

Rapport de Stage BTS SIO Option SLAM

Du 28 novembre 2016 au 23 décembre 2016



IMAGINA
- Connect -

**Vivez une expérience
connectée**

Charlotte DAVID
GRETA Vannes
Année 2016-2017

SOMMAIRE

Introduction.....	2
--------------------------	----------

1. Présentation de Imagina_.....	3 à 5
---	--------------

1-1 Présentation de l'entreprise.....	3
---------------------------------------	---

1-2 Descriptif du projet «Imagina Connect».....	4
---	---

2.Présentation du travail effectué.....	6 à 14
--	---------------

2-1 Projet Personnel.....	6
---------------------------	---

2-2 Expérience apportée.....	14
------------------------------	----

INTRODUCTION

Dans le cadre de ma formation en BTS Services Informatiques aux Organisations option Solutions logiciels et applications métiers, j'ai effectué un stage au sein de la Start up Imagina durant un mois du 28 novembre 2016 au 23 décembre 2016.

Le projet principal qui m'a été confié fût le développement de la barre latérale ou «sidebar» pour le nouveau site de Imagina Connect. Elle devait contenir différents blocs affichant diverses propositions comme des personnes et des lieux à suivre et des intérêts à renseigner.

Dans un premier temps , je présenterai l'entreprise, puis j'expliquerai en quoi consistait mon travail. Je finirai par une conclusion en développant mon point de vue sur cette expérience professionnelle et ce que cela m'a apporté.

1. Présentation de Imagina

1-1. Présentation de l'entreprise

Au commencement en 2013, «Imagina» est un projet personnel de Steve Cotonnec alors qu'il est encore étudiant. Il crée une plateforme qui permettait aux utilisateurs de laisser leur avis sur les magasins ou les restaurants de Vannes.

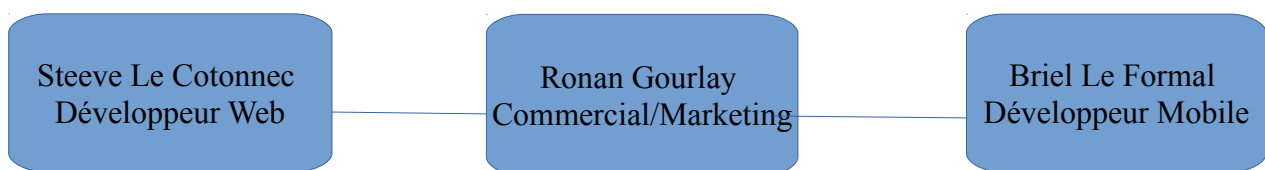
Il s'associe ensuite à deux de ses amis: Brieg Le Formal, développeur mobile et Ronan Gourlay, commercial.

En 2014, Imagina remporte le concours « Start-ubs » qui met à l'honneur les start-up Vannetaises.

En 2016, Imagina lance officiellement son application mobile «Imagina Connect» qui devient le support d'utilisation indispensable pour leur projet. Imagina Connect équipe alors trois festivals en été 2016 dans le Morbihan.

Cependant les revenus de l'entreprise ne leur permet pas actuellement de louer un local , ni de se verser un salaire, ils travaillent donc chez Steve et sont soutenus financièrement par la technopole Vipe de Vannes , la région Bretagne et la banque publique d'investissements.

Organigramme

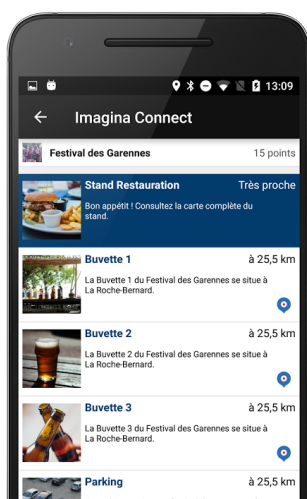


1.2 Descriptif du projet «Imagina Connect»

Le Concept est né à partir du site internet qui permettait aux utilisateurs de laisser leur avis sur les activités commerciales de Vannes.

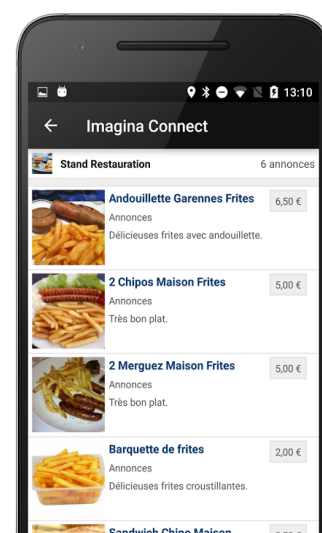
Imagina Connect lance son application mobile dans le but de «connecter» un événement.

D'un point de vue technique, cela nécessite l'installation de puces bluetooth sur chaque stand pour créer les différents point de localisation.



Après activation de la géolocalisation sur son smartphone , l'utilisateur reçoit les informations les plus utiles selon sa position et selon les horaires.

