

Etudiant
Bilal BENLARBI

Tuteur
Pierre CLAIRET

Maître de stage
Eddy MAINPAIN

IUT d'Orléans - Département informatique

Licence professionnelle réseaux et télécommunications
spécialité développement web et mobile - 2013/2014

CMRP Montargis (45)



RAPPORT DE STAGE

Développement web et mobile



REMERCIEMENTS

Je tiens à remercier Manuel Ribeiro - dirigeant de CMRP - pour la confiance qu'il m'a accordé en m'intégrant à son équipe. Je remercie aussi Eddy Mainpain, mon maître de stage, pour son accueil, son aide et sa collaboration.

Plus généralement, je suis reconnaissant envers l'entreprise CMRP.

Par ailleurs, je remercie l'IUT d'Orléans pour la qualité des enseignements proposés et la formation dispensée.

SOMMAIRE

Sommaire

REMERCIEMENTS	2
SOMMAIRE	3
I. INTRODUCTION	5
II. PRESENTATION DE L'ENTREPRISE	6
II.1 Historique de l'entreprise	6
II.2 Marques de l'entreprise	6
II.3 Cibles de l'entreprise	7
II.4 Equipe	7
II.4.1 Pôle CMRP	7
II.4.2 Pôle Kopy Minute	8
II.5 Environnement de travail	8
III. DEVELOPPEMENT MOBILE : APPLICATION MOBILE DE L'ESPACE CLIENT SELFONE	9
III.1 Description du besoin	9
III.2 Analyse du besoin	9
III.2.1 Système de connexion	10
III.2.2 Tableau de bord	10
III.3 API	14
III.3.1 Organisation	14
III.3.2 Travailler avec les bases de données	14
III.3.3 Analyse	14
III.3.4 Développement	15
III.4 Application native (Android)	17
III.4.1 Configuration de l'environnement de développement	17
III.4.2 Développement	17
III.5 Application web	26
III.5.1 Remarque	26
III.5.2 Analyse	26
III.5.3 Développement	27
IV. DEVELOPPEMENT WEB	29
IV.1 Configuration de l'environnement de développement	29
IV.1 Refontes de sites web	30

IV.1.1 DOMAINE PINSON	30
IV.1.2 AVOCATS VIGNET	32
IV.2 Création de sites web	34
IV.2.1 CMRPGRAPHICDESIGN	34
IV.2.2 BLOG D'ASSISTANCE TECHNIQUE SELFSEVEUR	36
IV.3 Campagne d'emailing	39
IV.3.1 Description du besoin	39
IV.3.2 Analyse du besoin	40
IV.3.3 Réalisation.....	40
IV.3.4 Déploiement	40
IV.4 Campagne de référencement naturel de Selfserveur.com	41
IV.4.1 Description du besoin	41
IV.4.2 Organisation.....	41
IV.4.3 Outils utilisés	41
IV.4.4 Réalisation.....	41
IV.4.5 Résultat	41
V. CONCLUSION	42
V.1 Conclusion technique	42
V.2 Conclusion générale	43

I. INTRODUCTION

Dans le cadre de la licence professionnelle Réseaux et Télécommunications spécialité Développement Web et Mobile de l'IUT d'Orléans, j'ai effectué un stage au sein de l'entreprise CMRP. Cette entreprise est composée de plusieurs pôles que je détaillerai par la suite, à savoir : imprimerie, accès Internet, téléphonie par IP, hébergement de serveurs, développement web.

Un des principaux objectifs du stage est d'adapter l'espace client Selfone au format mobile. Les autres objectifs sont le développement de sites web, dont des refontes, mais aussi une campagne d'emailing et une campagne de référencement naturel.

Le stage, d'une durée de 16 semaines, s'est déroulé du lundi 17 février au vendredi 6 juin 2014.

Je commencerai par présenter l'entreprise CMRP : son histoire, ses marques, ses cibles et son équipe, avant de décrire mon environnement de travail. Ensuite, je traiterai des projets. Cette partie sera scindée en deux, avec d'une part le développement mobile et d'autre part le développement web. Enfin, je conclurai avec un bilan personnel et professionnel de cette expérience.

II. PRESENTATION DE L'ENTREPRISE

II.1 Historique de l'entreprise

CMRP est une entreprise créée en 1999 par M. Ribeiro et située à Montargis (45).

La vocation première de cette entreprise est la création, le développement et l'hébergement de sites web professionnels.

Très vite, M. Ribeiro décide de s'orienter vers l'hébergement de sites grâce à ses propres serveurs, basés en Floride (les infrastructures françaises manquant de fiabilité à cette époque). En 2002, l'hébergement web professionnel constitue une marque : Selfserveur.

En 2004, les deux services sont fusionnés sous la SARL CMRP.

En 2008, un autre service s'ajoute à CMRP. Il s'agit de Selfone, qui propose des services de téléphonie par IP avec notamment la vente de forfaits illimités et de numéros SDA (numéros téléphoniques publics).

Des services viendront s'ajouter aux différents pôles comme le cloud, l'hébergement de solutions professionnelles.

II.2 Marques de l'entreprise

CMRP vend de nombreux services aux professionnels, sous le nom de plusieurs marques :

- CMRP : accès Internet haut et très haut débit (ADSL, SDSL, fibre optique) et création de sites Internet (ce domaine n'est que peu actif, il s'agit d'une volonté de M. Ribeiro).
- Selfserveur : hébergement de solutions professionnelles (cloud, serveurs dédiés).
- Selfone : téléphonie IP (forfaits illimités).
- Kopy Minute : imprimerie (Publication Assistée par Ordinateur, plastification, faire-parts, tampons, envois, fournitures, reliures spirales et thermocollées, vente d'objet publicitaire personnalisé, impression 3D).

II.3 Cibles de l'entreprise

Les cibles sont différentes suivant chaque service vendu :

- Pour Selfserveur et CMRP (accès Internet), les cibles principales sont les PME, PMI et grands comptes, sur toute la France, de tous secteurs. Le cœur de cible réside dans l'administrateur réseau.
- Pour Selfone, les cibles principales sont les PME/PMI, sur toute la France, de tous secteurs (de 3 à 50 salariés). Le cœur de cible ici est le responsable des achats, l'administrateur réseau.
- La partie création de site Internet de CMRP, quant-à elle, est plus locale (dans le Loiret) et cible tout type d'entreprise.
- Kopy Minute vend ses services aussi bien en commerce B to C (« Business to Consumer » : des entreprises aux particuliers) qu'en B to B (« Business to Business » : la clientèle visée est alors les entreprises).

II.4 Equipe

L'équipe de CMRP est composée de 9 personnes.

II.4.1 Pôle CMRP

- **Manuel Ribeiro** : Il est le dirigeant de l'entreprise CMRP. Il prend la plupart des décisions et gère les problèmes majeurs. Il fait fonctionner et vivre l'entreprise dans son ensemble.
- **Élodie Foucault** : Elle est l'assistante de M. Ribeiro, autant d'un point de vue administratif que commercial. En effet, elle gère un portefeuille de clients pour le service de numéros SDA et toute la partie Selfserveur. Elle est également standardiste puisqu'elle répond aux appels des clients et prospects.
- **Eddy Mainpain** : Il gère le support technique et le développement web. Il répond à toutes les questions et problèmes des clients de la partie Selfserveur et fait donc en sorte de régler chaque problème technique de chaque client dans un temps réduit. Il développe aussi des sites web et des scripts pour la téléphonie.
- **Éric Audoux** : Il est le commercial pour la partie Selfone. En effet, il gère un portefeuille de clients PME/PMI sur toute la France. Il vend principalement des services de téléphonie IP, notamment des forfaits illimités.
- **Mélanie Béchu** : Elle est chargée marketing de CMRP et s'occupe de toutes les marques de CMRP sur les réseaux sociaux (Facebook, Twitter, LinkedIn, Viadeo, Google+). Elle est aussi en charge de la communication de l'entreprise et réalise des plaquettes et des fiches de présentation.

