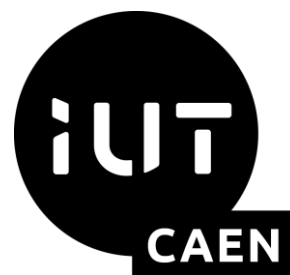




UNIVERSITÉ
CAEN
NORMANDIE



Université de Caen Normandie
Institut Universitaire de Technologie de Caen
Département Informatique

Diplôme Universitaire de Technologie
INFORMATIQUE

Gestion de fichier d'écritures comptables

Mémoire de conduite de projet de 2^e année

Lucas LECOMTE
Xavier DE CASENEUVE
Charly COLOMBU

Commanditaire : Marc LEVILLY

Tuteurs du projet : Fabienne JORT et Marc LEVILLY

Jury : Eric PORCQ et Philippe BRUTUS

Année universitaire 2017 / 2018

Gestion de fichiers d'écritures comptables

Le but de notre projet est de concevoir une application de gestion de fichiers d'écritures comptables (FEC) pour M. Levilly. Chaque année, toutes les entreprises regroupent les entrées et sorties d'argent dans un fichier. Le travail de M.

Levilly est de vérifier l'authenticité des données qui y sont inscrites. Notre application permettra de transférer un fichier

FEC vers une base de données, pour que les utilisateurs puissent faire des recherches en fonction de certains critères ou effectuer des tests sur certaines données.

The goal of this project is to build a web application that will manage accounting entry file (FEC in French) for Mr. Levilly. Every year, each company gather their accounting entry into those accounting entry file and the work of Mr. Levilly is to verify the authenticity of the written data. Our web application will allow him to transfer an accounting entry file into a database, so that he will later be able to search some entry with specific criteria or run some test on specific entries.

Mots-clés

Écritures comptables

Comptabilité

Fichier d'Écritures comptables

Base de données

Logiciel

Application web

Gestion de fichier d'écritures comptables

Remerciements:

Nous tenons à remercier Mme.Jort pour son investissement dans ce projet, pour nous avoir aidé à prendre certaines décisions et de manière générale pour nous avoir motivé à nous engager pleinement dans la réalisation de ce projet. Nous remercions également M.Levilly pour avoir répondu à nos questions concernant l'aspect comptabilité de ce projet. Nous souhaitons remercier aussi Mme.Hrarti-Drafate de nous avoir transmis un cours concernant le langage XML en première année et M.Porcq pour avoir répondu à nos questions concernant notre base de données .

Sommaire

Introduction	6
1 Début du projet	7
1.1 Compréhension du projet	7
1.2 Répartition des tâches	8
1.2.1 Diagrammes de Gantt	8
1.2.2 Backlog	10
1.3 Choix des outils	11
1.3.1 Choix des langages	11
1.3.2 Choix des logiciels	11
2. Réalisation du projet	12
2.1 Le transfert du fichier FEC	12
2.1.1 L'algorithme	12
2.1.2 Les problèmes rencontrés	12
2.1.3 Les solutions apportés	15
2.1.4 Les parties importantes du code	16
2.2 Les tests	18
2.2.1 Présentation des tests	18
2.2.2 Les algorithmes	19
2.3 Les formulaires	22
2.3.1 Fonctionnement	22
2.3.2 Le traitement PHP	23
2.4 L'interface	24
Conclusion	26
Bibliographie	28
Annexe I - Diagramme de cas d'utilisation	29
Annexe II - SNI	30
Annexe III - Maquettes	31
Annexe IV - Le code permettant de mettre les données dans les variables d'insertion	32
Annexe V - Calcul de taux de TVA	34
Annexe VI - Code du traitement PHP d'un formulaire	35
Annexe VII - Fiche de lecture	37

Introduction

A la fin de chaque année, toutes les entreprises doivent remettre un bilan de leur activité à l'état. Ce bilan prend la forme d'un Fichier d'Écritures Comptable (surnommé FEC). M. Levilly, notre commanditaire de projet, est expert-comptable ainsi que directeur général dans l'entreprise Cogediac depuis 2013. Cogediac est une entreprise spécialisée dans le secteur des activités comptables créée en 1993 et située à Caen.

En tant qu'expert-comptable, tous les ans, M. Levilly reçoit des fichiers d'écritures comptables de la part de ses clients. Ces fichiers répertorient la totalité des entrées et sorties d'argent d'une entreprise accompagnées d'une certaine quantité de détails, ils sont donc volumineux et difficile à traiter. Dans son travail M. Levilly est amené à travailler sur ces fichiers pour vérifier leur authenticité. Il nous a donc demandé de réaliser une application permettant de traiter ces données. Quelques logiciels existent déjà et présentent les mêmes fonctionnalités que notre application. Cependant M. Levilly souhaite savoir quel traitement est effectué sur les données envoyées à l'application et c'est pour cette raison que ce projet a été proposé.

Comment construire une application web de gestion de fichiers d'écritures comptables ergonomique et faisant appel à une base de données sachant que nous n'avons que très peu de connaissances en comptabilité mais que nos tuteurs de projets disposent des connaissances nécessaires pour nous apporter de l'aide ?

Nous nous pencherons dans un premier temps sur les prémices du projet, nous expliquerons comment nous avons débuté ce projet, la manière dont nous nous sommes répartis les tâches et le choix des outils pour réaliser ce projet. Ensuite nous exposerons la réalisation du projet en détail, parties par parties. Nous parlerons donc du transfert de FEC, des tests de validation, des formulaires de recherche et enfin de l'interface.

Le début de ce projet a surtout été de comprendre l'architecture d'un fichier d'écritures comptables en attendant d'en recevoir un anonymisé de la part de notre commanditaire afin de pouvoir créer la base de données pour accueillir les données du fichier. Les projets ont commencé fin Janvier et nous avons reçu le fichier le 4 Mai 2017. Cette première partie représente donc 4 mois de travail.

1 Début du projet

1.1 Compréhension du projet

Notre premier travail après le début du projet en première année a principalement été la compréhension du sujet. Ce projet requiert des notions d'économie et de comptabilité, il nous a fallu comprendre ces notions mais aussi comprendre la constitution d'un fichier FEC. Le fichier FEC est composé de 22 champs obligatoires, nous les avons répartis en 4 familles. Il y a ceux qui concernent la ligne d'écriture, la pièce d'écriture, le compte et le journal.

Le journal correspond au cahier dans lequel est rangé la ligne d'écriture, il peut par exemple y avoir un journal des achats ou un journal des ventes. Le compte correspond aux comptes bancaires utilisés pour la transaction. La pièce est l'élément permettant de justifier l'écriture comptable comme un ticket de caisse. L'écriture correspond à la transaction elle-même. Elle peut être un débit ou un crédit.

Comprendre le fichier FEC et le répartir en quatre familles nous a permis de créer le modèle relationnel des données sur le logiciel WinDesign. Elle était le deuxième point le plus important de ce début de projet car sans celle-ci nous n'aurions tout simplement pas pu réaliser le projet, le principe de notre application étant de fonctionner à partir d'une base de données remplie avec les données d'un fichier FEC.

Afin de bien comprendre le projet, notre principale tâche au départ était de réaliser des diagrammes pour faciliter la compréhension. Tout d'abord le diagramme des cas d'utilisation qui allait nous permettre de savoir les différentes fonctionnalités que les utilisateurs auront, grâce aux informations de notre commanditaire (cf. Annexe I). Ensuite, le SNI et les maquettes qui allaient nous permettre de réaliser l'application, que ce soit dans le design ou bien les transitions des fonctionnalités (cf. Annexe II et Annexe III).

Le principal problème de la gestion de FEC est la taille de ces fichiers. Plus l'entreprise est grande plus le fichier est volumineux. Ces fichiers peuvent donc atteindre des millions de lignes dans certains cas. L'intérêt premier de ce projet est donc de permettre à l'utilisateur de gérer et d'étudier facilement ces fichiers.

1.2 Répartition des tâches

Une fois la base de données conceptualisée et les concepts des FEC en tête, nous nous sommes réparti les tâches. Pour cela nous avons pris les fonctionnalités de l'application qui nous semblaient être les plus importantes et nous les avons répartis en trois.

L'un d'entre nous s'est donc occupé du transfert des données du fichier FEC vers la base de données. Il s'agit de l'élément central du projet qui sert de liens vers les autres parties.

Un autre s'est occupé d'implémenter des tests sur les données établis par le commanditaire. Cette partie est cruciale au projet puisque ces tests permettent l'analyse en profondeur du fichier. Ces tests sont souvent les révélateurs des anomalies ou des tentatives de fraudes.

Le dernier s'est occupé de créer les différents formulaires de recherche requis par le commanditaire. Cette partie, au même titre que la précédente, est essentielle à ce projet. Elle permet une analyse de surface mais aussi de se familiariser en explorant le fichier FEC.

Pour ce qui est de la partie restante, c'est à dire l'interface, nous avons décidé de nous en occuper en dernier mais avant tout ensemble pour éviter de créer des interfaces différentes chacun de notre côté et donc de perdre du temps.

1.2.1 Diagrammes de Gantt

Les diagrammes de Gantt permettent de représenter l'état d'avancement d'un projet. Nous avons créé une première version de ce diagramme au deuxième semestre. A ce moment le projet était encore dans sa première phase. Il est, par conséquent, peu précis sur la partie réalisation. Nous ne savions pas que nous serions amenés à nous séparer le travail de la manière dont nous l'avons fait. Cependant nous avons tout de même correctement prévu le fait d'avoir fini de concevoir les outils d'analyse en Décembre 2017. Nous avons fait une nouvelle version de ce diagramme à la fin du semestre 4. On peut voir sur cette version que nous nous sommes éparpillés un peu plus que prévu dans notre travail.

