exception :

- 3.1 L'utilisateur clique en dehors de la fenêtre (modal) ou sur la croix (X).
- 3.2 Le système n'affiche plus la fenêtre (modal) et rend de nouveau visible les résultats.

Afficher un clavier virtuel

Précondition : L'utilisateur se trouve sur un ordinateur.

1. Le système affiche un bouton « Clavier ».
2. L'utilisateur clique dessus.
3. Le système affiche un clavier directement sur le site web.

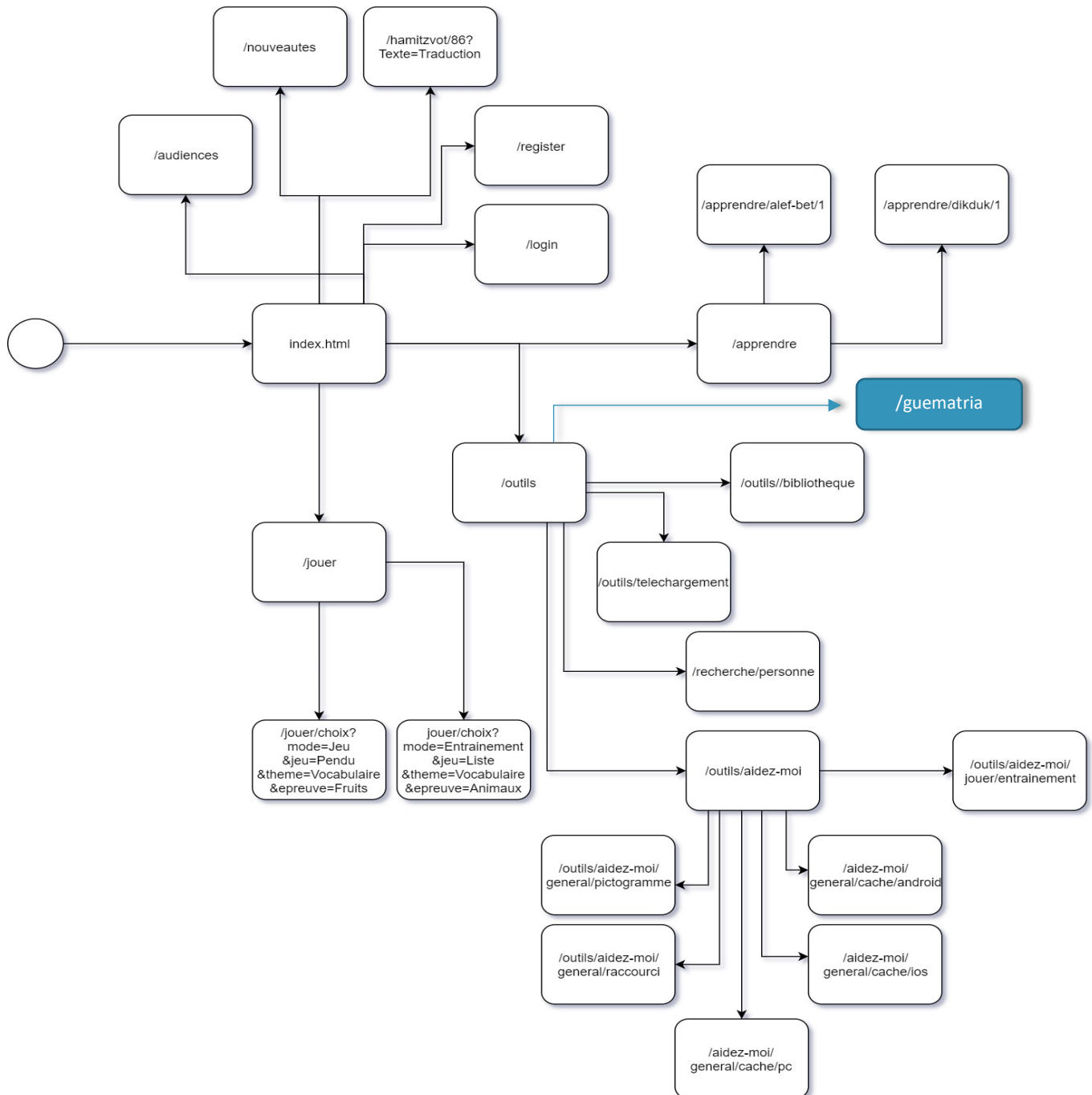
Imprimer les résultats

Précondition : L'utilisateur se trouve sur un ordinateur et affiche le clavier.

