

Rapport de stage

Première année - BTS SIO

Table des matières

1. Présentation de l'entreprise.....	2
2. Présentation des besoins.....	2
3. Liste des logiciels et matériels utilisés.....	3
4. Présentations, descriptions et méthodes des travaux réalisés.....	4
4.1. Création de l'application.....	4
4.2. Rendu de l'application.....	8
4.3. Lien avec la base de données.....	8
5. Documentation de l'application.....	12
6. Conclusion.....	12
7. Remerciements.....	13



1. Présentation de l'entreprise :



Le SILPC propose des services, des outils ainsi que des prestations d'expertise et d'accompagnement.

Le SILPC répond aux besoins des systèmes d'informations de santé en France métropolitaine et DOM TOM depuis plus de quarante ans.

Le SILPC est réparti dans deux grandes villes, Limoges et Poitiers, deux sites sont à Limoges pour un site à Poitiers.

Siège social GIP SILPC, 2 rue Jean Monnet, 87170 Isle

Second site sur Limoges, 8 rue Soyouz, 87280 Limoges

Pôle République III, 14 rue des Landes, 86000 Poitiers

Lors de la réalisation de mon stage de première année, j'ai eu la possibilité de découvrir le second site de Limoges, qui est très récent et très accueillant.

Durant ces cinq semaines de stage, j'ai rencontré plusieurs personnes, dont Madame Prengère Virginie, Monsieur Breda Laurent et Monsieur Lecompte Valentin. Ce sont les trois employés du SILPC avec qui j'ai passé le plus de temps. Ci-dessous un tableau les présentant.

Nom	Prénom	Rôle au sein du SILPC	Ancienneté
Prengère	Virginie	Informaticienne de diffusion - Coordinatrice	7 ans et demi
Breda	Laurent		
Lecompte	Valentin		

2. Présentation des besoins :

Un besoin nous a été présenté, le SILPC souhaite un outil de réservation des salles de réunions, et des bureaux, notamment pour les personnes qui alternent souvent entre présentiel et distanciel.









➔ Premier objectif : réaliser cette application liée à une base de données.

Ensuite, nous devons faire en sorte que cette application soit facilement déployable sur les postes des différents utilisateurs.

➔ Second objectif : adapter l'application pour une installation simple sur les postes.

3. Liste des logiciels et matériels utilisés :

Pour la réalisation complète du projet, nous avons eu besoins de plusieurs outils. J'ai choisi des les représenter sous forme de tableau.

Image	Nom	Description	Rôle dans le projet
	WampServer	WampServer est un environnement comprenant trois serveurs, un interpréteur de script, ainsi que phpMyAdmin pour l'administration Web des bases MySQL.	Dans notre projet, Wamp est utile pour accueillir PHP, Apache et une base de données MySQL.
	Symfony 5.2	Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP.	Dans le projet, Symfony sert de base. Il est capable de lire les routes et de trouver les fichiers rapidement. De plus il accède à la base de données sans faire de requête importante.
	MySQL	MySQL est un système de gestion de bases de données relationnelles.	MySQL nous sert de gestionnaire de base de données.
	Git	Git est un logiciel de gestion de versions décentralisé.	Git nous a servi en lien avec Gitlab, de manière à créer des versions et assembler des fonctionnalités.
	Visual Studio Code	Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.	Visual Studio code nous a servi d'IDE, de plus on a ajouté certaines extensions comme Twig ou Docker.
	Microsoft Teams	Microsoft Teams est une application de communication collaborative propriétaire officiellement lancée par Microsoft en novembre 2016.	Microsoft Teams nous a permis de rester en communication avec les différentes personnes qui s'occupent de nous, par exemple lorsqu'elles étaient en télétravail.
	Docker	Docker est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels.	Docker nous a servi pour la dernière partie de notre stage, dans le but d'automatiser le déploiement de notre application sur les machines.
	Win'Design	Solution de modélisation d'Architecture d'entreprise, multi-angles de vue qui couvre l'ensemble de vos Cartographies Métier & SI.	Win'design nous a permis de réaliser le MCD de notre base de données.

Je tiens à rajouter, que l'outil qui nous a le plus servi est principalement Google. Pour chaque souci que nous rencontrons nous pouvons trouver une solution.

Enfin, nous travaillons sur une machine Windows 10 liée au réseau du SILPC.

