
MÉMOIRE

Alternance du 17/09/14 au 16/09/15 au sein de



Titre RNCP Niveau II

Administrateur Systèmes, Réseaux et Base de Données

Session 2015 – Soutenance du 1^{er} juillet

8 juillet 2015

Remerciements

Je souhaite tout d'abord remercier l'entreprise XYMOX System de m'avoir accueilli au sein du pôle développement et permis d'effectuer mon contrat de professionnalisation dans d'excellentes conditions.

Plus précisément, je remercie Olivier Seine, Benoît Rolland et Sandrine Pauvert, respectivement Président, Directeur Général et Responsable du centre de Nantes, pour leur confiance et l'état d'esprit qu'ils diffusent chez XYMOX System.

Je remercie particulièrement Sandrine Pauvert, dont les nombreux « point planning » m'ont permis d'améliorer ma prise de note, mon organisation ainsi que ma gestion du temps. J'ai aussi particulièrement apprécié son mode de management.

Je remercie tout particulièrement mon tuteur, Jean Lemétayer, qui a fait preuve de beaucoup de patience, de disponibilité, et surtout de pédagogie pour me pousser à chercher par moi-même tout en me donnant des indices aux bons moments. Cette manière de procéder, au départ déroutante, s'est avérée finalement très efficace.

Mes remerciements vont aussi au reste de l'équipe de XYMOX System, qui contribue par sa simplicité, sa convivialité et son accueil chaleureux à la très bonne ambiance qui règne au sein de l'entreprise.



Enfin, mes remerciements s'adressent à toute l'équipe pédagogique et administrative de l'EPSI Nantes.

Sommaire

Introduction.....	3
XYMOX System	4
1. Effectif et structure	4
2. Philosophie.....	5
3. Activités	5
4. Produits	6
Environnement de travail.....	8
1. Conventions et bonnes pratiques.....	8
2. Principaux sites utilisés	8
3. Principaux logiciels et services utilisés.....	9
Mission 1 : refonte des temps prévisionnels	10
1. Situation initiale	10
2. Modèle et DAO	10
3. Intégration dans l'existant.....	11
4. Difficulté rencontrée	11
Mission 2 : refonte du module Statistiques.....	12
1. Objectif	12
2. Conception et planification	12
3. Modélisation et création des tables.....	13
4. Classes métiers et DAO	14
5. Gestion des droits d'accès	17
6. Fonctionnalité d'export	17
7. Contrôleur PHP et gestionnaire de vues HTML	18
8. Contrôleur JavaScript.....	19
9. Barre de filtres	20
10. Mise en forme des histogrammes	21
11. Mise en forme des tableaux	22
12. Recette et livraison.....	24
13. Difficultés rencontrées.....	25
Mission 3 : nouveau module Sondage	26
1. Besoin client.....	26
2. Spécifications fonctionnelles	26
3. Spécifications techniques.....	32
4. Difficulté rencontrée	33
Mission 4 : missions diverses.....	34
Conclusion.....	35
Glossaire	36
Annexes	38
ANNEXE 1 : rapport « Performance ».....	38
ANNEXE 2 : droits d'accès au niveau société.....	39
ANNEXE 3 : droits d'accès au niveau du rôle	40
ANNEXE 4 : Rapport « Temps passés »	41

Introduction

En septembre 2014, j'ai intégré le pôle développement de l'entreprise XYMOX System dans le cadre de ma troisième année d'informatique à l'EPSI Nantes. J'ai rejoint l'équipe dédiée au développement de l'outil Tick&Help, que XYMOX System souhaitait renforcer dans l'optique d'une commercialisation début 2015.

Ma première journée fut consacrée à la présentation de l'entreprise, du produit et à la découverte de mon environnement de travail. Je me suis ensuite formé sur le **framework PHP** de l'application en développant des petites évolutions (bouton au lieu de lien, couleur de fond de page, etc.). Puis j'ai été chargé de réaliser ma première mission : la refonte de la gestion des temps prévisionnels. Nous verrons dans cette synthèse que d'autres missions ont été réalisées suite à la finalisation de ce module. Cela m'a permis de mettre en pratique d'autres compétences acquises cette année, comme la création de **triggers** en **PGSQL**.

En premier lieu, je présenterai la société XYMOX System, son histoire, sa philosophie, son organisation structurelle, etc. J'accompagnerai cette présentation de son organigramme. Dans les parties suivantes je présenterai mes différentes missions, notamment le développement d'un nouveau module.

Enfin, j'apporterai en conclusion un regard sur mon parcours, la réalisation de mes missions ainsi que mes objectifs professionnels à court et moyen terme.

Pour plus de clarté, les mots en caractères **gras soulignés** sont définis dans le glossaire.

Je vous souhaite une bonne lecture !

XYMOX System

XYMOX System (SAS au capital de 36 000 €) est une **SSI** éditeur et société de services informatiques créée en 2001 par deux associés, puis acquise en 2012 par Olivier Seine via DIROSE, sa société de conseil en stratégie et de prise de participation.

L'entreprise divise ses activités entre ses deux centres :

1. Mulhouse : siège social spécialisé dans la partie réseau et l'hébergement,
2. Nantes : agence axée développement qui concentre toutes les autres activités.

1. Effectif et structure

XYMOX comprend 24 collaborateurs et a réalisé en 2014 2 millions d'euros de chiffre d'affaires, pour une croissance portée par ses fonds propres et une rentabilité nette supérieure à 10 %. Voici son organigramme en 2015 :



Figure : organigramme 2015 de XYMOX System

2. Philosophie

XYMOX System a été créée pour pallier des manques techniques et fonctionnels identifiés par les fondateurs dans leur société respective. La culture du « besoin client » a donc été au cœur du lancement et du développement de l'entreprise. La qualité des prestations fournies et la disponibilité des équipes sont les piliers de la relation client. Pour renforcer ces fondamentaux, XYMOX System s'est entourée d'une équipe compétente, professionnelle et disponible à qui elle reverse chaque année 1/3 de ses résultats, reconnaissance de la qualité de son travail et de son implication. L'entreprise contribue également à des actions pour la création d'entreprise en reversant une petite partie de son chiffre d'affaires à [Jeunes Pousses](#), une association qui aide de jeunes entrepreneurs à créer leur business.

3. Activités

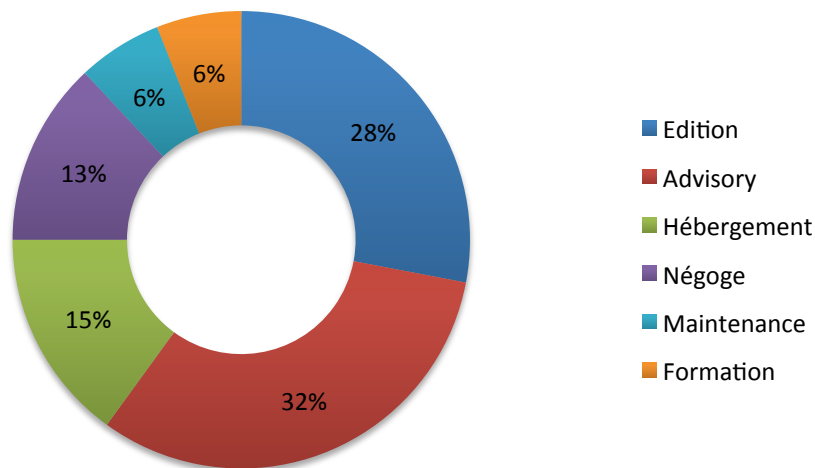


Figure : répartition du CA de XYMOX System par activité en 2014

4. Produits

MARINE

MARINE est un **ERP/PGI** qui garantit la couverture des activités d'un collecteur du 1 % logement et la simplicité d'exploitation, grâce à son intégration dans une solution globale ERP déjà existante. Cela assure une cohérence absolue des informations, une transversalité des flux et un pilotage efficace des activités. Le logiciel s'impose comme une solution intéressante sur le marché **ERP** pour les CIL, face à de grands éditeurs, grâce à des fonctionnalités spécifiques pour Action Logement, une migration éprouvée, un suivi permanent des évolutions réglementaires et des besoins utilisateurs.

MARINE est le produit principal de la société, développé en Progress et maintenu par la majorité des développeurs de XYMOX System. Il est actuellement utilisée par 8 clients (500 utilisateurs potentiels) et représente 31 % du CA annuel. Parmi les clients, on peut citer Logéo, CIL Val de Loire ou encore MNH.

EMMA

EMMA est un **PGI** pour les bailleurs sociaux et privés simple à mettre en œuvre et à utiliser. Ses différents modules métiers le transforment en un outil de pilotage de l'activité de gestion locative. EMMA comprend les modules suivants :

- Gestion du Patrimoine,
- Gestion de la Demande de Logement,
- Gestion Locative,
- Gestion Comptable et Financière,
- Gestion Budgétaire et Prévisionnelle,
- Comptabilité Analytique par Programme et par Opération,
- Gestion Financière des Opérations,
- Gestion des Projets d'Investissement,
- Gestion des Immobilisations,
- Editions des Etats Réglementaires.

Enfin, les outils graphiques de requêtes intégrés permettent aux collaborateurs de restituer l'ensemble des informations présentes dans la base de données directement sous Excel ou sous Word.

En raison de son ancienneté, aucune nouvelle évolution n'est apportée par XYMOX System, qui a stoppé sa commercialisation, mais continue le support client.

Tick&Help

Tick&Help est né d'un besoin de l'entreprise. Celle-ci recherchait un outil de suivi d'incident adapté aux besoins d'une PME. Cependant les outils déjà présents sur le marché n'étaient pas satisfaisants, souvent trop légers ou au contraire bien trop complexes et onéreux. C'est pour cela qu'en 2010 le développement de l'application a débuté. Au départ, l'application était destinée uniquement aux besoins internes. Au fil du temps, les dirigeants se sont rendu compte que Tick&Help pouvait intéresser d'autres PME. C'est alors que l'idée de commercialiser en mode **SaaS** l'application est apparue.

L'application dispose d'une gestion de tickets, de sociétés, de produits, d'utilisateurs, de planning et de statistiques. Elle se veut complète, mais relativement intuitive.

Aujourd'hui Tick&Help compte plusieurs « clients tests », c'est-à-dire des entreprises pour lesquelles l'outil a été mis en place. Les retours de ces « testeurs » permettent de faire évoluer l'application dans l'optique de toujours satisfaire au mieux les besoins des clients. Le logiciel est entré en phase de commercialisation en avril 2015.