1. Le système affiche un bouton d'impression (🖨) sur le clavier
2. L'utilisateur clique dessus.
3. Le système affiche la page d'impression ne contenant que le/les calcul(s) détaillé(s).

○ Spécification des IHM

• Arborescence des pages du site web



- Exemples de pages du site web

- ❖ Calculer la valeur d'une lettre ou d'un mot

Cette portion de page est affichée à l'utilisateur lorsqu'il sélectionne le module « Calcul de Guématria » dans le menu « outils » (non présenté dans le cadre de cette note de synthèse). Elle lui permet d'entrer un mot et de sélectionner la/les méthode(s) de calcul souhaitée(s).

Si l'utilisateur se trouve sur un ordinateur, il peut sélectionner le bouton « clavier » afin d'afficher un clavier virtuel sous la barre de saisie.

Compteur de lettres.

L'utilisateur a la possibilité de sélectionner une ou plusieurs méthodes de calcul à la fois.

Barre de saisie dans laquelle l'utilisateur peut entrer un mot ou une lettre. Les voyelles et les caractères d'une autre langue que l'hébreu sont dynamiquement retirés.

❖ Affichage des résultats

Les voyelles ne sont pas prises en compte

שלום Clavier

Lettres : 4 / 150

RAGUIL 376

DÉTAIL DU CALCUL

שלום
 ם + ו + ל + ש
 40 + 6 + 30 + 300

INTEGRATION 536

TM HAKOL LEI

L'affichage du résultat se fait sous la barre de saisie. Le calcul s'effectue dynamiquement, au fur et à mesure de la saisie du mot par l'utilisateur. De la même façon, les résultats sont affichés dynamiquement en fonction des méthodes sélectionnées plus haut sur la page.

Affichage du résultat. Par défaut, l'affichage est en mode simplifié. L'utilisateur peut afficher le détail du calcul en passant son curseur sur le résultat.

Permet de garder affiché les détails du calcul.

Affiche le détail lettre par lettre du calcul.

Permet d'afficher sous forme de fenêtre (modal) des informations complémentaires sur la méthode de calcul (récupérées depuis la base de données).

