



Université Blaise Pascal

UFR Sciences et Technologies
Master Informatique Décisionnelle et Logicielle



Agence d'application mobile
et d'application métier

Rapport du stage M2 informatique parcours GLIA



WEB SERVICES DE NOTIFICATIONS PUSH POUR APPLICATIONS MOBILES

Réalisé par:

M Xia WU, *Étudiant M2 GLIA IDL de l'UBP*
xia.wu.france@gmail.com / 06 27 87 61 25

Tuteur du stage:

M Damien RUIZ, *Président du DTWeb*

Encadré par:

Mme Fatiha BENDALI-MAILFERT, *Professeur de l'UBP*

Date de la soutenance: 2nd Septembre 2014

Du 28/03/2014 au 29/08/14

Remerciements

Tout d'abord, je souhaite remercier M Damien RUIZ le président de l'entreprise **DTWeb** pour l'opportunité du stage qu'il m'a offert, ainsi que le professeur Fatiha BENDALI-MALFERT à l'ISIMA qui a bien voulu être mon encadreur durant ce stage. Le projet Push4Apps est une bonne expérience professionnelle qui m'a permis de compléter mes connaissances dans le domaine du SOA et mes compétences sur la réalisation du serveur de web services.

J'aimerai également remercier Mickael, Soufiane, Thibaut ainsi que les autres collaborateurs pour leurs inspirations, leur aide lors du stage.

J'aimerai faire parvenir mes remerciements aux enseignants et aux administrateurs de l'université Blaise-Pascal, qui n'ont pas hésité à me conseiller tout au long de ce stage.

Résumé

Push4Apps est un serveur de **notification push mobile**, qui offre un ensemble de web services basés sur la protocole **HTTP**. Il reçoit des données venant d'un web service afin de les transmettre sur des appareils mobiles.

Push4Apps collecte des informations de clients pour établir des statistiques des envoies de push. La procédure du service du push est basée sur le modèle **BPM** qui sert à l'optimisation des charges dynamiques des web services. L'optimisation en question est établie sur trois niveaux:

Le premier concerne le modèle **BPM** réalisé par l'outil de modélisation Lucidchart, ce qui facilite le développement, la réalisation et la maintenance du serveur Push.

Le second niveau est basé sur la réalisation de web services. L'implémentation de web services est basé sur le framework **RESTeasy** qui permet de simplifier le développement de web services sur le protocole **HTTP**.

Le troisième niveau concerne l'**optimisation** et la coopération entre différents serveurs. Chaque serveur s'occupe de certaines tâches de web services en échangeant et partageant des données.

Mots clés:

notification push mobile, web service RESTful, RESTeasy, BPM, HTTP, optimisation

Abstract

Push4Apps is a **mobile push notification** server which provides a collection of **web services** based on **HTTP** protocol. It receives those datas come from a web service. After that, it sends thoses datas on smartphones.

Push4Apps collects its client's information in order to produce push sending statistics. The push sending process is based on **BPM** model which is used for the optimization of dynamic charge. So, it could be brick down in threee levels:

The first level is about those BPM model realized by Lucidchart. It helps the development, the realization and the maintenance.

The second level is based on web services realization. It used the framework **RESTeasy** which provides APIs in order to reduce developing task.

The third level is about **optimization** and cooperation between different servers of web service. Every server is charged by some tesk and they exchange and share resultats

Keywords:

mobile push notifications, web service RESTful, RESTeasy, BPM, HTTP, optimisation

Table des matières

Remerciements	2
Résumé	3
Abstract	3
Table des matières	4
Présentation de la société DTWeb	6
Problématique	8
Notification Push	8
Réalisation des web services basiques	8
Bureau d'arrière en multisupport	9
DaaS du Push4Apps	9
SSL du web services	10
Test à la grande échelle	10
Push4Apps Framework	11
Introduction du projet Push4Apps	12
Environnement de travail	13
IDE	13
Serveur d'application	15
Base de données	15
Framework	16
Langage de programmation	17
Partie I Web services basiques	19
1.1 MCD de la Base de données MySql du Push4Apps	19
1.2 Méthode de Scrum	20
1.3 Web services	21
1.3.1 Web services basiques	21
1.3.2 Web services pour la gestion des utilisateurs	23
1.3.3 Web services pour la gestion des applications	24
1.3.4 Web services pour push	25
1.3.5 Web services pour les statistiques	28
1.3.6 Web services pour les historiques	29
1.3.7 Web services pour les appareilles mobiles	29
1.3.8 Web servicee pour les Contacts	30
1.3.9 Web servicee pour les SDK	30
1.4 Système à la consommation des pushes	30
1.5 Système des alertes à la consommation du Push4Apps	31
1.6 Statistiques à la consommation des clients du Push4Apps	31
Partie II Back office du Push4Apps	32
2.1 Template des pages du Bureau d'arrière	32
2.2 Login	33