Tick&Help est développé en **PHP/HTML/CSS/JavaScript** sur la base d'un **framework** maison orienté objet **MVC 2**. Au niveau base de données, c'est un serveur **PostgreSQL** qui a été choisi. Le logiciel présente différents niveaux d'accès appelés **rôles**.

Afin d'améliorer le travail en équipe, un gestionnaire de version a été mis en place : dans un premier temps Subversion (SVN), puis **GIT**.

N°	Créé	Délai de rappel	Utilisateur	Société	Résumé	Produit
En cours 1 ticket(s)						
9530	22/10/2012 16:05					Marine
Analyse 138 ticket(s)						
19052	19/06/2015 15:24					Marine
19043	19/06/2015 12:13					Hébergement
19042	19/06/2015 11:18					Marine
19040	18/06/2015 16:21					Marine
19037	18/06/2015 14:42					Emma
19031	17/06/2015 17:26					Marine
19026	17/06/2015 14:17					Marine
19020	16/06/2015 14:31					Marine
19017	16/06/2015 11:08					Marine
19016	16/06/2015 10:54					Marine
19013	16/06/2015 10:02					Marine
19000	15/06/2015 11:06					Marine
18989	12/06/2015 13:15					Marine
18972	10/06/2015 14:43					Marine
18936	04/06/2015 12:17					Marine
18931	03/06/2015 16:13					Marine
18928	02/06/2015 18:06					WEB MARINE
18913	01/06/2015 17:21					Marine
18895	29/05/2015 14:31					Cabane Chic
18891	29/05/2015 09:54					Marine
18882	28/05/2015 15:22					Marine
18771	07/05/2015 16:21					Marine

Impression d'écran : page listant les tickets sur Tick&Help

Environnement de travail

Le choix de notre environnement de travail n'était pas spécialement imposé, mais XYMOX System fournit à tout nouveau développeur un ordinateur déjà configuré par M. Lemétayer, avec Windows 7, la base de données **PostgreSQL** avec PGAdmin, l'**IDE Eclipse** et d'autres logiciels spécifiques à la configuration réseau de l'entreprise. J'ai personnellement décidé d'utiliser PHPStorm comme IDE plutôt qu'Eclipse, ayant un meilleur ressenti à l'utilisation de ce dernier.

1. Conventions et bonnes pratiques

M. Lemétayer nous a demandé de respecter un certain nombre de règles dans notre code source :

1. Utilisation des espaces (doubles) à la place des tabulations,
2. Ecriture des commentaires et noms de tables en français,
3. Placement de l'accolade ouvrante d'une méthode sur la première ligne,
4. Commentaire multi ligne précédant chaque déclaration de propriété,
5. Indentation du code dans chaque condition/méthode/objet...,
6. Limitation à 80 caractères maximum par ligne,
7. Utilisation du camelCasing pour les attributs et méthodes,
8. Utilisation du PascalCasing pour le nom des classes,
9. Utilisation des majuscules et tirets pour les constantes,
10. Utilisation du snake_casing pour les colonnes de tables,
11. Les propriétés ou méthodes privées sont de préférence préfixées par un _.

2. Principaux sites utilisés

Nom	Utilité
T&H (dev)	Version locale de l'application (hébergée sur notre ordinateur), qui nous permet de tester nos développements avant de les envoyer sur le serveur GIT .
T&H (prod)	Version en production de l'application hébergée sur les serveurs de XYMOX System. Elle permet de prendre connaissance des tickets attribués, de l'avancement des tâches et de noter les heures effectuées.
T&H (tests)	Page recensant les pages de tests, par exemple pour vérifier le bon fonctionnement des DAO ou d'une fonction complexe.
Documentation	J'ai utilisé la documentation officielle de PHP , de MDN , de jQuery , du W3C , de PostgreSQL , de jQuery EasyUI ou encore de JQPlot .

3. Principaux logiciels et services utilisés

Nom	Définition - Utilisation
Google Chrome Mozilla Firefox Internet Explorer	Navigateurs utilisés en binôme pour comparer le rendu CSS lors de modifications des vues. Les extensions <u>Xdebug</u> pour Chrome et <u>Firebug</u> pour Firefox ont permis de lancer du pas à pas PHP . Pour la partie JavaScript, la console native des navigateurs pouvait suffire, même si <u>Firebug</u> est aussi à même de remplir la fonction sous Firefox. Pour les tests de comptabilité CSS, nous avons utilisé Internet Explorer 9.
PostgreSQL™ 8 PGAdmin	Nous utilisons PostgreSQL 8.4 pour des raisons de comptabilité avec l'existant. La mise à jour en version 9 est prévue, mais pas à l'ordre du jour étant donné la phase de commercialisation du produit. <u>PGAdmin</u> est un logiciel communautaire multi plateformes apportant une couche graphique à l'utilisation de cette base de données.
compass SASS	compass est un logiciel en ligne de commande capable de convertir des fichiers au format SASS en fichiers CSS lisibles par le navigateur. SASS est un préprocesseur CSS étendant les possibilités de base de CSS via mixins, variables, inclusions, etc.
WinSCP	Gratuitiel disponible sous Windows qui nous a permis depuis nos configurations d'envoyer les fichiers de l'application sur le serveur en SFTP (utilisation de SSH).
PHP 5	La version que nous utilisons est la 5.3.28 (2013), ceci pour des raisons de compatibilité avec l'ancien code de <u>Tick&Help</u> . La mise à jour en version 5.5 est prévue, mais pas à l'ordre du jour étant donné la phase de commercialisation du produit.
PuTTY	Client SSH qui est utilisé afin de se connecter à la base de données et exécuter les requêtes nécessaires.
Office 2007	Suite logicielle utilisée pour rédiger la documentation liée aux développements ou pour réaliser le <u>recettage</u> .
Microsoft Paint	Logiciel de dessin pixel utilisé pour réaliser les différents montages d'écrans et maquettes de nouvelles fonctionnalités
GIT	Nous utilisons les branches pour séparer le développement de plusieurs modules.
Skype	Très populaire chez XYMOX System, Skype est utilisé par pratiquement tout le monde et permet de communiquer avec le centre de Mulhouse ou d'envoyer facilement des fichiers d'un poste à un autre sans passer par mail ou clef USB.

Mission 1 : refonte des temps prévisionnels

Ma première mission a été de redévelopper la gestion des temps prévisionnels à partir d'un **CDC** déjà réalisé.

1. Situation initiale

Au moment de la refonte, la gestion des temps prévisionnels se fait via une simple colonne `forecast_time` dans la table `ticket`. Cela ne tient pas compte d'un opérateur ou encore d'une tâche.

2. Modèle et DAO

```
hl.forecast (#ticket_id, #membre_id, #tache_id, duration_mn)
```

J'ajoute dans la base de données la table `forecast_time`. Cette table permet de déterminer la durée prévisionnelle de chaque tâche par rapport à un Opérateur.

Je réalise ensuite le DAO, que je simplifie au maximum grâce à l'utilisation de la fonction **PGSQL** suivante :

```
CREATE OR REPLACE FUNCTION hl.forecast_managment(ticket_id_integer, membre_id_integer, tache_id_integer, duration_integer)
  RETURNS void AS
$BODY$
BEGIN
  -- Récupération des temps passés
  PERFORM *
  FROM    hl.forecast
  WHERE   ticket_id = ticket_id_
  AND     membre_id = membre_id_
  AND     tache_id = tache_id_;

  -- Insertion
  IF NOT FOUND AND duration_ != 0 THEN
    INSERT INTO hl.forecast(ticket_id, membre_id, tache_id, duration_mn)
      VALUES(ticket_id_, membre_id_, tache_id_, duration_);
  ELSE
    -- Suppression
    IF duration_ = 0 THEN
      DELETE
      FROM    hl.forecast
      WHERE   ticket_id = ticket_id_
      AND     membre_id = membre_id_
      AND     tache_id = tache_id_;
    -- Modification
    ELSE
      UPDATE  hl.forecast
      SET     duration_mn = duration_
      WHERE   ticket_id = ticket_id_
      AND     membre_id = membre_id_
      AND     tache_id = tache_id_;
    END IF;
  END IF;
END;
$BODY$
```

Capture d'écran : fonction de gestion des temps prévisionnels de [Tick&Help](#)

Cette fonction se comporte différemment en fonction des arguments passés en paramètre. Dans un premier temps, si le SELECT effectué à l'aide des paramètres ne retourne aucun résultat et que la durée passée en paramètre est différente de 0, elle insère un nouveau temps prévisionnel. Ensuite, si la durée est égale à 0 elle supprime le temps, sinon elle le modifie. Ce genre d'optimisation est un compromis qui permet de soulager les traitements côté applicatif, tout en alourdissant par conséquent ceux de la base de données.

3. Intégration dans l'existant

Ce développement a été réalisé par M. Lemétayer, tandis que je me suis consacré à la refonte du module Statistiques (voir mission suivante), étant donné l'importance de ce second projet. M. Lemétayer a ajouté une nouvelle section sur la page d'un ticket, visible sur l'image ci-après :



Temps prévisionnels et réalisés			
Tâches	Opérateurs	Temps prévisionnels	Temps passés
Totaux		-	-

Afficher les détails Masquer Ajouter Modifier Supprimer

Capture d'écran : section des temps sur la page du ticket dans [Tick&Help](#)

4. Difficulté rencontrée

Il a été difficile d'optimiser correctement la fonction `forecast_management`. Par habitude, j'avais commencé par développer un DAO complet et je n'avais pas pensé au moyen de simplifier ces traitements en passant par une fonction.

Mission 2 : refonte du module Statistiques

Le module Statistiques est un module qui existait déjà sur l'application à mon arrivée, mais celui-ci nécessitait une refonte complète pour la commercialisation de Tick&Help.

1. Objectif

L'objectif de ce développement était la création de plusieurs « rapports » destiné aux différents **rôles**. En fonction du rôle, le rapport a une finalité différente :

- Administrateur : permet d'avoir une vision globale de l'activité des équipes d'un point de vue performance, qualité et niveau de services,
- Opérateur : permet à une personne en charge du traitement de tickets d'avoir les informations lui permettant d'améliorer sa performance opérationnelle (nombre de tickets affectés par période, délai moyen de prise en charge, etc.),
- Référent/Utilisateur : permet aux utilisateurs de disposer de statistiques sur les tickets qu'ils ont créés (ou leur société dans le cas d'un Référent).

2. Conception et planification

XYMOX System a engagé Alexandre de Kermadec pour poser les bases de la conception de la nouvelle version du module Statistiques. M. de Kermadec a commencé par étudier le besoin réel de XYMOX System et de ses clients, puis il a rédigé un dossier de **SFD** qu'il a présenté et expliqué lors d'une réunion avec l'équipe en charge de Tick&Help.