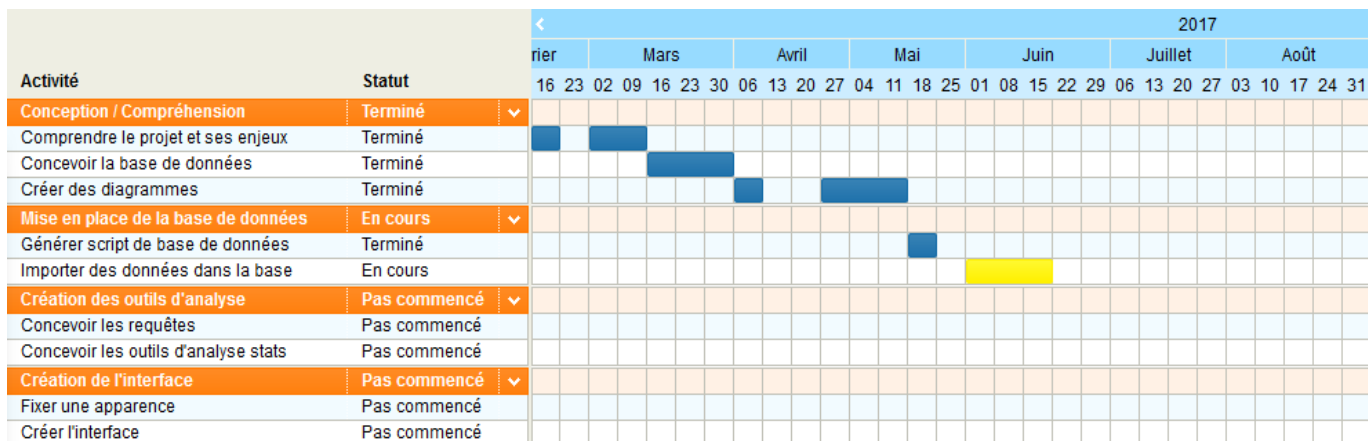


Diagramme de Gantt S2 - Février à Août 2017

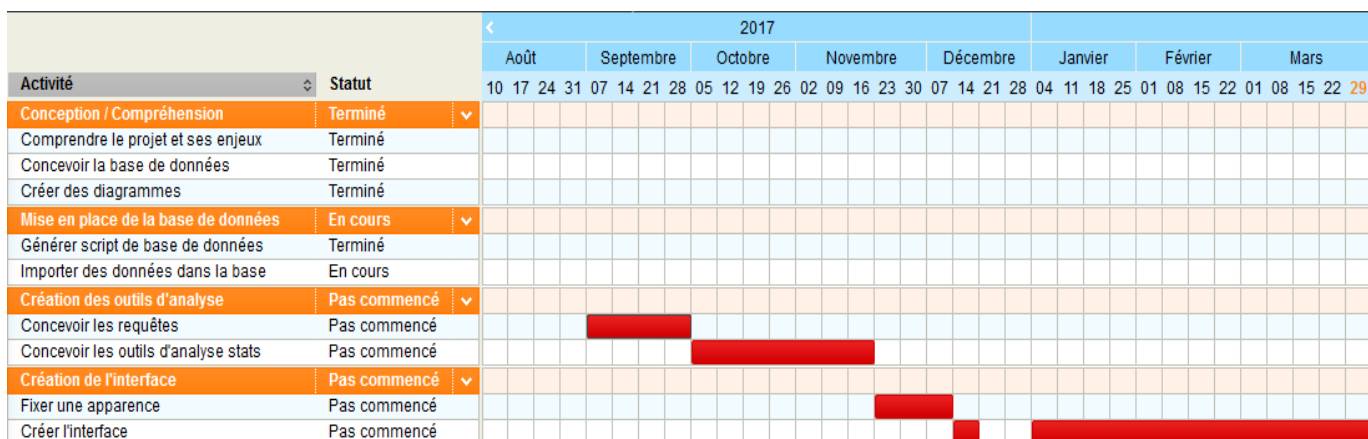


Diagramme de Gantt S2 - Septembre 2017 à Mars 2018

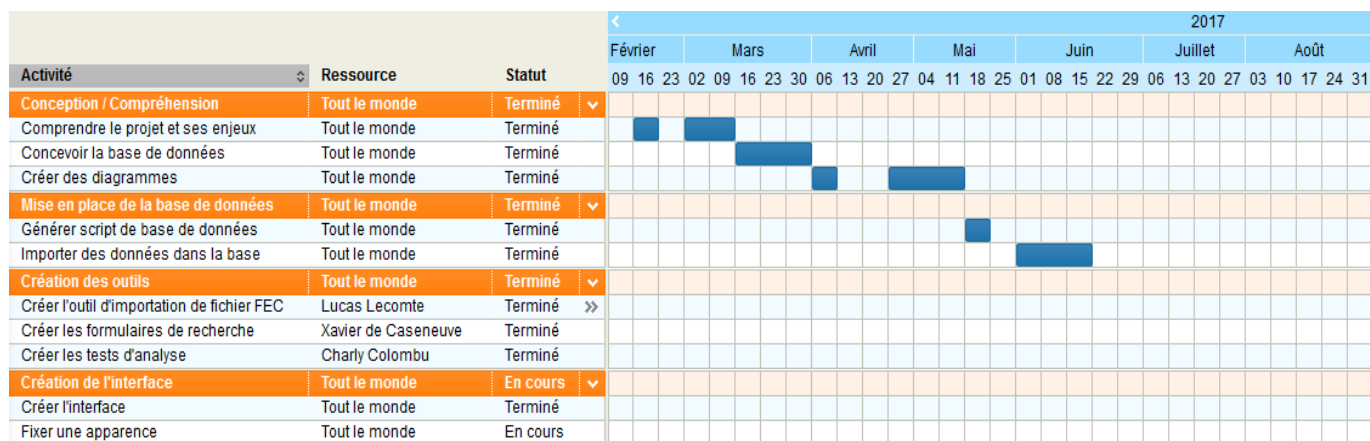


Diagramme de Gantt S4 - Février à Juin 2017

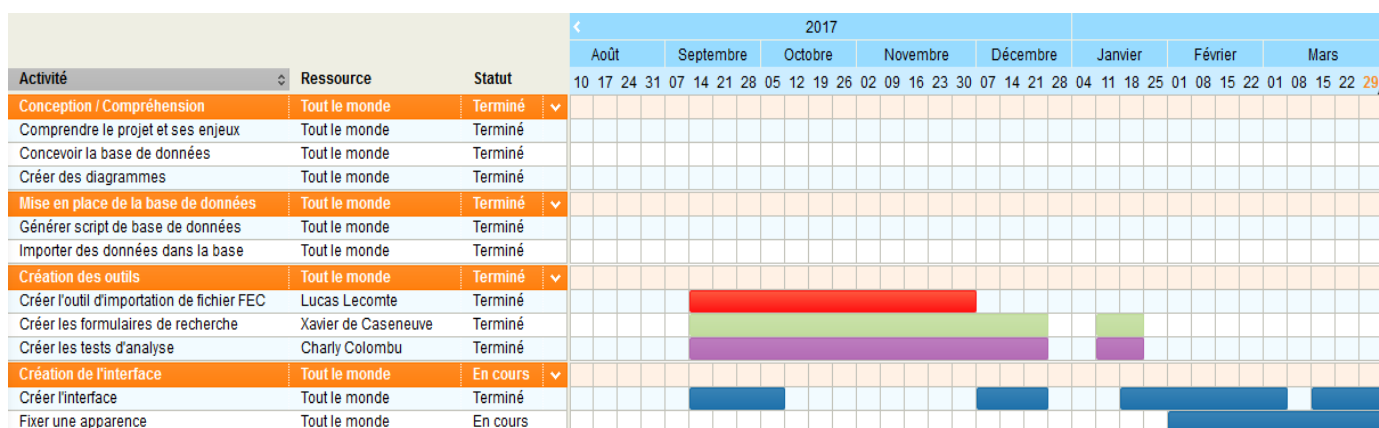


Diagramme de Gantt S4- Septembre 2017 à Mars 2018

1.2.2 Backlog

Numéro	Description	Priorité	Lot	Etat
1	Je veux pouvoir transférer mes fichiers d'écriture comptable XML dans l'outil.	1	Lot 2	terminé
2	Je veux pouvoir afficher ma base de données à partir de l'outil.	2	Lot 2	terminé
3	Je veux pouvoir analyser, calculer des ratios, effectuer des tests sur ma base de données.	3	Lot 3	terminé
4	Je veux une interface graphique intuitive.	3	Lot 4	terminé
5	Je veux pouvoir lire ma base de données dans un fichier CSV.	5	Lot 4	terminé
6	Je veux pouvoir rechercher des données particulières grâce à des formulaires	4	Lot 3	terminé

Backlog

Le backlog a été réalisé lors des cours de Gestion de projet informatique avec M.Lambert. Il a changé quelques fois depuis sa création mais nous n'avons pas pensé à sauvegarder les versions antérieures. Quelques descriptions ont été reformulées et quelques fonctionnalités ont été abandonnées par le commanditaire.

1.3 Choix des outils

1.3.1 Choix des langages

Au départ, l'application ne devait pas nécessairement être une application web, nous avons donc opté pour le langage JAVA. Nous avons fait ce choix car nous savions qu'en deuxième année nous allions apprendre comment gérer un fichier XML à partir d'un programme JAVA. Cependant après réflexion, nous avons opté pour les langages web. En effet, le PHP permettant de facilement gérer une base de données et le HTML possédant des balises d'input pour envoyer un fichier dans un formulaire, nous pouvions aisément transférer le fichier XML vers la base de données. De plus le but étant d'avoir une interface simple d'utilisation, le fait de pouvoir utiliser les feuilles de style CSS pour la mise en page et des fonctions JavaScript pour rendre les formulaires dynamiques était un avantage.