2.3 Page Accueil	34
2.4 Page Formulaire pour l'ajoute d'agence	38
2.5 Page Formulaire pour l'ajoute de client	39
2.6 Page Formulaire pour l'ajoute d'application	40
2.7 Page Formulaire de push	41
2.8 Page Statistique pour l'application	42
2.9 Page Statistique pour les histoires de notification mobile	43
2.10 Page GéoLive pour géo-localiser les appareilles mobiles	44
2.11 Page Push2Persp pour notifier les appareilles mobiles spécifiques	44
Partie III MBaaS du Push4Apps	45
3.1 DTWeb Manager avec Push4Apps	45
3.2 Statistiques d'application	45
3.3 Prédiction du push	48
Partie IV SSL du Push4Apps	49
4.1 Attaques typiques au serveur	49
4.2 Authentification	49
4.3 Autorisation	49
4.4 Session client et token sécurisé	49
4.5 Réalisation des mécanismes à la sécurité par OAuth2	49
Partie V Optimisation du Push4Apps	51
5.1 Work-bench de performance du Push4Apps	51
5.2 Stratégies à l'optimisation des web services	51
5.3 Parallélisme et Répartition des charges dynamiques	52
Conclusion	53
Glossaire	54
Annexe	57
Annexe 1 Schéma du projet	57
Annexe 2 Schéma de base de données	58
Annexe 3 L'arbre des problèmes techniques	58
Annexe 4 L'arbre des solutions aux problèmes techniques	58
Annexe 5 Diagramme de Gantt réel du Push4Apps	60
Annexe 6 la carte heuristique à l'organisation des web service du Push4Apps	61
Annexe 7 programme pour envoyer le push et la classe Push	62
Annexe 8 fonction récursive pour la table des utilisateurs	63
Annexe 9 la page de Géolive	64
Annexe 10 la page de Push2Perso	64
Annexe 11 la table Push	65
Annexe 12 la table StatePush et la table StatePushOpened	65
Annexe 13 Statistique par point spécifique	66
Annexe 14 Calculatrice de la distance entre deux points GPS	66

Présentation de la société DTWeb



Figure 1 - Logo de la société DTWeb

La société **DTWeb** est créée en 2010 par Damien Ruiz et Thierry Gaufryau. Ce dernier a cédé ses parts à M. Vargues qui est, par conséquent, devenu le nouvel associé de M. Ruiz. Elle est installée dans la pépinière d'entreprises de Lempdes-sur-Allagnon (43), en plein cœur de l'Auvergne, l'agence DT Web & Apps s'est installée en juin 2012. Depuis le 1er aout 2014, l'agence a déménagé dans de nouveaux locaux, dans le Parc Technologique de La Pardieu à Clermont-Ferrand (63), dans le but de sa rapprocher de sa clientèle et ainsi pouvoir offrir de meilleurs services.

Elle consiste à la communication spécialisée dans la création de sites internet, de référencement, d'application mobile et de web service afin d'améliorer la visibilité des entreprises. Son secteur d'activité est la communication et les médias sur internet. Elle est décomposée en trois grandes parties: un pôle de formation, un pôle technique et un pôle commercial. Sur le domaine du développement, DTWeb consiste à la refonte et développement de site internet. Sur le domaine des applications mobiles, elle contribue au développement d'applications iPhone, Android natives ainsi la mise à disposition des outils statistique. Au référencement, DTWeb propose l'audit, la préconisation et la stratégie au référencement de site, **SEO**, **SEM** et le travail de notoriété. Le modèle d'affaires de DTWeb est que **B2B**.

