



UNIVERSITÉ
DE LORRAINE



IUT Nancy Charlemagne
Département Informatique



IUT Nancy Charlemagne
Université de Lorraine
2 ter Boulevard Charlemagne
54052 Nancy Cedex
Dép. Informatique

Papeteries de Clairefontaine
19 rue de l'abbaye
88480 Etival Clairefontaine

*Développement d'un site intranet représentant la
production d'un atelier.*

Rapport de stage DUT informatique du 08/04/2013 au 14/06/2013
Christophe GERARD
2012-2013



IUT Nancy Charlemagne
Université de Lorraine
2 ter Boulevard Charlemagne
54052 Nancy Cedex
Dép. Informatique



Papeteries de Clairefontaine
19 rue de l'Abbaye
88480 Etival-Clairefontaine

*Développement d'un site intranet représentant la
production d'un atelier.*

Rapport de stage DUT informatique du 08/04/2013 au 14/06/2013

Christophe GERARD

Emmanuel CHRISMENT (tuteur en entreprise).

Michael LATHAM (parrain du stage).

Remerciements.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante et pleine d'intérêt qu'elles m'ont fait vivre pendant dix semaines au sein de l'entreprise Clairefontaine.

Tout d'abord, toute l'équipe pédagogique de l'IUT Nancy-Charlemagne qui m'a donné les connaissances (aussi bien techniques que générales) qui m'ont permises d'effectuer ce stage.

Emmanuel Chrisment, directeur informatique de l'entreprise et mon maître de stage tant pour son implication que pour son suivi régulier lors de mon séjour, sans qui je n'aurais pas pu vivre cette expérience.

Fabrice Aptel, responsable systèmes et réseaux, Régine Pezzetta chef de projet SAP et Philippe Willaume chef de projet informatique pour leur accueil, leur gentillesse, leur encadrement et leurs conseils durant ces dix semaines de stage.

Je remercie également toutes les personnes que j'ai rencontrées au sein de l'entreprise notamment leur accueil et l'intégration qu'ils m'ont apportée.

Pour finir, je remercie mon père, pour m'avoir aidé dans la démarche de recherches du stage et l'obtention d'un entretien au sein de l'entreprise Clairefontaine.

Sommaire.

I. Introduction	4
II. L'Entreprise	5
<i>II.1 Présentation générale</i>	5
<i>II.2 Le service informatique</i>	6
III. Le Stage	7
<i>III.1 Missions et objectifs du stage</i>	7
<i>III.2 Etude de l'existant</i>	8
<i>III.3 Conception</i>	13
<i>III.4 De la conception à la réalisation</i>	13
1) Partie administrative	15
2) Partie consultation	18
3) Partie graphique	19
4) Etude du programme existant	20
5) Améliorations	24
6) De nouvelles fonctionnalités	25
7) L'impression automatique	29
<i>III.5 De nouveaux besoins</i>	31
1) Bien connaître les besoins	31
2) Etude du calendrier des équipes	32
3) Finalité du projet	33
4) Migration de la base de données locale vers la base serveur	33
<i>III.6 Missions annexes</i>	34
<i>III.7 Bilan</i>	34
IV. Conclusion	35
V. Bibliographie	36
VI. Glossaire	37
VII. Annexes	40

I. Introduction

À la fin de ma seconde année de DUT informatique à l'IUT, j'ai été amené à effectuer un stage du 8 avril au 14 juin 2013. Le but de ce dernier était de valider les compétences acquises au cours de la formation.

Ce stage, je l'ai opéré dans l'entreprise Clairefontaine à Etival-Clairefontaine, en rapport avec mon intérêt pour l'informatique lié à l'industrie, j'ai pensé que la meilleure solution était d'effectuer un stage dans une papeterie qui a fait ses preuves et qui est connue de tout le monde.

Grandement intéressé par la programmation web, on m'a donné l'opportunité d'approfondir mes connaissances dans des langages spécifiques à ce domaine (HTML/CSS*, PHP*, JavaScript*), ainsi que de participer à l'analyse d'un projet déjà existant pour le concevoir et le réaliser.

J'ai donc eu pour mission de créer un site intranet* pour l'entreprise Clairefontaine qui devrait assurer le suivi de la production d'un atelier, ce même site serait consulté par les responsables de l'atelier, il devrait constituer des pages claires et générer des graphiques en accord avec les données que renvoient les machines. Chacun des graphiques pourrait s'imprimer automatiquement.

Le but de ce rapport est de vous présenter comment j'ai réagi aux différentes étapes du projet à réaliser, tout en respectant les besoins et les souhaits des demandeurs du site de l'entreprise.

Je commencerai donc par vous exposer l'entreprise où j'ai eu la chance d'être accueilli (Clairefontaine). Ensuite, je mettrai en évidence toute la partie analyse du projet, c'est-à-dire tout ce que j'ai dû comprendre avant d'en concevoir la suite, puis en troisième partie, je présenterai comment s'est déroulée la réalisation du projet. Pour ensuite finir, je parlerai de toutes les missions annexes qui m'ont été confiées lors de ce stage.

II. L'Entreprise

II.1 Présentation générale.

Papeteries de Clairefontaine est une entreprise qui existe depuis 1858, celle-ci est située dans les Vosges et emploie environ 600 personnes.

Sa capacité de production est de 165 000 tonnes de papier par an.

La papeterie propose des Produits très variés concernant le papier en lui-même, en voici une courte liste :

- ✚ Papiers pour transformation d'enveloppes(E.CF).
- ✚ Papiers publicitaires de vente par correspondance.
- ✚ Papiers pour imprimantes/traitement de texte.

L'entreprise dispose également d'un atelier dit « atelier de finition » qui transforme les bobines issues des machines à papier en produits commercialisables, sur lequel j'ai principalement travaillé.

L'atelier de finition dispose de dix machines de transformation qui sont divisées en deux espaces :

- ✚ « la salle » : deux équipes qui fonctionnent en 2x8.
- ✚ « le continu » : cinq équipes qui fonctionnent en 3x8.

Pour surveiller la production et connaître les points faibles des machines sur lesquels il faut travailler, des responsables de l'atelier consultent des courbes qu'un fichier EXCEL* permet de générer tous les matins (manuellement).

Pour avoir plus d'informations sur l'entreprise, notamment l'historique, le fonctionnement, le chiffre d'affaires et quelques chiffres clés, je vous conseille d'aller consulter l'annexe 1 spécialement dédiée à l'entreprise Clairefontaine.

II.2 Le service informatique.

Le service informatique se compose de :

- ✚ Un directeur informatique (Emmanuel Chrisment).
- ✚ Un Administrateur Systèmes et Réseaux (Fabrice Aptel).
- ✚ Deux chefs de projet (Philippe Willaume & Régine Pezzetta).

Ce service dispose de cinq bureaux avec un (ou plusieurs) ordinateur(s) qui selon les besoins utilisent un environnement Windows/MacOs ou Unix.

Tout au long de mon stage, j'ai effectué mon travail sur un ordinateur sous le système d'exploitation Windows.

L'environnement étant très calme et silencieux, cela a favorisé la concentration et l'efficacité du travail envisagé.

L'entreprise dispose également d'une salle sécurisée pour les serveurs et de salles remplies de disques durs qui permettent de sauvegarder les données et d'affecter de la mémoire aux serveurs si nécessaire.

Ce stage au sein d'une équipe complète m'a permis de me rendre compte de l'importance d'un tel service en entreprise, en effet, c'est un secteur aux allures passionnantes dans lequel on ne s'ennuie jamais.

III. Le Stage

III.1 Missions et objectifs du stage.

Comme je l'ai indiqué dans la présentation de l'entreprise, chaque jour un fichier EXCEL est généré et permet d'obtenir la production d'un atelier, cependant le système du fichier EXCEL est un peu lourd (j'en détaille le fonctionnement plus tard dans le rapport, à partir de la page 8).

Ma mission durant le stage consistait à réaliser un site web intranet qui avait au minimum les mêmes fonctionnalités que le fichier EXCEL, mais qui était moins lourd à utiliser. En effet le fichier EXCEL nécessitait beaucoup de manipulations fastidieuses de la part de l'utilisateur directement.

Je devais donc réaliser un site web qui représenterait la production d'un atelier, site qui serait consulté uniquement par les responsables de l'atelier. Ce site web devrait mettre en place des pages et des graphiques clairs représentant la production des différentes machines de l'atelier. Ces graphiques devraient s'imprimer automatiquement chaque matin.

Les objectifs du stage étaient donc les suivants :

- ✚ Réaliser un site web simple.
- ✚ Faciliter la tâche des responsables de l'atelier.
- ✚ Réaliser des graphiques complets.
- ✚ Rendre automatique l'impression des graphiques chaque matin.

Pour avancer correctement dans mon stage, j'ai construit un planning sous forme de diagramme de Gantt, ce diagramme est consultable en annexe 2.

III.2 Etude de l'existant.

Lors de mon arrivée dans l'entreprise, il existait déjà deux choses :

- ✚ Le fichier EXCEL qui génère les courbes.
- ✚ Le site déjà commencé.

Fichier EXCEL :

Cette feuille comporte plusieurs onglets :

- ✚ Graph X (3 graphes).
- ✚ List X (3 listes).
- ✚ Mise à jour et impression.
- ✚ Planning de toutes les machines (1 onglet pour 1 machine).

Les graphiques sont générés seulement si l'on a une présence de données dans l'onglet List correspondant.

Pour avoir des données dans l'onglet liste il faut cliquer sur le numéro de la machine dans l'onglet Mise à jour et impression (figure 1) :

Date

08/04/2013

Aujourd'hui

Mise à jour et impression
toutes machines

Cliquer sur un numéro de machine pour
obtenir le graphique de la machine
correspondante sur la journée saisie dans
le cadre "Date".

Ici on choisi la
machine
Will24

		Table valeurs	N° Feuille horaires	Format courbes	Vitesse	Ramette s	Gramma ge	Laize	Nb. Bobines	Tirée	Nb. Poses	Vitesse théorique	Echelle princ.	Echelle sec.
N°	Nom de machine													
1	Bielo25	dbo.Bielo25	8	C	P	S	S	P	S	P	S	N	2500	100
2	Bielo26	dbo.Bielo26	9	C	P	S	P	P	S	P	S	N	500	12
3	Coupeuse Bielo	dbo.CoupeuseBielo	10	C	P	S	D	P	S	P	S	N	1200	6
4	Dusenbery1	dbo.Dusenbery1	11	H	N	P	N	N	N	N	N	N	3	
5	Dusenbery3	dbo.Dusenbery3	12	H	N	P	N	N	N	N	N	N	3	
6	Etiqueteuse	dbo.Etiqueteuse	13	H	N	P	N	N	N	N	N	N	10	
7	GRM	dbo.GRM	14	H	N	P	N	N	N	N	N	N	130	
8	Massicot	dbo.Massicot	15	H	N	P	N	N	N	N	N	N	10	
9	Strecker	dbo.Strecker	16	C	P	N	P	P	S	P	D	P	1200	6
10	Will23	dbo.Will23	17	C	P	S	D	D	D	D	D	N	350	80
11	Will24	dbo.Will24	18	C	P	S	D	P	S	P	S	N	400	80
12	Trieuse	dbo.Trieuse	19	H	N	P	N	N	N	N	N	N		

Remarque : seules les données en vert sont modifiables

C = Courbe
H = Histogramme
P = Echelle principale
S = Echelle secondaire
N = Non pris en compte
D = Demandé non disponible

Important : l'ordre des colonnes de paramétrage des courbes doit rester le même que celui des colonnes de la requête. La colonne "Vitesse théorique" doit être la dernière car elle est calculée et non lue dans la base de données.

Figure 1 : fichier Excel onglet « Mise à jour et impression ».

Ensuite on peut aller vérifier que les données sont transmises dans les onglets List correspondant (ces données sont récupérées grâce à une macro* EXCEL en VBA*) :

List1 (figure 2) :

A2 fx 08/04/2013 05:00:32									
	A	B	C	D	E	F	G	H	I
1	Date	Vitesse	Ramettes	Grammage	Laize	Nbbob	Tiree	Nbposes	Vit.
2	08/04/2013 05:00:32	0	0			2	0	4	0
3	08/04/2013 05:01:33	0	0			2	0	4	0
4	08/04/2013 05:02:33	0	0			2	0	4	0
5	08/04/2013 05:03:33	0	0			2	0	4	0
6	08/04/2013 05:04:33	0	0			2	0	4	0
7	08/04/2013 05:05:33	0	0			2	0	4	0
8	08/04/2013 05:06:33	8	0			2	0	4	0
9	08/04/2013 05:07:33	8	1			2	0	4	0
10	08/04/2013 05:08:33	0	0			2	0	4	0
11	08/04/2013 05:09:33	0	0			2	0	4	0
12	08/04/2013 05:10:33	0	0			2	0	4	0
13	08/04/2013 05:11:33	8	1			2	0	4	0
14	08/04/2013 05:12:33	0	0			2	0	4	0
15	08/04/2013 05:13:33	14	2			2	0	4	0
16	08/04/2013 05:14:33	36	7			2	0	4	0
17	08/04/2013 05:15:33	52	11			2	0	4	0
18	08/04/2013 05:16:33	52	11			2	0	4	0
19	08/04/2013 05:17:33	29	6			2	0	4	0
20	08/04/2013 05:18:33	29	6			2	0	4	0
21	08/04/2013 05:19:33	29	6			2	0	4	0
22	08/04/2013 05:20:33	29	6			2	0	4	0
23	08/04/2013 05:21:32	29	6			2	0	4	0
24	08/04/2013 05:22:32	29	6			2	0	4	0
25	08/04/2013 05:23:32	82	17			2	0	4	0
26	08/04/2013 05:24:32	82	17			2	0	4	0
27	08/04/2013 05:25:32	82	17			2	0	4	0
28	08/04/2013 05:26:32	82	17			2	0	4	0
29	08/04/2013 05:27:32	82	17			2	0	4	0
30	08/04/2013 05:28:32	82	17			2	0	4	0
31	08/04/2013 05:29:32	82	17			2	0	4	0
32	08/04/2013 05:30:32	82	17			2	0	4	0
33	08/04/2013 05:31:32	82	17			2	0	4	0

Figure 2 : fichier Excel onglet « List ».

Puis le premier graphe est généré (figure 3) :

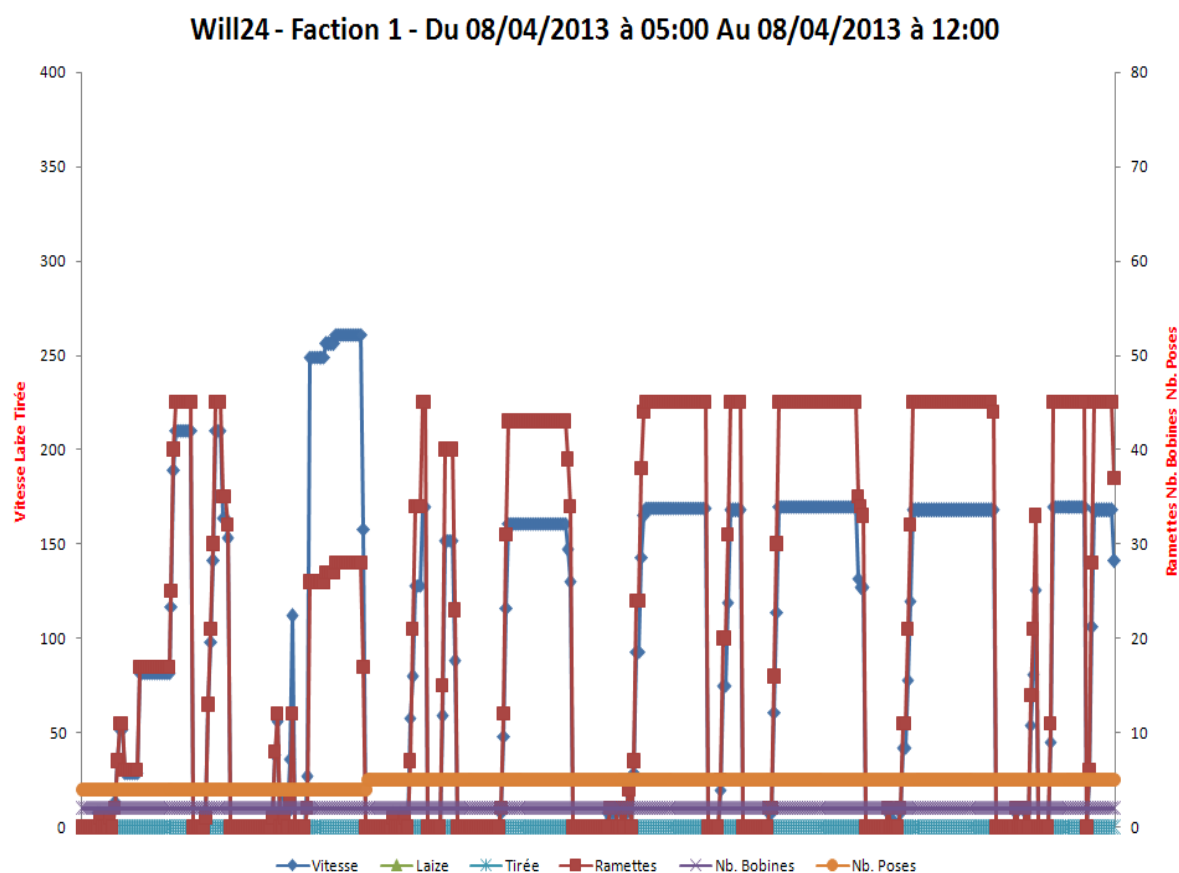


Figure 3 : fichier Excel onglet « Graph ».

