Etude, conception et implémentation d'une application Web sécurisée

Ing. V. DIRKEN
Ir L. BACLIN
ISICHt – Mons

Le travail de fin d'études s'est scindé en deux parties. La première partie est le développement d'une solution afin que les employés puissent indiquer leurs heures passées sur chaque projet. Auparavant, les informations étaient extraites manuellement à partir de fichiers *Excel*. Le projet consiste à analyser le système existant, générant des erreurs, et de proposer une solution plus performante et évolutive. Le choix s'est porté sur le développement d'une application Web. La seconde partie a porté sur un accès sécurisé à cette plateforme grâce à un « Réseau Privé Virtuel » (VPN).

Mots clés : timesheets, théorie des ensembles, application Web, HTML, CSS, PHP, MySQL, requêtes asynchrones, accès sécurisé, VPN, L2TP, IPsec, certificats X.509, réseaux chiffrés.

Final thesis has been split in two parts. In the first part, timesheets system had to be reconsidered allowing employees to communicate their working time. Information was manually extracted from *Excel* sheets. The project involves analysing the existing system, producing errors, for the purpose of to provide a more effective solution. The second part was focused on a secure access to this system using a virtual private.

Keywords : timesheets, set theory, web application, HTML, CSS, PHP, MySQL, asynchronous requests, secure access, virtual private network, L2TP, IPsec, X.509 certificates, encrypted networks.

1. Introduction

1.1. Contexte

Le sujet de ce travail porte d'une part sur la conception et l'implémentation d'une plateforme de gestion des temps de travail (*timesheets*) et, d'autre part, sur l'étude et l'implémentation d'un accès sécurisé à cette plateforme.
Les principales fonctions de l'application sont de suivre les temps de travail de chaque employé, d'analyser et d'extraire des résultats qui sont utilisés en interne et transmis à la sécurité sociale tout comme à des entreprises extérieures.

Cette publication est organisée en 3 parties. La première partie présente l'analyse du besoin, l'étude d'une solution et le choix des langages. La seconde partie est consacrée au développement d'une des fonctions de l'application Web. La troisième partie présente l'étude et l'implémentation d'une solution de sécurité afin d'accéder à la plateforme. Ce travail se termine sur l'analyse des résultats ainsi que les perspectives de ce projet.

1.2. Analyse du besoin

Initialement, le fichier *Excel* contenant la *timesheet* devait être présent sur la machine de l'employé. Si l'employé ne la remplissait pas correctement, le fichier ne fonctionnait pas lorsqu'il était envoyé sur le serveur central, ce qui empêchait la génération des résultats. En effet, à titre d'exemple, si une date était mise dans la colonne des heures prestées, le fichier central ne pouvait pas calculer, car les données étaient incohérentes. De plus, si le nom d'un projet devait changer, la *manager* devait ouvrir tous les fichiers *Excel* et renommer ce projet. Cette analyse prouve à quel point il est difficile de tenir ce système évolutif et efficace.
Plus précisément, ce système ne respecte pas l'intégrité référentielle.

Le cahier des charges n'impose pas d'architecture spécifique concernant la modélisation des données, mais celle-ci doit être justifiée et vérifiée.
Une analyse des systèmes disponibles sur le marché a été entreprise. Cependant, reprendre un produit existant, performant, gratuit et répondant strictement aux besoins de l'entreprise n'est pas possible et justifie l'élaboration et la mise en place de cette application.

Par ailleurs, les employés devaient modifier leur fichier sur un ordinateur de l'entreprise, ce qui était compliqué pour les personnes qui travaillent en consultance chez un client.

2. Application Web *timesheets*

2.1. Méthodologie de développement

Afin de développer au mieux l'application, il est nécessaire de convertir les besoins en une architecture informatique. Premièrement, une analyse fonctionnelle permet d'étudier en profondeur les besoins du projet et de décrire les fonctions qui devront être développées. Cependant, cette méthode ne donne aucune information sur les technologies à utiliser. C'est pourquoi une architecture informatique est également étudiée.

Cette étude fait appel aux différents langages et technologies disponibles selon plusieurs critères émis dans le cahier des charges : la gratuité et l'existence d'une application web tournant sur un serveur en interne.

2.2. Étude d'une solution et choix des langages

Le cahier des charges exprime que l'application Web est destinée à interagir avec des informations stockées dans plusieurs bases de données.

Une des bases de données contient toutes les informations relatives aux projets, aux financements, aux employés tandis que d'autres doivent être générées à la volée. En effet, une base de données est créée pour chaque année civile. Ce choix, décidé en interne, permet de verrouiller une base de données lorsque le bilan comptable est déposé.

L'employé doit pouvoir interagir avec ces données selon son niveau d'accès à travers des pages Web. En effet, chaque employé doit reporter ses heures prestées dans la sous-catégorie d'un projet. Il peut ajouter, éditer, lire et supprimer une tâche non verrouillée. Par ailleurs, chaque projet comporte un ensemble de paramètres comme le statut du projet, le financement alloué à ce projet, les sous-catégories ouvertes ainsi que les droits d'accès pour chaque sous-catégorie en fonction de l'employé.

À chaque fin de mois, la *manager* ainsi que le directeur veulent accéder à une partie restreinte de la plateforme. Cette partie a pour but d'analyser, de générer automatiquement des rapports financiers et des factures destinés aux tiers. Cette plateforme doit également proposer des indicateurs de performance qui sont utilisés en interne afin d'améliorer les composantes temps, qualité et coût d'un projet.

De plus, l'entreprise exige que le système soit ergonomique pour les utilisateurs et rapide lors de la génération des résultats.
Par ailleurs, l'entreprise souhaite que l'employé s'authentifie et qu'il puisse se connecter à travers un *smartphone* depuis l'extérieur de l'entreprise.