Aujourd'hui, Elle a intégré des locaux dans le Parc Technologique de La Pardieu, à Clermont-Ferrand (63), pour être plus proche de sa clientèle. Elle possède également un vrai savoir-faire en matière de communication web et utilise de nombreux leviers pour améliorer la visibilité des entreprises :

- **Création d'applications mobiles Android et iPhone:** une activité historique et un savoir-faire qui se sont développés avec la réalisation d'une quinzaine d'applications mobiles.
- **Développement web:** grâce au savoir-faire de développeurs expérimentés, DT Web & Apps propose la création de sites vitrines WordPress, de boutiques en ligne Prestashop et de sites web sur mesure développés en PHP. elle propose également la création de sites web optimisés pour les mobiles.
- **Création graphique :** affiches, flyers, bannières, cartes de visites, et même stickers pour smartphone, des solutions sur mesure pour développer votre visibilité.
- **Formation professionnelle :** formez-vous à la création web, au référencement ou aux applications mobiles pour devenir autonome.

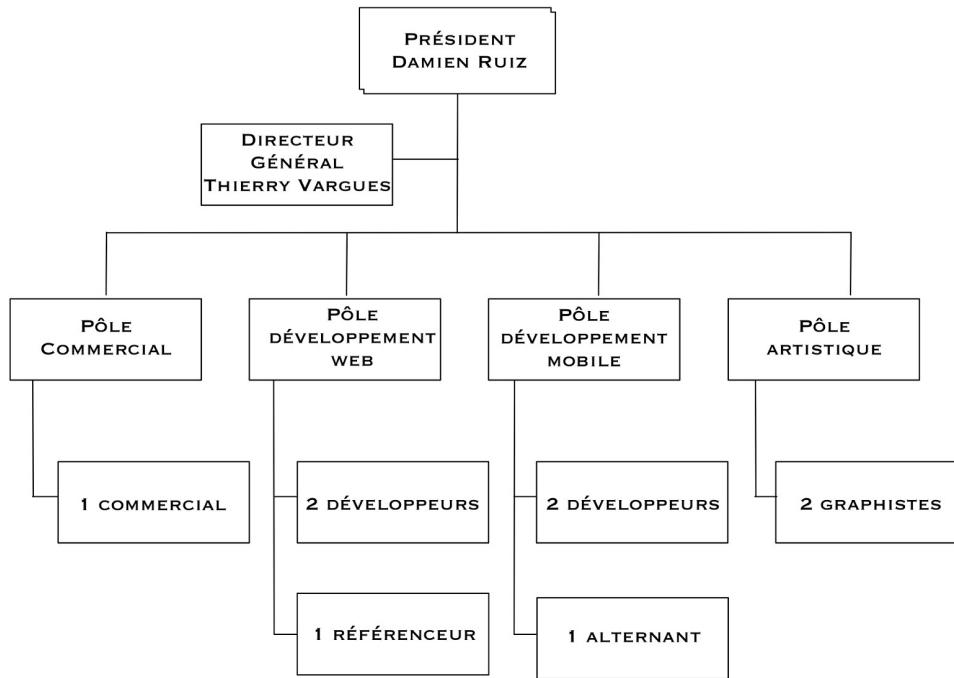


Figure 2 - Structure de l'entreprise DTWeb

Selon la figure du diagramme d'organisme, on peut trouver que la société DTWeb est dirigée par le président Damien RUIZ et le directeur général M Thierry VARGUES. Au Dessous, il y avait quatre pôles principales qui supporte la production de la société:

- Pôle **commercial** qui s'occupait la vente et la communication clientèle.
- Pôle du **développement web** qui s'occupaient la production des web site et des web sites.
- Pôle du **développement mobile** qui s'occupaient la production des applications mobiles.
- Pôle des **graphistes** qui s'occupaient le design de tous les interfaces d'utilisateur.

Problématique

Notification Push

Il existe plusieurs fournisseurs de services push pour envoyer des notifications aux applications installées sur différents appareils mobiles tel que **GCM** pour Android et **APNs** pour iOS. De plus, chaque fournisseur a développé sa propre interface et/ou librairie afin de simplifier l'appel du service push.