Les deux autres listes et graphes fonctionnent sur le même principe, la seule différence est l'horaire car l'entreprise Clairefontaine dispose de trois factions (factions qui diffèrent selon la machine).

Il y a aussi des onglets planning pour chaque machine, ils ont toute la même syntaxe (figure 4):

A	B	C	D	E	F	G	H	I	J	K
	Lundi		Mardi		Mercredi		Jeudi		Vendredi	
	Début	Fin	Début	Fin	Début	Fin	Début	Fin	Début	Fin
Faction 1	04:00	12:00	04:00	12:00	04:00	12:00	04:00	12:00	04:00	12:00
Faction 2	12:00	20:00	12:00	20:00	12:00	20:00	12:00	20:00	12:00	20:00
Faction 3	20:00	04:00	20:00	04:00	20:00	04:00	20:00	04:00	20:00	04:00
Sélection F1	08/04/2013 04:00	08/04/2013 12:00	08/04/2013 04:00	08/04/2013 12:00	08/04/2013 04:00	08/04/2013 12:00	08/04/2013 04:00	08/04/2013 12:00	08/04/2013 04:00	08/04/2013 12:00
Sélection F2	08/04/2013 12:00	08/04/2013 20:00	08/04/2013 12:00	08/04/2013 20:00	08/04/2013 12:00	08/04/2013 20:00	08/04/2013 12:00	08/04/2013 20:00	08/04/2013 12:00	08/04/2013 20:00
Sélection F3	08/04/2013 20:00	09/04/2013 04:00	08/04/2013 20:00	09/04/2013 04:00	08/04/2013 20:00	09/04/2013 04:00	08/04/2013 20:00	09/04/2013 04:00	08/04/2013 20:00	09/04/2013 04:00

Figure 4 : fichier Excel onglet « Planning Bielo 25 ».

Sur chaque planning, les horaires sont modifiables afin d'obtenir un résultat satisfaisant selon la demande.

Enfin, dans l'onglet Mise à jour et impression est disposé un bouton principal nommé « Mise à jour et impression toutes machines » (figure 5) :

Date: 08/04/2013 Aujourd'hui

Mise à jour et impression toutes machines

Cliquer sur un numéro de machine pour obtenir le graphique de la machine correspondante sur la journée saisie dans le cadre "Date".

Lancer l'impression.

N°	Nom de machine	Table valeurs	N° Feuille horaires	Format courbes	Vitesse	Ramette s	Grammage	Laize	Nb. Bobines	Tirée	Nb. Poses	Vitesse théorique	Echelle princ.	Echelle sec.
1	Bielo25	dbo.Bielo25	8	C	P	S	S	P	S	P	S	N	2500	100
2	Bielo26	dbo.Bielo26	9	C	P	S	P	P	S	P	S	N	500	12
3	Coupeuse Bielo	dbo.CoupeuseBielo	10	C	P	S	D	P	S	P	S	N	1200	6
4	Dusenbery1	dbo.Dusenbery1	11	H	N	P	N	N	N	N	N	N	3	
5	Dusenbery3	dbo.Dusenbery3	12	H	N	P	N	N	N	N	N	N	3	
6	Etiqueteuse	dbo.Etiqueteuse	13	H	N	P	N	N	N	N	N	N	10	
7	GRM	dbo.GRM	14	H	N	P	N	N	N	N	N	N	130	
8	Massicot	dbo.Massicot	15	H	N	P	N	N	N	N	N	N	10	
9	Strecker	dbo.Strecker	16	C	P	N	P	P	S	P	D	P	1200	6
10	Will23	dbo.Will23	17	C	P	S	D	D	D	D	D	N	350	80
11	Will24	dbo.Will24	18	C	P	S	D	P	S	P	S	N	400	80
12	Trieuse	dbo.Trieuse	19	H	N	P	N	N	N	N	N	N		

Remarque : seules les données en vert sont modifiables C = Courbe H = Histogramme P = Echelle principale S = Echelle secondaire N = Non pris en compte D = Demandé non disponible

Important : l'ordre des colonnes de paramétrage des courbes doit rester le même que celui des colonnes de la requête. La colonne "Vitesse théorique" doit être la dernière car elle est calculée et non lue dans la base de données.

Figure 5 : fichier Excel onglet « Mise à jour et impression ».

En cliquant sur ce bouton, chacun des graphes va être imprimé grâce à une macro d'EXCEL.

Afin de faciliter l'utilisation de l'interface, Mr Chrisment a commencé à développer un site web afin d'avoir le suivi des machines finition.

Présentation du site déjà existant :

Suivi des machines Finition						
Période d'historique du graphe : 4 heures						
	Vitesse	Cadence	Heure	Statut	Evénement	Signal
Bielo 25	263	71	09/04/2013 14:14:41	● Marche		⚡
Bielo 26	13	0.2	09/04/2013 14:14:42	● Marche		⚡
Coupeuse Bielo	222	2.67	09/04/2013 14:14:41	● Marche		⚡
Dusenbery 1	0	2	09/04/2013 14:12:10	● Marche		⚡
Dusenbery 3	0	0.7	09/04/2013 14:06:25	● Marche		⚡
Etiqueteuse	0	6	09/04/2013 14:12:14	● Marche		⚡
GRM	0	0	09/04/2013 14:12:10	● Arrêt		⚡
Massicot	0	0	09/04/2013 14:12:10	● Arrêt		⚡
Strecker	0	0	09/04/2013 14:14:41	● Arrêt	Couteau circ. capot	⚡
Will 23	256	55	09/04/2013 14:14:41	● Marche		⚡
Will 24	169	10	09/04/2013 14:14:42	● Marche		⚡
Trieuse	0	0.2	09/04/2013 14:15:09	● Marche		⚡

Figure 6 : page d'accueil du site existant.

Explication de la figure 6 :

Chaque lien souligné en bleu à gauche permet de consulter le graphique de la machine selon la période historique du graphe (ici 4, 8, 12, 24, 48 ou 72h).

La colonne statut permet de savoir si la machine est en fonction ou non.

Les événements permettent d'avoir des précisions lors d'arrêt de machine ou de dysfonctionnement.

Et la colonne signal permet de savoir si la machine communique toujours avec le serveur, ici toutes les machines sont en train de communiquer.

Chaque machine étant représentée par un lien, qu'est-ce que cela produirait si je clique sur un lien ?

Cliquons par exemple sur le lien Bielo 25, le résultat obtenu est le suivant (figure 7) :

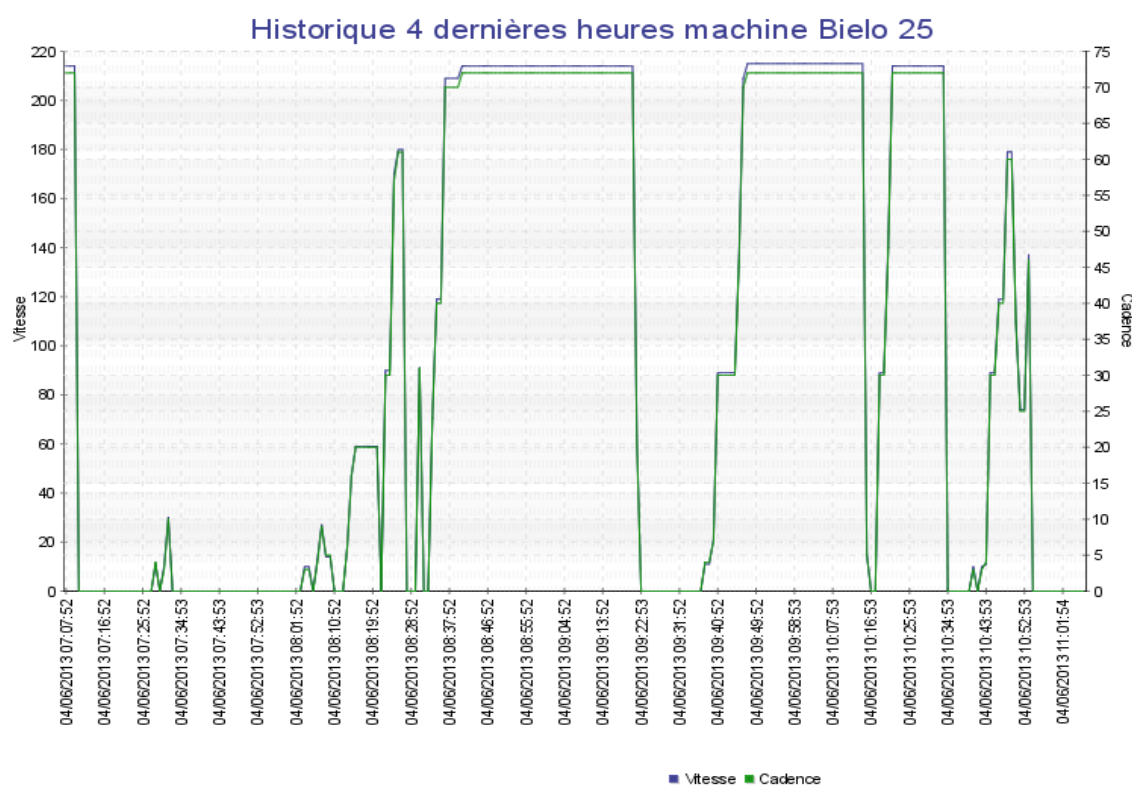


Figure 7 : graphe généré lors de mon arrivée.

Comme la période historique du graphe sélectionné est de 4h, on obtient le graphique des 4 dernières heures sur la machine Bielo 25.

Le site étant en cours de développement, il manquait des données à afficher sur la courbe.

III.3 Conception.

Une fois l'existant bien étudié et compris, j'ai commencer à concevoir mon projet, dans un premier temps je me suis dit que la meilleure chose à faire était de créer toute la partie administrative du site (les responsables de l'atelier m'ont demandé de pouvoir changer la priorité d'une courbe sur le graphique facilement par exemple). Pour ce faire, j'ai donc décidé de garder toute l'infrastructure du site et de juste rajouter un bouton qui permet d'accéder au paramétrage complet des machines, pour cela, il fallait également protéger les pages d'administration par mot de passe.

Le deuxième problème était que les graphiques générés pour le moment n'avaient pas toutes les données voulues à la base, j'ai donc décidé dans un deuxième temps d'étudier le programme qui effectuait le dessin du graphique et d'implanter toutes les données manquantes.






Troisièmement, il fallait aussi créer un système de gestion des plannings des machines et pouvoir changer le mot de passe si nécessaire.

Enfin avec Mr Chrisment, nous avons décidé de créer une nouvelle page de consultation des plannings des machines accessible par tout le monde.

Toutes ces conceptions devaient garder une grande cohérence avec le fichier EXCEL de base.

III.4 De la conception à la réalisation.

Pour pouvoir réaliser le site web, il me fallait avant tout des outils, les outils que j'ai choisis sont :

-  Le langage PHP.
-  Le langage HTML 5/CSS 3.
-  Du JavaScript.
-  Une base de données* locale.
-  Une base de données SQL* Server (déjà existante).

Afin de ne pas déranger le serveur* déjà mis en place qui est un serveur de production, Mr Chrisment m'a créé un serveur spécialement pour le développement du projet appelé SCLF-DEVOPC.

Sur ce serveur qui fonctionne sous un environnement Unix, nous avons installé plusieurs modules :

- ✚ Samba (il sert à partager des fichiers et des imprimantes sur les réseaux Microsoft).
- ✚ SSH (pour permettre la connexion à distance sur le serveur).
- ✚ Apache (pour permettre la lecture de page web par des clients*).

Une fois tous ces outils installés et à ma disposition, j'ai donc pu commencer à réaliser mon projet.

Comme j'ai respecté tout au long mon schéma de la conception, j'ai donc commencé par l'administration du site et la consultation des plannings, cette partie constituait quatre nouvelles pages à ajouter et deux boutons à ajouter sur la page d'accueil :

- ✚ Ajouter des boutons sur l'accueil.
- ✚ Page de gestion des paramètres.
- ✚ Page de gestion des plannings des machines.
- ✚ Page pour modifier le mot de passe.
- ✚ Page de consultation des plannings.

Avant de pouvoir commencer à programmer, j'ai d'abord établi une base de données sous phpmyadmin appelée « suivimachines ».

Dans cette base de données, j'y ai inclus les tables* suivantes :

- ✚ Paramètres.
- ✚ Planning.
- ✚ Password.

Voici leur structure (figure 8) :

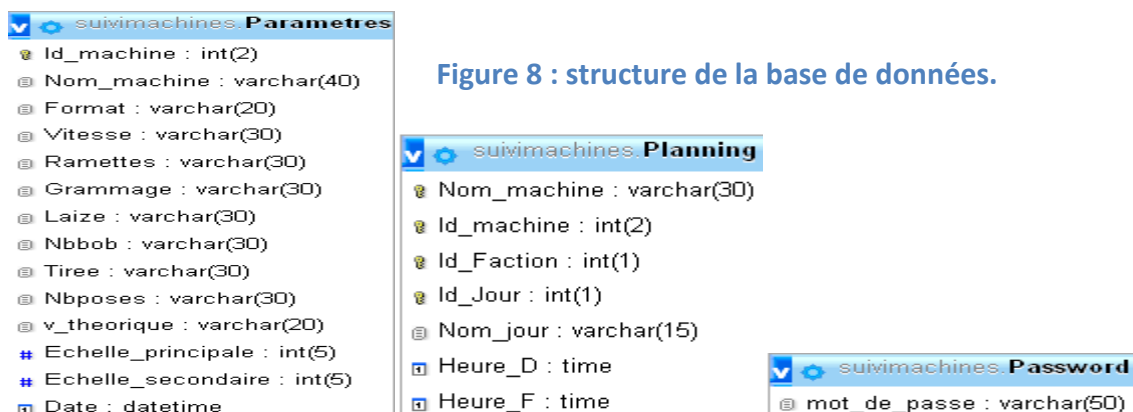


Figure 8 : structure de la base de données.

1) Partie administrative.

Pour respecter au maximum les contraintes du fichier EXCEL, j'ai initialisé toutes les données des tables Paramètres et Planning à l'aide de phpmyadmin (les utilisateurs ne pourront pas insérer ou supprimer de données mais juste les modifier).

Ajout des boutons sur l'accueil :

J'ai donc commencé par ajouter mes boutons, qui sont en vérité des liens camouflés avec du CSS :

Période d'historique du graphe : 4 heures Gestion des paramètres. Planning

Figure 9 : boutons de la page d'accueil.

Ainsi lorsque l'on clique sur gestion des paramètres, on est redirigé vers la page de gestion des paramètres et lorsqu'on clique sur planning, on est redirigé vers la page de consultation des plannings (voir annexe 3).

Page de gestion des paramètres :

Cette page faisant partie des pages administratives, il fallait la protéger par mot de passe. Ayant déjà effectué une protection par mot de passe dans le cadre de mon projet tuteuré, j'ai pu aisément mettre en place un tel système (voir annexe 3).

Une fois le système mis en place et le mot de passe rentré, on arrive à l'écran suivant :

 Paramétrages des machines

Nom de la machine	Format de la courbe	Vitesse	Ramettes	Grammage	Laize	Nb. bob	Tiree	Nb. pos	V theorique	Echelle principale	Echelle secondaire
Bielo 25	Courbe	Principale	Secondaire	Secondaire	Principale	Secondaire	Principale	Secondaire	Ignorer	<input type="text" value="2500"/>	<input type="text" value="100"/>
Bielo 26	Courbe	Principale	Secondaire	Principale	Principale	Secondaire	Principale	Secondaire	Ignorer	<input type="text" value="500"/>	<input type="text" value="12"/>
Coupeuse Bielo	Courbe	Principale	Secondaire	Indisponible	Principale	Secondaire	Principale	Secondaire	Ignorer	<input type="text" value="1200"/>	<input type="text" value="6"/>
Dusenbery 1	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text" value="3"/>	<input type="text"/>
Dusenbery 3	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text" value="10"/>	<input type="text"/>
Etiqueteuse	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text" value="30"/>	<input type="text"/>
GRM	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text" value="10"/>	<input type="text"/>
Massicot	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text" value="1200"/>	<input type="text" value="6"/>
Strecker	Courbe	Principale	Ignorer	Principale	Principale	Secondaire	Principale	Indisponible	Principale	<input type="text" value="350"/>	<input type="text" value="80"/>
Will 23	Courbe	Principale	Secondaire	Indisponible	Indisponible	Indisponible	Indisponible	Indisponible	Ignorer	<input type="text" value="400"/>	<input type="text" value="80"/>
Will 24	Courbe	Principale	Secondaire	Indisponible	Principale	Secondaire	Principale	Secondaire	Ignorer	<input type="text"/>	<input type="text"/>
Trieuse	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text"/>	<input type="text"/>
Signode	Histogramme	Ignorer	Principale	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	Ignorer	<input type="text"/>	<input type="text"/>

Valider Réinitialiser Accueil Modifier le mot de passe Gérer les planning

Figure 10 : page de gestion des paramètres.