II.4.2 Pôle Kopy Minute

- **Nathalie Ribeiro** : Elle est la directrice de l'agence Kopy minute, elle coordonne et gère l'entreprise et ses employés pour son bon fonctionnement.
- **Katerine Marieux-Chevrot** : Elle est la graphiste de l'agence et travaille l'image des clients sur divers supports. C'est donc elle qui gère la partie PAO.
- **Françoise Da Costa** : Elle accueille les clients et gère aussi d'autres tâches concernant l'impression.
- **Françoise** : Elle est la comptable et gère tous les comptes des entreprises. Elle ne vient qu'une fois par semaine.

II.5 Environnement de travail

Pendant le stage, j'ai travaillé sur mon ordinateur portable personnel sous Windows 7. Il a donc fallu se munir des logiciels adéquats. Les logiciels de développement web et les logiciels de développement mobile sont différents. Je développerai donc cette partie dans chacune des sections concernées.

J'ai partagé le bureau d'Eric Audoux (commercial) et Mélanie Béchu (chargée de communication), au sein du pôle CMRP.

III. DEVELOPPEMENT MOBILE : APPLICATION MOBILE DE L'ESPACE CLIENT SELFONE

III.1 Description du besoin

L'espace client existe déjà sur le site web de Selfone, mais il n'est pas responsive (adaptable sur mobile) et toutes les fonctionnalités disponibles ne sont pas nécessaires pour des clients en déplacement. Le but de l'application est de permettre aux clients en déplacement qui, par conséquent, ne disposent pas d'ordinateur, d'accéder à une partie seulement des fonctionnalités, comme le rechargement du crédit de communication et la gestion des tickets d'incidents.

L'interface de l'application doit être conçue pour une prise en main simple et rapide.

III.2 Analyse du besoin

Une fois l'objectif défini, j'ai pu commencer à réfléchir à l'élaboration de l'application. Pour m'aider dans ma réflexion, je me suis appuyé sur l'espace client déjà existant sur le site web de Selfone.

Tout d'abord, il parut évident de mettre en place un système de connexion à l'espace client.

III.2.1 Système de connexion

Pour se connecter, l'utilisateur doit saisir son email et son mot de passe.

Dans le cas où il a oublié son mot de passe, un module est prévu à cet effet. Il lui suffit de saisir son adresse mail et un mail comportant son nouveau mot de passe (généré automatiquement) lui sera envoyé.

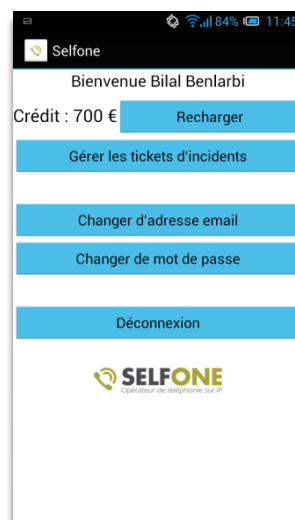
A noter qu'il n'y a pas de module de création de compte, car c'est l'équipe technique qui s'en charge.



III.2.2 Tableau de bord

Dès lors que l'utilisateur est connecté, il est automatiquement redirigé vers le tableau de bord. C'est là qu'il peut gérer toutes ses fonctionnalités.

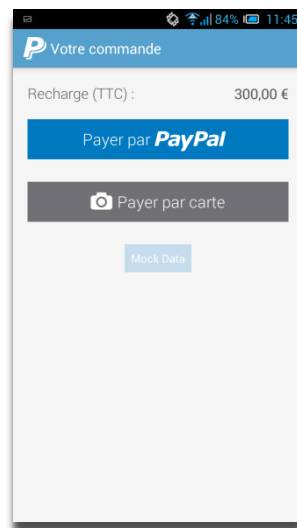
Lorsqu'il a terminé, il peut se déconnecter.



III.2.2.1 Gestion du crédit de communication

Pour recharger son crédit de communication, l'utilisateur doit choisir un montant de rechargement.

Le règlement du rechargement s'effectue par l'intermédiaire de l'API (interface de programmation) PayPal.



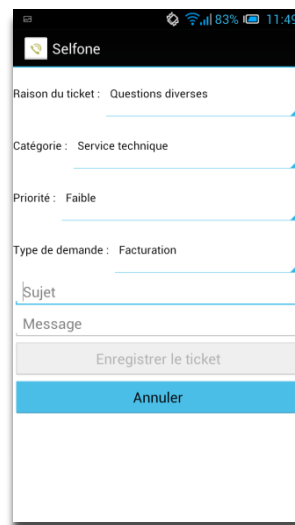
III.2.2.2 Gestion des tickets d'incident

Les tickets d'incidents sont des messages qui transitent entre le client et l'équipe Selfone. Ils ont plusieurs caractéristiques :

- raison (questions diverses, service dégradé, service interrompu)
- catégorie (service technique, service commercial, service comptabilité)
- priorité (faible, moyenne, élevée)
- type de demande (facturation, demande commerciale, modification de compte, SDA/DID et routage, coupure d'un appel entrant, coupure d'un appel sortant, problème de connexion à un compte SIP, problème de serveur, autre).
- sujet
- message

Pour créer un ticket d'incident, il faut remplir tous les champs ci-dessus. Ensuite, la communication entre le client et les différents services de Selfone s'établit via les messages. Ainsi, un seul ticket d'incident peut contenir plusieurs messages.

L'émission de ticket d'incident peut se faire dans les deux sens : du client vers l'équipe Selfone et inversement (par exemple, l'équipe technique informe les usagers lorsqu'il y a intervention).

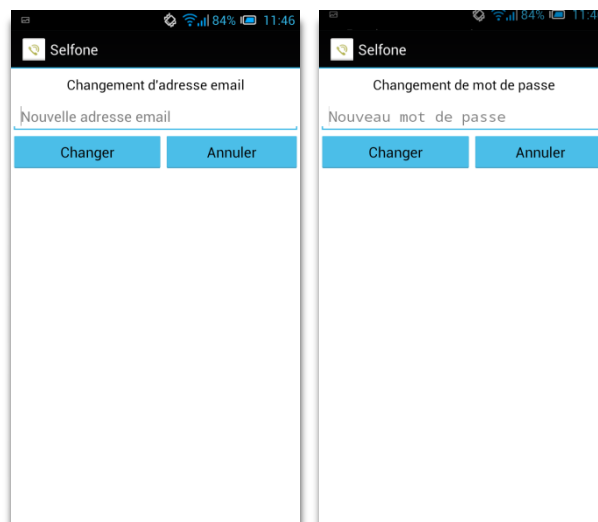


The screenshot shows the 'Selfone' app interface for creating a ticket. The form includes the following fields and options:

- Raison du ticket :** Questions diverses
- Catégorie :** Service technique
- Priorité :** Faible
- Type de demande :** Facturation
- Sujet :** (empty text field)
- Message :** (empty text area)
- Buttons:** 'Enregistrer le ticket' (grey) and 'Annuler' (blue).

III.2.2.3 Gestion du compte

Une section « gestion du compte » a été mise en place. Elle permet à l'utilisateur de modifier ses identifiants de connexion (adresse mail et mot de passe).



The two screenshots show the 'Selfone' app interface for account management. The left screenshot is for 'Changement d'adresse email' and the right is for 'Changement de mot de passe'.

- Changement d'adresse email:**
 - Field: Nouvelle adresse email
 - Buttons: 'Changer' (blue) and 'Annuler' (blue).
- Changement de mot de passe:**
 - Field: Nouveau mot de passe
 - Buttons: 'Changer' (blue) and 'Annuler' (blue).