4. Présentations, descriptions et méthodes des travaux réalisés :

Dans le cadre de ce stage de première année, monsieur Laurent Breda m'a fourni un projet très intéressant, dans un domaine qui me plaît car j'aimerais en faire mon métier.

Pour ce projet, je suis accompagné de Adam Repos, un stagiaire en BTS SIO spécialisé en SLAM du lycée Bahuet à Brive.

Comme indiqué dans la présentation des besoins, nous avons deux objectifs principaux. Le premier est de créer l'application et pour cela nous avons une contrainte : **utiliser le framework symfony**.

Pour le second objectif (faciliter le déploiement sur les machines) nous avons également une contrainte : **utiliser les containers Docker**.

(J'évoque le terme de contrainte, mais une fois les outils bien pris en main, nous sommes beaucoup plus productifs.)

4.1 Création de l'application :

Découverte de Symfony 5 :

Afin de créer l'application, je devais dans un premier temps découvrir le framework symfony qui me semble très puissant. Pour ce faire j'ai utilisé plusieurs méthodes :

1. Documentation de Symfony 5.2
2. Recherche Google
3. Tutoriaux YouTube

Globalement, j'ai réussi à comprendre Symfony, le reste de la compréhension se fera au cours de l'avancement du projet.

Pour utiliser Symfony correctement, j'ai utilisé le système d'annotation pour les routes. Une route est la reconnaissance d'un lien par Symfony, pour me rediriger vers la page demandée (en passant par un contrôleur).

```
class ReservationController extends AbstractController
{
    /**
     * @Route("/reservation", name="reservation")
     */
    public function Reservation(){}
}
```

Pour expliquer ce système d'annotation, on peut voir que Symfony va utiliser ce contrôleur dès que /reservation sera dans l'URL. La première fonction sera exécutée automatiquement dès l'entrée dans le contrôleur.

Ensuite, je dois rediriger l'utilisateur vers la vue. La vue est sous format twig.

Qu'est-ce que Twig ? (Wikipédia)

*Twig est un moteur de templates pour le langage de programmation **PHP**, utilisé par défaut par le framework Symfony.*

Ainsi, j'ai compris que Twig est une aide à PHP, et comme nous le verrons plus tard, cela m'aide bien.

Pour en revenir au fait, je dois diriger l'utilisateur vers la vue. Pour se faire, j'utilise la fonction `render()` incluse dans Symfony.

```
return $this->render("reservation.twig");
```

Le fichier `reservation.twig` est situé dans le dossier `Templates` à la racine du projet. Symfony peut directement trouver le fichier sans préciser le chemin d'accès.

Et voilà ! Ma route est configurée, pour rappeler les étapes :

1. L'utilisateur entre `/reservation` dans l'URL
2. Il est redirigé par Symfony vers le contrôleur
3. Le contrôleur redirige vers la vue au format twig.

Maintenant, l'Application est mise en place. Il faut donc créer cette page pour obtenir un premier aperçu. Pour se faire il va falloir utiliser les langages HTML / CSS et Javascript.

Pour m'aider d'une base, j'ai réalisé ce schéma, inspiré du schéma de présentation du projet de Laurent BREDA.

Schéma lors de l'ouverture de la page

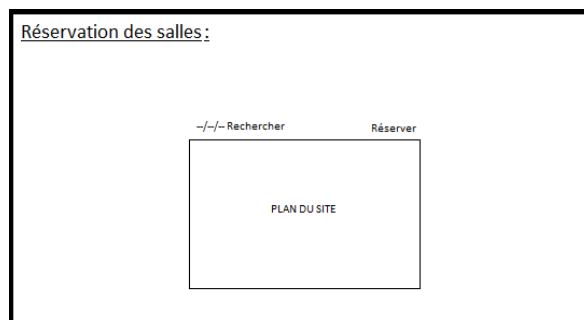
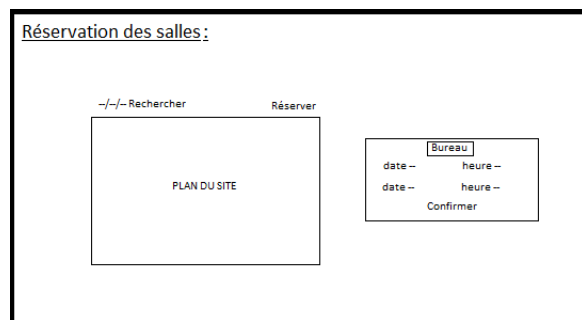