La table Paramètres étant initialisée, chaque sélecteur va chercher la valeur du paramètre dans cette même table, pour les échelles, si la valeur vaut NULL alors on affiche un champ vide, sinon on affiche la valeur de l'échelle (voir annexe 3).

Chacun des paramètres peut avoir la valeur suivante (sauf les échelles) : « Principale », « Secondaire », « Ignorer », « Indisponible ».

Sur la figure 10, on peut observer la présence de trois liens, un pour retourner à l'accueil, un autre pour modifier les mots de passe et un dernier pour gérer les plannings, ce sont simplement des liens qui redirigent vers d'autres pages PHP.

Page de gestion des plannings machine :

Cette page est uniquement accessible en passant par la page de gestion de paramètres, je ne trouvais pas utile de rajouter un bouton sur l'accueil concernant cette fonctionnalité. Cette page étant également administrative il fallait aussi la protéger par mot de passe.

Lorsqu'on arrive sur cette page on obtient l'écran suivant :

 Gestion du planning de la machine : Bielo 25

Sélectionner la machine pour en modifier le planning : Bielo 25

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin
Faction 1	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00
Faction 2	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00
Faction 3	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00

Valider Réinitialiser

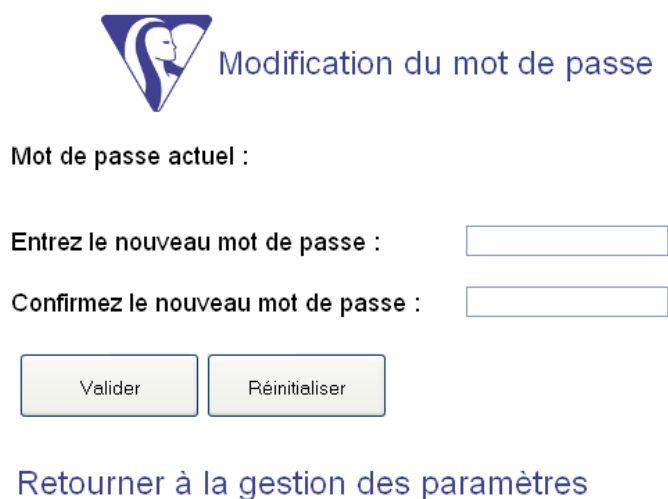
[Retourner à la gestion des paramètres](#)

Figure 11 : page de gestion des plannings machine.

Le fonctionnement est très similaire à la gestion des paramètres, la seule chose supplémentaire est la présence d'un sélecteur de machine, en effet, chaque machine pouvant avoir un planning différent, il était nécessaire de pouvoir changer la valeur des sélecteurs selon la machine. Si on change de machine, alors la page se rafraîchit et charge les horaires correspondants à la machine (voir annexe 3).

Page de modification du mot de passe :

Cette page permet de modifier le mot de passe dans la base de données, elle est aussi évidemment protégée par mot de passe. L'écran de modification des mots de passe a l'allure suivante :



Modification du mot de passe

Mot de passe actuel :

Entrez le nouveau mot de passe :

Confirmez le nouveau mot de passe :

[Retourner à la gestion des paramètres](#)

Figure 12 : page de modification du mot de passe.

Par soucis de sécurité et confidentialité, je ne divulgue pas le mot de passe dans mon rapport. Pour pouvoir modifier le mot de passe, il faut saisir deux fois le même mot de passe ce qui augmente la sécurité.

Un des gros problèmes de la partie administrative du site était à cause du mot de passe, le problème que j'ai rencontré a été que lorsque l'on tapait le mot de passe une fois, qu'on reculait, et qu'on retournait sur une page protégée, le mot de passe était de nouveau demandé (ce qui avait un côté peu pratique). Or j'étais censé réaliser un site web simple, c'est pourquoi je me suis renseigné sur le langage PHP et j'ai trouvé la solution suivante : les cookies*.

Un cookie est un petit fichier texte qui est écrit sur le disque dur du client afin de permettre au serveur web de mémoriser une information donnée par un client, (dans un formulaire par exemple) dans notre cas le cookie permet de mémoriser le mot de passe rentré et à une durée de dix minutes (voir annexe 3). Grâce au cookie l'utilisateur pouvait alors naviguer sur la partie administration pendant dix minutes sans retaper le mot de passe.

2) Partie consultation.

Page de consultation des plannings :

Cette page quand à elle ne nécessite aucun mot de passe, elle est accessible depuis l'accueil, un mot de passe n'était pas adéquat car consulter les plannings d'une machine doit être possible pour tout le monde.

Une fois que l'on accède à cette page, on obtient un écran similaire à celui-ci :

Consultation des plannings

Sélectionner la machine pour en obtenir le planning : Bielo 25

Planning de la machine : Bielo 25

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin
Faction 1	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00
Faction 2	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00
Faction 3	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00	20:00 04:00

[Accueil](#)

Figure 13 : consultation des plannings.

Pour cette page, le fonctionnement est quasiment le même que pour la gestion des plannings machine, le seul changement est qu'il n'y a pas de formulaire ni de sélecteurs, ici on favorise l'affichage plutôt que la modification.

Une fois toute la base du site créée j'ai pu m'atteler à la deuxième partie de ma conception, étudier le programme déjà existant et implémenter toutes les données manquantes.

3) Partie graphique.

Lors de mon arrivée, les seuls graphiques générés avaient l'allure de la figure 6.

J'ai parlé dans mes outils mis à disposition d'une base de données déjà existante, cette base de données à toute son importance dans cette partie.

La base de données est une base SQL Server 2008, elle se nomme OPCDATA et dispose des tables suivantes :

-  Dbo.Bielo25
-  Dbo.Bielo26
-  Dbo.CoupeuseBielo
-  Dbo.Dusenbery1
-  Dbo.Dusenbery3
-  Dbo.Etiqueteuse
-  Dbo.GRM
-  Dbo.Massicot
-  Dbo.Signode
-  Dbo.Strecker
-  Dbo.Trieuse
-  Dbo.Will23
-  Dbo.Will24
-  Dbo.confEvents
-  Dbo.confMachines

Toutes les tables sauf les deux dernières ont la même structure :

Dbo.nomdemachine (Date, Vitesse, Ramettes, Grammage, Laize, Nbbob, Tiree, Nbposes, Signal).

Dbo.confEvents (IDmachine, IDevent, type, description).





Dbo.confMachines (Machine, IDmachine, Table, Index, Alerte).

Chacune des tables machines enregistre des données toutes les minutes, notamment la vitesse, la laize, le grammage...etc.

4) Etude du programme existant.

Après une courte étude du script*, je me suis rendu compte qu'il utilisait la librairie* PHP pChart*, c'est pourquoi je me suis laissé quelques jours afin de me familiariser avec la librairie et ainsi maîtriser au mieux mon sujet.

Lors de l'étude de la librairie, j'ai déduit que le schéma pour créer des graphiques était toujours le même :

-  Création des données.
-  Remplissage des données.
-  Création de l'image et implémentation des données.
-  Rendu de l'image.

Création des données :

La création des données s'effectue de manière simple, il suffit d'écrire : `$mydata = new pData () ;`

C'est tout pour la partie création de données.

Remplissage des données :

Pour ajouter des données, il suffit d'écrire : `$mydata->AddPoints (LisedePoints, LabelAxe) ;`

Cette ligne permet d'ajouter aux données une liste de points pour un nom d'axe (attention c'est bien une chaîne de caractère qui est attendu et non un id).

Il faut évidemment indiquer à la librairie sur quel axe ajouter la liste de points : `$mydata->setSerieOnAxis (LabelAxe, IDAxe) ;`

Sachant que si IDAxe vaut 1, on est sur l'axe principal et s'il vaut 2, on est sur le secondaire.

Ensuite, on définit la position de l'axe : `$mydata->setAxisPosition (IDAxe, POS_AXE) ;`

Et pour finir dans le remplissage, on indique qui est l'axe des abscisses. Il apparaît évident que pour générer un graphe, il faut au moins une liste de points sur l'abscisse, et une sur l'ordonnée : `$mydata->SetAbscissa(LabelAxe) ;`

Création de l'image et implémentation des données :

Dans un premier temps, l'image est représentée par un objet `pImage` de la librairie :
\$myImg = new pImage (hauteur, largeur, donnees);

Une fois l'image créée, il faut définir la zone de dessin : *\$myImg->setGraphArea(x1, y1, x2, y2);*

Puis on dessine la légende des courbes ajoutées : *\$myImg->drawLegend(x, y);*







Et enfin, on dessine l'échelle : *\$myImg->drawScale ();*

Rendu de l'image :

Pour le rendu de l'image c'est très simple : *\$myImg->drawLineChart ();*

Ici on dessine des courbes, mais il est possible de dessiner d'autre type de graphiques (histogramme, des zones ...etc.), pour un fonctionnement plus détaillé, je vous conseille d'aller consulter l'annexe 4 traitant en partie de la bibliothèque `pChart`.

Une fois de nombreux tests effectués concernant la librairie, notamment sur les fonctions qu'elle proposait, j'ai alors établi le schéma que le script suivait pour créer des graphiques :

-  Connexion à la base de données.
-  Récupérer la machine depuis l'URL*.
-  Faire le lien entre la machine et la table à aller lire dans la base de données.
-  Définir l'intervalle de temps.
-  Lire les données dans la table.
-  Construction du graphique.

Connexion à la base de données :

Il existe, en PHP le moyen de se connecter à un serveur de données, en renseignant dans des variables, le nom du serveur, le nom d'utilisateur, le mot de passe pour accéder à la base : *lien = mssql_connect (serveur, utilisateur, mot_de_passe) ;*

Puis une fois connecter au serveur, on va sélectionner la base de données : *mssql_select_db (nom_base_de_donnees, lien);*

Récupérer la machine depuis l'URL :

Comme je l'ai précisé plus haut dans mon rapport, lorsque l'on clique sur un lien depuis l'accueil, on exécute un script qui à l'allure suivante : *graphe_machine.php ?m=nomdemachine*.

Or, en PHP il existe un tableau associatif appeler `$_GET` qui permet de récupérer un paramètre depuis une URL, ainsi, pour récupérer le nom de la machine avec laquelle va travailler le script on fait : *machine = \$_GET ['m'] ;*

Faire le lien entre la machine et la table à aller lire dans la base de données :

Pour effectuer ce lien, on utilise la table *Dbo.confMachines*, de cette façon, avec une simple requête SQL, on peut savoir dans quelle table aller lire :

```
Select * from Dbo.confMachines where Nom_machine = 'machine';
```

Définir l'intervalle de temps :

Pour définir cet intervalle, on utilise depuis l'index, une variable de session*. Une variable de session est une variable qui contient une valeur et qui ne change jamais (sauf changement voulu) de page en page. C'est pour cela, qu'il est facilement possible de récupérer l'intervalle sur lequel se calquer (sur la figure 6, cela correspond au sélecteur de la durée).

Lire les données dans la table :

Cette action s'effectue seulement une fois que l'on connaît l'intervalle de temps sur lequel se situer, une fois cet intervalle connu, on exécute la requête suivante :

```
Select * from Dbo.nommachine where Date between datedebut and datecourante Order by Date ASC;
```

Datedebut et *datecourante* correspondent à des types *DateTime* (voir annexe 4).

On stocke alors le résultat dans une variable, puis, s'il y a plus de 0 lignes en résultat alors on cherche à effectuer une boucle sur le nombre de lignes :

```
While (ligne = mssql_fetch_array (resultat_requete, CONST_ASSOC)) {
```


Puis dans des tableaux, on va ajouter les données voulues : *vitesse* [\$i] = *ligne* ['Vitesse'];

Lors de mon arrivée, on ne récupérait que la vitesse et la cadence.

Les données renvoyées concernent aussi les dates. Cependant, il y avait un problème, les tables renvoient des données qui sont enregistrées toutes les minutes dans la base, et les machines sont au format GMT, alors que nous enregistrons dans les tables les données au format GMT + 2, il m'a donc fallu gérer ce problème tout le long du script. Les dates étant de type DateTime, cela s'est avéré très simple de changer le format de date, il existe une fonction qui s'appelle SetTimeZone () et qui permet de définir le fuseau horaire sur lequel se caler. Il m'a alors suffi de définir un fuseau horaire GMT pour récupérer les données depuis les tables des machines, puis de remettre le fuseau sur GMT + 2 pour ajouter les dates dans le tableau de données (afin d'établir une cohérence entre les données récupérer et l'affichage graphique).

Construction du graphique :

En suivant le schéma de la librairie pChart, voici ce que l'on obtenait lors de mon arrivée (figure 14):

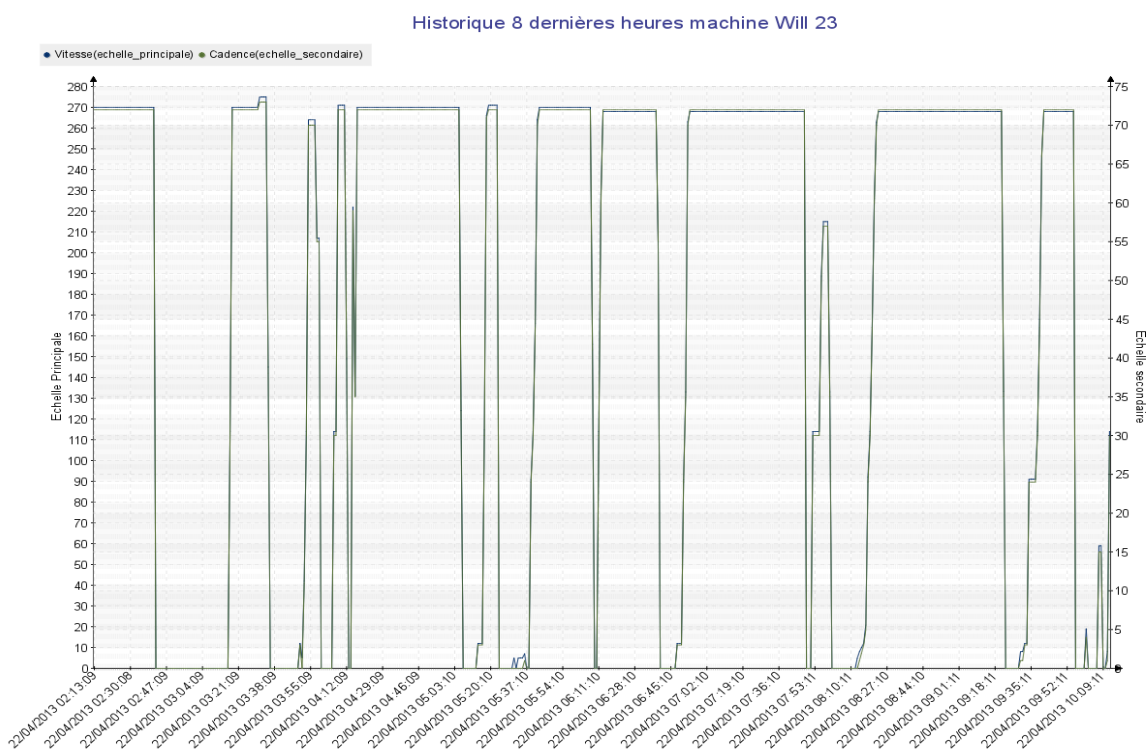





Figure 14 : exemple de graphique généré.

5) Améliorations.

Après avoir bien étudié le fonctionnement du script, j'ai pu facilement le modifier et ajouter tout ce qui était nécessaire pour que ça fonctionne au mieux.

Les améliorations que j'ai apportées au script sont les suivantes :

-  Ajout des données manquantes.
-  Gestion des axes.
-  Gestion des échelles.

Ajout des données manquantes :

Cela a été le plus simple à ajouter, il m'a suffi de créer un tableau par paramètres quelle que soit la machine (*laize[i] = ligne ['laize']* par exemple).

Gestion des axes.

Pour gérer les axes, je me suis servi de la base de données en locale et de la table Paramètres, j'ai d'abord testé la priorité du paramètre, est-ce que ce paramètre est : principal, secondaire, indisponible ou on l'ignore ?

Si le paramètre est « principal », alors je mets le tableau de données correspondant sur l'axe principal, si le paramètre est secondaire alors je le mets sur l'axe secondaire et sinon je ne fais rien ce qui m'évite d'ajouter une courbe pour rien (voir annexe 4).

Pour effectuer ce test, je me suis d'abord connecté à la base de données locale et récupérée la valeur à partir de la table paramètres, puis j'ai comparé le résultat et une chaîne de caractère (principale ou secondaire).

Exemple (récupération du paramètre Vitesse):

```
$query = "Select Vitesse from Parametres where Nom_machine = " . "" . $machine . "";
```

```
$result = mysql_query ($query);
```

```
$ligne = mysql_fetch_array ($result, MYSQL_ASSOC);
```

```
$lineV = $ligne ['Vitesse'];
```

```
Mysql_free_result ($result);
```

Gestion des échelles :

Les échelles sont gérées également grâce à la table Paramètres de la base de données locale, elle permet de définir pour certaines machines l'échelle maximale que peut atteindre l'axe principal ou/et secondaire. La librairie pChart étant très bien faite, elle génère automatiquement des échelles si elles ne sont pas définies au préalable, c'est pourquoi j'ai juste eu à gérer les différents cas des échelles (voir annexe 4).