Plus concrètement, d'un point de vue de utilisateur , après avoir installé l'application sur son smartphone , il accède à toutes les informations de l'événement en temps réel (espaces de ventes , animations , concours , concerts en cours .. etc)



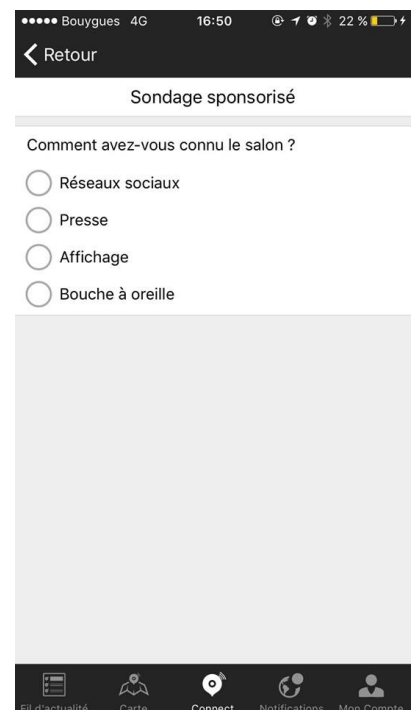


Il peut également partager son expérience avec les autres visiteurs (photos , publications , géolocalisation). Et enfin recevoir des notifications pour être alerté d'éventuelles promotions , rappels et sondages.

D'un point de vue des organisateurs cela permet de transformer son espace et potentiellement d'attirer plus de visiteurs et de développer les ventes.

L'envoi des notification permet de guider , enquêter et remercier les visiteurs.

Et enfin, il permet de faire des statistiques pour identifier la fréquentation , le flux , le temps passé et les informations consultés par les clients.

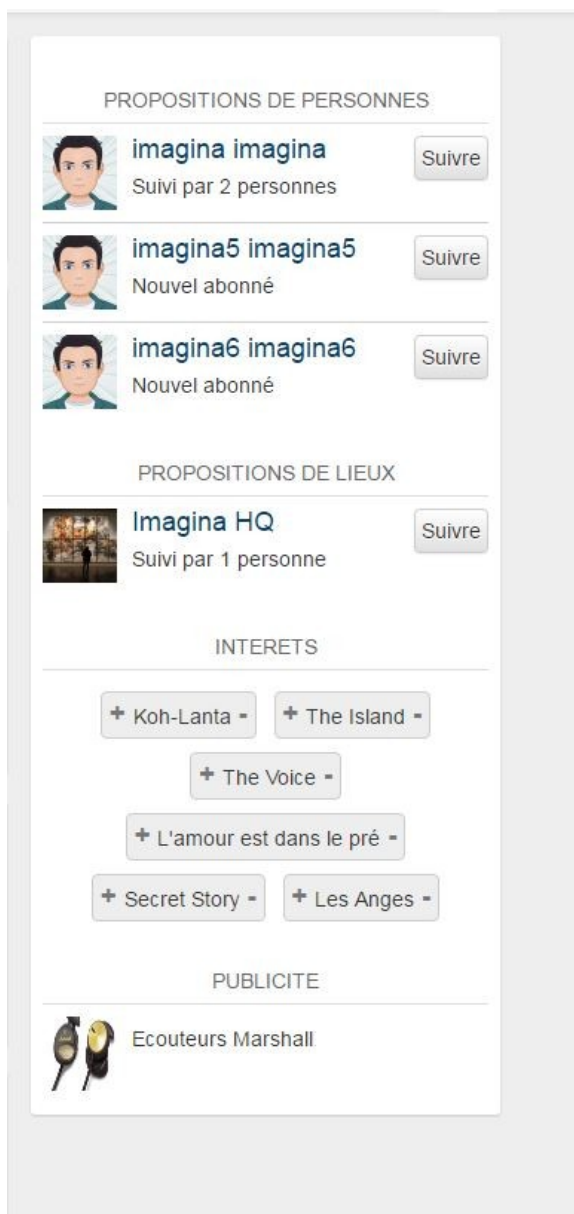


2.Présentation du travail effectué

2.1. Projet Personnel

Après la validation de mon arrivée, mon tuteur a décidé de me confier la mise en place de la sidebar du site internet qui doit sortir au printemps 2016.