III.2.3 Cycle d'utilisation

1. Connexion à l'espace client (connexion / mot de passe oublié ; pas de module d'inscription).

2. Accès au tableau de bord :

- gestion du crédit communication (consultation, rechargement)
- gestion des tickets d'incidents (consultation, édition, création)
- gestion du compte (modification des identifiants de connexion : mail et mot de passe)

(3. Déconnexion)

Evidemment pour que l'application fonctionne une connexion Internet (Wifi ou réseau mobile) est requise.

III.3 API

Pour fonctionner, l'application doit communiquer avec la base de données qui stocke les informations à propos du compte client, du crédit de communication et des tickets d'incidents. Cette communication s'établit par ce que l'on appelle l'API (Application Programming Interface, qui se traduit par interface de programmation).

III.3.1 Organisation

La logique de la base de données m'a été expliquée par mon maître de stage. Il m'a communiqué le nom des tables et des champs qui y interviennent. Dès lors, j'ai reproduit la base de données sur mon serveur personnel afin de ne pas perturber le serveur Selfone en production. Pour cela, j'ai utilisé l'interface phpMyAdmin. J'ai rempli les tables avec des données cohérentes et semblables à celles de la réalité pour simuler le fonctionnement normal.

III.3.2 Travailler avec les bases de données

Lors du développement j'ai dû réaliser des requêtes vers la base de données. Pour gagner en efficacité, avant d'écrire les requêtes dans les programmes PHP, je me suis assuré de leur conformité en les testant en direct sur l'interface web de phpMyAdmin. De cette façon, je segmente les différentes parties du programme pour mieux identifier les sources d'erreurs et mieux m'y retrouver dans mon travail.

III.3.3 Analyse

Le recours à la base de données intervient dans les cas suivants :

- Connexion à l'espace personnel : authentification de l'utilisateur pour vérifier que l'email et le mot de passe saisis concordent bien avec ceux attendus dans la base de données.
- Consultation / édition du crédit de communication disponible.
- Consultation / édition / création des tickets d'incidents et des messages.
- Modification des identifiants de connexion (adresse email / mot de passe).

III.3.4 Développement

Pour mettre en place ce système, j'ai utilisé des scripts PHP.

Dans un premier temps, il faut établir la connexion à la base de données. Puis, dans un second temps, on écrit les fonctions permettant les différentes requêtes.

Chaque besoin est centralisé dans une fonction. Les fonctions ont toutes le même squelette :

1. Requête
2. Vérification du résultat obtenu
3. Envoi du résultat à l'application

Par exemple, voici la fonction permettant l'authentification de l'utilisateur :

```
public function getUserByEmailAndPassword($email, $password) {  
    // query  
    $result = mysql_query("SELECT * FROM client WHERE email = '$email'") or die(mysql_error());  
    // check for result  
    $no_of_rows = mysql_num_rows($result);  
    if ($no_of_rows > 0) {  
        // user found  
        $result = mysql_fetch_array($result);  
        $salt = $result['salt'];  
        $encrypted_password = $result['encrypted_password'];  
        $hash = $this->checkhashSHA($salt, $password);  
        // check for password equality  
        if ($encrypted_password == $hash) {  
            // user authentication details are correct  
            return $result;  
        }  
    } else {  
        // user not found  
        return false;  
    }  
}
```

Ici, les deux **entrées** de la fonction sont l'adresse email *\$email* et le mot de passe *\$password* saisis par l'utilisateur.

La **requête** consiste en la sélection, dans la table *client*, de tous les champs (numéro d'identification, nom, prénom, ...) relatifs au client dont l'email est égal à celui transmis en entrée de la fonction *\$email*.

Le résultat de la requête est stocké dans la variable *\$result*.

On procède ensuite à l'**évaluation du résultat** en testant le nombre de lignes retournées par la requête et donc stockées dans *\$result*. Si le nombre de lignes n'est pas nul c'est que la requête a bien retourné un résultat cohérent et on peut continuer le traitement. Dans le cas contraire, on sort de la fonction en retournant un variable booléenne nulle : *false*. Lorsque le résultat est cohérent, on le sectionne pour obtenir un tableau dont chaque

cellule contient un champ du résultat de la requête. Dans ce cas particulier de l'authentification, on vérifie que le hachage du mot de passe saisi *\$hash* correspond bien au hachage du mot de passe attendu *\$encrypted_password*. Si c'est bien le cas, on **retourne** alors le résultat de la requête *\$result* à l'application.

III.4 Application native (Android)

III.4.1 Configuration de l'environnement de développement

Avant de commencer je disposais déjà de l'environnement de développement Eclipse avec le plugin d'outils de développement Android 22.3 (Android Development Tools : ADT). J'ai effectué la mise à jour du kit de développement (SDK : Software Development Kit) Android.

Pour pouvoir exécuter l'application, il faut soit : créer un émulateur de terminal (AVD : Android Virtual Device), soit connecter son téléphone à l'ordinateur. En plus de s'avérer très lent, l'AVD utilise un clavier qwerty. C'est pour cela que j'ai privilégié l'utilisation de mon téléphone personnel.

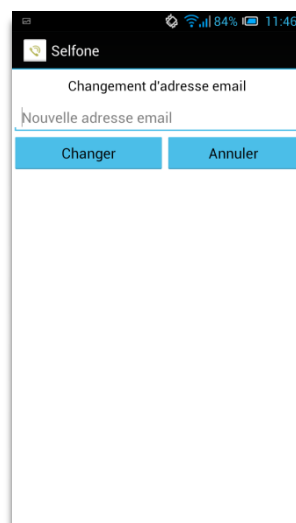
Enfin, j'ai installé le plugin Git permettant d'effectuer des sauvegardes de mon travail sur GitHub et donc de revenir en arrière en cas de problème.

III.4.2 Développement

Dès lors que les objectifs de l'application et les moyens mis en place pour y parvenir sont clairement définis, je peux passer au développement.

Sous Android, une fenêtre contient une partie visuelle (appelée « vue ») et une partie fonctionnelle (appelée « activité »). Ce qui veut dire que pour chaque fenêtre de l'application, il faut définir sa vue et son activité.

L'application Android suit la même logique quelque soit la fenêtre. Pour mieux la comprendre on se propose d'expliquer la création d'une fenêtre simple : celle permettant le changement d'adresse email.



La vue de cette page comporte plusieurs éléments :

- Un champ de texte contenant le titre de la vue : « Changement d'adresse email ».
- Une zone de texte éditable permettant la saisie de la nouvelle adresse email.
- Deux boutons : le bouton « Changer » permettant de valider le changement et le bouton « Annuler » pour annuler l'opération.

Pour créer cette vue, il a fallu créer les éléments qui la composent, comme on peut le voir dans le fichier XML suivant :

```
<!--Version et encodage du fichier -->
<?xml version="1.0" encoding="utf-8"?>

<!-- Pour placer les éléments dans une vue, il faut les inclure dans un conteneur -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff" >

    <!--Champ de texte contenant le titre -->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:gravity="center"
        android:text="Changement d'adresse email"
        android:textColor="#000000"
        android:textSize="17dp" />

    <!--Bouton Changer -->
    <Button
        android:id="@+id/btnChangeMail"
        style="@style/ButtonNiceBlueTheme"           ceci permet de styliser un élément
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/txtNewMail"
        android:layout_toLeftOf="@+id/btnAnnulerMail"
        android:text="Changer" />

    <!--Bouton Annuler -->
    <Button
        android:id="@+id/btnAnnulerMail"
        style="@style/ButtonNiceBlueTheme"
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/btnChangeMail"
        android:layout_alignBottom="@+id/btnChangeMail"
        android:layout_alignParentRight="true"
        android:text="Annuler" />
```

```

<!--Zone de texte éditable -->
<EditText
    android:id="@+id/txtNewMail"
    style="@style/EditTextNiceBlueTheme"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView"
    android:ems="10"
    android:hint="Nouvelle adresse email"
    android:inputType="textEmailAddress"
    android:textColor="#000000"
    android:textCursorDrawable="@null" />

</RelativeLayout>