Une fois toutes ces améliorations intégrées, le résultat du nouveau graphique donnait quelque chose comme ceci (figure 15) :

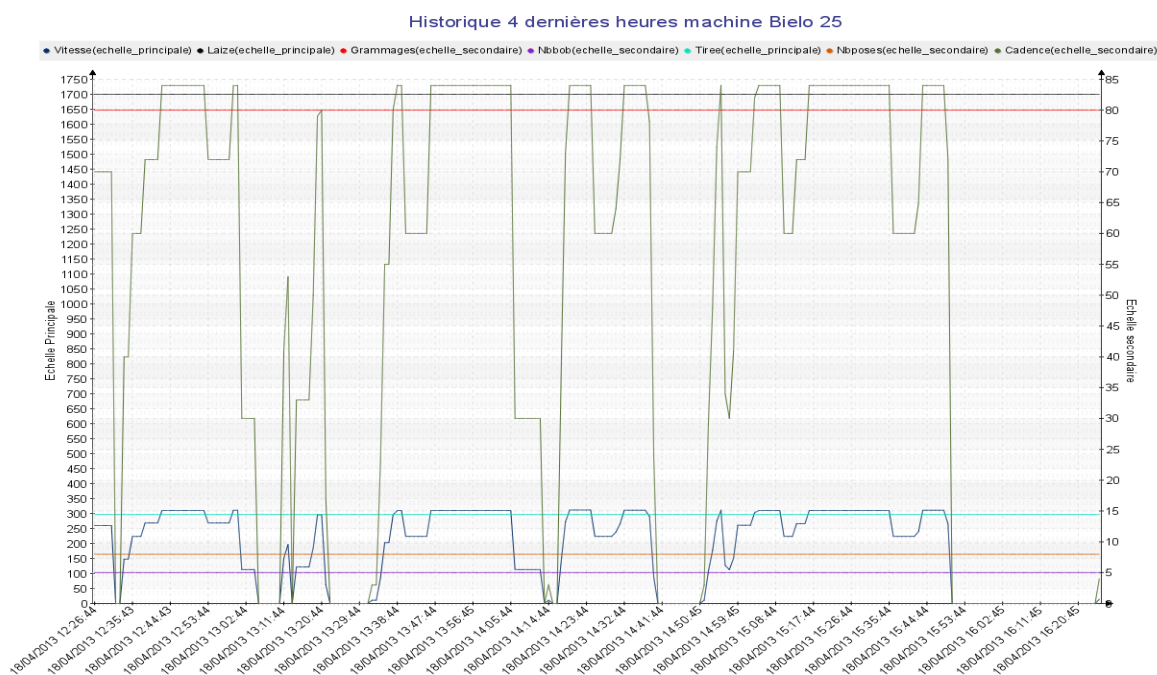


Figure 15 : nouveau graphique généré.

6) De nouvelles fonctionnalités.

Une fois tout cela mis en place, nous avons discuté d'améliorations possibles du site et de fonctionnalités que l'on pourrait mettre en place avec Mr Chrisment.

Il est en ressorti les choses suivantes :

- ✚ Ajouter un journal des événements sur la page du graphique.
- ✚ Consulter un graphique depuis les plannings (par faction).
- ✚ Consulter les graphiques d'une journée depuis les plannings.

Ajouter un journal des événements :

Cet ajout a été possible grâce à la table Dbo.confEvents, table qui permet d'obtenir la description d'un événement qui s'est passé à partir d'un Idmachine et d'un IDevent.

Le script qui crée les graphiques dispose d'une page HTML directement intégrée, ainsi, on peut facilement afficher l'image sur la page grâce à une balise img et pour le journal des événements il suffit de créer une table en HTML (voir annexe 5).

Pour remplir cette table HTML (qui se remplit dynamiquement via PHP), il faut aller lire dans la table Dbo.confMachines l'id de la machine passé en paramètres de l'URL, puis aller chercher tous les événements dans la table Dbo.confEvents sur l'intervalle demandé (voir annexe 5).

Consulter un graphique depuis les plannings :

Pour cette amélioration, il m'a fallu repenser un petit peu la conception, en effet, comment pourrais-je obtenir toutes les informations nécessaires depuis la page des plannings vers le script qui génère les graphiques ?

Il me fallait au minimum :

- ✚ La date.
- ✚ Le nom de la machine.
- ✚ La faction correspondante.
- ✚ Une façon de distinguer l'appel du script depuis l'accueil de celui des plannings.
- ✚ L'heure de départ de la faction.
- ✚ L'heure de fin de la faction.

À ce moment là, je me suis souvenu que le script de base utilisait le tableau \$_GET, je me suis donc attelé à construire une URL qui réunirait toutes les données nécessaires :

graphe_machine.php?m=machine&hd=heuredeb&hf=heurefin&bool=false&date=date&isfaction=idfaction

Avec cette composition d'URL, je m'assurais non seulement d'avoir toutes les données nécessaires pour construire un graphe, mais aussi pour différencier l'appel depuis l'accueil de celui des plannings.

Seulement, pouvoir consulter des graphiques par faction engendrait de nouvelles contraintes sur la consultation des plannings. En effet, mon URL nécessitant une date, il me fallait alors implémenter une façon d'avoir toujours la date correspondante lors d'un clic sur la faction.

C'est pourquoi j'ai tout d'abord rajouté un champ de saisie, afin de pouvoir saisir une date, cependant, il fallait que la saisie soit la plus automatique possible, j'ai donc pensé à créer un calendrier qui s'afficherait dynamiquement grâce à la librairie JQuery* (voir annexe 5), le JavaScript n'étant pas nécessairement activé chez le client, il restait quand même possible de saisir la date à la main.

À la demande de Mr Chrisment, j'ai rajouté en dessous du champ de saisie la date du lundi de la semaine choisie, pour se repérer au niveau visuel c'était plus simple.

Cette fonctionnalité m'a demandé de créer quelques fonctions, fonctions que je détaille en annexe 5.

Après implémentation de la nouvelle fonctionnalité, regardons les changements apportés à la page de consultation des plannings (figure 16) :

Consultation des plannings

Sélectionner la machine pour en obtenir le planning : Bielo 26

Planning de la machine : Bielo 26

Date: 02/06/2013

Début de la semaine : Lundi 27/05/2013

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin	Début Fin
Faction 1	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	04:00 12:00	N/A N/A	N/A N/A
Faction 2	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	12:00 20:00	N/A N/A	N/A N/A
Faction 3	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A	N/A N/A

Accueil

Figure 16 : nouvelle page de consultation des plannings.

Exemple :

Cliquons par exemple sur la case 4h | 12 h du lundi de la faction 1 sur la machine Bielo 26 à la date du 27/05/2013, on obtient l'URL suivante :

Graphe_machine.php?m=Bielo+26&hd=04:00&hf=12:00&bool=false&date=27-05-2013&idfaction=1

Avec cela correspond le graphe suivant :

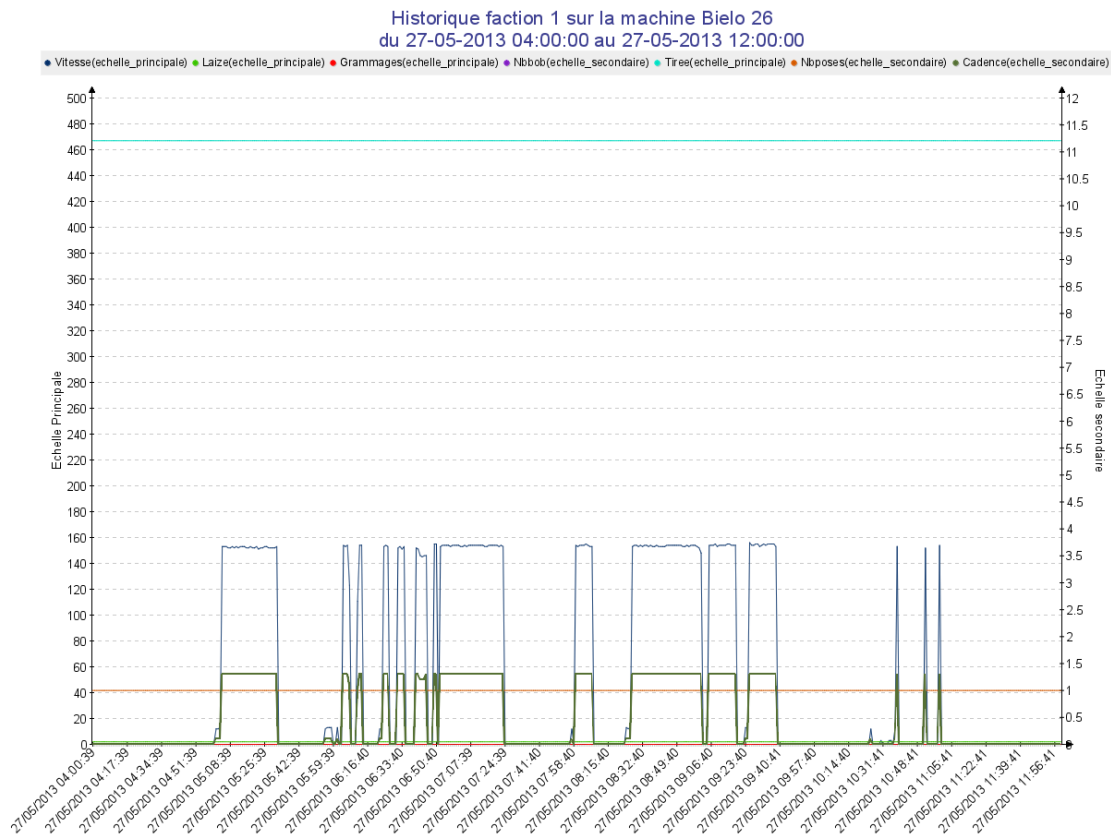


Figure 17 : graphique généré depuis les plannings.

Consulter les graphiques d'une journée depuis les plannings :

Le principe de cette fonctionnalité est très simple, au lieu de cliquer sur l'horaire d'une faction, on clique sur lundi, mardi, mercredi ... dimanche. Et lorsque l'on clique sur la journée, il faudra alors sortir les graphiques correspondant tous sur la même page.

Dans un premier temps, j'ai essayé de faire une boucle for et pour toutes les factions, d'appeler le script qui crée les graphiques avec les paramètres dans l'URL (même structure qu'avant). Seulement, les résultats n'étaient pas du tout ceux auxquels-je m'attendais, en effet, le fait d'appeler le même script en même temps à créé de gros problème dans le rendu des graphiques, notamment au niveau des échelles qui n'avaient plus aucune cohérence avec les données enregistrées, j'ai donc cherché autre une solution sur internet.

La solution que j'ai trouvée et qui était pour moi la plus adéquate a été d'utiliser la bibliothèque cURL*.

cURL permet de communiquer avec des sites de façon distante, ce qui implique qu'il est possible de récupérer la source d'une page externe, autrement dit, avec cette bibliothèque j'ai pu appeler trois fois de suite (boucle for) mon script avec les paramètres de l'URL qui correspondent à la journée sur laquelle on a cliqué.

Cependant, il me fallait un moyen de dissocier la consultation des plannings via une faction et via la journée, pour cela j'ai donc créé un nouveau script qui a l'URL suivant :

url = construit_graphe.php ?m=machine&date=date

Les deux paramètres s'expliquent du fait que lorsque l'on clique sur la journée, on a au moins besoin du nom de la machine et la date pour pouvoir aller récupérer dans la base de données les horaires de départ et de fin pour chaque faction.

Ainsi, dans mon script, j'ai fait en sorte d'appeler distinctement le script de création de graphique grâce à cURL.

Le résultat obtenu prenant beaucoup de place, je l'ai intégré à l'annexe 6.

Une fois les fonctionnalités précédentes mises en place, Mr Chrisment m'a alors demandé d'étudier la possibilité de rendre l'impression des graphiques automatique.

7) L'impression automatique.

Cette partie du projet a été pour moi non seulement une découverte de la puissance d'un langage de programmation mais aussi d'en voir les limites, c'est une partie qui m'a posé beaucoup de problème et qui m'a nécessité beaucoup de réflexion.

Dans un premier temps, j'ai d'abord cherché à générer plusieurs graphiques de manière automatique, j'ai donc fait une boucle qui allait de 1 à 3 (pour les factions) et appelé le script de génération de graphique. Or, cette méthode ne posait pas de problème pour générer deux graphiques de suite, mais, à partir du 3^{ème} les données étaient complètement faussées et tout le rendu n'avait aucun sens.

J'ai donc décidé de réutiliser la bibliothèque cURL afin de générer des graphiques, seulement, dans cette fonctionnalité, on ne cherche pas à afficher les graphiques sur un page mais à en faire un fichier qui pourra ensuite être imprimable.

C'est pourquoi cette partie m'a pris beaucoup de temps, j'ai dû me renseigner sur les possibilités de cURL, mais aussi comment faire pour pouvoir créer une page HTML à partir d'un résultat cURL et enfin réussir à convertir toutes les pages HTML en PDF* et les rassembler dans un seul document.

Les solutions que j'ai trouvées à ces problèmes ont été la méthode `shell_exec ()` et l'objet* `DOMDocument` que je détaille en annexe 6.

Grâce à ces solutions, j'ai pu en dérouler la démarche suivante :

- ✚ Récupérer les données nécessaires.
- ✚ Construction de l'URL.
- ✚ Récupérer le résultat de cURL dans un fichier HTML (qu'on enregistre sur le serveur).
- ✚ Convertir le fichier HTML en PDF.
- ✚ Concaténer tous les PDF en un seul PDF final.

Afin de pouvoir réaliser tout cela, il fallait aussi des outils qui le permettent, or, le serveur étant sous Unix, il a été très facile de demander à Mr Chrisment d'ajouter les paquets nécessaires (n'ayant pas toujours les droits je ne pouvais pas le faire seul).

Les paquets que j'ai demandés à installer sont :

- ✚ `Wkhtmltopdf` (conversion HTML vers PDF).
- ✚ `Pdftk` (concaténer* des PDF).
- ✚ `Pdf2ps` (transformer un PDF en fichier PostScript).

Le dernier des paquets est intéressant, en effet, il se peut qu'une imprimante ne gère pas le format PDF, c'est pourquoi il faut au préalable le transformer en fichier PostScript.

Une fois tous ces outils et toute cette démarche faite, mon script d'impression a alors très bien marché et généré des graphiques convenables.

Cependant, à ce moment, il fallait encore lancer le script à la main, alors que le but était de rendre l'impression automatique, je me suis donc de nouveau renseigné sur internet.

J'ai alors découvert qu'il existait un programme sous Unix qui permettait d'éditer des tables appelées crontab. Ces tables permettent de définir pour un horaire et une date, une exécution de programme (voir annexe 7).

La possibilité de rendre l'appel automatique au script à permis de programmer une impression automatique tous les matins à 07h00 sur l'imprimante des responsables de l'atelier.

III.5 De nouveaux besoins.

1) Bien connaître les besoins.

Lorsque le script à commencé à marcher, nous avons effectué une nouvelle réunion avec les utilisateurs et Mr Chrisment afin de savoir quelles fonctionnalités pourraient être intéressantes d'intégrer.

Lors de la réunion nous avons conclu ce qui suit :

- ✚ Ajouter une possibilité de gérer les équipes.
- ✚ Ajouter le nom des conducteurs et du contremaître sur les graphes.
- ✚ Ajouter une consultation du calendrier des équipes.

Tous ces ajouts incluait de rajouter des tables dans la base de données, c'est pourquoi j'ai ajouté les tables suivantes :

- ✚ Equipe (Num_Equip, Nom_Contremaître).
- ✚ Conducteur (Nom_Machine, Num_conducteur, Eq1, Eq2, Eq3, Eq4, Eq5).
- ✚ Cycle (Num_Equip, date_deb, date_fin, cycle).
- ✚ CalendrierEquipe (Date, Faction_cycle1, Faction_Cycle2, Num_Equip, contremaître, CB25, CCB, CW23).

Après avoir ajouté toutes ces tables j'ai d'abord commencé par créer l'interface sur le site. Dans un premier temps je devais rajouter des boutons sur l'accueil, puis construire une page de gestion des conducteurs, une de gestion des contremaîtres. Toutes ces pages étant des pages administratives, je les ai donc aussi protégées par mot de passe (Ces écrans étant très similaires à la partie administration de la partie III.4, j'ai décidé de mettre toutes les pages écran en annexe 7).

Ces fonctionnalités ont été assez dures à implémenter, du fait que je n'avais pas de connaissances concernant l'organisation du calendrier des équipes et de comment il était créé.

C'est pourquoi avant tout développement, j'ai d'abord étudié comment était conçu le calendrier des équipes.

2) Etude du calendrier des équipes.

