

## AMELIORATION DE LA LISTE INCENDIE

Rapport de stage de BTS SIO



Stage : 03/01/2022 – 18/02/2022

Société : BIC Rasoirs

Tuteur : Pascal Hennion – Responsable logistique et informatique – BIC Rasoirs

Tuteur : Jérémy Brassart – Chef de projets informatiques – BIC Rasoirs

Responsable pédagogique : Agnès Kintzler – Enseignante – Lycée Jean Rostand

## Table des matières :

I.	Présentation de l'entreprise .....	2
1.	BIC de 1945 à aujourd'hui.....	2
2.	BIC Rasoirs, l'usine de production .....	4
3.	Organisation de BIC Rasoirs.....	6
4.	Le MIS, le service informatique .....	7
I.	Mes Missions.....	9
1.	Liste des missions .....	9
2.	Environnement de travail .....	9
II.	Description de la mission « Amélioration de la liste incendie » .....	9
1.	Genèse du projet .....	9
2.	Planning .....	12
3.	Développement des améliorations sur la liste incendie .....	12
a.	Fonctionnement de la liste incendie.....	12
b.	Résultat final .....	14
c.	Présentation du fichier PDF.php.....	17
d.	Présentation du fichier ConnexionBDD.xml .....	26
e.	Présentation du fichier liste_incendie.php.....	28
III.	Description de la mission « Configuration et installation de pc Windows 10 » .....	30
1.	Genèse du projet .....	30
2.	Matériel mis à disposition .....	30
3.	Planning .....	31
4.	Formatage du PC avec clé bootable .....	31
5.	Processus de configuration des pc .....	33
6.	Mise en place des nouveaux ordinateurs.....	35
IV.	Description de la mission « Amélioration de la saisie de la packing list ».....	38
1.	Genèse du projet .....	38
2.	Planning .....	39
3.	Développement des améliorations de la saisie de packing list .....	39
a.	Résultat final .....	39
b.	Présentation du script BRISE.....	40

c.	Présentation du script PHP .....	45
V.	Description de la mission « Ajout d'une fonctionnalité dans AP2M » .....	50
1.	Genèse du projet .....	50
2.	Planning .....	50
3.	Résultat final .....	51
4.	Présentation du script.....	51
VI.	Bilan .....	60
VII.	Remerciement .....	61
VIII.	Annexes.....	62
	Annexe 1 : Fichier liste_incendie.php initiale .....	62
	Annexe 2 : Fichier verification.php .....	63
	Annexe 3 : Fichier Horoquartz_générateur_pdf.bat .....	63
	Annexe 5 : Fichier CSV non Standard SANDVIK mis au format Excel .....	64
	Annexe 6 : Fichier CSV non Standard JINDAL mis au format Excel avec indication des colonnes du format standard.....	64
	Annexe 7 : La fin du contenu du fichier Preparation_auto_packing_list.php .....	65
	Annexe 8 : Méthode bfp_envoyer_email.....	68

## I. PRESENTATION DE L'ENTREPRISE

### 1. BIC de 1945 à aujourd'hui....

**BIC** est une société française fondée le 25 octobre 1945 par Marcel BICH et son associé Edouard BUFFARD, dont le siège social se trouve à Clichy.

En premier lieu nommée **PPA** pour **P**orte-plume, **P**orte-mines et **A**ccessoires et fondée dans le but de produire des pièces détachées pour stylos plume et porte-mines, dans une usine à Clichy, elle se renommera définitivement BIC en 1953 pour une meilleure identification et prononciation dans toutes les langues du monde.



Marcel BICH  
(1914-1994)



Edouard BUFFARD  
(1908-1996)



Premier logo de BIC  
(1953)

En 1950 BIC se fait une place parmi les plus grands acteurs du marché mondial du stylo grâce à la commercialisation d'un produit : le **BIC Cristal**. Vendu à plus de 100 milliards d'exemplaires, il reste encore aujourd'hui le stylo le plus vendu dans le monde. La société élargit son activité en lançant dès 1973 la fabrication et la commercialisation de briquets jetables, ensuite celle des rasoirs en 1975, suivi d'une nouvelle branche de société, BIC Sport, spécialisé dans la fabrication de planches à voile et de planches de surf. Pour finir, sa dernière innovation est la commercialisation de tablette numérique BIC en partenariat avec Intel, disponible en France depuis le premier trimestre 2013, elle est destinée principalement à un usage pédagogique dans les écoles primaires. Aujourd'hui la société s'articule autour de 3 domaines principaux : les articles de papeterie, les briquets et les rasoirs.



Vente BIC par jour dans le monde

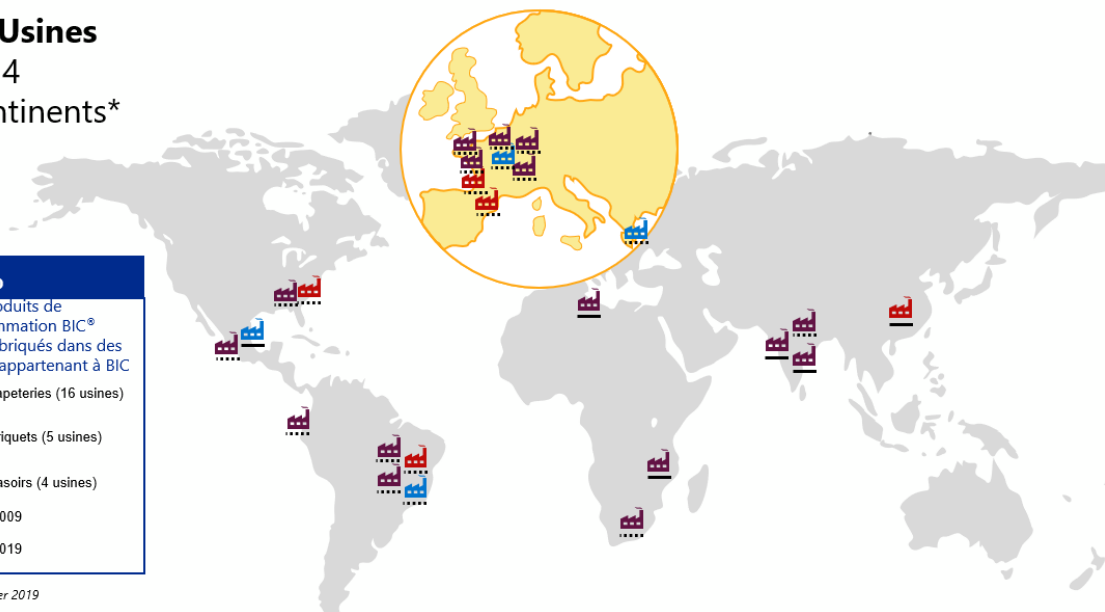
De plus, la société s'est toujours projetée sur le marché mondial en s'implantant dans différents pays au fil des années : en Italie en 1954, au Brésil en 1956, au Royaume-Uni, Australie, Nouvelle-Zélande et Scandinavie en 1957, aux États-Unis, au Moyen-Orient en 1958 et en Asie en 1997.

Aujourd'hui BIC est présent sur les cinq continents, dans plus de 162 pays employant ainsi au total plus de **9.600 personnes**, dont environ 3.900 en Europe. Elle possède également 3,2 millions de points de vente qui distribuent les produits BIC et 9 700 sociétés collaborent avec BIC à travers le monde.

**26 Usines**  
sur 4  
Continents\*



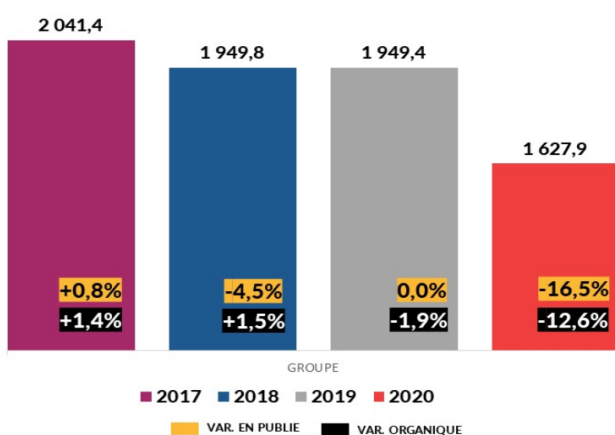
\*En Février 2019



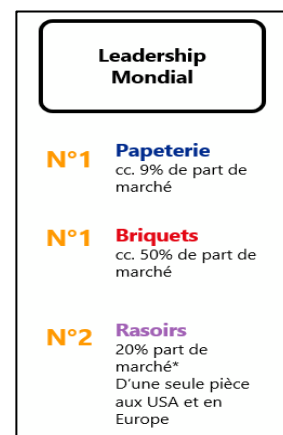
Empreinte de fabrication internationale

Les productions du groupe BIC sont généralement destinées au grand public et représentent 85 % de son chiffre d'affaires. Ses produits sont fabriqués dans 26 usines réparties sur 4 continents (dont 7 en France).

Réalisant en 2019 un chiffre d'affaires de 1 949,4 millions d'euros, BIC comme beaucoup d'autres entreprises n'a pas été épargnée par la crise sanitaire et a vu son chiffre d'affaires de 2020 chuter à 1 627,9 millions. Cela n'empêcha pas la société de rester leader sur le marché en étant n°1 en Papeterie en Europe, en Inde et Amérique latine ainsi qu'en Océanie par exemple.



Chiffres d'affaires Groupe (en millions d'euros)



Pour finir, le groupe BIC possède de nombreuses filiales comme BIC Rasoirs spécialisée dans la fabrication de rasoirs de la firme, Bic Kids qui est la marque de produits de coloriage du Groupe BIC destinée aux enfants ou encore BIC Graphic qui est le principal fournisseur de produits promotionnels de la marque.

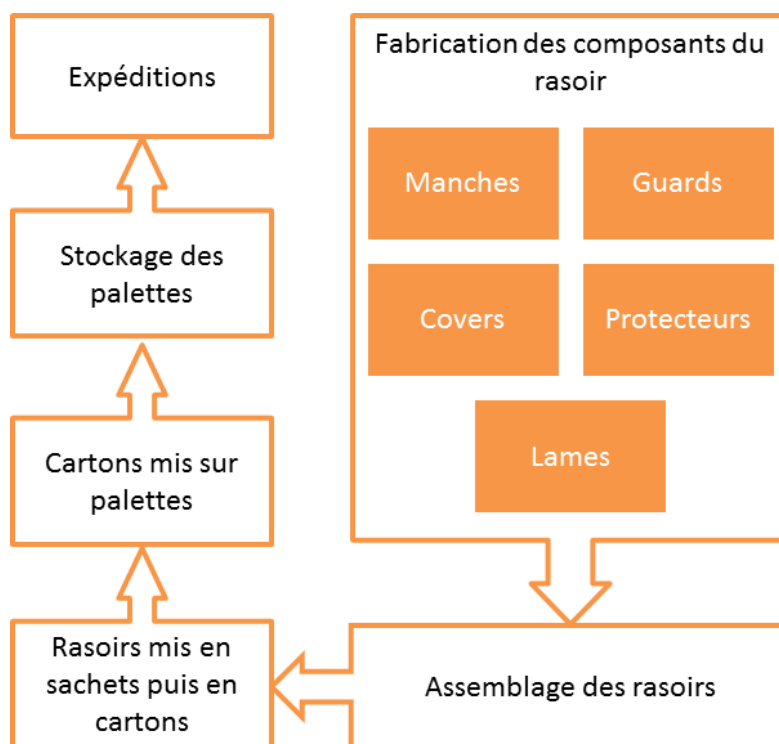
## 2. BIC Rasoirs, l'usine de production

Mon stage s'est effectué dans l'entreprise **BIC Rasoirs**, une SASU (Société par Actions Simplifiée Unipersonnelle) filiale du groupe BIC. Cette filiale est située au 6 Rue du Port Salut, 60126 Longueil-Sainte-Marie.

D'abord, utilisée en 1973 pour la conception de pièces de briquet et de composant d'articles d'écriture, elle est aujourd'hui uniquement consacrée à la production de rasoirs depuis l'année 1975. En outre, l'usine couvre une superficie de 20 000 m<sup>2</sup> et comprend 4 tranches d'activités. Employant environ 200 employés, elle fonctionne 5 jours sur 7 avec 3 équipes de 8 heures.

L'usine est divisée en 4 secteurs d'activité :

1. Une activité **moulage** : première étape de la création d'un rasoir, elle consiste à transformer du polystyrène granulé en composant des rasoirs au moyen de presse à injecter. Ce service crée donc toutes les pièces moulées nécessaires aux rasoirs comme le manche ;
2. Une activité **lames** : deuxième étape de la création d'un rasoir, elle permet de créer les lames indispensables pour un rasoir à partir d'une bobine de feuillard en acier. Celle-ci, avant d'atteindre le statut de lame, est perforée avec des outils de découpe montés sur des presses. On réalise ensuite un traitement thermique à la bobine, appelé trempe, pour rigidifier les lames et en améliorer les propriétés mécaniques. Le feuillard circule ensuite entre des meules afin d'affûter l'acier et réaliser le fil de la lame. Quand les lames atteignent le poste de polissage, un couteau les sépare contre une enclume. Elles seront alors contrôlées manuellement et en ligne ;
3. Une activité **assemblage** : troisième et dernière étape de la création d'un rasoir, elle consiste à assembler les différentes pièces créées précédemment pour assembler le rasoir final. Les différents composants sont chargés dans des systèmes de distribution qui cadencent leur arrivée dans la machine d'assemblage. Les composants plastiques sont disposés dans des trémies de grandes capacités (alimentées au moyen du chariot élévateur) alors que les lames sont distribuées. La machine permet alors l'assemblage des composants entre eux et de façon suivante : la tête du rasoir est assemblée indépendamment puis soudée sur le manche.  
Une fois assemblés, les rasoirs sont emballés dans un sachet (appelé « pouch ») puis disposés dans un carton. Le carton est pesé afin de s'assurer que le bon nombre de rasoirs sont présents ;
4. Une partie **expéditions** et **entreposage** des produits finis : enfin les rasoirs sont ensuite stockés dans l'entrepôt avant d'être expédiés vers différentes destinations.



Cycle de production des rasoirs

Cette usine est le cœur de la production du rasoir BIC et de sa technologie en France. Les rasoirs sont exportés dans les pays de l'Union Européenne, Afrique, Moyen-Orient, Amérique du Nord et Centrale et certains pays d'Amérique du Sud et Latine. Le principal client de BIC Rasoirs est BIC S.A.

BIC Rasoirs a réalisé un chiffre d'affaires de 32 943 300,00 € sur l'année 2018.

Bic Rasoirs est la seule usine en France à fabriquer les rasoirs de la marque. Il existe 2 autres usines à l'étranger les produisant : une à Athènes en Grèce, et une à Manaus au Brésil. Elles communiquent entre elles et s'échangent leur production pour vérifier l'état et la conformité de la production (le visuel, le dimensionnel, le packaging) puisque les 3 usines ne produisent pas les mêmes types de rasoirs.

Pour finir, l'usine produit principalement 2 types de rasoirs : le **BIC 1** (630 M/an) et le **BIC Comfort Twin** (80 M/an). Ce sont des rasoirs jetables produits grâce aux différents secteurs d'activités présentées précédemment. 3 millions de rasoirs et 6 millions de lames sont produits chaque jour et ce sont 11 millions de rasoirs qui sont vendus chaque jour dans le monde, soit près de 7700 chaque minute.

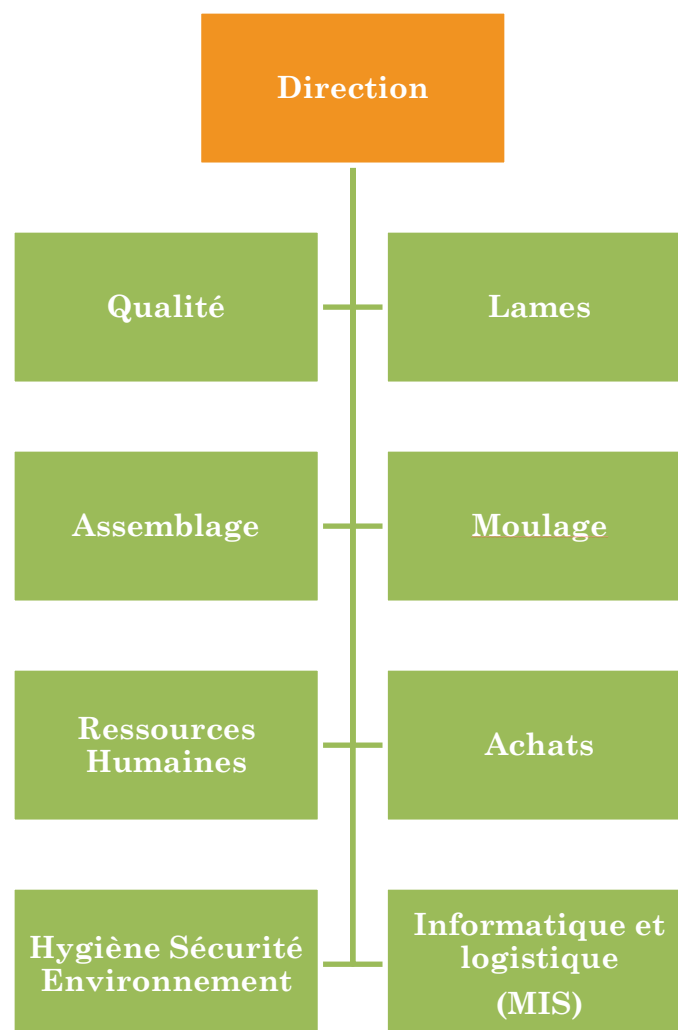
Comme toute entreprise, BIC Rasoirs possède de nombreux concurrents. Ses principaux concurrents sur le marché sont la marque américaine de rasoirs Gillette et la marque anglaise Wilkinson. En ce qui concerne la recherche et le développement, BIC Rasoirs ne conçoit pas de nouveaux produits. Cette activité est centralisée à BIC Violex à Athènes.

Par ailleurs, l'usine dispose de plusieurs certifications ISO de normalisation internationale et peut garantir à ses produits et services un label de référence reconnu. Par conséquent, grâce à la norme ISO, les produits de l'usine peuvent facilement se positionner sur le marché intérieur et international. Ces certifications sont les suivantes :

- ISO 9001/2000 : elle spécifie les exigences du système de management de la qualité où un organisme doit démontrer sa capacité à fournir systématiquement un produit qui répond aux exigences des clients et des réglementations ;
- ISO 14 001 : elle concerne le management environnemental et repose sur le principe d'amélioration continue de la performance environnementale par la maîtrise des impacts liés à l'activité de l'entreprise ;
- ISO 45001 : c'est une norme définissant des exigences précises concernant la sécurité et la santé au travail.

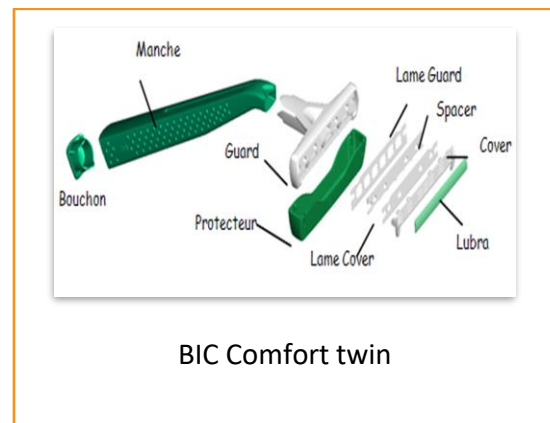
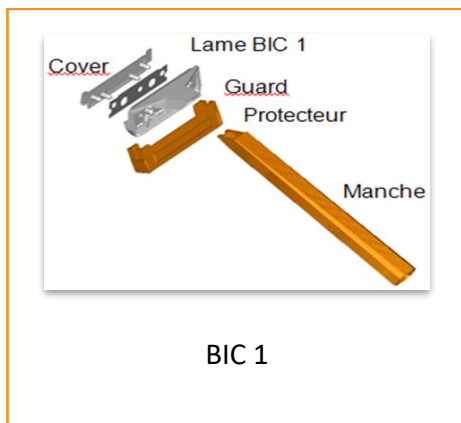
### 3. Organisation de BIC Rasoirs

L'organigramme suivant montre la composition de l'entreprise avec ses différents services, dont le service informatique dans lequel je me trouvais durant le stage :



Organigramme BIC Rasoirs



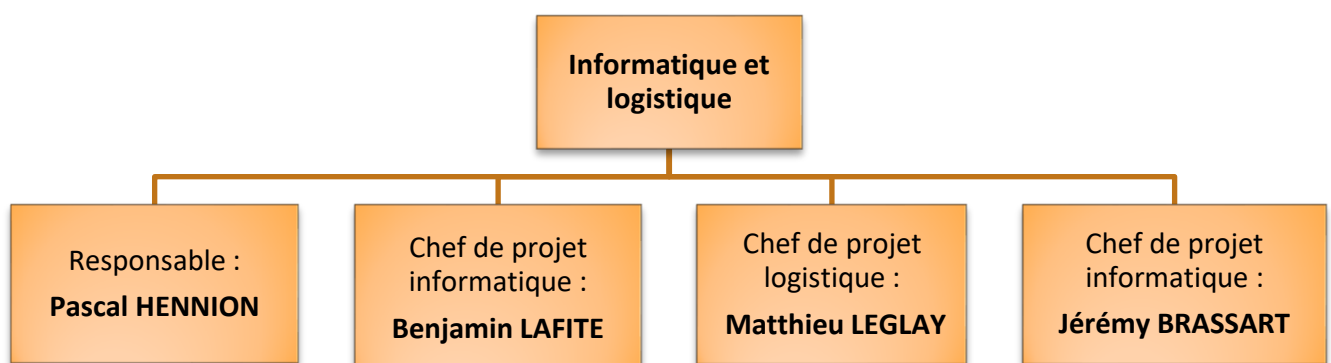


Vue éclatée des rasoirs produits

#### 4. Le MIS, le service informatique

Le service informatique ou MIS pour **Management Information System** assure la gestion et le fonctionnement du réseau sur le site ainsi que l'administration des bases de données. Il a la charge du développement de nouveaux outils utiles pour l'usine. Ces outils permettent des enregistrements, des consultations sur les bases de données. Il doit aussi assurer la sécurisation et l'intégrité des données pour l'ensemble des logiciels utilisés.

Voici un organigramme présentant les différentes personnes travaillant dans le service ainsi que leurs rôles :



Organigramme du MIS de BIC Rasoirs

Le MIS est là aussi pour former les employés et leur permettre un accès aux données dans les meilleures conditions possibles. Il est en communication avec l'ensemble des services de l'usine. De plus, il contribue à la réduction des impressions papier, action qui va dans le sens de la norme ISO 14 001 pour l'environnement. Le service n'a recours à aucun prestataire extérieur, la majeure partie se fait en interne.

Pour finir, les missions du MIS s'orientent autour de 3 domaines majeurs : le réseau et système, le développement et les projets.

#### Réseau et système :

- Installation et gestion du parc informatique (environ 200 postes)
- Configuration et installation d'équipements réseau (switch et bornes)
- Administration des serveurs, essentiellement Windows server 2012 à 2019 (environ 20 serveurs)
  - Serveurs Web, Applicatifs, Impression, Active Directory, Fichiers, GMAO, SQL, ...
  - Serveurs virtualisés sous VMware
- Administration des bases de données
  - Création de base de données
  - Sauvegarde
  - Mise à jour
- Administration de la téléphonie

#### Développement :

- Création d'applicatif web pour l'ensemble des services (outils de développement présentés dans ma partie environnement de travail)
  - AP2M, outils de supervision des machines de production pour l'assemblage
  - SYCODOM, système de collecte de données moulage
  - SS2L, systèmes des lots lames
- Formation aux utilisateurs

#### Projets :

- Etablissement de note de cadrage
- Démarchage de prestataire
- Chef de projet MOE/MOA
- Conduite du changement

## I. MES MISSIONS

### 1. Liste des missions

- Amélioration de la liste incendie
- Configuration et installation de pc Windows 10
- Amélioration de la saisie de packing list
- Ajout d'une fonctionnalité dans AP2M

### 2. Environnement de travail

Lors de mon stage, j'ai reçu un ordinateur portable HP pour pouvoir travailler sur le projet et utiliser les outils du MIS (base de données, programme, ...). En outre pour la mise en œuvre du projet, j'ai utilisé Notepad++ et SQL Server 2012.