Afin de répondre au mieux au cahier des charges, le projet est scindé en deux parties. L'application Web d'une part et d'autre part l'accès sécurisé depuis l'extérieur.

Le rendu d'une page Web utilise le langage HTML (Hypertext Markup Language). Ce langage est sémantique et repose sur des balises afin de mettre en forme le contenu des éléments. Les pages HTML peuvent intégrer des éléments multimédias comme les images. Néanmoins, dans ce cas, l'affichage doit être dynamique. En effet, il est impensable d'écrire chaque page HTML pour chaque employé. De plus, il est impossible au premier abord de calculer les résultats et de générer des rapports. Ce constat montre qu'il est nécessaire de réfléchir à une architecture répondant mieux à la demande.

L'étude montre qu'il est possible de générer des pages HTML à la volée en utilisant par exemple le langage Java, ASP ou PHP. Le choix s'est porté sur le langage PHP (Hypertext Preprocessor) qui est gratuit et répond, sans complexité à la demande, contrairement au Java ou l'ASP qui est payant.
En effet, ce langage permet de générer des pages HTML ainsi que du CSS et du JavaScript.
Le langage PHP permet notamment de communiquer avec un serveur de bases de données, il offre la possibilité de gérer le résultat d'une requête SQL, de gérer des sessions d'accès, des cookies, chiffrer des données ou de générer des condensats de messages comme le SHA-1 qui permet de rendre illisible et non réversible un mot de passe dans une base de données.

Ce langage a fait ses preuves dans de nombreux domaines et est implémenté sur de nombreux sites Web grand public comme Google, Facebook ou Wikipédia.

Le langage CSS (Cascading Style Sheets) soutient intimement le HTML. Le langage CSS manie la mise en forme du HTML dans le navigateur. En effet, afin d'éviter l'écriture redondante, il est possible de découpler la mise en forme dans un fichier qui est importé par le fichier HTML.

La dernière étape est de manipuler les informations. L'enjeu fondamental est d'éviter à l'avenir les problèmes d'intégrité référentielle, la redondance et la mise à jour des données. Ces données sont agencées grâce à la modélisation des bases de données faisant appel à la théorie des ensembles.

Les informations contenues dans ces bases de données sont organisées et extraites grâce au langage SQL (Structured Query Language). SQL permet de récupérer, d'insérer, de mettre à jour et de supprimer des données. Ce langage a la capacité de créer des bases de données à la volée et de gérer les jointures. Une jointure est l'association de minimum deux tables contenant des informations reliées entre elles par un paramètre appelé clé.

La base de données est en MySql dont le moteur de stockage est innoDB. Ce moteur de stockage propose d'ajouter des contraintes relationnelles entre plusieurs tables afin de garantir l'intégrité des données. La modélisation et ces contraintes suppriment définitivement les problèmes inhérents de l'ancien système. À titre d'exemple, le nom du projet se trouve à un seul endroit dans la base de données. De plus, il est dorénavant impossible de supprimer ce projet si des sous-catégories relatives à celui-ci existent. En effet, si un projet n'existe plus, il est verrouillé et n'est plus visible auprès des utilisateurs et ce grâce à une gestion de droits d'accès, mais il ne sera jamais supprimé s'il existe un élément sous-jacent relié au projet (heures prestées, sous-catégories...)
Ce choix s'est porté sur des bases de données MySql, car le langage PHP peut manipuler les données contenues dans ces bases de données.

L'entreprise demande également que l'application Web soit ergonomique et rapide. Les recherches ont conclu que le JavaScript est un candidat très intéressant parmi ses concurrents dans le domaine des applications Web. De plus, le moteur JavaScript est implémenté nativement dans tous les navigateurs récents. L'analyse montre que de plus il est possible d'augmenter considérablement l'ergonomie en couplant JavaScript à du PHP en utilisant les requêtes asynchrones AJAX (Asynchronous JavaScript and XML).
Le principal avantage est que le code JavaScript peut rendre la page sensible à des évènements comme le clic de souris. Il peut également interagir avec le rendu de la page en modifiant des parties de celles-ci ou en modifiant des propriétés CSS. Pour ce faire, JavaScript utilise l'objet *Document Object Model* pour atteindre les éléments de la page HTML hiérarchisés.
Néanmoins, l'implémentation de cette ergonomie suggère une très bonne maîtrise du langage JavaScript, HTML, PHP et SQL.

Dans une seconde partie, l'entreprise spécifie, dans le cahier des charges, un accès sécurisé depuis l'extérieur grâce à un *smartphone*. Une des solutions est de développer une application pour Android en Java et en Swift/Objective C pour iOS. Le temps de développement ainsi que le coût d'exploitation font qu'une alternative doit être trouvée. Le choix s'est porté sur le développement d'un fichier CSS capable d'afficher correctement l'application Web sur un *smartphone*. Le navigateur Web détecte la taille de l'écran et bascule sur le CSS correspondant.

L'application Web prend en charge les *smartphones*, mais ne permet pas d'accéder depuis l'extérieur au réseau interne de l'entreprise. Afin de régler ce problème, une solution VPN est implémentée afin de sécuriser l'accès aux *timesheets* en passant par un serveur CentOS (distribution Unix) qui est imposé par le cahier des charges.

2.3. Les bases de données

Modélisation de la base de données

Les informations contenues dans une base de données peuvent évoluer dans le temps. En effet, les informations sont ajoutées, supprimées, modifiées. Afin de garder une certaine cohérence dans les informations, il est nécessaire de respecter l'intégrité référentielle.

L'intégrité référentielle est un des principes fondamental des bases de données relationnelles qui fait appel à une clé étrangère liée à une autre clé indexée sous forme d'une contrainte. Ce lien permet de garantir la cohérence des informations. La clé étrangère garantit que les valeurs contenues dans une colonne ou un ensemble de colonnes provenant d'une table, existent dans une ou plusieurs colonnes provenant d'une autre table.