Schéma avec le formulaire de réservation :



Utilisation d'HTML / CSS et Javascript :

HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Réservation des salles - SILPC</title>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<body>
  <h1>Réservation des salles</h1>
  <div class="line"></div>
  <div class="main">
    <div class="buttons">
      <div class="selecteur-date">
        <form>
          <input type="datetime-local" name="date-selecteur">
          <input type="submit" name="rechercher" value="Rechercher">
        </form>
      </div>
      <div class="reserver">
        <button class="button-reserver">Réserver</button>
      </div>
    </div>

    <div class="plan">
      
    </div>
  </div>
</body>
</html>
```

Voici le code source HTML pour la page d'accueil, il sera plus tard introduit dans le fichier Twig, accompagné du formulaire de réservation.

Ce code source est composé d'un titre, d'une image et de quelques inputs.

CSS:

En ce qui concerne mon fichier CSS, il est assez imposant, je vais donc présenter le principal.

```
body{
  margin: 0;
  padding: 0;
  font-family: 'Montserrat', sans-serif;
}

h1{
```

```

    font-size: 38px;
    font-weight: 1000;
    margin-left: 5px;
}

.line{
    background-color: black;
    height: 10px;
    width: 25vw;
    border-top-right-radius: 5px;
    border-bottom-right-radius: 5px;
}

.plan{
    margin-top: 10px;
    height: 676px;
    border: 4px solid black;
}

.selecteur-date input[type=datetime-local]{
    height: 30px;
    width: 200px;
}

```

Sur ce code source représentant les éléments principaux, nous pouvons voir que je stylise le titre (plus gros et un peu sur la droite), j'ajoute une ligne, je place le plan et je lui mets une bordure et enfin, j'agrandi pour bouton de sélecteur de date.

Javascript :

La partie Javascript est composé d'une librairie nommée TimelineMax, créée par GreenSock. Elle a pour but de gérer des courtes animations sous formes de Timeline.

```

var AffichageReservation = new TimelineMax({paused: true});

AffichageReservation
.to(main, 1, {x: -800})
.from(formulaire, 1,{visibility: "hidden"}, "-=1")
.from(formulaire, 1,{opacity: 0})

boutonReservation.addEventListener("click", function(){
    AffichageReservation.timeScale(1.5);
    if(ouvert){
        AffichageReservation.timeScale(4);
        AffichageReservation.play();
        AffichageReservation.reverse();
    }
});

```

```

    ouvert = false;
  } else{
    AffichageReservation.play();
    ouvert = true;
  }
});