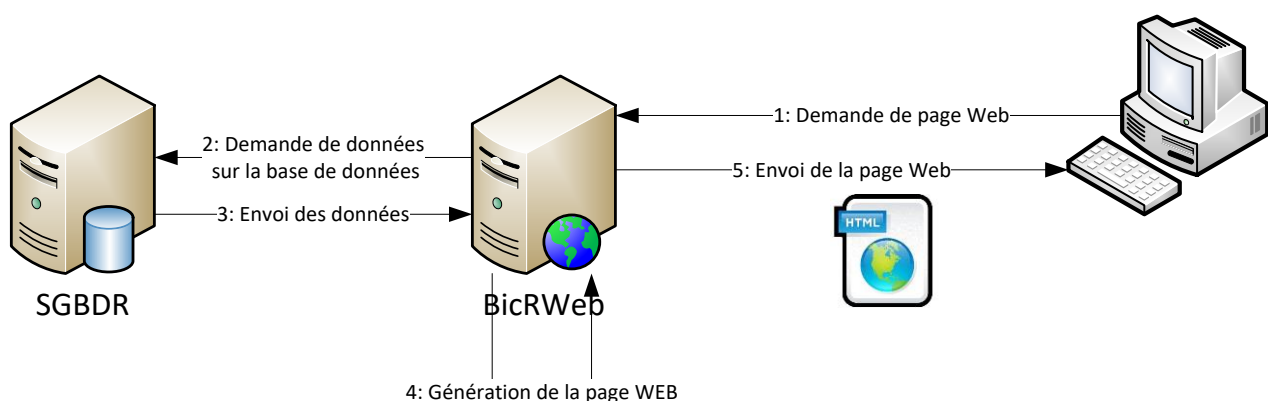
Notepad++ est un éditeur de texte libre générique gratuit qui intègre la coloration syntaxique de code source pour les langages et fichiers comme XML, SQL, PHP, ...

BIC Rasoirs, utilise le SGBDR SQL Server qui apporte une meilleure gestion de la sécurité, de la protection des données, de la traçabilité et de la performance que les autres SGBDR sur le marché. Ce SGBDR permet de gérer les informations collectées à travers les applicatifs web au sein de l'usine et de les stocker dans une base de données. Le choix d'un autre serveur SGBD (tel qu'Oracle) aurait été surdimensionné pour les besoins internes car au sein du site, il y a très peu de requêtes par jour (environ 7000). De plus, le choix d'utiliser le SGBD SQL Server est une décision prise par le siège.

J'ai reçu une copie de chacune des bases de données du MIS nécessaire à mon projet.

Enfin dans mes deux dernières missions, j'ai utilisé la bibliothèque PHP **BRISE** (Bic Rasoirs Interrogation Saisie et Enregistrement) qui est une bibliothèque PHP pour BicRWeb, dédiée à la définition de formulaire.

Pour conclure, tous mes projets sont accessibles à l'usine sur le serveur Web BicRWeb. Voici son fonctionnement en schéma :



## II. DESCRIPTION DE LA MISSION « AMELIORATION DE LA LISTE INCENDIE »

### 1. Genèse du projet

En cas d'incendie dans l'usine, les salariés doivent se rendre à l'extérieur (vers le parking) au point de rassemblement. Celui-ci se compose de 3 points qui répartissent les employés en groupes d'appels en fonction de la première lettre de leur nom de famille. Le premier point accueille les employés de A à D, le second de E à M et le dernier de N à Z. Lors de ce rassemblement, certains employés occupent une fonction spécifique pour assurer le bon déroulement de cette évacuation. L'une d'elles consiste à utiliser une liste d'appel incendie pour faire l'appel de tous les employés présents sur site. Cet appel est effectué sur chaque point de rassemblement par une personne désignée et permet alors de vérifier si aucun employé n'est resté dans l'usine. Enfin elle est remise aux sapeurs-pompiers dès leur arrivée sur les lieux.

La liste incendie est un document légal et obligatoire qui doit être accessible à tout moment dans l'usine en cas d'incendie. Cette liste ne contient pas tous les employés de l'usine, mais elle est générée à partir du premier pointage des employés arrivés à l'usine. En outre la génération de la liste incendie dynamiquement par le pointage des employés permet un gain de temps considérable lors de l'appel car l'usine compte environ 200 employés ayant des horaires de journées, de soirées et de nuits. Ainsi lors de l'appel seul les employés vraiment présents dans l'usine sont appelés.

Avant mon intervention, la liste incendie se composait initialement de 3 listes de groupes d'appels (A à D, E à M et N à Z) contenant le numéro de la ligne, le nom, le prénom, la date de d'arrivé de l'employé et l'heure de pointage de l'employé et une dernière colonne permettant d'indiquer la présence de l'employé.

32	DIOT	Clément	2022-01-21 07:58	
----	------	---------	------------------	--

Exemple de ligne de la liste incendie

Ces 3 groupes d'appels contenaient chacun une liste d'ajout manuel dans le cas où un employé n'aurait pas pointé. Ne pas pointer signifiant sa non-présence lors de la génération de la liste. Pour finir, ses listes étaient dupliquées pour les sapeurs-pompiers et une légende des rôles incendies des employés était présente sur chaque page.

Cette première ébauche a été effectuée l'année dernière par un stagiaire (annexe 1) pour le compte de Mme PIERRA, chef de projet HSE (Hygiène Sécurité Environnement) à l'usine, avec l'aide de M. BRASSART. Cependant lors d'une réunion Teams avec Mme Pierra, moi-même et Mr BRASSART. Mme PIERRA nous a transmis de nouvelles améliorations concernant le projet de la liste incendie de l'usine. Avec l'aide de mon tuteur Mr BRASSART nous avons défini les besoins de Mme PIERRA qui sont les suivants :

- Suppression des duplicatas des listes à destination des sapeurs-pompiers. Dorénavant les listes seront imprimées en double directement pour les pompiers
- Utiliser une couleur de police propre à chaque groupe d'appel pour une meilleure performance visuelle
- Ajout du texte « ESI : Equipier de Seconde Intervention » dans la légende
- Harmonisation des prénoms en majuscules ainsi que leurs positions et ceux des noms de familles à gauche de leurs cellules dans la liste



De plus, M. BRASSART m'a confié la mission d'ajouter des commentaires au code ainsi que d'apporter une architecture MVC au code. Le projet achevé permettrait une utilisation beaucoup plus optimale de la liste et une meilleure compréhension du code pour de futures améliorations.

## 2. Planning

Pour mener à bien mon projet, un planning a été défini :

Tâches :	Semaines :		
	1	2	3
Installation des logiciels et appropriation du code			
Modification et amélioration de la liste			
Présentation, formation et mise en production			
Rédaction du rapport de stage			

## 3. Développement des améliorations sur la liste incendie

### a. Fonctionnement de la liste incendie

Horoquartz est un éditeur de logiciels qui aide les entreprises à construire une relation entreprise-salariée saine et durable. BIC a recours à cette société pour la gestion des temps, des plannings et des activités de ses employés. Pour ce faire, Horoquartz propose une suite logicielle eTemptation qui d'appuie sur une base de données nommée eTemptation2. Cette dernière collecte plusieurs informations via le logiciel, y compris des données sur le pointage des employés qui nous intéressent ici.

Les données de pointage sont récupérées lorsqu'un employé se présente avec son badge sur la pointeuse pour la première fois de la journée et sont stockées dans la base de données eTemptation2. Les pointeuses, installées à l'entrée de chaque zone d'activité de l'usine, sont destinées à enregistrer les temps de présence et à décompter automatiquement les heures travaillées.



Pointeuse eTSmart  
d'Horoquartz

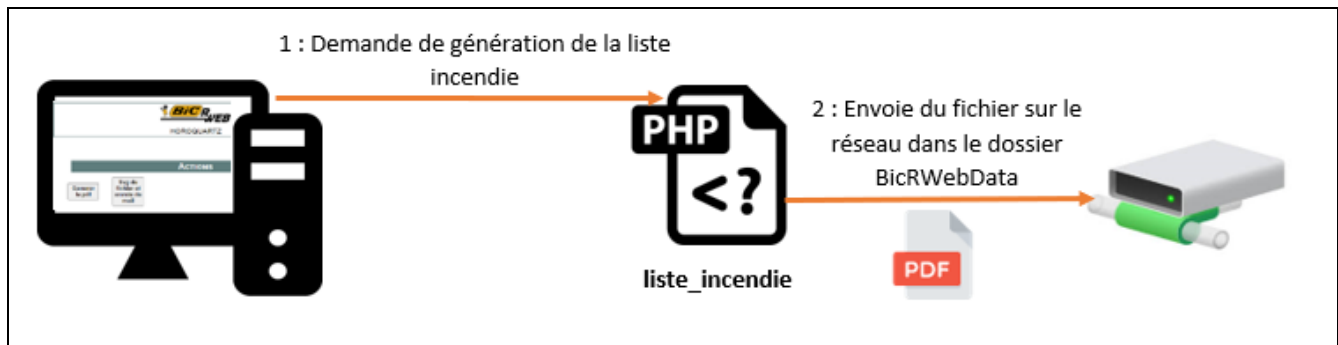


Exemple de badge  
d'employée

Un programme PHP nommé liste\_incendie.php récupère le nom, le prénom, la date et l'heure d'arrivée des employés depuis la base de données eTemptation2 et les place dans une liste de type PDF. Ces données apparaissent lors du premier pointage et disparaissent lors du deuxième pointage du salarié signifiant la fin de sa journée dans l'usine.

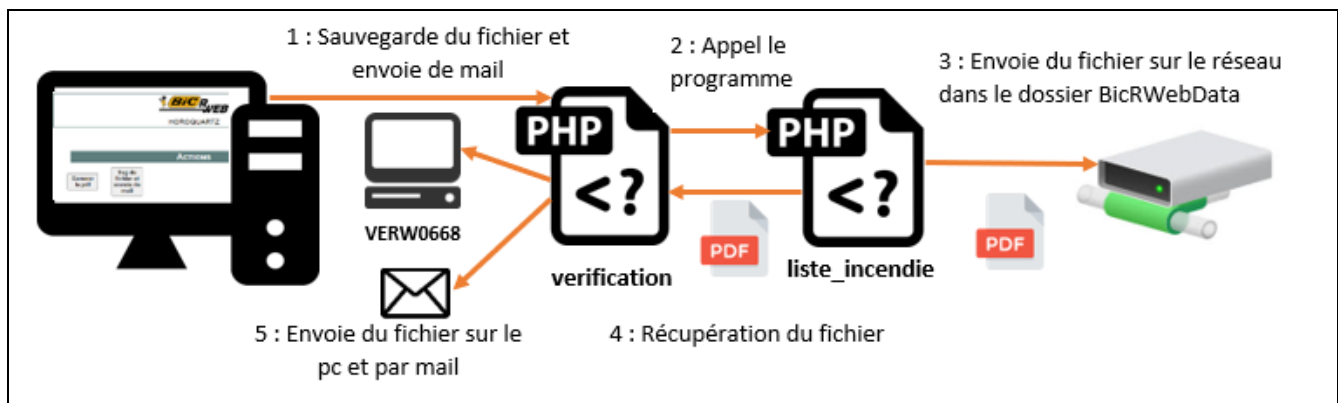
Maintenant nous allons voir la génération et l'accès à la liste incendie initiale, qui est divisée en plusieurs étapes utilisant différents fichiers. Il existe trois manières principales de générer la liste incendie.

- La première consiste à se rendre sur BicRWeb (interface web de l'usine) et lancer sa génération en cliquant sur le bouton « Générer le pdf » :



L'utilisateur voulant générer la liste incendie doit se rendre sur BicRWeb puis cliquer sur le bouton « Générer le pdf » dans l'onglet Horoquartz. Ce bouton lancera alors le programme liste\_incendie.php qui génère la liste au format pdf et qui l'envoie sur l'un des réseaux de l'usine dans le dossier BicRWebData. Enfin, l'utilisateur accède au réseau à l'aide d'un lecteur réseau associé au réseau contenant le dossier BicRWebData dans l'explorateur de fichiers, afin que la liste incendie puisse être récupérée ou consultée.

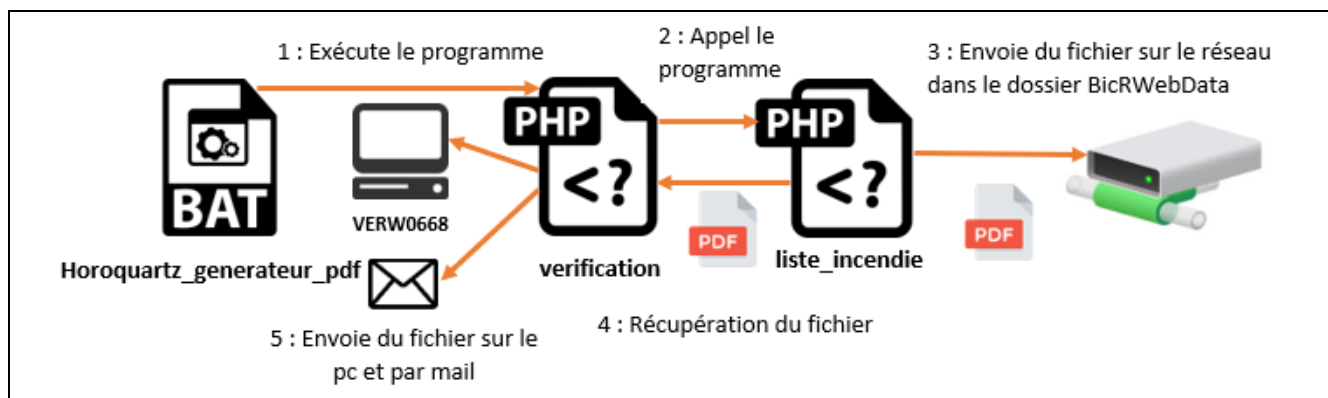
- La seconde consiste également à se rendre sur BicRWeb et sélectionner le deuxième bouton mis à disposition dans l'onglet Horoquartz, le bouton « Svg du fichier et envoi de mail » :



Un clic sur ce bouton par l'utilisateur exécutera le programme verification.php (annexe 2). Ce programme appelle le programme liste\_incendie.php, récupère la liste incendie et l'envoie sur l'ordinateur VERW0668. Cet ordinateur est situé à l'extérieur de l'usine, réduisant la possibilité qu'il soit touché lors d'un incendie dans l'usine. Il l'envoie également par courriel à une adresse spécialement conçue pour la réception des listes incendies, auxquelles seules certaines personnes ont accès. Si le pc VERW0668 prend également feu, ce mail assure la redondance afin qu'il soit possible de récupérer la liste sur un téléphone à l'aide d'Outlook.



- La troisième s'effectue automatiquement grâce à un fichier .bat :



Un fichier bat est l'extension des fichiers de commande MS-DOS. La création d'un tel fichier permet de concevoir des scripts qui seront interprétés par un "Shell" ou un interpréteur de commandes (command.com ou cmd.exe) pour exécuter des fichiers .EXE ou .COM. Ici, le script du fichier Horoquartz\_générateur\_pdf (annexe 3) exécute le fichier verification.php, en effectuant les mêmes étapes ci-dessus toutes les 900 secondes.

#### b. Résultat final

Voici l'apparence finale de la liste incendie de l'usiné généré le 21-01-2022 à 08H10 :


BIC		Liste incendie		Page 1/8
				Généré le : 2022-01-21 08:10
Liste d'appel A-D				
N°	Nom	Prénom	Date et heure de pointage	Présent
1	(ESI - S)			
2				
3	(EI)			
4	(EI - S)			
5	(S)			
6				
7	(SF)			
8				
9	(EI)			
10				
11	(ESI)			
12				
13	(ESI)			
14	(EI - S)			
15	(G)			
16				
17				
18				
19				
20				
21	(ESI - S)			
22				
23	(G)			
24				
25	(G)			

**Rôle incendie :**  
 >Guide(G)/Serre file(SF) : voir autres points rassemblement  
 >Equipes d'intervention(EI) : vérifier avec le gardien  
 >Equipier de Seconde Intervention(ESI)  
 >Secouristes(S)

BIC		Liste incendie		Page 2/8
				Généré le : 2022-01-21 08:10
26				
27	(ESI -S)			
28				
29	DELHALLE	MATHIS	2022-01-21 07:58	
30				
31	(SF)			
32	DIOT	CLÉMENT	2022-01-21 07:58	
33	(SF)			
34				
35				
36				
37				
38	(SF)			


**Rôle incendie :**  
 >Guide(G)/Serre file(SF) : voir autres points rassemblement  
 >Equipes d'intervention(EI) : vérifier avec le gardien  
 >Equipier de Seconde Intervention(ESI)  
 >Secouristes(S)



 **Liste incendie** Page 3/8  
Généré le : 2022-01-21 08:10


Ajout Manuel de Liste d'appel A-D			
N°	Nom	Prénom	Matif (soubli, pointage,...)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

**Notes incendie :**  
 > Guide (G) / Guide file (GF) : voir autres points d'assemblage  
 > Équipe d'intervention (E) : vérifier avec le gardien  
 > Équipe des Secondes Interventions (ESI)  
 > Réserve (R)

 **Liste incendie** Page 4/8  
Généré le : 2022-01-21 08:10


Liste d'appel E-M				
N°	Nom	Prénom	Date et heure de pointage	Présent
1	(SF)			
2				
3				
4				
5	(G - S)			
6				
7				
8				
9	(SF)			
10				
11	(G)			
12	(G - S)			
13	(SF - S)			
14				
15	(ESI)			
16				
17				
18	(G)			
19	(RI)			
20				
21				
22				
23	(SF)			
24				
25	(SF)			

**Notes incendie :**  
 > Guide (G) / Guide file (GF) : voir autres points d'assemblage  
 > Équipe d'intervention (E) : vérifier avec le gardien  
 > Équipe des Secondes Interventions (ESI)  
 > Réserve (R)

 **Liste incendie** Page 5/8  
Généré le : 2022-01-21 08:10


26			2022-01-21 07:36	
27	(N)		2022-01-21 07:34	
28			2022-01-21 07:35	
29	(SF)		2022-01-21 08:47	
30	(G - S)		2022-01-21 08:46	

**Notes incendie :**  
 > Guide (G) / Guide file (GF) : voir autres points d'assemblage  
 > Équipe d'intervention (E) : vérifier avec le gardien  
 > Équipe des Secondes Interventions (ESI)  
 > Réserve (R)

 **Liste incendie** Page 6/8  
Généré le : 2022-01-21 08:10


Ajout Manuel de Liste d'appel E-M			
N°	Nom	Prénom	Matif (soubli, pointage,...)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

**Notes incendie :**  
 > Guide (G) / Guide file (GF) : voir autres points d'assemblage  
 > Équipe d'intervention (E) : vérifier avec le gardien  
 > Équipe des Secondes Interventions (ESI)  
 > Réserve (R)

 **Liste incendie** Page 7/8  
Généré le : 2022-01-21 08:10

Liste d'appel N-Z				
N°	Nom	Prénom	Date et heure de pointage	Présent
1				
2				
3				
4				
5	(G - S)			
6	(RI)			
7	(S)			
8				
9				
10	(SF)			
11				
12				
13				
14	(R)			
15				
16	(R - S)			

**Notes incendie :**  
 > Guide (G) / Guide file (GF) : voir autres points d'assemblage  
 > Équipe d'intervention (E) : vérifier avec le gardien  
 > Équipe des Secondes Interventions (ESI)  
 > Réserve (R)

 **Liste incendie** Page 8/8  
Généré le : 2022-01-21 08:10

Ajout Manuel de Liste d'appel N-Z			
N°	Nom	Prénom	Matif (soubli, pointage,...)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

**Notes incendie :**  
 > Guide (G) / Guide file (GF) : voir autres points d'assemblage  
 > Équipe d'intervention (E) : vérifier avec le gardien  
 > Équipe des Secondes Interventions (ESI)  
 > Réserve (R)

Suite à mon intervention, la liste incendie répond à tous les besoins exprimés par Mme PIERRA car :

- Elle ne contient plus les duplicatas des groupes d'appels ;
- Chaque groupe d'appel possède une couleur qui lui est propre (groupe d'appels A-D->bleu, E-M->vert et N-Z->orange) ;
- L'ajout de la définition d'un ESI dans la légende du pied de page est effectué ;
- Les noms et prénoms sont tous en majuscules et mis à gauche de leurs cellules.

Durant une réunion organisée par mes soins avec M. BRASSART et Mme PIERRA, cette liste a été validée marquant la fin du projet et la mise en production de cette nouvelle liste incendie dans l'usine.

Pour aboutir à ce résultat, je suis intervenu uniquement sur le programme en PHP liste\_incendie et reçu la base de données eTemptation2. Afin de mettre en place une architecture MVC, j'ai séparé en trois parties le code dans des fichiers différents. Le premier fichier de type PHP nommé **PDF** contient la partie objet (POO) du code donc notre classe PDF avec son attribut privé, son constructeur et ses méthodes. Dans un modèle MVC, ce fichier serait alors le fichier entity de la partie Model. Ensuite, le deuxième fichier de type XML nommée **ConnexionBDD** contient la partie SQL du code. Dans un modèle MVC, ce fichier serait la combinaison entre le fichier config et le fichier repository de la partie Model. Pour finir, le troisième fichier de type PHP nommée **liste\_incendie** contient le programme principal servant principalement à se connecter à la base de données, instancier un objet de la classe PDF et utiliser les méthodes pour mettre en forme cet objet sous forme de la liste incendie avec les données suivantes : nom, prénom, date et heure de pointage. Dans un modèle MVC, ce fichier serait la fusion entre le fichier Controller et le fichier View.

A partir de maintenant, je vais vous expliquer le fonctionnement et l'utilisation de ses différents fichiers. En réponse à un autre besoin, qui est de fournir des commentaires pour le code, tout le code des fichiers est commenté afin qu'il puisse être compris par des personnes extérieures au programme et que ses futures évolutions soient plus simples à effectuer. Ainsi, il est possible pour une personne ayant des bases en programmation de comprendre le fonctionnement du programme grâce aux commentaires.