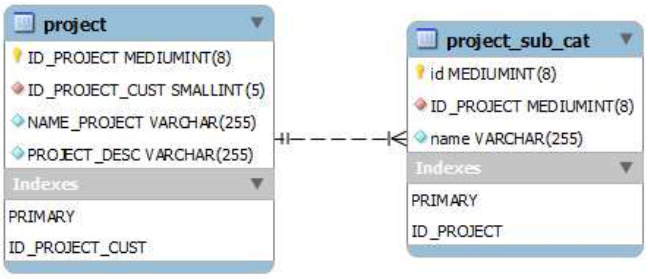

Figure 1 - Modélisation de deux tables

Une contrainte référentielle est ajoutée entre ces deux tables [9] :

```
ALTER TABLE `project_sub_cat`
ADD CONSTRAINT `acti_pr` FOREIGN KEY (`ID_PROJECT`)
REFERENCES `project` (`ID_PROJECT`) ON UPDATE CASCADE;
```

Ainsi, lorsque la clé primaire d'une ligne est modifiée (*UPDATE*), sa valeur est propagée à toutes les clés étrangères liées par la contrainte. C'est l'action référentielle en cas d'*UPDATE*.
ID_PROJECT de la table *project_sub_cat* est la clé étrangère de *ID_PROJECT* provenant de la table *project*.

Développement d'une fonction

En raison de la quantité de fonctions à développer dans cette application, une seule fonction est décrite et développée dans cette partie. Cette section propose la modélisation de la base de données [4] ainsi que la création dynamique des pages dans l'objectif de les afficher en fonction de plusieurs paramètres.

À titre d'exemple, un projet contenant des sous-catégories se présente de la manière suivante :

- Projet ADC14b1MHZ :
  - Concept Evaluation
  - Feasability
  - Development – Design
  - Development – Layout
  - Prototyping
  - Qualification
  - Production
  - End Of Life

Un second projet :

- Projet LVDS
  - Concept Evaluation
  - Feasability
  - Development – Design
  - Development – Layout
  - Prototyping
  - Qualification
  - Production
  - End Of Life

Par ailleurs, le projet Business a les sous-catégories suivantes :

- Prospect
- Finance
- Marketing

Un projet suit ce gabarit, mais il est possible de supprimer ou d'ajouter des sous-catégories selon les besoins comme pour le projet Business.

Une ou plusieurs sous-catégories provenant d'un projet peuvent être financées par une structure extérieure (banque, subsides) ou par des fonds propres.
De plus, une sous-catégorie peut être soumise à un précompte professionnel et/ou liée à un partenaire de recherche.
L'objectif est de tenir compte de la topologie financière de chaque sous-catégorie afin de modéliser la base de données.

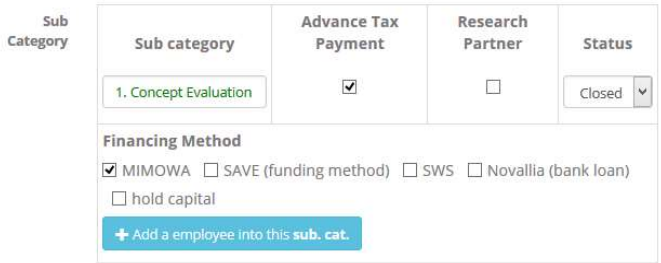

Figure 2 - Aperçu d'une sous-catégorie et de ses paramètres

La figure 2 illustre les informations liées à une sous-catégorie. Cette figure affiche également un statut en lien à la sous-catégorie ainsi qu'un bouton afin d'ajouter des employés à cette sous-catégorie. Cette partie est gérée grâce à d'autres algorithmes.

La modélisation des bases de données pour cette fonction est *partiellement* visible à la figure 3.

