

Paris
Matthieu
BTS1 SLAM

Rapport de Stage : Bluesquare Computing



Sommaire :

I)Présentation de l'entreprise et de mes activités :

A) Présentation de l'entreprise :

B) Présentation de mes activités :

II)Vue en détails de mes activités :

A) Projet Air&Jobs :

B) Projet WebService :

III) Conclusion:

IV) Lexique:

V) Annexe:

I)Présentation de l'entreprise et de mes activités :

A) Présentation de l'entreprise :

Bluesquare Computing est une jeune société de conseil en technologies en forte croissance fondée par des ingénieurs.

Les activités de l'entreprise se déclinent en 3 catégories :

- **Activités digitales** : Applications mobiles et plateformes web
- **Data Mining*** : R&D externalisée et analyses prédictives
- **Start-ups* en interne** : Production technologique et conseil et structuration

Bluesquare Computing réalise aussi bien des missions dans le **domaine du digital** (sites vitrines, sites e-commerces, plateformes sur mesures, web app, app natives iOS et Android ...), mais va également plus loin dans le domaine du **big data*** ou de la **R&D** où Bluesquare agit comme laboratoire externalisé pour ses partenaires.

Son équipe projet est constituée :

- d'un chef de projet
- de trois développeurs
- de trois web designer

La réalisation est divisée en différentes parties :

Zoning et Graphisme	Intégration et création contenu texte	Déploiement continu des différents modules	Recettage final et création du BackOffice	Mise en ligne
<p>Zoning</p> <p><i>Allers-retours illimités</i> Maquettage schématique des différentes zones et pages du site et validation</p> <p>Graphisme</p> <p><i>Allers-retours illimités</i> Réalisation de l'univers graphique .psd pour les pages principales et validation</p>	<p>Intégration web</p> <p>Les développeurs de Bluesquare traduisent le .psd créé par la graphiste en langage web responsive design.</p>	<p>Déploiement continu des différents modules</p> <p>Intégration continue des différentes pages. Etant donné le zoning et le graphisme défini en 1, Bluesquare crée l'ensemble des pages de la plateforme. Le recettage est fait avec le client</p>	<p>Recettage final</p> <p>L'ensemble du site est revue et recettée par Bluesquare et le client avant sa mise en ligne. Le recettage est réalisé toujours en méthode agile. Le backoffice est créé par Bluesquare. Il permet au client et à son équipe d'administrer le contenu du site</p>	<p>Mise en ligne</p> <p>Déploiement final et lancement de la période de maintenance</p>

Bluesquare Computing a travaillé sur différents projets pour différentes entreprises tels que :

- **Ornikar**, une start-up française se lançant dans le permis de conduire low-cost.
Comptant dans ses premiers investisseurs Xavier Niel, Jacques-Antoine Granjon, et Marc Simoncini, Bluesquare Computing réalisera plusieurs missions d'audits, de production et de structuration technologique.

The screenshot displays the Ornikar website's main offerings. At the top, the Ornikar logo is accompanied by the text 'N° d'agrément E16.044.00090'. Navigation links include 'Ornikar', 'Offres', 'Tarifs', 'Pédagogie', 'J'enseigne la conduite', and 'Connexion'. The central headline reads 'Le permis de conduire moins cher' with subtext 'Entraînez-vous au code de la route en ligne et réservez des leçons de conduite près de chez vous'. Below this, two primary offers are highlighted: 'Code de la route' for 49,90 € (conforme à l'ETG 2016) and 'Conduite' for 34,90 € (première heure de conduite). Each offer includes a 'Je découvre l'offre' button and a list of features. A chatbot interface for 'Ornikar' is visible on the right, and a footer link 'Comment fonctionne Ornikar ?' is at the bottom center.

Offre	Prix	Détails
Code de la route	49,90 €	Conforme à l'ETG 2016, 1600 questions d'entraînement, Livre de code édition 2016, Dossier d'inscription à l'examen, Conseils d'un enseignant
Conduite	34,90 €	Première heure de conduite, 1 ^{ère} heure de conduite, Packs d'heures accessibles après la première heure, Enseignants diplômés, Sans engagement

- **Welkeys**, principal concurrent français de BnbSiPer, le Airbnb de la conciergerie

Confiez-nous vos clés, partez l'esprit léger !

Plus de 1000 concierges disponibles partout en France dès **16€^{TTC}**

☒ ARRIVÉE DES VOYAGEURS
 ☐ DÉPART DES VOYAGEURS
 ☐ MÉNAGE & LINGE

Votre code postal **Trouver mon concierge**

Découvrez notre solution de conciergerie clé en main: **WELKEYS Premium**

Comment ça marche ?

Pour vos locations courte-durée (Airbnb, Abritel...), Welkeys gère pour vous: ménage, accueil des locataires, à partir de 16€^{TTC}

- **Studionet**, agence digitale ayant plus de 500 sites web à son actif et 10 ans d'expérience. Bluesquare Computing réalise une aide à la production

s studionet
AGENCE DIGITALE

L'AGENCE SITE INTERNET STRATÉGIE DIGITALE RÉALISATIONS

CONTACTEZ-NOUS

ROUEN 02 78 77 00 05
PARIS 01 84 16 66 22

L'agence web qui pimente vos projets

*** et qui assure le service !**

- **Jpal**, En partenariat avec l'éducation nationale, J-PAL réalise une étude auprès de plus de 100.000 élèves de collèges pour évaluer diverses méthodes pédagogiques. Bluesquare Computing a réalisé l'application questionnaire liée à l'étude et la plateforme SAAS permettant de traiter les données



Réponse aux questions :

- Existe-t-il un logiciel d'inventaire des éléments informatiques dans l'organisation dans laquelle vous réalisez votre stage (ou dans votre service) ? Si oui, lequel ?
Non
- Existe t-il un logiciel de gestion des incidents ? Si oui, lequel ?
Non
- Existe t-il un outil de gestion des configurations ? Si oui, lequel ? Quel(s) aspect(s) des configurations sont gérés par ce(s) mécanisme(s) ?
Non
- Existe-t-il une cellule dédiée à la résolution des incidents dans l'organisation (ou dans votre service) ?
Non
- Concernant les postes de travail, quelle est la description précise matérielle et logicielle de la machine (ou des machines) que vous utilisez ?
Utilisation de Mac mini et portable, ainsi que des ordinateurs portables sous Windows 10.
- Comment l'organisation procède-t-elle pour installer de nouveaux postes de travail ? Existe-t-il des mécanismes d'automatisation des installations ? Si oui, lesquels ?
Aucune installation, chaque employé ramène son ordinateur personnel
- Quel(s) système(s) de gestion de base de données (SGBD) est (sont) utilisé(s) ? Qui réalise des requêtes SQL ?
L'entreprise utilise phpmyadmin comme SGBD.

B) Présentation de mes activités :

J'ai appris durant ce stage **le framework Symfony2*** et j'ai ensuite appliqué les connaissances acquises sur un projet nommé **Air&Jobs**.

J'ai donc rencontré différentes situations professionnelles qui sont :

- La création d'une interface graphique sur un compte administrateur à l'aide du framework Symfony2 sur le projet Air&Jobs ainsi qu'une fonction permettant d'envoyer un mail aux différents administrateurs du projet.

Air&Jobs est une application disponible sur IOS et Android qui a pour but de faciliter le recrutement dans l'aéronautique. Il y a donc un côté client et un côté recruteur. Cette application est soutenue par Air & Cosmos, média référence du secteur.

En tant que candidat, vous pouvez balayer vers la droite les offres qui vous intéressent et entrez directement en relation avec le recruteur. Un CV ainsi qu'une lettre de motivation n'est pas nécessaire. Vous pouvez choisir des offres selon différents critères : métier, localisation, salaire, ...

En tant que recruteur, vous pouvez balayer vers la droite les candidats qui vous intéressent et ainsi rentrer en relation avec celui-ci. La communication peut se faire via chat ou par un appel vidéo.

- La création d'une page web sur un site déjà disponible qui a pour fonction de présenter l'entreprise. Ce site en question est mis à la disposition du client lorsque celui-ci confie un projet à Bluesquare. Il permet de suivre le projet. Lorsqu'une entreprise externe fait appel à Bluesquare pour réaliser un projet, celle-ci recevra un mail contenant un pdf présentant l'entreprise ainsi que les différentes méthodes de travail de celle-ci puis un lien redirigeant sur la page web en question qui représentera donc le pdf mais sous forme digital.