### c. Présentation du fichier PDF.php

```
<?php

//appel de la classe FPDF
require('../lib-php/fpdf/fpdf.php');

//la classe PDF hérite de la classe FPDF
class PDF extends FPDF
{
    //attribut privé de la classe
    private $jour;           //le jour courant

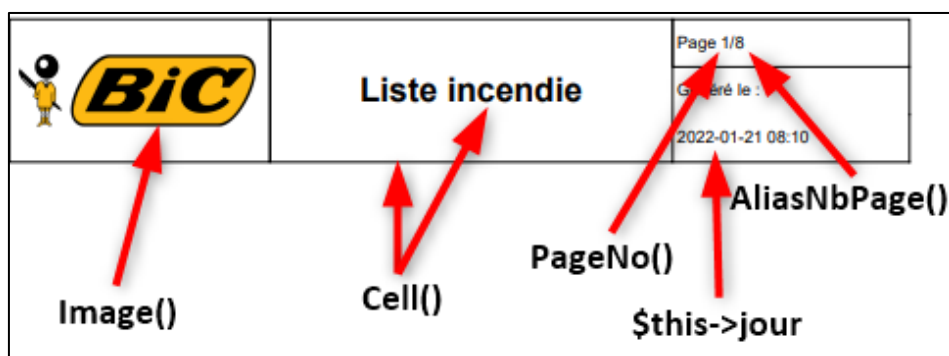
    //constructeur de la classe PDF,
    //parent::__construct(arguments) permet d'appeler le constructeur parent FPDF,
    //ce qui change le constructeur de la classe qui accepte les arguments de la classe FPDF ($orientation, $unit, $size)
    public function __construct($orientation, $unit, $size, $jour) {
        parent::__construct($orientation, $unit, $size);
        $this->jour = $jour;
    }

    //déclaration de la méthode Header() pour traiter l'en-tête de page,
    //l'implémentation de Header() dans FPDF est vide,
    //donc il faut redéfinir la méthode pour avoir un traitement particulier de l'en-tête
    public function Header()
    {
        //déclaration de la méthode Footer() permet de définir le pied de page,
        //l'implémentation de Footer() dans FPDF est vide,
        //donc il faut redéfinir la méthode pour avoir un traitement particulier du pied de page
        public function Footer()
        {
            //déclaration de la méthode mise_en_page(PDF $pdf, ODBC $success, string $en_tete, int $num_tableau_associe_entete)
            //qui retourne l'objet PDF avec la mise en page de la liste
            //Paramètres : $pdf = objet de la classe PDF
            //                  $success = contient la commande SQL obdc_exec()
            //                  $en_tete = l'en tête de la liste
            //                  $num_tableau_associe_entete = le numéro associé à l'en-tête
            function mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete)
            {
            }
        }
    }
}
```

J'ai créé une classe PDF qui hérite de la classe FPDF qui permet de générer des fichiers. Le F de FPDF signifie Free donc nous sommes libres de l'utiliser et de la modifier comme on le souhaite. Cette classe se télécharge gratuitement sur ce site : [www.fpdf.org](http://www.fpdf.org) qui m'a permis de comprendre le fonctionnement de cette classe et l'utilisation de ses méthodes. La classe PDF redéfinit les méthodes Header() et Footer() pour traiter les en-têtes et pieds de page car leur implémentation dans la classe FPDF n'existent pas. Donc pour les utiliser nous devons les redéfinir.

La classe dérivée PDF me permet aussi de déclarer un nouvel attribut privé qui me servira à récupérer le jour courant. Son constructeur public valorise cet attribut et appelle le constructeur parent de la classe FPDF.

La méthode Header() configure l'en-tête du fichier de la liste incendie et est appelée automatiquement donc il n'est pas nécessaire de l'appeler dans le programme principal. Cette méthode génère l'en-tête dans le fichier de la liste incendie avec cette apparence :



Et voici la méthode Header() qui nous permet d'obtenir ce résultat :

```
//déclaration de la méthode Header() pour traiter l'en-tête de page,
//l'implémentation de Header() dans FPDF est vide,
//donc il faut redéfinir la méthode pour avoir un traitement particulier de l'en-tête
public function Header()
{
    //la méthode Image(string file, float x, float y, float w) imprime le logo BIC
    //Paramètres : LogoBic.png = chemin du logo,
    //              12 = abscisse du coin supérieur gauche,
    //              17.5 = ordonnée du coin supérieur gauche,
    //              50 = ordonnée du coin supérieur gauche
    $this->Image('LogoBic.png',12,17.5,50);

    //la méthode Cell(float w, float h, string txt, mixed border, int ln, string align) imprime une cellule (zone rectangulaire) autour du logo
    //Paramètres : 55 = largeur de la cellule,
    //              30 = hauteur de la cellule,
    //              '' = aucune chaîne à imprimer
    //              1 = indique que la cellule est encadré
    //              0 = indique la position courante à droite après l'appel de la méthode
    //              C = centre le logo dans la cellule
    $this->Cell(55,30,'',1,0,'C');

    //la méthode SetFont(string family, string style, float size) fixe la police utilisée pour imprimer le titre
    //Paramètres : Arial = famille de la police
    //              B = la police en gras
    //              20 = taille de police
    $this->SetFont('Arial','B',20);

    //on affiche le titre dans une cellule au centre de l'en-tête
    $this->Cell(85,30,'Liste incendie',1,0,'C');

    //la police Arial en taille 10 pour l'affichage du N° Page
    $this->SetFont('Arial','',10);

    //la méthode PageNo() renvoie le numéro de page courant,
    //nb : définit un alias pour le nombre total de page
    //on affiche le N° de page courant et maximum en haut à droite
    $this->Cell(50,10,'Page '.$this->PageNo().'/{nb}',1,1,'L');

    //on crée une cellule pour afficher la date de génération,
    //il ne faut pas utiliser l'encodage UTF-8 avec les polices standards,
    //elles attendent de l'ISO-8859-1 ou windows-1252.
    //Donc on peut effectuer une conversion en ISO-8859-1 grâce à utf8_decode()
    //et utiliser des caractères spéciaux, accents,...
    $this->Cell(140,10,'',0,0,'C');
    $this->SetFont('Arial','',10);
    $this->Cell(50,10,utf8_decode("Généré le :\n"),'R',1,'L');
    $this->Cell(140,10,'',0,0,'C');
    $this->Cell(50,10,''.$this->jour,'BR',1,'L');

    //la méthode Ln(float h) effectue un saut de ligne
    //Paramètre : 10 = amplitude du saut de ligne
    $this->Ln(10);
}
```

Cette méthode publique n'a besoin d'aucun paramètre et utilise différentes méthodes héritées de la classe FPDF pour former le contenu de l'en-tête. L'en-tête de la liste incendie permet une reconnaissance immédiate de la nature du document. Le logo désigne la société, le titre indique la nature

du document et on informe également sur le nombre de pages, la date et l'heure de la génération de la liste. Ces deux dernières informations sont importantes pour l'évacuation car elle indique si la personne a bien récupéré toutes les pages de la liste incendie en cas d'appel et si elle a été mise à jour récemment.

Pour représenter les cellules et la disposition des titres et logo, j'ai utilisé la méthode Cell(). Pour afficher le logo BIC, j'utilise la méthode Image() qui va chercher l'image du logo sur le dossier où se situe le programme. Pour afficher le numéro de la page courante, j'utilise la méthode PageNo() et pour afficher le nombre total de pages j'utilise la méthode AliasNbPages() qui sera exprimée dans le programme principal. Enfin, « \$this->jour » permet d'utiliser l'attribut privé « \$jour » valorisé par le constructeur, sous la forme ('Y-m-d H:i'), indiquée lors de l'instanciation de l'objet PDF dans le programme principal. Bien évidemment, les commentaires visibles dans les captures écran indiquent les paramètres obligatoires et le fonctionnement des différentes méthodes.

La méthode Footer() configure le pied de page du fichier de la liste incendie et est appelée automatiquement (il n'est donc pas nécessaire de l'appeler dans le programme principal). Cette méthode génère le pied de page dans le fichier de la liste incendie avec cette apparence :

Rôle incendie :

- >Guide(G)/Serre file(SF) : voir autres points rassemblement
- >Equipes d'intervention(EI) : vérifier avec le gardien
- >Equipier de Seconde Intervention(ESI)
- >Secouristes(S)

Voici la méthode Footer() qui nous permet d'obtenir ce résultat :

```
//déclaration de la méthode Footer() permet de définir le pied de page,
//l'implémentation de Footer() dans FPDF est vide,
//donc il faut redéfinir la méthode pour avoir un traitement particulier du pied de page
public function Footer()
{
    //la méthode SetY(float y) fixe l'ordonnée, la où commence le pied de page
    //Paramètre : y = valeur de l'ordonnée
    $this->SetY(-25);

    //on imprime la légende des différents rôles lors d'un incendie
    $this->SetFont('Arial','U',10);
    $this->Cell(0,5,utf8_decode('Rôle incendie :'),0,0,'L');
    $this->Ln(1);

    $this->SetFont('Arial','',10);
    $this->Cell(0,13,'>Guide(G)/Serre file(SF) : voir autres points rassemblement',0,0,'L');
    $this->Ln(1);

    $this->Cell(0,20,utf8_decode('>Equipes d\'intervention(EI) : vérifier avec le gardien'),0,0,'L');
    $this->Ln(1);

    $this->Cell(0,27,utf8_decode('>Equipier de Seconde Intervention(ESI)'),0,0,'L');
    $this->Ln(1);

    $this->Cell(0,34,'>Secouristes(S)',0,0,'L');
}
```

Cette méthode publique n'a besoin d'aucun paramètre et utilise différentes méthodes héritées de la classe FPDF pour former le contenu du pied de page. Le pied de page dans la liste incendie est une

légende permettant de connaître le rôle spécifique d'un employé durant un incendie, ce rôle est caractérisé par une lettre ou un sigle à côté de son nom de famille. Tout d'abord, la méthode SetY() fixe la position sur l'axe Y pour la position de la légende. Ensuite, la méthode SetFont() fixe la police du texte ainsi que sa taille et la méthode Cell() permet l'écriture des différents textes qui composent la légende. Enfin, la méthode Ln() fixe un saut de ligne.

Nous allons voir la méthode mise\_en\_page() qui configure la mise en page de l'instance de la classe PDF donc la mise en page de la liste incendie. En outre, elle est déclarée en dehors de la classe car elle utilise l'instance de la classe PDF pour utiliser les méthodes de FPDF pour sa mise en page. Cette méthode génère alors les 3 listes de groupe d'appel avec leurs listes d'ajout manuel dans le fichier de la liste incendie avec cette apparence :

Liste d'appel N-Z					Ajout Manuel de Liste d'appel N-Z			
N°	Nom	Prénom	Date et heure de pointage	Présent	N°	Nom	Prénom	Motif (oubli,pointage,...)
1					1			
2					2			

Voici la méthode mise\_en\_page () qui nous permet d'obtenir ce résultat :

3

```

//déclaration de la méthode mise_en_page(PDF $pdf, ODBC $success, string $en_tete, int $num_tableau_associe_entete)
//qui retourne l'objet PDF avec la mise en page de la liste
//Paramètres : $pdf = objet de la classe PDF
//              $success = contient la commande SQL obdc_exec()
//              $en_tete = l'en tête de la liste
//              $num_tableau_associe_entete = le numéro associé à l'en-tête
function mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete)
{
    #Création de la liste d'appel trier par première lettre du nom de famille#
    #Création de l'en-tête de la liste#
    //on associe la couleur de fond de l'en-tête de la liste à son numéro correspondant
    switch($num_tableau_associe_entete){
        case 1:
            //la méthode SetFillColor(int r, int g, int b) fixe la couleur pour toutes les opérations de remplissage,
            //Paramètres : r = composante de rouge
            //              g = composante de vert
            //              b = composante de bleu
            //couleur bleu
            $pdf->SetFillColor(0,100,255);
            break;
        case 2:
            //couleur verte
            $pdf->SetFillColor(14,107,14);
            break;
        case 3:
            //couleur orange
            $pdf->SetFillColor(255,100,0);
            break;
    }

    //la méthode Rect(float x, float y, float w, float h, string style) dessine un rectangle à partir de son coin supérieur gauche,
    //Paramètres : 10 = abscisse du coin supérieur gauche,
    //              50 = ordonnée du coin supérieur gauche,
    //              190 = la largeur,
    //              10 = la hauteur,
    //              F = le remplissage du rectangle
    $pdf->Rect(10,50,190,10,'F');
    //la méthode SetTextColor(int r, int g, int b) fixe la couleur pour le texte de la liste
    //Paramètres : r = composante de rouge
    //              g = composante de vert
    //              b = composante de bleu
    $pdf->SetTextColor(255,255,255);
    $pdf->SetFont('Times','B',14);
    $pdf->Cell(190,10,$en_tete,1,1,'C');
    $pdf->SetFont('Times','B',11);
    $pdf->SetTextColor(0,0,0);
}
    
```

**Première partie**



```
#Création de l'intitulé des colonnes#
//couleur jaune
$pdf->SetFillColor(255,220,0);
$pdf->Rect(10,60,190,10,'F');
$pdf->Cell(15,10,utf8_decode('N'),1,0,'C');
$pdf->Cell(50,10,'Nom',1,0,'C');
$pdf->Cell(50,10,utf8_decode('Prénom'),1,0,'C');
$pdf->Cell(45,10,'Date et heure de pointage',1,0,'C');
$pdf->Cell(30,10,utf8_decode('Présent'),1,1,'C');
$pdf->SetFillColor(255,255,255);

//on initialise une variable qui nous permettra de coloré une ligne sur 2
$nbactuel = 1;

//odbc_fetch_row(resource $statement) lit une ligne dans le résultat identifié par result_id et retourne true si la ligne existe, false sinon,
//Paramètre : $success = l'identifiant de résultat
//on effectue une condition qui s'arrêtera quand il n'y aura plus de ligne à lire
while(odbc_fetch_row($success)){

    //odbc_result(resource $statement, string $field) permet de lire un champ de résultat ODBC
    //Paramètres : $success = une ressource ODBC
    //                'Nom','Prénom',... = le nom du champ à récupérer
    //on récupère les différentes données de chaque ligne dans des variables qui seront ensuite affichées dans la liste ligne par ligne
    $Nom = ''.odbc_result($success,'Nom');
    $Prenom = ''.odbc_result($success,'Prénom');
    $Date = ''.odbc_result($success,'Date_Heure_Bipage2');
    $NumTableau = ''.odbc_result($success,'NumTableau');

    //on ajoute une couleur de fond 1 fois sur 2 suivant le numéro d'en_tête
    if($NumTableau == $num_tableau_associe_entete){
        if($nbactuel % 2 != 0){
            //couleur blanche
            $pdf->SetFillColor(255,255,255);
        }else{
            switch($num_tableau_associe_entete){
                case 1:
                    //couleur bleu
                    $pdf->SetFillColor(200,236,255);
                    break;
                case 2:
                    //couleur verte
                    $pdf->SetFillColor(189,236,182);
                    break;
                case 3:
                    //couleur orange
                    $pdf->SetFillColor(255,200,122);
                    break;
            }
        }

        //on renseigne les différentes données dans les colonnes de la liste
        $pdf->SetFont('Times','',13);
        $pdf->Cell(15,8,''. $nbactuel,1,0,'C',true);
        $pdf->SetFont('Times','B',12);
        $pdf->Cell(50,8,$Nom,1,0,'L',true);
        $pdf->SetFont('Times','',12);
        $pdf->Cell(50,8,$Prenom,1,0,'L',true);
        $pdf->Cell(45,8,''. $Date,1,0,'C',true);
        $pdf->Cell(30,8,''. $Présent,1,1,'C',true);
        $nbactuel++;
    }
}
```

## Deuxième partie

```
#Création de la liste d'ajout manuel de liste d'appel trier par première lettre du nom de famille#
$pdf->AddPage();
#Création de l'en-tête de la liste#
$pdf->SetFont('Times','B',12);
//on associe la couleur de fond de l'en-tête de la liste à son numéro correspondant
switch($num_tableau_associe_entete){
    case 1:
        $pdf->SetFillColor(0,100,255);
        break;
    case 2:
        $pdf->SetFillColor(14,107,14);
        break;
    case 3:
        $pdf->SetFillColor(255,100,0);
        break;
}
$pdf->Rect(10,50,190,10,'F');
$pdf->SetTextColor(255,255,255);
$pdf->SetFont('Times','B',14);
$pdf->Cell(190,10,'Ajout Manuel de '.$en_tete,1,1,'C');
$pdf->SetFont('Times','B',12);
$pdf->SetTextColor(0,0,0);

#Création de l'intitulé des colonnes#
$pdf->SetFillColor(255,220,0);
$pdf->Cell(15,10,utf8_decode('N°'),1,0,'C',true);
$pdf->Cell(50,10,'Nom',1,0,'C',true);
$pdf->Cell(50,10,utf8_decode('Prénom'),1,0,'C',true);
$pdf->Cell(75,10,'Motif (oubli,pointage,...)',1,1,'C',true);
$pdf->SetFillColor(255,255,255);
$pdf->SetFont('Times','',12);

//on effectue une boucle pour créer 15 lignes vide pour la liste d'ajout manuel d'employé,
//avec 1 fois sur deux une couleur de fond associée à son numéro d'en-tête
$nla=15;
for($absl=1; $absl<=$nla; $absl++){
    if($absl % 2 != 0){
        //couleur blanche
        $pdf->SetFillColor(255,255,255);
    }else{
        switch($num_tableau_associe_entete){
            case 1:
                //couleur bleu
                $pdf->SetFillColor(200,236,255);
                break;
            case 2:
                //couleur vert
                $pdf->SetFillColor(189,236,182);
                break;
            case 3:
                //couleur orange
                $pdf->SetFillColor(255,200,122);
                break;
        }
    }

    //on créer les lignes vide avec le numéro de ligne
    $pdf->Cell(15,8,'',$absl,1,0,'C',true);
    $pdf->Cell(50,8,'',$absl,1,0,'C',true);
    $pdf->Cell(50,8,'',$absl,1,0,'C',true);
    $pdf->Cell(75,8,'',$absl,1,1,'C',true);
}

return $pdf;
```



Pour fonctionner cette méthode a besoin de quatre variables : la première de type objet, la seconde de type ODBC, la troisième de type string et la dernière de type int. Dans notre cas lors de son appel dans le programme principal, il sera passé en paramètre « \$pdf » l'instance de la classe PDF, « \$success » contenant une commande SQL `odbc_exec()`, « \$en\_tete » correspondant à l'en-tête de la liste selon si c'est un groupe d'appel ou une liste d'ajouts manuels et enfin « num\_tableau\_associe\_entete » contenant un numéro qui est associé à cet en-tête.

La première partie du code permet la mise en page de la liste des groupes d'appels avec les données de eTemptation2. Tout d'abord, je commence par la création de l'en-tête de chaque groupe d'appels avec sa couleur propre. C'est à ce moment-là que la variable passée en paramètre « num\_tableau\_associe\_entete » entre en jeu, cette variable passée dans le programme principal prendra la valeur '1' signifiant la liste de groupe d'appels A-D, la valeur '2' pour la liste de groupes d'appels E-M et la valeur '3' pour la liste de groupes d'appels N-Z. Un simple switch permet d'associer la bonne couleur pour l'en-tête de chaque groupe d'appels. Une fois la bonne couleur choisie pour l'en-tête, il me suffit juste de créer la cellule et d'incorporer le titre de l'en-tête passée en paramètre avec la variable « \$en\_tete ».

Voici le résultat quand « num\_tableau\_associe\_entete » est égale à 1 et que l'en-tête passée est « Liste d'appel A-D » :

Liste d'appel A-D				
-------------------	--	--	--	--

Maintenant quand « num\_tableau\_associe\_entete » est égal à 2 et que l'en-tête passée est « Liste d'appel E-M » :

Liste d'appel E-M				
-------------------	--	--	--	--

Enfin dans le troisième cas où « num\_tableau\_associe\_entete » est égal à 3 et que l'en-tête passée est « Liste d'appel N-Z » :

Liste d'appel N-Z				
-------------------	--	--	--	--

Ensuite, vient la création de l'intitulé des colonnes. Pour rappel, il nous faut 5 colonnes avec le numéro de la ligne, le nom, le prénom, la date et l'heure de pointage ainsi qu'une colonne pour cocher la présence. Ici rien de très compliqué, il suffit juste de créer cinq cellules avec la méthode `Cell()` avec le nom de l'intitulé et sur fond jaune.

Voici le résultat :

N°	Nom	Prénom	Date et heure de pointage	Présent
----	-----	--------	---------------------------	---------

Maintenant voyons la partie la plus complexe de la méthode qui récupère les informations de la base de données pour les mettre dans la liste selon le groupe d'appels. C'est à ce moment-là que rentre


en jeu la variable passée en paramètre « \$success ». Cette variable contient une commande `odbc_exec` initialisée dans le programme principal, la commande `odbc_exec` va permettre d'exécuter la requête SQL préparée dans le fichier ConnexionBDD. Donc pour le tri des employés selon leur première lettre de nom de famille cela s'effectue directement dans la requête SQL que nous verrons dans le fichier ConnexionBDD.

Pour commencer, j'effectue une condition sur cette requête SQL. Pour ce faire, je déclare un `while` avec une commande `odbc_fetch_row` qui permet de lire une ligne du résultat de `odbc_exec` renvoyant `true` si la ligne existe et `false` sinon. Ainsi si `false` est renvoyé au `while` la boucle s'arrête. Mais si `true` est renvoyé le `while` continue de lire la suite des lignes de résultat de l'`odbc_exec`. Puis je récupère les données de la ligne lue dans une variable grâce à la commande `odbc_result`. Dans cette commande, on renseigne « \$success » (la requête SQL) et le champ récupéré dans la requête SQL. Cette étape effectuée, il ne me reste plus qu'à mettre les données dans la bonne liste.

Pour une question d'esthétisme et de visualisation, une ligne sur deux sera colorisée par la couleur associée à la liste. Pour ce faire, j'initialise une variable nommée « \$nblactuel » qui prend la valeur 1. Je la mets ensuite dans une condition qui vérifie le résultat du modulo de la variable par 2. Ainsi si le modulo est différent de 0 la couleur de la ligne sera blanche sinon elle prendra la même couleur que sa liste, exprimée par la variable « \$num\_tableau\_associe\_entete ». Enfin j'incrémente cette variable à la fin du `while` pour remplir la condition une fois sur deux, rendant ainsi la liste colorisée une fois sur deux. Maintenant ayant récupéré les données, il ne me reste plus qu'à créer les cellules pour y mettre mes variables avec la méthode `Cell()`. De ce fait, la liste se remplit dynamiquement suivant le nombre de données qu'elle reçoit.

Les noms et prénoms étaient renseignés au milieu de cellule car un espace y était imposé sur l'ancien programme. Pour répondre au besoin, il suffit de le supprimer. Et voilà notre liste d'appel avec les 3 groupes d'appel mis en page.

Voici un exemple :

		<b>Liste incendie</b>	Page 1/8
			Généré le : 2022-01-21 08:10

Liste d'appel A-D				
N°	Nom	Prénom	Date et heure de pointage	Présent
1	DELHALLE	MATHIS	2022-01-21 07:58	
2	DIOT	CLÉMENT	2022-01-21 07:58	

Dans la dernière partie du code, il est maintenant question de la création des listes de groupes d'ajout manuel pour les 3 groupes d'appels. Ces listes se trouveront à la suite de leurs listes de groupe d'appel respectifs. Pour les générer à la suite de leur liste de groupe d'appels respectifs, j'utilise la

méthode Addpage() pour créer une nouvelle page avec cette liste dans la méthode. Ensuite j'utilise le même procédé que la première partie du code pour les en-têtes.

Voici le résultat quand « num\_tableau\_associe\_entete » est égal à 1 et que l'en-tête passé est « Ajout Manuel de Liste d'appel A-D » :

**Ajout Manuel de Liste d'appel A-D**

Voici le résultat quand « num\_tableau\_associe\_entete » est égal à 2 et que l'en-tête passé est « Ajout Manuel de Liste d'appel E-M » :

**Ajout Manuel de Liste d'appel E-M**

Enfin dans le troisième cas où « num\_tableau\_associe\_entete » est égal à 3 et que l'en-tête passé est « Ajout Manuel de Liste d'appel N-Z » :

**Ajout Manuel de Liste d'appel N-Z**

Ensuite pour les intitulés des colonnes sur fond jaune, deux colonnes ont été supprimées et une autre a été ajoutée nommée « Motif (oubli, pointage, ...) ». Voici le résultat :

N°	Nom	Prénom	Motif (oubli,pointage,...)
----	-----	--------	----------------------------

Pour finir, on crée une liste contenant 15 lignes vides. Tout d'abord, j'ai créé une boucle for avec une variable de valeur 15 que doit atteindre la boucle for. De plus, cette liste doit contenir aussi une ligne sur deux colorée donc je réutilise la même condition que pour la liste d'appels. Enfin la méthode Cell() me permet de créer 4 cellules composant une ligne, la boucle se répétant 15 fois, 15 lignes vides vont être créées. Voici le résultat de cette boucle :

Ajout Manuel de Liste d'appel A-D			
N°	Nom	Prénom	Motif (oubli,pointage,...)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

Pour finir, la méthode retourne l'instance de l'objet classe PDF.