```

activity_change_email.xml

Pour pouvoir manipuler cette vue, il faut l'associer à une activité. La voici :

```

// Définition du package
package com.cmrp.selfone;

// Importation des librairies requises
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;

import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.cmrp.selfone.library.UserFunctions;
import com.cmrp.selfone.R;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class ChangeEmailActivity extends Activity {

    private static String KEY_SUCCESS = "success";
    // variables permettant la communication avec la vue
    EditText txtNewMail; // zone de texte éditable
    Button btnChangeMail, btnAnnulerMail; // boutons

    /**

```

```

    * Called when the activity is first created.
    */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Dès lors que l'activité est créée, on lui associe sa vue.
        setContentView(R.layout.activity_change_email);

        // Mise en relation du bouton annuler de la vue avec la variable
        // correspondante
        btnAnnulerMail = (Button) findViewById(R.id.btnAnnulerMail);

        // Définition de l'action à réaliser lors de l'appui sur le bouton
        // annuler
        btnAnnulerMail.setOnClickListener(new View.OnClickListener(){
            public void onClick(View arg0){
                // Redirection de l'utilisateur vers le tableau de bord
                (Main.class)
                Intent annuler = new Intent(getApplicationContext(), Main.class);
                startActivity(annuler);
                finish();
            }
        });

        // Mise en relation de la zone de texte éditable et du bouton changer de
        // la vue
        // avec les variables correspondantes
        txtNewMail = (EditText) findViewById(R.id.txtNewMail);
        btnChangeMail = (Button) findViewById(R.id.btnChangeMail);

        // Définition de l'action à réaliser lors de l'appui sur le bouton
        // changer
        btnChangeMail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // exécution de la tâche asynchrone permettant la communication avec
                // la base de données
                NetAsync(view);
            }
        });
    }

    // Classe permettant la communication avec la base de données de manière
    // asynchrone
    // (car il est impossible d'effectuer des requêtes Internet de manière
    // synchrone)
    private class NetCheck extends AsyncTask<String,String,Boolean>{
        @Override
        protected void onPreExecute(){
            super.onPreExecute();
        }

        // Fonction permettant d'évaluer l'état de la connexion Internet du
        // téléphone
        @Override
        protected Boolean doInBackground(String... args){
            ConnectivityManager cm = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);

```

```

        NetworkInfo netInfo = cm.getActiveNetworkInfo();
        if (netInfo != null && netInfo.isConnected()) {
            try {
                URL url = new URL("http://www.google.com");
                HttpURLConnection urlc = (HttpURLConnection)
url.openConnection();
                urlc.setConnectTimeout(3000);
                urlc.connect();
                if (urlc.getResponseCode() == 200) {
                    return true;
                }
            } catch (MalformedURLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        return false;
    }
    @Override
    protected void onPostExecute(Boolean th){
        if(th == true){
            // Exécution du processus de changement d'email
            new ProcessChangeEmail().execute();
        }
        else{
            Toast.makeText(getApplicationContext(),
mobile).",
                                "Connexion Internet requise (Wi-Fi ou réseau
                                Toast.LENGTH_LONG).show();
                                //alert.setText("Error in Network Connection");
        }
    }
}

// Classe permettant le processus de changement d'email
private class ProcessChangeEmail extends AsyncTask<String, String,
JSONObject> {
    // chaîne de caractères comportant la nouvelle adresse email
    String newEmail;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Récupération de la nouvelle adresse mail
        newEmail = txtNewMail.getText().toString();
    }

    @Override
    protected JSONObject doInBackground(String... args) {
        UserFunctions userFunction = new UserFunctions();
        // Recours à la méthode changeEmail
        JSONObject json = userFunction.changeEmail(newEmail, Login.email);
        return json;
    }

    // Fonction qui s'exécute en dernière

```

```

// Cette fonction permet d'avertir l'utilisateur (par l'intermédiaire
d'un Toast)
// que l'action s'est bien (ou mal) déroulée
@Override
protected void onPostExecute(JSONObject json) {
    try {
        if (json.getString(KEY_SUCCESS) != null) {
            String res = json.getString(KEY_SUCCESS);
            if (Integer.parseInt(res) == 1) {

                Intent mainMenu = new Intent(getApplicationContext(),
Main.class);

                startActivity(mainMenu);

                Toast.makeText(getApplicationContext(),
                    "Adresse email modifiée.",
                    Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(),
                    "Une erreur est survenue.",
                    Toast.LENGTH_LONG).show();
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

public void NetAsync(View view){
    new NetCheck().execute();
}
}

```

ChangeEmailActivity.java

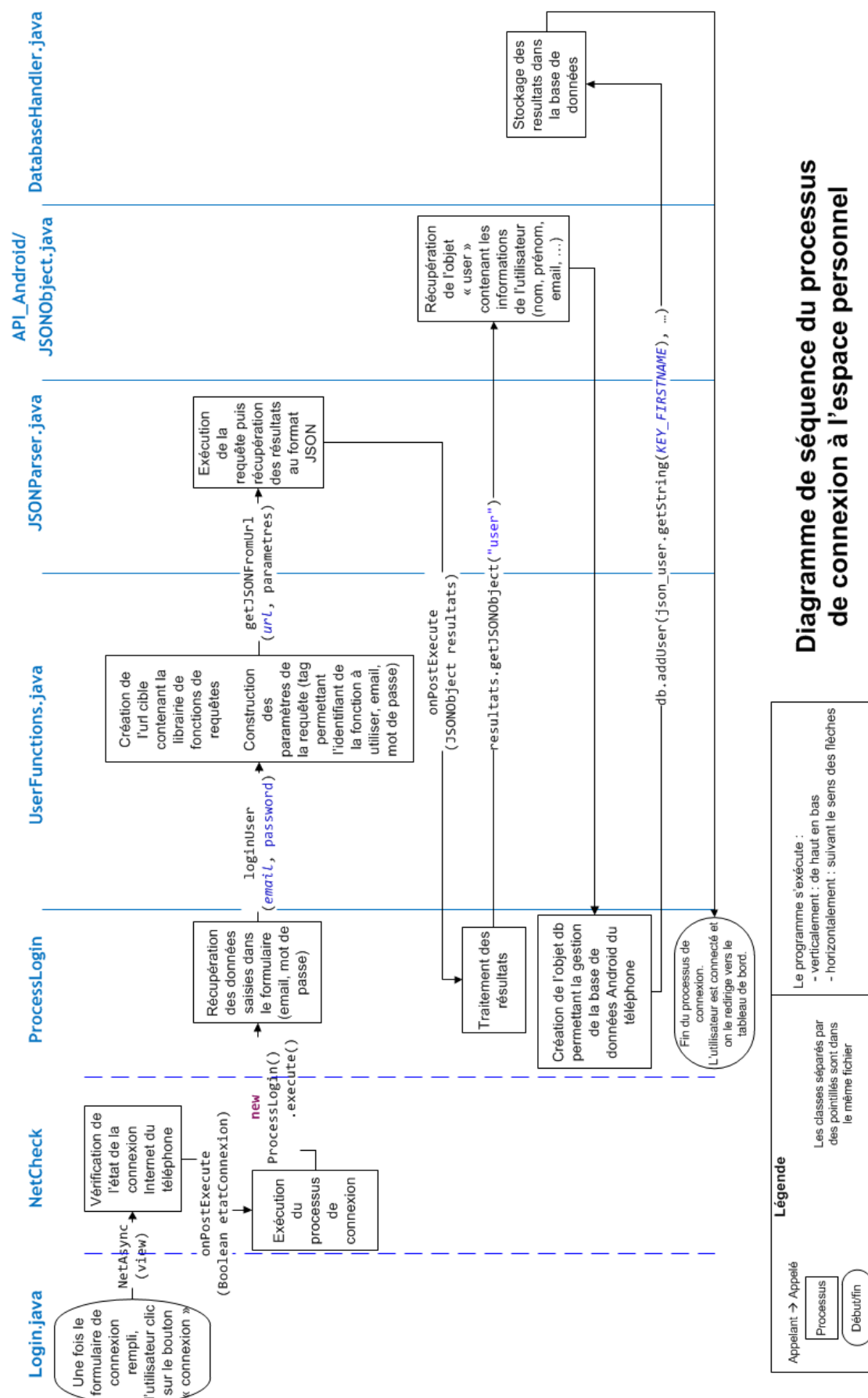
Ici, la classe ChangeEmailActivity permet d'associer les éléments de la vue à l'activité. En effet, on établit les relations entre les éléments de la vue et les variables pour être en mesure de manipuler la vue. On gère ainsi les boutons et le zone de texte éditable.