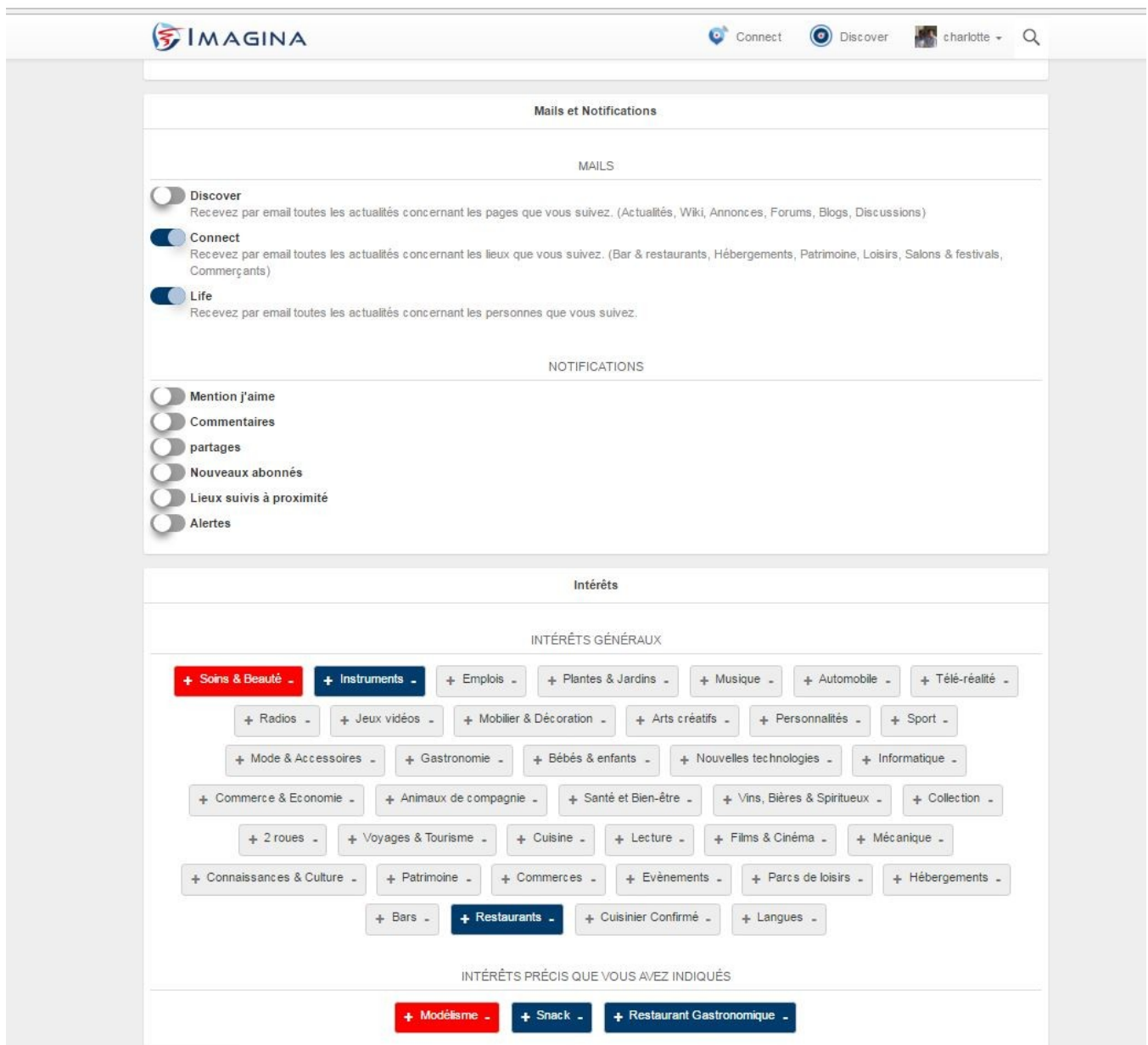
Un des projet de «Imagina connect» est de créer un réseau social. Le principe rejoint celui de Twitter© ou Facebook© avec une sidebar qui est incorporée dans de nombreuses pages du site qui permet à l'utilisateur d'y accéder facilement.



Celle-ci a pour but de présenter des personnes à suivre , des suggestions de lieux et différents intérêts pour personnaliser l'envoi de notification à l'utilisateur . Elle intégrera aussi quelques publicités.

Sidebar à la fin du Stage

Comme j'avais atteint mon objectif défini par mon tuteur avant la fin du stage, j'ai fait en extra la mise en forme de la page «Setting» qui permet à l'utilisateur d'avoir le résumé de ses intérêts «aimés» ou non. L'utilisateur peut également sur cette page choisir ou non l'envoi de mails et de notifications selon différents critères.



Affichage de la page « settings »

2.2 Outils et méthode de travail

Outils et langages utilisés:

- Framework: Symfony3©

Un Framework est un ensemble d'outils constituant les fondations d'un logiciel informatique ou d'applications web. Il est destiné autant à faciliter le travail qu'à augmenter la productivité du programmeur qui l'utilisera.



- Environnement de Développement ou IDE : PhpStorm©

Un IDE est une interface qui permet de développer, compiler et exécuter un programme dans un langage donné.



- Langages Utilisés : HTML 5 , CSS 3 , PHP



- Outil de communication interne : Slack ©



Méthodologie employée :

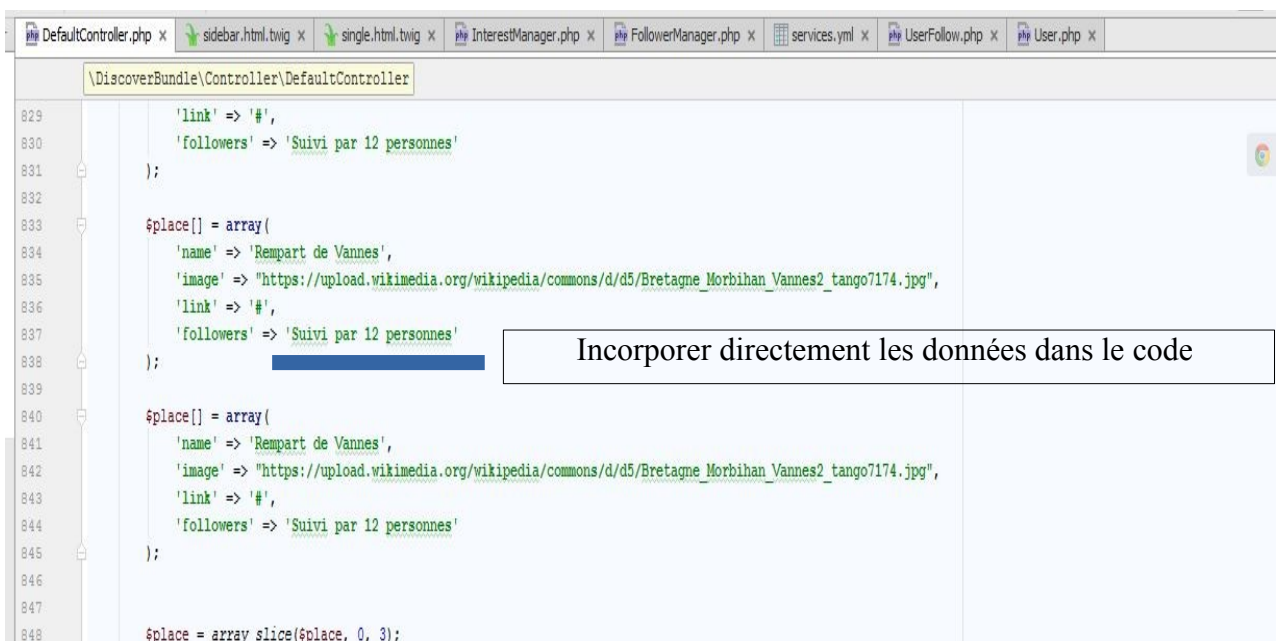
La première semaine m'a permis de prendre mes marques avec PhpStorm et avec l'équipe.