II) Vue en détails de mes activités :

Compétence référentiel BTS SIO SLAM :

- A1.1.1 : Analyse du cahier des charges d'un service à produire
- A1.3.1 : Test d'intégration et d'acceptation d'un service
- A1.3.3 : Accompagnement de la mise en place d'un nouveau service
- A1.4.1 : Participation à un projet
- A2.2.1 : Suivi et résolution d'incidents
- A2.3.1 : Identification, qualification et évaluation d'un problème
- A4.1.3 : Conception ou adaptation d'une base de données
- A4.1.2 : Conception ou adaptation de l'interface utilisateur d'une solution applicative

A) Projet Air&Jobs :

Ma tâche au sein de ce projet réalisé sur la première semaine de stage était de compléter l'interface graphique déjà présente afin d'y rajouter un onglet dans le menu qui nous renverrai sur une interface présentant un bouton d'envoi de mail vers les administrateurs.

Afin de réaliser ceux-ci, j'ai donc utilisé le framework Symfony2. J'ai donc créer un nouveau controller se nommant AdminRecruteurSwipeController qui contient la fonction « `envoieMail()` » permettant d'envoyer un mail. (voir image ci-dessous)

```
19      * - Affiche la page d'accueil du bundle
20      */
21      public function indexAction() {
22          return $this->render('AdminRecruteurBundle:AdminSwipe:swipe.html.twig');
23      }
24
25      private function envoieMail(){
26
27          //création d'un objet transport
28          $transport = \Swift_SmtpTransport::newInstance('smtp-relay.sendinblue.com', 25 )
29                  ->setUsername('michel@bluesquarecomputing.com')
30                  ->setPassword('EnTrwbCqf6yZxAaP');
31          //création d'un objet mailer
32          $mailer = \Swift_Mailer::newInstance($transport);
33          $message = \Swift_Message::newInstance();
34          $message->setSubject("Objet");
35          $message->setFrom('parismatthieu@yahoo.fr');
36          $message->setTo('francois.paris77@yahoo.fr');
37          // pour envoyer le message en HTML
38          $message->setBody('Hello world');
39          // pour envoyer le message en HTML
40          $message->setBody('<p>Hello world</p>', 'text/html');
41          //envoi du message
42          $mailer->send($message);
43
44
45
46          return $this->render('AdminRecruteurBundle:AdminCompany:listing.html.twig');
47      }
48  }
49  ?>
```

Ensuite, dans la vue contenant le menu du recruteur se nommant « `template-AdminRecruteur.html.twig` », j'ai créé une nouvelle liste « Réinitialisation Swipe ». (Voir image ci-dessous »)


```

<li>
  <a href="{{ path('admin_recruteur_offre_home') }}">
    <i class="fa fa-briefcase"></i>
    <span>Offres</span></a>
</li>
<li>
  <a href="{{ path('admin_recruteur_conversation_listing') }}">
    <i class="fa fa-comments-o"></i>
    <span>Conversations</span></a>
</li>
<li>
  <a href="{{ path('admin_recruteur_company_confirmed') }}">
    <i class="fa fa-user"></i>
    <span>Personnel de la boîte</span></a>
</li>
<li>
  .....
  <a href="{{ path('admin_recruteur_swipe') }}">
    <i class="fa fa-arrows-h"></i>
    <span>Réinitialisation swipe</span></a>
</li>
  .....

```

Entre la balise `` se trouve le nom du nouvel onglet cliquable qui va renvoyer sur un fichier « `routing.yml` » (Voir image ci-dessous) . Celui-ci va faire le lien entre la vue et le **controller***.

```

admin_recruteur_team:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_team.yml"

admin_recruteur_event:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_event.yml"

admin_recruteur_conversation:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_conversation.yml"

admin_recruteur_message:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_message.yml"

admin_recruteur_credit:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_credit.yml"

admin_recruteur_company_confirmed:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_company.yml"

admin_recruteur_offre:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_offre.yml"

admin_recruteur_swipe:
  resource: "@AdminRecruteurBundle/Resources/config/Admin/routing_swipe.yml"

```

« `admin_recruteur_swipe` » renvoie donc à un autre **routing*** se nommant
 « `routing_swipe.yml` »
 Celui-ci va contenir toute les fonctions présentes dans le controller

« AdminRecruteurSwipeController » (Voir image ci-dessous)

```
admin_recruteur_swipe:
  path:    /swipe
  defaults: { _controller: AdminRecruteurBundle:AdminRecruteurSwipe:index }

admin_recruteur_swipe_mail:
  path:    /
  defaults: { _controller: AdminRecruteurBundle:AdminRecruteurSwipe:envoi}
```

En cliquant donc sur l'onglet « Réinitialisation Swipe », l'utilisateur sera redirigé vers la **vue ou view*** «swipe.html.twig » grâce à « admin_recruteur_swipe » qui fait appel dans le controller AdminRecruteurSwipeController à la fonction index.

Sur la vue «swipe.html.twig », l'utilisateur pourra appuyer sur un bouton pour envoyer un mail. (Voir image ci-dessous)

```
{% block css %}
  <link rel="stylesheet" href="{{ asset('AdminBundle/css/style_nicolas.css') }}">
{% endblock %}

{% block title %}
  <div class="center">
    <span class="round-logo">
      <i class="fa fa-arrows-h"></i>
    </span>
    <h3 class="box-title">Demande de réinitialisation des swipes.</h3>
  </div>
{% endblock %}

{% block body %}
  <div class="center">
    <h5 style="margin-top:10%;"> Pour valider la demande, veuillez appuyer sur le bouton </h5>

    <form action="{{ path('admin_recruteur_swipe_mail') }}">
      <input type="submit" value="Valider" onClick="{{ path('admin_recruteur_swipe_mail') }}">
    </form>
  </div>
{% endblock %}
```

En appuyant sur le bouton, cela enclenchera l'action présente dans la balise <form>. Cela va faire appel à la fonction « admin_recruteur_swipe_mail » présente dans le fichier « routing_swipe.yml » (voir image pour le routing_swipe.yml) qui lui va faire appel à la fonction « envoieMail() » présente dans le controller et qui va générer le mail. (Voir image AdminRecruteurSwipeController).

A) Projet Webservice :

Ce projet consiste à retranscrire un fichier pdf en format numérique et responsive sur un site déjà existant se nommant « WebService ». Le pdf en question a pour fonction de présenter l'entreprise, les membres de l'équipe ainsi que le déroulement de la réalisation d'un projet. Au début du projet nous étions que 3 à travailler, dont notamment un chef de projet, qui nous confié les tâches à réalisés. Le chef de projet nous a donc expliqué ce que devait

contenir la page à réaliser, dont notamment, la présentation de l'équipe, le suivi du projet, notre méthode de travail ainsi que les documents téléchargeables tels que les factures. Pour réaliser ce projet, nous avons comme condition d'utiliser le framework Symfony2.

Ce projet s'est déroulé en plusieurs étapes :

- La première était de trouver un **template*** Symfony afin d'avoir une base au niveau de la structure du site.
- La deuxième était de trouver comment intégrer les différents éléments présents sur le pdf, puis commencer à coder mais pas directement sur le template, pour vérifier si l'intégration des éléments fonctionnaient correctement.
- La troisième était d'intégrer nos codes sur le template Symfony afin de réaliser un site statique.
- Puis pour finir, rendre le site dynamique en adaptant le site aux clients ayant fait une demande de projet.

1) Réalisation du site en version statique :

Les pages de notre page web seront divisées en deux grâce à du Bootstrap, à gauche nous avons une colonne en « col-md-3 » puis à droite « col-md-9 ».

Dans la colonne de droite de notre page d'accueil, nous avons décidé d'utiliser du javascript afin d'afficher le nom du client lettre par lettre en boucle. (Annexe 1)

Puis dans la colonne de gauche, nous avons décidé de mettre en téléchargement les différents dossiers concernant le client tels que la facture, les différentes démarches ainsi que le suivi de son projet.