#### d. Présentation du fichier ConnexionBDD.xml

```
<?xml version='1.0' encoding='utf-8'?>
<config>
  <connexion>
    <!--les éléments de connexion à la base de données-->
    <driver>SQL Server</driver>
    <server>\                </server>
    <database>etemption2</database>
    <utilisateur>            </utilisateur>
    <motDePasse>            </motDePasse>
    <!--la requête SQL permet de récupérer les noms et prénoms des employés ayant pointé une fois dans la journée
    ainsi que la date et l'heure du pointage convertit au format (YYYY-MM-DD hh:mm),
    on attribue un nombre unique pour les noms de famille commençant par les lettres de A à D, de E à M et de N à Z
    -->
    <rqsql>
      SELECT
        HOPEMPL.Nom,
        UPPER(HOPEMPL.Prenom) AS prenom,
        CONVERT(VARCHAR(16),DATEADD(second,CONVERT(INTEGER,LEFT(Heure,2))*3600+CONVERT(INTEGER,RIGHT(Heure,2))*60,DAT),121) AS Date_Heure_Bipage2,
        CASE
          WHEN SUBSTRING(HOPEMPL.Nom,1,1) BETWEEN 'A' AND 'D' THEN 1
          WHEN SUBSTRING(HOPEMPL.Nom,1,1) BETWEEN 'E' AND 'M' THEN 2
          WHEN SUBSTRING(HOPEMPL.Nom,1,1) BETWEEN 'N' AND 'Z' THEN 3
          ELSE 99 END NumTableau
      FROM HOPPOIN
      INNER JOIN HOPEMPL ON HOPPOIN.MATRI=HOPEMPL.MATRI
      INNER JOIN (
        SELECT
          HOPPOIN.MATRI AS M,
          MAX(DATEADD(second,CONVERT(INTEGER,LEFT(Heure,2))*3600+CONVERT(INTEGER,RIGHT(Heure,2))*60,DAT)) AS Date_Heure_Bipage
        FROM HOPPOIN
        GROUP BY MATRI, HOPPOIN.MATRI
      ) S ON M=HOPPOIN.MATRI AND Date_Heure_Bipage=DATEADD(second,CONVERT(INTEGER,LEFT(Heure,2))*3600+CONVERT(INTEGER,RIGHT(Heure,2))*60,DAT)
      WHERE SENS='0'
      AND DATEADD(second,CONVERT(INTEGER,LEFT(Heure,2))*3600+CONVERT(INTEGER,RIGHT(Heure,2))*60,DAT)>= DateAdd(day,-2,getdate())
      ORDER BY HOPEMPL.Nom,prenom,Date_Heure_Bipage2;
    </rqsql>
  </connexion>
</config>
```

Ce fichier est de type XML car il convient mieux pour le stockage de nos données. Etant fournie par Horoquartz, il m'est impossible de modifier la base de données eTemption2. Seule la consultation est possible. La première partie de ce fichier contient l'ensemble des informations de connexion à la base de données nécessaires pour effectuer la connexion odbc dans le programme principal. Pour l'effectuer il faut renseigner le driver spécifique à chaque type de base de données, le server où est située la base de données pour la connexion, le nom de la base de données, le nom d'utilisateur et le mot de passe de connexion à la base de données.

La requête SQL qui permet de récupérer nos informations de pointage pour les mettre dans la liste incendie a été réalisée par M. BRASSART. Elle récupère le nom et le prénom. Le prénom est mis en majuscule car dans la base de données le format du prénom est différent, certains prénoms étaient en majuscules d'autres en minuscules. Dans un souci de répondre aux besoins, j'ai ajouté la méthode UPPER() forçant la récupération des prénoms en format majuscules. Ensuite, il faut récupérer la date ainsi que l'heure du pointage. Ici le CONVERT sert à multiplier le champ heure de la base de données par 3600 pour récupérer l'heure et par 60 pour récupérer les minutes car étant dans format particulier dans la base de données. De plus, on convertit le format du champ DAT pour juste récupérer la date.

Par ailleurs, pour effectuer le tri selon la première lettre du nom de famille des employés pour les groupes d'appels, on utilise le CASE...ELSE. Si on utilise la valeur 1, on récupère tous les employés ayant comme première lettre de nom de famille de A à D, ensuite si on utilise 2 de E à M et enfin si on utilise 3 de N à Z. Dans notre programme, c'est « \$num\_tableau\_associe\_entete » qui renseigne le chiffre

permettant le tri. Ensuite on effectue des jointures, l'une permettant de joindre la table HOPEMPL à la table HOPPOIN, l'autre permettant de vérifier la date et l'heure de pointage. Pour finir, on applique une condition à la requête SQL pour récupérer les informations des dernières 48 heures. Lorsqu'un employé oublie de badger à la fin de son service, il reste enregistré présent dans l'usine dans la base de données. Ainsi notre condition permet en cas d'oubli d'avoir une liste incendie à jour. Enfin on regroupe les données par nom, prénom ainsi que par heure et date de pointage.

Voici un exemple des quatre colonnes de la table Hoppoin que la requête utilise :

	MATRI	DAT	HEURE	SENS
1	STA172596	2022-02-08 00:00:00.000	0757	0
2	STA252964	2022-02-08 00:00:00.000	0757	0

Voici un autre exemple des trois colonnes de la table Hopeml que la requête utilise :

	MATRI	Nom	PRENOM
1	STA172596	DELHALLE	Mathis
2	STA252964	DIOT	Clément

Le résultat final de l'exécution de notre requête SQL dans la base de données :

	Nom	prenom	Date_Heure_Bipage2	NumTableau
1	DELHALLE	MATHIS	2022-02-08 07:57	1
2	DIOT	CLÉMENT	2022-02-08 07:57	1

## e. Présentation du fichier liste\_incendie.php

```
<?php

//inclut la classe PDF
require('PDF.php');

#Préparation des données#
$debug=false;
//on récupère dans les variables les éléments de connexion à la base de données et la requête SQL contenue dans un fichier XML,
//la fonction simplexml_load_file prend le nom du fichier XML en argument et va stocker le résultat dans une variable
$xml = simplexml_load_file('connexionBDD.xml');
$driver=$xml->connexion->driver;
$server=$xml->connexion->server;
$database=$xml->connexion->database;
$user=$xml->connexion->utilisateur;
$password=$xml->connexion->motDePasse;
$query=$xml->connexion->reqsql;

#Connexion à la BDD#
//odbc_connect(string $driver, string $dsn, string $user, $string $password, int SQL_CUR_USE_DRIVER) permet une connexion à une source
//Paramètres : $driver = le driver spécifique à chaque type de base de données
//              $server = le nom source de la base de données pour la connexion,
//              $user et $password = le nom d'utilisateur et le mot de passe
//              SQL_CUR_USE_ODBC = fixe le type de pointeur de résultat utilisé pour cette connexion
$connexion = odbc_connect("Driver={$driver}; Server=.$server."; Database=.$database, $user, $password, SQL_CUR_USE_ODBC);

if($connexion != false){
    if($debug == true){
        echo 'Connexion réussie<br>';
        var_dump($success);
    }

    //on formate la date et l'heure courante de la génération du pdf
    $today = date('Y-m-d H:i');
    //instanciation de la classe dérivée
    $pdf = new PDF('P', 'mm', 'A4', $today);

    //retourne un alias numérique pour le nombre total de pages
    $pdf->AliasNbPages();

    #1er liste d'appel de nom de A à D#
    //la méthode AddPage(string orientation, mixed size, int rotation) crée une nouvelle page,
    //Paramètres : orientation (la valeur par défaut est celle indiquée dans le constructeur),
    //              size (la valeur par défaut est celle indiquée dans le constructeur),
    //              rotation (la valeur par défaut est 0)
    $pdf->AddPage();
    $en_tete = "Liste d'appel A-D";
    //odbc_exec(resource $connection_id, string $query) envoie une commande SQL à la source de données ODBC, représentée par connection_id,
    //Paramètres : $connection_id = l'identifiant de connexion ODBC,
    //              $query = la requête SQL
    $success = odbc_exec($connexion,$query);
    $num_tableau_associe_entete = 1;
    $pdf = mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete);

    #2ème liste d'appel de nom de E à M#
    $pdf->AddPage();
    $en_tete = "Liste d'appel E-M";
    $success = odbc_exec($connexion,$query);
    $num_tableau_associe_entete = 2;
    $pdf = mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete);

    #3ème liste d'appel de nom de N à Z#
    $pdf->AddPage();
    $en_tete = "Liste d'appel N-Z";
    $success = odbc_exec($connexion,$query);
    $num_tableau_associe_entete = 3;
    $pdf = mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete);

    //odbc_close((resource $connection_id) ferme la connexion avec la source de données représentée par l'identifiant de connexion connection_id.
    odbc_close($connexion);

    $file='../../../../Bicwebdata/horoquartz/';
    //la méthode Output(string dest, string name, boolean isUTF8) envoie le document vers la destination de $file,
    //Paramètres : dest = destination où envoyer le document ('F' : sauver dans un fichier local, avec le nom indiqué dans name (peut inclure un répertoire),
    //              name = le nom du fichier,
    //              isUTF8 = indique si name est encodé en ISO-8859-1(false) ou en UTF-8(true). Par défaut la valeur est false
    $pdf->Output($file.'Liste_incendie.pdf','F');
}else{
    echo 'Connexion non aboutie';
}

?>
```

Ce fichier est le programme principal. Pour commencer, il fait appel au fichier de la classe PDF classe fille de la classe FPDF pour pouvoir utiliser les méthodes de PDF ainsi que les méthodes de FPDF qu'elle hérite. Ensuite, on charge le fichier XML de la connexion à la base de données ainsi que la requête SQL avec la fonction `simplexml_load_file()` pour pouvoir exécuter la connexion ODBC. Une fois le fichier chargé, on récupère dans les variables les éléments de connexion à la base de données et la requête SQL

contenue dans le fichier. La connexion à la base de données s'effectue avec `odbc_connect`. On stocke cette connexion dans une variable pour pouvoir l'utiliser pour exécuter une requête SQL. Si la connexion échoue on arrête le programme avec un message d'erreur mais si la connexion on réussit on exécute l'autre partie du code.

Ensuite, on instancie l'objet de la classe PDF grâce au `new` en y passant les valeurs devant être renseignés. De plus, on retrouve la méthode `AliasNbPages()` qui permet de retourner le nombre de pages total du fichier de la liste incendie que l'on affiche dans l'en-tête.

En outre, pour créer les pages de nos différentes listes de groupes d'appels, j'utilise la méthode `AddPage()` qui crée une nouvelle page. Après la création d'une nouvelle page, on utilise alors notre méthode `mise_en_page` en passant ses valeurs obligatoires. La variable « `$success` » récupère `odbc_exec` qui envoie une commande SQL à la source de données ODBC en passant en paramètre la connexion `odbc_connect` et la requête SQL.

En résumé, elle permet que lorsque « `$success` » est appelée d'exécuter la requête SQL préparée dans le fichier `ConnexionBDD`. On crée ainsi les 3 listes de groupes d'appels et leurs listes d'ajouts manuels créées automatiquement dans la méthode à la suite d'eux. Enfin on ferme la connexion `odbc` avec la source de données puis on envoie le fichier sur le serveur dans le répertoire `BicRWebData\horoquartz`.



### III. DESCRIPTION DE LA MISSION « CONFIGURATION ET INSTALLATION DE PC WINDOWS 10 »

#### 1. Genèse du projet

Au sein de l'usine BIC, on ne dénombre pas moins de 210 ordinateurs portables ou fixe. Les portables sont généralement à destination des personnes en télétravail. Malheureusement certains de ces ordinateurs tournent encore sur Windows 7 (abandonné par Microsoft le 14 janvier 2020) et ne reçoivent plus de mise à jour système de sécurité Microsoft. Sans ces mises à jour régulières le risque de piratage est beaucoup plus élevé au sein du site.

Donc pour plus de sécurité et sous la directive du siège social situé à Clichy, tous les pcs du site tournant sous Windows 7 (surtout ceux connectés au réseau de l'usine) doivent être remplacés en version Windows 10 tenue à jour régulièrement par Microsoft. A mon arrivée, cette mission était déjà lancée et seuls 39 ordinateurs étaient encore sous Windows 7. Pour les changer, des ordinateurs fixes et portables sont régulièrement commandés jusqu'à l'élimination de tous les ordinateurs sous Windows 7. De plus, un pc fixe du bureau d'études devait être formaté avec une clé bootable avec Windows 10 car étant sous Windows 7.

Les missions à effectuer dans l'ordre pour la configuration d'un pc Windows 10 :

- Le nommage du PC commençant par « VERW0 » avec un numéro s'incrémentant à chaque nouveau pc mis en place ;
- Le choix de la localisation du site de l'usine BIC ;
- La mise à jour de l'anti-virus Symantec Endpoint Protection du pc ;
- La mise à jour de Windows Update pour avoir le pc à jour ;
- La mise à jour des pilotes du pc avec le logiciel du fournisseur Dell Command Update ;
- Mettre le pc du sur le domaine du site ('eu') ;
- Mettre les comptes ver.g.admins et cli.g.mis\_pcscs dans le groupe Administrateurs ;
- Pour certains utilisateurs, remettre certains logiciels et données de l'ancien pc.



#### 2. Matériel mis à disposition

Pour réaliser cette mission, j'étais accompagné de mon collègue Mathis DELHALLE pour la configuration des pcs et leurs mises en place. A notre arrivée, nous avions à notre disposition cinq pc fixes et quatre pc portables. Pour faciliter la configuration de Windows 10 au démarrage, une image préparée est chargée sur le pc lors de la commande de celui-ci pour quelques euros de plus.

Cette image permet d'esquiver la configuration basique de Windows 10. Au démarrage l'image se lance automatiquement demandant la configuration de la langue du clavier ainsi que la localisation du site et le nommage du pc. Enfin un compte administrateur nous est donné en amont et arrivé sur le bureau, les différents logiciels du constructeur ainsi que l'antivirus sont installés.



Voici les caractéristiques des pcs que nous avons installés :

Spécificité	PC fixe	PC portable
Image		
Nom	OPTIPLEX 3080 Micro Form Factor	PRECISION 3560
Marque	Dell	Dell
Processeur	I5-10500T 2,30GHz	I7-1165G7 2,80GHz
RAM	16GO	32GO
Carte Graphique	Intel UHP Graphics 630	Nvidia T500 2GO
Stockage	SSD 512GO	SSD 1TO
Prix	Environ 600€	Environ 2000€

### 3. Planning

Pour mener à bien mon projet, un planning a été défini :

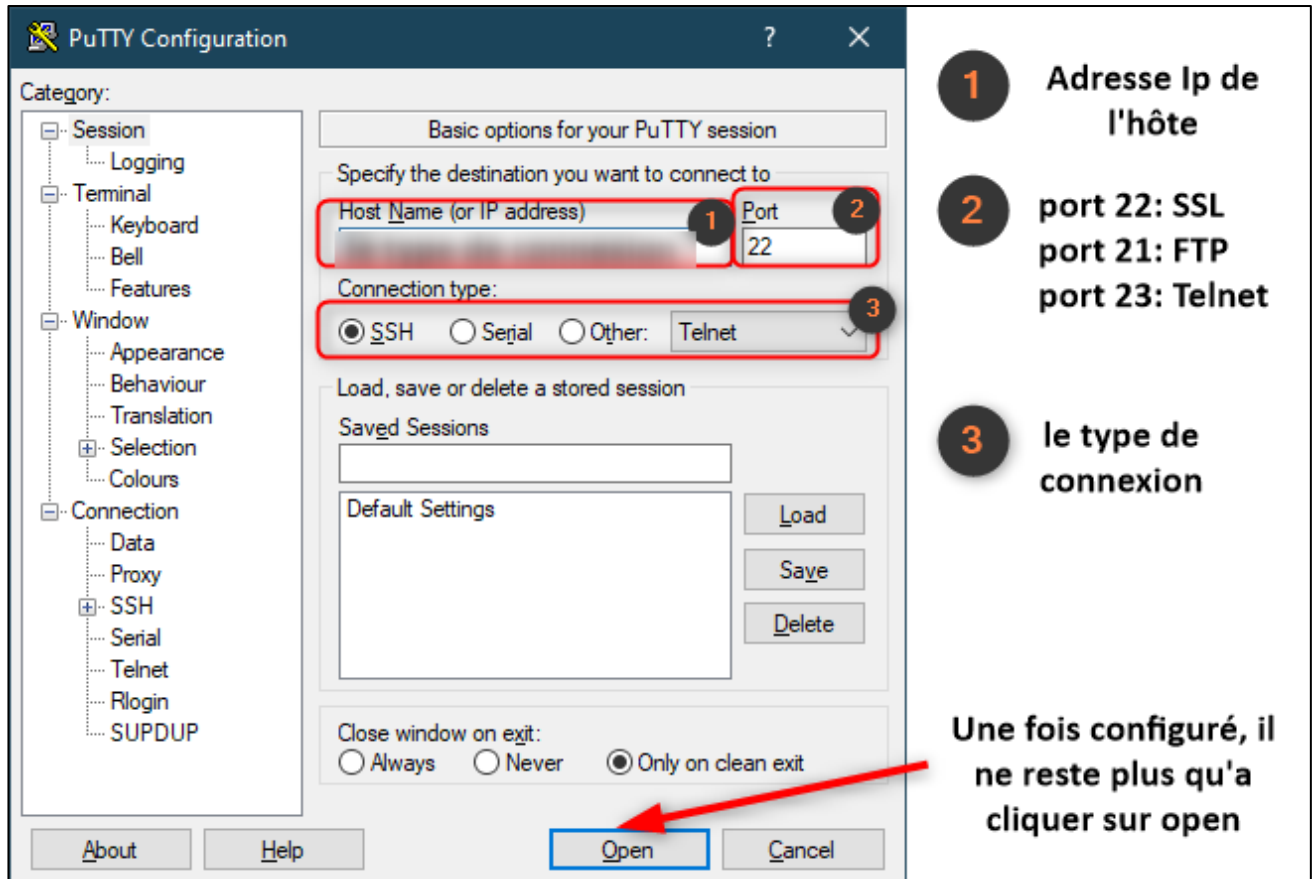
Tâches :	Semaines :				
	1	2	3	4	5
Formatage du pc et mise en place					
Configuration des ordinateurs					
Mise en place des ordinateurs					
Rédaction du rapport de stage					

### 4. Formatage du PC avec clé bootable

Le formatage du pc avec une clé bootable est simple à réaliser. Pour ce faire il faut mettre sous tension le pc en question puis brancher la clé USB qui contient l'image de Windows 10 sur l'un de ses ports et démarrer le pc sur le boot menu en appuyer frénétiquement sur le bouton F2 lors du démarrage du pc. Lorsque le menu apparaît, il faut sélectionner la clé USB et l'image se lance. Pour effectuer l'installation de Windows, il a fallu ensuite supprimer toutes les partitions de disque et en créer une nouvelle en faisant un clic droit « nouveau volume » renommé « disk data » sur le disque 1 non alloué. Puis définir la langue du clavier, le compte utilisateur, les règles de sécurité et enfin choisir le réseau du pc.

Ce dernier point retient tout particulièrement mon attention car nous avons avec mon collègue utilisé un câble ethernet relié à un port réseau au MIS. Cependant, Mr BRASSART a fermé l'ensemble des ports réseau non utilisés au sein du MIS pour éviter les vols de données par les personnes malveillantes

s'introduisant sur le site et pouvant s'introduire sur le réseau de l'usine en se branchant sur un des ports réseau du site. Donc il nous a fallu ouvrir un nouveau port au sein du MIS. Pour ce faire nous avons utilisé PuTTY. PuTTY est un émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet, rlogin, et TCP brut. Il permet également des connexions directes par liaison série RS-232. Voici l'interface de PuTTY :



**1** Adresse Ip de l'hôte

**2** port 22: SSL  
port 21: FTP  
port 23: Telnet

**3** le type de connexion

Une fois configuré, il ne reste plus qu'à cliquer sur open

Dans notre cas, nous avons utilisé le port SSH donc 22. Une fois cliqué sur « Open » cela ouvre une invite de commande. La première commande à réaliser est un show run qui permet d'afficher toutes les interfaces des ports sur le réseau hôte. Puis il faut vérifier que l'interface de notre port en question soit en mode access ou en mode trunk car le mode access est utilisé pour la connexion terminale d'un périphérique (pc, imprimante, serveur, ...) appartenant à un seul vlan. Le mode trunk est utilisé dans le cas où plusieurs vlan doivent circuler sur un même lien. Dans notre cas il était en mode trunk donc pour utiliser le port il nous faut le passer en mode access.

Pour passer d'un mode trunk à access voici les commandes sur PuTTY à réaliser :

- Conf t (permet la configuration)
- Int gi 1/0/47 (le port en question)
- No switchport mode trunk (la commande no permet d'annuler, on enlève tous les protocoles de sécurité également) no spanning-tree portfast edge, ...
- Ensuite on réécrit ce que l'on souhaite : switchport mode access

- Par exemple pour changer de vlan :
  - Conf t
  - Int gi 1/0/47
  - Switchport mode access vlan 50

Une fois le port configuré en mode access on remet tous les protocoles associés au port (spanning-tree,...). Puis on a renouvelé l'adresse ip du pc suite à l'ouverture du port et son branchement par câble. Il faut se rendre dans le cmd puis taper les lignes de commande suivantes :

- Ipconfig (affiche les valeurs de la configuration actuelle du réseau TCP/IP)
- Ipconfig/release (la configuration IP actuelle sera enlevée)
- Ipconfig/renew (le serveur DHCP attribue une nouvelle adresse IP à l'ordinateur ou la même si elle est disponible)
- Ipconfig (pour vérifier le changement d'adresse ip)

Pour finir, nous avons effectué la même procédure sur le pc formaté qu'au pc que nous avons configuré avec Mathis. Cette procédure est expliquée dans la partie suivante. Enfin nous avons transféré les anciennes données du pc puis mis en place le pc au bureau d'étude.

## 5. Processus de configuration des pc

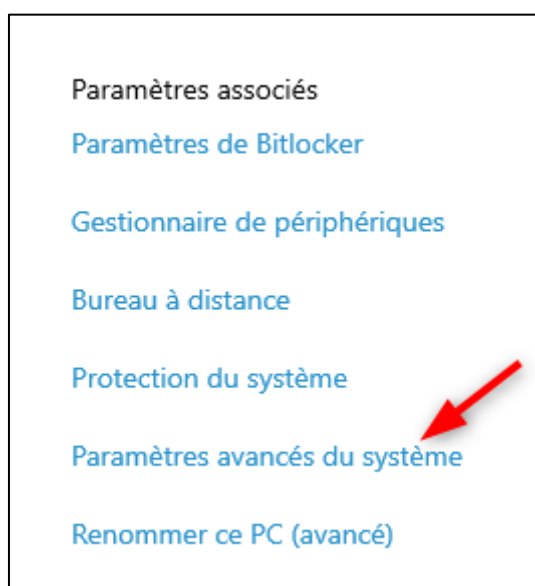
Le processus que je vais vous expliquer a été réalisé de la même façon sur tous les ordinateurs que nous avons configurés à quelques rares exceptions. Pour commencer après l'image initialisée au démarrage du pc, je commence par effectuer les mises à jour de l'antivirus. Il faut se rendre sur Symantec End point Protection, une fois installée, et cliquer sur « LiveUpdate » qui effectue les mises à jour de l'antivirus :



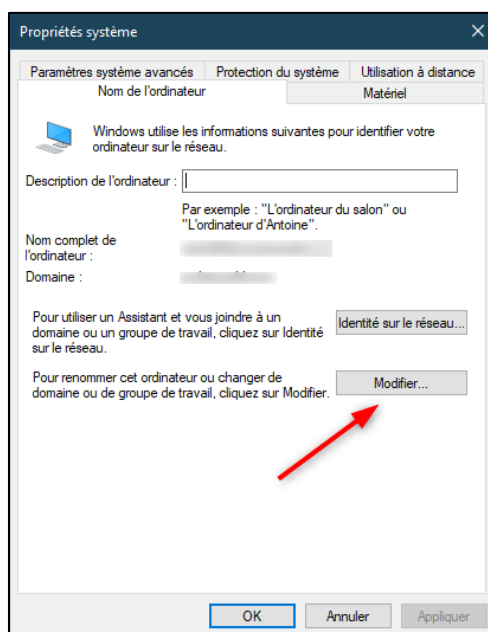
Ensuite, pour effectuer les mises à jour Windows Update, il faut se rendre dans les paramètres du pc->Mise à jour et sécurité et effectuer une recherche des mises à jour pour avoir les dernières versions. Le minimum étant la version 21H1 de Windows 10.