Ensuite, on met en place la classe NetCheck pour établir le lien entre l'application et la base de données. En fait, il s'agit d'une tâche asynchrone qui permet de vérifier l'état de la connexion Internet du terminal (en l'occurrence, un téléphone sous Android).

Enfin, la classe ProcessChangeEmail est une tâche asynchrone qui permet la communication avec la base de données en utilisant le format de données textuelles JSON.

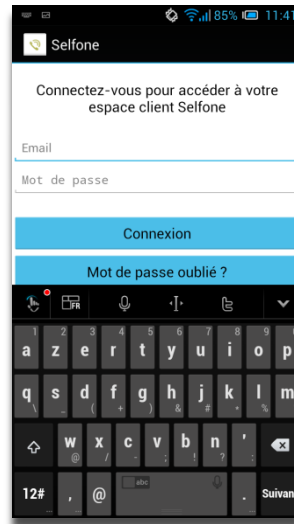
Cette logique est identique quelque soit la fenêtre. Elle est illustrée par le diagramme de séquence de la page suivante.

III.4.2.1 Diagramme de séquence du processus de connexion à l'espace personnel



III.4.2.2 Ergonomie

Les champs de saisis de l'application ont été passés au crible afin de rendre l'expérience utilisateur plus simple. Ainsi, chaque champ de saisi possède ses propres propriétés (mail, mot de passe, texte, ...).

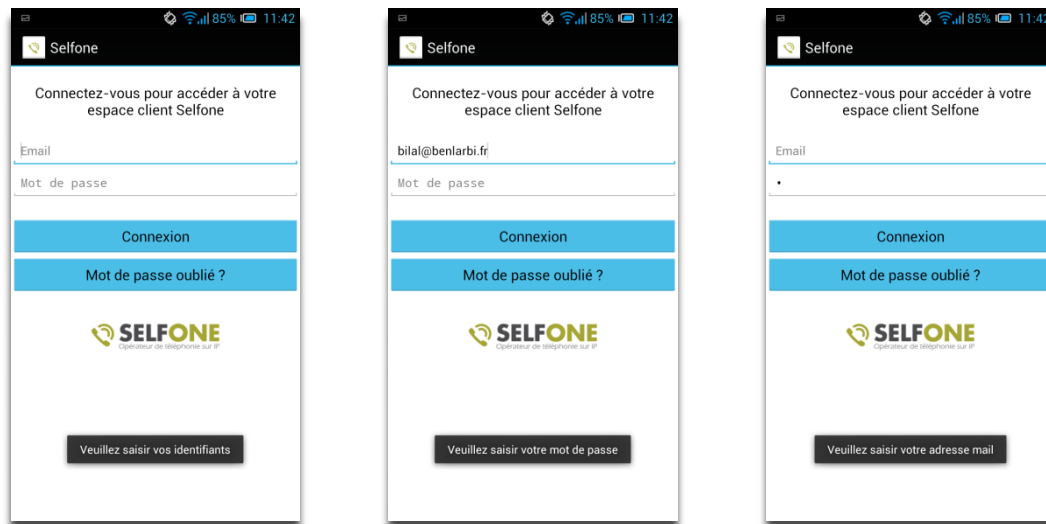


Sur cette capture d'écran on voit bien que le clavier s'adapte au type de champ à éditer. Ainsi, la saisie d'adresse mail provoque l'apparition du raccourci au symbole @ sur le clavier.

III.4.2.3 Vérifications préalables

Avant d'exécuter les scripts, on s'assure du bon remplissage des champs. Cela permet de n'exécuter les scripts qu'au moment nécessaire pour ne pas faire attendre l'utilisateur d'un traitement qui n'aura pas lieu.

Sur les captures d'écran de la page suivante on voit bien qu'un message d'erreur apparaît lorsque l'utilisateur essaye de valider un formulaire sans remplir tous les champs.



III.4.2.4 Problèmes rencontrés

J'ai été confronté à un problème lorsque j'ai voulu observer le contenu de la base de données créée sur le téléphone contenant les données concernant l'utilisateur (informations personnels, crédit de communication, tickets d'incidents).

Pour y parvenir, plusieurs solutions existent et nécessitent :

- soit un téléphone rooté
- soit une mémoire externe sur le téléphone (carte SD)

Or, dans mon cas, je n'ai ni de téléphone rooté, ni un emplacement de mémoire externe.

J'ai donc pensé à utiliser l'émulateur (AVD), mais le clavier de l'émulateur est en QWERTY et il semble impossible à paramétrer en AZERTY. J'ai donc essayé de continuer avec l'émulateur malgré tout, mais il fut impossible pour moi de trouver les caractères "." et "@" contenu dans l'adresse mail utilisé pour les tests. J'ai alors choisi de déplacer le problème en créant un nouvel utilisateur de test avec une adresse mail simple à utiliser avec le clavier QWERTY. Lors de la connexion, l'émulateur est resté bloqué (il a « freeze ») plus de 10 minutes. Je renouvelé l'opération qui a fonctionné cette fois ci, mais je n'ai malgré tout pas pu observer la base de données...

Finalement, je ne suis pas parvenu à résoudre ce problème. J'ai toutefois pu continuer à avancer et j'ai dû faire preuve d'abstraction pour bien comprendre les mécanismes des bases de données Android intégrée au téléphone.

III.5 Application web

L'espace client Selfone doit être accessible à tous les clients Selfone, et ce, quel que soit le système d'exploitation de leur smartphone.

Ceci étant dit, l'application Android seule ne suffit pas, il faut la même application pour les principales plateformes du marché (Apple et Microsoft).

Or, pour développer des applications iOS (iPhone), le matériel Apple est requis : un ordinateur sous Mac avec Xcode (environnement de développement d'applications iOS) et un terminal mobile (iPhone). Ni l'entreprise, ni moi ne disposons de ce type de matériel. Le développement d'application iOS native est donc impossible.

Comme nous ne pouvons pas faire l'impasse sur les iPhones, il a fallu trouver une parade. C'est là que la solution de l'application web a été envisagée.

En effet, une application web ne nécessite que le navigateur Internet d'un terminal connecté à la toile. Côté développement, j'ai opté pour la technologie jQuery Mobile pour sa puissance et sa compatibilité multiplateforme.

III.5.1 Remarque

Avec du recul, je réalise que j'aurais dû anticiper la question du matériel pour développer sous iOS avant même de commencer à développer l'application Android. En effet, l'application Android est fonctionnelle, au même titre que l'application web qui a été développée beaucoup plus rapidement et qui offre un rendu plus harmonieux en termes de visuels pour des fonctionnalités identiques. Finalement, cela m'a permis de me rendre compte, dans la réalité, qu'il faut toujours réfléchir avant d'agir. En fait, il ne faut pas perdre de vue la problématique pour rester cohérent et efficace dans son travail.

III.5.2 Analyse

Avant de commencer à développer l'application web, il faut penser à la sécurité.

Comme il s'agit d'une application web consultable depuis un navigateur, il faut s'assurer de rendre toutes les pages inaccessibles à un utilisateur non connecté.

De plus, comme le détail d'un ticket d'incident s'affiche dans une page de type GET (ticket-detail.php?id=12345), il faut restreindre l'accès aux tickets des autres utilisateurs.