RAGUIL

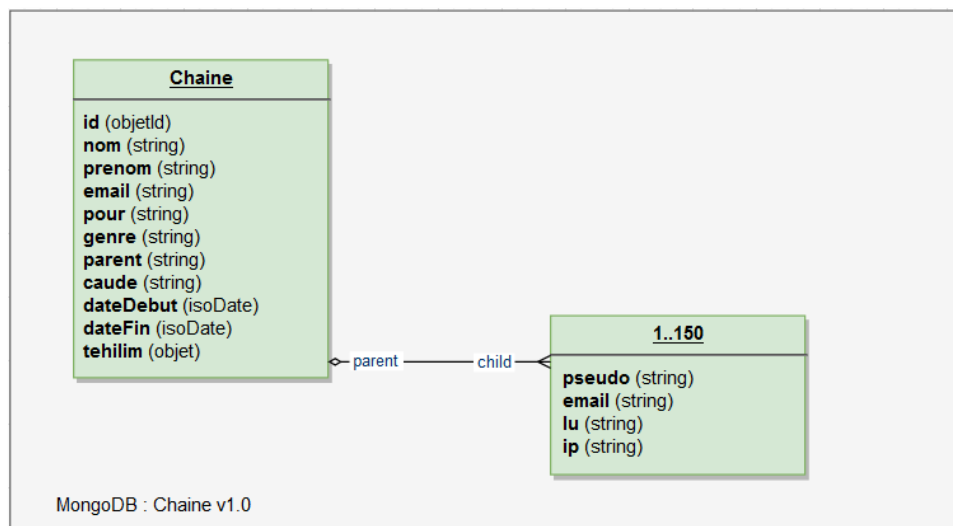
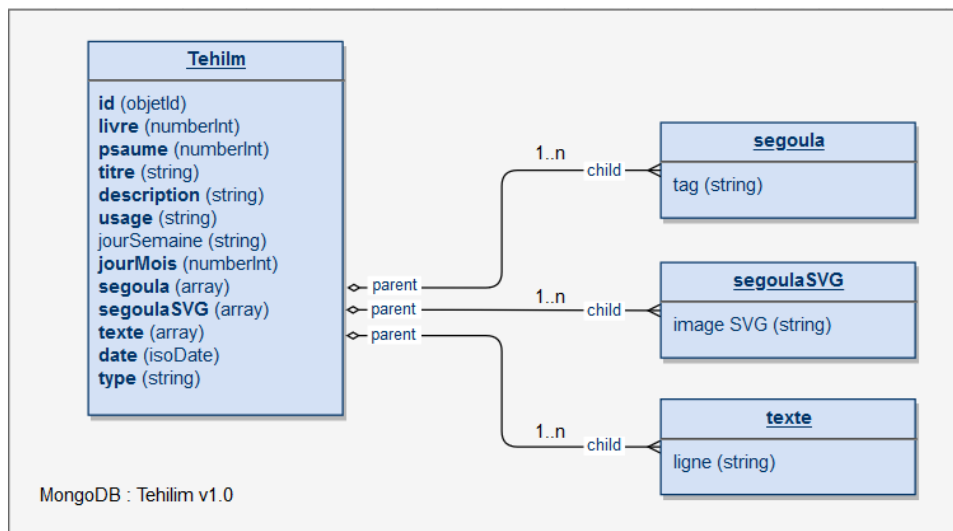
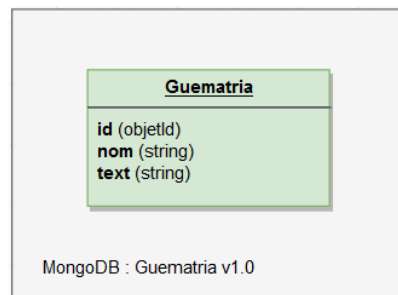
C'est la méthode traditionnelle pour l'attribution des valeurs des lettres. « Mispar gadol » signifie « grande valeur », on l'appelle aussi le « mispar raguil ».

Selon cette méthode, la lettre Shin, ש, qui vaut 300 est équivalente du nombre obtenu par la somme des valeurs numériques des lettres des mots אלהים רוח, Ruach Elohim, Esprit ou Souffle d'Elohim ; et est ainsi un symbole de l'esprit d'Elohim. Resh=200, Vav=6, Heth=8, Aleph=1, Lamed=30, Hé=5, Iod=10, Mem=40 ; total = 300.

○ Modèle de données

MongoDB est une base de données NoSQL. Celle-ci n'étant pas relationnelle, il est impossible de réaliser un MCD. Son modèle d'organisation s'appuie sur la notion de document, de modèle clé-valeur et de collection.