Réalisation des web services basiques

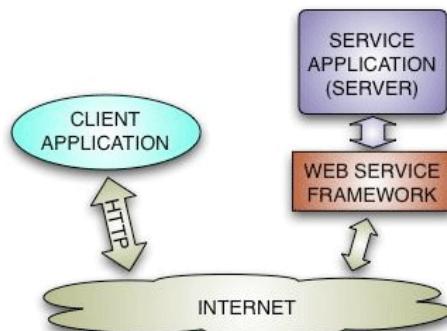


Figure 3 - Schéma de web service général RESTeasy

Afin de réaliser des fonctionnalités principales du Push4Apps comme expédier les notifications aux appareilles mobiles. D'abord, on devrait choisir **RESTful**. En suite, On a choisi le framework RESTeasy. Enfin, il devrait développer un serveur de web services qui peut supporter tout le protocole. Cet langage permet de réaliser des web services en protocole RESTful en Java orientés aux un ou plusieurs différents coté clientèle.

Bureau d'arrière en multisupport

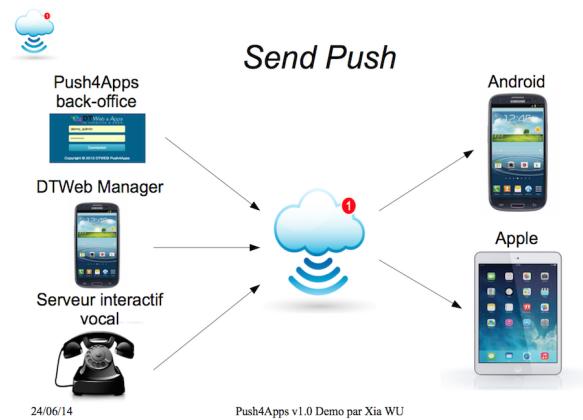


Figure 4 - schéma du web service “Send Push”(envoyer notification)

Un des plus important avantage du web service en RESTfull est l'adaptabilité et la versatilité. Le web service en RESTfull peut utilisé et interrogé par n'importe quel outil utilisant le protocole HTTP.

Un des objectifs principaux du push4apps est de permettre d'envoyer des notifications push à partir n'importe quel plateforme.

DaaS du Push4Apps

La récolte des données de push permet de réaliser des diagrammes statistiques. Ces derniers seront utilisés par la suite pour établir des études de marché vis-à-vis d'une application mobile.

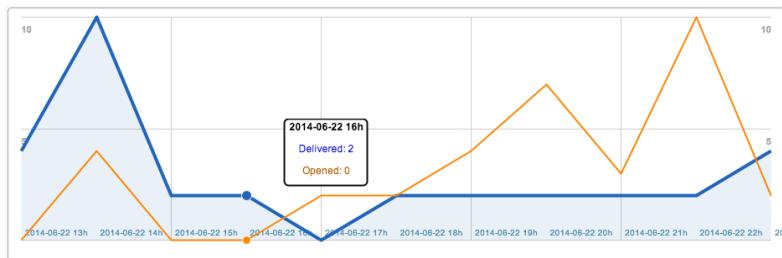


Figure 5 - Statistique par la date et par l'heure

Les données statistiques peuvent être interprétré par rapport à une durée ou une date, mais également en fonction d'une zone géographique.



Figure 6 - Statistique par les pays et par les villes

Par autre manière, le géo-statistique par les pays et par les villes nous permet de trouver le plus gros marketing de l'application.

SSL du web services

Après l'implémentation de Push4Apps, on arrive sur le niveau du **SSL** du serveur. Il reste à réaliser un mécanisme qui permet de sécuriser les données des web services.

Au niveau des appelles aux web services avec le protocole **RESTfull**, on doit masquer les données pour éviter toute menace d'intrusion et de piratage sur les web services. C'est dans ce but que l'encryptage des login et mots de passe a du être réalisé grâce à la méthode de hachage **SHA-256**.

SHA-256 est une méthode de encryptage populaire dans le domaine web parce qu'il est assez complexe pour protéger les données encryptées.

Par ailleurs, il faut associer l'appel des web services aux sessions clientes. Sinon, le serveur Push4Apps ne peut pas défendre l'attaque **DDoS** (attaque par déni de service).

DDoS : Une attaque par déni de service distribué qui sert à envoyer massivement demandes afin d'écraser le serveur.

Test à la grande échelle

Afin d'optimiser les performances du serveur à grande échelle, il a fallut utiliser un algorithme parallèle **MapReduce**, ce qui a permis d'alléger la charge de travail sur le serveur.