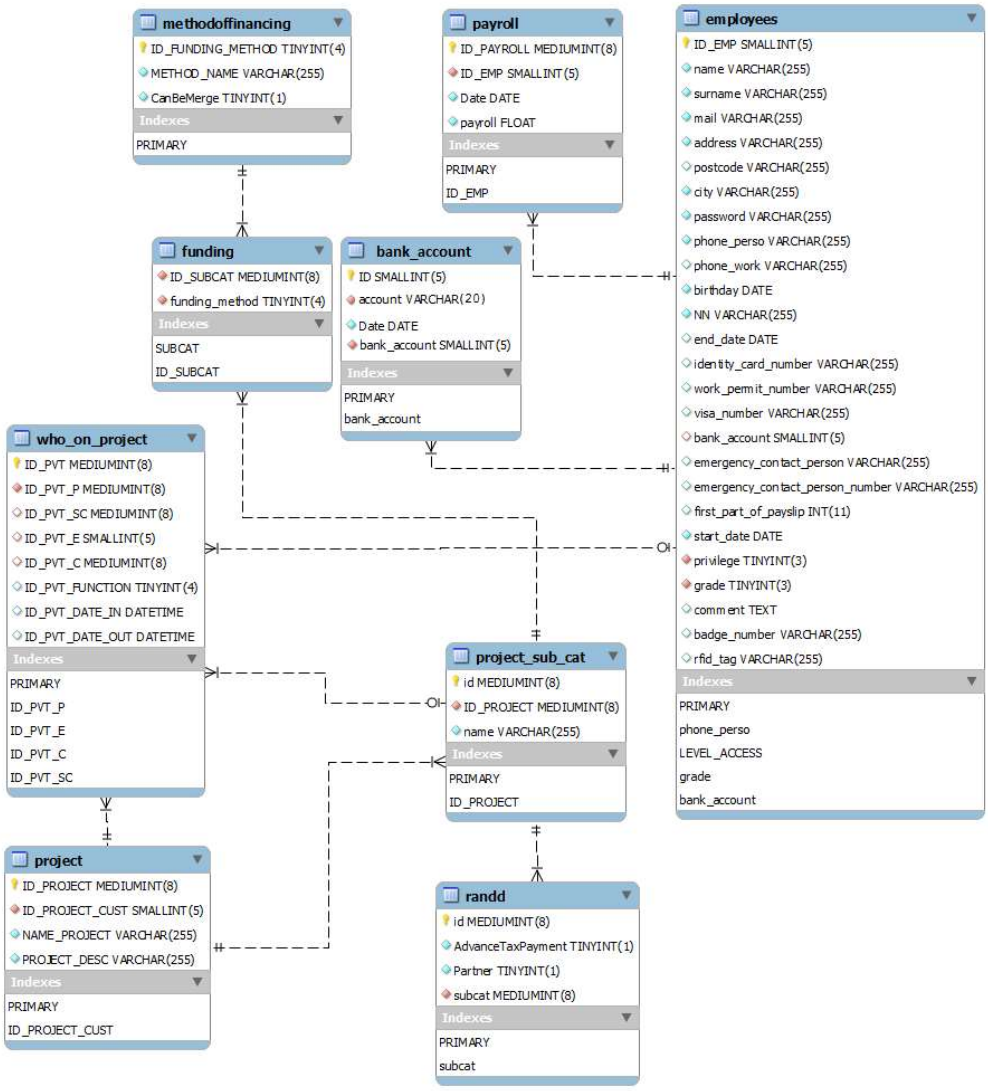

Figure 3 - Modélisation de la fonction

La figure 3 exprime la modélisation relationnelle [5]. Effectivement, toutes les tables sont reliées par au moins un lien. Comme prévu, cette architecture garantit une cohérence en évitant que les données soient redondantes. De plus, elle améliore considérablement la mise à jour des informations.
Afin d'extraire les financements liés à chaque sous-catégorie, une requête SQL est écrite en prenant compte les heures prestées sur chacune d'elles.

La requête utilise des opérateurs relationnels, dont les jointures, et les opérateurs ensemblistes (UNION).

T_EMP	T_R	NAME_PROJECT	name	T_SUB_CAT	METHOD_NAME	Employee
9	08:00:00	ADC14b1MHZ	Developpement design	18	MIMOWA	Valentin Dirken
8	08:00:00	ADC14b1MHZ	Feasibility	17	MIMOWA	Thomas Sevrin
8	08:00:00	LVDS	Developpement design	45	FORTIS	Thomas Sevrin
8	01:30:00	Business	Finance	97	Bank Loan	Thomas Sevrin
8	02:00:00	ADC14b1MHZ	Developpement design	18	MIMOWA	Thomas Sevrin
9	04:00:00	ADC14b1MHZ	Developpement design	18	MIMOWA	Valentin Dirken
5	08:00:00	Business	Marketing	98	Bank Loan	Julie Renard
1	08:00:00	LVDS	Developpement design	45	FORTIS	Maxime Forges
5	08:00:00	Business	Marketing	98	Bank Loan	Julie Renard
8	07:00:00	ADC14b1MHZ	Concept Evaluation	16	MIMOWA	Thomas Sevrin

Figure 4 - Résultat de la requête MySQL qui permet d'associer pour chaque sous-catégorie le(s) financement(s)

La figure 4 est le résultat de la requête. Une sous-catégorie, reliée au même projet, est présente plusieurs fois dans le résultat. C'est évident, car cette dernière peut être financée de plusieurs manières.

2.4. Affichage et mise en page des résultats

Le résultat MySQL est affiché, grâce au langage serveur PHP [8], dans une page HTML embellie par le CSS [7].

Project	Sub. Cat.	Funding	Month	Sum. days	Total Hours	Hours on it	%	Payroll	Montant à activer
No items	No items	No items	No items	No items	No items	No items	No items	No items	No items
[illegible]	1. Concept Evaluation	SWS	1	19.00	152:00	01:00	0.66	3000	19.74
[illegible]	Development	Novallia (bank loan)	1	19.00	152:00	40:00	26.32	3000	789.47

Figure 5 Affichage HTML et PHP de la requête SQL

Par ailleurs, l'entreprise souhaite réaliser des calculs en fonction des résultats générés. Par exemple, n'afficher qu'un financement ou les calculs relatifs à un seul employé.

Une requête supplémentaire, à chaque calcul, oblige de recharger la page. Une solution pour ne pas le faire, est d'utiliser la page générée en PHP en incluant une couche ergonomique. L'utilisation du JavaScript ainsi que les propriétés CSS permettent de pallier ce problème tout en améliorant les performances de l'application Web. En effet, le CSS permet de cacher ou non des lignes selon certains critères de recherche tandis que le JavaScript calcule le total des lignes visibles en se référant aux propriétés CSS de chaque ligne.

Ces fonctions implémentées calculent en temps réel comme sur un fichier *Excel*. Les valeurs calculées sont la somme du « Montant à activer », du nombre de lignes encore visibles ainsi que d'une moyenne.

Par ailleurs, des options de recherche comme trouver le début d'un texte, sélectionner un ou plusieurs critères sont ajoutées. Ces options utilisent pleinement JavaScript, CSS ainsi que les expressions régulières afin d'afficher ou non une ligne.