1.3.2 Choix des logiciels

Pour la création de la base de données, nous avons utilisé WinDesign car c'est le logiciel que nous avons appris à utiliser pendant nos cours. Pour la programmation des pages web nous avons utilisé Notepad++ lorsque nous étions sur les machines de l'IUT et quelques fois Atom ou SublimeText lorsque nous travaillions sur nos propres machines. Pour ce qui est de la mise en place de la base de données, nous avons utilisé PhpMyAdmin. Nous avons également utilisé FileZilla pour le transfert de fichier vers notre serveur de développement. Pour la rédaction du mémoire nous avons utilisé Google Docs de manière à pouvoir tous travailler sur le mémoire en même temps sans avoir à regrouper les différentes parties plus tard. Nous avons utilisé le gestionnaire de version mis en place par l'université principalement pour enregistrer notre travail dans un endroit sécurisé.

Nous avons reçu un FEC le 4 Mai 2017 de la part de notre commanditaire. C'est l'événement qui nous a permis de nous lancer dans la production de l'application.

2. Réalisation du projet

2.1 Le transfert du fichier FEC

Le transfert du fichier dans la base de donnée est une étape cruciale puisqu'elle permet de faire fonctionner les tests et les recherches sur le fichier. Un fichier est chargé en permanence dans la base de données et il est remplacé quand un nouveau fichier est importé. L'utilisateur charge son FEC depuis la page `disp_acceuil.php`, la page d'accueil de l'application. Cette page appelle ensuite le traitement PHP contenu dans le fichier `trmt_insertion.php`.

2.1.1 L'algorithme

Le transfert se fait en plusieurs étapes :

- L'utilisateur soumet son fichier, si l'extension n'est pas correcte l'opération est annulée
- Le fichier est nommé à la date du jour (exemple : 2018-03-29_16:18:20.mxl)
- La base de donnée est supprimée ainsi que toutes les données contenues dedans puis reconstruite
- Une fonction parcourt le fichier FEC et prépare des commandes d'insertion en SQL pour toutes les lignes d'écritures
- Les commandes d'insertion sont exécutées, toutes les données sont insérées
- Retour à la page d'accueil et affichage de quelques informations de prévisualisation

2.1.2 Les problèmes rencontrés

Le fichier que nous avons reçu de notre commanditaire a été anonymisé en utilisant le logiciel Excel. Nous avons donc travaillé avec un fichier XMS qui est l'équivalent de XML mais pour les documents Microsoft. Cependant les structures XML et XMS apportent quelques problèmes.

Le premier problème est que la structure XML du fichier contient un nombre de lignes beaucoup plus important que le nombre d'écritures.

En effet, il y a 22 informations sur chaque ligne et chacune de ces informations sont codées sur trois lignes de la manière suivante :

- Balise ouvrante
- Information
- Balise fermante

Le calcul est simple, pour notre fichier FEC de 757 lignes d'écritures nous nous retrouvons avec 49 962 lignes dans sa version XML.

Ensuite la structure XMS ajoute énormément de lignes qui ne contiennent pas d'informations relatives au fichier FEC. Le fichier que nous avons contient 60 425 lignes. Il y a donc 10463 lignes inutiles soit 17,3%.

756	VT	VENTES	336
757	VT	VENTES	336
758			

Nombre de lignes pour un FEC ouvert avec un tableur

```

60419  <Row ss:Index="1048576" ss:Height="14.9953">
60420    <Cell ss:Index="1024" ss:StyleID="ce1" />
60421  </Row>
60422  </Table>
60423  <x:WorksheetOptions />
60424  </Worksheet>
60425  </Workbook>

```

Nombre de lignes pour un FEC ouvert avec un éditeur de texte

De plus, les balises codant des cellules vides ne sont pas fermées correctement. Il y a une balise <Data/> au lieu de deux balises <Data></Data>.

```

<Cell>
  <Data ss:Type="String" />
</Cell>
<Cell>
  <Data ss:Type="String" />
</Cell>

```

Balises pour des cellules vides

Enfin la plus grosse difficulté était l'irrégularité de la manière de noter les cases vides. Nous allons prendre l'exemple des champs CompAuxNum et

CompAuxLib qui sont vides dans tout le fichier et des champs EcritureLet et DateLet qui sont parfois tous les deux vides parfois tous les deux remplis.

CompAuxNum et CompAuxLib sont codés par deux groupes de balises <Cell> </Cell> et <Data> </Data> et les balises <Data> sont vides. A l'inverse EcritureLet et DateLet sont codés par un seul groupe de balises <Cell> </Cell> et <Data> </Data> mais le champ suivant récupère dans sa balise <Cell> un argument "ss:index" qui indique dans quelle colonne se trouve le champ.

Ce problème a pour effet de décaler tous les champs, l'application essayait donc de les insérer dans la mauvaise table de la base de données.

```
<Cell ss:StyleID="ce1">
  <Data ss:Type="String">RESTAURANTS</Data>
</Cell>
<Cell>
  <Data ss:Type="String" />
</Cell>
<Cell>
  <Data ss:Type="String" />
</Cell>
<Cell ss:StyleID="ce1">
  <Data ss:Type="Number">1</Data>
</Cell>
<Cell ss:StyleID="ce1">
  <Data ss:Type="Number">20130902</Data>
</Cell>
```

1	CompteLib	CompAuxNu	CompAuxLib	PieceRef	PieceDate
2	RESTAURANTS			1	20130902
3	BANQUE 1			1	20130902
4	RESTAURANTS			2	20130903
5	BANQUE 1			2	20130903
6	CARBURANTS			3	20130903
7	BANQUE 1			3	20130903
8	FOURNISSEURS D			1518278	20130903
9	BANQUE 1			1518278	20130903
10	RESTAURANTS			5	20130904
11	BANQUE 1			5	20130904
12	RESTAURANTS			6	20130905
13	BANQUE 1			6	20130905

Les champs CompAuxNum et CompAuxLib

```
:Cell ss:StyleID="ce1">
  <Data ss:Type="String">RESTAURANTS</Data>
:/Cell>
:Cell ss:StyleID="ce1">
  <Data ss:Type="Number">39.1</Data>
:/Cell>
:Cell ss:StyleID="ce1">
  <Data ss:Type="Number">0</Data>
:/Cell>
:Cell ss:StyleID="ce1">
  <Data ss:Type="String" />
:/Cell>
:Cell ss:Index="16" ss:StyleID="ce1">
  <Data ss:Type="Number">20151116</Data>
:/Cell>
```

1	EcritureLib	Debit	Credit	EcritureLet	DateLet	ValidDate
2	RESTAURANTS	39,1	0			20151116
3	BANQUE 1	0	39,1			20151116
4	RESTAURANTS	55	0 AAB	20130903		20151116
5	BANQUE 1	0	55			20151116
6	CARBURANTS	70,18	0 AAA	20130903		20151116
7	BANQUE 1	0	70,18			20151116
8	ABC	11362	0 AAB	20130903		20151116
9	ABC	0	11362			20151116
10	RESTAURANTS	7,6	0 AAC	20130904		20151116
11	BANQUE 1	0	7,6			20151116
12	RESTAURANTS	25	0 AAD	20130905		20151116
13	BANQUE 1	0	25			20151116

Les champs EcritureLet et DateLet

Tous ces problèmes auraient rendu la fonction de récupération des données plus compliquée s'ils n'étaient pas traités.

2.1.3 Les solutions apportés

Le problème du nombre de lignes ne peut pas être résolu à moins de créer une structure nouvelle pour les fichiers d'écritures comptables cependant cela ne correspondait pas à ce projet.

Le problème des lignes inutiles dues à la structure XMS peut être réglé en créant une fonction qui parcourt le fichier et supprime les lignes inutiles cependant cela prendrait plus de temps à faire que de simplement les ignorer.

Le problème des balises de cellules vides fermés incorrectement est réglé avec une fonction qui parcourt le fichier et ferme correctement les balises concernées.

```
//remplace $ceci par $cela dans un $fichier grace a un fichier $temp
function remplacerPar($fichier, $temp, $ceci, $cela){

    $reading = fopen($fichier, 'r');
    $writing = fopen($temp, 'w');

    $replaced = false;

    while (!feof($reading)) {
        $line = fgets($reading);
        if (stristr($line,$ceci)) {
            $line = $cela;
            $replaced = true;
        }
        fputs($writing, $line);
    }
    fclose($reading); fclose($writing);
    // might as well not overwrite the file if we didn't replace anything
    if ($replaced) {
        rename($temp, $fichier);
    } else {
        unlink($temp);
    }
}
```

La fonction remplacerPar du traitement PHP

Le problème de l'irrégularité de notation des cellules vides est géré dans la fonction qui crée le tableau à partir du FEC. Cette fonction remet les champs à leur place dès qu'elle détecte la chaîne de caractère "ss:index".