Spécifications fonctionnelles

M. de Kermadec propose dans son dossier un ensemble de 5 rapports :

1. Le rapport « Performance » : permet de vérifier que le niveau de services correspond aux engagements (flux de tickets correctement traités). Ce rapport présente, par exemple, sur une période choisie par l'utilisateur, le nombre de nouveaux tickets créés, le nombre tickets résolus, le délai moyen de résolution, etc. (ANNEXE 1).
2. Le rapport « Temps passés » : permet le pilotage interne du traitement des tickets en matière d'avancement, de charge de travail et d'écart prévisionnel/réel. Parmi les indicateurs qu'il présente, on peut trouver le temps passé sur la période, le temps passé total (depuis la création du ticket) et le temps prévisionnel (temps de travail estimé pendant la période). Ces indicateurs sont très utiles pour la facturation.
3. Le rapport « Opérateur » : apporte une vision synthétique de l'activité d'un Opérateur sur une période donnée. On retrouve des indicateurs tels que le nombre de tickets affectés sur la période ou encore le nombre de tickets en stock (tickets non résolus en fin de période).
4. Le rapport « Temps opérateur » : sur la base du rapport « Temps passés », précise le champ d'application à un seul Opérateur.

5. Le rapport « Utilisateur » : apporte à un Utilisateur des statistiques sur les tickets qui le concerne (auteur), toujours avec le choix de la période étudiée. Il peut ainsi voir sur une période donnée le nombre de tickets qu'il a créé, ceux qui ont été résolus, etc.

Ces rapports sont proposés avec une base commune : la barre de filtres. Les seuls éléments non fixes sont les filtres, qui sont fonction du rapport. Le bouton « Export » est prévu pour permettre d'exporter les données dans un fichier Excel.

Pour filtrer les accès aux statistiques, M. de Kermadec ajoute une gestion des droits d'accès séparée en deux écrans :

- Ecran 1 : tableau permettant de définir quel rôle dans la société peut activer/désactiver ou définir par défaut un rapport (ANNEXE 2),
- Ecran 2 : tableau permettant de définir quel client de la société utilisatrice de Tick&Help peut accéder aux rapports (ANNEXE 3).

Spécifications techniques

Le travail effectué par M. de Kermadec a permis à M. Lemétayer d'estimer les charges. Nous avons donc formé un binôme avec Benoit Sauvage (« Benoit »), également en contrat de professionnalisation chez XYMOX System, et rédigé dans un premier temps le **CDC**. Puis M. Lemétayer a découpé le développement en deux parties, une concernant les droits d'accès aux différents rapports et une autre concernant tous les rapports.

3. Modélisation et création des tables

```
TABLE h1.rapport
id          INTEGER          NOT NULL (PK)
libelle     VARCHAR         NOT NULL UNIQUE
client_only BOOLEAN        NOT NULL DEFAULT false
```

```
TABLE h1.rapport_societe
rapport_id  INTEGER          NOT NULL (PK, FK)
societe_nom VARCHAR         NOT NULL (PK, FK)
role        h1.trole         NOT NULL (PK)
is_default  BOOLEAN          NOT NULL DEFAULT false
```

Capture d'écran : modélisation des tables du module

La présence d'une ligne dans h1.rapport_societe indique qu'une société a accès au rapport pour le rôle défini.

4. Classes métiers et DAO

Le couple classe métier/DAO représente la partie Model d'un framework MVC, en séparant la couche d'accès aux données (DAO) de celle de déclaration de l'identité (classe métier). Cette séparation d'intérêt contribue à la maintenabilité de l'application et à son évolutivité.

Classes métier

```
class Rapport extends ABusiness {
    private $_libelle;
    private $_id;

    public function __construct() {...}
    protected function getIdLibelle() {...}
    protected function equals() {...}
    protected function __get() {...}
    protected function __set() {...}
    protected function __toString() {...}
    protected function __isset() {...}
    protected function getIds() {...}
    protected function getObjectVars() {...}
}
```

```
class RapportSociete {
    private $_rapport; # instance de Rapport
    private $_societe_nom; # (String)
    private $_accessRoles; # ('Réfèrent' => true, 'rôle' => is_default)

    public function __construct($params) {...}
    protected function getRapport() {...}
    protected function getSocieteNom() {...}
    protected function getAccessRoles() {...}
    protected function hasAccess(Role $role) {...}
    protected function isDefault(Role $role) {...}
    protected function equals() {...}
    protected function __toString() { ...}
    protected function __isset() {...}
    public function setAccessRoles($accessRoles) {...}
}
```

Classes DAO

```
class DAORapport {  
    public function get($id) {...}  
    public function getAll() {...}  
    private function _getRapportFromPDOArray(array $result) {...}  
}
```

La méthode `get()` récupère un rapport, `getAll()` les récupère tous, et la dernière fonction transforme les données brutes en un objet `Rapport` grâce à la classe business « `Rapport` ».

```
class DAORapportSociete extends ADAOTransaction {  
    private $_instanciedRapports;  
  
    public function get(Societe $societe, Rapport $rapport) {...}  
    public function getDefaultRapport($societe, Role $role) {...}  
    public function getAllFromRapport(Rapport $rapport) {...}  
    public function getAllFromSociete($societe) {...}  
    public function getAllFromRoleAndSociete($societe, Role $role) {...}  
    private function _getAll($query) {...}  
    private function _getRapportSocieteFromPDOArray(array $rows) {...}  
    private function _getRapportFromRapportId($id) {...}  
    public function add($societe, Rapport $rapport, Role $role) {...}  
    public function update($societe, Rapport $rapport, Role $role) {...}  
    public function delete($societe, Rapport $rapport, Role $role) {...}  
}
```

La propriété `$_instanciedRapports` permet de stocker en mémoire les objets `RapportSociete`. Ainsi, lorsqu'un utilisateur effectue un `get()`, l'on vérifie d'abord que le rapport en question n'est pas déjà stocké. Cela permet de limiter le nombre de requêtes vers la base de données. Les méthodes `get[...]()` retournent l'objet souhaité en fonction des valeurs passées en paramètre. Les méthodes commençant par des `_` sont des méthodes privées qui exécutent une partie intermédiaire du traitement et peuvent être appelées dans plusieurs méthodes publiques. Enfin, on retrouve le reste des opérations **CRUD** classiques dans les dernières méthodes : `add()`, `update()` et `delete()`.

Procédure de tests

Pour tester le bon fonctionnement des méthodes, nous avons ajouté dans le dossier /tests à la racine du projet un fichier de test par **classe métier**, puis nous avons testé différents appels corrects et incorrects, à l'image des **tests unitaires**.

```
1 <?php
2
3 require_once 'offline-sites/db/db_host.php';
4 require_once 'offline-sites/business/role.php';
5 require_once 'offline-sites/business/membre.php';
6 require_once 'offline-sites/business/agence.php';
7 require_once 'offline-sites/business/agence_full.php';
8 require_once 'offline-sites/business/societe.php';
9 require_once 'offline-sites/business/societe_full.php';
10 require_once 'offline-sites/business/groupe.php';
11 require_once 'offline-sites/business/groupe_full.php';
12 require_once 'offline-sites/dao/utils/dao_factory.php';
13
14 $role = Role::operateur();
15
16 print_r("\t== Test ajout ==\n");
17 $rand = rand();
18 $societe = new Societe('133t'.$rand, '0000000000', '0000000000',
19     "133t$rand@mail.com");
20 print_r("\t\t= add('$societe') =\n");
21 try {
22     $dao_societe->add($societe);
23     $societe = $dao_societe->get('133t'.$rand);
24     echo "$societe\n";
25 } catch(DAOSocieteException $daose) {
26     echo "$daose\n";
27 }
28
29
```

Capture d'écran : extrait du fichier tests_dao_societe.php de [Tick&Help](#)

5. Gestion des droits d'accès

Comme expliqué auparavant, la gestion des droits est séparée en deux parties. Une première gestion est faite au niveau de la société, société possédant l'application Tick&Help.

Cette page est présentée sous forme d'un tableau avec différents boutons :

- Activer : est inactif si toutes les cases sélectionnées sont déjà activées,
- Désactiver : est inactif si toutes les cases sélectionnées sont déjà désactivées,
- Définir par défaut : est inactif notamment à partir du moment où, pour une même société et un même rôle, plusieurs rapports sont sélectionnés.

S'il n'y a qu'un seul rapport pour une société et un rôle, ce rapport est automatiquement celui par défaut.

L'autre partie de la gestion se fait au niveau des clients de la société possédant l'application Tick&Help. Cette partie comprend exactement les mêmes éléments (tableau, couleurs, boutons).

La communication client/serveur est effectuée lorsque l'utilisateur valide ses modifications via le bouton « Valider ». L'ensemble des lignes du tableau qui ont été modifiées est alors envoyé en **JSON** à la méthode `manageRights()` du `ControllerStats`, qui convertit le tableau JSON en tableau d'objets **PHP**, et qui, pour chaque ligne de ce tableau, appelle la méthode correspondante si nécessaire (`add()`, `update()` ou `delete()`).

6. Fonctionnalité d'export

En raison d'une forte lenteur lors de la génération de fichiers Excel (environ 1 minute par chargement de fichier), une autre solution a été choisie : celle d'utiliser le format **CSV**. Il s'agit d'un format standard et donc compatible avec la grande majorité des tableurs disponibles sur le marché. De plus, sa légèreté a permis d'augmenter la rapidité de génération des exports. Enfin, **PHP** le supporte nativement via la méthode `fputcsv()`. Ce choix a donc grandement simplifié le code, qui utilisait au départ la librairie PHPExcel. Ce format standardisé permet d'être plus flexible aux différences de **SI** chez les clients de Tick&Help.

7. Contrôleur PHP et gestionnaire de vues HTML

```
3 class ControllerStats extends TControllerAction {
4
5     protected displayRapport<TypeRapport>(Array $params) {...}
6     protected getRapportData<TypeRapport>(Array $params) {...}
7     protected displayRapportRights_Societe() {
8         # Affiche l'écran de gestion des droits au niveau société
9     }
10    protected displayRapportRights_Client() {
11        # Affiche l'écran de gestion des droits au niveau du client
12    }
13    protected manageRights(JSON Array $data) {
14        # Appelle pour chaque ligne du tableau JSON la fonction
15        # _addRights(), _editRights() ou _deleteRight()
16    }
17    private _addRights($rapport, $role, $societe, $is_default) {
18        # Ajoute un droit d'accès au $rapport pour les utilisateurs ayant
19        # $role de la $societe (possibilité d'en faire le rapport par défaut)
20    }
21    private _editRights($rapport, $role, $societe, $is_default) {
22        # Modifie un droit d'accès au $rapport pour les utilisateurs ayant le
23        # $role de la $societe (possibilité d'en faire le rapport par défaut)
24    }
25    private _deleteRight($rapport, $role, $societe, $is_default) {
26        # Suppression d'un droit d'accès au rapport
27    }
28
29 }
```

Capture d'écran : simplification du code du ControllerStats de [Tick&Help](#)

Lorsqu'une page de statistique est appelée, la méthode `displayRapport<TypeRapport>()` est appelée afin de charger la vue correspondante.