2.5. Vue générale d'une *timesheet*

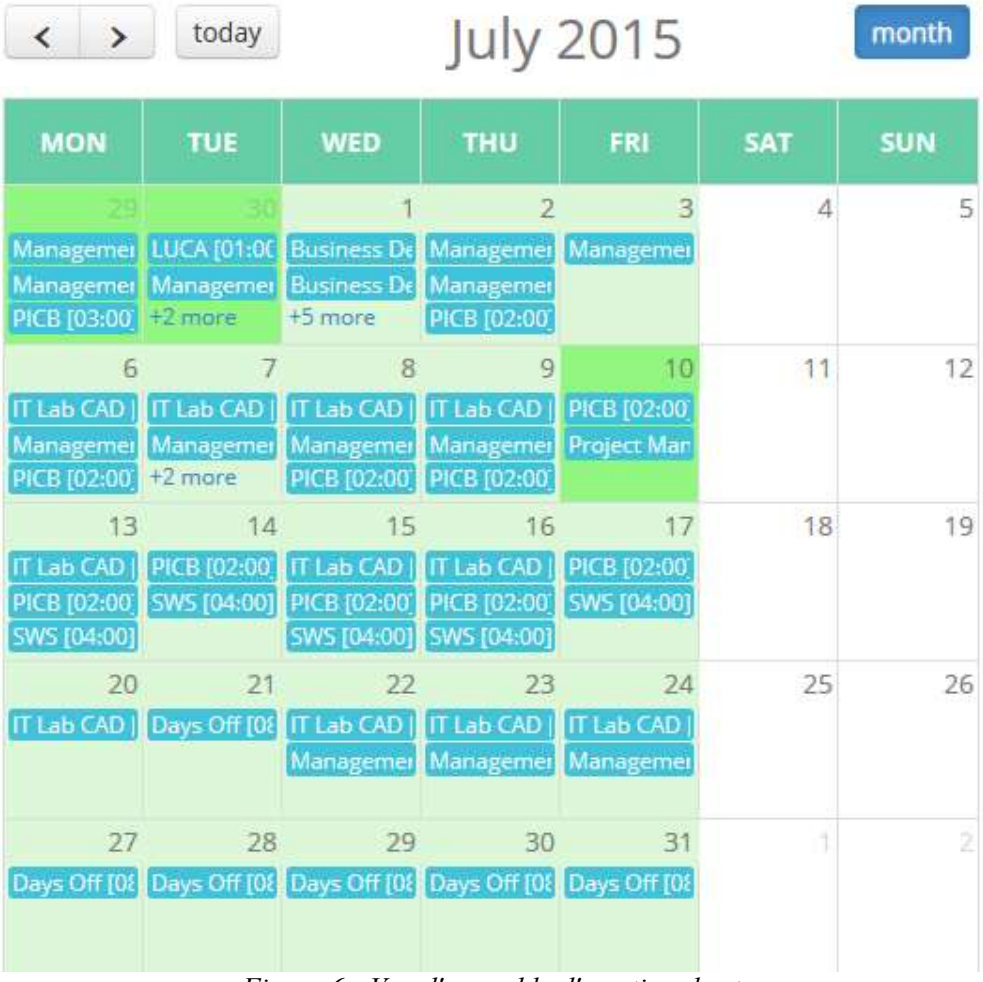

Figure 6 - Vue d'ensemble d'une timesheet

La figure 6 présente la vue générale d'une *timesheet* mensuelle d'un employé.

Afin que le système soit agréable à utiliser, l'utilisateur peut sélectionner une plage de jours. Lorsqu'il lâche le clic de souris, une fenêtre modale apparaît avec les dates déjà remplies.
L'avantage est double. D'une part, l'application améliore l'ergonomie de l'utilisateur, car la fenêtre modale permet à l'utilisateur de se concentrer sur l'information et de limiter temporairement ses actions sur la *timesheet*. D'autre part, ce choix augmente les performances de l'application Web en faisant appel aux requêtes asynchrones tout en évitant d'ouvrir une nouvelle page.

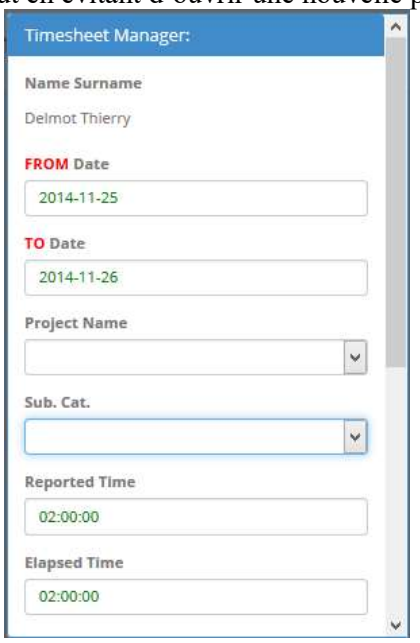

Figure 7 - Aperçu d'une fenêtre modale.

Il existe la méthode classique signifiant qu'à chaque action de l'utilisateur, une requête est envoyée au serveur qui renvoie, à son tour, le résultat sous forme d'une nouvelle page Web chargée vers le navigateur.
L'utilisateur attend que la page soit affichée avant qu'il puisse envoyer une nouvelle requête.

Cependant, il est possible d'utiliser JavaScript pour gérer les échanges entre le navigateur et le serveur Web sans que la page soit rechargée. C'est la méthode AJAX.

La méthode AJAX repose sur plusieurs technologies conjointes :

- *Document Object Model* offre la structure à l'utilisateur afin d'ajouter, modifier, supprimer du contenu d'une page Web déjà chargée et d'en modifier son CSS si nécessaire.

- JavaScript, pleinement utilisé dans l'application Web, permet d'exploiter le *Document Object Model*.
- *XMLHttpRequest* est un objet de programmation qui était initialement un composant ActiveX. Ce composant est fondamental dans une architecture AJAX. C'est l'élément qui permet de contacter le serveur Web sans que la page soit rechargée en créant une communication asynchrone entre le serveur Web et le navigateur.

Les avantages sont multiples : le temps de latence est diminué et la page n'est plus rechargée, ce qui améliore l'ergonomie.
C'est pourquoi l'application *timesheet* intègre pleinement cette technologie qui rend le système plus interactif, réactif et simple à l'utilisation.