2.1.4 Les parties importantes du code

```
function genererTab($fichier, $temp){

    $reading = fopen($fichier, 'r');
    $writing = fopen($temp, 'w');

    $tab = array();

    $nbLignes = 0;

    $replaced = false;//va degager

    while (!feof($reading)) {
        $line = fgets($reading);

        // https://stackoverflow.com/questions/4366730/how-do-i-check-if-a-string-
        // contains-a-specific-word
        if (strpos($line, '<Row ') !== false) {
            $numLigne = 0;
            $nbLignes++;
        }
        if (strpos($line, '<Data ') !== false) {
            $numLigne++;
            $tab[$nbLignes][$numLigne] = get_string_between($line, '>', '<');
        }
        if (strpos($line, '<Cell ss:Index') !== false) {
            $index = get_string_between($line, '"', '"');
            if($index <= 22){
                $nbAAjouter = $index - $numLigne - 1;
                $line = "<Cell><Data></Data></Cell>";
                $numLigne++;
                $tab[$nbLignes][$numLigne] = "";
                for($i = 0 ; $i < $nbAAjouter - 1 ; $i++){
                    $line = $line."<Cell><Data></Data></Cell>";
                    $numLigne++;
                    $tab[$nbLignes][$numLigne] = "";
                }
                $line = $line."<Cell>";
            }
        }

        fputs($writing, $line);
    }
    fclose($reading); fclose($writing);
    // might as well not overwrite the file if we didn't replace anything
    if ($replaced) {
        rename($temp, $fichier);
    } else {
        unlink($temp);
    }
    return $tab;
}
```

La fonction generTab du traitement PHP

La fonction generTab est une des fonctions les plus importante du traitement PHP. Elle permet de mettre toutes les données dans un tableau pour les traiter par la suite. Il s'agit d'une version modifiée d'une fonction proposée par un utilisateur sur le site www.stackoverflow.com .

```

$ligneDEcriture = "INSERT INTO LIGNE_DECRITURE(NUM_LIGNE, CODE_JOURNAL, NUM_EC,
DATE_LIGNE_EC, NUM_COMPTE, REFERENCE, LIBELLE_LIGNE_EC, MONTANT_DEBIT, MONTANT_CREDIT,
LETTRAGE, DATE_LETTRAGE, DATE_VALIDATION_LIGNE_EC, MONTANT_DEVISE, ID_DEVISE,
DATE_REGLEMENT, MODE_REGLEMENT, NATURE_OPERATION, ID_CLIENT, DEB_CRED) VALUES(";
$ecriturePiece = "INSERT INTO ECRITURE_PIECE(REFERENCE, DATE_PIECE) VALUES(";
$compte = "INSERT INTO COMPTE(NUM_COMPTE, LIBELLE_COMPTE, NUM_COMPTE_AUX,
LIBELLE_COMPTE_AUX) VALUES(";
$journal = "INSERT INTO JOURNAL(CODE_JOURNAL, LIBELLE_JOURNAL) VALUES(";

```

Les variables d'insertion.

Ces variables d'insertion sont de type chaîne de caractère. On ajoute les données du FEC à la fin avant de les exécuter sur le serveur pour les insérer dans la base de données.

```

foreach($ligne as $colonne => $valeur){
    switch($i){
        case 1:
            $journal = $journal.""".addslashes($valeur)."', ";
            $ligneDEcriture = $ligneDEcriture.""".addslashes($valeur)."', ";
            break;
        case 2:
            $journal = $journal.""".addslashes($valeur)."', ";
            break;
        case 5:
            $compte = $compte.""".addslashes($valeur)."', ";
            $ligneDEcriture = $ligneDEcriture.""".addslashes($valeur)."', ";
            break;
        case 6:
        case 7:
        case 8:
            $compte = $compte.""".addslashes($valeur)."', ";
            break;
        case 9:
            $ecriturePiece = $ecriturePiece.""".addslashes($valeur).".
".$tab_ref_pour_ecriture[$num_ec]."', ";
            $ligneDEcriture = $ligneDEcriture.""".addslashes($valeur).".
".$tab_ref_pour_ecriture[$num_ec]."', ";
            break;
    }
}

```

Une partie de la boucle principale du traitement

Cette boucle foreach parcourt le fichier lignes par lignes et ajoute les données dans les variables d'insertion. (cf. Annexe IV)

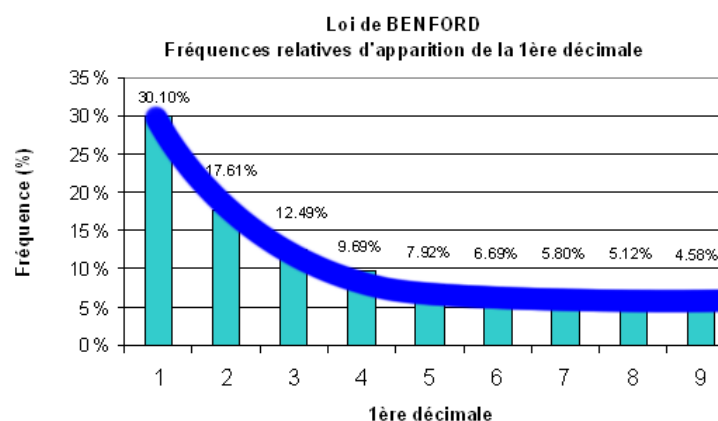
2.2 Les tests

Pour chaque fichier FEC envoyé, ce fichier est soumis à une analyse afin de contrôler la conformité du fichier. Cette analyse se fait par plusieurs tests : tel que le test de Benford par exemple. Ces tests peuvent prouver une erreur dans le fichier ou bien une fraude. Nous avons cinq tests : test de Benford, test des doublons, test de ruptures de séquences, calcul TVA et inversion hors-taxa et TVA

2.2.1 Présentation des tests

Ces tests permettent de vérifier l'intégrité du fichier FEC, il est donc important de pouvoir les intégrer.

La loi de Benford est une loi mathématique qui permet de détecter les fraudes financières ou comptables. Elle indique la fréquence d'apparition du premier chiffre significatif. Si les fréquences d'apparitions des chiffres significatifs, rangés de 1 à 9, dans le fichier FEC ressemble à celle proposée dans la loi de Benford, alors le fichier est conforme.



Fréquences de la Loi de Benford (source : etquefaire.fr)

Un fichier FEC ne peut avoir des doublons de transactions, il est donc essentiel de vérifier d'éventuels doublons dans le fichier pour prouver une conformité : c'est notre test des doublons. Ce fichier doit aussi avoir des écritures numérotées de manière croissante, avec un numéro changeant d'un en un. On ne peut avoir des numéros d'écritures 1 et 3 sans avoir une écriture de numéro 2, sinon c'est une rupture de séquence et donc un fichier non conforme.

EcritureNum	EcritureDate	CompteNum
3	20130903	401CARB
3	20130903	51210000
5	20130904	401RESTO
5	20130904	51210000

Exemple de rupture de séquence de numéro d'écriture

L'analyse des taxes dans un fichier FEC est aussi important pour vérifier la conformité du fichier. Le premier, l'inversion hors-taxe et TVA, vérifie que c'est bien les montants hors-taxes les plus élevés par rapport aux montants TVA. Le dernier test : le calcul de TVA, va permettre de vérifier que le taux de TVA appliqué est bien en accord avec le numéro de compte TVA (445).

2.2.2 Les algorithmes

Pour chaque test, il y a un bouton associé au test qui va permettre de lancer le traitement pour ensuite l'afficher. Ces boutons permettent de lancer une requête SQL pour récupérer toutes les données nécessaires afin d'y effectuer les manipulations nécessaires via des fonctions PHP ou bien JavaScript.

Pour le test de Benford, on récupère toutes les valeurs de crédit ou débit d'argent et on y effectue la manipulation suivante : récupérer seulement le premier chiffre de la valeur et compter le nombre de fois où l'on a un 1, un 2, etc. pour avoir un taux.

```
function premierChiffre(somme) {
  for(var i = 0; i < somme.length; i++) {
    while(somme[i] >= 10) {
      somme[i] = somme[i]/10;
    }
    somme[i] = parseInt(somme[i]);
  }
}
```

Algorithme récupérant les premiers chiffres significatifs

```

for(var i = 0; i < somme.length; i++) {
    frequence[somme[i]] = frequence[somme[i]] + 1;
    n++;
}
for(var i = 0; i < frequence.length; i++) {
    frequence[i] = 100 * frequence[i] / n;
    frequence[i] = frequence[i].toFixed(2);
}

```

Boucles qui transforment ces chiffres significations en fréquence

Pour le test des doublons, on a d'autres boutons permettant de choisir sur quel contenu spécifique du fichier on veut effectuer ce test : le libellé journal, le libellé compte ou la référence écriture. La fonction va ensuite vérifier toutes les données les unes par rapport aux autres pour vérifier des cas de doublons.

La rupture de séquence va récupérer tous les numéros d'écriture de la base de données et vérifier à l'aide d'une fonction JavaScript si on passe d'un numéro de ligne à une autre avec une différence supérieure ou égale à 2. Si oui, il affiche les valeurs où il y a une rupture de séquence.