Voici les trois tables créées dans le cadre de nos projets :



Réalisation

○ Choix techniques

Langages utilisés : HTML5, CSS3, PHP 7 (orienté objet), JavaScript, jQuery 3, Ajax, Twig



Framework : **Symfony 3**



Symfony

Editeur : **Visual Studio Code**



Visual Studio Code

Bibliothèque utilisée : **Zurb Fondation 6.5.3, Doctrine (accès à la base de données)**



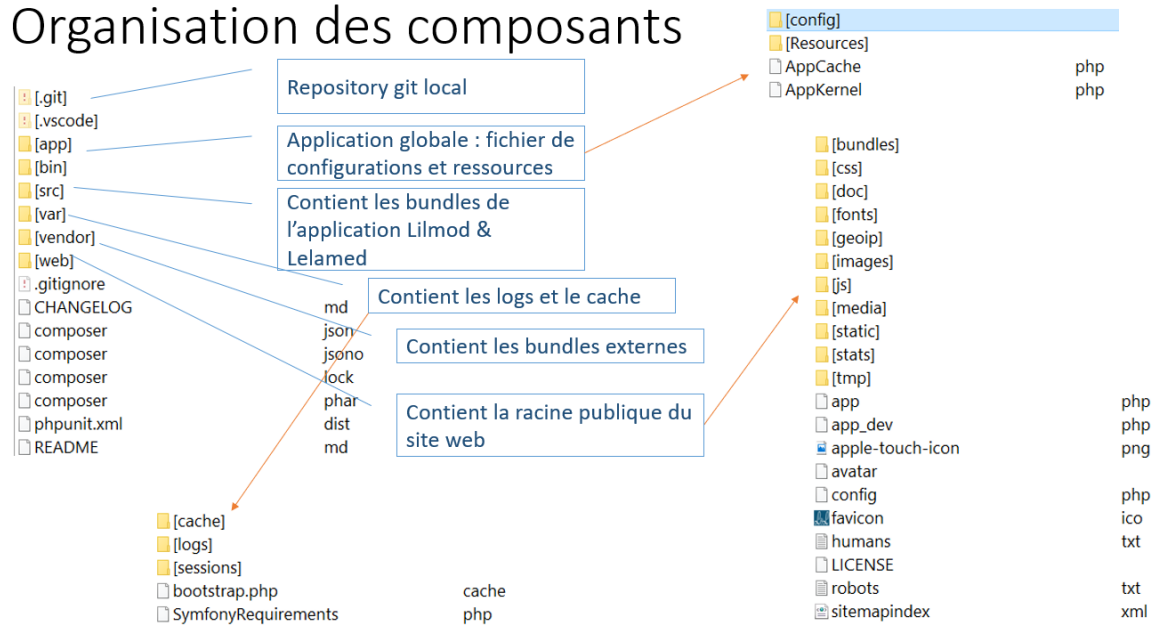
Outils : **Symfony Tools (debug), Git**

○ Conception du site web

Le site a été développé avec le Framework Symfony et suit un schéma MVC (Model – Vue – Contrôleur). Seule une partie du code du site nous a été remise afin de faciliter le développement de nos fonctionnalités. Le modèle MVC déjà mis en place nous a permis d'avoir une meilleure lisibilité de l'ensemble des fichiers et de leurs interactions au sein du site web.

❖ Fichiers existants sur le site web

Organisation des composants



/Src/PortailBundle/Controller

```

v src
v PortailBundle
v Controller
  Administration.php
  AjaxAfficheReferenceController.php
  AjaxAlefBetController.php
  AjaxAvatarController.php
  AjaxDikdukController.php
  AjaxJeuxListeController.php
  AjaxJeuxPenduController.php
  AjaxJeuxWordSearchController.php
  AjaxLivreController.php
  AjaxPetitMotController.php
  AjaxRechercheController.php
  ApprendreController.php
  ChangePasswordController.php
  ChangePasswordController.php.ori
  ConstructionController.php
  
```

Liste des contrôleurs (33).
En CamelCase majeur.