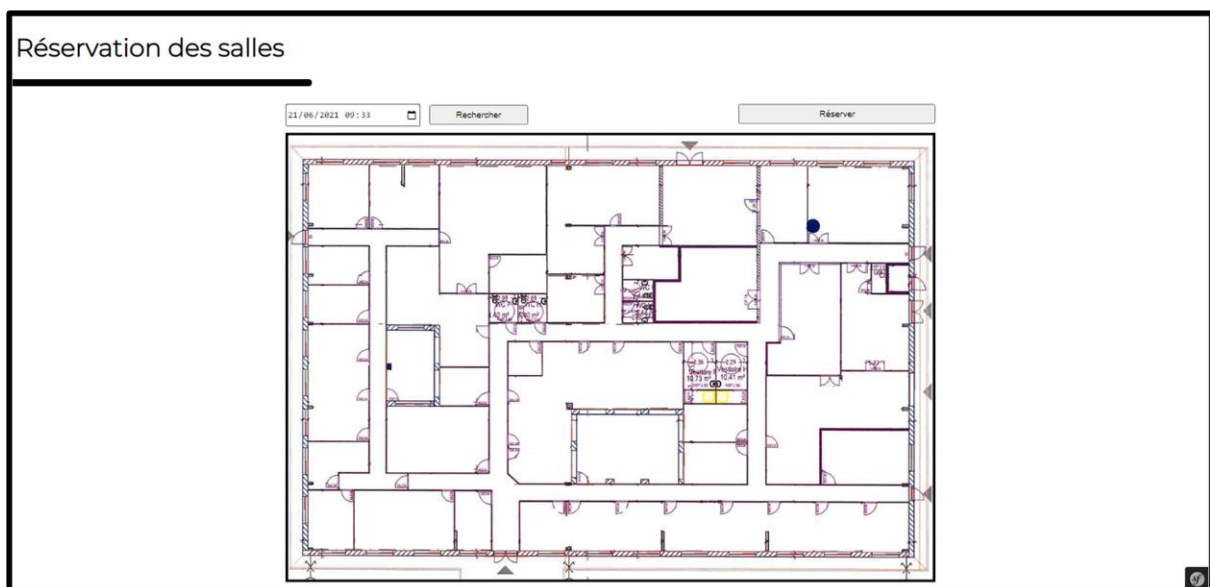
```

Cette partie de code Source est simple à comprendre. Globalement, je vais créer une variable de Timeline, puis je configure ses animations, opacité, visibilité et vitesse d'apparition.

Puis j'ajoute un événement, lorsque je clique sur l'élément boutonReservation, je démarre l'animation, par ailleurs l'animation va plus vite pour se fermer que pour s'ouvrir.

4.2 Rendu de l'application :

L'application terminée physiquement ressemble à ceci :



Dans la prochaine section, nous allons afficher les bureaux en utilisant la base de données, et intégrer le formulaire de réservation.

4.3 Lien avec la base de données :

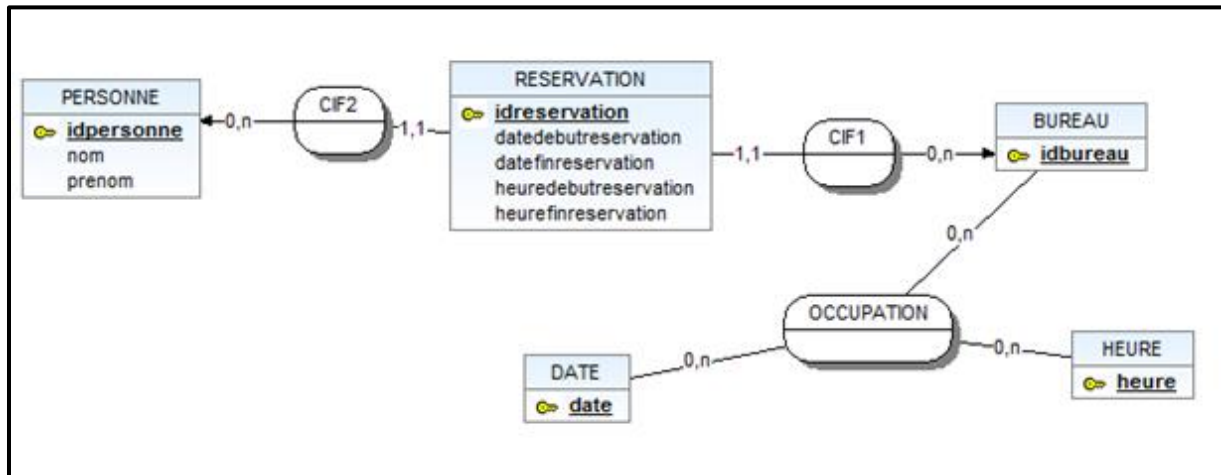
De manière à ajouter les différents bureaux sous formes de carrés, nous devons faire plusieurs étapes :

1. Créer la base de données.

- Repérer les positions largeur / hauteur des bureaux sur le plan (+ Les inscrire dans la base de données).
- Développer une fonction Javascript pour permettre de les afficher.
- Choisir la couleur en fonction de la disponibilité.
- Gérer le formulaire de réservation.

Dans un premier temps, je vais créer la base de données. Pour se faire j'utilise Win'design pour créer un premier aperçu, puis l'intégrer avec PhpMyAdmin.

MCD sous Win'design :



Base de données sur PhpMyAdmin :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
bureaux	Parcourir Structure Rechercher Insérer Vider Supprimer	92	InnoDB	latin1_swedish_ci	16,0 kio	-
etat	Parcourir Structure Rechercher Insérer Vider Supprimer	3	MyISAM	latin1_swedish_ci	2,1 kio	-
reservation	Parcourir Structure Rechercher Insérer Vider Supprimer	12	InnoDB	latin1_swedish_ci	16,0 kio	-
3 tables	Somme	107	MyISAM	latin1_swedish_ci	34,1 kio	0 0

Table Bureau :

		id_bureaux	largeur	hauteur
<input type="checkbox"/>	Éditer Copier Supprimer	ACC01	277	645
<input type="checkbox"/>	Éditer Copier Supprimer	DEM01	410	120
<input type="checkbox"/>	Éditer Copier Supprimer	DEM02	410	140
<input type="checkbox"/>	Éditer Copier Supprimer	DEM03	430	120

Table Etat :

		id_etat	nom_etat
<input type="checkbox"/>	Éditer Copier Supprimer	0	Disponible
<input type="checkbox"/>	Éditer Copier Supprimer	1	Occupé
<input type="checkbox"/>	Éditer Copier Supprimer	2	Fixe

Table Réservation :

		id_reservation	jour_debut	jour_fin	heure_debut	heure_fin	id_bureau	nom	prenom	id_etat
<input type="checkbox"/>	Éditer Copier Supprimer	3	2020-12-31	5000-01-01	00:00:00	23:59:59	DIR01	Direction	Direction	2
<input type="checkbox"/>	Éditer Copier Supprimer	4	2020-12-31	5000-01-01	00:00:00	23:59:59	DIR02	Direction	Direction	2
<input type="checkbox"/>	Éditer Copier Supprimer	5	2020-12-31	5000-01-01	00:00:00	23:59:59	DIR03	Direction	Direction	2
<input type="checkbox"/>	Éditer Copier Supprimer	68	2021-06-03	2021-06-03	16:32:00	17:32:00	REU03	test	test	1
<input type="checkbox"/>	Éditer Copier Supprimer	69	2021-06-04	2021-06-04	09:08:00	09:08:00	REU01	essa	esss	1

Développement de la fonction Javascript pour l'affichage :