Pour l'inversion HT/TVA, on récupère les débits et crédits grâce à une requête SQL en fonction des lignes d'écriture et vérifie si la valeur pour les comptes 6 ou 7 est supérieure à celle du compte 445.

```

foreach($num2 as $cle => $valeur) {
    if(strstr($valeur['num_compte'], '445')) {
        $tva = $valeur['montant_debit'];
        if($ht < $tva) {
            echo("Fraude inversion pour l'écriture numéro  

            ".$valeur['num_ec']);
        }
    }
    if ($valeur['num_compte'][0] == '6' || $valeur['num_compte'][0] == '7' ||  

    $valeur['num_compte'][0] == '2') {
        $ht = $valeur['montant_debit'];
    }
}

```

Boucle vérifiant une inversion hors-tax et TVA

Pour le calcul TVA, une requête SQL récupère tous les débits et crédits en fonction de la ligne d'écriture, et calcule la TVA en divisant les valeurs débit/crédit des lignes avec un numéro de compte 445 par rapport aux valeurs des numéros de compte 6 ou 7 (cf. Annexe V). On aura donc la TVA appliquée pour chaque ligne. On récupère toutes les valeurs de TVA qui ont été appliquées ainsi que les numéros de comptes hors-taxes, c'est à dire tous les numéros comptes autres que 445 qui ont une TVA qui a été appliquée, enfin d'y faire des recherches par la suite. Il y a deux méthodes de recherche : par taux de TVA ou par numéro de compte. Ce système de

recherche est géré par une liste déroulante, qui effectue par la suite une requête SQL et un traitement PHP pour avoir les résultats souhaités.

```
<form name="TVA_ContrôleTaux" action="<?php echo $_SERVER['PHP_SELF']; ?>"
      method="post" enctype="multipart/form-data">
  <?php
    //Crée une liste déroulante à partir des taux de tva qui ont été
appliqués
    echo('<select name="taux_tva" id="taux_tva">');
    echo '<option value="all">-----</option>';
    foreach($tvaTab as $cle => $valeur) {
      echo '<option value="' . $valeur . '>' . $valeur . '</option>';
    }
    echo('</select>');
  ?>
  <button type="submit" name="submit_tvaContrôleTaux">Contrôle sur taux de
TVA</button>
</form>
```

Exemple de formulaire avec une liste déroulante des taux TVA appliqués

2.3 Les formulaires

Pour les formulaires de recherche notre commanditaire nous a demandé de faire différentes familles. Il y a par exemple des formulaires concernant le montant de l'écriture, le numéro de compte où a lieu le transfert ou encore sur différentes dates. Le traitement des formulaires est exécuté avec la méthode post afin d'éviter l'affichage des données dans l'URL de la page web. La récupération des données entrées par l'utilisateur est faite en PHP. Pour avoir des formulaires un peu plus dynamique nous utilisons quelque fonction JavaScript.

CODE_JOURNAL	REFERENCE	NUM_COMPTES	NUM_LIGNE	NUM_EC	DEB	CRED	MONTANT_CREDIT	MONTANT_DEBIT	LETTRAGE	DATE_LETTRAGE	MONTANT_DEVISE	ID_DEVISE	DATE_REGLEMENT	MODE_REGLEMENT	NATURE_OPERATION	ID_CLIENT	DATE_LIG
B1	1	401RESTO	1	1	d	0.00	39.10				EUR	0000-00-00					2013-09-01
B1	1	51210000	2	1	c	39.10	0.00				EUR	0000-00-00					2013-09-01
B1	10	401RESTO	1	10	d	0.00	23.40				EUR	0000-00-00					2013-09-01
B1	10	51210000	2	10	c	23.40	0.00				EUR	0000-00-00					2013-09-01
B1	100	401CARB	1	100	d	0.00	46.58				EUR	0000-00-00					2013-11-27
B1	100	51210000	2	100	c	46.58	0.00				EUR	0000-00-00					2013-11-27
B1	101	401DIVERS	1	101	d	0.00	45.80				EUR	0000-00-00					2013-11-27
B1	101	51210000	2	101	c	45.80	0.00				EUR	0000-00-00					2013-11-27
B1	102	401PEAGES	1	102	d	0.00	24.50				EUR	0000-00-00					2013-11-27
B1	102	51210000	2	102	c	24.50	0.00				EUR	0000-00-00					2013-11-27
B1	103	401PARKING	1	103	d	0.00	1.40		AAD	2013-11-25	EUR	0000-00-00					2013-11-27
B1	103	51210000	2	103	c	1.40	0.00				EUR	0000-00-00					2013-11-27
B1	104	401RESTO	1	104	d	0.00	113.50				EUR	0000-00-00					2013-11-27
B1	104	51210000	2	104	c	113.50	0.00				EUR	0000-00-00					2013-11-27
B1	105	401HOTEL	1	105	d	0.00	48.00		AAA	2013-11-25	EUR	0000-00-00					2013-11-27
B1	105	51210000	2	105	c	48.00	0.00				EUR	0000-00-00					2013-11-27
B1	106	401PARKING	1	106	d	0.00	3.20				EUR	0000-00-00					2013-11-27
B1	106	51210000	2	106	c	3.20	0.00				EUR	0000-00-00					2013-11-27
B1	107	401RESTO	1	107	d	0.00	18.05				EUR	0000-00-00					2013-11-27
B1	107	51210000	2	107	c	18.05	0.00				EUR	0000-00-00					2013-11-27
B1	108	401RESTO	1	108	d	0.00	37.60				EUR	0000-00-00					2013-11-27
B1	108	51210000	2	108	c	37.60	0.00				EUR	0000-00-00					2013-11-27
B1	109	401RESTO	1	109	d	0.00	14.70		AAJ	2013-11-27	EUR	0000-00-00					2013-11-27

Exemple de formulaire de recherche

2.3.1 Fonctionnement

Chaque formulaire fonctionne sur le même principe l'utilisateur entre les données de son choix à l'aide de bouton radio, de checkbox ou encore de champ de texte puis envoie les données sélectionnées à l'aide du bouton d'envoi. Ensuite le traitement PHP se charge de créer la requête SQL et l'exécute sur la base de données. La partie JavaScript nous permet de proposer à l'utilisateur d'entrer plusieurs données, par exemple dans le formulaire de recherche à partir de date, l'utilisateur peut choisir de lancer une requête entre deux dates s'il choisit cette option avec le bouton radio correspondant. Un deuxième champ date va donc apparaître. L'apparition et la disparition du champ en fonction de sa nécessité nous a aussi faciliter la tâche pour sa prise en compte dans la création de la requête SQL. Le JavaScript a également permis la création de la fonction qui sélectionne la totalité du tableau renvoyé par la recherche permettant ainsi de le copier dans le presse-papier

2.3.2 Le traitement PHP

Le traitement PHP récupère tous les champs qui ont été remplis par l'utilisateur et les met dans différentes variables, pour ensuite les envoyer dans une fonction qui créera la requête SQL. Cette fonction concatène chaque variable dans l'ordre afin de donner une requête valide, elle vérifie si les champs d'option, par exemple le champ select présent dans le formulaire de recherche, à partir d'un montant et supérieur à 1, la fonction ajoute un AND dans la requête et prendra en compte le deuxième champs d'opération et de valeur à chercher, puis les concatènera à la suite du AND.

```
//construit la totalité de la requêtes a l'aide des champs fournie dans le formulaire correspondant
function construirerequete($debcred,$nb_operation,$operation,$champU,
                          $champD,$operationlog,$operationdeux,$champT,$champQ){

    $sql = 'select * from LIGNE_DECRITURE
            join COMPTE using (num_compte)
            join ECRITURE_PIECE using (reference)
            join JOURNAL using (code_journal) where ' ;

    if($nb_operation == 1){
        if ($operation != "between"){
            $sql .= $debcred." ".$operation." ".$champU;
        }
        else{
            $sql .= $debcred." ".$operation." ".$champU." AND ".$champD;
        }
    }
    else{
        if ($operation != "between" && $operationdeux != "between"){
            $sql .= $debcred." ".$operation." ".$champU." ".$operationlog." ".$debcred."
".$operationdeux." ".$champT;
        }
        else if ($operation == "between" && $operationdeux != "between"){
            $sql .= $debcred." ".$operation." ".$champU." AND ".$champD."
".$operationlog." ".$debcred." ".$operationdeux." ".$champT;
        }
        else if ($operation != "between" && $operationdeux == "between"){
            $sql .= $debcred." ".$operation." ".$champU." ".$operationlog." ".$debcred."
".$operationdeux." ".$champT." AND ".$champQ;
        }
        else{
            $sql .= $debcred." ".$operation." ".$champU." AND ".$champD."
".$operationlog." ".$debcred." ".$operationdeux." ".$champT." AND ".$champQ;
        }
    }
    return $sql;
}
```

Partie du code utiliser dans le traitement PHP (cf. Annexe VI)

Une fois tout cela fait, la requête est exécutée et le résultat est récupéré sous forme de tableau qui est par la suite affiché. Au-dessus du tableau la requête SQL est affichée. Au départ cette affichage a été implémentée pour vérifier les requêtes SQL renvoyer par la fonction et devait être retiré par la suite. Cependant à la demande du commanditaire nous l'avons laissé. Avant la fin du traitement PHP une dernière fonction est utilisée, celle-ci reprend le tableau résultat et l'enregistre dans un fichier xml nommé avec la date et l'heure à laquelle la recherche a été effectuée ainsi qu'avec

Copier

```
select * from LIGNE_DECRITURE
      join COMPTE using (num_compte)
      join ECRITURE_PIECE using (reference)
      join JOURNAL using (code_journal) where MONTANT_DEBIT = 39.10
```

CODE_JOURNAL	REFERENCE	NUM_COMPTES	NUM_LIGNE	NUM_EC	DEB_CRED	MONTANT_CREDIT	MONTANT_DEBIT	LETTAGE	DATE_LETTAGE	MONTANT_DEVISE	ID_DEVISE	DATE_REGLEMENT	MODE_REGLEMENT	NATURE_OPERATION	ID_CLIENT
B1	1	401RESTO	1	1	d	0.00	39.10				EUR	0000-00-00			

le nom du formulaire.

Exemple de résultat après envoie d'une requête

2.4 L'interface

Pour l'interface nous avons décidé que la page d'accueil ressemblerait à un tableau de bord qui permettrait de voir les éléments importants du fichier implémenté dans la base de données tel que le débit et le crédit total du fichier. Cela peut permettre à l'utilisateur de voir en un premier coup d'œil si le fichier possède des anomalies, en reprenant l'exemple du débit et du crédit si les deux valeurs diffèrent c'est qu'il manque probablement une ou plusieurs lignes d'écriture au fichier FEC.

Dogediac
Accueil Tests Formulaire

Actuellement chargé :

débit total : 78019.93

crédit total : 78019.93

3 journaux

32 comptes

336 écritures

756 lignes d'écriture

Votre fichier :

Aucun fichier sélectionné.