Les requêtes asynchrones sont utilisées, essentiellement dans les fenêtres modales contenant des formulaires, afin d'effectuer des actions spécifiques comme la suppression, la duplication, l'ajout et la modification de données.
À titre d'exemple, la page est chargée complètement. Lors d'un clic sur un élément de la page, une requête asynchrone est exécutée en joignant un paramètre grâce à AJAX.
Ce paramètre est une variable PHP qui transite par l'URL vers la page *ajax_tms.php.* Une des fonctions écrites dans *ajax_tms.php* est sensible à cette requête et exécute, à son tour, une requête SQL selon le paramètre transmis.
Le résultat est renvoyé au navigateur en utilisant la méthode HTTP POST sans recharger complètement la page. Ajax met à jour une partie de la page Web sans la recharger complètement.

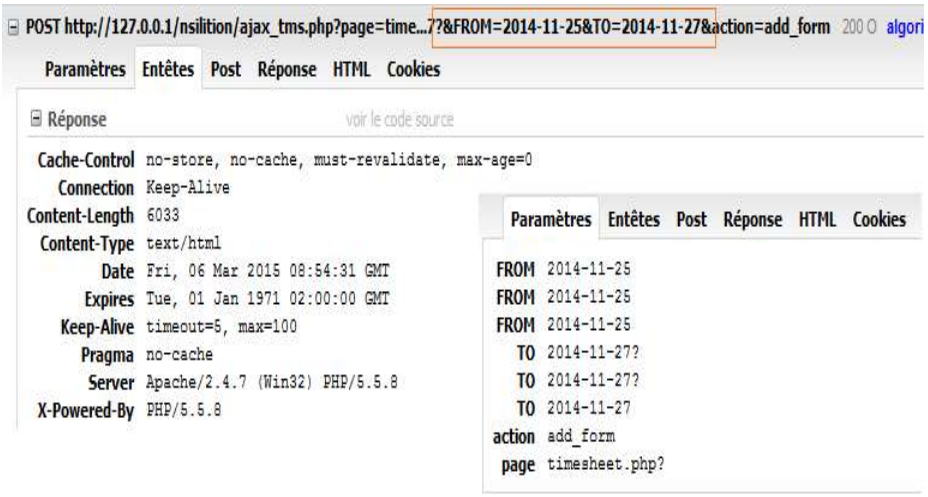

Figure 8 - Requête asynchrone

3. Accès sécurisé aux données

3.1. Introduction

Afin d'accéder de manière sécurisée aux *timesheets*, plusieurs solutions sont envisagées.
Pour rappel les 5 propriétés d'un système sécurisé [6] sont :

- Authentification : seuls les employés peuvent accéder aux ressources. Cette authentification se réalise grâce à un identifiant-mot de passe.
- Confidentialité : Les données ne sont pas accessibles aux personnes qui n'ont pas été identifiées.
- Intégrité : assure que l'information lue est correcte.
- Disponibilité : maintient la disponibilité et l'accès aux ressources. Ce point utilise notamment les firewalls.
- Non répudiation : garantit irrévocablement la personne qui s'est connectée.

3.2. État de l'art – connexion à distance

Voici une présentation générale des différentes solutions d'accès aux *timesheets*.
La première solution est la réplication partielle des données. Cette réplication est circulaire et se fait entre deux serveurs dont un est accessible depuis l'extérieur de l'entreprise.
Une seconde solution est que l'utilisateur utilise une connexion SSH, mais ajoute un paramètre (-X) afin qu'il communique avec le serveur d'affichage X11.
Une autre possibilité est d'utiliser un réseau privé virtuel en utilisant l'Internet comme un réseau de transport afin d'accéder au réseau interne de l'entreprise.
Plus précisément, les informations empruntent des infrastructures publiques de communication, mais des mécanismes d'identification, d'authentification, de chiffrement et de gestion des flux garantissent aux échanges la même confidentialité et la même fluidité qu'une ligne réellement privée.
Le résultat d'un VPN peut être assimilé à une connexion filaire entre deux réseaux.
Il existe 3 grandes solutions pour les VPN :

- **Des services** comme les FAI (Belgacom...). Ils utilisent les protocoles comme ATM ou MPLS et permettent d'interconnecter l'ensemble de deux sites distants de manière complètement sécurisée. La connexion se fait directement par le réseau opérateur. Les données ne transitent pas sur le Web. Une ligne physique est dédiée à la communication.
- **Des solutions matérielles** conçues spécifiquement pour cette tâche comme le Cisco VPN 3000
- **Des solutions logicielles** utilisant des protocoles comme l2TP couplé à IPSec.

3.3. Choix de la solution

La réplication des tables implique que certaines le soient également à distance.
En effet, afin que les utilisateurs puissent remplir leurs *timesheets*, ils ont besoin de connaître le nom des projets, le nom du client... Ce problème exige une réflexion globale concernant la réplication. À titre d'exemple, il est nécessaire d'importer à distance, le nom des projets, ainsi que les sociétés.

On ne peut pas autoriser cela, car rien que le nom du projet permet de comprendre ce que l'entreprise fait. De plus, les lois entre sociétés interdisent de divulguer des informations confidentielles (la sous-traitance peut altérer les stratégies de l'entreprise).

Une idée proposée est d'utiliser des noms de projets différents ainsi que des noms de sociétés différentes.
Exemple : LX = Linux. Microsoft : MCST, cependant, cette méthode est rejetée de suite, car une approche dans les télécommunications prouve que si la clé est connue, l'information peut être comprise.

Une autre méthode est de chiffrer le tout en AES (Advanced Encryption Standard) qui est une méthode de chiffrement symétrique. En d'autres mots, toutes les informations sont chiffrées avec cette clé. Cette solution n'est pas optimale vu qu'il faut stocker la clé et cela oblige de chiffrer à nouveau toutes les données sur le serveur si une personne quitte l'entreprise.
De toute évidence, il faudrait chiffrer la communication de bout en bout en respectant les critères de la sécurité.
La solution suggérée est que les employés se connectent auprès du serveur central qui contient toutes les informations de manière sécurisée. Cette méthode évite les réplications ainsi que la présence d'informations confidentielles disponibles sur Internet, même chiffrées sur un serveur distant.