Puis il faut maintenant exécuter les mises à jour des pilotes (bios, souris, Realtek, ...), il faut se rendre sur le logiciel Dell Command Update, lancer la recherche de nouvelles mises à jour et les exécuter.

Maintenant que l'ordinateur est à jour, il faut le mettre dans le domaine de l'entreprise. Il faut se rendre dans les «paramètres»-> «Système»-> «A propos de» et sélectionner « paramètres avancés du système » :

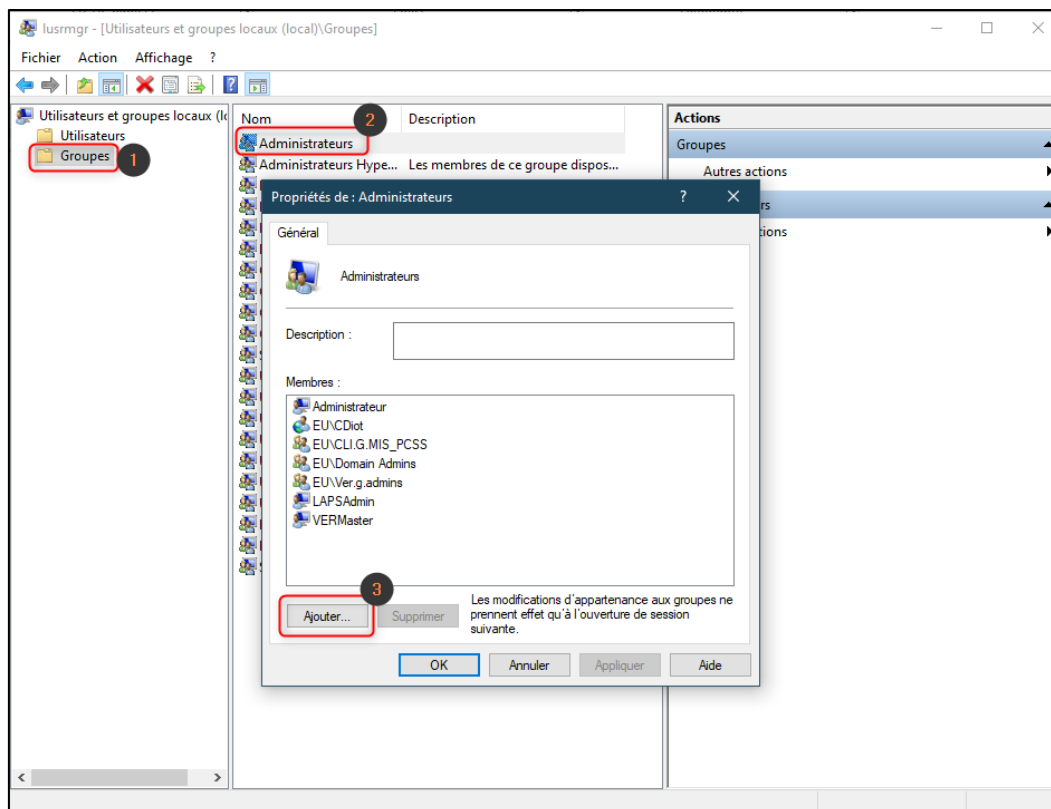


Puis se rendre dans l'onglet Nom de l'ordinateur et sélectionner cette option :



On arrive sur un nouvel onglet, ici on coche « Domaine » dans la partie « Membre d'un » et on entre « eu ». Après un redémarrage, le pc est enfin sous le domaine « eu » pour « eu.bicworld.com ».

Pour finir, il ne me reste plus qu'à mettre les comptes indiqués dans la genèse du projet dans le groupe administrateur. Il faut se rendre dans le « panneau de configuration »-> « modifier le type de compte » -> l'onglet « option avancées » et cliquer sur « avancé ». Une nouvelle page s'ouvre et j'ai suivi les instructions de la capture d'écran :



Enfin, on saisit les différents comptes administrateurs et on les applique. Pour finir, on remet les anciennes données du pc que nous allons changer (méthode expliquée dans mon rapport de l'année dernière). Avec dans certains cas, des logiciels à réinstaller.

## 6. Mise en place des nouveaux ordinateurs

Une fois tous les ordinateurs configurés, il faut maintenant les mettre en place dans l'usine et récupérer les anciens ordinateurs sous Windows 7. Pour ce faire, M. BRASSART nous a listé avec mon collègue Mathis tous les lieux où se trouvent des pc Windows 7 ainsi que les responsables de ces secteurs. Connaître les responsables des secteurs en question où se trouve les anciens pc, permet de planifier un rdv pour changer le pc. De plus, cela permet aux employés du secteur en question d'être au courant de nos changements et présence. Également, cela permet de savoir si des dangers existent pour la mise en place du pc (prise proche de produits dangereux ou de lames, ...) car BIC Rasoirs met en avant tout la sécurité de ses employés.

Voici un tableau récapitulatif de tous les ordinateurs changés :

Lieu	Personne à contacter	Données à transférer	Nom de l'ancien pc	Nom du nouveau pc	Danger
<b>Outillage</b>	Marina CHAVERNOZ (Chef de projet)	Bureau, favoris, documents, vidéos, images, One drive, GMAO (Saisie de données et monter d'informations)	VERW0441 (Pc fixe)	VERW0690 (Pc fixe)	Pont élévateur
<b>Dégraissage</b>	Adrien LEVILLY (Responsable production lames) + Elodie CONFRERE (Responsable HSE)	Bureau, favoris, documents, vidéos, images, One drive	VERW0472 (Pc fixe)	VERW0691 (Pc fixe)	Pas de danger
<b>Maintenance lames</b>	Xavier GRIFFON (Chef de projet)	Bureau, favoris, documents, vidéos, images, One drive	VERW0365 (Pc fixe)	VERW0699 (Pc fixe)	Pas de danger
<b>Comfort Twin</b>	Michael POUETTRE (Project Manager)	Bureau, favoris, documents, vidéos, images, One drive	VERW0391 (Pc fixe)	VERW0702 (Pc fixe)	Vérification de l'accessibilité de l'arrivée électrique
<b>Fours</b>	Adrien LEVILLY (Responsable production lames)	Bureau, favoris, documents, vidéos, images, One drive	VERW0448 (Pc fixe)	VERW0700 (Pc fixe)	Pas de danger
<b>Contrôle d'entrée</b>	Brice DESACHY (Responsable qualité lames)	Bureau, favoris, documents, vidéos, images, One drive, Centrage Presse (logiciel d'impression)	VERW0419 (Pc fixe)	VERW0684 (Pc fixe)	Pas de danger
<b>Maintenance générale</b>	Yannick BERDON (Ouvrier)	Bureau, favoris, documents, vidéos, images, One drive	VERW0705 (Pc fixe)	VERW0380 (Pc fixe)	Pas de danger
<b>MEB</b>	Laurence PECQUERY (Contrôle international)	Bureau, favoris, documents, vidéos, images, One drive, MarkWare (logiciel	VERW0364 (Pc fixe)	VERW0706 (Pc fixe)	Pas de danger

		d'impression d'étiquettes)			
HSE	Fanny PIERRA (Chef de projet HSE)	Bureau, favoris, documents, vidéos, images, One drive		VERW0701 (Pc portable)	Pas de danger
Bureau de maintenance automatisation				VERW0707 (Pc portable)	Pas de danger
Bureau de maintenance automatisation				VERW0708 (Pc portable)	Pas de danger

#### IV. DESCRIPTION DE LA MISSION « AMELIORATION DE LA SAISIE DE LA PACKING LIST »


##### 1. Genèse du projet

BIC Rasoirs dispose de nombreux fournisseurs pour l'approvisionnement de matières premières comme l'acier pour la fabrication de lames de rasoir ou encore le plastique pour le moulage des pièces. Mon projet se concentre sur les fournisseurs d'acier de l'usine. Lors de leurs livraisons, ces fournisseurs fournissent une packing list avec la marchandise. C'est une liste contenant les informations fournisseur de la caisse + de la bobine. De plus, cette packing list est propre à chaque fournisseur.

Aujourd'hui cette packing list est transmise au format CSV lors de la livraison. Ce format est obligatoire au programme qui permet d'enregistrer cette packing list dans la base de données Lames\_SS2L (SS2L = Suivi des lots lames). Ainsi lorsqu'un défaut est présent sur une bobine d'acier, il est alors possible de trouver chez quel fournisseur ce défaut est apparu. En déposant ce fichier CSV dans le répertoire « SS2L »-> « Packing »-> « Auto » puis en cliquant sur l'onglet « Auto » dans « SS2L » sur BicRWeb, le programme va alors interpréter le fichier CSV et insérer les informations aux bons endroits. Remarque : un seul packing list peut être déposé à chaque fois dans ce répertoire pour que les données soient récupérées par SS2L.

Ce programme comprend deux parties. Une partie BRISE pour le choix du format et la saisie d'informations complémentaires et une partie PHP qui interprète le fichier CSV pour plus tard insérer les données dans la base de données avec des requêtes SQL. Cependant, ce fichier est différent chez chaque fournisseur. En outre, cette tâche est effectuée par Madame Sylvie BONNIN (Gestion – Approvisionnement et chargée de la récupération des données fournisseurs des packing lists) :

Voici l'apparence du module web SS2L dans BicRWeb et pour utiliser la récupération des données d'une packing list, il suffit de cliquer sur le bouton « Auto » :

 SS2L - PACKING LIST FEUILLARD ACIER	
<b>SAISIES</b>	
Saisie d'une packing	
Confirmation d'une packing	
Réception d'une packing	
Auto	
<b>CONSULTATIONS</b>	
Voir la saisie en cours	

BIC a créé un fichier standardisé de packing list calqué sur leur premier fournisseur SANDVIK. Ce fichier standard sert dans deux cas :

- Si le fournisseur n'envoie pas de packing list, le fichier standard permet alors de créer une packing list à partir des papiers fournis avec la commande



- Dans le cas où le fournisseur fournit une packing list, le fichier standard va permettre de tester sa compatibilité. Ainsi on peut constater si une donnée manque dans la packing list du fournisseur par rapport aux données essentielles de la packing list au format standard. Cette packing list sera alors mise dans notre programme au format fournisseur ne collant pas au format standard (manque d'une ou plusieurs données)

Ce format standard est construit avec les colonnes suivantes : Lot, Heat, Order, Pos, Custom Order, Packing slip, Case, Coil, Strand, Weight, Coil size, Master Sample. Or certains fournisseurs choisissent d'utiliser leurs propres formats. Donc mon projet consiste à adapter le programme pour qu'il puisse prendre en compte le format standard et fournisseur. Ce projet permettra de faire gagner à Madame BONNIN 3H de saisie.

## 2. Planning

Pour mener à bien mon projet, un planning a été défini :

Tâches :	Semaines :				
	1	2	3	4	5
Modification et amélioration du programme					
Présentation, formation et mise en production					
Rédaction du rapport de stage					

## 3. Développement des améliorations de la saisie de packing list

### a. Résultat final

```

Nouvelle Ligne
15474;562628;502-41895;01;21000080 DU 18.3.2;4759756;5615393;1;4A;31.3;13,74 x 0,074;

Nouvelle Ligne
15474;562628;502-41895;01;21000080 DU 18.3.2;4759756;5615393;1;4A;31.3;13,74 x 0,074;
Requete SQL "query_acier_id":
Select acierid From Aciers WHERE AcierDescription='13.74 x 74µ'
Requete SQL "query_lame_id":
Select MAX(lameId) as Lameld,LameAcierId From Lames WHERE lameAcierId IS NOT NULL and LameDebutProcess=1 and lameAcierId=NULL GROUP BY LameAcierId

array(14) {
  ["PackingBobineRefFourLot"]=>
  string(14) "562628 ; 15474"
  ["PackingBobineRefFourBobine"]=>
  string(6) "1 ; 4A"
  ["PackingCaisseFournisseurId"]=>
  string(1) "3"
  ["PackingCaisseDateLivraison"]=>
  string(8) "20220215"
  ["PackingCaisseCodeCommande"]=>
  string(18) "21000080 DU 18.3.2"
  ["PackingCaisseNumero"]=>
  string(7) "5615393"
  ["PackingCaissePoidsNet"]=>
  int(0)
  ["PackingBobineIS"]=>
  int(0)
  ["PackingBobinePoidsLivre"]=>
  float(31.3)
  ["PackingBobineEstimerPoids"]=>
  int(0)
  ["PackingBobineEtapeFluxStatut"]=>
  string(3) "PKL"
  ["acier"]=>
  string(12) "13.74 x 74µ"
  ["PackingBobineAcierId"]=>
  NULL
  ["PackingBobineLameId"]=>
  NULL
}

```

Cette page marque la fin de ma partie sur ce projet et sa mise en production. La suite du programme a été effectuée par M. BRASSART. Cette page montre que ma partie du programme a bien récupéré les informations de la ligne du fichier CSV et les a associées avec les bonnes variables pour ensuite effectuer leurs insertions dans la base de données. Ce programme récupère dans le fichier CSV les données des colonnes Lot, Heat, Customer Order, Case, Coil, Strand, Weight, Coil Size et le Master Sample.

## b. Présentation du script BRISE

```

<?xml version='1.0' encoding='utf-8'>
<brise titre="SS2L : Packing List - Saisie de la livraison" debut='0' debug='oui' retour='index.php' >

  <alias>
    <externe nomFichier='../config.xml' xpath='/root/connexion' />
  </alias>

  <var nom='idFournisseur' />
  <var nom='dateLivraison' />
  <var nom='dateLivraisonSansTiret' />
  <var nom='nom_fournisseur' />
  <var nom='format' />
  <var nom='commande' />

  <var nom='erreur' />

  <source nom='Fournisseur'>
    <connexion type='alias' nom='ConnexionSql' >
      SELECT
        [FournisseurId]
        , [FournisseurNom]
        , CONVERT (VARCHAR, FournisseurId) + '_' + FournisseurNom AS Cle
      FROM Fournisseurs
      ORDER BY FournisseurNom
    </connexion>
  </source>

  <source nom='FormatFournisseur'>
    <connexion type='alias' nom='ConnexionSql' >
      SELECT
        [FournisseurId]
        , [FournisseurNom]
        , CONVERT (VARCHAR, FournisseurId) + '_' + FournisseurNom AS Cle
      FROM Fournisseurs
      WHERE FournisseurFormat=1
      ORDER BY FournisseurNom
    </connexion>
  </source>

  <!-- Saisie information concernant la livraison -->
  <etape nom='0' matriceL='10'>
    <affichage type='texte' val='$erreur$' test='$erreur$' #DIF# '' aspect='erreur grand gras' />
    <saisie var='format' type='liste' label='Format' >
      <champ nom='' cle='' />
      <champ nom='Standard' cle='1' />
      <champ nom='Fournisseur' cle='2' />
    </saisie>
    <bouton nom='Suivant' label='Saisie Fournisseur' etape='verifFormat' />
  </etape>

  <etape nom='verifFormat'>
    <calcul var='erreur' val='' />
    <calcul var='erreur' test='$format$' #EG# '' val='Saisir un format CSV !' />
    <saut test='$erreur$' #EG# '' etape='ConditionFormat' />
    <saut test='$erreur$' #DIF# '' etape='0' />
  </etape>

  <etape nom='ConditionFormat'>
    <saut test='$format$' #EG# 1' etape='FormatStandard' />
    <saut test='$format$' #EG# 2' etape='FormatFournisseur' />
  </etape>

```

```

<etape nom='FormatStandard'>
  <saisie var='format' type='ligne' testM='false' label='Format CSV' val='Standard' />
  <chargement source='Fournisseur' />
  <calcul var='dateLivraison' valCalc='date("Y-m-d")' />
  <affichage type='texte' val='$erreur$' test='$erreur$' #DIF# '' aspect='erreur grand gras' />
  <saisie var='idFournisseur' type='liste' label='Fournisseur' testF='true' />
  <champ nom='' cle='' />
  <champSrc source='Fournisseur' nom='FournisseurNom' cle='Cle' />
</saisie>
<saisie var='dateLivraison' type='date' label='Date' />
<bouton nom='Retour' label='Retour' etape='0' />
<bouton nom='Suivant' label='Saisie caisse' etape='VerifSaisieS' />
</etape>

<etape nom='VerifSaisieS'>
  <calcul var='erreur' val='' />
  <calcul var='erreur' test='$idFournisseur$' #EG# '' val='Saisir un fournisseur !' />
  <calcul var='erreur' test='$dateLivraison$' #EG# '' val='Saisir une date !' />
  <calcul var='erreur' test='preg_match('^[0-9]{4})-([0-9]{2})-([0-9]{1})([0-9]{1})$', '$dateLivraison$') #EG#0' val='Saisir une date au format aaaa-mm-jj !' />
  <calcul var='dateLivraisonSansTiret' valCalc='str_replace('-', '', $dateLivraison$)' />
  <saut test='$erreur$' #EG# '' etape='SaisieCaisse' />
  <saut test='$erreur$' #DIF# '' etape='FormatStandard' />
</etape>

<etape nom='FormatFournisseur'>
  <saisie var='format' type='ligne' testM='false' label='Format CSV' val='Fournisseur' />
  <chargement source='FormatFournisseur' />
  <calcul var='dateLivraison' valCalc='date("Y-m-d")' />
  <affichage type='texte' val='$erreur$' test='$erreur$' #DIF# '' aspect='erreur grand gras' />
  <saisie var='idFournisseur' type='liste' label='Fournisseur' testF='true' />
  <champ nom='' cle='' />
  <champSrc source='FormatFournisseur' nom='FournisseurNom' cle='Cle' />
</saisie>
<saisie var='dateLivraison' type='date' label='Date' />
<bouton nom='Retour' label='Retour' etape='0' />
<bouton nom='Suivant' label='Suivant' etape='VerifSaisieF' />
</etape>

<etape nom='VerifSaisieF'>
  <calcul var='erreur' val='' />
  <calcul var='erreur' test='$idFournisseur$' #EG# '' val='Saisir un fournisseur !' />
  <calcul var='erreur' test='$dateLivraison$' #EG# '' val='Saisir une date !' />
  <calcul var='erreur' test='preg_match('^[0-9]{4})-([0-9]{2})-([0-9]{1})([0-9]{1})$', '$dateLivraison$') #EG#0' val='Saisir une date au format aaaa-mm-jj !' />
  <calcul var='dateLivraisonSansTiret' valCalc='str_replace('-', '', $dateLivraison$)' />
  <saut test='$erreur$' #EG# '' etape='VerifSaisieCommande' />
  <saut test='$erreur$' #DIF# '' etape='FormatFournisseur' />
</etape>

<etape nom='VerifSaisieCommande'>
  <saut test='$idFournisseur$' #EG# "5_JINDAL" etape='SaisieCommande' />
  <saut test='$idFournisseur$' #DIF# "5_JINDAL" etape='SaisieCaisse' />
</etape>

<!-- si le fournisseur est JINDAL -->
<etape nom='SaisieCommande'>
  <saisie var='commande' type='ligne' label='Numéro de commande' />
  <bouton nom='Retour' label='Retour' etape='FormatFournisseur' />
  <bouton nom='Suivant' label='Saisie Caisse' etape='SaisieCaisse' />
</etape>

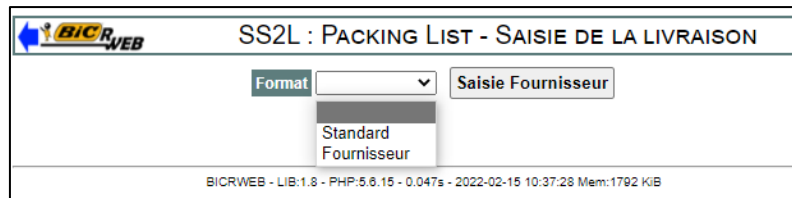
<etape nom='SaisieCaisse'>
  <branchement lien='Preparation_auto_packing_list.php?dateLivraison=$dateLivraisonSansTiret&camp;fcl=$idFournisseur&camp;formatcle=$format&camp;tab_Param=$commande' />
</etape>
rise>

```

☐ Si le format est Standard

☐ Si le format est Fournisseur

Cette partie BRISE permet de connaître le format du fichier CSV, le fournisseur, la date et le numéro de commande dans le cas du format fournisseur JINDAL car cette donnée n'est pas renseignée dans le fichier CSV. Le fichier contenant le script BRISE se nomme « SS2L\_Packing\_Saisie\_Livraison\_auto.xml ». Tout d'abord, j'initialise des requêtes SQL avec des variables et fais commencer mon script à l'étape « 0 ». Cette étape affiche un message avec une liste demandant la sélection du format du fichier CSV. Ici deux choix sont possibles : Standard ou Fournisseur. Dans le cas où « Standard » est choisi la valeur 1 est renvoyée et dans le cas où « Fournisseur » est choisi la valeur 2 est renvoyée. Enfin le bouton « Saisie Fournisseur » nous amène alors à l'étape « verifFormat » :

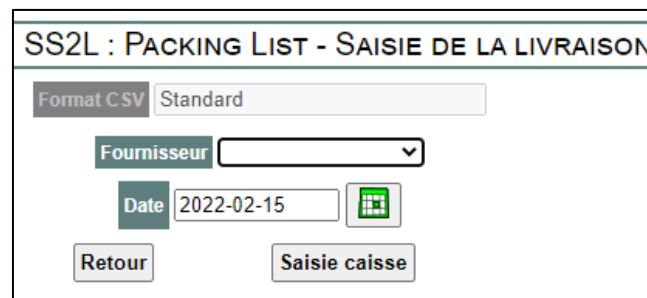


Une fois sur l'étape « verifFormat » celle-ci va vérifier si l'utilisateur a bien sélectionné un format, dans le cas contraire un message d'erreur apparaît. Si aucune erreur est détectée, le script passe à l'étape « ConditionFormat » qui amène l'utilisateur à l'étape « FormatStandard » si le format Standard a été sélectionné sinon à l'étape « FormatFournisseur » si le format choisi était le format Fournisseur.