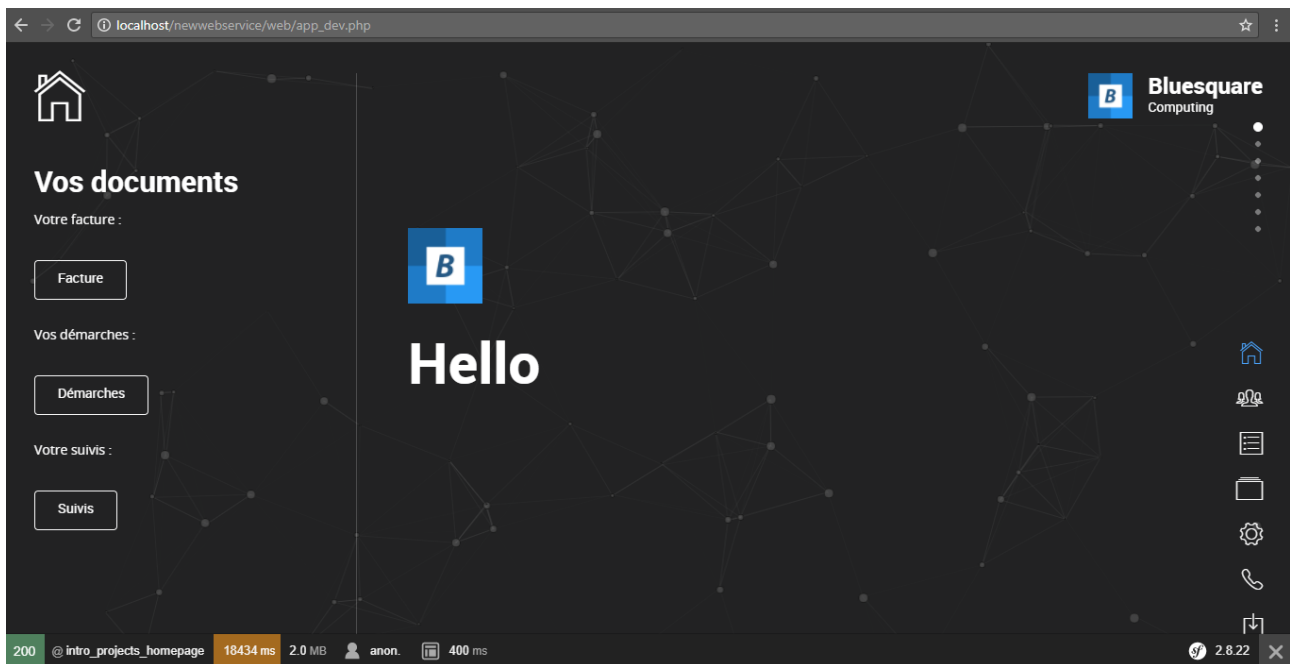
Tout cela est donc intégré dans un fichier se nommant « index.html.twig » se trouvant dans le dossier view de notre Bundle.

(Le code se trouve en annexe)

Ensuite, le client a accès un menu permettant de se rediriger soit vers :

- Home
- Team
- Suivi
- Projets
- Méthodes
- Contact
- Download

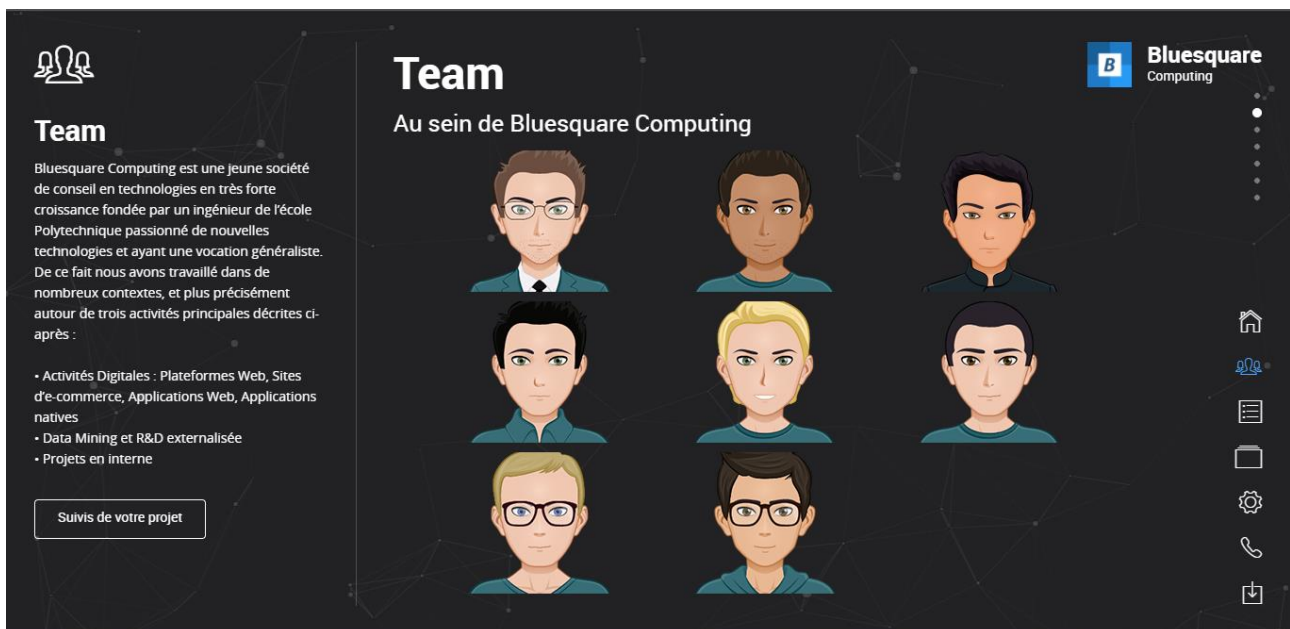
Aperçu de la page d'accueil :



Ensuite passons à la section dédiée à l'équipe :

Dans la colonne de gauche, nous avons donné une description de l'entreprise ainsi que ses secteurs d'activités. (Annexe 2)

Dans la colonne de droite, nous avons créé des avatars afin de représenter les différents membres de l'équipe. (Annexe 3)



Ensuite, j'ai mis en place un hover afin de pouvoir afficher le nom et prénom d'un membre. Puis pour finir, j'ai mis en place un popup qui permet de visualiser la description du membre concerné. Ce popup s'ouvre lorsque l'on clique sur le nom et prénom de celui-ci.

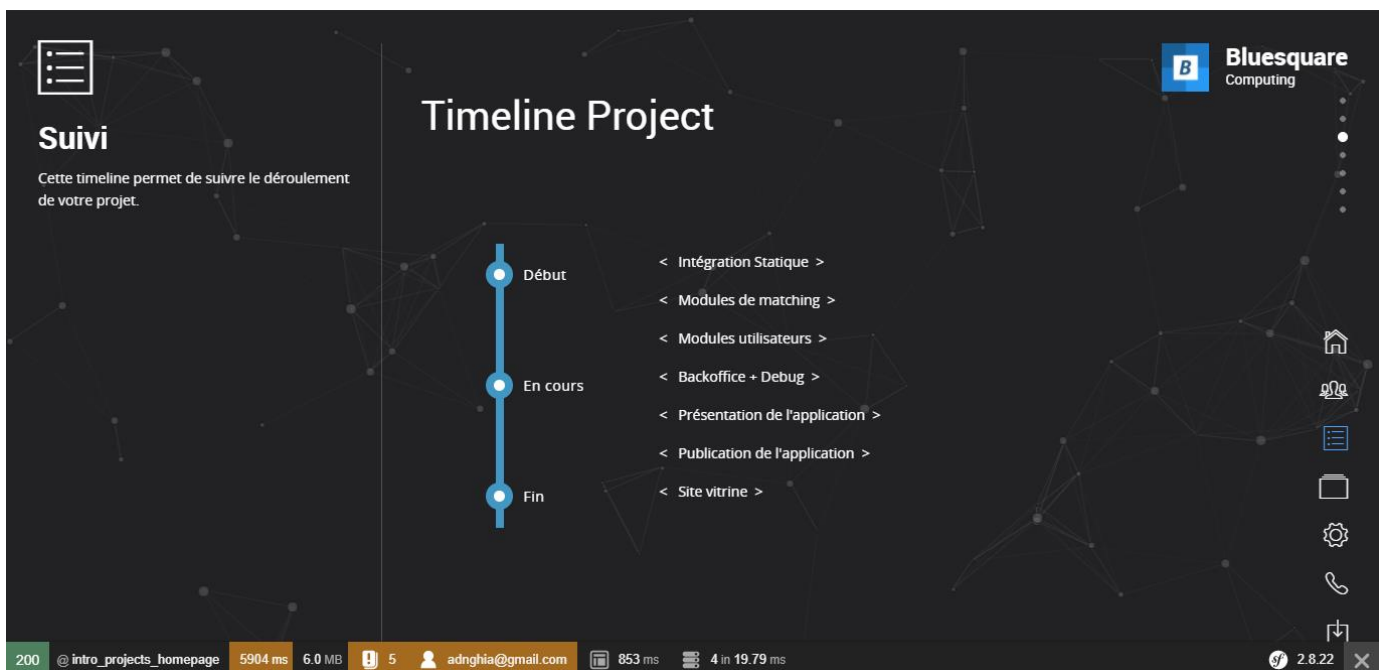


Passons à la section dédiée au suivi du projet :

Dans cette section, j'ai mis en place une timeline verticale divisé en trois parties :