Pour communiquer et retenir les idées plus facilement , nous utilisons le logiciel Slack © qui se présente comme un tchat avec différents groupes (général , développement , marketing par exemple) et qui permet d'envoyer des fichiers, photos.. etc ; Ce qui s'est avéré pratique lorsque mon tuteur était en déplacement ou pour lui envoyer des lignes de code à corriger.

Mon travail durant cette première semaine a été principalement de la mise en page/forme de la sidebar en utilisant HTML et CSS comme langages.

J'ai dans un premier temps codé en dur pour pouvoir ajuster l'affichage de la sidebar.

Coder en dur est le fait d'intégrer les données directement dans le code source du programme sans les obtenir de sources extérieures.



```
829         'link' => '#',
830         'followers' => 'Suivi par 12 personnes'
831     );
832
833     $place[] = array(
834         'name' => 'Rempart de Vannes',
835         'image' => "https://upload.wikimedia.org/wikipedia/commons/d/d5/Bretagne_Morbihan_Vannes2_tango7174.jpg",
836         'link' => '#',
837         'followers' => 'Suivi par 12 personnes'
838     );
839
840     $place[] = array(
841         'name' => 'Rempart de Vannes',
842         'image' => "https://upload.wikimedia.org/wikipedia/commons/d/d5/Bretagne_Morbihan_Vannes2_tango7174.jpg",
843         'link' => '#',
844         'followers' => 'Suivi par 12 personnes'
845     );
846
847
848     $place = array_slice($place, 0, 3);
```

Incorporer directement les données dans le code

Progressivement , j'ai intégré des boucles for en Twig pour réduire le code et permettre l'arrivée des fonctions permettant de proposer des intérêts et des personnes depuis la base de donnée.

```

1 <style...>
114 <div id="sidebar">
115 <div class="block" style="min-height: 300px;background: #FFF;width: inherit !important;margin-top: 16px;padding: 0;top: 50px;" data-spy="affix" data-offset-top="0" data-offse-
116 <div class="content">
117
118 <div class="sub-block">
119 <div class="sub-title">Propositions de personnes </div>
120 <div class="sub-content">
121 {% for peoples in people %}
122 <div class="media">
123 <div class="media-left">
124 <a href="#">
125 
126 </a>
127 </div>
128 <div class="media-body">
129 <h4 class="media-heading">{{ peoples.name }} {{ peoples.firstname }}</h4>
130 Suivi par 100 personnes
131 </div>
132 <div class="media-right">
133 <a href="#" class="btn btn-default" style="padding: 4px;">Suivre
134 </a>
135 </div>
136 </div>
137 {% endfor %}
138
139 </div>
140
141 </div>
142 <div class="sub-block">
143 <div class="sub-title">Propositions de lieux </div>
144 <div class="sub-content">
145 <div class="media">
146 <div class="media-left">
147 <a href="#">

```

Page HTML de la side-bar



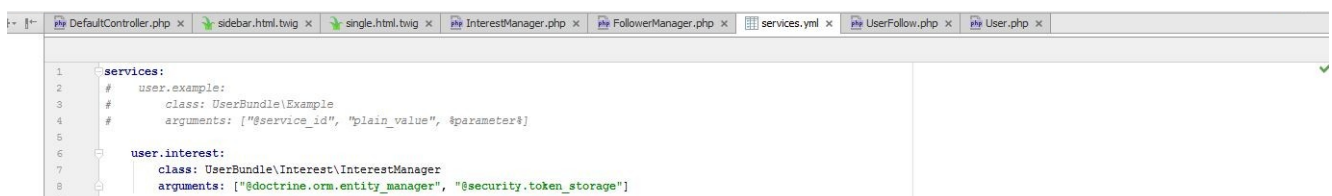
Rendu escompté pour la sidebar avec la mise en place des blocs (proposition de personnes, propositions de lieux et publicité) accueillant les différentes propositions faites à l'utilisateur.

Les semaines suivantes , j'ai travaillé sur la fonction permettant de proposer des intérêts à l'utilisateur de façon aléatoire et personnalisée, d'autres fonctions pour proposer des personnes à suivre selon leurs points communs et une pour proposer des suggestions de lieux et/ ou d'événements à suivre selon les intérêts «aimés» par l'utilisateur.