Maintenant, nous allons prendre le cheminement d'un utilisateur qui choisit le format « Standard ». Donc le script se retrouve alors à l'étape « FormatStandard » qui demande alors la saisie de la date de livraison ainsi que le nom du Fournisseur dans une liste. Les données de la liste sont générées grâce à la requête SQL de la source « Fournisseur » qui sélectionne l'id et le nom du fournisseur ainsi que leur conversion en chaînes de caractères avec l'id du fournisseur tiret le nom du fournisseur (par exemple : 1\_Hitachi) de la table Fournisseur. Ici étant un format Standard tous les fournisseurs sont récupérés :

FournisseurId	FournisseurNom
1	Hitachi
2	Sandvik
3	Outokumpu
4	Uddeholm
5	JINDAL
6	Voestalpine
7	Bic Violex Hitachi
8	Bic Violex Sandvik

Cette liste affiche alors le nom des fournisseurs et renvoie à leur sélection idFournisseur\_FournisseurNom. J'affiche également pour un meilleur confort visuel la sélection du format. Enfin deux boutons sont mis à disposition, l'un « Retour » pour retourner à l'étape précédente et l'autre « SaisieCaisse » permettant de passer à l'étape « VerifSaisieS » :



Lors du clic sur la liste :

SS2L : PACKING LIST - SAISIE DE LA LIVRAISON

Format CSV Standard

Fournisseur

Date 2021

Retour

- Bic Violex Hitachi
- Bic Violex Sandvik
- Hitachi
- JINDAL
- Outokumpu
- Sandvik
- Uddeholm
- Voestalpine

CRWEB - LIB:1.8 - PHP:5.6.15 3 Mem:1792 KiB

Ensuite, l'étape « verifSaisieS » permet de vérifier que l'utilisateur a bien sélectionné un fournisseur, une date de livraison et que celle-ci est composé de caractères numériques avec 4 chiffres pour l'année et 2 chiffres pour le mois et le jour. Enfin pour faciliter la suite du code, on utilise la méthode `str_replace()` pour enlever les tirets dans la date que l'on récupère. Si une erreur est détectée un message d'erreur apparaît sinon le script va à l'étape « SaisieCaisse ».

Maintenant voyons le cas d'un utilisateur ayant choisi le format « Fournisseur ». Il se retrouve alors automatiquement à l'étape « FormatFournisseur » qui procède exactement comme la méthode « FormatStandard » mais celui-ci alimente sa liste de fournisseur avec la source « FormatFournisseur ». Cette requête comparée à celle de la source « Fournisseur » est la même mais une condition est imposée sur la colonne FournisseurFormat de la table Fournisseur. Cette colonne a été créée par mes soins renseignant la possession du format fournisseur par l'usine. Si la donnée « 1 » y est inscrite cela signifie que BIC possède le format fournisseur sinon la colonne prend la valeur « 0 ». Ici comme exprimé dans la genèse du projet, BIC possède le format fournisseur uniquement pour SANDVIK et JINDAL :

FournisseurId	FournisseurNom	FournisseurFormat
1	Hitachi	0
2	Sandvik	1
3	Outokumpu	0
4	Uddeholm	0
5	JINDAL	1
6	Voestalpine	0
7	Bic Violex Hitachi	0
8	Bic Violex Sandvik	0

Voici l'apparence de la méthode « FormatFournisseur » :



Le bouton « Suivant » amène alors à l'étape « verifSaisieF » qui comme l'étape « verifSaisieS » vérifie la saisie de l'utilisateur et récupère la date de livraison sans tiret. Une fois cette étape exécutée sans erreur, le script reprend à l'étape « VerifSaisieCommande » car comme indiqué dans la genèse du projet au format fournisseur, JINDAL ne possède pas de numéro commande nécessaire à l'exécution des requêtes SQL du code PHP. Donc cette étape permet de vérifier si l'utilisateur a sélectionné le fournisseur JINDAL. Si ce n'est pas le cas, l'utilisateur passe à l'étape « SaisieCaisse » sinon si JINDAL a été sélectionné l'utilisateur passe à l'étape « SaisieCommande ». Cette étape permet alors uniquement pour le format fournisseur de JINDAL d'entrer le numéro de commande, elle possède également deux boutons, l'un pour revenir à l'étape précédente (Retour) et l'autre pour se rendre à l'étape « SaisieCaisse » (Saisie Caisse) :



Dans tous les cas, à la fin l'utilisateur se retrouvera à l'étape « SaisieCaisse » constituant la dernière étape du script BRISE. Cette étape sert uniquement à un branchement avec un lien pointant sur l'autre script PHP. Dans cette URL sont passés les paramètres saisis par l'utilisateur dans le BRISE donc la date de livraison sans tiret, la variable idFournisseur composée de l'id du fournisseur et de son nom (par exemple, 2\_Sandvik), le format (1 pour le standard et 2 pour le fournisseur) et enfin le numéro de commande dans le cas de JINDAL.

### c. Présentation du script PHP

☐ **Mes ajouts**

```

<?php
//// Librairie BicRWeb ////
require_once '...\lib-php\bfp.inc.php';
$debug=true;
/* $emplacement_fichier_a_traiter='C:/Tests_dev/'; */
$emplacement_fichier_a_traiter='../../BicRWebData/SS2L/Packing/Auto/';
$emplacement_backup='../../BicRWebData/SS2L/Packing/Backup/';

if(!is_dir($emplacement_fichier_a_traiter)){
    echo 'Création du répertoire '.$emplacement_fichier_a_traiter.BR;
    mkdir($emplacement_fichier_a_traiter);
}
if(!is_dir($emplacement_backup)){
    echo 'Création du répertoire '.$emplacement_backup.BR;
    mkdir($emplacement_backup);
}

$fichier_query='Query_SS2L_packing.xml';
$conn=bfp_odbc_connect($fichier_query,'databasel');

$fichier_erreur='Erreur_packing_list_'.date('Ymd_his').'.txt';
$flag_erreur=false;

$nom_fichier_erreur=$emplacement_fichier_a_traiter.'.'.$fichier_erreur;
$monfichier_erreur = fopen($nom_fichier_erreur, 'w+');
$ligne='Lot;Heat;Order;Pos;Customer Order;PackingSlip;Case;Coil;Strand;Weight;Coil Size;Master Sample'.CHR(13);
fputs($monfichier_erreur, $ligne);
$ligne='';

//Recuperation des fichiers
$format='0';
$fournisseur_id='0';
$fournisseur_nom='';
$date_livraison='';
$commandeJindal='0';

// Recuperation des paramètres
if(isset($_GET['dateLivraison'])) $date_livraison=$_GET['dateLivraison'];

if(isset($_GET['fclle'])){
    $tab_fourn=explode('_',$_GET['fclle']);
    $fournisseur_id=$tab_fourn[0];
    $fournisseur_nom=$tab_fourn[1];
}

//1 : Standard
//2 : Fournisseur
if(isset($_GET['formatcle'])) $format=$_GET['formatcle'];

//Récupération du numéro de commande pour JINDAL dans le format Fournisseur
if ($format==2){
    if(isset($_GET['tab_Param'])) $commandeJindal=$_GET['tab_Param'];
}
    
```



```
if($dossier=opendir($emplacement_fichier_a_traiter)){
    $dh = opendir($emplacement_fichier_a_traiter);
    // Recuperation des fichiers
    while (false != ($filename = readdir($dh)) {
        $files[] = $filename;
    }
    closedir($dh);

    //traitement des fichiers
    foreach($files as $f){
        $extension_fichier=substr($f,-3);
        //recuperation seulement des fichiers csv
        if(strtoupper($extension_fichier)=='CSV'){
            $nom_fichier_a_traiter=substr($f,0,strlen($f)-4);
            $fichier=$emplacement_fichier_a_traiter.$nom_fichier_a_traiter.'.csv';
            $tab_caisses_present_dans_db=array();
            $tab_bobines_present_dans_db=array();

            if($debug) echo 'Fichier : '.$fichier.BR;

            $i=1;
            //Lecture du fichier
            $monfichier = fopen($fichier, 'r');
            while (!feof($monfichier)) {
                $ligne = fgets($monfichier);
                if($debug) echo BR.'Nouvelle Ligne'.BR.$ligne.BR;
                if($ligne!=''){
                    // Je ne traite pas la première ligne contenant les en-tetes
                    if($i<>1){
                        // Initialisation des variables
                        $tab_donnee=array();
                        $commande='';
                        $ref_four_lot='';
                        $ref_four_bobine='';
                        $caisse='';
                        $acier='';
                        $coulee='';
                        $lot='';
                        $bande='';
                        $brin='';
                        $poids=0.0;
                        $ms=0;
                        $tab_donnee=explode(";", $ligne);

                        //traitement du csv selon le format standard
                        if($format==1){
                            //decoupage du csv pour insertion
                            if( count($tab_donnee)==12){
                                $commande=$tab_donnee[4];
                                $caisse=$tab_donnee[6];
                                $a=$tab_donnee[10];
                                $coulee=$tab_donnee[1];
                                $lot=$tab_donnee[0];
                                $bande=$tab_donnee[7];
                                $brin=$tab_donnee[8];
                                $poids=$tab_donnee[9];
                                if(ltrim(rtrim($tab_donnee[11]))=='X'){
                                    $ms=1;
                                }
                                //traitement pour Hitachi et BIC Violex Hitachi
                                if($fournisseur_id==7 or $fournisseur_id==1){
                                    $ref_four_lot=$coulee;
                                    $ref_four_bobine=$lot.' ; '.$bande.' ; '.$brin;
                                }
                                //traitement pour JINDAL
                                }else if($fournisseur_id==5){
                                    $ref_four_lot=$coulee.' ; '.$lot;
                                    $ref_four_bobine='- ; '.$brin;
                                }else{
                                    $ref_four_lot=$coulee.' ; '.$lot;
                                    $ref_four_bobine=$bande.' ; '.$brin;
                                }
                                $acier=$a;
                                //Gestion de l'acier car la description de l'acier dans la base de données n'est pas la même que le fou
                                if($a=='16,76 x 0,10') $acier='16,76 x 0,1';
                                if($a=='13,74 x 0,074') $acier='13.74 x 74µ';
                                if($a=='22,10 x 0,0761') $acier='22,10 x 76µ';
                                if($a=='16,76 x 0,074') $acier='16.76 x 74µ';
                            }
                        }
                        //traitement du csv selon le format fournisseur
                    }
                }
            }
        }
    }
}
```

3

```
//traitement du csv selon le format fournisseur
}else{
    switch($fournisseur_id){
        //traitement de sandvik
        case 2:
            $commande=$tab_donnee[4];
            $caisse=$tab_donnee[6];
            $a=$tab_donnee[10];
            /* $coulee=$tab_donnee[5]; */
            $coulee=$tab_donnee[1];
            $lot=$tab_donnee[0];
            $bande=$tab_donnee[7];
            $brin=$tab_donnee[8];
            $poids=$tab_donnee[9];
            if(ltrim(rtrim($tab_donnee[11]))=='X'){
                $ms=1;
            }
            $ref_four_lot=$coulee.' ; '.$lot;
            $ref_four_bobine=$bande.' ; '.$brin;
            $acier=$a;
            //Gestion de l'acier car la description de l'acier dans la base de données n'est pas la même que le fournisseur
            if($a=='16,76 x 0,10') $acier='16,76 x 0,1';
            if($a=='13,74 x 0,074') $acier='13.74 x 74p';
            if($a=='22,10 x 0,0761') $acier='22,10 x 76p';
            if($a=='16,76 x 0,074') $acier='16.76 x 74p';
            break;

            //traitement de jindal
            case 5:
                $commande=$commandeJindal;
                $caisse=$tab_donnee[9];
                $a=$tab_donnee[5].' x '.$tab_donnee[4];
                $coulee=$tab_donnee[3];
                $lot=$tab_donnee[1];
                /* $bande=$tab_donnee[8]; */
                $brin=$tab_donnee[2];
                $poids=$tab_donnee[6];
                if(ltrim(rtrim($tab_donnee[10]))=='M.S'){
                    $ms=1;
                }
                $ref_four_lot=$coulee.' ; '.$lot;
                $ref_four_bobine='- ; '.$brin;
                $acier=$a;
                //Gestion de l'acier car la description de l'acier dans la base de données n'est pas la même que le fournisseur
                if($a=='22,10 x 0,076') $acier='22,10 x 76p';
                break;
            }
        }

        //Preparation des variables pour les requetes
        $par['PackingBobineRefFourLot']=$ref_four_lot;
        $par['PackingBobineRefFourBobine']=$ref_four_bobine;
        $par['PackingCaisseFournisseurId']=$fournisseur_id;
        $par['PackingCaisseDateLivraison']=$date_livraison;
        $par['PackingCaisseCodeCommande']=$commande;
        $par['PackingCaisseNumero']=$caisse;
        $par['PackingCaissePoidsNet']=0;
        $par['PackingBobineMS']=$ms;
        $par['PackingBobinePoidsLivre']=floatval($poids);
        $par['PackingBobineEstimerPoids']=0;
        $par['PackingBobineEtapeFluxStatut']='PKL';
    }
}
```

Ce fichier PHP nommée Preparation\_auto\_packing\_list.php permet de récupérer les données dans le fichier CSV pour les stocker dans des variables qui seront ensuite insérées dans la base de données Lames\_SS2L. Ce fichier a été créé par M. BRASSART et les encadrés rouges représentent mes améliorations sur le script pour prendre en compte le type de format principalement.

Pour commencer, je déclare une variable « \$format » qui récupérera le choix du format de l'utilisateur et la variable « \$commandeJindal » qui lui récupère le numéro de commande saisi dans le cas de JINDAL (étape 1). Ensuite pour récupérer ces valeurs, elles sont passées en GET dans l'url pointant vers ce fichier PHP. Donc en utilisant \$\_GET on récupère les informations passées dans l'url. Je rajoute donc

cette action pour récupérer le format ainsi que le numéro de commande dans le cas d'un format Fournisseur pour JINDAL (étape 2).

Dans un second temps à la suite de l'initialisation des variables pour récupérer les données, j'effectue une condition selon le format. Si le format est égal à « 1 » donc standard alors les variables initialisées dans un tableau avec la méthode array() au-dessus iront chercher les données obligatoirement (étape 3). La position des données à aller chercher dans le fichier CSV est exprimée par le numéro entre crochet. De plus, le programme lit le fichier CSV ligne par ligne. En résumé (voir annexe 4) :

- \$commande récupère les données de la colonne Customer Order à la position [4] dans un fichier standard ;
- \$caisse récupère les données de la colonne Case à la position [6] dans un fichier standard ;
- \$a récupère les données de la colonne Coil Size à la position [10] dans un fichier standard ;
- \$coulee récupère les données de la colonne Heat à la position [1] dans un fichier standard ;
- \$lot récupère les données de la colonne Lot à la position [0] dans un fichier standard ;
- \$bande récupère les données de la colonne Coil à la position [7] dans un fichier standard ;
- \$brin récupère les données de la colonne Strand à la position [8] dans un fichier standard ;
- \$poids récupère les données de la colonne Weight à la position [9] dans un fichier standard ;
- Si la colonne Master Sample contient la valeur 'X' à la position [11], \$ms prend la valeur « 1 ».

Cependant, un traitement est à effectuer différemment selon les fournisseurs car dans la base de données les lots et bobines ne sont pas composés de la même façon selon les fournisseurs. La plus commune est d'utiliser la coulée, le lot pour créer le lot fournisseur et la bande, le brin pour créer la bobine fournisseur. Mais chez les fournisseurs Hitachi, JINDAL et Bic Violex Hitachi ce traitement est différent comme le montre la table Fournisseur de la base de données Lames\_SS2L :

FournisseurId	FournisseurNom	FournisseurExempleLot	FournisseurExempleBobine	FournisseurFormat
1	Hitachi	Coulée	Lot ; Bande ; Brin	0
2	Sandvik	Coulée ; Lot	Bande ; Brin	1
3	Outokumpu	Coulée ; Lot	Bande ; Brin	0
4	Uddeholm	Coulée ; Lot	Bande ; Brin	0
5	JINDAL	Coulée ; Lot	- ; Brin	1
6	Voestalpine	Coulée ; Lot	Bande ; Brin	0
7	Bic Violex Hitachi	Coulée	Lot ; Bande ; Brin	0
8	Bic Violex Sandvik	Coulée ; Lot	Bande ; Brin	0

Dans ma condition il suffit alors de composer les variables « \$ref\_four\_lot » et « \$ref\_four\_bobine » différemment selon l'id d'Hitachi, de Bic Violex Hitachi et JINDAL. Enfin je fais la gestion de l'acier car la description de l'acier dans la base de données n'est pas la même que le fournisseur.

Pour finir le « else » correspond au choix du format « fournisseur ». Ainsi en déclarant un switch sur l'id du fournisseur passé, je peux traiter le format de SANDVIK et de JINDAL. On peut remarquer que pour SANDVIK le placement des informations entre crochets est la même que dans le format Standard car comme dit dans la genèse du projet, le format Standard est calqué sur le format SANDVIK. Donc dans

le cas où l'id du fournisseur est égal à « 2 » pour SANDVIK, les informations sont récupérées avec un ordre précis de récupération de données entre crochets.

Sinon dans le cas où l'id du fournisseur est égal à « 5 » pour JINDAL le positionnement des informations des colonnes à récupérer est différent. De plus, on passe dans la variable « \$commande », la variable saisie par l'utilisateur dans le script BRISE et stockée dans la variable « \$commandeJindal » (voir annexe 6).

Et voilà maintenant le programme est capable de prendre en compte le format ainsi que les différents fournisseurs.

## V. DESCRIPTION DE LA MISSION « AJOUT D'UNE FONCTIONNALITE DANS AP2M »

### 1. Genèse du projet

Ce projet est à l'initiative de Michaël POUETTRE (Assembly Production Manager) pour la partie assemblage de l'usine. Il consiste à ajouter une fonctionnalité dans BicRWeb dans les différents groupes de machines d'assemblage dans AP2M (Assembly Production Management Modernization). Cette fonctionnalité permettra d'envoyer un mail lorsqu'une machine de l'assemblage est en arrêt à cause d'un manque de matière. Ce mail sera envoyé à Michael POUETTRE et Matthieu LEGLAY (Project Manager) et contiendra les informations suivantes : le numéro de la machine, l'heure du début d'arrêt de la machine, le temps d'arrêt total en minutes ainsi que la matière manquante. Ces données sont alors saisies par l'employé de la machine. A terme, cela permettra de connaître qui est en cause dans l'arrêt de la machine.

Monsieur POUETTRE nous a fourni la description du fonctionnement de la nouvelle fonctionnalité :

1. Un bouton « Déclaration manque matière »
2. Quand la personne clique sur ce bouton une nouvelle page apparaît avec la question : « Votre machine a-t-elle été arrêtée ? »  
Les opérateurs doivent cliquer sur une case au choix :
  - d. « NON » : le message affiche « Il n'y a donc pas de déclaration à faire », et un bouton « Retour » permet de revenir à la page d'accueil AP2M de la machine
  - e. « OUI » : on passe alors au point 3
3. « Quel est le numéro de machine ? » : une liste de choix avec les machines. Idéalement peut-on avoir seulement les machines du groupe sur lequel est le PC ? La personne sélectionne la machine. On passe au point suivant
4. « Quelle heure était-il ? Et combien de temps la machine a-t-elle été arrêtée ? » : la personne doit taper l'heure du début d'arrêt et le temps d'arrêt en minutes
5. « Quelle matière manquait sur votre machine ? » : la personne a une liste de choix avec les éléments suivants : Manche, Guard, Cover, Protecteur, Carton, Film, Autre
6. Ensuite il y a une page de synthèse des éléments choisis par l'opérateur, avec une demande de validation
7. Quand l'opérateur valide, un mail est envoyé à Matthieu et moi

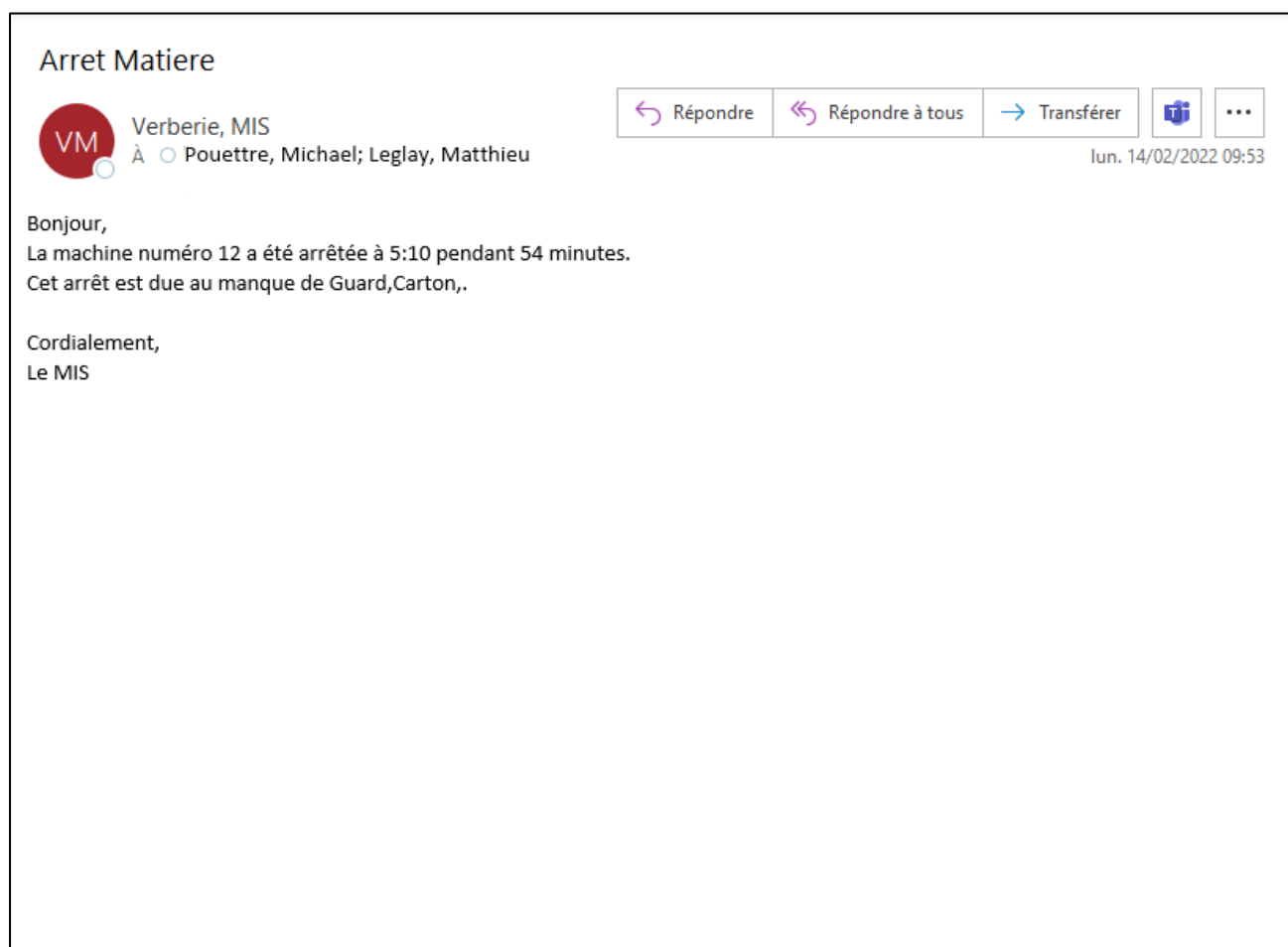
### 2. Planning

Ce projet m'a été donné l'avant-dernière semaine de stage, et sa réalisation était estimée à environ 1 semaine :

	Semaines :						
Tâches :	1	2	3	4	5	6	7
Réalisation du script							
Présentation, formation & mise en prod							
Rédaction du rapport de stage							

### 3. Résultat final

Voici le résultat final du projet, un mail envoyé à Michael POUETTRE et Matthieu LEGLAY avec les informations utiles sur l'arrêt de la machine d'assemblage.