MapReduce sert à la distribution des tâches. Il permet d'accélérer la vitesse.

la procédure du MapReduce contient trois phases: La première phase, il distribue dynamiquement des tâches aux web services. La deuxième phase, il collecte des résultats des web services. La troisième phase, il fusionne les résultats.

L'avantage du MapReduce est qu'il passe à grande échelle. L'inconvénient est que la distribution des tâches et la collection des résultats.

Push4Apps Framework

Il existe plusieurs Framework pour faciliter l'implémentation des web services, comme WSDL, RESTeasy en Java, WSO2 en php et Eve en python. Selon le besoin du client, Push4Apps peut offrir des web service pour des notifications personnalisées.

Donc, On devrait compiler les urls des web services dans une librairie en donnant les macros utilisé par les développeurs de la notification mobile afin de simplifier ainsi de sécuriser les appelles des web service du Push4Apps. À la fois, on peut aussi générer les marcos en format XML qui permet d'utiliser le framework en n'importe quel langage de programmation ainsi un outil graphique à la génération des modèles BPM au Push4Apps.

Introduction du projet Push4Apps

Selon le schéma du projet([voir annexe 1](#)), le but principal du projet Push4apps est de diffuser des messages courts sur les appareils mobiles. D'abord, les clients de Push4apps peuvent entrer dans le back-office par son login et son mot de passe. Le serveur Push4Apps envoi le push par rapport au temps défini par le client. Dès la réception d'un push, le smartphone va retourner des données qui seront traités par la suite, afin de concevoir des diagrammes statistiques. Les statistiques concernent la géolocalisation et la date de réception et d'ouverture d'un push.

Le serveur Push4Apps présente les fonctionnalités suivantes :

- Envoyer les messages sur les appareils mobiles.
- Gérer des données statistiques.
- Fournir des web services pour la notification push.

On a mit des expressions fonctionnelles du besoin au cahier des charges:

J'ai pu énumérer les besoins au cachier des charges:

- Gérer des comptes utilisateurs.
- Envoyer des notifications push via une interface web.
- Afficher l'historique des pushes envoyés.
- Afficher les statistiques de push.
- Permettre d'utiliser le Push4Apps à travers différents types de plateforme.

J'ai proposé aussi des solutions pour répondre à ce besoin:

- Réaliser des services web pour l'administration des clients du Push4Apps serveur.
- Réaliser le service qui permet de créer et diffuser la notifications.
- Réaliser les services web pour l'analyse de l'efficacité de la diffusion de la notification selon l'histoire et les statistiques.
- Développer un bureau d'arrière pour l'administration du serveur Push4Apps.

J'ai également pris l'initiative de proposer des solutions d'optimisation pour répondre aux besoins du cahier des charges, dans le but de simplifier l'administration de l'interface web de Push4Apps.

Environnement de travail

IDE

Sur le domaine de la programmation informatique, un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels¹.

Eclipse Kepler



logo de l'eclipse Kepler

Eclipse Kepler est un projet consisté aux développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels **libre** qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java².

Netbeans



logo de l'eclipse Kepler

NetBeans est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).³

¹ http://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9

² [http://fr.wikipedia.org/wiki/Eclipse_\(projet\)](http://fr.wikipedia.org/wiki/Eclipse_(projet))

³ <http://fr.wikipedia.org/wiki/Netbeans>

SoapUi & LoadUI



logo de SoapUI

SoapUI est une application open source permettant le test de web service dans une architecture orientée services (SOA). Ses fonctionnalités incluent l'inspection des web service, l'invocation, le développement, la simulation, le mocking, les tests fonctionnels, les tests de charge et de conformité. Une version commerciale, *SoapUI Pro*, qui se concentre principalement sur des fonctionnalités conçues pour améliorer la productivité, a également été mise au point par eviware software. En 2011, eviware a été racheté par SmartBear Software.⁴

LoadUI



logo de LoadUI

LoadUI est un outil effectif consistant à lancer en temps réel des load testings à la performance des web services ou des pages web.⁵

Filezilla



logo de FileZilla

FileZilla est gestionnaire des sessions FTP qui permet l'**upload et le download** des fichiers sur un site, un serveur ou un hôte FTP, de sauvegarder en cache les arborescences pour une navigation plus rapide, et aussi de gérer les favoris.⁶