Fonction qui permet de proposer des intérêts à l'utilisateur:

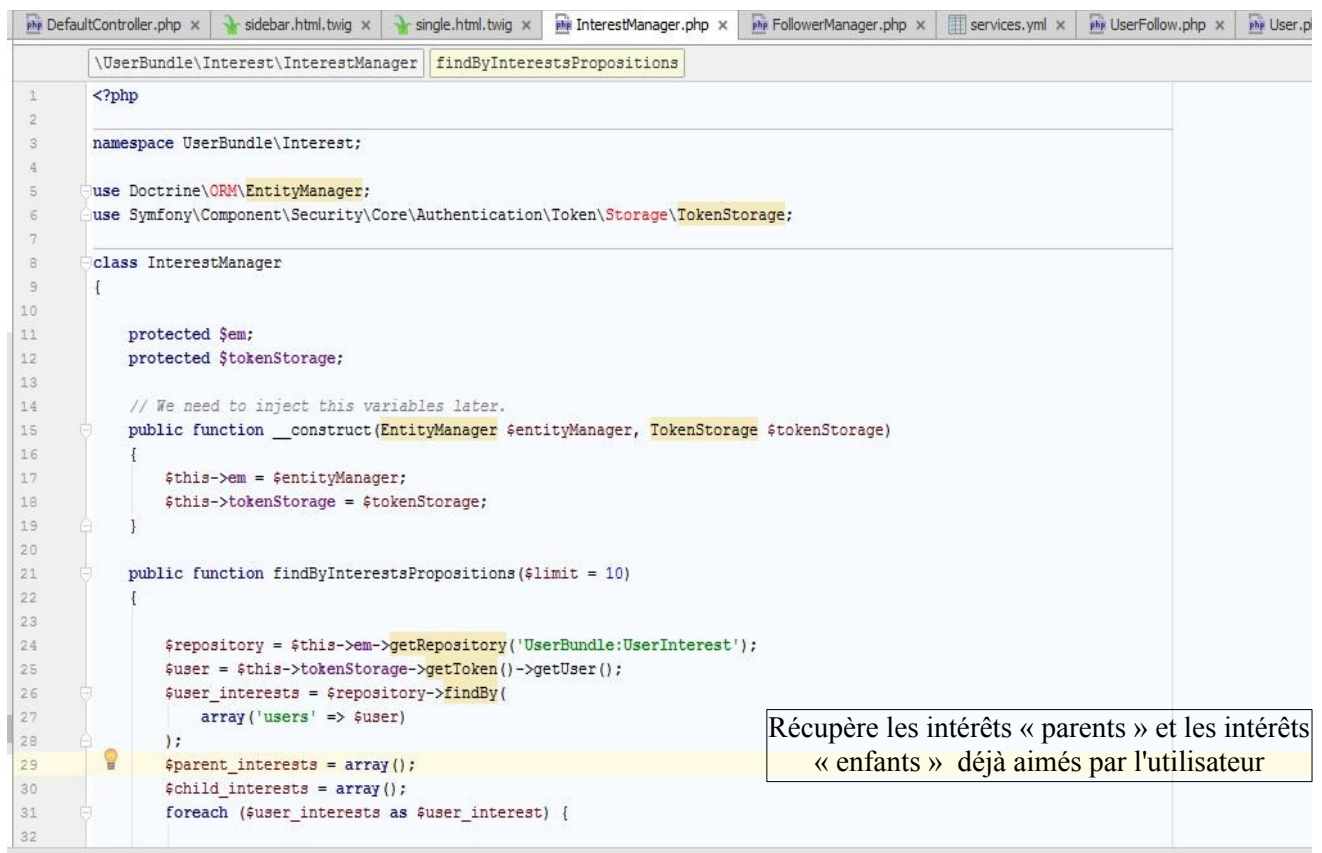
Cette fonction sera exécutée à plusieurs endroits dans le code: pour faciliter la chose Symfony permet de créer un service.

Un service aide à instancier, organiser et récupérer les nombreux objets de l' application. Ce dernier permet de normaliser et de centraliser la façon dont les objets sont construits.



```
1 services:
2   # user.example:
3   #   class: UserBundle\Example
4   #   arguments: ["%service_id%", "plain_value", %parameter%]
5
6   user.interest:
7     class: UserBundle\Interest\InterestManager
8     arguments: ["%doctrine.orm.entity_manager%", "%security.token_storage%"]
```

Création du service : user.interest



```
1 <?php
2
3 namespace UserBundle\Interest;
4
5 use Doctrine\ORM\EntityManager;
6 use Symfony\Component\Security\Core\Authentication\Token\Storage\TokenStorage;
7
8 class InterestManager
9 {
10
11     protected $em;
12     protected $tokenStorage;
13
14     // We need to inject this variables later.
15     public function __construct(EntityManager $entityManager, TokenStorage $tokenStorage)
16     {
17         $this->em = $entityManager;
18         $this->tokenStorage = $tokenStorage;
19     }
20
21     public function findByInterestsPropositions($limit = 10)
22     {
23
24         $repository = $this->em->getRepository('UserBundle:UserInterest');
25         $user = $this->tokenStorage->getToken()->getUser();
26         $user_interests = $repository->findBy(
27             array('users' => $user)
28         );
29         $parent_interests = array();
30         $child_interests = array();
31         foreach ($user_interests as $user_interest) {
```

Récupère les intérêts « parents » et les intérêts « enfants » déjà aimés par l'utilisateur