Le calendrier fonctionne de la façon suivante : il y a cinq équipes qui tournent sur un système de faction, chaque jour, trois équipes effectuent une faction (matin, après-midi, nuit) et deux équipes sont en repos (voir annexe 8).

Cependant, si peu de certitude ne me suffisait pas, j'ai donc décidé de pousser l'étude du calendrier plus loin et j'en ai déduit les choses suivantes :

- ✚ Le calendrier suit une logique du lundi qui succède aux trois semaines de vacances d'août de l'année N jusqu'aux vacances de juillet de l'année N+1.
- ✚ Pendant les trois semaines de vacances, chaque équipe ne doit pas faire plus de douze factions.
- ✚ Le nombre d'heures des équipes doit être également réparti sur une année.
- ✚ Les équipes tournent sous un système de cycle.

Les cycles :

Dans l'entreprise, les équipes tournent sous deux cycles, le premier cycle qui est quasiment celui de toute l'année tourne en 6-4-6-4-6-4 (6 factions, 4 repos) et le deuxième qui est seulement sur une période de l'année en 6-2-6-2-6-8 (6 factions, 2 repos, 6 factions, 2 repos, 6 factions, 8 repos).

Après avoir établi mes déductions, j'ai donc déterminé qu'il serait impossible de créer un planning automatiquement sur plusieurs années, en raison des vacances entre juillet et août. C'est pourquoi j'ai créé un écran qui permet d'initialiser le calendrier et un autre qui permet d'ajouter le calendrier d'une semaine, sachant que cela ne serait à faire qu'une seule fois par an (voir annexe 8).

Lorsque l'on initialise le calendrier, la table CalendrierEquipe se remplit et ainsi on peut récupérer tous les jours les noms de contremaître/conducteurs.

La page de consultation du calendrier de l'équipe permet comme pour la consultation des plannings, d'aller voir le planning d'une semaine selon la date saisie.

3) Finalité du projet.

Une fois que toutes les fonctionnalités ont été implémentées, nous avons pu observer le résultat en générant un graphique complet (figure 18) :

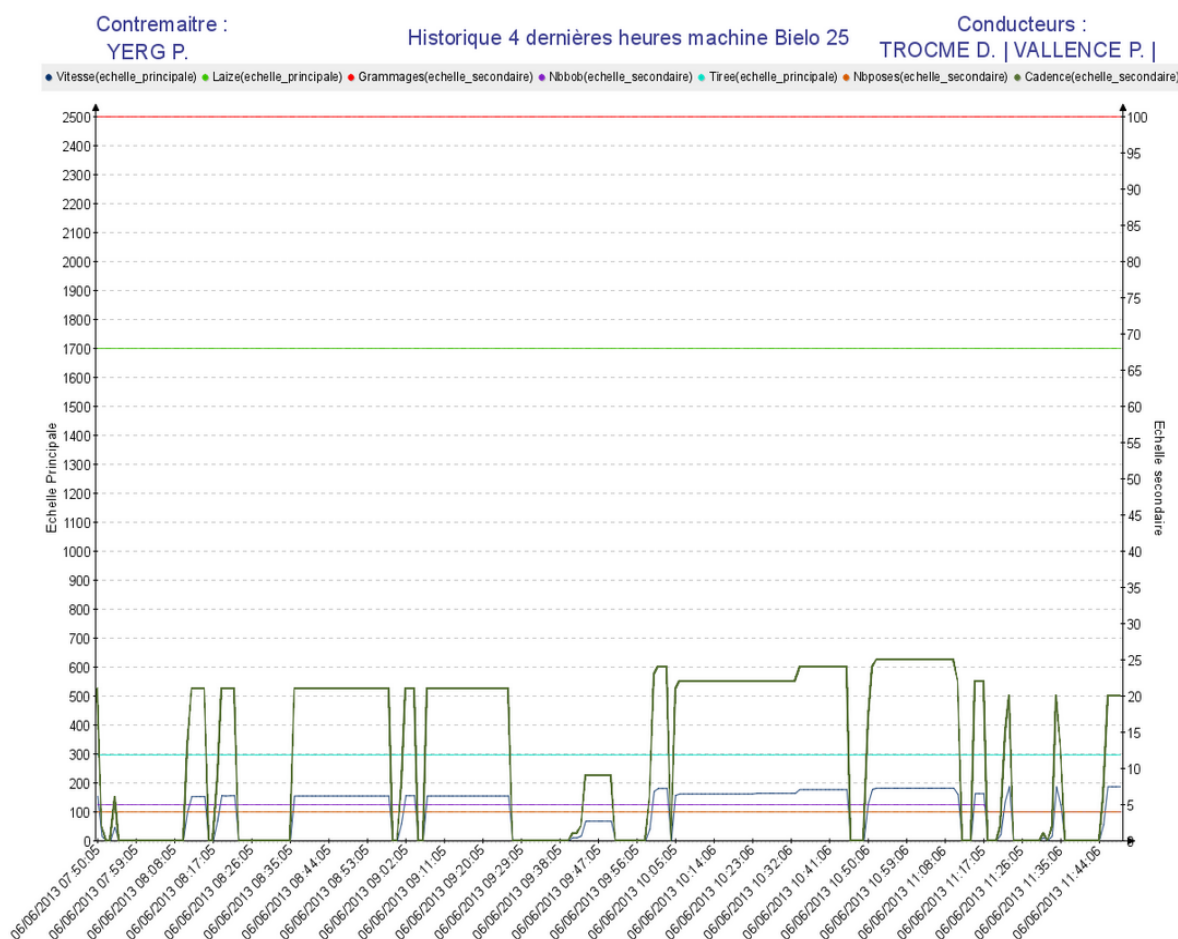


Figure 18 : graphique final.

4) Migration de la base de données locale vers la base serveur.

Une fois que le site web était en place et que toutes les fonctionnalités ont été intégrées, Mr Chrisment a importé la base locale sur le serveur directement, de cette façon, je n'avais plus qu'à changer certaines parties du programme afin de le réadapter.

Puis lorsque tout ceci a été fait, nous avons déployé le système sur le serveur de production qui lui, était déjà effectif (je travaillais sur un serveur à part).

III.6 Missions annexes.

Une fois que le projet a été terminé, Mr Chrisment m'a confié la mission de déployer le système sur un deuxième serveur, qui fonctionnait sous l'environnement Windows. Un serveur Windows étant beaucoup moins pratique qu'un serveur Unix, il m'a été impossible de créer le script d'impression depuis ce serveur.

Afin de pouvoir quand même imprimer le rapport des machines tous les matins, nous avons choisi avec Mr Chrisment d'appeler le script d'impression depuis le serveur de production, ainsi nous avons évité toutes les contraintes présentées par un serveur Windows.

Lors de mon stage, j'ai aussi été confronté à beaucoup de problèmes de fonctionnement, certains cas sont inconcevables au niveau théorique mais arrivent rapidement dans la réalité. C'est de cette façon que j'ai assuré une maintenance du site en compagnie de Mr Chrisment afin de pouvoir régler les problèmes qui arrivaient au fur et à mesure dans les meilleurs délais.

III.7 Bilan.

Ce projet à été pour moi, une expérience très instructive, non seulement pour les apports personnels que cela à engendré mais également pour tout ce que cela m'a fait apprendre au niveau du monde de l'entreprise dans le secteur de l'informatique, en effet, étant dans le service informatique directement, j'ai pu voir de mes propres yeux ce que pouvait être celui-ci dans une entreprise.

J'ai pu notamment approfondir mes connaissances dans différents langages de programmation (HTML/CSS, PHP), mais aussi voir tout le déroulement d'un projet au sein d'une entreprise.

IV. Conclusion

Cette expérience m'a en premier lieu permis d'approfondir mes connaissances dans des langages que je connaissais déjà (PHP, HTML/CSS) et d'apprendre par moi-même de nouveaux outils de programmation (jQuery, Ajax, pChart, JavaScript).

Ensuite, cela m'a également permis de comprendre que pour satisfaire pleinement les demandeurs, il faut régulièrement organiser des réunions avec ceux-ci pour avoir une idée très proche de leurs attentes. En effet, étant informaticien et donc avant tout programmeur, nous avons tendance à nous focaliser sur les fonctionnalités qui peuvent être développées, or, en ajoutant la contribution des utilisateurs, le résultat final intègre de bonnes fonctionnalités et une interface très simple.

Enfin, ce stage est la meilleure expérience professionnelle que j'ai eue tout au long de ma formation. Cette expérience en entreprise a révélé en moi une véritable passion pour le développement web, aussi bien dans l'analyse et la conception que dans la réalisation même d'un projet.

En dernier lieu, je tiens encore une fois à remercier Emmanuel Chrisment pour m'avoir suivi tout au long du stage et m'avoir laissé évoluer dans une autonomie complète ainsi que Fabrice APTEL sans qui tout cela n'aurait jamais été possible.

V. Bibliographie

Sites web consultés :

- + pChart 2.0, 10/04/13, <http://wiki.pchart.net/>.
- + JQuery, 20/05/13, <http://api.jquery.com/>.
- + cURL, 25/04/13, <http://fr2.php.net/manual/fr/book.curl.php>.

Rapports lus/consultés :

- + Halipré, Anthony, « création d'un site internet pour CrédoSanté », support internet, 2009/2010, DUT informatique, Université de Lille A, 66 pages.
- + Coste, Marie, « Réalisation d'un portail intranet pour la gestion du matériel réseau », archives Clairefontaine, 2009/2010, DUT informatique, Université Henri Poincaré Nancy 1, 44 pages.
- + Grebert, Vincent, « Archivage de données de production », archives Clairefontaine, 2002/2003, DUT informatique, Université Henri Poincaré Nancy 1, 42 pages.

VI. Glossaire

Chacun des termes cités ci-dessous est expliqué d'un point de vue informatique.

EXCEL : C'est un programme informatique qui permet de réaliser des tâches avec des feuilles de calculs.

HTML : Cela signifie HyperText Markup Language et c'est un format de données conçu pour représenter des pages web. Il permet d'organiser des données de façon sémantique.

CSS : Signifie feuille de style en cascade, on forme souvent le couple HTML/CSS car les deux langages sont très souvent utilisés ensemble dans le développement web, le CSS permet de définir des propriétés affectées à la page HTML.

PHP : Hypertext Preprocessor, c'est un langage de programmation qui permet de produire des pages web dynamiques.

JavaScript : C'est un langage de programmation qu'on utilise dans les pages web afin de les rendre interactives.

Intranet : C'est un réseau informatique utilisé à l'intérieur de l'entreprise qui fonctionne comme internet mais seulement au niveau de l'entreprise.

Macro : Une macro est l'association d'un texte de remplacement à un identificateur.

VBA : Visual Basic Application, c'est un langage de programmation spécifique à EXCEL qui permet de commander les fonctionnalités du logiciel.

Base de données : C'est un conteneur informatique qui permet de stocker l'intégralité des informations en rapport avec une activité. Elle permet de stocker et de retrouver un ensemble d'informations de plusieurs natures ainsi que les liens qui existent entre elles. Elle est constituée d'un ensemble de tables.

SQL : Structured Query Language (langage de requête structurée) est un langage informatique qui sert à effectuer des opérations sur des bases de données.

Serveur : C'est un dispositif informatique matériel ou logiciel qui offre des services, à différents clients, les services les plus courants sont : partage d'imprimantes, accès aux pages web...etc.

Client : Un client est le logiciel qui envoie des demandes au serveur. Par exemple pour l'accès à une page web, le logiciel client est un navigateur internet (Mozilla Firefox, Google Chrome...etc.).

Table : Une table est constituée de lignes et de colonnes, chaque colonne correspond à un champ de données. Chaque ligne correspond à un enregistrement.

Cookie : C'est un petit fichier texte enregistré sur le disque dur du client par le serveur web, son utilisation se fait surtout pour mémoriser certaines données concernant des utilisateurs.

Librairie (ou bibliothèque) : C'est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans les réécrire.

pChart : C'est une librairie en PHP qui permet de créer facilement des graphiques sur un serveur web, il y a notamment la possibilité de l'afficher directement dans le navigateur ou de l'enregistrer dans un fichier PDF.

URL : Uniform Resource Locator est un format de nommage universel pour définir une ressource sur internet (dans notre cas intranet).

Variable de session : Elle consiste à la mémorisation de certaines données concernant un utilisateur, cependant ce n'est pas comme un cookie, le cookie garde en mémoire les informations pour une durée donnée tandis que la variable de session existe à partir du moment où l'utilisateur se connecte et jusqu'à ce qu'il ferme la connexion (quitter le navigateur).

jQuery : C'est une bibliothèque qui porte sur l'interaction entre JavaScript et HTML, elle permet surtout de simplifier des commandes communes à JavaScript.

cURL : C'est une bibliothèque en PHP qui permet de récupérer le contenu d'une ressource accessible par un réseau informatique. La ressource est désignée à l'aide de l'URL.

PDF : Qui signifie Portable document format, est un format de fichier qui permet de conserver les styles définis par le créateur quelque soit le logiciel ou système d'exploitation utilisé.

Objet : Un objet est un conteneur symbolique, qui possède sa propre existence et incorpore des informations et des mécanismes.

Concaténer : C'est le fait d'assembler plusieurs choses pour n'en faire plus qu'une.

Script : Programme informatique qui ne nécessite pas d'être compilé avant d'être exécuté. Pour fonctionner, les scripts doivent être interprétés par un programme ou un serveur dédié au langage dans lequel ils ont été écrits.

VII. Annexes

Table des annexes.

Annexe 1	41
1) L'histoire de Clairefontaine	41
2) L'histoire du papier	42
3) Les matières premières	43
4) La fabrication du papier	44
5) L'atelier finition	45
6) La qualité Clairefontaine	45
7) Environnement et écologie	46
8) Quelques chiffres	47
9) Les ressources humaines	47
Annexe 2	48
Annexe 3	49
1) Redirection HTML	49
2) Protection par mot de passe	49
3) Fonction pour récupérer la valeur d'un paramètre	50
4) Sélecteur de machine	50
5) Intégration du cookie	51
Annexe 4	53
1) La bibliothèque pChart	53
2) L'objet DateTime	53
3) Gestion des axes	54
4) Gestion des échelles	55
Annexe 5	56
1) Une table en HTML	56
2) Une table qui se remplit dynamiquement	56
3) La bibliothèque JQuery	58
4) Fonctions temporelle	60

Annexe 6	62
1) La librairie cURL	62
2) Shell_exec() et l'objet DOMdocument	64
Annexe 7	65
1) La table de cron	65
2) Deuxième partie administrative.....	65
Annexe 8	68
1) Le calendrier	68
2) Bien gérer le calendrier	68

Annexe 1

1) L'Histoire de Clairefontaine.

Tout commence en **1858** quand Jean-Baptiste Bichelberger choisit, pour y installer une fabrique de papier en fonctionnement continu sur le site d'Etival. Celui-ci réunit les conditions les plus favorables. L'eau, indispensable à toutes les étapes de la production, est fournie par la rivière qui assure en même temps une source d'énergie suffisante. De plus, la présence d'un moulin à papier permet à l'entrepreneur de s'entourer immédiatement d'ouvriers qualifiés. Ainsi naissent, avec la souscription de différents actionnaires, les « Papeteries de Clairefontaine ».

En **1877**, Paul Bichelberger et Emile Champon, introduisent l'utilisation du bois comme matières premières. Ensemble, ils lancent la fabrication des enveloppes et cahiers.

De **1904 à 1912** Louis Nusse et Etienne Bodet dirigent ensemble l'entreprise dont les effectifs atteignent plus de 1100 personnes.

Léon Daridan fait redémarrer l'usine en **1919**. Il en assume également la reconstruction après la seconde guerre mondiale.

C'est sous l'impulsion, de Charles Nusse que Clairefontaine se développe considérablement de **1950 à 1971** notamment grâce à l'installation de machines à papier modernes. C'est à lui que la marque doit aussi son célèbre logo, devenu depuis synonyme de qualités pour les spécialistes de l'industrie papetière, comme pour le grand public.

Clairefontaine aujourd'hui, dirigée par Frédéric Nusse son P.D.G, propose plus de 3000 produits destinés aux écoliers, aux étudiants ainsi qu'aux adultes et aux entreprises. Des cahiers piqués, brochés ou spiralés aux copies perforées, des répertoires aux agendas en passant par les papiers bureautiques, on retrouve la qualité de fabrication qui fait le renom de la marque.

Clairefontaine est aujourd'hui le seul fabricant à faire lui-même et à garantir son propre papier pour ses produits, ce papier extra blanc et très résistant, est réputé pour son extrême douceur à l'écriture.

2) L'histoire du papier.

En l'an **105**, T'Sai LUN mandarin du régime impérial chinois, codifie la technique de fabrication du papier. Celle-ci avait été inventée dans son pays, 3 siècles plus tôt. Le processus de fabrication (artisanale) se faisait à partir de fibre de mûrier et bambou.

Pour les Arabes, la fabrication du papier à base de chiffon arrive en Europe au 10^{ème} siècle. Les moulins à papier s'y développent notamment à partir de l'invention de l'imprimerie par Gutenberg vers **1440**.

Le pilon à maillet, invention italienne datant du 11^{ème} siècle permet en utilisant la force motrice de l'eau, d'améliorer le broyage des chiffons et ainsi augmente la productivité des premiers moulins à papier.