Ex. Contrôleurs Ajax.

```

EtudeController.php
FooterController.php
JouerController.php
OutilAidezMoiController.php
OutilController.php
OutilLivreController.php
PaypalController.php
PortailController.php
ProfileController.php
ProfileController.php.ori
RegistrationController.php
RegistrationController.php.ori
ResettingController.php
ResettingController.php.ori
SecurityController.php
SecurityController.php.ori
ZTestController.php
  
```

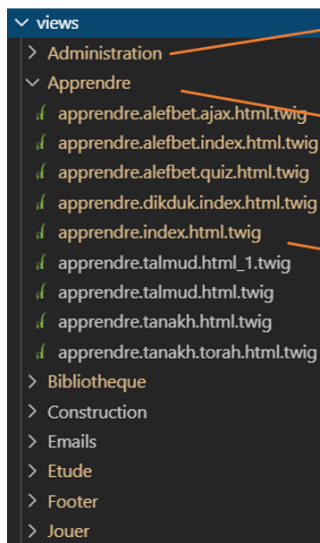
Class FOS.
Gestion des utilisateurs.

Class de tests.

Le contrôleur contient la logique métier et l'accès aux données.

Le contrôleur porte les règles de gestion, les routes et la sécurité.

Src/PortailBundle/Ressources/view



Module administration

Module Apprendre

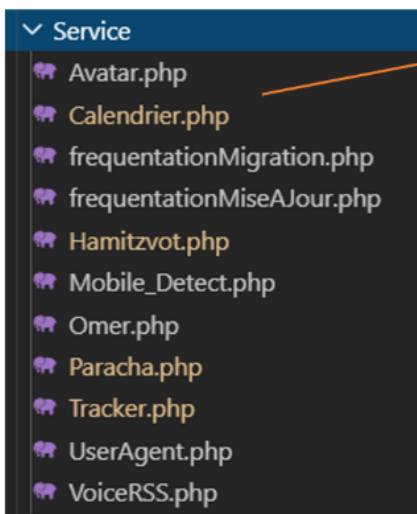
Pages de présentation au format twig pour le module.

Il existe un dossier pour chaque module qui regroupe les pages HTML de présentation (template TWIG).

Le nommage suit la règle :

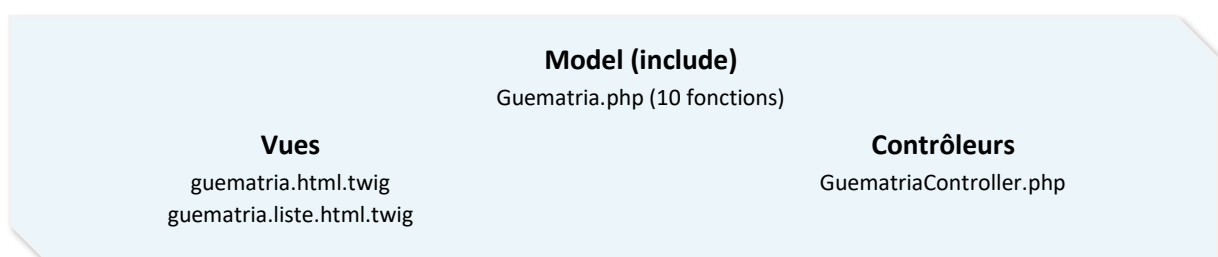
- *module.page.index.html.twig* → page principal
- *module.page.html.twig* → sous-page
- *module.page.ajax.html.twig* → fragment pour Ajax

Src/PortailBundle/Services



Liste des services PHP utilisés dans les différents contrôleurs.