<pre> 3 foreach (\$user_interests as \$user_interest) { 4 5 if (!is_null(\$user_interest->getInterests()->getParent())) { 6 \$child_interests[] = \$user_interest->getInterests(); 7 } else { 8 \$parent_interests[] = \$user_interest->getInterests(); 9 } 10 } </pre>	<p>Remplissage des tableaux des intérêts parents et enfants déjà aimés</p>
<pre> 11 if (empty(\$parent_interests) rand(1, 2) == 3) { 12 \$question = 'Quels sont vos intérêts ?'; 13 \$repository = \$this->em->getRepository('UserBundle:Interest'); 14 \$interests = \$repository->findBy(15 array('parent' => NULL) 16); </pre>	<p>On récupère tous les intérêts parent inscrit dans la base de donnée</p>
<pre> 17 \$proposition_interests = array(); 18 foreach (\$interests as \$interest) { 19 if (!in_array(\$interest, \$parent_interests)) { 20 \$proposition_interests[] = \$interest; 21 } 22 } </pre>	<p>Si l'intérêt n'est pas déjà aimé par l'user on le met dans le tableau « <i>proposition_interests</i> »</p>
<pre> 23 shuffle(\$proposition_interests); 24 \$proposition_interests = array_slice(\$proposition_interests, 0, \$limit); 25 } </pre>	<p>Mélange les propositions et n'affiche qu'une liste égale à \$limit, ici 10.</p>
<pre> 26 } else { 27 shuffle(\$parent_interests); 28 \$question = 'Vous avez indiqué comme intérêt "'.parent_interests[0]->getLabel().'". Qu\'aimez-vous précisément ?'; 29 30 \$repository = \$this->em->getRepository('UserBundle:Interest'); 31 \$interests = \$repository->findByParent(\$parent_interests[0]); 32 \$proposition_interests = array(); 33 foreach (\$interests as \$interest) { 34 if (!in_array(\$interest, \$child_interests)) { 35 \$proposition_interests[] = \$interest; 36 } 37 } </pre>	<p>On fait pareillement pour remplir le tableau de proposition d'intérêts enfants</p>
<pre> 38 shuffle(\$proposition_interests); 39 \$proposition_interests = array_slice(\$proposition_interests, 0, \$limit); 40 } </pre>	<p>Le return permet de renvoyer le résultat de la fonction.</p>
<pre> 41 return array('question' => \$question, 'propositions' => \$proposition_interests); 42 } </pre>	

Fonction : *FindByInterestsPropositions()* dans la classe InterestManager

```

847     $place = array_slice($place, 0, 3);
848
849     $interest_manager = $this->container->get('user.interest');
850     $proposition_interets = $interest_manager->findByInterestsPropositions(6);
851
852     $follower_manager = $this->container->get('user.follower');
853     $proposition_people = $follower_manager->findPeoplePropositions(3);
854
855     return $this->render(
856         'sidebar.html.twig',
857         array( 'proposition_people' => $proposition_people, 'places' => $place, 'proposition_interets' => $proposition_interets['propositions'] )
858     );
859 }
860
861 }
862
863 }
864

```

Nous appelons la fonction dans le DefaultController

J'ai employé la même méthode (création d'un service) pour les deux autres fonctions.

La fonction *FindPeoplePropositions()* permet de proposer aléatoirement des personnes avec qui l'utilisateur a le plus d'intérêts en commun (Cf ANNEXE 1).

La fonction *FindLocalePropositions()* permet de proposer aléatoirement des lieux/ événements dont les «tags» correspondent aux intérêts de l'utilisateur (cf ANNEXE 2).

Difficultés rencontrées:

La principale difficulté que j'ai rencontré était le commencement de la fonction donc de l'algorithme. Je suis facilement déstabilisé par le mot algorithme, les boucles et les conditions.

Je me suis rendu compte qu'il fallait bien définir l'intérêt de la fonction d'une façon globale et de décomposer ce dont nous avons besoin de récupérer au fur et à mesure.

La meilleure solution a été de tout écrire sur papier les différentes étapes.

Qu'est-ce que je voulais? Par quel moyen?

J'ai également appris à tester mon code au fur et à mesure de ma fonction pour me permettre de voir si je partais du bon pied et éviter de chercher l'erreur partout alors qu'elle était au début par exemple.

2.3 Expérience apportée

Ce premier stage m'a permis d'être immergée totalement dans le quotidien professionnel d'un développeur, ce qui me conforte dans mon choix professionnel. J'y trouve beaucoup d'avantages, comme la réflexion intellectuelle, la collaboration avec les autres développeurs, la créativité et la satisfaction d'avoir créé quelque chose du début jusqu'à la fin.

D'un point de vue théorique, j'ai rencontré et fait beaucoup de choses différentes, utilisé plusieurs langages. Et même si certains langages ne se ressemblent pas, la logique reste la même. Je pense avoir pris des automatismes dans ma façon de travailler. Il est plus facile pour moi de voir la structure du code avec le contrôler, les classes et les vues.

De plus, j'ai pu apercevoir les difficultés rencontrées par une jeune entreprise, et toute la mise en place d'un projet avec ses moments de satisfaction et de doute.

Pour l'anecdote, j'avais déjà téléchargé l'application avant de les rencontrer lors d'un de leur premier festival équipé, ce qui m'a permis de ne pas être qu'une stagiaire développement mais de participer également à la réflexion autour du projet, des événements qui seraient intéressants de connecter, des moyens de communication pour promouvoir leur application.