- Début
- En Cours
- Fin

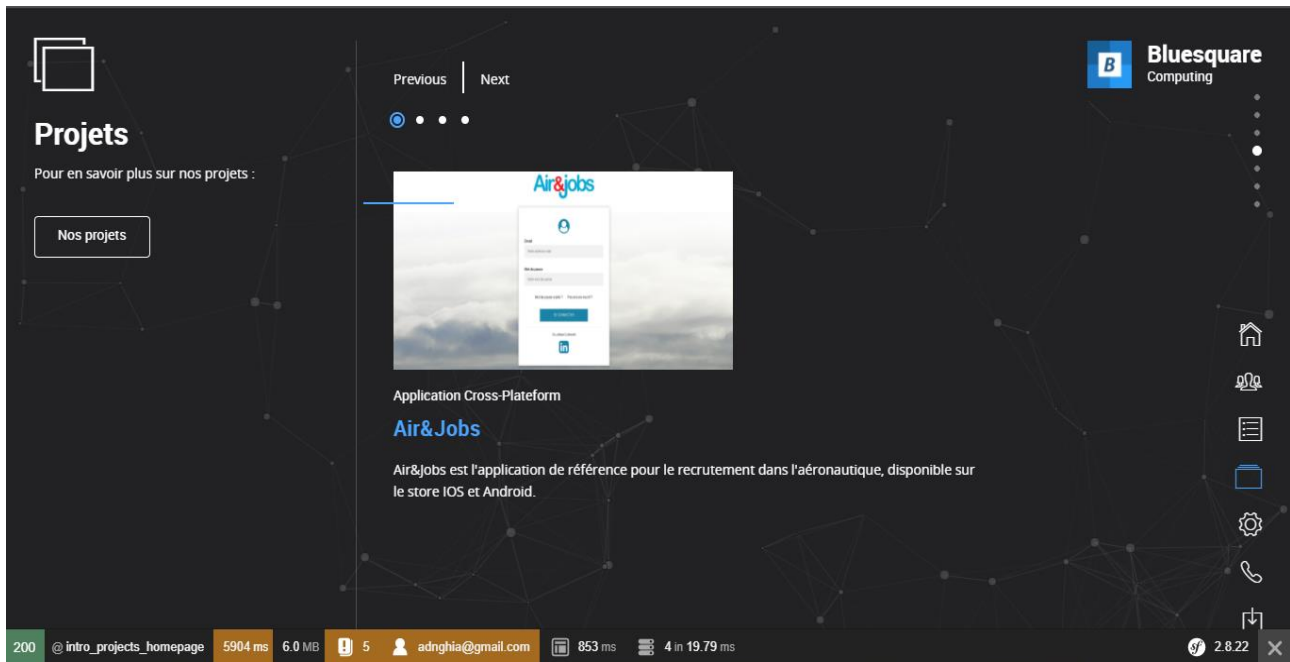
Celle-ci va permettre d'afficher l'état d'avancement du projet destiné au client. Lorsque le client va cliquer sur « En Cours », cela va faire apparaître les tâches qui sont en cours sur le projet. Afin de les afficher, j'ai utilisé du javascript permettant de changer le display des différents blocs contenant les informations, c'est-à-dire de passer du display :none au display : block ainsi que l'inverse. (Annexe 4)



Pour la page Projets :

On a gardé sur cette section, ce que le template nous fournissait. J'ai juste remplacé les

images qui étaient déjà présentes ainsi que les différents titre et paragraphe. (Annexe 5)

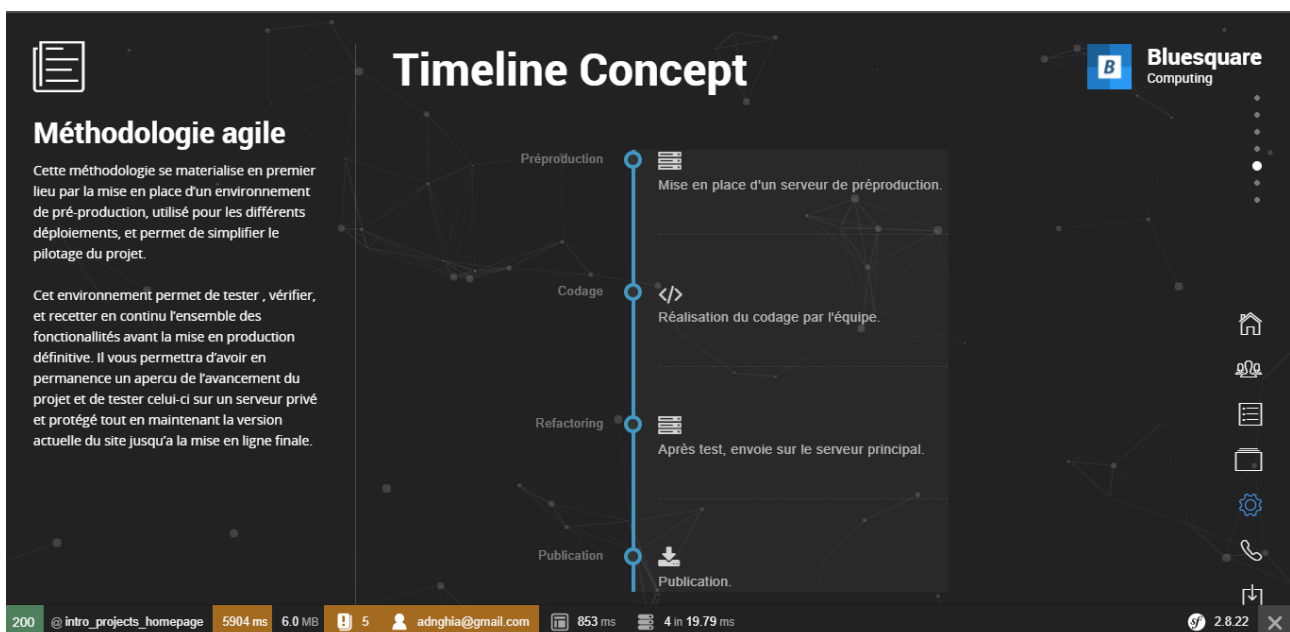


Pour la section Méthodes :

Cette section a pour fonction de décrire le déroulement d'un projet. C'est-à-dire de sa conception à sa publication finale.

Dans la colonne de gauche, nous avons définis en quoi consiste la méthodologie agile utilisé par Bluesquare.

Puis dans la colonne de droite, une nouvelle timeline verticale. (Annexe 6)

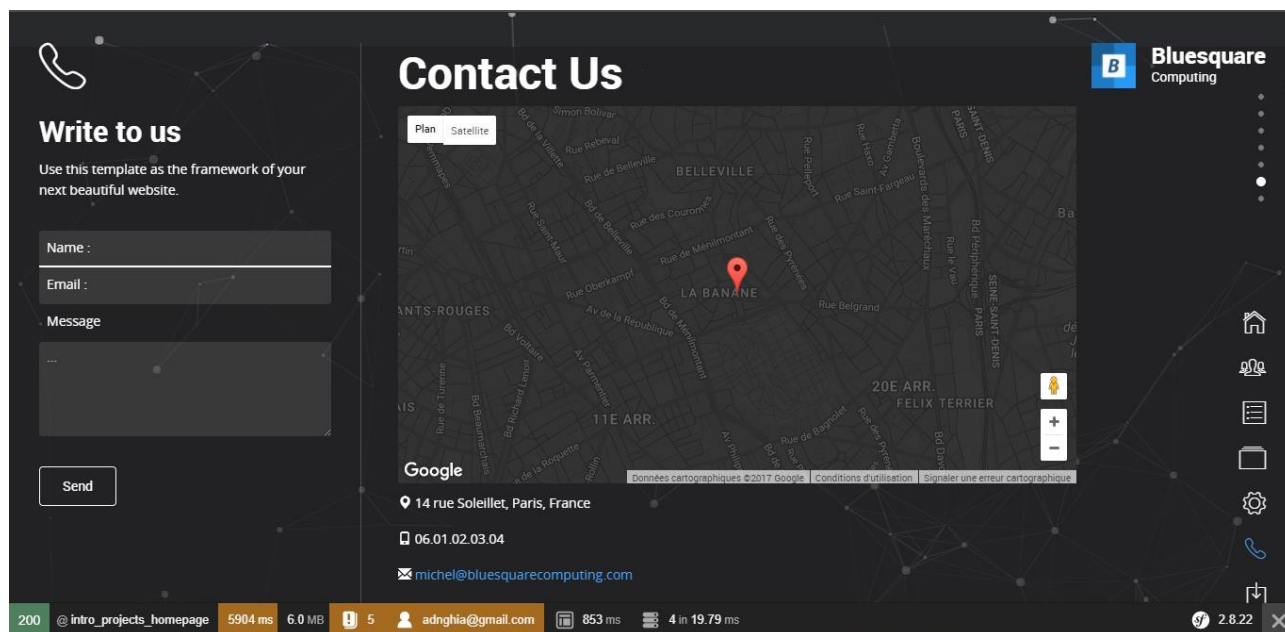


Pour la section Contact :