III.5.3 Développement

L'application à développer est en tout point similaire à l'application Android. J'ai donc repris l'API permettant la liaison à la base de données de l'application Android.

Concernant la navigation, j'ai conservé la même logique que celle de l'application Android (cf. p.13 : *III.2.3 Cycle d'utilisation*).

Il a fallu développer chaque page, une par une ; avant des les interconnecter entre elles. En fait, il s'agit de pages web simples qui font appel à jQuery Mobile.

Pour mieux comprendre, vous trouverez le code source de la page de connexion sur la page suivante.



The image shows a mobile web interface for 'Espace client Selfone'. The title is 'Connexion à l'espace client Selfone'. It features two input fields: 'Email' with the value 'bilal.benlarbi@gmail.com' and 'Mot de passe' with a single dot. Below the inputs are two buttons: 'Connexion' with a right arrow icon and 'Mot de passe oublié ?' with a magnifying glass icon. At the bottom is the 'SELFONE' logo with the tagline 'Opérateur de téléphonie sur IP'.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8"/>
5      <meta name="viewport" content="width=device-width, initial-scale=1"/>
6      <title>Espace client Selfone</title>
7
8      <!-- Appels des feuilles de styles prédéfinies -->
9      <link rel="stylesheet" type="text/css" href="./assets/css/jquery.mobile-1.4.2.min.css" />
10     <link rel="stylesheet" type="text/css" href="./assets/css/main.css" />
11
12     <!-- Favicon -->
13     <link rel="shortcut icon" href="./assets/img/favicon.png">
14
15     <!-- Appels des scripts jQuery et jQuery Mobile -->
16     <script type="text/javascript" src="./assets/js/jquery-2.1.0.min.js"></script>
17     <script type="text/javascript" src="./assets/js/jquery.mobile-1.4.2.min.js"></script>
18 </head>
19 <body>
20     <div data-role="page">
21         <!-- Entête de la page -->
22         <div data-role="header">
23             <!-- Titre -->
24             <h1>Espace client Selfone</h1>
25         </div>
26
27         <!-- Contenu de la page -->
28         <div data-role="content">
29             <h2>Connexion à l'espace client Selfone</h2>
30
31             <!-- Formulaire de connexion -->
32             <form action="connect.php" method="post">
33                 <fieldset>
34                     <div data-role="fieldcontain">
35                         <label for="email">Email</label>
36                         <input type="email" name="email" id="email" placeholder="Email" value=""
37                             required/>
38                         <label for="password">Mot de passe</label>
39                         <input type="password" name="password" id="password" placeholder="Mot de
40                             passe" value="" required/>
41                         <br/>
42                         <input type="submit" name="submit" id="submit" data-icon="arrow-r" data-
43                             iconpos="right" value="Connexion">
44                         <a href="./forgotten-password.php" data-role="button" data-icon="search"
45                             data-iconpos="right">Mot de passe oublié ?</a>
46                     </div>
47                 </fieldset>
48             </form>
49         </div>
50
51         <!-- Pied de page -->
52         <div data-role="footer">
53             <!-- Lien vers le site Selfone.fr -->
54             <a style="display:block;" href="http://selfone.fr/" target="_blank">
55                 
57             </a>
58         </div>
59     </div>
60 </body>
61 </html>

```

IV. DEVELOPPEMENT WEB

CMRP développe des sites web pour ses clients. Lors de mon stage, il a été question de création et de refonte de sites web.

Je me suis chargé exclusivement du développement. Je n'ai donc pas été en relation avec les clients. C'est mon maître de stage qui m'a positionné sur des projets, au fur et à mesure de mon avancement.

IV.1 Configuration de l'environnement de développement

Pour le développement web, j'utilise les logiciels suivants :

- Navigateur :
 - pour le développement : Mozilla Firefox 29 avec l'extension Firebug 1.12 (pour inspecter les éléments d'une page web),
 - pour vérifier la compatibilité multi navigateurs, j'utilise les principaux navigateurs du marché : Internet Explorer 11, Google Chrome 34 et Safari 5.1.
- Editeur :
 - Sublime Text 2 avec l'extension Emmet (pour générer du code HTML et CSS en utilisant seulement des abréviations).
- Client FTP :
 - FileZilla 3.8.
- Editeur d'image :
 - Gimp 2.

Par ailleurs, lorsqu'il a fallu gérer une base de données, j'ai toujours travaillé sur mon serveur personnel (benlarbi.fr). En fait, mon maître de stage me donnait les noms des champs et des tables, afin de reproduire les bases à l'identique et de ne pas perturber les serveurs en production. Puis, j'ai inséré quelques lignes dans les bases de données reproduites afin de simuler le fonctionnement normal.

IV.1 Refontes de sites web

IV.1.1 DOMAINE PINSON



IV.1.1.1 Description du besoin

Le domaine Pinson est situé en Bourgogne. Il produit et commercialise des vins de Chablis. Il possède un site web ayant deux sections principales :

- la vitrine : avec l'histoire du domaine, ses actualités, ses techniques, ainsi qu'une page de contact,
- la vente en ligne de bouteilles de vins.

L'objectif de ce travail a été de convertir toute la partie vitrine de Flash en page web statique (HTML/CSS).

IV.1.1.2 Analyse du besoin

Le site vitrine contient essentiellement du texte et des images, donc aucun élément complexe à première vue.

La difficulté principale de ce projet a résidé dans la mise en forme, car il ne doit pas y avoir de différences visuelles entre le site de référence et le nouveau. Ainsi, une attention particulière a été portée à la disposition des éléments, mais aussi aux couleurs, tailles et police des textes.

De plus, les animations Flash ont été remplacées par des animations CSS similaires (images qui se contrastent et liens qui changent de couleurs au passage de la souris).

IV.1.1.3 Réalisation

Pour réaliser cette conversion, j'ai simplement créé des pages web statiques HTML/CSS reprenant le contenu nécessaire.

Le site ayant des parties invariables sur toutes les pages, j'ai eu recours aux *include* PHP. Ce procédé permet d'inclure du code d'un fichier dans un autre fichier. Je l'ai utilisé pour répéter entête et pied de page sur les différentes pages.

Centraliser le code permet l'harmonisation du site web et un gain de temps aussi bien pour le développement que pour le dépannage.

IV.1.2 AVOCATS VIGNET



IV.1.2.1 Description du besoin

Le cabinet d'avocats Vignet est situé dans l'Yonne. Ses membres ont demandé une refonte du site web vitrine, afin d'avoir un design plus harmonieux et moderne.

Le site web est constitué de deux parties. La première comporte les pages suivantes : accueil, honoraires, engagements, organisation du cabinet (description du personnel) et contact. La seconde partie comporte les pages décrivant les différents types de droits défendus par les avocats de ce cabinet : droit pénal, droit commercial et des affaires, droit du travail, droit de la famille, droit civil et droit rural.

IV.1.2.2 Organisation

Une maquette de page type (en l'occurrence, la page d'accueil) sous forme d'image (fichier Photoshop) m'a été fournie par mon maître de stage, qui l'a lui-même reçue d'un commanditaire en relation avec le client.

IV.1.2.3 Analyse du besoin

Il y a essentiellement du texte à mettre en forme et deux fonctionnalités particulières :

- carrousel sur l'entête du site
- Google map sur contact.php

J'ai dû allonger une image dans la hauteur pour qu'elle ait les bonnes proportions pour être intégrée au diaporama.

IV.1.2.4 Réalisation

Pour commencer, j'ai créé une page web à part entière (HTML/CSS) la plus fidèle possible à la maquette. Puis, le client a demandé d'apporter quelques modifications (taille/couleur de police, disposition de certains éléments) avant de valider mon travail et me faire parvenir le contenu de chaque page (texte, image, ...) par l'intermédiaire du commanditaire. C'est alors que j'ai pu réaliser le reste du site.