La connexion SSH utilisant le protocole X11 est intéressante et facile d'implémentation. Cependant, les attaques sont possibles :

- Déni de service qui consiste à paralyser le serveur en envoyant constamment des requêtes.
- Attaque par Brute Force qui consiste à essayer de se connecter avec un mot de passe.
- Une autre attaque connue est l'exploitation du canal caché.
- De plus, il est nécessaire qu'un logiciel soit installé comme Cygwin/X sur Windows. Cependant, Mac OS et Linux prennent en charge cette fonctionnalité nativement.

En conclusion, le choix s'est porté sur une connexion de type VPN. En effet, une réplication ne respecte pas la confidentialité des données. Un tunnel SSH est sensible aux attaques par Brute Force en l'occurrence. Lors des tests, le serveur a été attaqué par ce principe, mais il n'a pas été compromis.

3.4.Topologie du réseau

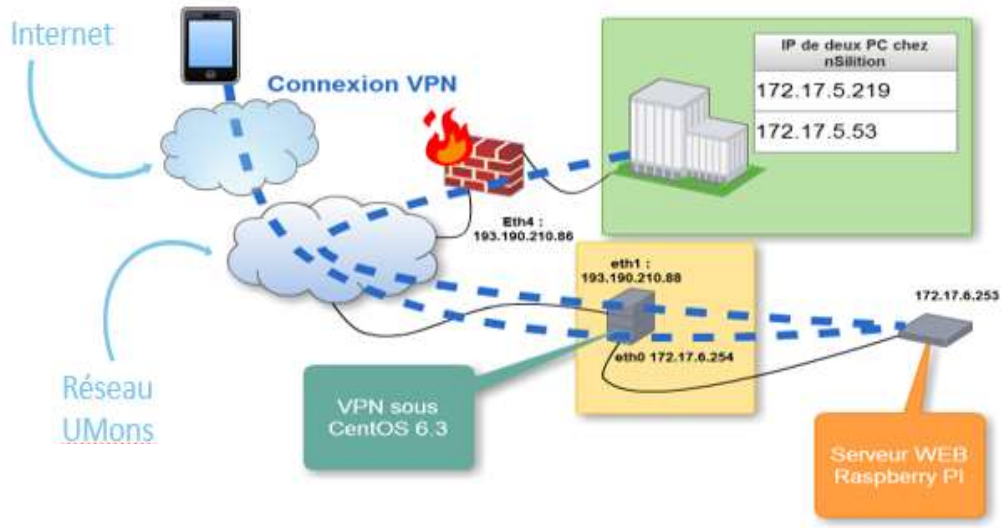

Figure 9 - Topologie de l'infrastructure

La topologie du réseau présentée ci-dessus est la solution mise en place au sein de l'entreprise lors des tests.

Ainsi, un employé se connecte à l'IP publique fournie par l'entreprise avec son *smartphone* ou son ordinateur. Lorsque la connexion est établie avec succès à travers le *firewall* [2] [3], il peut contacter l'adresse privée du serveur Web qui contient l'application Web sous une adresse locale non routable

3.5. Implémentation du VPN

L'infrastructure repose sur un serveur CentOS version 6.3 décidée en interne. Le software VPN utilisé dans le projet est OpenSwan [1]. Il est libre, européen et gratuit.

Différentes authentifications

Il existe 3 méthodes d'authentification utilisées dans OpenSwan pour authentifier les deux parties :

- La clé prépartagée PSK qui est une authentification simple. Une clé est définie entre le serveur et les employés. La négociation se fait en testant cette clé symétrique prépartagée. Si un employé quitte l'entreprise, il faut créer une nouvelle clé.
- Les clés asymétriques sont utilisées pour une architecture statique. Par exemple entre deux serveurs. Ce n'est pas notre cas. De plus, ce système n'est pas adéquat quand il faut gérer de plus en plus de machines. En effet, les machines sont configurées manuellement. Si une seule clé est utilisée, cela revient au même principe que le point précédent, mais avec des clés asymétriques.
- Les certificats X.509 sont utilisés lorsqu'il y a beaucoup de machines. Les certificats apportent, en plus d'une authentification par clés asymétriques, la non répudiation grâce à une signature numérique entre l'hôte et les utilisateurs, limitée dans le temps. Les certificats, signés par une autorité, sont hiérarchisés et peuvent être révoqués.

L'authentification utilisant des certificats X.509 est implémentée, car elle est la plus robuste, plus facile à mettre en place à grande échelle et respecte les 5 propriétés d'un système sécurisé.

Implémentation de la solution par certificats

La première étape est d'utiliser une autorité de certification (appelé CA en anglais). Le CA permet, après vérification de l'identité, de signer les certificats CSR (Certificate Signing Request) ainsi que les CRL (CRL signifie Certificate Revocation List) si besoin.

De nombreux CA sont disponibles comme Verisign (Symantec) ou, en Belgique, le Certipost qui délivre des certificats d'autorité. Ces certificats coûtent assez cher. Pour information, les certificats d'autorité coûtent aux alentours de 1500 euros pour celui de Verisign. C'est pourquoi le certificat CA est autosigné pour les besoins du test.

Dorénavant, deux méthodes sont possibles pour générer les certificats ainsi que les clés. La librairie *certutil* est utilisée, car elle crée les certificats et les gère directement dans la base de données NSS. La librairie *OpenSSL* peut aussi être utilisée. Le résultat est équivalent, seule l'écriture est différente.