⁴ <http://fr.wikipedia.org/wiki/SoapUI>

⁵ <http://www.loadui.org/web-testing/testing-web-pages.html>

⁶ <http://filezilla.fr/>

Serveur d'application

JBoss EAP 6



logo de JBoss

JBoss Application Server (ou WildFly) est un serveur d'applications Java EE Libre écrit en **Java**, publié sous licence GNU LGPL. Étant écrit en Java, JBoss Application Server peut être utilisé sur tout système d'exploitation fournissant une machine virtuelle Java(JVM).⁷

Postman



logo de postman

Postman est un plug-in du web navigateur Chrome qui consiste à simplifier le développement avec les APIs des web services se basant sur le protocole HTTP.

Base de données

Mysql & Mysql workbench



logo de Mysql

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde.⁸

⁷ <http://fr.wikipedia.org/wiki/JBoss>

⁸ <http://fr.wikipedia.org/wiki/MySQL>



logo de Mysql workbench

MySQL Workbench (anciennement *MySQL administrator*) est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.⁹

Framework

Resteasy



logo du RESTeasy

RESTEasy est un projet du JBoss qui offre des frameworks consistés à simplifier le développeur des web services. Il utilise JAX-RS API qui offre des API au développement des web services RESTful sur le protocole HTTP.¹⁰

Fatfree



logo du framework fatfree

Cadre de Fat-Free est un framework d'application web open source distribué sous la GNU General Public License et hébergé par GitHub et Sourceforge. Son code source est presque entièrement écrit en PHP et conçus spécifiquement avec l'expérience de l'utilisateur et la facilité d'utilisation que ses objectifs principaux de la conception.¹¹

JQuery et Ajax



logo du JQuery

⁹ http://fr.wikipedia.org/wiki/MySQL_Workbench

¹⁰ <http://resteasy.jboss.org/>

¹¹ <http://fatfreeframework.com/home>

L'architecture informatique Ajax (acronyme d'Asynchronous JavaScript and XML) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

Ajax combine JavaScript, les CSS, XML, le DOM et le XMLHttpRequest afin d'améliorer maniabilité et confort d'utilisation des Applications Internet Riches.¹²

Font Awesome



logo du font Awesome

Font Awesome est une librairie CSS très populaire et ouverte des icons pour le développement des web site.¹³

Google Map & Google Geocoder



logo de géocodage du Google

Le "géocodage" consiste à affecter des coordonnées géographiques (longitude/latitude) à une adresse. Ce procédé conduit à la mise en place de traitement automatisés de manière ponctuelle ou sur des fichiers d'adresses (individus, entreprises, points d'intérêt, etc.). Les coordonnées géographiques permettent de positionner chaque adresse sur une carte numérique via un Système d'informations Géographiques (SIG). Le géocodage est une des techniques de géolocalisation ou de géoréférencement.¹⁴

Langage de programmation

Java



logo du java

¹² [http://fr.wikipedia.org/wiki/Ajax_\(informatique\)](http://fr.wikipedia.org/wiki/Ajax_(informatique))

¹³ <http://fortawesome.github.io/Font-Awesome/>

¹⁴ <http://fr.wikipedia.org/wiki/G%C3%A9ocodage>

Le langage **Java** est un langage de programmation informatique orienté objet.¹⁵

Php



logo du php

PHP: Hypertext Preprocessor, plus connu sous son sigle PHP(acronyme récursif), est un langage de programmation libre⁴ principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

JavaScript



logo du JavaScript

JavaScript (souvent abrégé JS) est un langage de programmation descriptif principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.¹⁶

CSS



logo du CSS 3.0

Les **feuilles de style en cascade**, généralement appelées **CSS** de l'anglais *Cascading Style Sheets*, forment un langage informatique qui décrit la présentation des documents HTML et XML. Il ne s'agit cependant pas de styles déterminés par l'auteur du document.¹⁷

¹⁵ [http://fr.wikipedia.org/wiki/Java_\(langage\)](http://fr.wikipedia.org/wiki/Java_(langage))

¹⁶ <http://fr.wikipedia.org/wiki/JavaScript>

¹⁷ http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

Partie I Web services basiques

Tous les web services sont réalisés et développés selon les fonctionnalités du Push4Apps. Ils permettent que le serveur Push4Apps puisse traiter les données stockées dans la base de données. Tous les clients du Push4Apps demandent les services à partir du côté client.