Cela se fait via l'appel à `ViewContentStats`, un fichier qui incorpore les scripts CSS et JavaScript dans la vue `template_rapport_<type_rapport>.php`. Cela permet de disposer par exemple de « **combobox** » en lieu et places des balises `select`.

Lors d'un clic sur le bouton « Lancer », la méthode `getRapportData<TypeRapport>()` est appelée via le contrôleur JavaScript. Celle-ci récupère les données qu'elle formate et renvoie au contrôleur JavaScript sous forme d'un tableau.

8. Contrôleur JavaScript

Le contrôleur JavaScript est un fichier JavaScript permettant d'associer des événements dans le **DOM** JavaScript à des actions particulières. C'est ce dernier qui exécute une action au clic sur le bouton « Export CSV » ou encore connaît la période étudiée lors de l'envoi d'informations au contrôleur **PHP**.

Le contrôleur JavaScript construit un JSON valide, puis envoie les données via la fonction `sendDataToPhp()` à la méthode `getRapportData<TypeRapport>()` du contrôleur **PHP**.

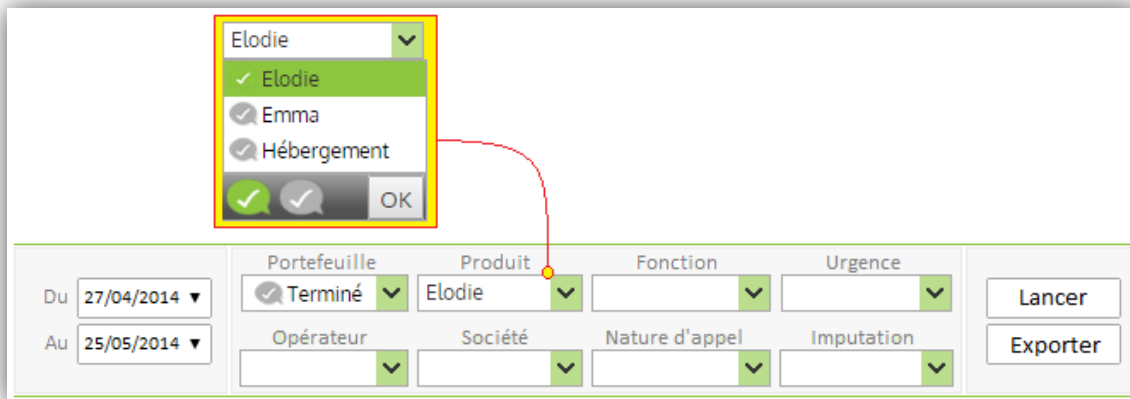
```
244  /**
245   * Envoie des données à la méthode 'getDataRapport{N}' du ControllerStats
246   */
247  function sendDataToPhp(filters, onSuccessCallback) {
248    $.ajax({
249      url:      completeURLParams({action: 'ControllerStats', event: gEvent}),
250      type:     'POST',
251      data:     {'filters': JSON.stringify(filters), 'rapport': gRapport},
252      dataType: 'json'
253    })
254    .done(function(res) {
255      if(!res) {
256        alert('Echec du chargement du rapport. Pas de retour');
257        console.log('Echec ControllerStats');
258        return;
259      }
260      if(res.ERR_INFO) {
261        alert('Echec du chargement du rapport.\n\n' + res.ERR_INFO);
262        console.log('Echec ControllerStats : ' + res.ERR_INFO);
263        return;
264      }
265      if(res.OK_INFO == null || res.OK_INFO == false || res.OK_INFO == 'noData') {
266        console.log('Echec ControllerStats');
267        showErrorMsg();
268        return;
269      }
270      if(onSuccessCallback != null) {
271        onSuccessCallback(res.OK_INFO, res['display']);
272      }
273    })
274    .fail(showErrInfoFromjqXHR);
275  }
276
```

Capture d'écran : extrait du contrôleur JavaScript du module statistiques de [Tick&Help](#)

`sendDataToPhp()` récupère ensuite en retour et affiche le résultat du traitement.

9. Barre de filtres

Les rapports implémentent une barre de filtre en en-tête, qui comprend 3 zones distinctes : à gauche la sélection de la période (non limitée dans le temps), au milieu la liste des filtres disponibles, et à droite le choix entre l’affichage direct ou l’export **CSV**.



Ecran : barre des filtres des sondages dans Tick&Help

La barre prend la largeur de la page de rapport et s’adapte dynamiquement en hauteur à la résolution. Par exemple lorsque la résolution de la page diminue, les filtres passent les uns après les autres en dessous. Et sur une grande résolution, ils sont listés sur une seule ligne. Les parties gauches et droites utilisent la propriété CSS float pour s’aligner aux bords de l’écran lors de changements de résolution.

Au niveau du code JavaScript, chaque balise select est transformée via les événements onReady() du **DOM** par un « **combobox** ».

Dès la conception, nous avons décidé de créer un fichier unique pour la barre de filtre, dont la structure est commune à tous les rapports. Ainsi le fichier contient tous les filtres disponibles pour tous les rapports, mais à la construction de la vue, seuls ceux qui sont pertinents sont conservés (conditions **PHP** déterminant l’ajout ou non de la balise select dans la vue HTML).

10. Mise en forme des histogrammes

Une partie des rapports nécessitait un affichage graphique. Suite à une rapide recherche, c'est le plugin JQPlot qui a été retenu. Ce dernier est gratuit, évolutif et facilement adaptable à l'existant. Le changement de type d'affichage (camembert, lignes, barres) se fait en modifiant une seule ligne dans les options.

Pour fonctionner, le plugin prend en paramètre un tableau de nombres ([12, 25, 8, 37]) et affiche chaque pourcentage de l'histogramme en fonction des valeurs. Il est possible d'associer un code couleur par valeur.

Lors du clic sur une légende de colonne, un nouvel onglet s'ouvre et affiche sur une page de recherche avancée l'ensemble des tickets concernés par la colonne. La méthode utilisée est `advancedSearch()` du `ControllerSearch()`.

La légende est affichée en dehors du plugin, par défaut lors de la création de la vue. Une propriété CSS permet simplement de la désactiver, pour mieux la réactiver lors du retour d'une requête AJAX suite au clic sur le bouton de validation.

11. Mise en forme des tableaux

L'autre partie des rapports (« Temps passés » et « Temps passés opérateur ») utilise un affichage des résultats sous forme de tableaux. Pour enrichir les capacités de base d'une balise table et ainsi proposer une expérience utilisateur plus convaincante, nous avons opté pour l'utilisation de l'objet « **datagrid** » de jQuery EasyUI, ce dernier étant déjà intégré dans l'application.

Un objet JavaScript **datagrid** se construit en déclarant sous forme de propriétés les colonnes (identifiant et intitulé), les données (rows) ou encore des spécificités comme le non support de la sélection multiple (singleSelect: true). Une option footer permet de définir la ou les ligne(s) de résultat à afficher dans le pied de page de chaque colonne. Enfin, il est possible de grouper des résultats par une des valeurs communes (exemple : numéro de ticket) grâce à la propriété groupFormatter.

J'ai utilisé cette propriété pour créer des groupes de tickets dans le tableau **datagrid**. Sur chaque rapport de type tableau, les tâches sont ainsi réparties par ordre croissant de numéro de ticket. Un clic sur le « - » permet de réduire le groupement et ne plus afficher son contenu. Un clic sur l'intitulé vert du numéro de ticket affiche sa page dans un nouvel onglet.

Ticket	Tâche	Opérateur	Portefeuille	Produit	Fonction	Société	Nature d'appel
- T16766 Enquete de satisfaction							
	Analyse - Conception	BLANCHARD Malo	Evo planifiées	Tick&Help		SYXPERIANE	
- T17622 Mail de résolution							
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Mails	XymoX	Evolution
- T18198 STATS3 - Rapport 2 : Temps passés							
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	XymoX	Evolution
	Formation interne don...	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	XymoX	Evolution
	Recette	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	XymoX	Evolution
- T18200 STATS3 - Rapport 4 : Temps opérateur							
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Rapport 4 - Temps ...	XymoX	Evolution
- T18202 STATS4 - Recherche avancée							
	Recette	BLANCHARD Malo	Terminé	Tick&Help	Recherche	XymoX	Evolution
- T18671 Stats temps passé : Intitulé ticket							
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	XymoX	Evolution

Capture d'écran : tableau du rapport « Temps passés » de Tick&Help

J'ai aussi mis en place un formatage des temps, ce qui permet à l'utilisateur de passer instantanément d'un affichage heures:minutes à celui en jours (un jour et demi = 1,5). Ceci a été rendu possible par l'utilisation de la méthode `reload()` qui recharge le contenu du tableau, et de la fonction JavaScript suivante :

```
1 /**
2  * Formate l'affichage du temps à partir de minutes
3  */
4  var _timeFormatter = function(minutesNbr) {
5      return _displayInMinutes ? formatMinutesToLocalTimeString(minutesNbr) :
6          _wkTimeCvr.formatMinutesToWorkDayFloat(minutesNbr);
7  };
```

Capture d'écran : fonction de formatage des temps du module statistiques de [Tick&Help](#)

Cette fonction transforme des minutes en fonction de l'état de l'attribut booléen privé `_displayInMinutes`. Elle utilise pour cela une condition ternaire.

Enfin, j'ai ajouté le calcul des totaux de chaque colonne de type « temps » dans le footer (ANNEXE 4).

12. Recette et livraison

Etape essentielle avant mise en production, la recette a été effectuée de manière inversée : j'avais pour objectif de tester les rapports développés par Benoit et vice-versa. J'ai donc effectué la recette des rapports 3 et 5.

La recette consiste à établir des situations de test complémentaires permettant d'établir tous les cas de figure possibles pour une fonctionnalité. C'est une forme de **tests d'intégration**.