J'ai apprécié leur ouverture d'esprit et c'était agréable pour moi de partager mes idées, de sentir qu'elles étaient prises en considération et ainsi avoir apporté ma contribution à l'élaboration de leur projet.

ANNEXE

Annexe 1: Fonction des personnes à suivre

Annexe 2: Fonction des lieux à suivre

Annexe 3: Code HTML et CSS pour la sidebar

Annexe 4: Page d'accueil de l'ancien site de «Imagina»

Annexe 5:Flyer «Imagina Connect» à destination des professionnels.

Annexe 1: Fonctions des personnes à Suivre

```
DiscoverBundle\...DefaultController.php x sidebar.html.twig x ImaginaBundle\...DefaultController.php x UserInterest.php x InterestManager.php x FollowerManager.php x UserFollow.php x User.php x
\UserBundle\Follower\FollowerManager findPeoplePropositions
1 <?php
2
3 namespace UserBundle\Follower;
4
5 use Doctrine\ORM\EntityManager;
6 use Symfony\Component\Security\Core\Authentication\Token\Storage\TokenStorage;
7 use ImaginaBundle\Utils\Tool;
8
9 class FollowerManager
10 {
11     protected $em;
12     protected $tokenStorage;
13     protected $tool;
14
15     // We need to inject this variables later.
16     public function __construct(EntityManager $entityManager, TokenStorage $tokenStorage, Tool $tool)
17     {
18         $this->em = $entityManager;
19         $this->tokenStorage = $tokenStorage;
20         $this->tool = $tool;
21     }
22
23     public function findPeoplePropositions($limit = 4)
24     {
25         $user = $this->tokenStorage->getToken()->getUser();
26         $user_interests = $user->getInterests();
27
28         $usersScores = array();
29         foreach ($user_interests as $interest) {
30             $usersForInterest = $interest->getInterests()->getUsers();
31
32             foreach ($usersForInterest as $user) {
33                 if (!empty($usersScores[$user->getUsers()->getId()])) {
34                     $usersScores[$user->getUsers()->getId()] = $usersScores[$user->getUsers()->getId()] + 1;
35                 } else {
36                     $usersScores[$user->getUsers()->getId()] = 1;
37                 }
38             }
39         }
40
41         arsort($usersScores);
42         $usersScores = array_slice($usersScores, 0, (intval($limit) * 2), true);
43         $this->tool->shuffle($usersScores);
44         $usersScores = array_slice($usersScores, 0, $limit, true);
45
46         $repository = $this->em->getRepository('UserBundle:User');
47
48         arsort($usersScores);
49         $usersScores = array_slice($usersScores, 0, (intval($limit) * 2), true);
50         $this->tool->shuffle($usersScores);
51         $usersScores = array_slice($usersScores, 0, $limit, true);
52
53         $repository = $this->em->getRepository('UserBundle:User');
54
55         $propositions_users = array();
56         foreach($usersScores as $user_id => $propositions ){
57             $user = $repository->findOneById($user_id);
58             $propositions_users[] = $user;
59         }
60
61         return $propositions_users;
62     }
63 }
```

Annexe 2: Fonction des lieux à suivre

```
public function findLocalePropositions($limit = 4)
{
    $user = $this->tokenStorage->getToken()->getUser();
    $user_interests = $user->getInterests();

    $repository = $this->em->getRepository('UserBundle:UserInterest');
    $userInterests = $repository->findBy(['state' => 'like', 'user' => $user]);

    $locationScores = array();
    foreach ($userInterests as $userInterest) {
        $repository = $this->em->getRepository('ConnectBundle:LocationTag');
        $query = $repository->createQueryBuilder('t')
            ->where('t.value LIKE :value')
            ->setParameter('value', '%' . $userInterest->getInterest()->getLabel() . '%')
            ->getQuery();
        $value_locations = $query->getResult();

        foreach ($value_locations as $value_location) {
            if (!empty($locationScores[$value_location->getLocation()->getId()])) {
                $locationScores[$value_location->getLocation()->getId()] = $locationScores[$value_location->getLocation()->getId()] + 1;
            } else {
                $locationScores[$value_location->getLocation()->getId()] = 1;
            }
        }
    }

    arsort($locationScores);
    $locationScores = array_slice($locationScores, 0, (intval($limit) * 2), true);
    //mixed the results
    $this->tool->shuffle($locationScores);
    $locationScores = array_slice($locationScores, 0, $limit, true);

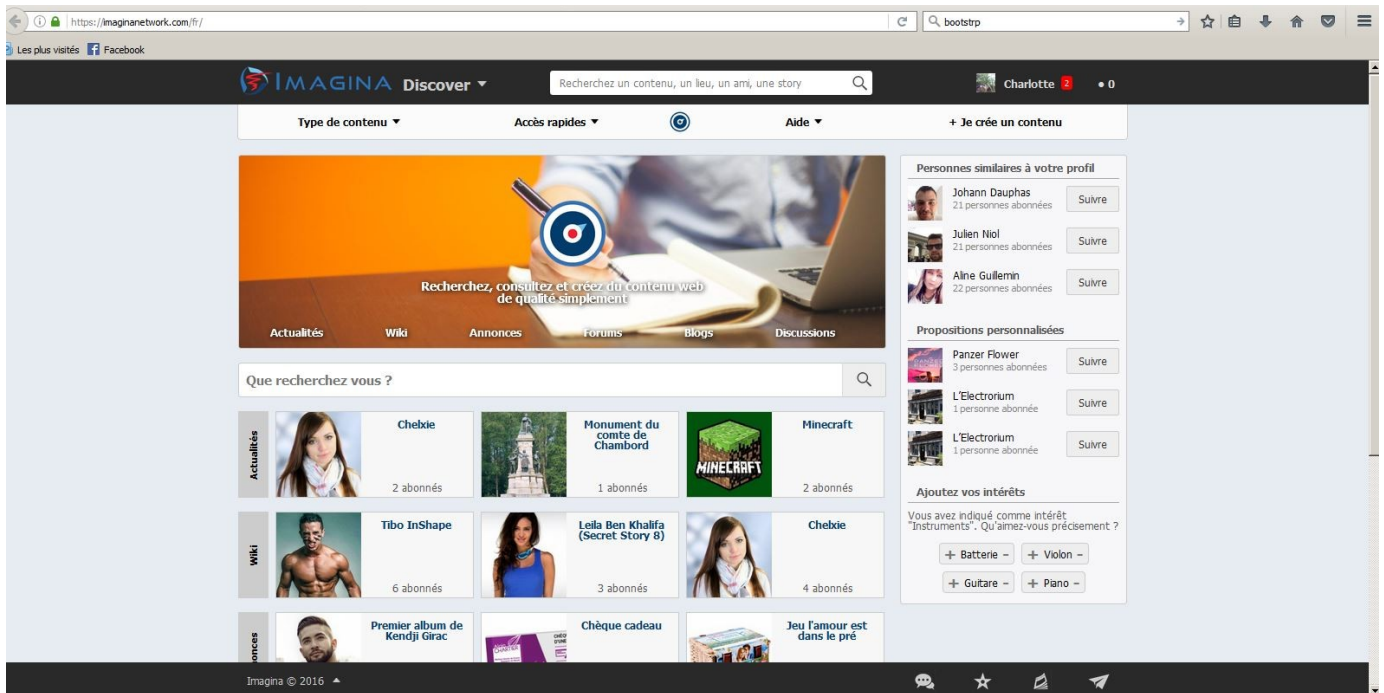
    $repository = $this->em->getRepository('ConnectBundle:Location');
    $propositions_locations = array();
    foreach($locationScores as $location_id => $propositions ){
        $locale = $repository->findOneById($location_id);
        $propositions_locations[] = $locale;
    }

    return $propositions_locations;
}
```