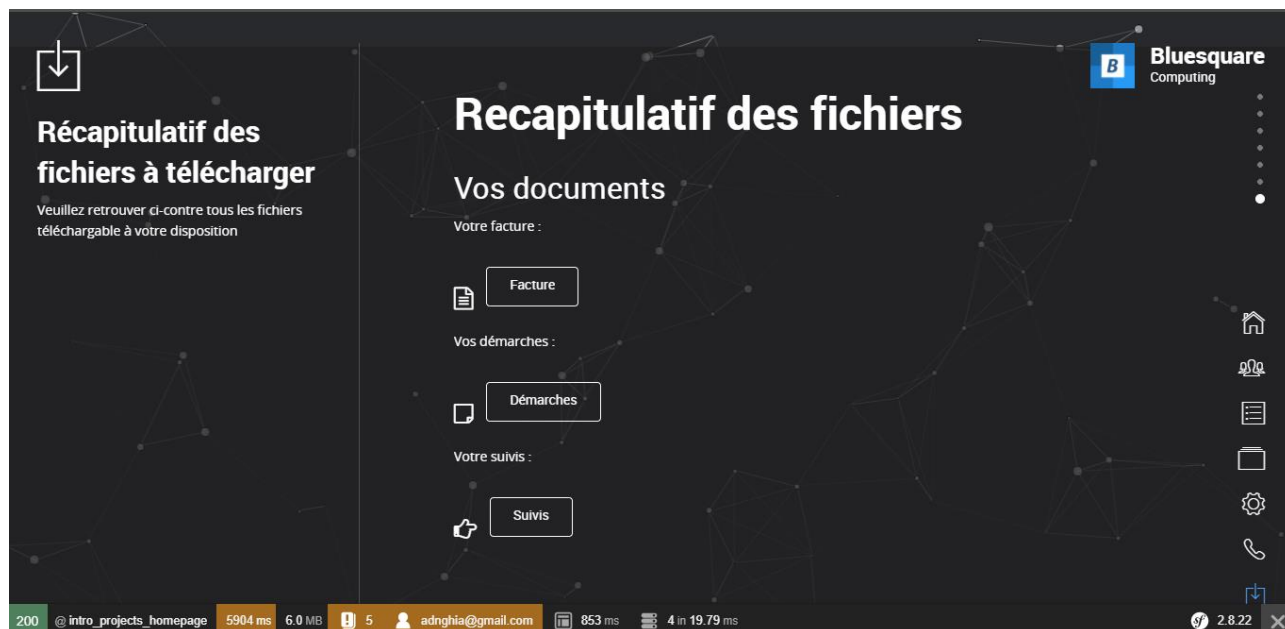
Dans la colonne de gauche, nous avons un formulaire de contact basique, avec nom, email

et commentaire.

Puis à droite, j'ai ajouté une carte google map ainsi que les différentes coordonnées de l'entreprise grâce à Google Map Javascript API. J'ai donc défini le centre de la carte, en mettant la longitude et la latitude de l'emplacement des locaux, ainsi qu'un marqueur. Pour les couleurs, je voulais une carte qui soit en harmonie avec le template, j'ai donc trouvé un site présentant des google map personnalisé (lien : <https://snazzymaps.com/>). J'ai ensuite mis dans le javascript le style pour changer les couleurs. (Annexe 7)



Puis pour finir, la dernière section du site contient toutes les informations que le client peut télécharger. (Annexe 8)



2) Réalisation du site en version dynamique :

Après avoir finis la mise en place de la page web en version statique, nous nous attaquons à

la version dynamique. On a pour objectif de faire en sorte que cette page web s'adapte pour chaque client se connectant au site web « WebService ». Pour se faire, nous avons dû prendre connaissance de la base de données déjà existante sur phpmyadmin.

Différentes modalités sont à changer tels que :

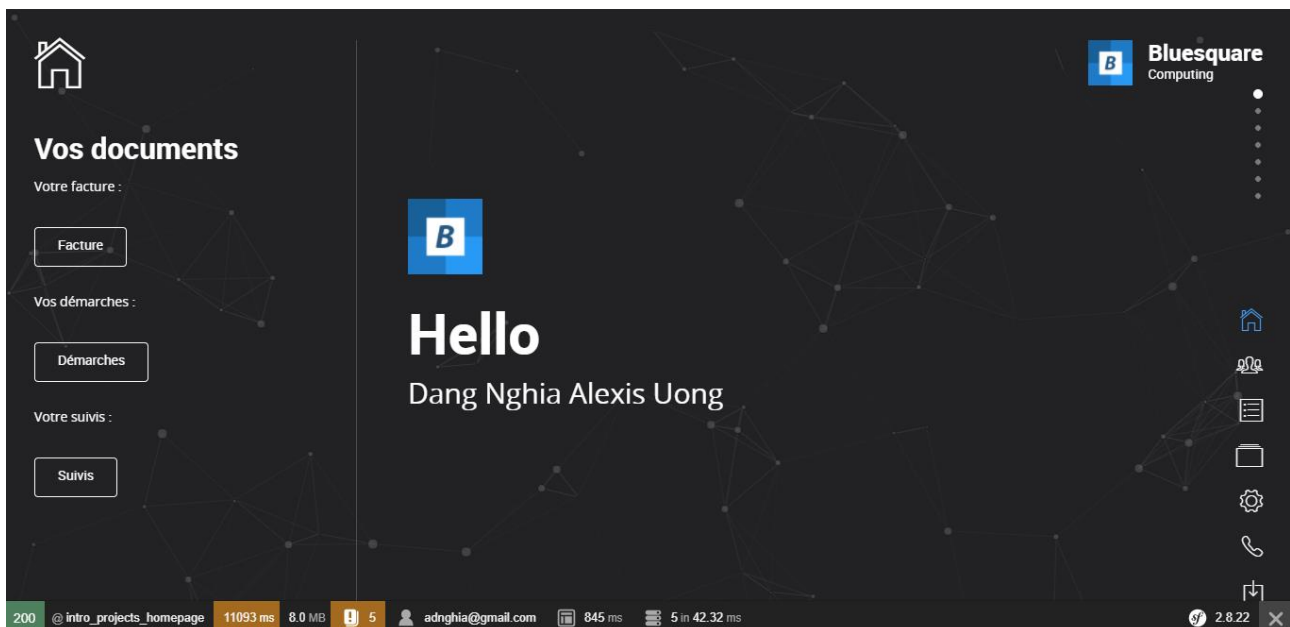
- L'affichage sur l'accueil du nom et prénom du client connecté
- Afficher dans la section suivi, l'avancement du projet destiné au client
- Rendre dynamique la présentation de l'équipe ainsi que les popups avec les employés présents dans la BDD
- Mettre à disposition du client les différentes factures, ainsi que ses démarches.

Pour effectuer cela, j'ai dû codé différentes fonctions en php avec le framework Symfony2 dans un controller.

Pour l'accueil :

Pour afficher le nom et le prénom du client connecté, j'ai utilisé « getToken() » ainsi que « getUser() » afin de récupérer l'id correspondant au client. Ensuite avec Doctrine, je décide d'aller trouvé et de récupérer les informations dans la table User correspondant à l'id récupérer avec « findOneBy(array("id" => \$userCo)) ». Puis je retourne un tableau se nommant « user » comprenant les informations du client sur la vue.

Puis j'affiche sur la vue le nom et prénom du client avec « {{ user.firstname }} » et « {{ user.lastname }} ». Firstname et lastname correspond à nos noms de colonnes présentent dans la table User.