	A	B	C	D	E	F	G	H	I	J	
1	RESULTATS :										
2											
3	avec uniquement l'opérateur "FAVREAU Arnaud" :				=> correspond au graphique						
4	Date début	Date fin	Opérateurs	Tickets affectés	Dont en attente	Tickets modifiés	Dont tickets traités	Tickets en stock			
5	01/01/2015	28/02/2015	FAVREAU Arnaud	78	4	80	70	12			
6											
7	avec uniquement l'opérateur "LEMETAYER Jean" :				=> correspond au graphique						
8	Date début	Date fin	Opérateurs	Tickets affectés	Dont en attente	Tickets modifiés	Dont tickets traités	Tickets en stock			
9	01/01/2015	28/02/2015	LEMETAYER Jean	19	9	20	13	21			
10											
11	avec les opérateurs "Jean LEMETAYER & Arnaud FAVREAU" :				=> correspond au graphique et aux résultats de la recette						
12	Date début	Date fin	Opérateurs	Tickets affectés	Dont en attente	Tickets modifiés	Dont tickets traités	Tickets en stock			
13	01/01/2015	28/02/2015	FAVREAU Arnaud	78	4	80	70	12			
14	01/01/2015	28/02/2015	LEMETAYER Jean	19	9	20	13	21			
15					97	13	100	83	33		
16											
17											
18	+ vérification de l'accès au combo opérateurs selon le profil: OK										
19	+ vérification des caractères spéciaux : NA (que des ids)										
20											
21											
22											
23											
24											
25											
26											
27											

Capture d'écran : recette du rapport 3, cas 1, onglet des résultats

Chaque fichier de recette comprend plusieurs onglets. Les premiers contiennent des jeux de données récupérés depuis la base de données (exemple : le couple rapport/société), et les derniers permettent (voir image ci-dessus) de vérifier que les résultats des tests obtenus dans le fichier correspondent à ceux de l'application en condition réelle. Si les résultats concordent, cela signifie que la programmation du module semble correcte. D'autres cas de tests permettent de corroborer ou non cette hypothèse.

J'ai eu aussi l'occasion d'effectuer la recette d'un autre module, le module « Recherche avancée ».

13. Difficultés rencontrées

La principale difficulté rencontrée fut celle de la maîtrise délicate du **framework** de Tick&Help, dont la logique est un peu plus complexe que celles des **framework** traditionnels comme Symfony ou Laravel. Par manque d'habitude, nous avons commencé avec Benoit la conception technique du module en ne reproduisant pas à la lettre ce qui est déjà en place dans l'existant. Cela a posé des problèmes d'intégration et rendra plus difficile la maintenance ou l'évolution du module.

Je me suis aussi rendu compte de mon manque d'expérience vis-à-vis de la reprise de projets existants. Dans le cadre de mes études, je suis souvent amené à développer une nouvelle application, mais jamais à greffer une nouvelle fonctionnalité sur un code existant et dont il n'est pas possible de modifier le cœur. Cela a entraîné une difficulté d'adaptation supplémentaire.

Le passage de SVN vers **GIT** a apporté son lot de déconvenues, avec par exemple des parties de code supprimées lors de « merges », des espaces transformés par des tabulations suivant la configuration et qui provoquent un conflit dans tout un fichier non modifié, etc.

Enfin, nous nous sommes rendu compte avec Benoit que le plugin JQPlot comporte des limites, notamment au niveau des légendes d'axes des colonnes. Celles-ci sont difficilement personnalisables et il a fallu les masquer pour les remplacer par des légendes manuelles dont les dimensions s'adaptent en résolution et en contenu. Etant donné que les rapports sont chargés en AJAX, il a fallu utiliser les méthodes du **DOM** `resize()` et `window()` pour adapter les dimensions des légendes par rapport au graphique JQPlot déjà généré.

Mission 3 : nouveau module Sondage

Le module à créer répond à une demande d'évolution faite par Mme. CAILLÉ du groupe SYXPERIANE le 13/11/2014, dans le cadre d'une démarche ISO 9001 de son entreprise. Le module n'est pas encore développé, mais la conception est terminée et validée.

1. Besoin client

Le groupe souhaite pouvoir mesurer la satisfaction client lorsqu'un ticket est clôturé. Cela passe par un module permettant de créer, modifier, ou supprimer un sondage, et surtout d'afficher des statistiques par rapport à un sondage en particulier.

2. Spécifications fonctionnelles

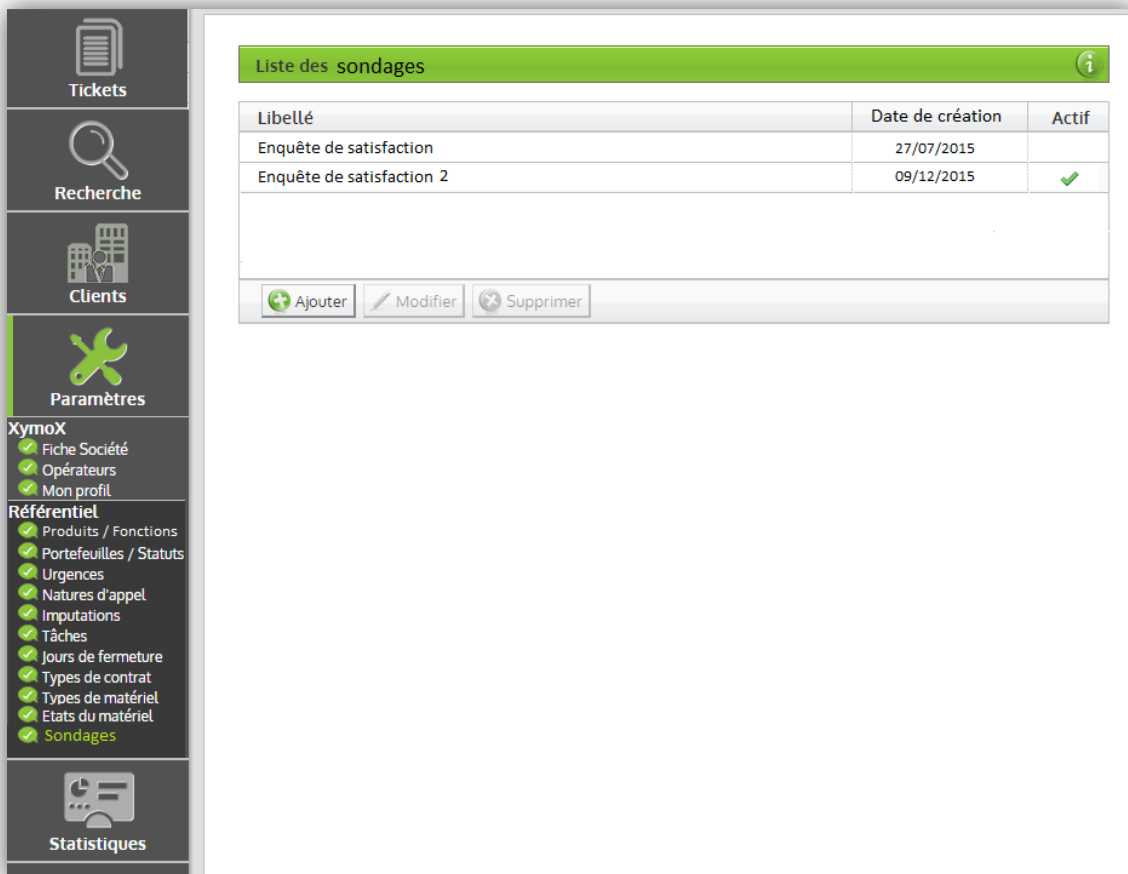
J'ai commencé par réaliser un fichier des spécifications fonctionnelles (**SFD**) à destination de Mme. CAILLÉ. Ce travail m'a pris une semaine, durant laquelle M. Lemétayer m'a fait des retours en vue d'éclaircir certains points. Le logiciel Microsoft Paint m'a été d'une grande utilité pour créer rapidement des croquis et les modifier facilement.

Nous avons avec M. Lemétayer défini les principales règles de gestion, visibles dans le tableau ci-dessous :

Société	Rôle	Création & Modification sondage	Suppression sondage	Réponse au sondage sur le ticket	Lecture résultats sur le ticket
SYXPERIANE	Administrateur	Oui	Oui		Oui
	Opérateur				Oui
	Créateur du ticket			Oui	Oui
Clientes	Référent				
	Utilisateur				
	Créateur du ticket			Oui	Oui
	Collègues du créateur				Oui

Accès aux sondages

La société utilisatrice de Tick&Help disposera d'une nouvelle interface accessible depuis les paramètres pour gérer les sondages, dont voici le visuel à l'ouverture :



Libellé	Date de création	Actif
Enquête de satisfaction	27/07/2015	
Enquête de satisfaction 2	09/12/2015	✓

Capture d'écran : accueil de la page de gestion des sondages

Le principe est simple : il est possible de définir autant de sondages que désiré, mais seul un sondage est actif.

Des règles spécifiques à cette page s'appliqueront :

- Rendre un sondage actif désactive automatiquement tous les autres,
- Seul le créateur du ticket clôturé pourra accéder et répondre au sondage actif,
- Un seul sondage peut être sélectionné à la fois dans la « Liste des sondages »,
- Le sondage sélectionné par défaut sur la page des statistiques est le sondage actif.

Gestion des sondages

Sur l'interface de gestion, la sélection d'un sondage ou du clic sur le bouton « Ajouter », affiche une nouvelle section, visible sur l'image ci-après :

The screenshot displays a web application interface for managing surveys. On the left is a vertical navigation menu with icons and labels for Tickets, Recherche, Clients, Paramètres, XymoX, Référentiel, and Statistiques. The main content area is divided into two sections.

The first section, titled "Liste des sondages", contains a table with the following data:

Libellé	Date de création	Actif
Enquête de satisfaction	27/07/2015	
Enquête de satisfaction 2	09/12/2015	✓

Below the table are three buttons: "Ajouter", "Modifier", and "Supprimer".

The second section, titled "Enquête de satisfaction", shows the configuration for a selected survey. It includes a "Libellé" field with the value "Enquête de satisfaction" and a "Texte dans l'email" field containing the text: "Parce que votre satisfaction est notre priorité, nous vous invitons à répondre au questionnaire de satisfaction suivant :".