### 4. Présentation du script

Ce script est réalisé avec BRISE. Pour rappel, **BRISE** pour **Bic Rasoirs Interrogation Saisie et Enregistrement** est une bibliothèque PHP pour BicRWeb, dédiée à la définition de formulaire (cf. rapport de l'année dernière). Tout d'abord, j'ai dû créer 2 fichiers, un fichier de type PHP pour faire appel à la bibliothèque BRISE et un autre fichier XML permettant l'utilisation des balises XML faisant référence à la bibliothèque. Mon premier fichier s'appelle **AP2M\_appel\_manque\_matiere.php**. Ce script en PHP contient seulement l'appel du BRISE dans la librairie BicRWeb et son lancement dans mon autre fichier XML permettant ainsi de l'utiliser :

```
<?php

//// Librairie BicRWeb ////
require_once '..\..\lib-php\bfp.inc.php';
require_once REP_BRISE.'brise.php';

//// Lancer la consultation
brise('AP2M_appel_manque_matiere.xml');
```

Et mon autre fichier nommé **AP2M\_appel\_manque\_matiere.xml** de type XML contenant le script BRISE. Les cadres sur les captures d'écran correspondent aux différents points du cahier des charges :

```

<?xml version='1.0' encoding='utf-8' ?>
<brise titre="AP2M - Déclaration Manque Matière" debug='non' debut='debut' >

  <alias>
    <externe nomFichier='../config.xml' xpath='/config/connexion' />
  </alias>

  <source nom='lesMachines'>
    <connexion type='alias' nom='master' >
      SELECT CASE WHEN LEFT('$Machine$',2) #DIF#'00' THEN LEFT('$Machine$',2) ELSE '-' END AS M
      UNION(
      SELECT CASE WHEN RIGHT('$Machine$',2) #DIF#'00' THEN RIGHT('$Machine$',2) ELSE '-' END AS M
      )
    </connexion>
  </source>

  <var nom='erreur' />

  <var nom='numMachine' />
  <var nom='horaireTempsArret' />
  <var nom='heureTempsArret' />
  <var nom='minuteTempsArret' />
  <var nom='minuteTotalArret' />
  <var nom='matiere' />
  <var nom='lesMatiere' />
  <var nom='Machine' type='auto' valParam='Machine' />
  <var nom='envoi' type='texte' val='' />
  <var nom='Objet' val='Arret Matière' />

  <var nom='Destinataire' val='michael.pouettre@bicworld.com,matthieu.leglay@bicworld.com' />
  <var nom='Expediteur' val='mis.verberie@bicworld.com' />

  <etape nom='debut'>
    <affichage val="Votre machine a-t-elle été arrêtée ?" aspect='gras' />
    <bouton nom='NON' label='NON' etape='aucuneDeclaration' />
    <bouton nom='OUI' label='OUI' etape='saisieNumeroMachine' />
  </etape>

  <etape nom='aucuneDeclaration'>
    <affichage val="Il n'y a donc pas de déclaration à faire" aspect='erreur' />
    <affichage val="Veuillez fermer la page !" aspect='gras' />
  </etape>

```



```
<etape nom='saisieNumeroMachine'>
  <affichage type='texte' val='$erreur$' test='$erreur$' #DIF# '' aspect='erreur grand gras' />
  <affichage val="Quel est le numéro de machine ?" aspect='gras' />
  <chargement source='lesMachines' />
  <saisie var='numMachine' type='liste' label="Numéro de machine" >
    <champ nom='' cle='' />
    <champSrc source='lesMachines' nom='M' cle='M' />
  </saisie>
  <bouton nom='Retour' label='Retour' etape='debut' />
  <bouton nom='Suivant' label='Suivant' etape='verifMachine' />
</etape>

<etape nom='verifMachine'>
  <calcul var='erreur' val='' />
  <calcul var='erreur' test="$numMachine$" #EG# "" val='Sélectionner un numéro de machine !' />
  <calcul var='erreur' test="$numMachine$" #EG# "-" val='Ce numéro de machine n est pas valable !' />
  <saut test='$erreur$' #EG# "" etape='saisieHoraire' />
  <saut test='$erreur$' #DIF# "" etape='saisieNumeroMachine' />
</etape>
```

```
<etape nom='saisieHoraire'>
  <affichage type='texte' val='$erreur$' test='$erreur$' #DIF# '' aspect='erreur grand gras' />
  <affichage val="Quel heure était-il ?" aspect='gras' />
  <saisie var='heureTempsArret' type='ligne' label='Heure début arrêt (HH)' />
  <saisie var='minuteTempsArret' type='ligne' label='Minute début arrêt (mm)' />

  <affichage val="Et combien de temps la machine a été arrêtée ?" aspect='gras' />
  <saisie var='minuteTotalArret' type='ligne' label='Temps total arrêt (en minutes)' />
  <bouton nom='Retour' label='Retour' etape='saisieNumeroMachine' />
  <bouton nom='Suivant' label='Suivant' etape='verifHoraire' />
</etape>

<etape nom='verifHoraire'>
  <calcul var='erreur' val='' />
  <calcul var='erreur' test="$minuteTempsArret$" #EG# "" val='Saisir minute de début !' />
  <calcul var='erreur' test="$heureTempsArret$" #EG# "" val='Saisir heure de début !' />
  <calcul var='erreur' test="$heureTempsArret$" #SUP# "24" val='Saisir heure de début compris entre 0 et 23 !' />
  <calcul var='erreur' test="$minuteTempsArret$" #SUP# "59" val='Saisir minute de début compris entre 0 et 59 !' />
  <calcul var='erreur' test="$minuteTotalArret$" #EG# "" val='Saisir un temps arrêt en minutes !' />
  <calcul var='erreur' test='!is_numeric("$minuteTotalArret$")' val="Saisir un temps arrêt numérique en minutes !" />
  <saut test='$erreur$' #EG# "" etape='saisieMatiere' />
  <saut test='$erreur$' #DIF# "" etape='saisieHoraire' />
</etape>
```

```
<etape nom='saisieMatiere'>
  <affichage type='texte' val='$erreur$' test='$erreur$' #DIF# '' aspect='erreur grand gras' />
  <affichage test="$lesMatiere$" #DIF# "" val='Matiere manquante : $lesMatiere$' aspect='gras fondbleu' />
  <affichage val="Quelle matière manquait sur votre machine ?" aspect='gras' />
  <saisie var='matiere' type='liste' label="Matière" >
    <champ nom='' cle='' />
    <champ nom='Manche' cle='Manche' />
    <champ nom='Guard' cle='Guard' />
    <champ nom='Cover' cle='Cover' />
    <champ nom='Protecteur' cle='Protecteur' />
    <champ nom='Carton' cle='Carton' />
    <champ nom='Film' cle='Film' />
    <champ nom='Autre' cle='Autre' />
  </saisie>
  <bouton nom='Retour' label='Retour' etape='saisieHoraire' />
  <bouton nom='Suivant' label='Suivant' etape='verifMatiere' />
</etape>

<etape nom='verifMatiere'>
  <calcul var='erreur' val='' />
  <calcul var='erreur' test="$Matiere$" #EG# "" val='Sélectionner une matière !' />
  <calcul var='lesMatiere$' valCalc="$matiere$"."$lesMatiere$" />
  <saut test='$erreur$' #EG# "" etape='autreSaisieMatiere' />
  <saut test='$erreur$' #DIF# "" etape='saisieMatiere' />
</etape>

<etape nom='autreSaisieMatiere'>
  <affichage val="Voulez-vous saisir une autre matière ?" aspect='gras' />
  <bouton nom='NON' label='NON' etape='recapitulatif' />
  <bouton nom='OUI' label='OUI' etape='saisieMatiere' />
</etape>
```


```
<etape nom='recapitulatif'>
  <calcul var='horaireTempsArret' valCalc="$heureTempsArret$"."H"."$minuteTempsArret$" />
  <affichage val="Numéro de machine : $numMachine$" aspect='gras fondbleu' />
  <affichage val="Heure début arrêt : $horaireTempsArret$" aspect='gras fondbleu' />
  <affichage val="Temps arrêt (en minutes) : $minuteTotalArret$" aspect='gras fondbleu' />
  <affichage val="Matière manquante : $lesMatiere$" aspect='gras fondbleu' />
  <bouton nom='Retour' label='Retour' etape='autreSaisieMatiere' />
  <bouton nom='Valider' label='Valider' etape='Envoi_email' />
</etape>
```

```
<etape nom='Envoi_email'>
  <calcul var='envoi' valCalc="bfp_envoyer_email(array(
    'smtp'=>'eusmtpl.eu.bicworld.com',
    'message'=>'
      Bonjour,&lt;/br&gt;
      La machine numéro $numMachine$ a été arrêtée à $horaireTempsArret$ pendant $minuteTotalArret$ minutes.&lt;/br&gt;
      Cet arrêt est dû au manque de $lesMatiere$.
      &lt;/br&gt;
      &lt;/br&gt;
      Cordialement,&lt;/br&gt;
      Le MIS
    ',
    'objet'=>'$Objet$',
    'email_expéditeur'=>'$Expéditeur$',
    'expéditeur'=>'$Expéditeur$',
    'mail_reponse'=>'$Destinataire$',
    'destinataire'=>'$Destinataire$'
  ),null)
  "/>
  <saut etape='Fin' />
</etape>

<etape nom='Fin' >
  <affichage val="E-mail envoyé" />
</etape>

</brise>
```

Pour commencer avant de parler du script, il faut créer le bouton « Déclaration manque matière » (point numéro 1 du cahier des charges) dans le menu AP2M de toutes les machines d'assemblage sauf les machines de Zone US et 91-92. Dans le menu AP2M, les machines sont regroupées par paires sauf dans le cas de la machine 16 :

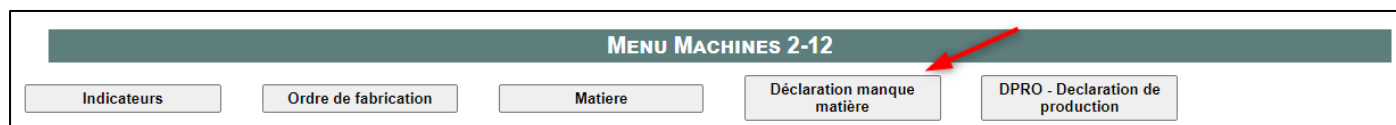
 AP2M				
GROUPES				
10-11	02-12	05-09	03-06	01-04
07-08	13-14	16	21-22	23-24
Zone US	31-32	91-92		

Ce bouton devra être donc rajouté dans chaque groupe de machines. Pour ce faire, il faut se rendre dans le fichier index de chacun des groupes de machines, par exemple ici nous allons prendre le fichier du groupe de machines 02-12. Ce fichier configure le menu à afficher du groupe de machines donc pour afficher notre bouton il suffit de le créer dans ce script. Je configure ainsi le bouton pour pointer, lors du clic, sur mon script BRISE :

```
<?xml version='1.0' encoding='utf-8'?>
<brame titre="AP2M" >

<affichage val="Menu Machines 2-12" />
<bouton nom="Indicateurs" lien="Communication/AP2M_visu_Indicateur_v2.php?%23RETOUR=%23close&Machine=0212" dest='ext' />
<bouton nom="Ordre de fabrication" lien="Communication/AP2M_visu_Ofs.php?%23RETOUR=%23close&Machine=0212" dest='ext' />
<bouton nom="Matiere" lien="Communication/AP2M_visu_matiere.php?%23RETOUR=%23close&Machine=0212" dest='ext' />
<bouton nom="Déclaration manque matière" lien="Communication/AP2M_appel_manque_matiere.php?%23RETOUR=%23close&Machine=0212" dest='ext' />
<bouton nom="DPRO - Declaration de production" lien=".../SiDERA/scripts/FormBRISE/SiDERA_DPRO.php?%23ASPECT=fond_vert_b&Machine=0212" dest='ext' />
```

De plus, je passe en paramètre dans l'url pointant vers le fichier BRISE la valeur 0212 correspondant au groupe dans lequel se trouve le bouton. Cette valeur nous servira lors du point numéro 3 du cahier des charges. Voici le résultat de ce script :



Le bouton mis en place, passons sur la partie script BRISE. Pour effectuer le deuxième point du cahier des charges, j'utilise BRISE. J'initialise mon script pour qu'il commence à l'étape « début » et créer des variables. Dans cette étape je demande si la machine a été arrêtée avec deux boutons, si la réponse est « OUI » on passe alors au point 3 du cahier des charges avec l'étape « saisieNumeroMachine » et si la réponse est « NON » le bouton amène alors le script sur l'étape « aucuneDéclaration » qui invite l'utilisateur à fermer la fenêtre de la fonctionnalité.

Voici l'étape « début » du script :

AP2M - DÉCLARATION MANQUE MATIÈRE	
Votre machine a-t-elle été arrêtée ?	
<input type="button" value="NON"/>	<input type="button" value="OUI"/>

Si la réponse choisie est « NON » :

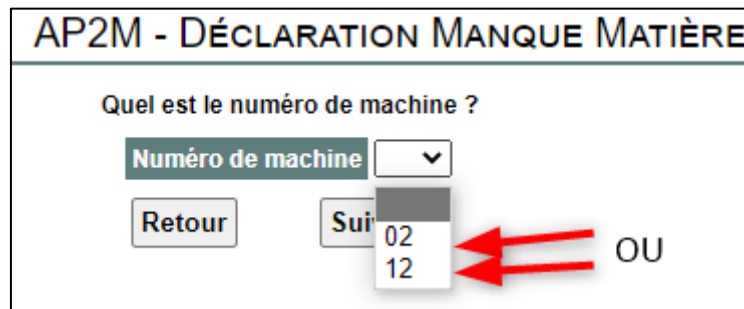
AP2M - DÉCLARATION MANQUE MATIÈRE
<b>Il n'y a donc pas de déclaration à faire</b>
Veuillez fermer la page !

Pour continuer le script, prenons l'exemple d'un utilisateur qui choisit « OUI ». En faisant cela il se retrouve à l'étape « saisieNumeroMachine » qui demande de choisir le numéro de la machine du groupe. Dans notre cas, soit la machine 02 soit la machine 12. Ce paramètre est récupéré dans en GET dans l'url avec valParam puis stocké dans la variable Machine (0212). Ensuite pour pouvoir récupérer les numéros des deux machines distinctes, j'utilise une requête SQL qui récupère les deux premiers caractères de gauche la variable Machine (dans notre cas 02) et si celle-ci est égale à 00, on remplace par un tiret. J'effectue de nouveau cette requête mais ici en allant chercher les deux derniers caractères de droite (dans notre cas 12). Cette condition dans la requête SQL avec le cas d'une chaîne de caractères 00 est important dans le groupe de machine 16 car celle-ci est envoyée dans l'url avec la valeur 0016 ainsi cela permet à l'utilisateur de ne pas se tromper lors de la saisie des informations de la fonctionnalité.

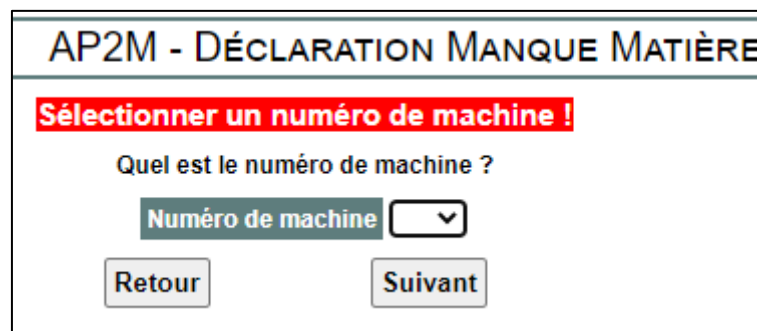
Pour revenir à l'étape « saisieNumeroMachine », j'affiche les numéros de machine dans une liste avec un texte et deux boutons. L'un permettant de revenir à l'étape précédente et l'autre permettant d'avancer dans le script en vérifiant la sélection du numéro de machine dans l'étape « verifMachine ».

Par exemple, cette étape vérifie si l'utilisateur a sélectionné un numéro ou encore si le numéro est égal au tiret. Si c'est le cas un message d'erreur apparaît et empêche l'utilisateur de continuer.

Voici l'étape « saisieNumeroMachine » du script :

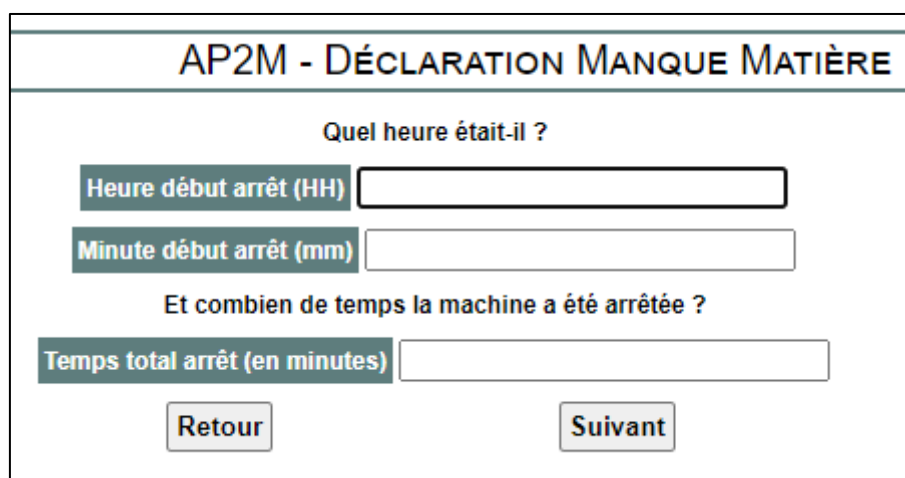


Par exemple, si l'utilisateur commet une erreur en ne sélectionnant rien cela lance l'étape « verifMachine » :



L'étape de vérification de la saisie machine effectué, l'utilisateur se retrouve automatiquement sur l'étape « saisieHoraire » qui consiste à saisir l'heure du début d'arrêt de la machine ainsi que le temps d'arrêt total en minutes. Plus deux boutons, l'un menant à l'étape précédente et l'autre menant à l'étape « verifSaisieHoraire » pour vérifier la saisie.

Voici comment se présente l'étape « saisieHoraire » :



En cliquant sur « Suivant » le script se retrouve alors à « verifSaisieHoraire », vérifiant si les données ont été saisies, si les champs sont bien remplis avec des valeurs numériques, si l'heure ne dépasse pas 23 et si les minutes ne dépassent pas 60. Par exemple, voici l'étape en action si l'utilisateur commet une erreur en ajoutant des lettres dans le champ « Temps total arrêt (en minutes) » :

### AP2M - DÉCLARATION MANQUE MATIÈRE

**Saisir un temps arrêt numérique en minutes !**

Quel heure était-il ?

Heure début arrêt (HH)

Minute début arrêt (mm)

Et combien de temps la machine a été arrêtée ?

Temps total arrêt (en minutes)

Retour
Suivant

Suite à l'étape de vérification de la saisie des horaires, l'utilisateur se retrouve automatiquement sur l'étape « saisieMatiere » qui consiste en la sélection du ou des matières manquantes ayant provoqué l'arrêt de la machine. Dans cette étape l'utilisateur peut choisir la matière dans une liste prédéfinie avec deux boutons, l'un menant à l'étape antérieure et l'autre à l'étape « verifMatiere ». Une fois la matière sélectionnée et un clic sur le bouton « Suivant », le script se trouve sur l'étape « verifMatiere » qui vérifie si l'utilisateur a bien sélectionné une matière et concatène la valeur de la matière dans une autre variable dans le cas où dans la prochaine étape l'utilisateur décide d'ajouter une autre matière manquante. Ainsi la variable pourra stocker toutes les matières sélectionnées.

Voici l'étape « saisieMatiere » du script :

### AP2M - DÉCLARATION MANQUE MATIÈRE

Quelle matière manquait sur votre machine ?

Matière

Retour Suivant

EB - LIB:1.8 - PHP:5.6.15 - C
20:18:21 Mem:1792 KIB

Et voici le cas où l'utilisateur commet une erreur en ne sélectionnant aucune valeur :

AP2M - DÉCLARATION MANQUE MATIÈRE	
<b>Sélectionner une matière !</b>	
Quelle matière manquait sur votre machine ?	
Matière	<input type="text"/>
<input type="button" value="Retour"/>	<input type="button" value="Suivant"/>

A la suite de l'étape de vérification de la matière, il est proposé automatiquement à l'utilisateur de saisir une autre matière dans le cas où plusieurs matières manquantes ont causé l'arrêt de la machine :

AP2M - DÉCLARATION MANQUE MATIÈRE	
Voulez-vous saisir une autre matière ?	
<input type="button" value="NON"/>	<input type="button" value="OUI"/>

Si l'utilisateur choisit « NON » le script amène à l'étape « recapitulatif » récapitulant comme son nom l'indique les données saisies par l'utilisateur dans cette fonctionnalité. Si son choix est « OUI », alors le script retourne à l'étape précédente de la sélection de matière pour que l'utilisateur choisisse une autre matière. En outre, la concaténation permettra alors de stocker la deuxième matière puis la troisième, etc. Dans tous les cas, l'utilisateur se retrouvera par cliquer sur « NON » et se retrouver à l'étape « recapitulatif ». Cette étape affiche alors les variables saisies par l'utilisateur :

AP2M - DÉCLARATION MANQUE MATIÈRE	
Numéro de machine : 2	
Heure début arrêt : 5H10	
Temps arrêt (en minutes) : 51	
Matière manquante : Protecteur,Manche,	
<input type="button" value="Retour"/>	<input type="button" value="Valider"/>

Après avoir constaté l'exactitude des informations saisies, l'utilisateur peut alors cliquer sur « Valider » qui lance l'étape « Envoie\_email ». Cette étape utilise la méthode bfp\_envoyer\_email (annexe) qui envoie un mail en renseignant le serveur smtp, le message, l'objet, le courriel expéditeur, l'expéditeur, le mail réponse et le destinataire. Les variables créées au début du script contiennent les informations des destinataires ainsi que l'objet du mail et dans la partie message on trouve les informations saisies par l'utilisateur. Pour finir sur ce script, j'affiche un message attestant l'envoi du mail :

AP2M - DÉCLARATION MANQUE MATIÈRE
E-mail envoyé



## VI. BILAN

Ce projet m'a beaucoup apporté sur l'aspect technique et humain. Tout d'abord sur l'aspect technique, j'ai pu approfondir mes connaissances en SQL, PHP et également en réseau. Augmentant ainsi mon portefeuille de compétences et ma curiosité. J'ai pu découvrir la bibliothèque de classe FPDF ainsi que le logiciel PuTTY. De plus, durant ce projet, j'ai pu consolider mes bases en PHP et apprendre de nouvelles commandes comme celles de type ODBC,...

Ensuite sur l'aspect humain, j'ai pu découvrir le travail en équipe sur un projet dans une entreprise et comment dans un projet en équipe le travail se répartit,... Une équipe permet d'élargir sa pensée et d'utiliser des procédés auxquels nous n'aurions jamais pensé. Par exemple pour la configuration du logiciel d'impression MakWare, après plusieurs essais mon collègue Mathis a eu l'idée de vérifier sur l'un des pc possédant ce logiciel si la clé produite s'y trouvait et c'était le cas. Cette idée nous débloqua ainsi de ce problème.

Le fait de savoir que son projet va servir à l'entreprise ajoute une plus-value ainsi qu'une très grande satisfaction pour son travail. J'ai appris également que le travail personnel et l'autonomie permettent d'augmenter ses compétences personnelles. Enfin, j'ai pu découvrir ce que c'était d'être chef de projet temporairement et à faible échelle en concevant plusieurs projets qui sont à chaque fois présentés aux clients par le biais de réunions. J'ai également pu apprendre que le besoin du client est le plus important et que parfois le client exprime d'autres besoins non contenus dans le cahier des charges initial. Également, le rapport humain est très important au sein du monde du travail. Lors de nos interventions comprendre correctement le problème du client est indispensable et cela passe par la communication. Ce point-là m'a beaucoup plu et m'a permis d'améliorer mon contact humain ainsi que mon élocution orale.

Pour finir, les plus grandes difficultés de ce stage ont été le bug du chargement d'une image installé sur l'un des pc fixes à configurer. Cette image permettant la configuration initiale du pc plus rapidement. Ce bug nous a empêché de terminer la configuration du pc et nous a handicapés d'un pc en moins car avec le bug de l'image, il faut un compte administrateur pour se connecter au pc. Compte que nous n'avons pas. La seule solution est alors de se tourner vers le service client DELL et le siège social Clichy. Ces procédures sont assez longues à réaliser. Et l'autre fut l'absence durant 1 semaine de mon collègue Mathis DELHALLE. Son absence a ralenti le projet de configuration des pc qui est possible seulement en présentiel mais j'ai pu m'adapter et tenir au courant mon collègue de mon évolution pour qu'il soit ainsi à jour au moment où il revient. Là encore l'esprit d'équipe est très important.