Page d'accueil

Pour les formulaires nous avons décidé de faire afficher la totalité de la base de données sous forme de tableau. Nous avons dans l'idée que cela pour aider l'utilisateur à choisir les données sur lesquelles il voulait faire la recherche. Seul le formulaire général ne possède pas ce tableau car celui-ci sert à afficher la totalité des données sur une ou plusieurs tables et un ou plusieurs champs de la base ou encore la base complète. Pour respecter la demande du commanditaire nous avons fait en sorte que les tableaux possèdent deux couleurs différentes le blanc et le gris qui s'intercalent à chaque fois, cela permet de faciliter la lecture du tableau. Toujours dans l'optique de simplifier la lecture du tableau nous avons décidé de changer la couleur de la cellule sur laquelle se trouve le curseur de la souris.

Conclusion

Pour conclure nous pouvons dire que ce projet a été une manière pour nous d'acquérir quelques connaissances en comptabilité mais aussi de compléter notre formation sur les langages de programmation web.

L'objectif de ce projet était de concevoir une application permettant de contenir, parcourir et analyser un FEC. L'application devait donc offrir une interface homme machine ainsi que des outils de gestion de FEC.

Après avoir considéré les avantages et les inconvénients d'une application web et d'une application JAVA nous avons choisi de développer une application web. Pour ce faire nous avons créé un algorithme, en PHP, d'insertion des données dans une base de données MySQL. Nous avons également créé des pages web, en HTML et JavaScript, pour parcourir, tester et analyser un FEC. Enfin nous avons créé une interface homme machine qui est simple d'utilisation en utilisant une feuille de style CSS et les éléments de structuration HTML.

Les résultats que nous escomptions au début du projet sont assez fidèles aux résultats effectifs. Les seuls points de divergence sont des fonctionnalités qui ont été abandonnées par le commanditaire ou bien des choix de langages.

Ce projet nous a permis de mettre en œuvre plusieurs des connaissances acquises au cours de ces deux années. Nous avons approfondi notre expérience dans les langages web, la gestion de base de données, l'algorithmique, la gestion de projet et la création d'interfaces homme-machine.

Ce projet a été enrichissant sur le plan personnel puisqu'il nous a permis d'approfondir notre formation dans plusieurs domaines mais aussi d'apprécier un peu plus le monde professionnel et sa liaison avec l'informatique. Nous avons également acquis de l'expérience dans la gestion de projet informatique et c'est très certainement une des choses les plus essentielles que ce projet nous a apporté.

La plus grosse difficulté à laquelle nous nous sommes heurtés a été de comprendre le fonctionnement et la structure des FEC. Beaucoup de ressources sont disponibles sur internet cependant elles apportent des informations très précises qui ne sont pas faites pour une introduction au sujet. Nous avons heureusement surmonté cette difficulté assez facilement après avoir reçu un exemple anonymisé d'un FEC de notre commanditaire. Il a également été difficile de manier la structure XMS puisque presque tous les problèmes de cette partie ont été engendrés par la structure XMS. De plus nous avons eu des difficultés avec la fonction qui permet de copier le tableau dans le presse papier. Enfin la mise en page a été le sujet de nombreuses réflexions puisque nous n'avions pas de directives précises.

L'application pourrait être sujette à certaines améliorations telles que l'ajout d'un système d'utilisateurs, l'ajout d'un système qui permettrait de gérer plusieurs fichiers à la fois ou encore l'approfondissement des fonctionnalités déjà existantes. Il faudrait également tester l'application avec une variété de FEC différents afin de s'assurer que l'algorithme fonctionne sur chacun d'entre eux.

Bibliographie

Programmation :

Forum phpfrance, "Importer des données XML dans une base de données MySQL", 7 octobre 2008. Consultable sur Internet :
<<https://forum.phpfrance.com/faq-tutoriels/importer-des-donnees-xml-dans-une-base-donnees-mysql-t243506.html>>

Site MySQL, "Load XML Syntax", *MySQL 5.5 Reference Manual*. Consultable sur Internet : < <https://dev.mysql.com/doc/refman/5.5/en/load-xml.html> >

MBIANDOU Douglas, "Accès base de données MySQL", *Tutoriel Java*, n°12. Consultable sur Internet :
< <http://objis.com/tutoriel-java-n12-acces-base-de-donnees-mysql/> >

Comptabilité :

CLERMONT Thibaut, "Le fichier des écritures comptables", *Les obligations comptables*. Consultable sur Internet : <<https://www.compta-facile.com/fichier-des-ecritures-comptables-fec-definition-contenu-utilite/>>

LAMORT DE GAIL Marc, "Le fichier des écritures comptables", *RFComptable*, n°416. Consultable sur Internet :
<<http://www.fidulorraine.fr/documents/publications/fec.pdf>>

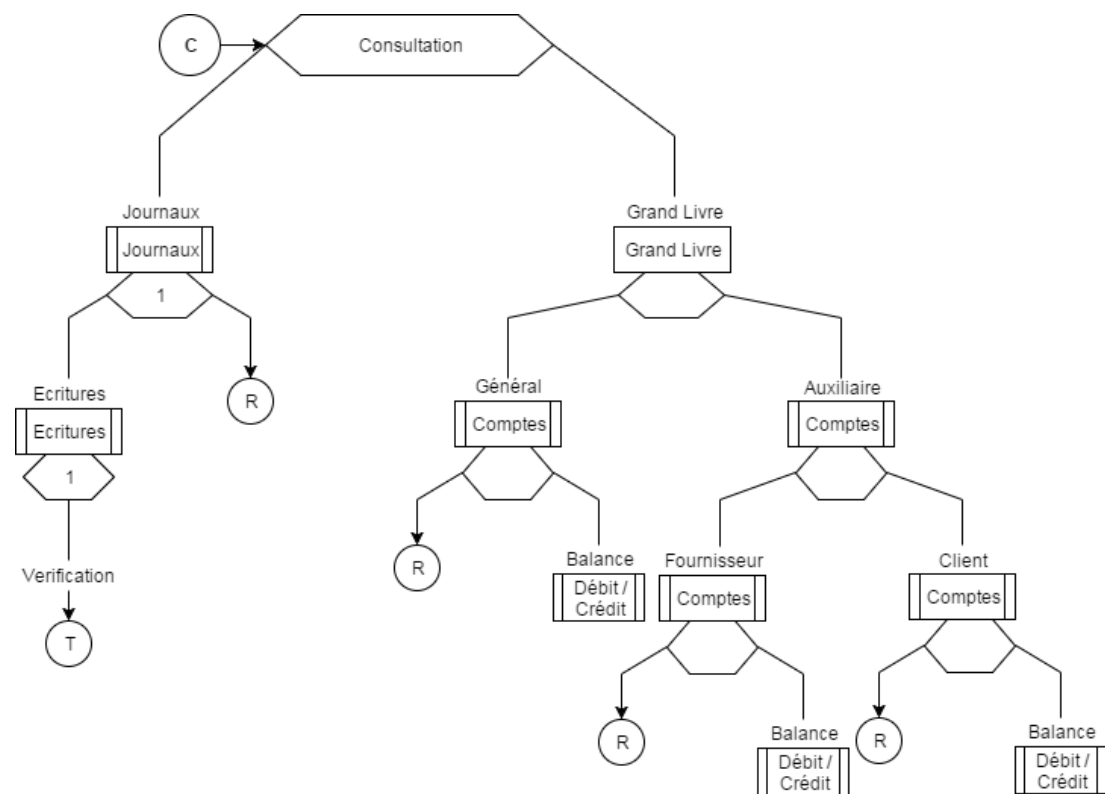
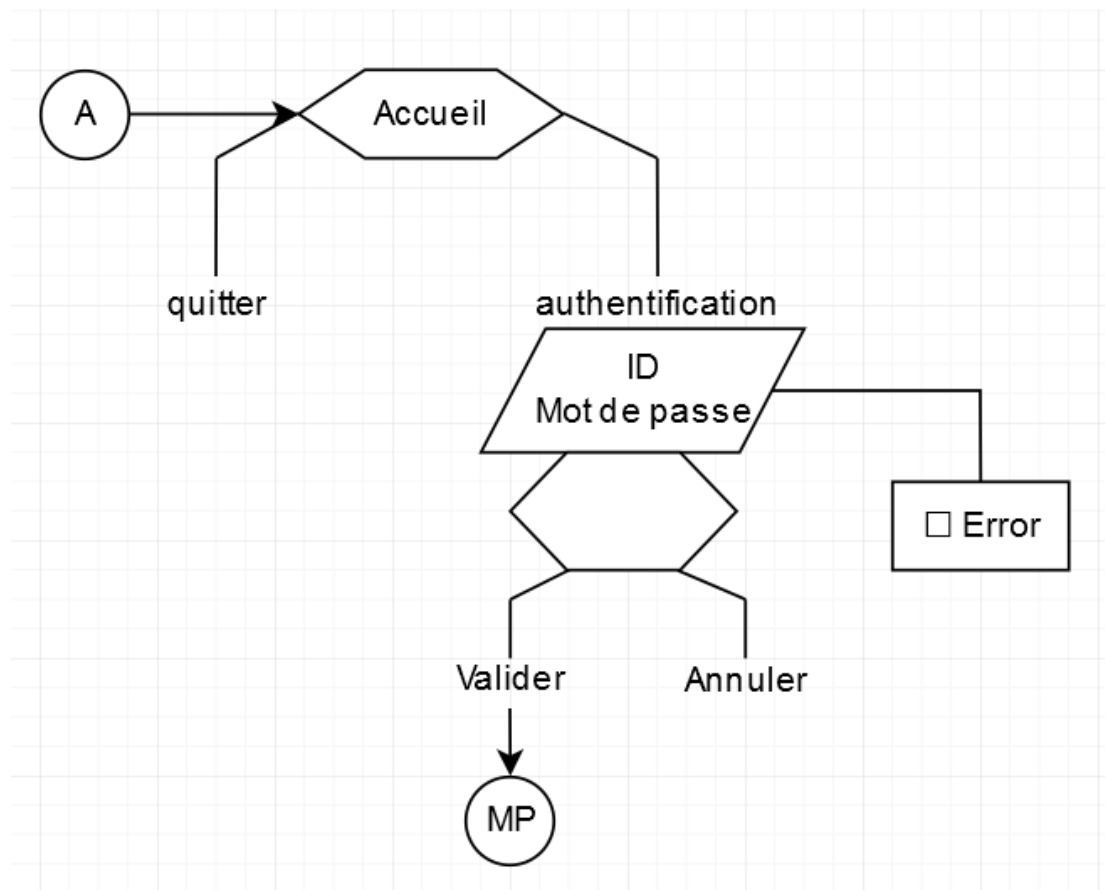
Cours :