❖ Fichiers créés pour le projet suivant le model MVC



○ Exemples de codages du site web

❖ **Controller (GuematriaController.php)**

Le controller récupère le texte saisi par l'utilisateur, puis effectue les calculs nécessaires grâce au service Guematria.php (page 21). Le résultat est ensuite renvoyé en Json pour l'Ajax.

```
/**
 * @Route("/guematria/ajax", name="guematriaAjax", options = { "utf8": true })
 */
public function AfficherAlphabetAjax(Request $request)
{
    $guematria = new Guematria;
    $data = $request->get('data');
    $methods = $request->get('methods');
    $mot = $data["data"];
    $valeurDeReponse = array();
    $mot = $guematria->clean($mot);
    foreach ($methods as $method) {
        $valeur = $guematria->$method($mot);
        if ($valeur != 0) {
            if ($method == "anteriorite") {
                $method = "antériorité";
            } elseif ($method == "imhakollel") {
                $method = "Im haKollel";
            }
            $valeurDeReponse[$method]["titre"] = $method;
            $valeurDeReponse[$method]["valeur"] = $valeur["valeur"];
            $valeurDeReponse[$method]["detail"] = $valeur["detail"];
        }
    }
    $aucun = 0;
    if (empty($valeurDeReponse)) {
        $aucun = 1; // Si aucune methode de calcul n'est selectionnée
    }
    $mot = implode($mot);
    $response = new JsonResponse();
    $response->setData($this->renderView('PortailBundle:Guematria:guematria.liste.html.twig',
        ['resultats' => $valeurDeReponse,
         'aucun' => $aucun,
         'mot' => $mot]));
    return $response;
}
```

❖ API (GuematriaController.php)

L'API a été réalisée dans le controller. Celle-ci peut être utilisée sur l'ensemble du site lorsque l'utilisateur sélectionne un mot par exemple. La valeur de ce mot est ensuite calculée avant d'être renvoyée en JSON afin de l'afficher dynamiquement sur une autre page pendant la navigation (en Ajax).

```
/**
 * @Route("/guematria/api" , name ="gueMatriaApi", options = { "utf8": true })
 */

public function apiGuematria(Request $request)
{
    $guematria = new Guematria;
    $data = $request->get('data');
    $mot = $data["data"];
    $mot = $guematria->clean($mot);
    $valeur = $guematria->raguil($mot);
    $laValeur = $valeur["valeur"];
    $detail = $valeur["detail"];
    $response = new JsonResponse();
    $response->setData(['valeur' => $laValeur, 'detail' => $detail]);
    return $response;
}
}
```

❖ Affichage des résultats (vue : guematria.liste.html.twig)

Cette vue permet l'affichage des résultats, du détail et des infos concernant les méthodes de calcul (affichage d'un modal).

```
{# Affichage des resultats #}
{% if resultats is defined %}
  {% for resultat in resultats %}
    <article class="grid-container grid-x event">
      <div class="callout callout-hover-reveal cell" data-callout-hover-reveal
        style="text-align: center; clear both;">
        <div class="callout-body" id="texte">
          {# modal #}
          <p style="float: right"><a id="infos-{{ resultat.titre|capitalize }}"
            class="float-right lilmod-question02"
            style="font-size:2rem;"
            tabindex="0"
            data-open="DetailGuematriaModal"
            onclick="afficher('{{ resultat.titre|capitalize }}')"></a></p>

          {# modal #}
          <h3 id="titre" style="text-align: center;">
            {{ resultat.titre }}
          </h3>
          <div class="OpenSans text-justify"
            style="padding-left:0.5rem; padding-right:0.5rem; color:#00445b;text-
            align: center;">
            {{ resultat.valeur }}
          </div>
          <span id="detail{{ resultat.titre|replace({' ': ''}) }}" class="float-right lilmod-
            cadenasouvert"
            style="font-size:2rem;"
            onclick="ouvrir('#call{{ resultat.titre|replace({' ': ''}) }}','#{detail{{ resultat.
            titre|replace({' ': ''}) }}')"></span>
          <br/>
          <div class="callout-footer" id="call{{ resultat.titre|replace({' ': ''}) }}"
            style="text-align:center; clear: both; margin-right: 2rem">
          <h4><b>Détail du calcul</b></h4>
          <div style="color:#00445b; overflow-wrap: break-word;">

          <span class="motEntre">
            {{ mot }}
          </span>
          <br>
          {% set compte = 1 %}
          {% for detail in resultat.detail %}
            {{ detail.lettre }}
            {% if compte < resultat.detail|length %}
              +
              {% set compte = compte + 1 %}
            {% endif %}
          {% endfor %}
          <br>
          {% set compte = 1 %}
          {% for detail in resultat.detail|reverse %}
            {{ detail.valeur }}
            {% if compte < resultat.detail|length %}
              +
              {% set compte = compte + 1 %}
            {% endif %}
          {% endfor %}
        </div>
      </div>
    </article>
  {% endfor %}
{% endif %}
```

```

        </div>
    </div>
</div>
</div>
</article>
<hr/>
{% endfor %}
{% endif %}

<script type="text/javascript">
$(function () {
    $('[data-callout-hover-reveal]').hover(function () {
        $(this).find('.callout-footer').slideDown(250);
    }, function () {
        $(this).find('.callout-footer').slideUp(250);
    });
});

function ouvrir(id, icone) {
    var context = document.querySelector(id);
    if (context.classList.contains("callout-footer")) {
        context.classList.remove("callout-footer");
        document.querySelector(icone).classList.remove("lilmod-cadenasouvert");
        document.querySelector(icone).classList.add("lilmod-cadenasferme");
    } else {
        context.classList.add("callout-footer");
        document.querySelector(icone).classList.remove("lilmod-cadenasferme");
        document.querySelector(icone).classList.add("lilmod-cadenasouvert");
    }
};

// Pour le modal ajax
function afficher(titre) {
    var data = titre;
    $.ajax({
        type: 'POST',
        url: "{{ path('modal') }}",
        data: {
            'data': data,
        },
        dataType: 'json',
        timeout: 3000,
        success: function (response) {
            $('#NomGuematria').html(response.nom);
            $('#TexteGuematria').html(response.text);
        },
        error: function () {
            $('#resultat').html(
                '<br />\n
                <div class="grid-container ">\n
                <div class="alert callout grid-x" data-closable="slide-out-right">\n
                <p style="color:#e03522;" class="OpenSans cell">Oups! Il y a un problème, je n
                e comprends pas la demande.</p>\n
                <button class="close-button" aria-label="Dismiss alert" type="button" data-
                close><span aria-hidden="true">&times;</span></button>\n
                </div>\n
                </div>'
            ).show("slow");
        }
    });
}
</script>