Pour cela, j'ai récupéré les éléments visuels (images de fond, photos, logos, ...).

IV.1.2.5 Problème

Le script que j'ai utilisé pour mettre en place le diaporama a provoqué des erreurs de compatibilité multi navigateurs. En effet, l'affichage du diaporama a provoqué des effets indésirés sur un poste Mac navigant sur Safari (images rognés, menu mal disposé).

J'ai utilisé des hacks CSS spécialement conçus pour Safari pour corriger ce problème.

IV.2 Création de sites web

IV.2.1 CMRPGRAPHICDESIGN



IV.2.1.1 Description du besoin

La marque Kopy Minute de CMRP ne disposait pas jusqu'alors de site web. La tâche de le créer m'a été confiée. Il s'agit en fait d'un site vitrine où la marque Kopy Minute expose ses domaines de compétences.

Katerine (graphiste de Kopy Minute) m'a transmis la maquette (en format Photoshop) qu'elle a réalisée et je l'ai reproduite en page web statique (HTML/CSS).

IV.2.1.2 Analyse du besoin

Sur la maquette fournie, du texte et plusieurs images de fond se chevauchent. Les difficultés principales résident dans la disposition des éléments et la mise en place de galeries d'images.

IV.2.1.3 Problème

La compatibilité multi navigateurs n'a pas été évidente de prime abord. En effet, le rendu graphique fut différent d'un navigateur à l'autre. La gestion des marges a posé problème. Il a fallu recourir à des hacks CSS spéciaux pour Firefox, Internet Explorer et Chrome sur les éléments problématiques.

D'autres problèmes de compatibilité navigateurs sont apparus, mais, cette fois, sur le navigateur Firefox sous Mac. Pour pallier ce problème, j'ai mis en place un script JavaScript permettant la modification du style en fonction du navigateur et du système d'exploitation utilisés. Comme je ne dispose pas d'un ordinateur sous Mac, pour pouvoir observer le problème j'ai pris la main à distance sur le Mac de Katherine à l'aide du logiciel TeamViewer.

Pour mettre en place les galeries d'images, j'ai utilisé un script que j'ai trouvé sur Internet et je l'ai customisé pour qu'il corresponde bien à notre besoin. Ainsi, j'ai modifié le style et le texte (traduction en français) de la galerie.

IV.2.2 BLOG D'ASSISTANCE TECHNIQUE SELFSEVEUR



The screenshot shows the homepage of the 'Assistance Selfserveur' blog. The header features the title 'Assistance Selfserveur' and the subtitle 'Blog d'assistance technique de Selfserveur'. Navigation links include 'Accueil', 'Linux', 'Réseaux', and 'Serveur VOIP'. The main article is titled 'Installation et configuration d'un serveur SMTP sur serveur Centos', dated '23 avril 2014', and categorized under 'Linux'. The article content includes instructions to 'Désinstaller le serveur sendmail installé par défaut' using the command '# yum remove sendmail', to 'Installer postfix' using '# yum install postfix', and to 'Sauvegarder le fichier principal'. A sidebar on the right lists 'Articles récents' and 'Catégories'.

IV.2.2.1 Description du besoin

L'équipe technique de Selfserveur s'occupe, entre autres, de l'administration de serveurs. Il n'est pas rare que les opérations à effectuer soient identiques d'un serveur à l'autre. Les techniciens en charge des opérations cherchent souvent la manière d'opérer sur Internet. Ainsi, il est courant qu'un technicien ait à effectuer la même opération sur deux serveurs différents, à des moments différents. La solution à mettre en place est donc la même et le technicien doit se souvenir de ce qu'il a fait la première fois afin de le répéter. Dans le cas où il ne s'en souvient pas entièrement il va rechercher une nouvelle fois la solution sur Internet.

Le fait de chercher la même information plusieurs fois n'est pas optimal, quand on peut l'enregistrer et la retrouver à tout moment.

La mise en place d'un blog répertoriant les routines à effectuer est donc une solution pour un gain de temps et d'efficacité. Le blog serait donc alimenté par ses contributeurs qui sont des membres de l'équipe technique. La consultation du blog est cependant ouverte à tous, puisque le blog est disponible sur Internet.

Par ailleurs, le blog permet de faire des « backlinks » sur selfserveur.com. Cela contribue donc à mieux référencer le site selfserveur.com.

De plus, c'est une porte d'entrée supplémentaire pour un nouveau client.

IV.2.2.2 Analyse du besoin

Pour mettre en place ce blog, le CMS WordPress a été adopté pour sa simplicité d'utilisation. Le fait que je sois familier à cette technologie a aussi pesé dans la balance.

Nous avons choisi de désactiver les commentaires pour permettre à l'équipe technique de rester concentrée ses objectifs principaux.

Comme le blog contient pour l'essentiel du texte, il ne nécessite pas de fonctionnalité particulière. Le recours à des plugins n'est pas nécessaire et a été évité au maximum, ce qui permet d'avoir un blog WordPress léger et optimal.

IV.2.2.3 Administration serveur

J'ai assisté à l'installation, par M. Ribeiro, du serveur dédié privé (VPS : Virtual Private Server) qui héberge le blog. Une fois le VPS installé, j'ai été en charge de l'administration du serveur.

Il a fallu gérer les droits d'accès aux fichiers/dossiers.

Pour cela, j'ai utilisé le client SSH (Secure Shell : terminal sécurisé) PuTTY.

IV.2.2.4 Développement

Le thème Twenty Thirteen est responsive (adapté aux mobiles) et simple et correspond à notre besoin. Pour personnaliser les apparences du site, j'ai créé un thème enfant de Twenty Thirteen (menu responsive, entête du site personnalisé qui se rapproche de celui du site de Selfserveur).

IV.2.2.5 Problème

Nous pensons avoir été victime d'une attaque par dénis de service, car M. Ribeiro a été réveillé en pleine nuit par des avertisseurs de serveur en dysfonctionnement (on dit que le serveur est « down »). En fait, de nombreuses requêtes vers le blog ont été demandées ce qui provoqua un crash sur le serveur parent du VPS (le VPS est comme un fils du serveur parent). Pour en savoir plus, nous nous sommes penchés du côté des logs et il s'avéra que les lignes critiques ont été effacées.

IV.2.2.6 Solution

Pour corriger ce problème, j'ai procédé à la sécurisation du nouveau blog mis en place :

- Modification du nom du répertoire wp-content
- Mise en place d'un .htaccess / .htpasswd sur le répertoire wp-admin
- Plugin Login Lockdown pour limiter le nombre de tentatives de connexion au panel admin (attaques par force brute)
- Suppression du compte admin, créé par défaut par Wordpress
- Remplacement du préfixes des tables de la bdd WP (wp-... => n2t-...)
- Sauvegardes (backups) régulières de la base de données WordPress et du site
- Mises à jour de WordPress
- Mot de passe plus complexe
- Ajout de clés de sécurité secrète dans wp-config.php
- Masquage de la version de WordPress
- Masquage des erreurs de connexion
- Désactivation de Windows Live Writer (inutile dans ce contexte et source d'insécurité)
- Suppression du fichier readme.html

IV.2.2.7 Remarque

C'est la première fois que j'ai été confronté à un tel problème. Le hacking fait pleinement partie de l'Internet et en être victime permet de garder à l'esprit l'importance de la sécurisation d'un site web. Le fait d'être attaqué constitue une sorte de défi entre le hackeur et moi-même qui doit tout mettre en œuvre pour lui rendre la tâche la plus difficile possible.

IV.3 Campagne d'emailing



Version web | Se désinscrire

Accédez au très haut débit avec la fibre optique

Cette révolution technologique possède de réels avantages

- Débits **surboostés** à plus de 2 Gb/s (**1000 fois supérieur** à l'ADSL) ce qui donne accès à une facilité et une rapidité de navigation Internet et une meilleure qualité téléphonique.
- Ne subit qu'une **très faible perte de signal** sur la distance. Ainsi, la fibre comporte moins de latence ce qui offre un meilleur délai de transmission pour Internet et la téléphonie.