C'est de **1799** que date la grande révolution de l'industrie papetière avec la réalisation de la première machine à papier en continu : **La machine Robert**.

Inventée par le français Louis Nicolas Robert, elle est le prototype dont s'inspirent les installations modernes.

Schéma de fonctionnement :

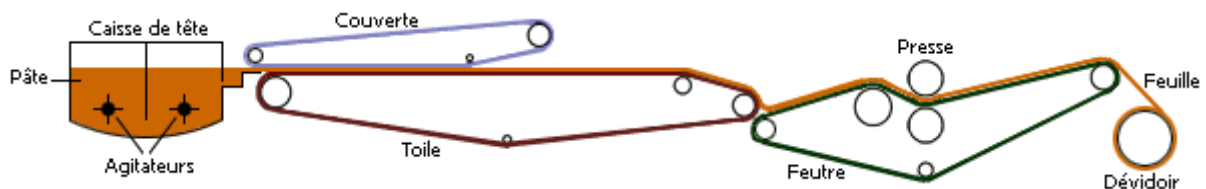
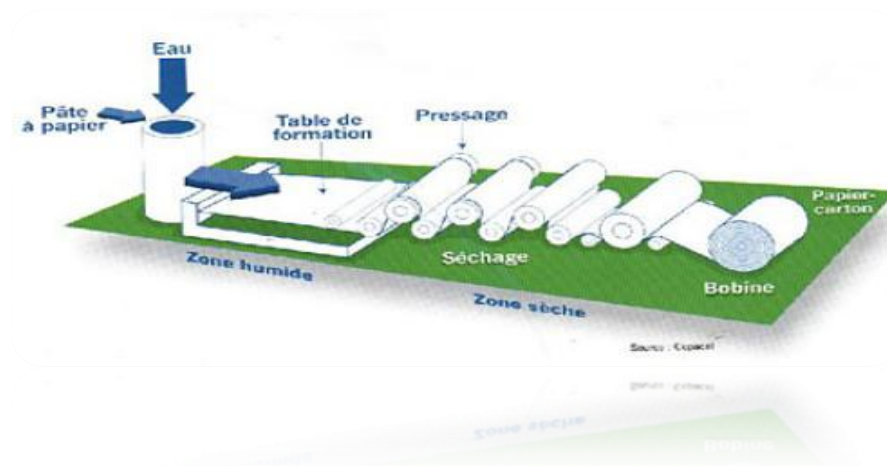


Schéma machine à papier (moderne) :



3) Les matières premières.

Fibres courtes et longues :

Le papier se compose essentiellement de fibres de bois : des fibres de feuillus, appelées fibres courtes qui confèrent au papier les propriétés d'imprimabilité, d'opacité, et des fibres de résineux, appelées fibres longues, qui apportent la résistance au papier. Grâce à un mélange subtil entre ces deux types de fibres, on obtient un papier de qualité.

Les pâtes à papier :

Les pâtes à papier, qui ont l'aspect de carton blanchi, sont dirigées vers le pulpeur et mélangées à de l'eau. Des lames métalliques en rotation permettent aux fibres de se séparer les unes des autres. Puis suit le raffinage : les fibres subissent un frottement violent, qui entraîne leur fibrillation et leur coupe pour les plus longues.

Ce traitement a pour but d'augmenter les caractéristiques mécaniques du papier et de donner une belle formation de feuille. Des colles et des colorants sont alors ajoutées pour apporter aux papiers ses caractéristiques d'imprimabilité. De même, des charges minérales, fabriquées dans l'usine sont jointes pour donner de l'opacité et de l'éclat.

Balles de pâtes :



4) La fabrication du papier.

Pour transformer la pâte, Clairefontaine dispose de deux machines à papier de 3,40 mètres de largeur. Les vitesses de production atteignent 900 mètres par minutes pour l'une, et 500 mètres pour l'autre. Chaque année, plus de 165 000 tonnes de papier sont ainsi fabriquées sur le site d'Etival-Clairefontaine. Des améliorations sont sans cesse apportées sur ces deux machines afin d'assurer en permanence le plus haut niveau de qualité.

A son arrivée dans la machine, la pâte contient 99% d'eau. Elle est projetée sous la forme d'un jet parfaitement plat et égal sur toute la largeur d'une toile en mouvement.

La suspension fibreuse obtenue vient s'égoutter, par aspiration à travers deux toiles. Cette opération a pour but d'obtenir une parfaite symétrie et une teinte identique sur les deux faces de la feuille, propriété essentielle pour l'impression recto-verso du papier.

Tout l'art du pressage consiste, grâce au passage de la feuille entre des presses et des feutres absorbants, à extraire un maximum d'eau en limitant la diminution d'épaisseur du papier. A l'issue du pressage, le séchage est effectué au contact de dizaines de cylindres chauffés à la vapeur.

La feuille est imprégnée d'amidon, afin d'améliorer l'imprimabilité, tout en maintenant un profil d'humidité régulier.

En sortie de machine, le papier est lissé pour obtenir un état de surface correspondant à son utilisation, puis bobiné sur un mandrin d'acier. Il formera ainsi les bobines dites « mères ».

Ces bobines sont ensuite découpées en bobine « filles » de différentes largeurs, suivant les besoins des ateliers auxquels elles sont destinées.

Machine à papier :



Bobine mère :



5) L'atelier finition.

Cet atelier est l'atelier auxquels j'ai été affecté, il consiste comme son nom l'indique dans la finition du papier. Chaque machine à un but précis, certaines machines serviront à découper le papier, certaines le trierons...etc.

Coupeuse et conditionnement ramettes :



Chaque machine de l'atelier est équipée d'un capteur, celui-ci renvoi des données qui sont enregistrés toutes les minutes dans la base de données.

6) La qualité Clairefontaine.

Pour l'entreprise qui bénéficie de la norme ISO 9001 depuis 1997, la qualité elle l'affaire de tous dans l'entreprise.











La qualité de l'entreprise repose sur les choses suivantes :

- ✚ Une stricte sélection des matières premières.
- ✚ Des contrôles à la réception.
- ✚ Des moyens de production.
- ✚ Des jauges de mesures en ligne et des systèmes de régulation.
- ✚ Par des prélèvements de papier.
- ✚ Par la formation et l'engagement de tous.

7) Environnement et écologie.

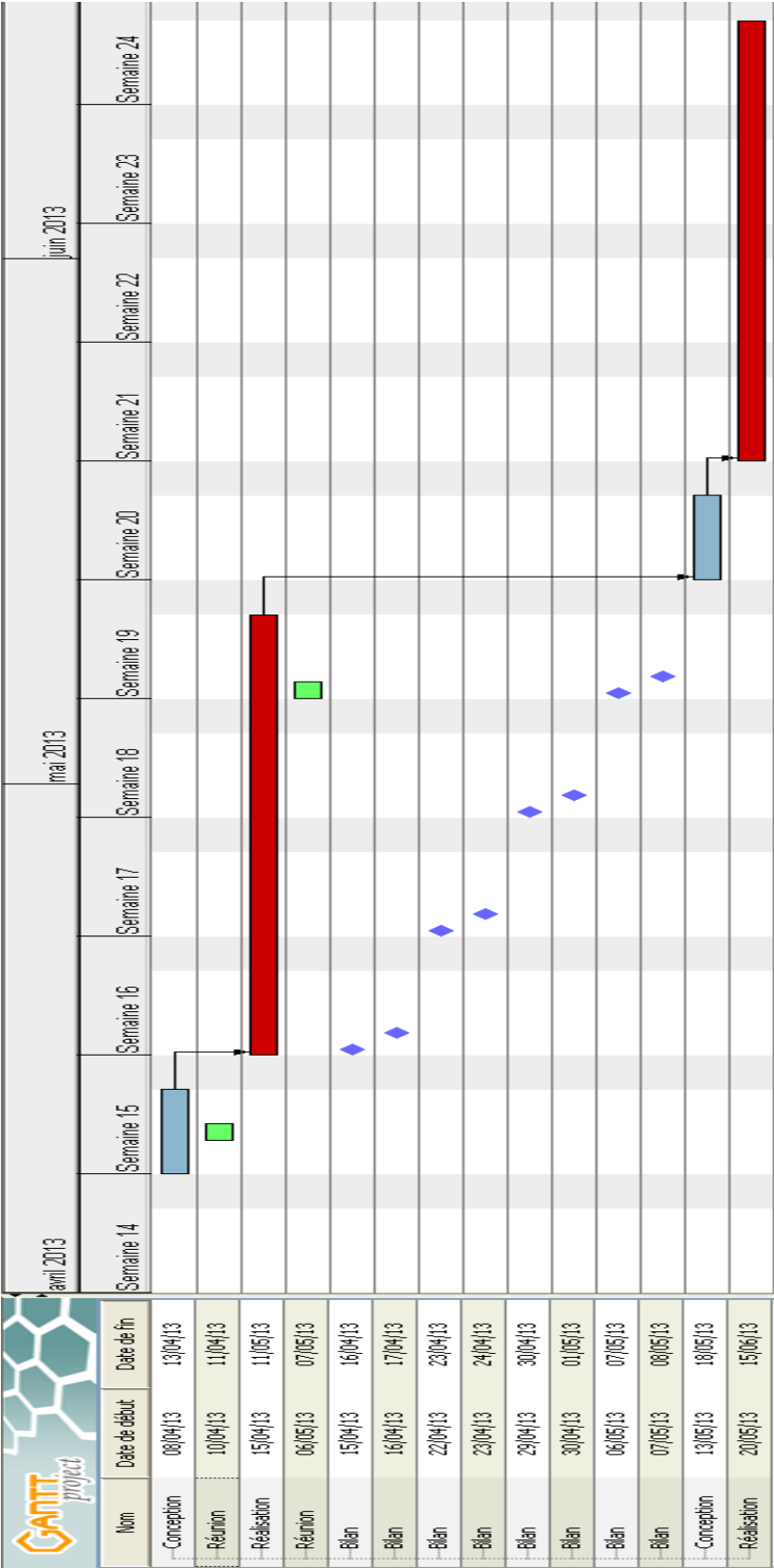
Clairefontaine n'a pas attendu la tendance actuelle pour intégrer l'écologie dans sa stratégie d'entreprise.

Résumons leurs actions en quelques dates :

-  **1974** : Clairefontaine a adopté la production des ses papiers en pH neutre.
-  **1981** : Précurseur dans le traitement biologique des eaux de production avant leur retour à la rivière, Clairefontaine reçoit en 1988 le « Trophée de l'eau » pour le bassin Rhin-Meuse.
-  **1985** : Utilisation du gaz naturel pour fournir en vapeur les machines à papier. La combustion permet de supprimer les résidus de soufre.
-  **1992** : Mise en place de la première unité de carbonate de calcium précipité (PPC) en réutilisant le CO₂ émis par nos chaudières.
-  **1994** : Clairefontaine favorise l'utilisation des fibres recyclées (vieux papiers) au travers de son usine d'Everbal.
-  **1997** : L'usine est une des premières à posséder une installation de cogénération de vapeur et d'électricité.
-  **1999** : Mise en service de la première unité de compostage des boues de station d'épuration pour supprimer les nuisances d'odeur et d'aspect.
-  **2001** : Grâce aux moyens mis en place pour le respect de l'environnement, Clairefontaine bénéficie de la certification internationale ISO 14001.
-  **2002** : Clairefontaine met tout en œuvre pour limiter progressivement l'émission de composés organiques volatils et de substances dangereuses dans les encres, les colles et les autres solutions utilisées pour l'impression. En 2008, l'objectif est atteint à 80%.
-  **2005** : Les pâtes à papier proviennent exclusivement de forêts certifiées pour leur gestion environnementale (PEFC, FSC). Celle-ci favorise la croissance des arbres et permet l'absorption du gaz carbonique.

Annexe 2

Diagramme de Gantt :



Annexe 3

1) Redirection HTML.

```
<td><a id = "verif" href = "gerer/gere_param.php">Gestion des paramètres.<a></td>
<td><a id = "planning" href = "planning.php">Planning machines.</a></td>
```

2) Protection par mot de passe.

```
$query = "Select mot_de_passe from Password";
$result = mssql_query($query) or die ("Query failed " . mssql_get_last_message());
$ligne = mssql_fetch_array($result,MSSQL_ASSOC);
$line = $ligne['mot_de_passe'];
```

```
<body>
<?php
if((isset($_POST['pass']) && !empty($_POST['pass']) && $_POST['pass'] == $line)
{
?>
```

Code HTML

```
<?php
}
else
{
?>
    <div id = "mdp"> Quel est le mot de passe ?</div>
    <form name = "protect" id = "protect" method = "POST" action = "">
    <input type = "password" name = "pass" value = "">
    <input type = "submit" name = "Valid_protect" value = "Valider" >

    </form>
<?php
}
?>
```

Ecran correspondant :

Quel est le mot de passe ?

3) Fonction pour récupérer la valeur d'un paramètre.

```
<td>
<select name = <?php echo "Vitesse_" . $i ;?>>
<option <?php getParam('Vitesse',$i,'Principale');?>>Principale</option>
<option <?php getParam('Vitesse',$i,'Secondaire');?>>Secondaire</option>
<option <?php getParam('Vitesse',$i,'Ignorer');?>>Ignorer</option>
<option <?php getParam('Vitesse',$i,'Indisponible');?>>Indisponible</option>
</select>
</td>
```

```
function getParam($param,$id,$chaine)
{
    $query = "Select " . $param. " from Parametres where Id_machine=" . $id;
    $result = mssql_query($query) or die ("Query failed " . mssql_get_last_message());
    $line = mssql_fetch_array($result,MSSQL_ASSOC);
    $ligne = $line["$param"];
    $cmp=strcmp($ligne,$chaine);
    if($cmp == 0)
    {
        echo "selected='selected'";
    }
    mssql_free_result($result);
}
```

4) Sélecteur de machine.

```
<select style = "width : 135px;" name = "Machine" id = "Machine" onChange =
"this.form.submit()" >

<?php $query = "Select * from ConfMachines";
    $result = mssql_query($query);
    while($ligne = mssql_fetch_array($result))
    {
        ?>
        <option <?php $machine = $ligne['Machine'] ; $chaine = $_SESSION['Machine'] ; $cmp =
        strcmp($machine,$chaine) ; if($cmp==0){echo "selected='selected'";}?>> <?php echo
        $ligne['Machine'] ;?></option>
    }
    ?>
</select>
```

5) Intégration du cookie.

```
$query = "Select mot_de_passe from Password";
$result = mssql_query($query);
$ligne = mssql_fetch_array($result);
$line = $ligne['mot_de_passe'];
mssql_free_result($result);
// On test si le cookie est existant ou si le mot de passe a été rentré
if((isset($_POST['pass']) && !empty($_POST['pass'])) OR (isset($_COOKIE['pass']) &&
!empty($_COOKIE['pass'])))
{
    if(isset($_POST['pass']) && !empty($_POST['pass']))
    {
        if($_POST['pass'] == $line)
        {
            $cook = $_POST['pass'];
            setcookie("pass",$cook,time()+600);
        }
    }
}
```

Ce rajout impliquait d'établir un nouveau test pour chaque page protégée :

```
<body>
<?php
if((isset($_POST['pass']) && !empty($_POST['pass']) && $_POST['pass'] == $line) ||
(isset($_COOKIE['pass']) && !empty($_COOKIE['pass'])))
{
?>
```

De cette façon, j'ai pu tester s'il le cookie existait ou s'il était rentré pour la première fois.

Annexe 4

1) La bibliothèque pChart.

Code minimal pour crée un graphique :

```
include("pChart/class/pData.class.php");
include("pChart/class/pDraw.class.php");
include("pChart/class/pImage.class.php");

// Création des données
$mydata = new pData();
// Remplissage des données
$mydata->AddPoint(array(1,2,3,4,5,6,7,8,9,10,11,12,13), 'Test_échelle');
$mydata->SetSerieOnAxis('Test_échelle',0);
$mydata->addpoints(array(8,15,28,14,03), 'abscisse');
$mydata->setAbscissa('abscisse');
// Création de l'image et implémentation des données
$myimg = new pImage(800,600,$mydata);
$myimg->drawScale(); // Il est possible de définir une échelle personnalisée
// Rendu de l'image
$myimg->drawLineChart();
$myimg->Render('graphe.png');
```

La fonction drawLineChart() permet de dessiner une courbe, cependant le paramètre de certaines machines était histogramme. La bibliothèque proposait une multitude de modèle à dessiner et notamment des histogrammes :

```
$myimg->drawBarChart();
```

Il est aussi possible d'ajouter du texte sur le graphique :

```
$myimg->drawText(420,35,'Exemple de texte écrit en coordonnées(420;35)');
```

Et enfin, il est possible d'enregistrer le fichier sur le serveur comme dans le code minimal ou alors d'afficher directement le rendu dans un navigateur web :

```
$myimg->Stroke();
```

2) L'objet DateTime.

Comme je l'évoque dans mon rapport, chaque machine est réglée selon le format GMT, or, nous sommes en GMT + 2, je me suis donc servi de quelque méthode de l'objet DateTime afin d'établir les bonnes mesures.