```

❖ Extrait du service (Guematria.php)

C'est dans ce fichier que l'ensemble des calculs sont effectués. La fonction calcul(\$mot, \$methode) permet de calculer la valeur d'un mot selon une méthode entrée en paramètres. Chaque méthode de calcul a sa fonction pouvant appeler calcul() en cas de nécessité.

```
// Une fonction avoirTableau($methode) est disponible. Elle renvoie un tableau associant une lettre à sa valeur selon la méthode choisie.

public function calcul($mot, $methode)
{
    $j = 0;
    $tableau = $this->avoirTableau($methode);
    $valeurMot = 0;
    foreach ($mot as $i => $value) {
        if (array_key_exists($value, $tableau)) {
            $valeurMot += $tableau[$value];
            if ($methode == "nistar") {
                $detailCalcul[$j]["valeur"] = $tableau[$value] - 1;
            } else {
                $detailCalcul[$j]["valeur"] = $tableau[$value];
            }
            $detailCalcul[$j]["lettre"] = $value;
            $j++;
        }
    }
    if ($valeurMot != null && $detailCalcul != null) {
        $valRetourne["valeur"] = $valeurMot;
        $valRetourne["detail"] = $detailCalcul;
    } else {
        $valRetourne = 0;
    }
    return $valRetourne;
}

// Calcul selon la méthode dite « classique ».
public function raguil($mot)
{
    $resultat = $this->calcul($mot, "raguil");
    return $resultat;
}

// Des fonctions integration, antériorité et nistar sont disponibles sur le site.

// Calcul selon la méthode « quaternion ».
public function quaternion($mot)
{
    $valQuaternion = 0;
    $valRaguilQ = 0;
    $TabRaguil = $this->avoirTableau("raguil");
    $j = 0;
    foreach ($mot as $i => $value) {
        if (array_key_exists($value, $TabRaguil)) {
            $valRaguilQ += $TabRaguil[$value];
            $valQuaternion += $valRaguilQ;
            $detailQuaternion[$j]["lettre"] = $value;
            $detailQuaternion[$j]["valeur"] = $valRaguilQ;
            $j++;
        }
    }
    if ($valQuaternion != 0 && $detailQuaternion != 0) {
```

```

        $valRetourne["valeur"] = $valQuaternion;
        $valRetourne["detail"] = $detailQuaternion;
    } else {
        $valRetourne = 0;
    }
    return $valRetourne;
}

// Calcul selon la méthode « Im Hakollel ».
public function imhakollel($mot)
{
    $valRaguilIm = $this->calcul($mot, "raguil");
    if($valRaguilIm >0) {
        $valRaguilIm["valeur"] += 1;
        $detail = $valRaguilIm["detail"];
        $nbDetail = count($detail);
        $detail[$nbDetail]["valeur"] = 1;
        $detail[$nbDetail]["lettre"] = implode($mot);
        $valRetourne["valeur"] = $valRaguilIm["valeur"];
        $valRetourne["detail"] = $detail;
        return $valRetourne;
    }
    else {
        return 0;
    }
}