M1104 Introduction au base de données
M2104 Modélisation par objets pour l'analyse et la conception
M2105 Introduction aux IHM
M2106 Programmation et administration des bases de données
M2204 Gestion de projet informatique
M3103 Algorithmique avancée
M3104 Programmation Web côté serveur
M3106C Bases de données avancées
M4203 Expression-communication - Communiquer dans les entreprises

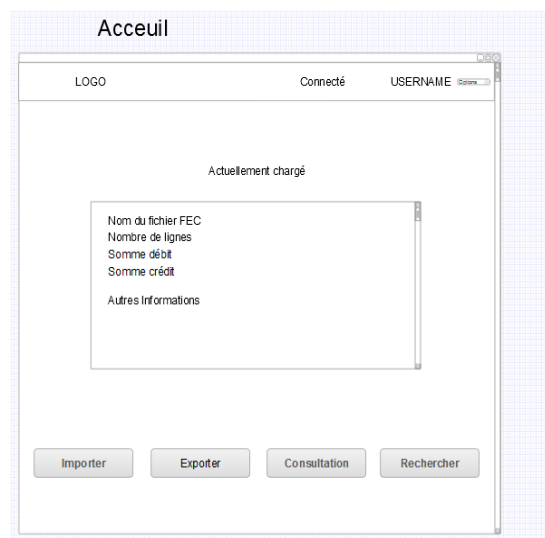
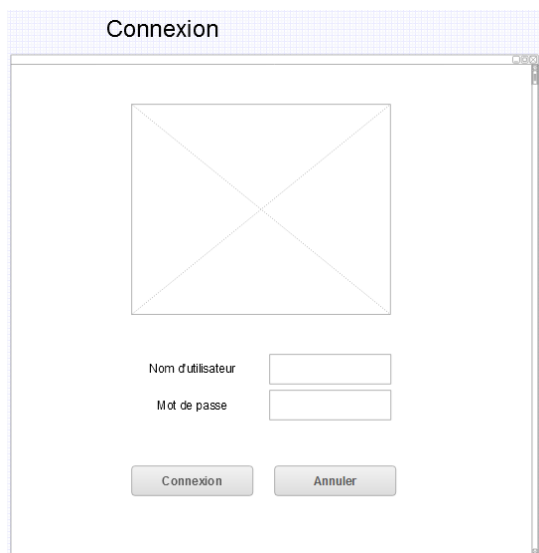
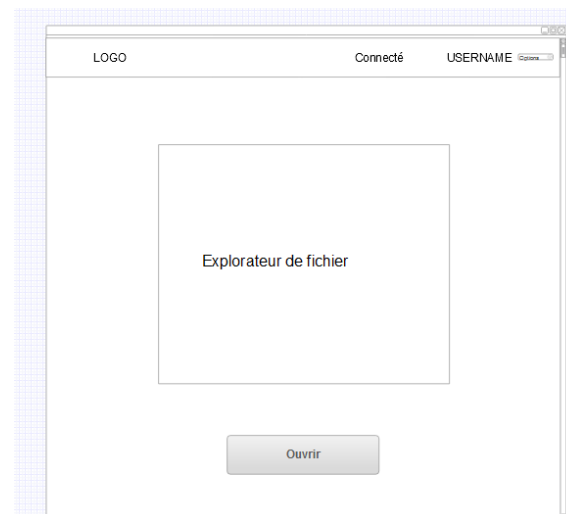
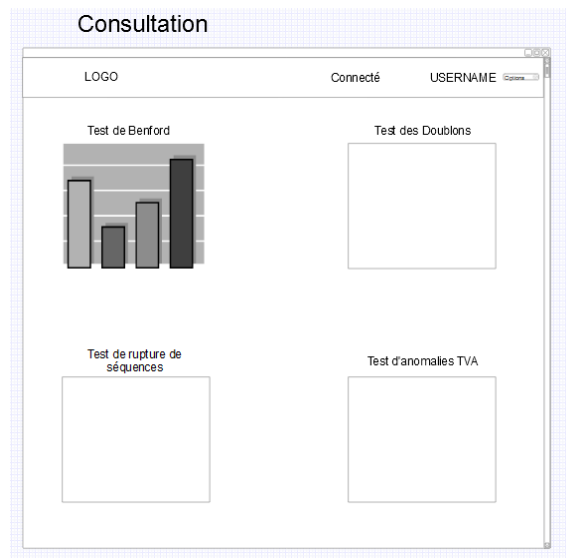
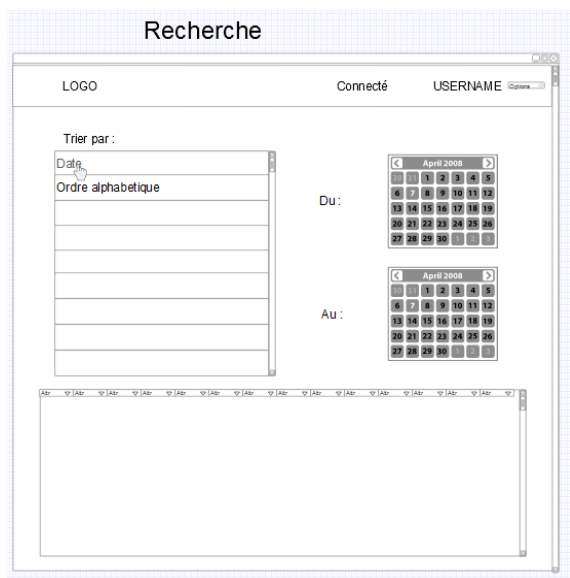
Annexe I - Diagramme de cas d'utilisation



Annexe II - SNI



Annexe III - Maquettes



Annexe IV - Le code permettant de mettre les données dans les variables d'insertion

```
if($num_ligne_actuel != 1){
    $i = 1;
    foreach($ligne as $colonne => $valeur){
        switch($i){
            case 1:
                $journal = $journal."".addslashes($valeur).", ";
                $lignedEcriture =
$lignedEcriture."".addslashes($valeur).", ";
                break;
            case 2:
                $journal = $journal."".addslashes($valeur).", ";
                break;
            case 5:
                $compte = $compte."".addslashes($valeur).", ";
                $lignedEcriture =
$lignedEcriture."".addslashes($valeur).", ";
                break;
            case 6:
            case 7:
            case 8:
                $compte = $compte."".addslashes($valeur).", ";
                break;
            case 9:
                $ecriturePiece = $ecriturePiece."".addslashes($valeur).
".$tab_ref_pour_ecriture[$num_ec].", ";
                $lignedEcriture = $lignedEcriture."".addslashes($valeur).
".$tab_ref_pour_ecriture[$num_ec].", ";
                break;
            case 10:
                if($valeur != ""){
                    $data = substr_replace($valeur, "-", 4, 0);
                    $ecriturePiece =
$ecriturePiece."".substr_replace($data, "-", 7, 0).", ";
                }
                else
                    $lignedEcriture = $lignedEcriture."null, ";
                break;
            case 12:
            case 13:
                $lignedEcriture = $lignedEcriture.addslashes($valeur).", ";
                break;
            case 3:
                $num_ec = $valeur;
                $lignedEcriture =
$lignedEcriture."".addslashes($valeur).", ";
                break;
            case 4:
            case 15:
            case 16:
                if($valeur != ""){
                    $data = substr_replace($valeur, "-", 4, 0);
```

```

                                $lignedEcriture =

$lignedEcriture.""substr_replace($data, "-", 7, 0)."', ";
                                }
                                else
                                $lignedEcriture = $lignedEcriture."NULL, ";
                                break;
                                case 11:
                                case 14:
                                case 17:
                                case 18:
                                case 19:
                                case 20:
                                case 21:
                                case 22:
                                $lignedEcriture =
$lignedEcriture.""addslashes($valeur)."', ";
                                break;
                                default :
                                echo "bug($i)";
                                }
                                $i++;
                                }
}