```
function desk(left, top, color, bureau, date, heure){
    var panel = document.querySelector('.dot-plan');
    var cube = document.createElement('button');
    cube.setAttribute('class', 'button-desk bureau');
    cube.setAttribute('id', 'bureau');
    cube.style.left = left + 'px';
    cube.style.top = top + 'px';
    cube.style.backgroundColor = color;
    cube.value = bureau;
    cube.dataset.date = date;
    cube.dataset.heure = heure;
    panel.appendChild(cube);
}
```

Cette fonction va créer un élément « cube », puis lui ajouter les class **button-desk** et **bureau**. Puis mettre les différents paramètres de la fonction en place.

Une fonction similaire existe pour créer les salles de réunions, le seul paramètre qui change est la class, pour afficher un bouton un peu plus large.

Je vais pour la première fois utiliser le fichier Twig. Je vais récupérer toutes les réservations, tous les états et tous les bureaux depuis le contrôleur, pour l'envoyer dans le Twig. Puis je vais analyser ces résultats pour choisir la couleur à afficher.

Voici mon Contrôleur principal, qui redirige vers le fichier Twig en incluant les trois tableaux de la base de données :

```
$bureaux = $this->getDoctrine()->getRepository(Bureaux::class)->findAll();
$reservations = $this->getDoctrine()->getRepository(Reservation::class)->findAll();
$etat = $this->getDoctrine()->getRepository(Etat::class)->findAll();

return $this->render("reservation.twig", ['bureau'=>$bureaux, 'reservations'=>$reservations, 'etats'=>$etat]);
```

Et je l'analyse pour choisir correctement les paramètres de la fonction javascript :

```
{% set couleur = '#5DE659' %}

{% for bur in bureau %}
    {% for res in reservations %}
        {% if bur.idBureaux == res.idBureau %}
            {% if post.selecteur is defined %}
                {% if (res.jourDebut|date("Y-m-d")) == post.selecteur|date("Y-m-d") and
```

```

                                res.heureDebut|date("H:i") <= post.selecteur|date("H:i"))
and
                                (res.jourFin|date("Y-m-d") == post.selecteur|date("Y-m-
d") and
                                res.heureFin|date("H:i") >= post.selecteur|date("H:i")) or
                                (res.jourDebut|date("Y-m-d") < post.selecteur|date("Y-m-
d") and
                                res.jourFin|date("Y-m-d") > post.selecteur|date("Y-m-
d") and
                                (post.selecteur|date("H:i") > "00:00" and post.selecteur|d
ate("H:i") < "23:59")) %)
                                {% for etat in etats %}
                                    {% if etat.idEtat == res.idEtat %}
                                        {% if etat.nomEtat == "Disponible" %}
                                            {% set couleur = '#5DE659' %}
                                        {% endif %} {%# Fin du if Disponible #}
                                        {% if etat.nomEtat == "Occupé" %}
                                            {% set couleur = '#E13B3B' %}
                                        {% endif %} {%# Fin du if Occupé #}
                                        {% if etat.nomEtat == "Fixe" %}
                                            {% set couleur = '#C1C1C1' %}
                                        {% endif %} {%# Fin du if Fixe #}