```

❖ Fonction de nettoyage de mot (Guematria.php)

Cette fonction permet à l'utilisateur de ne saisir des mots qu'en hébreu, et nettoie chaque lettre saisie de l'ensemble de ses voyelles.

```

public function clean($mot)
{
    $mot = $this->str_split_unicode($mot);
    $alphabet = ["א", "ב", "ג", "ד", "ה", "ו", "ז", "ח", "ט", "י", "כ", "ל", "מ", "נ", "ס", "ע", "פ", "צ", "ק",
    ", "ר", "ש", "ת", "ך", "ם", "ן", "ף", "ץ"];
    $i = 0;
    $tmp = array();
    while ($i < count($mot)) {
        $lettre = $mot[$i];
        if ($lettre == "ב") {
            $lettre = "ב";
        }
        if ($lettre == "כ") {
            $lettre = "כ";
        }
        if ($lettre == "פ") {
            $lettre = "פ";
        }
        if ($lettre == "ש" || $lettre == "ו") {
            $lettre = "ש";
        }
        if (in_array($lettre, $alphabet)) {
            array_push($tmp, $lettre);
        }
        $i++;
    }
    $mot = $tmp;
    unset($tmp);
    return $mot;
}

```

Conclusion

Durant cette seconde expérience dans le monde du travail, j'ai acquis de nombreuses compétences et amélioré mes connaissances dans plusieurs domaines tels que le JavaScript ou l'utilisation de Symfony. De nombreuses difficultés de différentes natures sont survenues durant ce stage telles que la gestion de projets en équipe ou la découverte d'un nouvel environnement de travail (MongoDB, gestion d'un réel projet sous Symfony...).

Nous avons donc, dans un premier temps décidé de nous former la première semaine afin d'avoir une utilisation plus aisée des différents outils mis à notre disposition. Puis, lorsque nous avons divisé les groupes afin de mettre en place chacun de nos projets, nous avons (Nathan et moi-même) décidé d'organiser la réalisation des tâches à accomplir afin d'avancer plus efficacement. J'ai donc acquis grâce à ce second stage des notions dans la gestion de projets en équipe, et le respect scrupuleux des délais pour chacune des tâches à accomplir.

Le module web créé a pour finalité d'être utilisé par n'importe quelle personne souhaitant approfondir ses connaissances dans la langue hébraïque. Toutes les fonctions demandées ont été mises en place dans les temps, ce qui nous a laissé la possibilité d'ajouter des fonctionnalités supplémentaires telles que la fonction d'impression des résultats. Cet ajout a été très apprécié par notre maître de stage. Ce dernier nous a assuré que notre projet sera mis en production dans quelques mois.

L'outil mis en place a été pensé pour être facile à maintenir, et à faire évoluer en rajoutant notamment de nouvelles méthodes de calcul.