Below the text field is a "Questions" section with a list of questions:

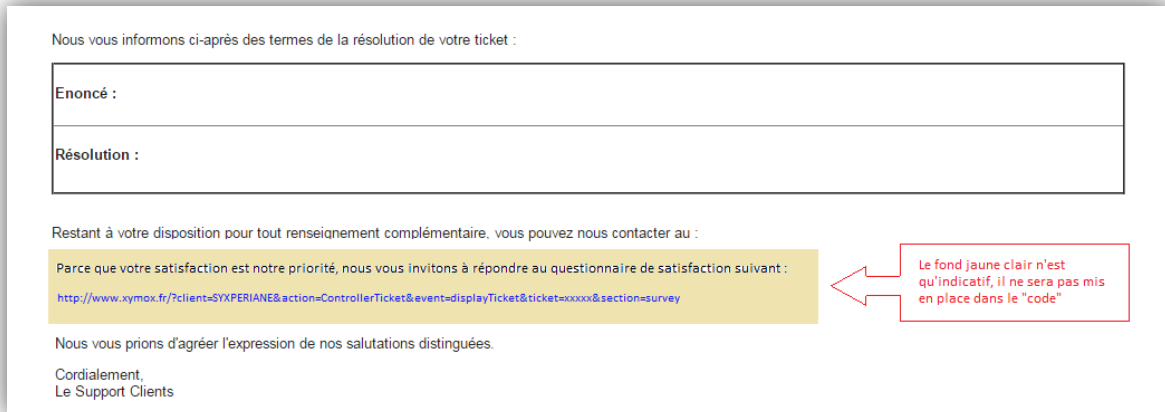
- 1° Accueil téléphonique
- 2° Délai de prise en charge

At the bottom of this section are "Ajouter", "Modifier", and "Supprimer" buttons. At the very bottom right of the interface are "Enregistrer" and "Annuler" buttons.

Capture d'écran : page de gestion des sondages

Email de résolution

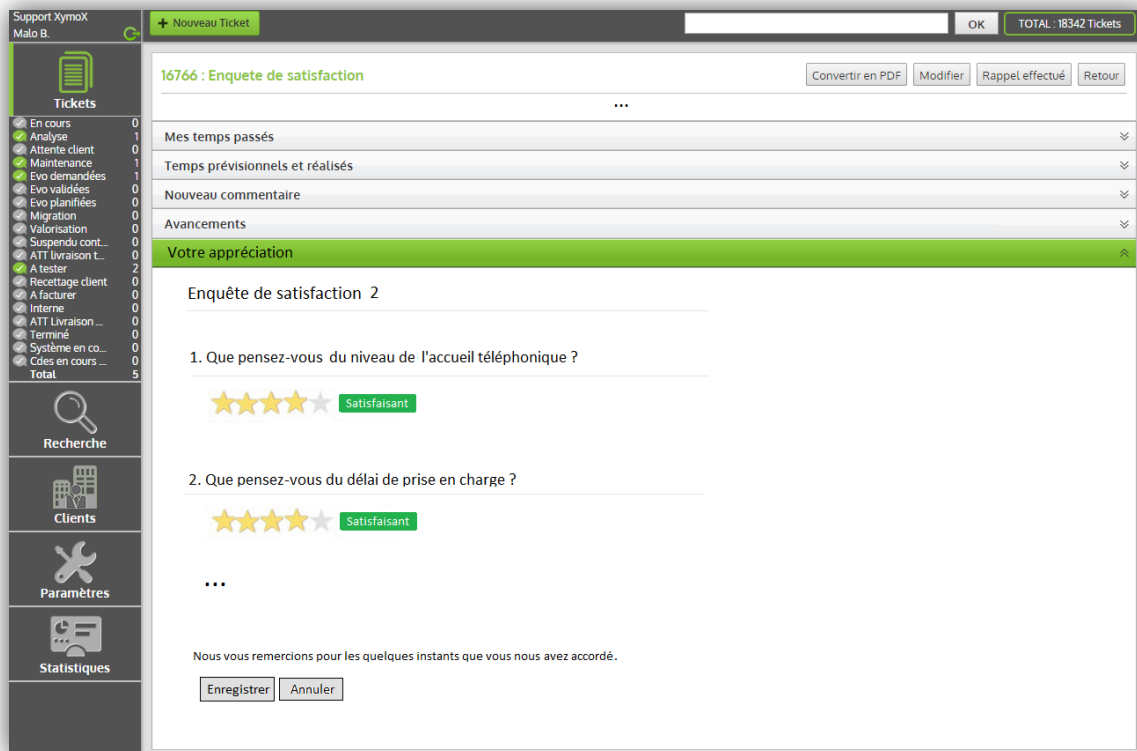
Lorsqu'un ticket est résolu, un mail est envoyé à l'utilisateur auteur de ce ticket pour l'informer de cette résolution et détaillant les opérations effectuées pour résoudre son problème. Le lien n'apparaît pas dans le mail s'il n'y a pas de rapport actif.



Capture d'écran : extrait du squelette d'un mail de résolution

Formulaire de réponse au sondage

Lorsqu'un utilisateur clique sur le lien dans le mail, il est redirigé vers la page du ticket concerné, dans la section « Votre appréciation ». Cette section contiendra le formulaire visible dans l'image ci-dessous s'il ne l'a pas déjà renseigné, ou un visuel de ses réponses dans le cas contraire.



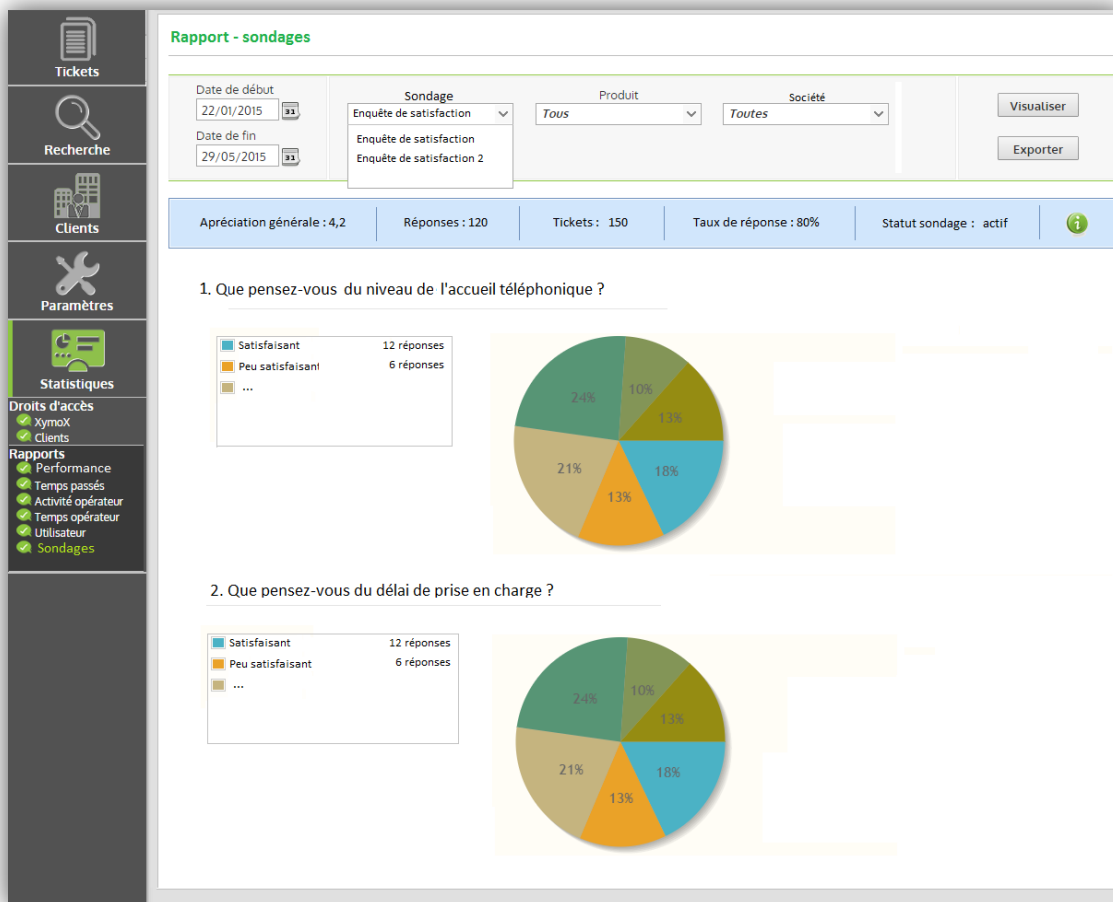
The screenshot shows a web application interface for ticket management. The top bar includes the user name 'Malo B.', a '+ Nouveau Ticket' button, and a 'TOTAL: 18342 Tickets' indicator. The left sidebar contains a 'Tickets' menu with various status filters (e.g., 'En cours', 'Analyse', 'Attente client') and other navigation options like 'Recherche', 'Clients', 'Paramètres', and 'Statistiques'. The main content area displays ticket details for '16766 : Enquete de satisfaction'. The 'Votre appréciation' section is active, showing a survey titled 'Enquête de satisfaction 2' with two questions. Each question has a 5-star rating system, and both are marked as 'Satisfaisant' (Satisfying). The survey concludes with a thank-you message and 'Enregistrer' (Save) and 'Annuler' (Cancel) buttons.

Capture d'écran : page d'un ticket avec la section « Votre appréciation » ouverte

Les réponses aux sondages seront graduées sur une échelle de 1 à 5 et symbolisées par des étoiles.

Lorsque l'utilisateur valide le formulaire, cela enrichit les statistiques du sondage actif. Les statistiques des sondages sont un nouveau type de rapport ajouté au module Statistiques.

Statistiques



Capture d'écran : page du statistiques des sondages après appui sur le bouton « Visualiser »

Comme pour tout rapport, les filtres permettent de sélectionner le sondage à analyser. Une fois les statistiques chargées, une barre bleutée en haut de la section indique des informations agrégées tel que l'appréciation générale (moyenne de toutes les notes à toutes les questions de tous les formulaires de réponses concernés par la recherche). Puis, pour chaque question, un camembert utilisant le plugin [JQPlot](#) affiche en pourcentage le ratio de chaque réponse, avec à sa gauche une légende plus précise.