- Installation du routeur par un professionnel de votre région.

Test d'éligibilité

TESTER MON ÉLIGIBILITÉ ➔

Testez sans plus tarder votre éligibilité à ce service **fiable** et bénéficiez de la connexion Internet la plus **performante** du marché !

La fibre optique répond à tous vos besoins professionnels

La technologie de la fibre optique couvrira de façon plus pertinente vos besoins en matière de bande

IV.3.1 Description du besoin

Afin de développer l'activité de fournisseur d'accès à Internet de CMRP par la fibre optique, nous avons mis en place une campagne d'emailing à objectif commercial.

L'objectif de cette campagne est de faire faire au destinataire un test d'éligibilité à la fibre optique, dans lequel il renseigne ses coordonnées. Cela permet d'élargir la base de contacts en faisant connaître l'activité fibre optique de CMRP.

IV.3.2 Analyse du besoin

Ma mission sur ce projet a été globale : de la rédaction à la réalisation technique, en passant par la mise en forme. Il s'est agi de mettre en exergue les avantages du service fibre optique de CMRP.

IV.3.3 Réalisation

Pour réaliser cette newsletter, j'ai eu recours à un template (gabarit) du site Campaign Monitor. Ce template a été choisi, car, en plus d'être esthétique et gratuit, il est responsive (adapté au format mobile). Ces caractéristiques sont des points forts, car, d'une part, l'image de l'entreprise passe par l'esthétique, et, d'autre part, de plus en plus de professionnels utilisent leur smartphone pour consulter leur courrier électronique.

Par ailleurs, il existe plusieurs logiciels de messagerie et plusieurs webmails (interfaces web permettant la consultation d'emails) qui utilisent des moteurs de rendus différents. C'est un facteur important à prendre en compte pour fournir un email compatible avec le maximum de ces outils.

Après plusieurs tests sur différents webmails, navigateurs, logiciels de messagerie et terminaux mobiles, il semble que le résultat soit concluant.

IV.3.4 Déploiement

Pour déployer cette campagne d'emailing, j'ai utilisé l'outil Mail Chimp qui est assez performant. Il permet d'envoyer l'emailing à une liste de destinataire de façon simple et gratuite dans une limite de 2 000 destinataires.

IV.4 Campagne de référencement naturel de Selfserveur.com

IV.4.1 Description du besoin

Mettre en avant le site de Selfserveur permet d'obtenir de nouveaux clients potentiels. Pour ce faire, nous avons mené une campagne de référencement naturel.

IV.4.2 Organisation

Cette campagne de référencement a été menée en collaboration avec Mélanie Béchu (chargée marketing). Pour travailler efficacement, nous nous sommes répartis les tâches selon nos compétences. Ainsi, Mélanie s'est occupée de la partie rédactionnelle en mettant en avant les mots clefs. Pour ma part, je me suis chargé de la partie technique.

IV.4.3 Outils utilisés

Outil d'analyse : <http://www.site-analyzer.com>

Outil d'analyse de mots clefs : <http://www.outiref.com/>

Outil d'aide de définition de mots clefs : <http://www.google.fr/trends/>

IV.4.4 Réalisation

Voici les principales actions que j'ai réalisées :

- validation de toutes les pages du site selon les normes W3C
- suppression de certaines balises de titre incohérentes
- suppression des liens en double au sein d'une même page
- correction des attributs « alt » des images
- correction du lien vers l'accueil : non pas index.php mais ./
- mise en exergue des mots clefs avec les balises strong, u et em
- corrections des titres et des descriptions Google
- ajout des liens réseaux sociaux
- mise en place de robots.txt & sitemap.xml.

IV.4.5 Résultat

Après avoir apporté toutes les corrections nécessaires, le score indiqué par site-analyzer est passé de 45% de réussite à 79%. Nous ne pouvons pas améliorer ce score, car il est donné à titre indicatif ; certains conseils ne sont pas adaptés au contexte.

V. CONCLUSION

V.1 Conclusion technique

Lors de mon stage chez CMRP, j'ai eu la chance de travailler sur des projets de natures différentes. J'ai développé des sites et une application web, une application Android, mais j'ai aussi mené une campagne d'emailing et une campagne de référencement.

Cela m'a permis d'approfondir les connaissances que j'ai acquises durant ma formation à l'IUT. J'ai découvert l'emailing qui m'était jusqu'alors complètement étranger. Cela m'a intéressé, car c'est un élément important dans le monde de l'Internet. Je suis fier d'avoir pu mettre en place une campagne d'emailing sans connaissances préalables sur ce sujet. Je me suis servi d'Internet pour apprendre et j'ai abouti à un résultat très satisfaisant pour l'entreprise.

En outre, j'ai dû travailler en autonomie et apprendre de nouvelles choses par moi-même. Pour cela, j'ai utilisé les notions acquises au cours de ma formation que j'ai étoffées grâce à Internet qui est un outil très puissant. J'ai essentiellement eu recours :

- au moteur de recherche Google,
- à la librairie Android qui est une référence officielle,
- au forum de développement très actif Stack Overflow,
- à OpenClassrooms (Site du Zéro) et à Developpez.com qui regroupent des cours d'informatique et des forums,
- à l'encyclopédie en ligne Wikipédia.org.

Pendant mon stage, j'ai été amené à utiliser l'anglais tous les jours. J'ai utilisé cette langue pour les recherches sur Internet. C'est la langue internationale : les développeurs du monde entier communiquent en anglais. Les ressources disponibles en anglais sont largement plus nombreuses que les ressources en français. Les langages de programmation étant écrits en anglais, j'ai aussi utilisé l'anglais pour coder.

Avant le stage, j'ai essentiellement développé des sites web à l'aide du framework Bootstrap qui est compatible avec les dernières versions des navigateurs majeurs. Je n'ai donc été que peu confronté aux problèmes de compatibilité multi navigateurs. Le stage m'a permis de me rendre compte que cet élément est important et qu'il faut constamment le garder à l'esprit.

V.2 Conclusion générale

J'ai effectué mon stage licence professionnelle développement web et mobile au sein de l'entreprise CMRP. Lors de ce stage de 16 semaines, j'ai pu mettre en pratique les connaissances et les méthodes acquises durant ma formation à l'IUT d'Orléans.

J'ai réussi à mener à bien les missions qui m'ont été confiées.

Travailler sur des projets professionnels concrets m'a beaucoup motivé. Cela a donné du sens à mes études, et je suis fier que mon travail soit utile à l'entreprise.

Ce stage en entreprise a été l'occasion pour moi d'entrer en contact avec des professionnels. J'y ai entrevu le quotidien de différents corps de métiers comme les commerciaux, les développeurs et les chargés de communication.

Ce stage m'a permis de découvrir le travail dans une petite structure de 6 salariés. Mon précédent stage s'étant déroulé dans une entreprise de 180 personnes, je peux désormais me faire une idée des différences entre ces deux types de structures. Il en ressort que dans une petite structure on travaille sur plusieurs projets dans des domaines différents, alors que dans des plus grosses entreprises, le travail est segmenté et on travaille généralement sur de gros projets à plus long terme.

Par ailleurs, l'ambiance à l'entreprise CMRP est plutôt familiale. On se tutoie et il n'y a pas de stress, ce qui permet de travailler efficacement.

Je suis reconnaissant envers l'entreprise de m'avoir donné ma chance et m'avoir permis d'avoir une première expérience professionnelle dans l'Internet et la mobilité, ce qui me conforte dans mon orientation professionnelle à venir.

Par ailleurs, avant d'arriver en stage, j'hésitais entre terminer mes études après la licence professionnelle, donc au niveau Bac +3, et poursuivre mes études jusqu'au niveau ingénieur (Bac +5). Le stage en développement web et mobile m'a convaincu de poursuivre mes études, en alternance, dans ce domaine. Cela me permettra de me positionner en tant qu'« expert » sur des projets innovants.