```

Annexe V - Calcul de taux de TVA

```
foreach($num2 as $cle => $valeur) {
    //Si le numéro de compte commence par 445
    if(strpos($valeur['num_compte'], '445')) {
        //Si le montant débit n'est pas égal à 0
        if(floatval($valeur['montant_debit'] != 0)) {
            //On récupère la tva
            $tva = floatval($valeur['montant_debit']);
            //Et on calcule le taux
            $tauxTva = round($tva / $ht * 100, 1);
            //On récupère le num compte pour la tva
            $numcompteTVA = $valeur['num_compte'];
            //On ajoute numéros de compte, numéro d'écriture et le taux tva
            appliqué dans un tableau
            $écritureHT = array(1 => $numec, $numcompteHT, $tauxTva);
            array_push($écritureTva2, $écritureHT);
            $écritureTVA = array(1 => $numec, $numcompteTVA, $tauxTva);
            array_push($écritureTva2, $écritureTVA);
            //On ajoute le taux de tva dans un tableau, s'il n'a pas déjà été
            ajouté

            if(in_array($tauxTva, $tvaTab) == false)
                array_push($tvaTab, $tauxTva);
            //On ajoute les numéros de compte, s'il n'ont pas déjà été ajoutés
            if(in_array($numcompteHT, $compteTab) == false)
                array_push($compteTab, $numcompteHT);
            if(in_array($numcompteTVA, $compteTab) == false)
                array_push($compteTab, $numcompteTVA);
        }
    }
    if ($valeur['num_compte'][0] == '6' || $valeur['num_compte'][0] == '7' ||
    $valeur['num_compte'][0] == '2') {
        //On récupère la valeur hors taxe
        $ht = floatval($valeur['montant_debit']);
        //On récupère le numéro d'écriture et le numéro de compte pour le compte
        produit / charge
        $numec = $valeur['num_ec'];
        $numcompteHT = $valeur['num_compte'];
    }
}
```

Annexe VI - Code du traitement PHP d'un formulaire

```
include('../insert/pdo.php');
include("../insert/insert_info_connexion.php");

//construit la totalité de la requêtes a l'aide des champs fournie dans le formulaire
correspondant
function construirerequete($debcred,$nb_operation,$operation,$champU,
                          $champD,$operationlog,$operationdeux,$champT,$champQ){

    $sql = 'select * from LIGNE_DECRITURE
            join COMPTE using (num_compte)
            join ECRITURE_PIECE using (reference)
            join JOURNAL using (code_journal) where ' ;

    if($nb_operation == 1){
        if ($operation != "between"){
            $sql .= $debcred." ".$operation." ".$champU;
        }
        else{
            $sql .= $debcred." ".$operation." ".$champU." AND ".$champD;
        }
    }
    else{
        if ($operation != "between" && $operationdeux != "between"){
            $sql .= $debcred." ".$operation." ".$champU." ".$operationlog."
".$debcred." ".$operationdeux." ".$champT;
        }
        else if ($operation == "between" && $operationdeux != "between"){
            $sql .= $debcred." ".$operation." ".$champU." AND ".$champD."
".$operationlog." ".$debcred." ".$operationdeux." ".$champT;
        }
        else if ($operation != "between" && $operationdeux == "between"){
            $sql .= $debcred." ".$operation." ".$champU." ".$operationlog."
".$debcred." ".$operationdeux." ".$champT." AND ".$champQ;
        }
        else{
            $sql .= $debcred." ".$operation." ".$champU." AND ".$champD."
".$operationlog." ".$debcred." ".$operationdeux." ".$champT." AND ".$champQ;
        }
    }
    return $sql;
}

if (!empty($_POST ))
{
    //connection à la base de données
    $conn = ConnectorPDO($db_address,$db_username, $db_password);

    //récupération des données rentré dans le formulaire
    echo "<PRE>";
    echo "<link rel='stylesheet' href='../insert/style2.css'> ";
    if(!empty ($_POST['debcred'])){
        $debcred = $_POST['debcred'];
    }
    if(!empty ($_POST['nb_operation'])){
        $nb_operation = $_POST['nb_operation'];
    }
}
```

```

}
if(!empty ($_POST['operation'])){
    $operation = $_POST['operation'];
}
if(!empty ($_POST['champ1'])){
    $champU = $_POST['champ1'];
}
if(!empty ($_POST['champ2'])){
    $champD = $_POST['champ2'];
}
else{
    $champD = "none";
}
if(!empty ($_POST['operationlog']) && $nb_operation > 1){
    $operationlog = $_POST['operationlog'];
}
else{
    $operationlog = "none";
}
if(!empty ($_POST['operationdeux']) && $nb_operation > 1){
    $operationdeux = $_POST['operationdeux'];
}
else{
    $operationdeux = "none";
}
if(!empty ($_POST['champ3']) && $nb_operation > 1){
    $champT = $_POST['champ3'];
}
else{
    $champT = "none";
}
if(!empty ($_POST['champ4']) && $nb_operation > 1){
    $champQ = $_POST['champ4'];
}
else{
    $champQ = "none";
}

$sql = construirerequete($debcrid,$nb_operation,$operation,$champU,
                        $champD,$operationlog,$operationdeux,$champT,$champQ);

$rep;
$cur = preparerRequete($conn,$sql);
$nbligne = LireDonneesPDO1($conn,$sql,$rep);
echo "<fieldset>".$sql."</fieldset>";
if($nbligne == 0){
    echo ("la requête ne renvoie rien !");
    afftout();
}
else{
    echo "<button id='copier' class='bouton_submit'>Copier</button>";
    AfficherDonneeTAB($rep);
    echo "<script src='copier.js'></script>";
    CreerFichierXML($rep,"montant");
}
unset($_POST);
echo "</PRE>";
}
else{
    include('requetes_montant.html');
}
}

```

Annexe VII - Fiche de lecture

FICHE DE LECTURE
de mémoire de conduite de projet 2A
établie par Paula GRENTE, Sophia HERVIEU, Salomé MENARD
et transmise le 23/03/2018

Intitulé du projet : Gestion de fichier d'écriture comptable

NOMS et Prénoms des membres du groupe de projet :

Lucas LECOMTE ; Xavier DE CASENEUVE ; Charly COLOMBU

1. La forme du mémoire

→ Le respect des normes de présentation

- **Couverture :** La première de couverture respecte les normes. La quatrième de couverture comporte le bon nombre de lignes, est en Français et en Anglais mais les mots-clés ne sont pas inscrits en Anglais. La quatrième de couverture est au mauvais endroit, elle doit être entre la page de titre et la page de remerciements.
- **Liminaires :** Page de garde présente, page de titre aussi, remerciements présents mais un peu succincts, sommaire présent et cliquable, pas de tables présentes, pas de glossaire / lexique.
- **Corps du mémoire :** Introduction présente et pertinente, quelques « trous » dans le développement : des parties sont incomplètes et parfois, le style appliqué est incohérent, conclusion pertinente, pas de renvois ni de notes de bas de page.
- **Visuels :** Les illustrations ne sont pas toutes légendées (problème d'adaptation au texte?) mais sont plutôt bien introduites dans le contenu. Les visuels sont pertinents.
- **Appareil de référence :** Pas d'appareil de référence, les sources des images sont indiquées dans la légende.
- **Respect du volume de pages indiqué :** Le mémoire compte vingt pages contre vingt à trente pages pour le corps du mémoire *uniquement*.

→ La lisibilité

- **Langue :** Il y a des fautes d'accord un peu partout dans le mémoire, cependant le vocabulaire est accessible, la syntaxe est bonne et la ponctuation régulière ce qui rend le tout, malgré les fautes récurrentes, tout de même compréhensible.
- **Adaptation du discours au destinataire :** Le tout est compréhensible et assez bien expliqué, les explications sont assez vulgarisées pour être accessibles à tous.
- **Mise en page et typographie :** Pas de pagination, certains titres sont mal formatés, mais bien hiérarchisés. Les marges sont quelquefois irrégulières. Les alinéas, paragraphes et interlignes sont bien harmonisés.

- **Qualité des visuels** : Le mémoire est illustré de manière pertinente, les visuels sont de qualité, peut-être penser à une illustration sur la couverture ? (facultatif)

2. Le contenu du mémoire

→ La cohérence

- **Plan** : Le plan est cohérent, suit le développement de l'application dans l'ordre chronologique. Bonne distinction entre le début du projet et la partie développement. Les titres sont explicites sauf « Des bouts de code ?? » (nous pensons que cette partie est encore en développement et que ce titre sera provisoire). Encore une fois, attention aux accords !
- **Logique du discours** : Le discours est bien articulé et logique, on ne perd pas le fil du discours.
- **Introduction** : Il manque la problématique, nous avons vu que vous avez laissé un commentaire à ce propos.
- **Conclusion** : La conclusion résume bien le projet et dresse un bilan de l'expérience vécue. Le seul bémol que nous voyons est l'orthographe en particulier les accords (« un » à la place de « une » par exemple).

→ La pertinence

- **Caractère globalement analytique du propos** : Les rédacteurs ont mis assez de distance dans l'explication, bonne utilisation de la troisième personne.
- **Problématisation autour de la mission à effectuer** : La problématique n'est pas encore énoncée.
- **Capacité à présenter l'existant puis le projet ou la mission** : Aucun problème à pointer.
- **Qualité de la réflexion** : La réflexion est logique et aboutie.
- **Recours pertinent à des connaissances, des compétences et des outils universitaires** : Vous montrez que vous avez eu recours aux connaissances issues cours dispensés par l'IUT, mais aussi acquises lors du projet.

Points ajoutés à notre mémoire:

La bibliographie a été ajoutée suite à la fiche de lecture. Toutes nos illustrations ont été légendées, ce point a donc été pris en compte. Les appareils de référence (annexes et bibliographie) ont été pris en compte ainsi que la source des images quand elles étaient nécessaires. La page de remerciements a été modifiée car elle était bien trop courte. La problématique et la pagination ont elles aussi été ajoutées.

Points non ajoutés à notre mémoire:

Les mots clés en Anglais, car ils ne sont pas nécessaires. La quatrième de couverture doit être après la page de garde, d'après la méthodologie des mémoires et soutenances, nous n'avons donc pas modifié sa place malgré la fiche de relecture. Nous n'avons pas mis d'illustration sur la couverture, car nous n'avions pas de logo spécifique à mettre.