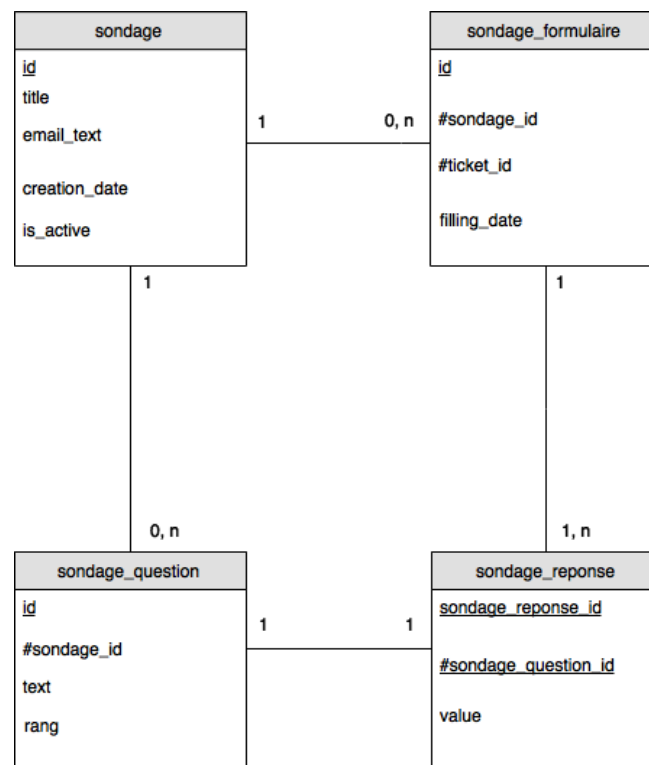
L'export **CSV** permet d'exporter à plat les données brutes comme sur l'exemple ci-après :

A	B	C	D	E	F	G
Date début	Date fin	Sondage	Ticket	Libellé	Produit	Société
22/01/2015	29/05/2015	Enquête de satisf.	16786	Reprise libellé ticket	WEB-Consulting	ECMS
22/01/2015	29/05/2015	Enquête de satisf. 2	16789	Evo. WEB-Consulting	WEB-Consulting	ECMS

3. Spécifications techniques

Une fois mon travail sur la solution fonctionnelle validée par M. Lemétayer, j'ai envoyé le fichier en question à Mme CAILLÉ par l'intermédiaire de Tick&Help. Son retour fut positif et j'ai donc entamé la définition des spécifications techniques (**CDC**).

Modélisation et création des tables



Capture d'écran : Modèle Conceptuel des Données (Merise)

Règles de gestion :

- La table sondage liste tous les sondages créés,
- Un sondage doit au moins contenir un titre pour exister,
- Les questions d'un sondage sont listées dans la table sondage_question,
- Un formulaire de réponse associe ticket avec sondage et est stocké dans sondage_formulaire,
- La colonne value dans sondage_reponse accepte un entier de 1 à 5, correspondant au niveau de satisfaction de l'utilisateur, du plus mauvais (1) au meilleur (5),
- Il y a le même nombre de réponses dans un sondage_formulaire que de questions dans un sondage.

Classes métiers et DAO

Le principe est le même que pour le module Statistiques. Le nombre de tables détermine le nombre de classes.

DAO :

1. DAOSondage : **CRUD** d'un sondage et récupération de tous les sondages,
2. DAOQuestion : Liste des questions liées à un sondage et opérations CRUD,
3. DAOFormulaire : Opérations sur un formulaire de réponse à un sondage,
4. DAOReponse : Ajoute les réponses d'un formulaire dans la base de données, retourne la liste de toutes les réponses d'un formulaire avec pour chacune la valeur associée.

Classes métier :

- Sondage,
- Question,
- Formulaire,
- Reponse.

Contrôleur PHP et gestionnaire de vues HTML

Le contrôleur, véritable chef d'orchestre, fera communiquer les modèles et le gestionnaire de vues HTML à l'image des autres modules déjà en place. Ce gestionnaire de vue sera appelé « ViewContentSondage » et se chargera d'afficher les différentes pages HTML en y intégrant les données précédemment récupérées.

4. Difficulté rencontrée

La principale difficulté de cette mission a été de me mettre dans la tête d'un client qui ne connaît pas les termes informatiques techniques. C'est pourquoi j'ai pris du temps pour faire et refaire le document des spécifications fonctionnelles (**SFD**). La phase de conception technique a ensuite été bien plus aisée.

Mission 4 : missions diverses

J'ai été amené à réaliser différentes petites missions annexes aux principales. Entre autres, j'ai été amené à :

- Former un membre de l'équipe de communication aux bases du CSS,
- Effectuer diverses améliorations graphiques de Tick&Help,
- Modifier le pied de page de l'application,
- Aider l'équipe lors du remplacement des disques durs de nos configurations de travail par des disques SSD,
- Effectuer la recette du module « Recherche avancée »,
- Modifier plusieurs icônes, notamment celle de création d'un nouveau ticket,
- Modulariser certaines portions de code CSS.

Conclusion

Parcours

Cette année en contrat de professionnalisation m'a permis de consolider mes compétences en développement web au travers de l'utilisation récurrente de **PHP** orienté objet, JavaScript/jQuery et HTML/CSS.

J'ai également pu découvrir en situation réelle la base de données **PostgreSQL**, qui m'a d'ailleurs permis de mettre en pratique mes cours d'Oracle au travers de scripts **PGSQL**.

Pour la première fois, j'ai eu l'opportunité de travailler sur un projet existant et d'utiliser des plugins JavaScript tels que jQplot ou encore jQuery EasyUI. Leur utilisation régulière m'a permis de mieux appréhender la lecture des documentations techniques en ligne.

Les différents modules à créer ou rénover ont été pour moi l'occasion de prendre beaucoup d'expérience en conception, domaine qui me plaît tout particulièrement par son côté créatif, mais néanmoins rigoureux.

Les différents enseignements de l'EPSI m'ont aidé à devenir plus autonome et à savoir mieux m'organiser, notamment en planifiant mes tâches du jour sur un bloc-notes.

Au-delà des compétences, j'ai développé un sens de l'écoute et amélioré ma force de proposition, lié aux nombreuses discussions techniques et réflexions sur la conception de fonctionnalités.

Axes d'amélioration

Cependant je ne suis pas encore tout à fait autonome, et mes questions n'ont pas toujours très explicites. Ma compréhension n'est pas toujours parfaite et je manque parfois de confiance en moi pour oser poser des questions.

Objectifs professionnels

Je souhaite poursuivre ma formation à l'EPSI en cycle « Ingénieur » en alternance. Mon souhait est me perfectionner dans les technologies JavaScript (NodeJS, AngularJS, ReactJS, etc.) et de me diriger vers le monde des start-up.

A moyen terme, je suis motivé par l'idée de créer ma propre entreprise, mais pas forcément dans le développement informatique.

Glossaire

Les définitions proviennent principalement de l'encyclopédie en ligne [Wikipédia](#).

CDC (Cahier Des Charges)

Documentation technique destinée à guider le développeur lors de la programmation.

Classe métier

Synonyme d'entité, c'est la définition d'un objet que l'on pourra ensuite instancier dans l'application.

Combobox

Objet du plugin [EasyUI](#) qui enrichit la balise select en y intégrant entre autre support des sélections multiples. Si toutes les options disponibles sont sélectionnées, le libellé est remplacé par « Tous les ... » ou « Toutes les... ».

CRUD (Cread Read Update Delete)

Se sont les 4 opérations basiques sur des données persistantes : création, lecture, modification et suppression.

CSV (Comma Separated Value)

Format de fichier léger et standardisé permettant de stocker des informations.

DAO (Data Access Object)

Couche d'abstraction de la base de données permettant d'effectuer les requêtes par rapport à une entité et de séparer la partie métier de la partie de traitement des données.

Datagrid

Objet JavaScript présent dans [EasyUI](#) qui enrichit les possibilités de base d'une balie table en lui apportant par exemple le support de groupement ou de la sélection multi lignes.

Dépôt

Le dépôt désigne l'endroit où les modifications apportées aux fichiers sont stockées, dans le cadre de l'utilisation du logiciel GIT.

DOM (Document Object Model)

Modèle décrivant les éléments d'une page HTML.

ERP/PGI (Progiciel de Gestion Intégré)

Ensemble de briques logiciels formant un ensemble cohérent de gestion et de suivis d'activité d'une entreprise, à la base de son système d'information (SI).

Framework

Ensemble cohérent de composants structurant les composants et les interactions des modules d'une application. Cela standardise toutes les étapes de conception d'une application, et favorise ainsi le travail en équipe.

GIT

Logiciel de gestion de versions permettant le travail en équipe dans un projet informatique. Ce dernier est initialisé sous forme d'un **dépôt**. Chaque développeur possède une version du **dépôt** en local sur son ordinateur. La version « à jour » du projet est hébergée sur un dépôt distant. Lorsqu'une personne a terminé de modifier en local son code, elle « commit » puis « push » ses fichiers sur le dépôt distant. Les autres utilisateurs devront alors « pull » les modifications afin d'être à jour.

IDE (Integrated Development Environment)

Ensemble de briques logicielles formant un environnement de développement intégré. On peut citer l'exemple de PHPStorm ou encore celui d'Eclipse.

Intégration continue

Ensemble de pratiques consistant à vérifier à chaque modification du code source que le résultat ne produit pas de régression dans l'application développée.

JSON (JavaScript Object Notation)

Format de fichier standard conçu pour être facilement lisible et proposer une alternative au XML.

MVC (Model View Controller)

Principe de programmation très populaire en PHP depuis l'apparition de la programmation orientée objet, et qui consiste à séparer en trois couches le code.

PGSQL

Language procédural utilisé par PostgreSQL et supportant également le langage PLSQL d'Oracle. Ce dernier apporte le support d'opérations plus complexes que ce que le SQL permet d'origine.

PHP (Hypertext Preprocessor)

PHP est un langage de programmation libre créé en 1994 par Rasmus Lerdorf et utilisé pour produire des pages Web dynamiques. PHP est un langage orienté objet qui supporte le procédural et dispose d'un typage faible.

PostgreSQL

Système de Gestion de base de données Relationnelles open-source développé à l'université de Californie. Il supporte une grande partie du standard SQL tout en offrant des fonctionnalités modernes comme les **triggers**, les vues modifiables, le support du **PGSQL**, etc.

Rôle (Tick&Help)

L'application distingue via des rôles le niveau d'accès des utilisateurs : Administrateur, Opérateur, Référent et Utilisateur.

SaaS (Service as a Software)

Modèle d'exploitation commerciale des logiciels dans lequel ils sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur.