Changement de fuseau horaire :

```
$w_deb = new DateTime();  
$w_deb->setTimezone(new DateTimeZone('GMT'));
```

Affichage de la date et du temps au format particulier :

```
$w_deb->format('Y/m/d H:i:s') ; // affiche par exemple-> 2013/05/01 12:00:00  
$w_deb->format('Y-m-d H:i:s') ; // affiche par exemple-> 2013-05-01 12:00:00  
$w_deb->format('d/m/Y H:i:s') ; // affiche par exemple-> 01/05/2013 12:00:00
```

Cet objet m'a énormément servi lors de ce stage, pour pouvoir établir mes requêtes dans la base de données par exemple.

3) Gestion des axes.

```
// Gestion de la sélection des courbes  
// Pour la vitesse  
$query = "Select Vitesse from Parametres where Nom_machine = " . "" . $machine . "" ;  
$result = mssql_query($query);  
$ligne = mssql_fetch_array($result, MSSQL_ASSOC);  
$lineV = $ligne['Vitesse'];  
mssql_free_result($result);
```

Traitement pour le paramètre Vitesse :

```
// On va chercher sur quel axe mettre chaque paramètres  
$cmp_P = strcmp($lineV, "Principale");  
$cmp_S = strcmp($lineV, "Secondaire");  
if($cmp_P == 0) // Si principale  
{  
    $myData->addPoints($vitesse, "Vitesse(echelle_principale)");  
    $serieSettings = array("R"=>4, "G"=>47, "B"=>104);  
    $myData->setPalette("Vitesse(echelle_principale)", $serieSettings);  
    $myData->setSerieOnAxis("Vitesse(echelle_principale)", 0); // Alors on le met sur l'axe principale  
}  
else if($cmp_S == 0) // Sinon si c'est secondaire  
{  
    $myData->addPoints($vitesse, "Vitesse(echelle_secondaire)");  
    $serieSettings = array("R"=>4, "G"=>47, "B"=>104);  
    $myData->setPalette("Vitesse(echelle_secondaire)", $serieSettings);  
    $myData->setSerieOnAxis("Vitesse(echelle_secondaire)", 1); // Axe secondaire et sinon on ajoute pas la données  
}
```


4) Gestion des échelles.

On récupère d'abord les données depuis la base :

```
$query_settings = "Select Echelle_principale,Echelle_secondeaire from Parametres where  
Nom_machine='$machine' ";  
$result_settings = mssql_query($query_settings);  
$ligne = mssql_fetch_array($result_settings,MSSQL_ASSOC);  
$principale = $ligne['Echelle_principale'];  
$secondeaire = $ligne['Echelle_secondeaire'];
```

Puis on effectue les tests :

Si les deux échelles sont définies :

```
if($principale != NULL && $secondeaire != NULL)  
{  
$AxisBoundaries=  
array(0=>array("Min"=>0,"Max"=>$principale),1=>array("Min"=>0,"Max"=>$secondeai  
re)  
Settings=array("Mode"=>SCALE_MODE_MANUAL,"ManualScale"=>$AxisBoundaries) );  
}
```

Sinon si la principale existe mais pas la secondeaire :

```
else if($principale != NULL && $secondeaire == NULL)  
{  
$AxisBoundaries = array(0=>array("Min"=>0,"Max"=>$principale));  
$Settings =  
array("Mode"=>SCALE_MODE_MANUAL,"ManualScale"=>$AxisBoundaries);  
}
```

Sinon si la principale n'existe pas et la secondeaire existe :

```
else if($principale == NULL && $secondeaire != NULL)  
{  
$AxisBoundaries = array(1=>array("Min"=>0,"Max"=>$secondeaire));  
$Settings =  
array("Mode"=>SCALE_MODE_MANUAL,"ManualScale"=>$AxisBoundaries);  
}
```






Sinon cela signifie que les deux échelles sont nulles :

```
else  
{  
$Settings = array("Mode"=>SCALE_MODE_START0); // Définition auto de l'échelle  
}
```

Annexe 5

1) Une table en HTML.

Une table en html dispose de plusieurs balises :

-  `<table>` : Qui permet de définir la structure de la table.
-  `<thead>` : Qui représente l'entête de la table.
-  `<th>` : Qui s'intègre au thead pour définir en général des noms de colonnes.
-  `<tr>` : Pour définir les limites d'une ligne.
-  `<td>` : Pour définir chaque colonne d'une ligne.

Exemple :

```
<table id = "calendrier">
<thead>
  <th style="border : none;"></th>
  <th>Lundi</th>
  <th>Mardi</th>
  <th>Mercredi</th>
  <th>Jeudi</th>
  <th>Vendredi</th>
  <th>Samedi</th>
  <th>Dimanche</th>
</thead>
<tr id = "lignedeb">
  <td style = "border:none;"> </td>
  <td>Début / Fin </td>
  <td>Début / Fin </td>
  <td>Début / Fin </td>
  <td>Début / Fin </td>
  <td>Début / Fin </td>
  <td>Début / Fin </td>
  <td>Début / Fin </td>
</tr>
```

2) Une table qui se remplit dynamiquement.

Comme je l'évoque dans mon rapport, la table HTML qui établit le journal des erreurs chaque jour est rempli de manière dynamique. Ce qui évite de prendre une place trop importante sur la page lorsqu'il y a beaucoup de données.

Code de la table dynamique :

```

<?php
$w_deb = new DateTime();
$w_deb->modify("-". $nbh. " hour");
$w_deb->setTimezone(new DateTimeZone('GMT'));
$ws_deb = date_format($w_deb, 'Ymd H:i:s');
$query = "SELECT [IDEvent],[Date] FROM [OPCData].[dbo].[Events] WHERE [Date]
BETWEEN ". $ws_deb . " AND GETDATE() AND [IDMachine] = ". $Idmachine . " ORDER BY [Date] ASC";
$result = mssql_query($query);
$nb_lignes = mssql_num_rows($result);
?>

<div id="Journal_des_log">
    <table id="table_log">
        <?php
        echo
        "<thead>
        <th> Date de l'evenement </th>
        <th> Description de l'evenement</th>
        </thead>" ;
        while($line = mssql_fetch_array($result,MSSQL_ASSOC))
        {
            // Gestion de la description de l'event
            $Id_event = $line['IDEvent'];
            $date = $line['Date'];
            $date_courante = new DateTime($date,new DateTimeZone('GMT'));
            $date_courante->setTimezone(new DateTimeZone('Europe/Paris'));
            $date_courante->format('d/m/Y H:i:s');
            $query_event = "Select [Description] FROM [OPCDATA].[dbo].[ConfEvents]
            WHERE IDMachine = ". $Idmachine . " AND IDEvent = ". $Id_event ;
            $result_event = mssql_query($query_event);
            $nb_lignes_event = mssql_num_rows($result_event);
            if($nb_lignes_event > 0) // Si on a des événements
            {
                while($line_event = mssql_fetch_array($result_event,MSSQL_ASSOC)) // Tant qu'on
                en a
                {
                    echo "<tr><td>" . $date_courante->format('d/m/Y H:i:s') . "</td> ". "<td>" .
                    $line_event['Description'] . "</td></tr>" ; // On les affiche
                }
            }
        }
    }
?>
</table>
</div>

```

On obtient alors le résultat suivant :

Journal des erreurs du 10-06-2013 07:18:47 au 10-06-2013 11:18:47 sur la machine Bielo 25

Date de l'événement	Description de l'événement
10/06/2013 07:37:45	Incident emballage ligne 1
10/06/2013 07:50:23	Incident emballage ligne 1
10/06/2013 07:56:09	Incident emballage ligne 1
10/06/2013 08:13:51	Incident emballage ligne 1
10/06/2013 08:44:18	Incident emballage ligne 1
10/06/2013 09:39:06	Surveillance papier devant coupeuse transversale
10/06/2013 09:42:59	Surveillance papier après coupeuse transversale
10/06/2013 09:44:49	Surveillance cylindre pression
10/06/2013 09:57:38	Porte protection couteau circulaire
10/06/2013 09:59:00	Surveillance cylindre pression
10/06/2013 09:59:35	Porte protection couteau circulaire
10/06/2013 10:01:43	Surveillance papier devant coupeuse transversale
10/06/2013 10:10:30	Surveillance papier devant coupeuse transversale
10/06/2013 10:13:20	Confirmation portes de protection couteaux circulaires
10/06/2013 10:13:29	Confirmation portes de protection couteaux circulaires
10/06/2013 10:15:08	Porte protection service puits fermée
10/06/2013 10:19:32	Aspiration des rognures ne marche pas
10/06/2013 10:30:53	Incident emballage ligne 2
10/06/2013 10:39:27	Incident emballage ligne 1

3) La bibliothèque jQuery.

Cette bibliothèque regroupe toutes les fonctionnalités de JavaScript, le principe de la bibliothèque est très simple, écrire le moins de code possible pour faire le maximum de choses.

C'est notamment grâce à cette bibliothèque que j'ai pu insérer un champ de saisie interactif sur la consultation des plannings :

<script>

```
$(function() {  
$( "#datepicker" ).datepicker();  
});  
  
jQuery(function($){  
$.datepicker.regional['fr-CH'] = {  
closeText: 'Fermer',  
prevText: '&#x3C;Préc',  
nextText: 'Suiv&#x3E;',  
currentText: 'Courant',  
monthNames: ['Janvier','Février','Mars','Avril','Mai','Juin',  
'Juillet','Août','Septembre','Octobre','Novembre','Décembre'],  
monthNamesShort: ['Jan','Fév','Mar','Avr','Mai','Jun',  
'Jul','Aoû','Sep','Oct','Nov','Déc'],
```

```

    dayNames: ['Dimanche', 'Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi'],
    dayNamesShort: ['Dim', 'Lun', 'Mar', 'Mer', 'Jeu', 'Ven', 'Sam'],
    dayNamesMin: ['Di', 'Lu', 'Ma', 'Me', 'Je', 'Ve', 'Sa'],
    weekHeader: 'Sm',
    dateFormat: 'dd/mm/yy',
    firstDay: 1,
    isRTL: false,
    showMonthAfterYear: false,
    yearSuffix: '';
    $.datepicker.setDefaults($.datepicker.regional['fr-CH']);
  });
</script>

```

```

<p id = "Date">Date: <input value = <?php $date = getDateCourante(); echo $date; ?>
type="text" id="datepicker" onChange = "this.form.submit()" name =
"datepicker"/></p>

```

```

function getDateCourante()
{
    if(isset($_POST['datepicker']) && !empty($_POST['datepicker']))
    {
        $date = $_POST['datepicker'];
    }
    else
    {
        $date = getdate();
        if($date['mon'] < 10)
        {
            $date['mon'] = "0" . $date['mon'];
        }
        if($date['mday'] < 10)
        {
            $date['mday'] = "0" . $date['mday'];
        }
        $date = $date['mday'] . "/" . $date['mon'] . "/" . $date['year'];
    }

    return $date ;
}

```

Voici le résultat :

Date:

Début  **Juin 2013** 

Lu	Ma	Me	Je	Ve	Sa	Di
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

4) Fonctions temporelle.

// Fonction qui renvoie le numero de semaine a partir de la date courante au format JJ/MM/AAAA

```
function date_format_fr()
{
    if(isset($_POST['datepicker']) && !empty($_POST['datepicker']))
    {
        $date = $_POST['datepicker'];
        $numweek = dateFR2WeekNum("$date");
    }
    else
    {
        $date = getdate();
        $jour = $date['mday'] ;
        $mois = $date['mon'];
        $annee = $date['year'] ;
        $numweek = dateFR2WeekNum("$jour/$mois/$annee");
    }
    return $numweek ;
}
```

// Fonction qui renvoi le numéro de semaine à partir d'une date

```
function dateFR2WeekNum($date)
{
    list($day, $month, $year) = explode('/', $date);
    $weekNum = (date('W', mktime(0, 0, 0, $month, $day, $year)) * 1);
    return $weekNum;
}
```

// Fonction qui renvoi le numéro de jour à partir d'une date

```
function getNumJour($date)
{
    list($day, $month, $year) = explode('/', $date);
    $numjour = date('w', mktime(0,0,0,$month,$day,$year));
    return $numjour ;
}
```

// Fonction qui renvoi le lundi de la semaine à partir d'une date

```
function getLundiDate($date)
{
    list($day,$month,$year) = explode('/', $date);
    $date = new DateTime("$month/$day/$year");
    $format = $date->format("d/m/Y");
    $num = getNumJour($format);
    switch($num)
    {
        case 0:
            $date->modify("-6 day");
    }
```

```

    break;

    case 2:
    $date->modify("-1 day");
    break;

    case 3:
    $date->modify("-2 day");
    break;

    case 4:
    $date->modify("-3 day");
    break;

    case 5:
    $date->modify("-4 day");
    break;

    case 6:
    $date->modify("-5 day");
    break;
}

return $date->format("d/m/Y");
}

```

Annexe 6

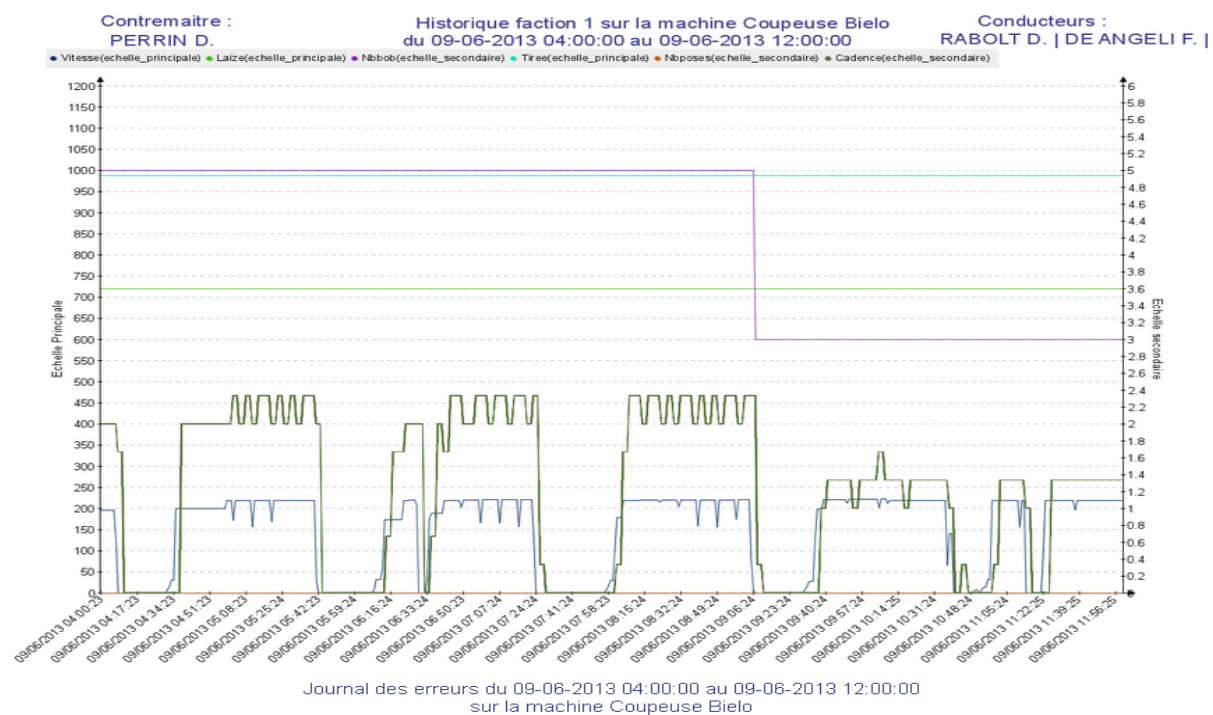
1) La librairie cURL.