## VII. REMERCIEMENT

Pour commencer, je tiens à remercier Pascal HENNION de m'avoir accepté de nouveau au sein du MIS pour effectuer mon stage, de tous ses bons conseils précieux sur le monde de l'informatique, sur mon orientation scolaire et de sa bonne humeur ainsi que l'accueil offert tout au long de mon stage. Ce fut une chance et un honneur de retravailler au sein de cette formidable équipe dans l'entreprise BIC Rasoirs.

Ensuite, je voudrais tout particulièrement remercier Jérémy BRASSART pour m'avoir accompagné tout au long de ce stage. Le remercier pour m'avoir trouvé des missions toujours cohérentes avec mon cursus scolaire et ayant une réelle utilité au sein de l'entreprise. Ainsi que de m'avoir appris de nouvelles choses sur l'informatique consolidant mes bases en développement et réseau. Et enfin d'avoir pris le temps de répondre à toutes mes questions et de m'avoir guidé sur mon orientation scolaire.

Je souhaite également remercier mon collègue Mathis DELHALLE pour m'avoir épaulé durant ce stage sur les différentes missions et pour sa joie de vivre.

Enfin, je tiens également à remercier tous les collaborateurs de BIC Rasoirs pour leur accueil, leur gentillesse et leur humour, qui ont rendu mon stage très agréable.

## VIII. ANNEXES

### Annexe 1 : Fichier liste\_incendie.php initiale

```

<?php
require('..\lib-php\fpdf\fpdf.php');

class PDF extends FPDF
{
    private $jour;

    function __construct($jour) {
        parent::__construct();
        $this->jour = $jour;
    }

    //fonction Header() pour traiter les en-têtes
    function Header()
    {
        // Pied de page
        function Footer()
        {
        }

    function mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete) {

    function mise_en_page_absent($pdf,$success,$en_tete,$num_tableau_associe_entete) {

    /*
    -----
    Programme Principale-----
    -----
    */

    // Preparation des donnees
    $debug=false;
    $size='';
    $password='';
    $server='';
    $database='etemplation2';
    $driver='SQL Server';
    $query = "
    SELECT
        HOPEMPL.Nom,
        HOPEMPL.Prenom,
        CONVERT (VARCHAR(16), DATEADD(second, CONVERT (INTEGER, LEFT (Heure, 2)) *3600+CONVERT (INTEGER, RIGHT (Heure, 2)) *60, DAT), 121) AS Date_Heure_Bipage2,
        CASE
            WHEN SUBSTRING(HOPEMPL.Nom, 1, 1) BETWEEN 'A' AND 'D' THEN 1
            WHEN SUBSTRING(HOPEMPL.Nom, 1, 1) BETWEEN 'E' AND 'M' THEN 2
            WHEN SUBSTRING(HOPEMPL.Nom, 1, 1) BETWEEN 'N' AND 'Z' THEN 3
            ELSE 99 END NumTableau
    FROM HOPOIN
    INNER JOIN HOPEMPL ON HOPOIN.MATRI=HOPEMPL.MATRI
    INNER JOIN (
        SELECT
            HOPOIN.MATRI AS M,
            MAX (DATEADD(second, CONVERT (INTEGER, LEFT (Heure, 2)) *3600+CONVERT (INTEGER, RIGHT (Heure, 2)) *60, DAT)) AS Date_Heure_Bipage
        FROM HOPOIN
        GROUP BY MATRI, HOPOIN.MATRI
    ) S ON M=HOPOIN.MATRI AND Date_Heure_Bipage=DATEADD(second, CONVERT (INTEGER, LEFT (Heure, 2)) *3600+CONVERT (INTEGER, RIGHT (Heure, 2)) *60, DAT)
    WHERE SENS='0'
    AND DATEADD(second, CONVERT (INTEGER, LEFT (Heure, 2)) *3600+CONVERT (INTEGER, RIGHT (Heure, 2)) *60, DAT) >= DateAdd(day, -2, getdate())
    ORDER BY HOPEMPL.Nom, HOPEMPL.Prenom, Date_Heure_Bipage2
    ";

    /* Connexion BDD */
    $connection = odbc_connect("Driver={$driver};Server={$server};Database={$database}; user={$user};password={$password};SQL_CUR_USE_ODBC");
    $success = odbc_exec($connection,$query);
    if($connection != false) {
        if($debug) {
            echo 'Connexion reussie<br>';
            var_dump($success);
        }

        // Fin des preparations des donnees

        // Instanciation de la classe dérivée
        $today = date('Y-m-d H:i');
        $en_tete="Liste d'appel A-D";
        $num_tableau_associe_entete=1;
        $pdf = new PDF($today);
        $pdf->AliasNbPages();
        //1 ere liste
        $pdf->AddPage();
        $pdf->mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete);
        //liste absent AD
        $pdf->AddPage();
        $en_tete="Liste des personnes à transmettre au gardien de A-D";
        $success = odbc_exec($connection,$query);
        $num_tableau_associe_entete=1;
        $pdf->mise_en_page_absent($pdf,$success,$en_tete,$num_tableau_associe_entete);

        //2" liste
        $pdf->AddPage();
        $en_tete="Liste d'appel E-M";
        $success = odbc_exec($connection,$query);
        $num_tableau_associe_entete=2;
        $pdf->mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete);
        //liste d'absent EM
        $pdf->AddPage();
        $en_tete="Liste des personnes à transmettre au gardien de E-M";
        $success = odbc_exec($connection,$query);
        $num_tableau_associe_entete=2;
        $pdf->mise_en_page_absent($pdf,$success,$en_tete,$num_tableau_associe_entete);

        //3" liste
        $pdf->AddPage();
        $en_tete="Liste d'appel N-Z";
        $success = odbc_exec($connection,$query);
        $num_tableau_associe_entete=3;
        $pdf->mise_en_page($pdf,$success,$en_tete,$num_tableau_associe_entete);
        //liste absent NZ
        $pdf->AddPage();
        $en_tete="Liste des personnes à transmettre au gardien de N-Z";
        $success = odbc_exec($connection,$query);
        $num_tableau_associe_entete=3;
        $pdf->mise_en_page_absent($pdf,$success,$en_tete,$num_tableau_associe_entete);

        odbc_close($connection);
        if ($debug)
        $file='.....\Bicwebdata\horloguartz\';
        $pdf->Output($file."Liste_incendie.pdf",'F');
    }
    else {
        echo 'Connexion non aboutie';
    }
}

```

- 1 Présentation du fichier PDF.php
- 2 Présentation du fichier connexionBDD.xml
- 3 Présentation du fichier liste\_incendie.php

Annexe 2 : Fichier verification.php

```
<?php
$dossier='..\..\..\..\Bicwebdata\horoquartz\Liste_incendie.pdf';
$destination='\\VERW0668\Horoquartz\Liste_incendie.pdf';
// $destination='..\..\..\..\Bicwebdata\Liste_incendie.pdf';

require ('liste_incendie.php');
require ('..\lib-php\bfp_divers.inc.php');
if (file_exists($dossier)) {
    echo "Le fichier existe ";
    copy($dossier,$destination);
    if (file_exists($destination)){
        echo "et a été transmis";
    }else{
        echo "mais n'a pas etait transmis";
    }
}

//Envoi de mail
$today = date('Y-m-d H:i');
$envoie=''; // <= a remplir pour l'email d'envoi
$pieces_jointes=array($dossier);
$email_param=array('smtp'=>'',
    'message'=>'Bonjour,<br>Envoi de la liste incendie datant du '.$today,
    'objet'=>'Liste_incendie',
    'email_expediteur'=>'',
    'expediteur'=>'',
    'mail_reponse'=>$envoie,
    'destinataire'=>$envoie,
    'accuse_reception'=>$envoie
);

bfp_envoyer_email($email_param, $pieces_jointes);
} else {
    echo "Le fichier n'existe pas.";
}
-?>
```

Annexe 3 : Fichier Horoquartz\_generateur\_pdf.bat

```
:start

start ..\..\..\..\xampp\php\php.exe "verification.php"
timeout 900

GOTO start
```

Annexe 4 : Fichier CSV Standard mis au format Excel avec indication placement des données en référence avec le programme PHP.

[0] Lot	[1] Heat	Order	Pos	[4] Customer Order	Packing Slip	[6] Case	[7] Coil	[8] Strand	[9] Weight	[10] Coil Size	[11] Master Sample
15474	562628	502-41895		1 21000080 DU	4759756	5615393		1 4A	31.3	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615393		1 4A	31.3	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615394		1 15B	32.3	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615394		1 15B	32.3	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615394		1 16B	32.3	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615394		1 16B	32.3	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615393		1 6A	31.4	13,74 x 0,074	
15474	562628	502-41895		1 21000080 DU	4759756	5615393		1 6A	31.4	13,74 x 0,074	

Annexe 5 : Fichier CSV non Standard SANDVIK mis au format Excel

Lot	Heat	Order	Pos	Customer Or	Packing Slip	Case	Coil	Strand	Weight	Coil Size	Master Sample
15764	563832	502-44490		1 21000258 DU	586556	5623371		1 23A		32 13,74 x 0,074	
15764	563832	502-44490		1 21000258 DU	586556	5623372		1 5C		33,3 13,74 x 0,074	
15764	563832	502-44490		1 21000258 DU	586556	5623372		1 7C		33,3 13,74 x 0,074	
15764	563832	502-44490		1 21000258 DU	586556	5623371		1 2B		32,2 13,74 x 0,074	
15764	563832	502-44490		1 21000258 DU	586556	5623371		1 4B		32,2 13,74 x 0,074	
15764	563832	502-44490		1 21000258 DU	586556	5623372		1 8C		33,3 13,74 x 0,074	

Annexe 6 : Fichier CSV non Standard JINDAL mis au format Excel avec indication des colonnes du format standard.

Packing Slip	Lot	Strand	Heat	Coil Size	Coil Size	Weight	Master Sample	Case		
Sr. No.	Coil No.	Slit No.	Heat No.	Thickness	Width	Net Wt.	Joint	Drum No.	Pallet No.	Packet No.
1	S85766	4	2109J0263	0,076	22,1	47,45	M.S	1	126950	126950-1
2	S85766	3	2109J0263	0,076	22,1	47,45		1	126950	126950-1
3	S85766	2	2109J0263	0,076	22,1	47,4		1	126950	126950-1
4	S85766	1	2109J0263	0,076	22,1	47,42		1	126950	126950-1
5	S85765	10	2109J0263	0,076	22,1	47,52		1	126950	126950-1
6	S85765	11	2109J0263	0,076	22,1	47,46	M.S	1	126950	126950-1
7	S85766	11	2109J0263	0,076	22,1	47,51		1	126950	126950-1

## Annexe 7 : La fin du contenu du fichier Preparation\_auto\_packing\_list.php

```
//recherche de l'acier
$par['acier']=$acier;
$reqSql=bfp_query_depuis_xml($fichier_query,'query_acier_id',$par,true);
if($debug){
    echo 'Requete SQL "query_acier_id": '.BR.$reqSql.BR;
    $tabResultatReq=bfp_t2d_depuis_odbc($conn,$reqSql,true);
}else{
    $tabResultatReq=bfp_t2d_depuis_odbc($conn,$reqSql,true);
}

// recherche du type de lame
$par['PackingBobineAcierId']=$tabResultatReq[0]['acierid'];
$reqSql2=bfp_query_depuis_xml($fichier_query,'query_lame_id',$par,true);
if($debug){
    echo 'Requete SQL "query_lame_id": '.BR.$reqSql2.BR;
    $tabResultatReq2=bfp_t2d_depuis_odbc($conn,$reqSql2,true);
}else{
    $tabResultatReq2=bfp_t2d_depuis_odbc($conn,$reqSql2,true);
}
$par['PackingBobineLameId']=$tabResultatReq2[0]['LameId'];

if($debug){
    /* echo 'commande: '.$commande.BR;
    echo 'Caisse: '.$caisse.BR;
    echo 'acier: '.$acier.BR;
    echo 'coulee: '.$coulee.BR;
    echo 'lot: '.$lot.BR;
    echo 'bande: '.$bande.BR;
    echo 'brin: '.$brin.BR;
    echo 'poids: '.$poids.BR;
    echo 'ref_four_lot: '.$ref_four_lot.BR;
    echo 'ref_four_bobine: '.$ref_four_bobine.BR; */
    bfp_var_dump($par);
}

// Preparation insertion BDD
if($ref_four_lot<>'')
    $ref_four_bobine<>' '
    $fournisseur_id<>0
    $date_livraison<>' '
    $commande<>' '
    $poids<>0
    $acier<>' '
    $par['PackingBobineLameId']<>0
}

// Preparation de la caisse
//verification que la caisse n'existe pas dans la table caisses
$reqSql3=bfp_query_depuis_xml($fichier_query,'query_recherche_caisse',$par,true);
if($debug){
    echo 'Requete SQL "query_recherche_caisse": '.BR.$reqSql3.BR;
    $tabResultatReq3=bfp_t2d_depuis_odbc($conn,$reqSql3,true);
}else{
    $tabResultatReq3=bfp_t2d_depuis_odbc($conn,$reqSql3,true);
}
$nb_caisse_in_table_caisses=$tabResultatReq3[0]['NbCaisse'];

//verification que la caisse n'existe pas dans la table Packingcaisses
$reqSql4=bfp_query_depuis_xml($fichier_query,'query_recherche_packingcaisse',$par,true);
if($debug){
    echo 'Requete SQL "query_recherche_packingcaisse": '.BR.$reqSql4.BR;
    $tabResultatReq4=bfp_t2d_depuis_odbc($conn,$reqSql4,true);
}else{
    $tabResultatReq4=bfp_t2d_depuis_odbc($conn,$reqSql4,true);
}
$nb_caisse_in_table_packingcaisse=$tabResultatReq4[0]['NbCaisse'];

//0: insertion impossible
//1: insertion
//2 : update
$insert_ou_update_packingcaisse=0;

if($nb_caisse_in_table_caisses==0){
    if($nb_caisse_in_table_packingcaisse>0){
        $insert_ou_update_packingcaisse=2;
    }else{
        $insert_ou_update_packingcaisse=1;
    }
}
```

```

}
}else{
    $tab_caisse_present_dans_db[$par['PackingCaisseNumero']]=$par['PackingCaisseNumero'];
}

if($debug) echo 'insert_ou_update_packingcaisse '.$insert_ou_update_packingcaisse.BR;

// Preparation des Bobines
//verification que la caisse n'existe pas dans la table bobines
$reqSql7=bfp_query_depuis_xml($fichier_query,'query_recherche_bobine',$par,true);
if($debug){
    echo 'Requete SQL "query_recherche_bobine": '.$BR.$reqSql7.BR;
    $tabResultatReq7=bfp_t2d_depuis_odbc($conn,$reqSql7,true);
}else{
    $tabResultatReq7=bfp_t2d_depuis_odbc($conn,$reqSql7,true);
}
$nb_bobine_in_table_bobine=$tabResultatReq7[0]['NbBobines'];

$reqSql8=bfp_query_depuis_xml($fichier_query,'query_recherche_packingbobine',$par,true);
if($debug){
    echo 'Requete SQL "query recherche packingbobine": '.$BR.$reqSql8.BR;
}
//0: insertion impossible
//1: insertion
$insert_ou_update_packingbobine=0;
if($nb_bobine_in_table_bobine==0){
    if($nb_bobine_in_table_packingbobines==0){
        $insert_ou_update_packingbobine=1;
    }
    else{
        $tab_bobines_present_dans_db[$par['PackingCaisseFournisseurId'].'_'.$par['PackingBobineRefFourLot'].'_'.$par['PackingBobineRefFourBobine']]=$par['PackingCaisseFournisseurId'].'_'.$par['PackingBobineRefFourLot'].'_'.$par['PackingBobineRefFourBobine'];
    }
}else{
    $tab_bobines_present_dans_db[$par['PackingCaisseFournisseurId'].'_'.$par['PackingBobineRefFourLot'].'_'.$par['PackingBobineRefFourBobine']]=$par['PackingCaisseFournisseurId'].'_'.$par['PackingBobineRefFourLot'].'_'.$par['PackingBobineRefFourBobine'];
}
if($debug) echo 'insert_ou_update_packingbobine '.$insert_ou_update_packingbobine.BR;

//Gestion des bobines qui ne sont pas insérées
if($insert_ou_update_packingbobine==0){
    /* echo 'Erreur la bobine '.$ref_four_lot.' '.$ref_four_bobine.' ne peut pas être ajoutée !'. BR;
    echo $ligne.BR; */
    fputs($monfichier_erreur, $ligne);
    $flag_erreur=true;
}

//Insertion de la caisse si elle n'existe pas
if($insert_ou_update_packingcaisse==1){
    $reqSql5=bfp_query_depuis_xml($fichier_query,'query_insert_packingcaisse',$par);
    if($debug){
        echo 'Requete SQL "query_insert_packingcaisse": '.$BR.$reqSql5.BR;
        echo 'insertion de la caisse n°'.$caisse.' !'. BR;
    }
    else{
        odbc_exec($conn, $reqSql5);
    }
}
}
}

```

```

//Si la caisse existe et si la bobine est à inserer alors je mets à jour
//L'update de la caisse doit seulement servir à mettre le poids à jour
if($insert_ou_update_packingcaisse==1){
    if($insert_ou_update_packingbobine==1){
        $reqSql6=bfp_query_depuis_xml($fichier_query,'query_update_packingcaisse',$par);
        if($debug){
            echo 'Requete SQL "query_update_packingcaisse": '.$BR.$reqSql6.BR;
        }
        odbcc_exec($conn, $reqSql6);
    }
}

//On peut seulement inserer des bobines qui n'existe pas
if($insert_ou_update_packingbobine==1){
    $reqSql9=bfp_query_depuis_xml($fichier_query,'query_insert_packingbobine',$par);
    if($debug){
        echo 'Requete SQL "query_insert_packingbobine": '.$BR.$reqSql9.BR;
    }
    odbcc_exec($conn, $reqSql9);
}

}else{
    echo 'Impossible de traiter les lignes, toutes les informations ne sont pas renseignées'.BR;
    echo $ligne.BR;
    fputs($monfichier_erreur, $ligne);

    echo 'ref_four_lot '.$ref_four_lot;
    echo ' ;ref_four_bobine '.$ref_four_bobine;
    echo ' ;fournisseur_id '.$fournisseur_id;
    echo ' ;date_livraison '.$date_livraison;
    echo ' ;commande '.$commande;
    echo ' ;ms '.$ms;
    echo ' ;rpoids '.$poids;
    echo ' ;acier '.$acier;
    echo ' ;PackingBobineLameId '.$par['PackingBobineLameId'].BR; */
    $flag_erreur=true;
}

}
$ii++;
}
}
fclose($monfichier);
}

```



```

if(!empty($tab_caisses_present_dans_db)){
    asort($tab_caisses_present_dans_db);
    echo 'Attention Erreurs !!!!!!!'.BR;
    echo 'Recap des caisses qui sont déjà présentes dans la table caisse'.BR;
    echo '<table style="border: 1px solid #333">';
    echo '<tr><th>Caisses</th></tr>';

    foreach ($tab_caisses_present_dans_db as $v){
        echo '<tr>';
        echo '<td style="border: 1px solid #333">'. $v. '</td>';
        echo '</tr>';
    }
    echo '</table>';
}

if(!empty($tab_bobines_present_dans_db)){
    asort($tab_bobines_present_dans_db);
    echo 'Attention Erreurs !!!!!!!'.BR;
    echo 'Recap des bobines qui sont déjà présentes dans la table bobines ou packingbobines'.BR;

    echo '<table style="border: 1px solid #333">';
    echo '<tr><th style="border: 1px solid #333">Ref_Four_lot</th>';
    echo '<th style="border: 1px solid #333">Ref_Four_bobines</th></tr>';

    foreach ($tab_bobines_present_dans_db as $v){
        echo '<tr>';
        $tab_ref_bob_pres=explode("_",$v);

        echo '<td style="border: 1px solid #333">'. $tab_ref_bob_pres[0]. '</td>';
        echo '<td style="border: 1px solid #333">'. $tab_ref_bob_pres[1]. '</td>';
        echo '</tr>';
    }
    echo '</table>';
}

if(!$debug){
    //Backup du fichier
    copy ($emplacement_fichier_a_traiter.$nom_fichier_a_traiter.'.csv',$emplacement_backup.$nom_fichier_a_traiter.'.');
    unlink ($emplacement_fichier_a_traiter.$nom_fichier_a_traiter.'.csv');
}
}
}

}else{
    echo 'Dossier introuvable !!';
}

fclose($monfichier_erreur);
odbc_close($conn);

//Si aucune erreur je supprime le fichier creer
if($flag_erreur==false){
    unlink ($nom_fichier_erreur);
    //redirection pour voir la saisie
    echo '<meta http-equiv="Refresh" content="1;URL=VoirSaisie.php?fournisseurnom='.$fournisseur_nom.'">';
}

```

## Annexe 8 : Méthode bfp\_envoyer\_email

```

//////////////////////////////////////
// 2014-02-03 : bfp_envoyer_email : (PH) Paramètre "$smtp" par défaut
//////////////////////////////////////
function bfp_envoyer_email($email_param, $pieces_jointes=null){

    //---- traitement des paramètres :
    if(isset($email_param['smtp'])) {
        $smtp=$email_param['smtp'];
    }else{
        $smtp='eusmtpl.eu.bicworld.com';    /// Serveur par défaut
    }

    $message=$email_param['message'];
    $objet=$email_param['objet'];
    $email_expéditeur=$email_param['email_expéditeur'];
    $expéditeur=$email_param['expéditeur'];
    $destinataire=$email_param['destinataire'];

    ini_set('sendmail_from', $email_expéditeur);
    ini_set('SMTP',$smtp);

    $frontiere = "_lafrontiere";

    $headers = "From: <$expéditeur> \r\n";
    $headers .= "MIME-Version: 1.0\r\nContent-Type: multipart/mixed; boundary=\"$frontiere\"\r\n";
    if(isset($email_param['mail_reponse'])){ $headers .= 'Reply-To: '.$email_param['mail_reponse']."\r\n"; }
    $headers .= 'From: <'.$expéditeur.>'.\r\n';
    $headers .= 'Delivered-to: '.$destinataire.\r\n';
    if (isset($email_param['copie'])) { $headers .= 'Cc: '.$email_param['copie']."\r\n"; } // permet de mettre plusieurs destinataires
    if (isset($email_param['copie_cachée'])) { $headers .= 'Bcc: '.$email_param['copie_cachée']."\r\n"; } // permet de mettre plusieurs destinataires mais de cacher les adresses
    if (isset($email_param['accuse_reception'])) { $headers .= 'Disposition-Notification-To: '.$email_param['accuse_reception']."\r\n"; }

    //---- Message html
    $corps = "--". $frontiere ."\r\n";
    $corps .= "Content-Type: text/html; charset=utf-8\r\n\r\n";
    $corps .= $message ."\r\n";

    //---- pieces jointes
    if(isset($pieces_jointes)){
        foreach($pieces_jointes as $val){
            $corps .= "--". $frontiere . "\r\n";

            $fichier=$val;
            $piece_jointe = file_get_contents($fichier); //
            $piece_jointe = chunk_split(base64_encode($piece_jointe));

            $corps .= "Content-Type: application/msword; name=\"$fichier\"\r\n";
            $corps .= "Content-Transfer-Encoding: base64\r\n";
            $corps .= "Content-Disposition: inline; filename=\"$fichier\"\r\n\r\n";
            $corps .= $piece_jointe ."\r\n";
            $corps .= "\r\n\r\n";
        }
    }

    return mail($destinataire, $objet, $corps, $headers); // Envoi du message
}

```