SFD (Spécifications Fonctionnelles Détaillées)

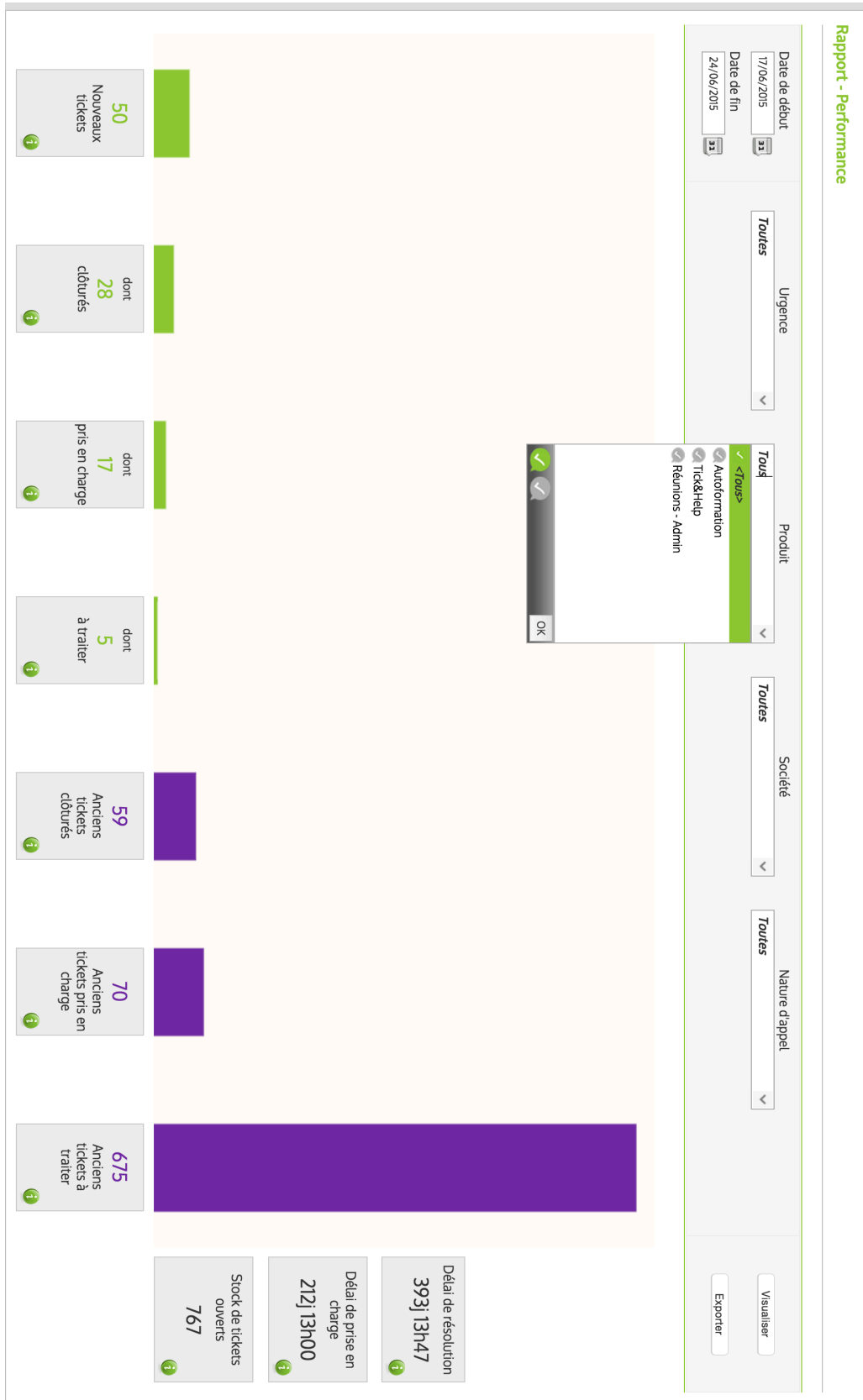
Fichier décrivant précisément les fonctions d'un logiciel en vue de sa réalisation.

Test unitaire

Procédure atomique visant à vérifier que le comportement d'une méthode est conforme aux attentes suivant les situations d'appel.

Annexes

ANNEXE 1 : rapport « Performance »



ANNEXE 2 : droits d'accès au niveau société

Gestion des droits

Accès aux rapports ?

Niveau Société

	Administrateur	Opérateur
Rapport 1 : Performance	✔ Par défaut	
Rapport 2 : Temps passés		
Rapport 3 : Activité opérateur		✔ Par défaut
Rapport 4 : Temps opérateur		
Rapport 5 : Utilisateur		

Annuler Valider

Barre d'outils: + Nouveau, Activer, Désactiver, Définir par défaut, OK, TOTAL : 16854 Tickets

Menu déroulant: Sélection d'une ou plusieurs cases inactives (Activer, Désactiver, Définir par défaut), Sélection d'une ou plusieurs cases actives (Activer, Désactiver, Définir par défaut)

ANNEXE 3 : droits d'accès au niveau du rôle

+ Nouveau Ticket

Activer

Désactiver

Définir par défaut

OK

TOTAL : 16854 Tickets

Gestion des droits

Accès aux rapports ?

	Rapport n°5		Rapport n°6	
	Réfèrent	Utilisateur	Réfèrent	Utilisateur
Société 1	✔ Par défaut	Sélectionner toute la colonne	✔ Par défaut	
Société 2	✔ Par défaut			
Société 3		✔ Par défaut		
Société 4				✔ Par défaut
Société 5				

Annuler

Valider

Sélection d'une ou plusieurs cases inactives

Activer

Désactiver

Définir par défaut

Sélection d'une ou plusieurs cases actives

Activer

Désactiver

Définir par défaut

Sélection d'une ou plusieurs cases inactives et actives

Activer

Désactiver

Définir par défaut

ANNEXE 4 : Rapport « Temps passés »

Rapport - Temps passés

Date de début: 17/04/2015
 Date de fin: 24/06/2015
 Opérateur: BLANCHARD Malo
 Portefeuille: Tous
 Nature d'appel: Toutes
 Société: Société
 Produit: Tous
 Imputation: Toutes
 Fonction: Fonction

Ticket	Tache	Opérateur	Portefeuille	Produit	Fonction	Société	Nature d'appel	Imputation	Temps						
									passé total	passé période	prévisionnel	écart	facturé		
Enquete de satisfaction															
- T16166	Analyse - Conception	BLANCHARD Malo	Evo planifiées	Tick&Help		SYXPERIANE			69h50	69h50	12h00	57h50	00h00		
- T17622	Mail de résolution								05h30	05h30	01h00	04h30	00h00		
- T18198	STAT33 - Rapport 2 : Temps passés														
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Mails	Xymox	Evolution	Imputable XYMOX	12h45	08h30	01h00	04h30	-		
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	Xymox	Evolution	Imputable XYMOX	07h45	03h30	00h00	-	-		
	Formation Interne don...	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	Xymox	Evolution	Imputable XYMOX	01h30	01h30	00h00	-	-		
	Recette	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	Xymox	Evolution	Imputable XYMOX	03h30	03h30	00h00	-	-		
- T18200	STAT33 - Rapport 4 : Temps opérateur														
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Rapport 4 - Temps ...	Xymox	Evolution	Imputable XYMOX	11h00	06h30	00h00	-	00h00		
- T18202	STAT34 - Recherche avancée														
	Recette	BLANCHARD Malo	Terminé	Tick&Help	Recherche	Xymox	Evolution	Imputable XYMOX	39h00	21h00	00h00	-	00h00		
- T18671	Stats temps passé - Intitulé ticket														
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Rapport 2 - Temps ...	Xymox	Evolution	Imputable XYMOX	00h30	00h30	00h00	-	00h00		
- T18672	Mes temps passés - Visu du libellé ticket														
	Développement	BLANCHARD Malo	Terminé	Tick&Help	Tickets	Xymox	Evolution	Imputable XYMOX	14h00	14h00	00h00	-	-		
	Support - Assistance	BLANCHARD Malo	Terminé	Tick&Help	Tickets	Xymox	Evolution	Imputable XYMOX	00h10	00h10	00h00	-	-		
Total									152h45	126h00	19h00	62h20	77h00		

DÉVELOPPEUR WEB - ALTERNANCE

MALO BLANCHARD, BAC+3



COMPÉTENCES

- ▶ NodeJS
- ▶ PHP, Laravel
- ▶ Bootstrap, Jade
- ▶ Vagrant, Ansible
- ▶ OSX, Linux
- ▶ AngularJS
- ▶ Trello
- ▶ SASS, LESS
- ▶ Photoshop
- ▶ PostgreSQL
- ▶ RabbitMQ
- ▶ Git
- ▶ Grunt, compass
- ▶ Jenkins
- ▶ bash

INFORMATIONS

- 📅 24 ans
- ☎ 06 63 05 68 92
- 📍 5 rue de Miséricorde
44000 NANTES
- 🚗 Permis B
- ✉ malo.blanchard
- 🌐 maloblanchard.com
- ✉ pro@maloblanchard.com

EXPÉRIENCES PROFESSIONNELLES

<p>SEPT. 2014</p>	<p>XYMOX SYSTEM</p>	<p>Concepteur développeur en alternance 1 an</p> <p>Co-réalisation module statistiques + maintenance et évolutions du nouveau produit web de la société</p> <ul style="list-style-type: none"> ▪ conception et développement (rédaction CdC, framework PHP, PostgreSQL, jQuery, SASS) ▪ maintenance du produit (gestion des retours clients) ▪ mise en place d'évolutions (conception, réalisation, recette, livraison)
<p>JAN. 2014</p>	<p>Bringr</p>	<p>Développeur web stagiaire 2 mois</p> <p>Co-réalisation d'un module NodeJS d'envoi d'alertes en temps réel pour les clients de la start-up.</p> <ul style="list-style-type: none"> ▪ conception (API REST, MCD, diagrammes) ▪ développement backend (NodeJS, nodechildren, nodeunit, RabbitMQ, MySQL) ▪ développement front-end (Bootstrap, AngularJS, routerUI, restangular)
<p>MAI 2013</p>	<p>HistoireDeFormes <small>Habilite les femmes en formes</small></p>	<p>Développeur web stagiaire 7 sem.</p> <ul style="list-style-type: none"> ▪ Migration du site e-commerce de Wordpress vers Prestashop ▪ Nouvelle charte graphique + nouveau logo

FORMATION

<p>2016</p> <p>2014 ● 2015</p> <p>2012 ● 2014</p> <p>2010 ● 2012</p> <p>2009 ● 2010</p> <p>JUIN ● 2009</p>		<p>Ingénieur informatique - 4^{ème} année <i>futur</i></p> <p>Bachelor informatique - 3^{ème} année <i>en cours</i></p> <p>BTS Services Informatiques aux Organisations (SIO) <i>obtenu</i></p> <p>Licence Économie et Gestion - 1^{ère} et 2^{ème} années</p> <p>BTS MUC (Management des Unités Commerciales) - 1^{ère} année</p> <p>Baccalauréat série Sciences et Techniques de Gestion (STG) - Option mercatique</p>
--	--	---

LANGUES

- ANGLAIS Lu, écrit et parlé
- ESPAGNOL Lu et parlé

CENTRES D'INTÉRÊT

- SPORTS tennis, vélo, badminton, football
- HISTOIRE sociétés médiévales, aviation militaire 20^{ème} siècle