Par ailleurs, les employés se connectent au VPN avec une IP dynamique, ce qui implique qu'il est nécessaire de gérer des connexions *roadwarriors*.
Le VPN mis en place est un tunnel L2TP over IPsec utilisant une authentification par certificats et MS-CHAPv2 (couple nom d'utilisateur – mot de passe).

Les certificats sont créés selon la procédure suivante :

- Générer la clé privée du CA
- Créer un CSR CA
- Autosigner le CA avec la clé privée CA.
- Créer une clé privée ainsi qu'un CSR pour le VPN
- Signer le certificat VPN grâce au CSR, sa clé privée ainsi que le CA autosigné.
- Réitérer les deux points précédents pour chaque client en changeant l'identité.
- Exporter le certificat client, sa clé privée ainsi que le CA autosigné dans un format PKCS#12
- Cacher la clé privée du CA afin d'éviter de compromettre l'infrastructure à clés publiques mise en place. (PKI : *Public Key Infrastructure)*
  Ensuite, le certificat est importé dans l'ordinateur de l'employé.

Enfin, il est nécessaire de configurer les fichiers IPsec, L2TP, … d'Openswan.

Résultats

La connexion à un autre réseau local, à travers l'Internet, en utilisant un tunnel VPN est un succès.
Cependant, les iPhone ne gèrent pas la certification x509 nativement (sans installation de programmes tiers) contrairement aux smartphones sous Android et aux ordinateurs sous Windows/Unix.

Comportement du tunnel

Afin de vérifier l'infrastructure mise en place, la connexion VPN par certificats est analysée.

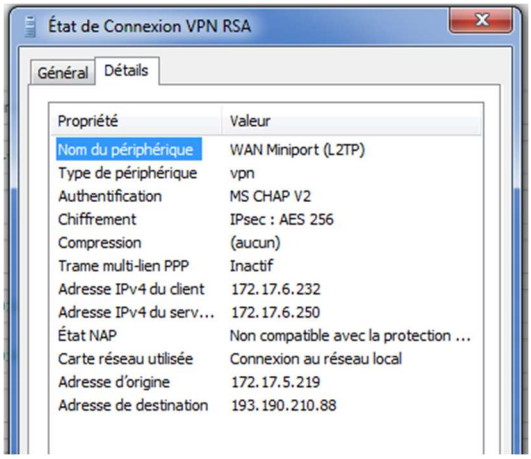

Figure 10 - État de la connexion VPN par certificats

La figure 10 illustre les points précédents :

- Il existe bien une authentification de type MS-CHAPv2
- Le chiffrement des paquets est en IPsec AES256 (grâce à une propriété écrite dans l'IPsec).
- L'adresse IPV4 du client est 172.17.6.232. En effet, le service L2TP a attribué à cette connexion la première adresse disponible sur sa plage. La configuration de la plage se trouve dans le fichier *etc/xl2tpd/xl2tpd.conf*
- L'adresse du serveur qui est 172.17.6.250 est spécifiée dans le fichier *etc/xl2tpd/xl2tpd.conf*
- L'adresse d'origine est en fait l'adresse locale de l'ordinateur
- L'adresse de destination est celle du serveur VPN : 193.190.210.88
- Une analyse du réseau prouve que tous les paquets sont chiffrés en ESP (IPsec). Ce chiffrement est configuré dans le fichier IPsec.

4. Conclusion

Le projet a pour but, d'une part de réaliser une application Web permettant d'ajouter les temps de travail des employés et ce depuis une IP extérieure à l'entreprise.
La conduite du projet a nécessité 3 grandes étapes.

La première étape est une analyse des besoins. Cette étape a permis de décomposer ce projet en plus petites étapes, en plusieurs fonctions.

La seconde étape est de développer l'application Web. Cette étape comprend l'analyse, la conception et l'implémentation de fonctions qui sont la modélisation des bases de données et le développement de l'application grâce aux langages PHP, MySQL, HTML et CSS. Une couche d'ergonomie est ajoutée grâce au langage JavaScript qui utilise également les requêtes asynchrones.

Le système *timesheets* peut évoluer en incluant par exemple une authentification LDAP.

D'une autre part, le projet consiste à étudier et implémenter un accès sécurisé à l'application. Premièrement, afin de définir la meilleure solution, les protocoles de communication ainsi que les architectures possibles ont été étudiés. Le choix s'est porté sur un réseau privé virtuel (VPN). Son implémentation s'est faite avec une méthode par certificats qui garantit un meilleur niveau de sécurité par rapport à une clé prépartagée ou une clé asymétrique. À contrario, cela implique une plus grande complexité de mise en œuvre. Une amélioration possible serait d'augmenter la sécurité en utilisant des certificats multi-niveaux, une authentification par challenge comme le *One Time Password* ou des tunnels comme IKEv2.

5. Sources

[1] Paul Wouters and Ben Bantoft, Openswan : Building and Integrating Virtual Private Networks: Learn from the developers of Openswan how to build industry standard, military grade VPNs with Windows, MacOSX, and other VPN vendors, 2006.

[2] Gregor N. Purdy, *Linux iptables Pocket Reference*. O'Reilly Media, 2004.

[3] Elizabeth D. Zwicky, *Building Internet Firewalls*. O'Reilly Media, 2nd edition, 2000.

[4] Thomas M. Connolly, Carolyn Begg and Anne Strachan, *Database Systems: A Practical Approach to Design, Implementation and Management*. Addison-Wesley, 1995.

[5] Jean-Luc Hainaut, Bases de données : Concepts, utilisation et développement. Dunod, 2009.

[6] Jandrew Tanenbaum and David Wetherall, *Réseaux*. Pearson, 5ème edition, 2011

[7] *W3Schools Online Web Tutorials*, (Mars 2015), http://www.w3schools.com/

[8] *PHP: Documentation*, (Mars 2015), http://php.net/docs.php

[9] *MySQL 5.7 Reference Manual*, (Mars 2015), http://dev.mysql.com/doc/refman/5.7/en/