Annexe 3: Code HTML , CSS pour la mise en forme de la sidebar

```
</div>
</div>
<div class="sub-block">
  <div class="sub-title">Propositions de lieux </div>
  <div class="sub-content">
    {% for place in proposition_locale %}
      <div class="media">
        <div class="media-left">
          <a href="#">
            
          </a>
        </div>
        <div class="media-body">
          <h4 class="media-heading">{{ place.title }}</h4>
          {% if place.followers|length ==0 %} Nouveau lieu
          {% elseif place.followers|length ==1 %} Suivi par 1 personne
          {% else %} Suivi par
            {{ place.followers|length }} personnes
          {% endif %}
        </div>
        <div class="media-right">
          <a href="{{ place.guid }}" class="btn btn-default " style="...">Suivre</a>
        </div>
      </div>
    {% endfor %}
  </div>
</div>
```

Annexe 4: Page d'accueil de l'ancien site de «Imagina»



Annexe 5: Flyer «Imagina Connect» à destination des professionnels



IMAGINA - Connect -

Événements

Dynamisez vos événements sportifs, festivals, salons et expositions pour les rendre uniques.

Parcs de Loisirs

Guidez et informez les visiteurs en temps réel dans vos parcs animaliers, parcs d'attractions et musées.

Villes & Territoire

Transformez votre ville en Smart City. Connectez les lieux historiques et culturels du patrimoine et les événements locaux.

Commerces

Optimisez vos points de vente. Attirez des visiteurs dans vos boutiques, grandes surfaces et galeries marchandes.

Campus

Connectez votre école ou université aux étudiants, professeurs et communiquez instantanément.

Services

Innovez en équipant les moyens de transport, aéroports, campings et ports de plaisance.

Offrez à vos visiteurs
une expérience connectée

54
20h
120m

Informez vos visiteurs
 Vos visiteurs obtiennent les informations de votre espace en temps réel. La barre de recherche leur permet d'accéder en un instant à ce qui les intéresse vraiment.

Guidez grâce aux iBeacons

Une fois ces puces disposées dans votre espace, l'application les détecte et propose à vos visiteurs des informations pertinentes au bon endroit et au moment opportun.

Alertez avec les notifications

Ces messages sont envoyés avant un événement précis à l'intérieur de votre espace. Ils se présentent sous la forme d'un message texte, d'une photo, d'une vidéo ou d'un sondage.

Dynamisez les échanges

En se déplaçant dans votre espace, vos visiteurs peuvent publier des messages, partager des photos et se géolocaliser pour se retrouver entre amis ou en famille.

Personnalisez selon votre espace

L'application est entièrement modulable. Organisez votre contenu en fonction de votre espace. Choisissez le type d'informations que vous souhaitez mettre en avant.

Améliorez grâce aux données

Les statistiques vous permettent d'identifier la fréquentation et le temps passé par vos visiteurs aux différents points de votre espace. Analyser les flux de vos visiteurs et leurs réponses aux sondages.

09 81 74 15 14

www.imagina.io

contact@imagina.io

1 Place Albert Einstein
56000 VANNES