Pour le suivi :

Grâce à la partie juste au-dessus, j'ai donc réussi à récupérer l'id du client qui s'authentifie

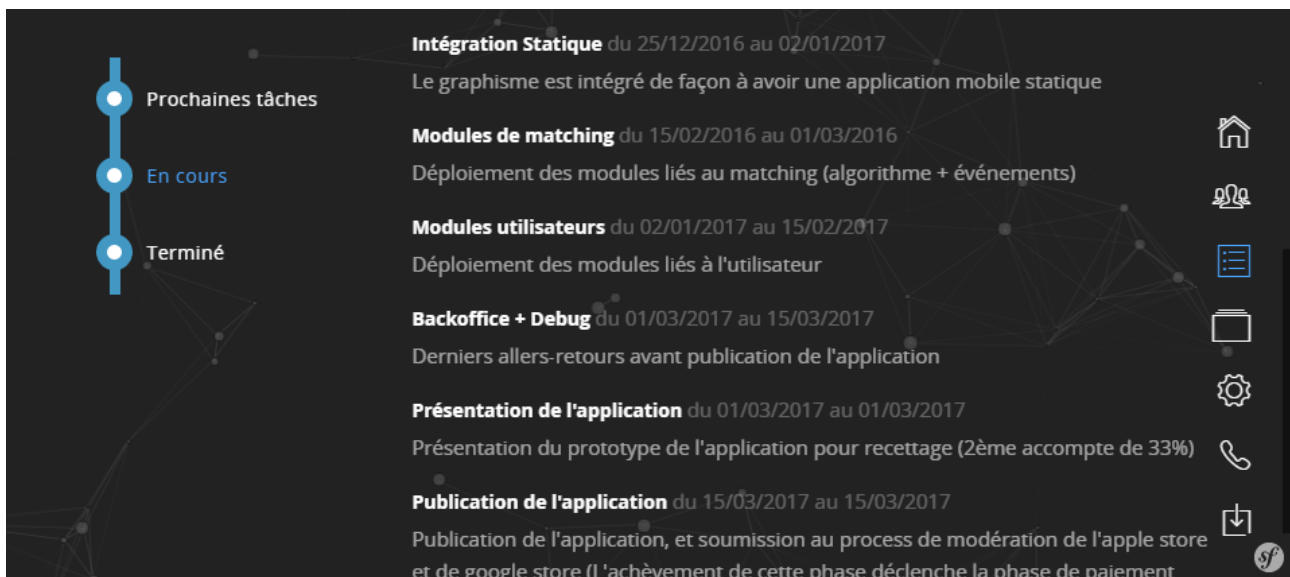
sur le site web, ce qui va me faciliter la tâche pour affiché le suivi du projet. Dans la BDD, chaque client est associé à un projet grâce à une clé étrangère. La table projet possède donc toutes les différentes tâches a effectué sur le projet. On peut distinguer celles qui sont finis ou en cours avec l'aide d'une colonne se nommant status. Si la valeur du status vaut 0, alors la tâche est en cours, puis si elle vaut 1, alors elle est finie.

J'ai donc utilisé deux requêtes afin de récupérer les tâches en cours sur le projet puis celle qui sont finis.

J'effectue donc les requêtes suivantes :

- \$timeline =
\$this->getDoctrine()->getRepository('BCInvoiceBundle:Projet')->findBy(array("id" => \$userCo, "status" => 0));
- \$timeline =
\$this->getDoctrine()->getRepository('BCInvoiceBundle:Projet')->findBy(array("id" => \$userCo, "status" => 1));

Puis dans la vue, on affiche le titre de la tâche, sa date de début et de fin puis sa description.



Pour l'équipe :

Tout d'abord, chaque employé possède un avatar, cependant la BDD tel qu'elle n'est pas adaptée à cela, il faut donc permettre l'upload de l'image. Pour cela, il faut donc rajouter une colonne avatar à la table Employee. Pour se faire, on va éditer un fichier comprenant l'entité Employee.

On y rajoute les lignes suivantes :

```
/**
 * @var string
 *
 * @ORM\Column(name="avatar", type="text", nullable=true)
 */
private $avatar;

/**
```

```

* @param text $avatar
*/
public function setAvatar($avatar)
{
    $this->avatar=$avatar;
}

/**
* @return text
*/
public function getAvatar(){
    return $this->avatar;
}

```

Puis enfin, nous allons rajouté une ligne dans une fonction du nom « defSubdir » qui permet de définir l’endroit où va être upload l’image, donc son dossier de destination.

On obtient donc comme fonction :

```

public function defSubDir()
{
    return array(
        "cni"=>"InvoiceBundle/uploads/Employee/cni",
        "rib"=>"InvoiceBundle/uploads/Employee/rib",
        "contrat"=>"InvoiceBundle/uploads/Employee/contrat",
        "avatar"=>"InvoiceBundle/uploads/Employee/avatar"
    );
}

```

Puis pour finir, pour mettre a jour notre BDD, nous allons lancé la commande suivante :
 php app/console doctrine:schema:update --force

Ensuite, pour permettre l’envoi de l’image, nous allons rajouter un champ dans notre formulaire :

```

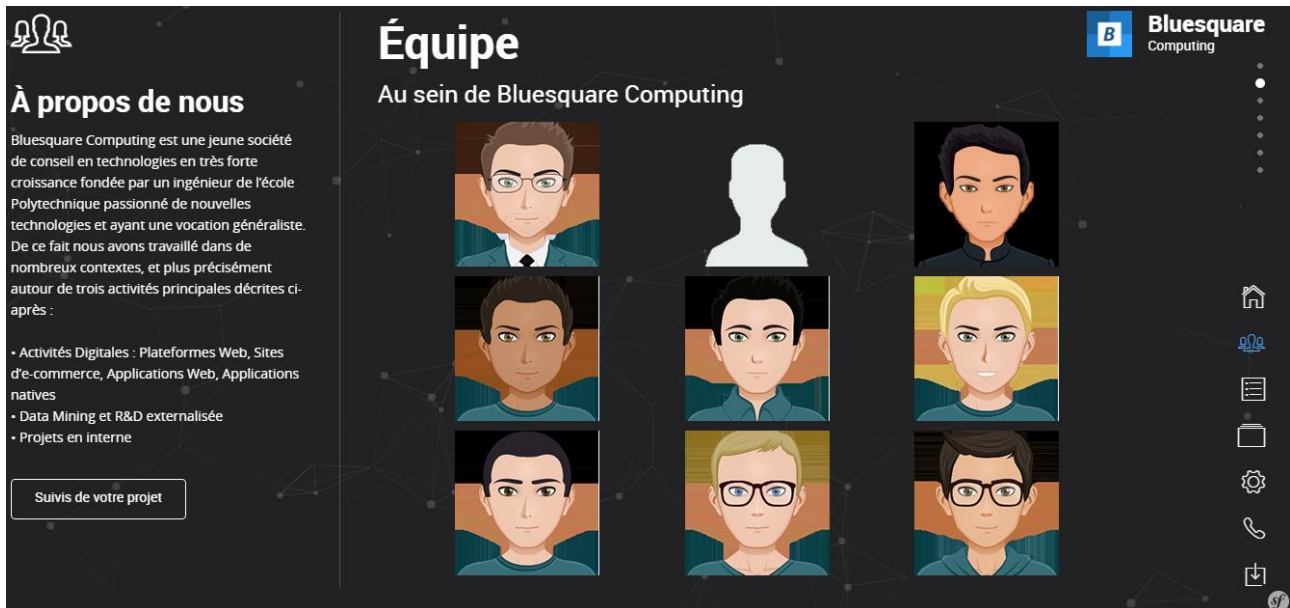
$builder->add('avatar','file', array('label' => 'avatar', 'required' => "", 'data_class' => null, 'attr'
=> array('class' => 'form-control') ));
        $builder->get("avatar")->addModelTransformer(new
CallbackTransformer(function ($originalDescription){return
"";},function($submittedDescription) {return $submittedDescription;}));

```

Ensuite, maintenant que l’upload de l’image est possible, j’ai décidé d’utiliser une requête afin de récupérer toutes les données présente dans la table Employee de notre BDD. On va donc y récupérer tous les employés. Puis dans la vue, on va effectuer une boucle ({% for equipe1 in equipe % } pour affiché tous les employés dont leur nom et prénom avec {{equipe1.firstname}} et {{equipe1.lastname}} (voir annexe 9)

La requête est :

```
- $equipe =  
$this->getDoctrine()->getRepository('BCInvoiceBundle:Employee')->findAll();
```



Puis pour finir, il faut que chaque employée puisse avoir son propre popup afin d'y ajouter leur description. Pour se faire, on a utilisé un plugin se nommant « magnific popup ». Celui-ci va nous permettre de faire appel à une vue contenant le popup.

On crée donc une nouvelle vue qui comprend un popup général mais qui s'adaptera en fonction de l'id de l'employée récupéré. (voir annexe 10)

Ensuite, on y rajoute une root dans notre fichier routing.yml (voir annexe 11)

Puis enfin, on ajoute une nouvelle fonction dans notre controller, qui va permettre d'envoyer dans cette nouvelle vue les informations comprises dans la table Employée et qui va permettre de renvoyer la vue. (voir annexe 12)

On obtient par exemple :



III) Conclusion :

J'ai donc appris durant ce stage le fonctionnement du framework symfony2. J'ai pris un grand plaisir à m'investir sur les différents projets cités. J'ai appris de jours en jours de nouvelles choses, ce fut très enrichissant. Je tiens à remercier Michel Jautzy qui m'a permis d'en arriver là ainsi que toute l'équipe de Bluesquare qui m'ont accompagné et aidé pendant tout le mois de Juin.