Comme je l'introduis dans mon rapport, cette bibliothèque permet de récupérer la source d'une page à distance, afin de bien comprendre comment cela fonctionne, je vais tout d'abord mettre le code correspondant au fait évoqué dans le rapport puis le résultat obtenu :

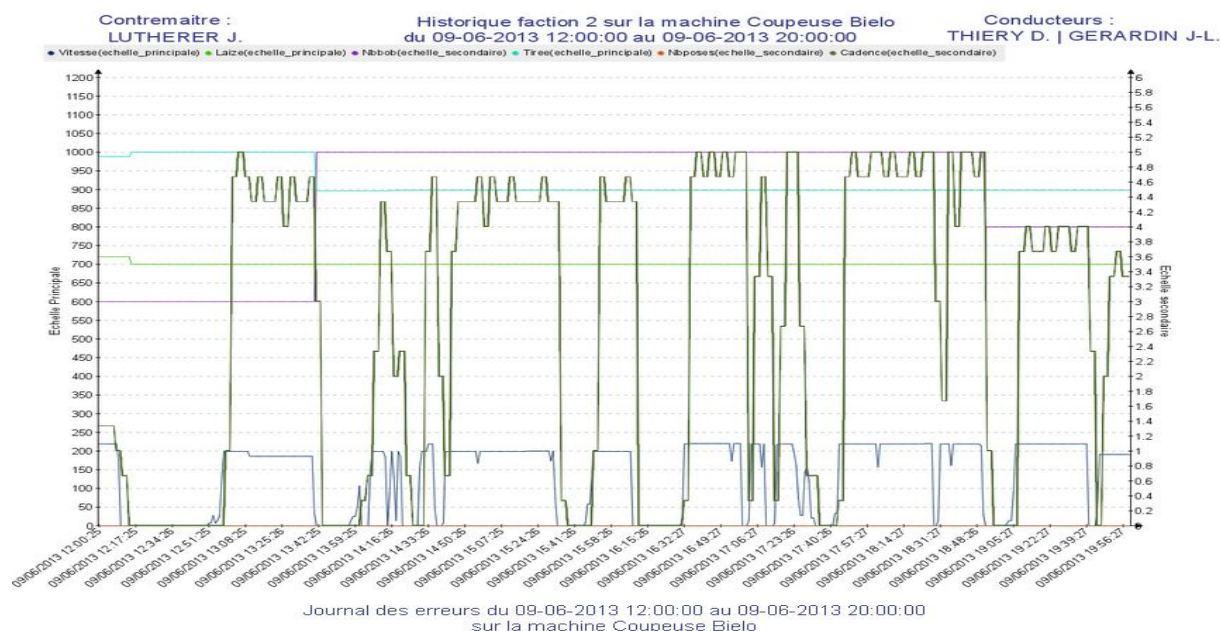
```
$date = new DateTime();
$dateconstruite = new DateTime($_GET['date']);
$date = $date->format('Ymd');
$dateconstruite = $dateconstruite->format('Ymd');

if(isset($_GET['idjour']) && isset($_GET['m']) && !empty($_GET['idjour']) &&
!empty($_GET['m']) && isset($_GET['date']) && !empty($_GET['date']) &&
$dateconstruite <= $date)
{
    $numjour = $_GET['idjour'];
    $machine = $_GET['m'];
    $date = $_GET['date'];
    for($i = 1 ; $i < 4 ; $i++)
    {
        $curl = curl_init();
        $query_sql = "Select Heure_D,Heure_F from Planning where
Nom_machine='$machine' AND Id_faction='.$i.'" AND Id_Jour = ".$numjour ;
        $result_sql = mssql_query($query_sql);
        $ligne_sql = mssql_fetch_array($result_sql,MSSQL_ASSOC);
        $heured= convert($ligne_sql['Heure_D']);
        $heuref= convert($ligne_sql['Heure_F']);
        $param = $date ;
        $machine_param = urlencode($machine);
        // Url construit pour les graphique automatique.
        $url = "http://sclf-
ocs/finition/graphe_machine.php?m=$machine_param&hd=$heured&hf=$heuref&bool
=false&date=$param&idfaction=$i";
        curl_setopt($curl,CURLOPT_URL,$url);
        curl_setopt($curl,CURLOPT_RETURNTRANSFER,true);
        $res = curl_exec($curl);
        if(curl_exec($curl) == false)
        {
            echo "Curl Error : " . curl_error($curl);
        }
        curl_close($curl);
        mssql_free_result($result_sql);
        echo $res . "<br /><br />";
    }
}
```

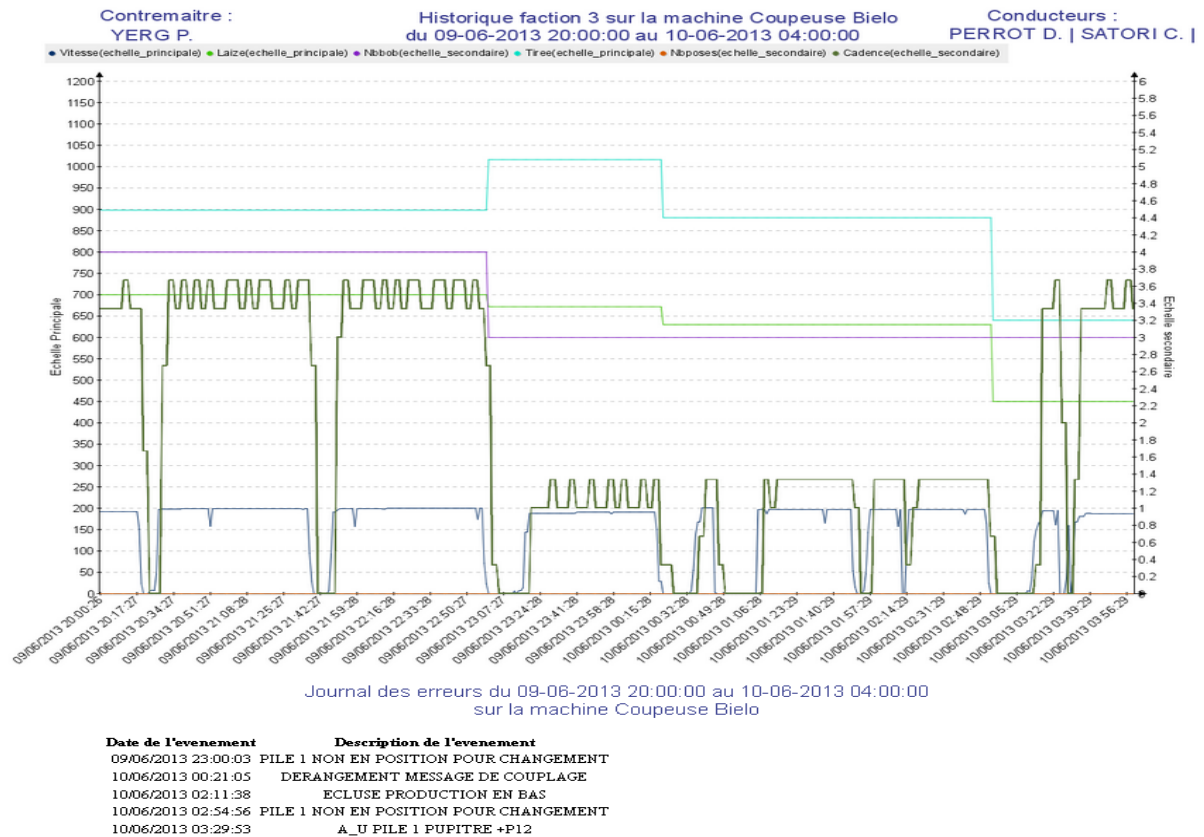

Exemple de résultat :



Date de l'événement	Description de l'événement
09/06/2013 04:07:28	SURVEILLANCE COUPEUSE TRANSVERSALE TRAIN
09/06/2013 04:30:31	SURVEILLANCE COUPEUSE TRANSVERSALE
09/06/2013 06:28:46	ARRET D'URGENCE CHEVAUCHEMENT
09/06/2013 06:32:35	EV RUBANS D'EXTRACTION AVANCE
09/06/2013 09:29:18	ROTATION TABLE NON EN POSITION
09/06/2013 09:29:27	ROTATION TABLE NON EN POSITION
09/06/2013 11:13:29	ECLUSE PRODUCTION EN BAS



Date de l'événement	Description de l'événement
09/06/2013 12:51:45	SURVEILLANCE COUPEUSE TRANSVERSALE TRAIN
09/06/2013 14:01:33	ECLUSE PRODUCTION EN BAS
09/06/2013 14:17:47	DERANGEMENT MESSAGE DE COUPLAGE
09/06/2013 14:20:23	A_U PILE 1 PUPITRE +P810
09/06/2013 14:36:17	SURVEILLANCE DE PORTE PILE 2
09/06/2013 16:59:07	A_U PILE 1 PUPITRE +P12
09/06/2013 17:10:10	PHOTO ELECTRONIQUE CHEVAUCHEMENT
09/06/2013 18:27:52	FDC BORRAGE AV AIGUILLAGE PILE 1 OU 2
09/06/2013 18:52:11	SURVEILLANCE COUPEUSE TRANSVERSALE TRAIN
09/06/2013 19:01:50	SURVEILLANCE COUPEUSE TRANSVERSALE
09/06/2013 19:39:55	ECLUSE PROTECTION EN BAS



2) Shell_exec() et l'objet DOMdocument.

Shell_exec() : C'est une fonction qui permet d'exécuter une commande comme ci celle-ci avait été faites depuis un invite de commande (ici on travaille sous Linux donc sous un shell).

C'est grâce à cette fonction que j'ai pu effectuer l'impression automatique depuis le programme PHP.

DOMdocument : C'est un objet PHP qui représente le corps d'une document HTML, dans mon projet je me suis uniquement servi de la méthode LoadHTML(nomfichier) à partir d'un résultat cURL pour pouvoir récupérer la source HTML et la métho SaveHTMLFile(nomfichier) pour enregistrer le fichier sur le serveur au format HTML.

Annexe 7

1) La table de cron.

Une crontab permet de planifier des exécutions de programme à des dates/heures précises, pour faire cela, il faut en outre utiliser une certaine syntaxe :

mm hh jj MMM JJJ tâche > log

Dans cette syntaxe :

- ✚ mm représente les minutes (de 0 à 59).
- ✚ hh représente les heures (de 0 à 23).
- ✚ jj représente le numéro de jour du mois (de 1 à 31).
- ✚ MMM Représente le numéro du mois (de 1 à 12) ou l'abréviation du mois (jan,fev,mar...etc).
- ✚ JJJ représente l'abréviation du jour ou son numéro dans la semaine (0 = Dimanche, 1 = Lundi).
- ✚ Enfin log représente un fichier dans lequel stocker le journal des opérations.

Ainsi pour programmer l'impression automatique j'ai utilisé la configuration cron suivante :

00 7 * * * * wget -q -O - http://localhost/finition/impression_auto.php >/dev/null 2>&1 → Ici nous imprimerons donc tous les jours les courbes à 7h00 du matin.

La commande wget permet de télécharger un fichier depuis le web ce qui permet d'appeler automatiquement le script d'impression automatique.

2) Deuxième partie administrative.

Cette partie se compose de plusieurs pages écran :

- ✚ Page de gestion des équipes.
- ✚ Page de gestion des conducteurs.
- ✚ Page de gestion des cycles.
- ✚ Page de consultation du planning des équipes (non protégée).

Page de gestion des équipes :



Gestion des équipes

N° Equipe	Nom du contremaître
1	BACHETTA L.
2	LUTHERER J.
3	GEHRIG S.
4	YERG P.
5	PERRIN D.

Valider

Réinitialiser

[Accueil](#)

[Gérer les conducteurs](#)

[Gérer les plannings](#)

[Initialiser le planning](#)

[Gérer les cycles](#)

Page de gestion des conducteurs :



Gestion des conducteurs machines.

Machine	N° Conducteur	Conducteur equipe 1	Conducteur equipe 2	Conducteur equipe 3	Conducteur equipe 4	Conducteur equipe 5
Bielo 25	1	LAZAROWICZ J.	ETIENNE J.	L HOTE D.	TROOME D.	MATHIEU A.
Bielo 25	2	OCANA J.	BRUNSPERGER F.	CLAUDE S.	VALLENCE P.	HUMBERT F.
Coupeuse Bielo	1	PETITNICOLAS D.	THIERY D.	LACHAMBRE D.	PERROT D.	RABOLT D.
Coupeuse Bielo	2	VERJUS G.	GERARDIN J-L.	BERNARD J-P.	SATORI C.	DE ANGELI F.
Will 23	1	MICLOT	ROCHEL	FREMINET P.	PERRIN A.	FAYS L.
Will 23	2	IDOUX F.	BIRKER	BAGARD L.	LENARS J-M.	CHENAL E.

Valider

Réinitialiser

[Accueil](#)

[Retourner à la gestion des équipes](#)

Page de gestion des cycles :



Gestion du cycle

N° Equipe	Date Début	Date de fin
1	01/03/2013	29/05/2013
2	05/03/2013	02/06/2013
3	27/02/2013	27/05/2013
4	03/03/2013	31/05/2013
5	07/03/2013	04/06/2013

Valider

Réinitialiser

[Accueil](#)

[Retourner à la gestion des équipes](#)

Les dates a renseignés sont les dates durant laquel les Factions sont sur les cycle 6-2-6-2-6-8.

Page de consultation du planning des équipes :



Planning des équipes

Date :

Planning du 10/06/2013 au 16/06/2013

Jour	Date	Eq1	Eq2	Eq3	Eq4	Eq5
Lundi	10/06/2013	R2	A2	R4	N2	M2
Mardi	11/06/2013	R3	N1	M1	R1	A1
Mercredi	12/06/2013	R4	N2	M2	R2	A2
Jeudi	13/06/2013	M1	R1	A1	R3	N1
Vendredi	14/06/2013	M2	R2	A2	R4	N2
Samedi	15/06/2013	A1	R3	N1	M1	R1
Dimanche	16/06/2013	A2	R4	N2	M2	R2

[Accueil](#)

Annexe 8

1) Le calendrier.

2013	BACETTA	LEVERRE	GERRIC	RIE	PERRIN	ZATORT	SERIALI	BOUCHERLE	2013	BACETTA	LEVERRE	GERRIC	RIE	PERRIN	ZATORT	SERIALI	BOUCHERLE	2013	BACETTA	LEVERRE	GERRIC	RIE	PERRIN	ZATORT	SERIALI	BOUCHERLE
N° Equipe	Eq 1	Eq 2	Eq 3	Eq 4	Eq 5				N° Equipe	Eq 1	Eq 2	Eq 3	Eq 4	Eq 5				N° Equipe	Eq 1	Eq 2	Eq 3	Eq 4	Eq 5			
Janvier	1 M								Mars	1 V								Mai	1 M							
2 M									2 S									2 J								
3 J									3 D									3 V								
4 V									4 L									4 S								
5 S									5 M									5 D								
6 D									6 M									6 L								
7 L									7 J									7 M								
8 M									8 V									8 M								
9 M									9 S									9 J								
10 J									10 D									10 S								
11 V									11 L									11 S								
12 S									12 M									12 D								
13 D									13 M									13 L								
14 L									14 J									14 M								
15 M									15 V									15 M								
16 M									16 S									16 J								
17 J									17 D									17 V								
18 V									18 L									18 S								
19 S									19 M									19 D								
20 D									20 M									20 L								
21 L									21 J									21 M								
22 M									22 V									22 M								
23 M									23 S									23 J								
24 J									24 D									24 V								
25 V									25 L									25 S								
26 S									26 M									26 D								
27 D									27 M									27 L								
28 L									28 J									28 M								
29 M									29 V									29 J								
30 M									30 S									30 M								
31 J									31 D									31 V								
Février	1 V								Avril	1 L								Juin	1 S							
2 S									2 M									2 D								
3 D									3 M									3 L								
4 L									4 J									4 M								
5 M									5 V									5 M								
6 M									6 S									6 J								
7 J									7 D									7 V								
8 V									8 L									8 S								
9 S									9 M									9 D								
10 D									10 M									10 L								
11 L									11 J									11 M								
12 M									12 V									12 M								
13 M									13 S									13 J								
14 J									14 D									14 V								
15 V									15 L									15 S								
16 S									16 M									16 D								
17 D									17 M									17 L								
18 L									18 J									18 M								
19 M									19 V									19 J								
20 M									20 S									20 M								
21 J									21 D									21 V								
22 V									22 L									22 S								
23 S									23 M									23 D								
24 D									24 M									24 L								
25 L									25 J									25 M								
26 M									26 V									26 J								
27 M									27 S									27 V								
28 J									28 D									28 S								
29 J									29 L									29 M								
30 J									30 M									30 D								

2) Bien gérer le calendrier.

Pour gérer le calendrier, j'ai créé deux pages :

- Une page d'initialisation du calendrier.
- Une page d'ajout de calendrier hebdomadaire.

La page d'initialisation permet d'ajouter la plus grosse période de l'année comme je l'explique dans mon rapport.

La page pour ajouter un calendrier hebdomadaire permet de gérer la particularité des cas des vacances d'août, comme je l'évoque dans mon rapport, chacune des factions ne doit pas effectuer plus de douze factions pendant les trois semaines de vacances.

Page d'initialisation :



Initialisation des équipes.

N° Equipe	Faction de départ	Date de départ	Date de fin
1	M1		
2	M1		
3	M1		
4	M1		
5	M1		

[Accueil](#) [Retourner à la gestion des équipes](#)

Pour initialiser le planning, il faut indiquer par quelle faction commence les équipes à la date de reprise après les 3 semaines de vacances d'août.
Il faut mettre en date de fin la date du début des vacances de l'année prochaine.
C'est une opération à effectuée une seule fois par an.

Page d'ajout :



Gestion du planning des équipes

Planning du 10/06/2013 au 16/06/2013

Date :

Jour	Date	Eq1	Eq2	Eq3	Eq4	Eq5
Lundi	2013-06-10	R2	A2	R4	N2	M2
Mardi	2013-06-11	R3	N1	M1	R1	A1
Mercredi	2013-06-12	R4	N2	M2	R2	A2
Jeudi	2013-06-13	M1	R1	A1	R3	N1
Vendredi	2013-06-14	M2	R2	A2	R4	N2
Samedi	2013-06-15	A1	R3	N1	M1	R1
Dimanche	2013-06-16	A2	R4	N2	M2	R2

[Accueil](#) [Retourner à la gestion des équipes](#)

Cette page inclut de l'Ajx, celui-ci permet de rafraîchir des éléments sur la page sans recharger entièrement la page, autrement dit :

Lorsque je change la date, le planning de la semaine est automatiquement rafraîchit et de cette façon je suis persuadé de modifier le bon planning.