                                        {% set dateDebutReserve = res.jourDebut|date("
Y-m-d") %}

                                        {% set heureDebutReserve = res.heureDebut|date
("H:i") %}

                                        {% set dateFinReserve = res.jourFin|date("Y-m-
d") %}

                                        {% set heureFinReserve = res.heureFin|date("H:
i") %}

                                    {% endif %} {%# Fin du if etat.idEtat == res.idEtat
#}

                                {% endfor %} {%# Fin du for etat in etats #}
                                {% endif %} {%# Fin de la vérification si le créneau de la
réservation est dans la date choisi #}
                                {% endif %} {%# Fin du if post.selecteur is defined #}
                                {% endif %} {%# Fin du if bur.idBureaux == res.idBureau #}
                                {% endfor %} {%# Fin du for res in reservations #}

```

Globalement, je vérifie la date et l'heure de chaque réservations de la base de données. Par rapport à l'heure sélectionner (post), puis je choisi la couleur à mettre en place.

Voici maintenant la fonction pour créer une nouvelle réservation :

```
function create(){
    $entityManager = $this->getDoctrine()->getManager();
    $reservation = new Reservation();
    $reservation->setJourDebut(\DateTime::createFromFormat('Y-m-
d', $_POST['datedebut']));
    $reservation->setJourFin(\DateTime::createFromFormat('Y-m-
d', $_POST['datefin']));
    $reservation-
>setHeureDebut(\DateTime::createFromFormat('H:i', $_POST['heuredebut']));
    $reservation-
>setHeureFin(\DateTime::createFromFormat('H:i', $_POST['heurefin']));
    $reservation->setIdBureau($_POST['choixbureaux']);
    $reservation->setNom($_POST['nom']);
    $reservation->setPrenom($_POST['prenom']);
    $reservation->setIdEtat("1");
    $entityManager->persist($reservation);
    $entityManager->flush();
}
```

Elle est directement reliée au formulaire de réservation.

5. Documentation de l'application :

Un de nos objectifs est de réaliser une documentation. Pour cela j'ai utilisé le système de Gitlab, en utilisant Markdown.

Nous devons présenter :

- L'outil
- Les prérequis
- Le guide d'utilisation
- Le modèle de données

Cette documentation est disponible en annexe.

6. Conclusion :

En conclusion de rapport, je peux dire que ce Stage d'environ 5 semaines m'a beaucoup plu, je suis fier d'avoir choisi l'entreprise qui me correspond.

J'ai appris beaucoup de chose, et je me rend compte qu'être sur un projet concret permet de découvrir encore plus, par exemple le framework Symfony, et j'ai également appris à savoir utiliser les documentations des éditeurs d'outils. Je sais aussi réaliser une documentation pour un projet que j'ai créé.

J'espère pouvoir retourner pour mon prochain stage dans cette entreprise, pour pouvoir continuer mon projet ou même en découvrir un nouveau.

Je ne regrette pas ce stage.

7. Remerciements :

Premièrement, je tiens à remercier Madame Prengère Virginie, qui est mon maitre de stage et qui a sû m'accompagner et s'intéresser à ce que je faisais, et quels sont mes projets en BTS et après. Elle m'a également mis en relation avec les personnes importantes pour mon stage.

Je souhaite remercier Monsieur Breda Laurent, le chef de projet qui a été capable de nous expliquer correctement les besoins, et nous accompagner sur les outils à utiliser et pourquoi. Il a également réaliser plusieurs réunions pour présenter le projet, dans le but de nous conseiller.

Puis, je remercie Monsieur Lecompte Valentin, qui a été capable de nous aider tout le long du projet, sur nos questions et nos problèmes au cours du développement de l'application. De plus, il m'a permis d'apprendre beaucoup par rapport aux fichiers de configurations, et Gitlab.

Enfin, je désire remercier Madame Bourgeois Agnes, Monsieur Pasqualini Claude et Monsieur Bogusz Thierry, qui m'ont entouré pendant ces semaines de stages, pour répondre à mes questions et mes besoins. Ainsi que pour me rendre visite en stage et s'intéresser au projet. Et biensur, à m'accompagner durant cette première année de BTS SIO.