IV) Lexique:

Data Mining : (le data mining regroupe l'ensemble des technologies susceptibles d'analyser les informations d'une base de données marketing pour y trouver des informations utiles à l'action marketing et d'éventuelles corrélations significatives et utilisables entre les données.)

Start up: (Une startup, ou jeune pousse, est une jeune entreprise innovante à fort potentiel de croissance qui fait souvent l'objet de levées de fonds.)

Framework : un **framework, cadre applicatif, cadre d'applications, cadriceil, socle d'applications**¹ ou encore une **infrastructure de développement**² désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.

Template: Le mot "template" est un anglicisme qui désigne un exemple sur lequel on se base pour concevoir un logiciel, un design de site.

Back-office : Sur internet, la notion de back-office correspond à la partie du site internet qui n'est visible que par l'administrateur du site et qui permet de **gérer le contenu**, les fonctionnalités.

Front-office : Le front-office est, lui, la partie d'un site internet qui est visible par les internautes.

Big-data : désignent des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données ou de gestion de l'information.

Symfony : Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

Controller : Un contrôleur contient la logique concernant les actions effectuées par l'utilisateur.

Routing : Permet de définir des URL .

View : Une vue contient la présentation de l'interface graphique.

V) Annexe:

Annexe 1 :

```
<header class="title">
  <h2><strong>Hello </strong></h2>
  <div><h3 id="IdAccueil" style="display: none;"> {{ user.firstname }} {{ user.lastname }}<script src="
  {{ asset('js/affiche.js') }}"></script></h3></div>
</header>
<!-- desc -->
<div id="hello" class="desc">
  <p></p>
</div>
```

Annexe 2 :

```
1  //var result = '';
2  var id='IdAccueil';
3  var str = document.getElementById(id).textContent;
4  var code = str.split("");
5  var index = 0;
6  var sentence = '';
7  var intervalId = setInterval(codein, 150);
8
9  console.log("text length = " + code.length);
10
11 function codein(){
12   sentence += code[index];
13   $('#hello p').text(sentence);
14   index++;
15   if (index == code.length + 1) {
16     $('#hello p').html("");
17     index = 0;
18     sentence = '';
19   }
20 }
```

```
<div class="header">
  <h3>Team</h3>
  <p>Bluesquare Computing est une jeune société de conseil en technologies en très forte croissance fondée
  par un ingénieur de l'école Polytechnique passionné de nouvelles technologies et ayant une vocation
  généraliste. De ce fait nous avons travaillé dans de nombreux contextes, et plus précisément autour de
  trois activités principales décrites ci-après :<br><br>
  • Activités Digitales : Plateformes Web, Sites d'e-commerce, Applications Web, Applications natives<br>
  • Data Mining et R&D externalisée<br>
  • Projets en interne</p>
</div>

<div class="cta-btns">
  <a class="btn arrow-circ-btn" href="#suivi">
    <span class="txt">Suivis de votre projet</span>
    <span class="arrow-icon"></span>
  </a>
</div>
```

Annexe 3 :

```
<!-- BEGIN Declaration of modals-->
<div id="imgContainer1" class="modal-img-container">
  <div id="modal-img1" class="modal-img">
    <div class="modal-img-header">
      <strong>Michel Jautzy</strong>
      <span class="close">X</span>
    </div>
    <div class="modal-img-body">
      
      <div class="descTeam">blabla</div>
    </div>
    <div class="modal-img-footer">
      <span class="close">Bluesquare Computing</span>
    </div>
  </div>
</div>
```

```
<header class="title-desc">
  <h2>Team</h2>
  <h4>Au sein de Bluesquare Computing</h4>
</header>
{#Première personne#}
<div class="imgTeam">
  <div id="myBtn1" class="overlayHover">
    <div class="textHover">Michel Jautzy</div>
  </div>
  

</div>
{#Deuxième personne#}
<div class="imgTeam">
<div id="myBtn2" class="overlayHover">
  <div class="textHover">Jonathan<br> Randrianarison</div>
</div>
  

</div>
{#Troisième personne#}
<div class="imgTeam">
<div id="myBtn3" class="overlayHover">
  <div class="textHover">Alexis Uong</div>
</div>
  

</div>
{#Quatrième personne#}
<div class="imgTeam">
<div id="myBtn4" class="overlayHover">
  <div class="textHover">Nicolas Boullet</div>
</div>
  

</div>
```



```

var container = document.getElementById('imgContainer1');
var modal = document.getElementById('modal-img1');
var img = document.getElementById("myBtn1");

var container2 = document.getElementById('imgContainer2');
var modal2 = document.getElementById('modal-img2');
var img2 = document.getElementById("myBtn2");

var container3 = document.getElementById('imgContainer3');
var modal3 = document.getElementById('modal-img3');
var img3 = document.getElementById("myBtn3");

var container4 = document.getElementById('imgContainer4');
var modal4 = document.getElementById('modal-img4');
var img4 = document.getElementById("myBtn4");

var container5 = document.getElementById('imgContainer5');
var modal5 = document.getElementById('modal-img5');
var img5 = document.getElementById("myBtn5");

var container6 = document.getElementById('imgContainer6');
var modal6 = document.getElementById('modal-img6');
var img6 = document.getElementById("myBtn6");

var container7 = document.getElementById('imgContainer7');
var modal7 = document.getElementById('modal-img7');
var img7 = document.getElementById("myBtn7");

var container8 = document.getElementById('imgContainer8');
var modal8 = document.getElementById('modal-img8');
var img8 = document.getElementById("myBtn8");

var mainMenu = document.getElementById('mainMenu');
var close = document.getElementsByClassName("close")[0];
var close2 = document.getElementsByClassName("close2")[0];
var close3 = document.getElementsByClassName("close3")[0];
var close4 = document.getElementsByClassName("close4")[0];
var close5 = document.getElementsByClassName("close5")[0];
var close6 = document.getElementsByClassName("close6")[0];
var close7 = document.getElementsByClassName("close7")[0];
var close8 = document.getElementsByClassName("close8")[0];

```

```

/*Début Première personne*/

// When the user clicks the button, open the modal
img.onclick = function() {
    container.style.display = "block";
    mainMenu.style.display = "none";
}

// When the user clicks on <span> (x), close the modal
close.onclick = function() {
    container.style.display = "none";
    mainMenu.style.display = "block";
}

container.onclick = function() {
    container.style.display = "none";
    mainMenu.style.display = "block";
}

/*Début Seconde Personne*/

img2.onclick = function() {
    container2.style.display = "block";
    mainMenu.style.display = "none";
}

close2.onclick = function() {
    container2.style.display = "none";
    mainMenu.style.display = "block";
}

container2.onclick = function() {
    container2.style.display = "none";
    mainMenu.style.display = "block";
}

/*Début troisième personne */

img3.onclick = function() {
    container3.style.display = "block";
    mainMenu.style.display = "none";
}

```

Annexe 4 :

```
<ul class="timeline">

  <li class="event EventHeight" onclick="javascript:CountryChange('textDebut');" style="height: 90px;">Début

</li>
<li class="event EventHeight" onclick="javascript:CountryChange('textenCours');" style="height: 90px;">En cours

</li>
<li class="event" onclick="javascript:CountryChange('textFin');">Fin

</li>
</ul>

<div class="bas">
  <div id="textDebut" style="display: none" >
    <p>Texte que serait affiché lorsque qu'on click sur l'onglet Début</p>
    <p>Texte que serait affiché lorsque qu'on click sur l'onglet Début</p>
    <p>Texte que serait affiché lorsque qu'on click sur l'onglet Début</p>
    <p>Texte que serait affiché lorsque qu'on click sur l'onglet Début</p>
    <p>Texte que serait affiché lorsque qu'on click sur l'onglet Début</p>
  </div>
  <div id="textenCours" style="display: none">
    {% for timelineEnCours1 in timelineEnCours %}
    <p>< &nbsp; {{ timelineEnCours1.title | nl2br }} &nbsp;</p>
    {% endfor %}

  </div>
  <div id="textFin" style="display: none">
    {% for timelineFinis1 in timelineFinis %}
    <p>< &nbsp; {{ timelineFinis1.title | nl2br }} &nbsp;</p>

    {% endfor %}
  </div>
</div>
```

```

function CountryChange(id) {
  var1 = 'textDebut';
  var2 = 'textenCours';
  var3 = 'textFin';
  if (id == var1){

    if (document.getElementById(var1).style.display == "block")
    {
      document.getElementById(var1).style.display = 'none';
      document.getElementById(var1).style.visibility = 'none';
    }
    else
    {
      document.getElementById(var1).style.display = 'block';
      document.getElementById(var2).style.display = 'none';
      document.getElementById(var3).style.display = 'none';
      document.getElementById(var1).style.visibility = 'visible';
    }
  }

  if (id == var2){

    if (document.getElementById(var2).style.display == "block")
    {
      document.getElementById(var2).style.display = 'none';
      document.getElementById(var2).style.visibility = 'none';
    }
    else
    {
      document.getElementById(var2).style.display = 'block';
      document.getElementById(var1).style.display = 'none';
      document.getElementById(var3).style.display = 'none';
      document.getElementById(var2).style.visibility = 'visible';
    }
  }
}

```

Annexe 5 :

```

<div class="items-slide">
  <ul class="c-features-list row swiper-wrapper my-gallery">
    <!-- item -->
    <li class="column swiper-slide">
      <a href="#project_url_1">
        <div class="item-img">
          <div class="img bg-img" data-image-src="{ { asset('images/img/items/airnjobs.jpg') } }"></div>
        </div>
        <div class="item-desc soulign">
          <h4>Application Cross-Plateform</h4>
          <h3 class="title soulign">Air8Jobs </h3>
          <div class="desc soulign">
            <p>Air8Jobs est l'application de référence pour le recrutement dans l'aéronautique, disponible sur le store IOS et Android.</p>
          </div>
        </div>
      </a>
    </li>
    <!-- item -->
  </ul>
</div>

```

Annexe 6 :

```
<div class="wrapper anim col-md-9">
  <!-- title and description -->
  <div class="title-desc">
    <h2>Timeline Concept</h2>
  </div>
  <div id="contentStructure">
    <ul class="timelineStructure">
      <li class="eventStructure" data-date="Préproduction">
        <!--<h3>Registration</h3>-->
        <i class="fa fa-server" aria-hidden="true" style="font-size:24px;"></i>
        <p class="pStructure"> Mise en place d'un serveur de préproduction.</p>
      </li>
      <li class="eventStructure" data-date="Codage">
        <!--<h3>Opening Ceremony</h3>-->
        <i class="fa fa-code" aria-hidden="true" style="font-size:24px;"></i>
        <p class="pStructure"> Réalisation du codage par l'équipe.</p>
      </li>
      <li class="eventStructure" data-date="Refactoring">
        <!--<h3>Main Event</h3>-->
        <i class="fa fa-server" aria-hidden="true" style="font-size:24px;"></i>
        <p class="pStructure"> Après test, envoi sur le serveur principal.</p>
      </li>
      <li class="eventStructure" data-date="Publication">
        <!--<h3>Closing Ceremony</h3>-->
        <i class="fa fa-download" aria-hidden="true" style="font-size:24px;"></i>
        <p class="pStructure"> Publication.</p>
      </li>
    </ul>
  </div>
</div>
```

Annexe 7 :

```
<div id="map"></div>
<div class="adresse"> <p><span class="glyphicon glyphicon-map-marker"></span> 14 rue Soleillet, Paris, France</p>
<p><span class="glyphicon glyphicon-phone"></span> 06.01.02.03.04</p>
<p><span class="glyphicon glyphicon-envelope"></span> <A HREF="mailto:michel@bluesquarecomputing.com">michel@bluesquarecomputing.com
</A></p></div>
```

```
function initMap() {
  var uluru = {lat: 48.8654688, lng: 2.39147339999999953};
  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 14,
    center: uluru,
    styles: [{"featureType": "all", "elementType": "labels.text.fill", "stylers": [{"saturation": 36}, {"color": "#000000"}, {"lightness": 40}]}]);
  var marker = new google.maps.Marker({
    position: uluru,
    map: map
  });
}
```

Annexe 8 :

```

<div class="wrapper anim col-md-9">
  <center><header class="title-desc">
    <h2 class="page-title">Recapitulatif des fichiers</h2>
    <div class="title">
      <br><h3>Vos documents</h3>
    </div>
    <p>Votre facture : </p>
    <div class="cta-btns">
      <i class="fa fa-file-text-o" aria-hidden="true" style="font-size: 24px; margin-right: 10px;"></i>
      <a class="btn arrow-circ-btn" href="#register">
        <span class="txt">Facture</span>
        <span class="arrow-icon"></span>
      </a>
    </div><br>
    <p> Vos démarches : </p>
    <div class="cta-btns">
      <i class="fa fa-sticky-note-o" aria-hidden="true" style="font-size: 24px; margin-right: 10px;"></i>
      <a class="btn arrow-circ-btn" href="#register">
        <span class="txt">Démarches</span>
        <span class="arrow-icon"></span>
      </a>
    </div><br>
    <p> Votre suivis : </p> <div class="cta-btns">
      <i class="fa fa-hand-o-right" aria-hidden="true" style="font-size: 24px; margin-right: 10px;"></i>
      <a class="btn arrow-circ-btn" href="#register">
        <span class="txt">Suivis</span>
        <span class="arrow-icon"></span>
      </a>
    </div><br>
  </center>
</div>

```

Annexe 9 :

```

<div class="wrapper anim col-md-9">
  <!-- title and description -->
  <header class="title-desc">
    <h2>Équipe</h2>
    <h4>Au sein de Bluesquare Computing</h4>
  </header>
  <% for membres in equipe %>
  <div class="imgTeam">
    <div id="myBtn1" class="overlayHover">
      <div class="textHover"><a href="{ { path('bc_popup_equipe', {'projectHash': projet.projectHash, 'id': membres.id}) } }" class="link-popup blanc">{ { membres.nom }}<br> { { membres.prenom }}</a></div>
    </div>
    <% if membres.avatar != null %>
      
    <% else %>
      
    <% endif %>
  </div>
  <% endfor %>
  <script type="text/javascript">
    $(document).ready(function () {
      $('.link-popup').magnificPopup({
        type: 'ajax',
        'closeOnContentClick': false,
        'closeOnBgClick': false
      });
      $(document).on('click', '.close-popup', function (e) {
        e.preventDefault();
        $.magnificPopup.close();
      });
    });
  </script>
</div>

```

Annexe 10 :

```
{% block body %}
<div id="imgContainer1" class="modal-img-container">
  <div id="modal-img1" class="modal-img">
    <div class="modal-img-header">
      <strong>{{equipe.nom}} {{equipe.prenom}}</strong>
      <span class="close-popup">X</span>
    </div>
    <div class="modal-img-body">
      {% if equipe.avatar != null %}
        
      {% else %}
        
      {% endif %}
      <p class="descTeam">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sit amet ligula non augue dictum dictum porta sit amet tortor. Vivamus ut consequat eros. Pellentesque eget hendrerit velit. Donec eu consectetur odio, in condimentum nulla. Nullam mattis est scelerisque, porttitor tortor et, iaculis massa. Nam vestibulum nibh at accumsan bibendum. Nam luctus non dui quis tincidunt. Aenean ac velit quis urna volutpat efficitur id ac urna. Proin laoreet nunc non ipsum venenatis efficitur. Vestibulum facilisis urna vel ex scelerisque, a bibendum elit volutpat. Maecenas viverra, nisi ac eleifend mattis, arcu erat pellentesque enim, vitae gravida magna tortor ac eros. Praesent lectus purus, rhoncus ut augue eu, sagittis mollis nibh. Donec feugiat mi et tellus accumsan ornare. Morbi dapibus et felis nec tincidunt.
      Donec sodales orci diam, eget aliquam elit bibendum nec. Maecenas rhoncus lorem et volutpat tempus. Pellentesque eu augue vitae ipsum ultrices tempor ut sit amet mi. Aenean tincidunt enim ac condimentum blandit. Mauris sollicitudin dapibus nulla, vitae imperdiet arcu ultricies sed. Etiam euismod convallis auctor. Nam malesuada vitae massa eget bibendum. In placerat risus eget magna eleifend, vitae placerat enim pulvinar. Suspendisse non arcu turpis. Morbi et volutpat felis, quis posuere lacus.</div>
    </div>
    <div class="modal-img-footer">
      <span class="close-popup">Bluesquare Computing</span>
    </div>
  </div>
</div>
{% endblock %}
```

Annexe 11 :

```
bc_popup_equipe:
  path: /popup/project/{projectHash}/{id}
  defaults: { _controller: BCInvoiceBundle:ClientProjectPage:popup }
  requirements:
    methods: POST
```

Annexe 12 :

```
public function popupAction($projectHash, $id){
    $projet = $this->getDoctrine()->getRepository('BCInvoiceBundle:Projet')->findOneBy(array("projectHash" => $projectHash));
    $equipe = $this->getDoctrine()->getRepository('BCInvoiceBundle:Employee')->find($id);
    return $this->render('BCInvoiceBundle:Projet:popup.html.twig', array('projet' => $projet, 'equipe' => $equipe));
}
```