1.1 MCD de la Base de données MySql du Push4Apps

Le modèle MCD (**voir annexe 2**) est basé sur des spécifications fonctionnelles de Push4Apps. Il permet de visualiser toutes les relations entre les objets de la base de données.

La table Application est le noyau du MCD car il contient la plupart des fonctionnalités de la base de données pour l'utilisation des web services.

Un Utilisateur(User) peut avoir un ou plusieurs applications et a la possibilité d'envoyer un ou plusieurs push via une application.

Une application peut être installée sur plusieurs plateformes.(Androïd, iOS).

La table Push aussi est importante du fait qu'elle contient le web service pour envoi des notifications mobiles (Push) sur des appareils android ou iOS.

Un push ne peut être associé qu'à une seule application. Il peut référencer plusieurs données statistiques pour l'ouverture et la réception des pushes.

La table Device permet de référencer les numéros d'identification des différents types d'appareil. Dans la table, on trouvera aussi la "Latitude" et la "Longitude" qui permettent de trouver la position géographique d'utilisateur via GoogleMap.

La table d'utilisateur(User) sert à gérer tout le profil des utilisateurs . Il permet aussi de hiérarchiser les utilisateurs. L'utilisateur numéro 1 est l'administrateur(Superuser) du Push4apps, il peut surveiller tous les autres utilisateurs. Les agences mobiles sont tous référencées par leurs clients pour la réalisation des applications en utilisant "IdAgence".

En résumé l'administrateur peut avoir un ou plusieurs agences ou des clients d'agence. De plus, une agence mobile peut avoir un ou plusieurs clients d'agence. Tous type de client peut ajouter une ou plusieurs applications. Aussi tout client peut envoyer des notifications sur leurs applications. Les données statistiques sont référencées par le numéro d'identité du Push "IdPush".

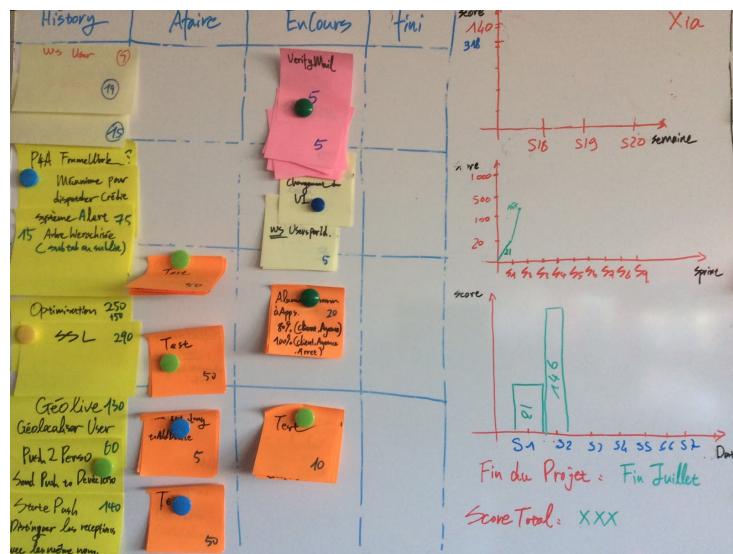
1.2 Méthode de Scrum

Il existe plusieurs méthodes Agile qui permettent d'optimiser la gestion de projet comme Waterfall, Kanban ou Scrum. Dans mon cas j'ai utilisé scrum afin de bien gérer mon projet. L'objectif est de décomposer le problème en tâches simples.

Dans un arbre de problèmes techniques(**voir annexe 3**), on a plusieurs tâches qui sont organisées par niveau d'importance , de priorité. Par exemple, il existe trois niveaux des tâches(Bleu: par important, Vert: normal, Orange: un peu important, Rouge: très important). Certaines tâches sont dépendantes entre elles c'est à dire une tâche ne peut pas commencer alors que des tâches sont en cours.

L'arbre aux solutions fonctionne (**voir annexe 4**) par le même principe que celui des tâches à solution, elle contient des tâches à solutions des différents niveaux hiérarchisées par le niveau de la difficulté à implémenter. Par exemple, le rouge signifie que cette tâche est difficile. Par contre, le bleu signifie que cette tâche est facile à implémenter. Les tâches à solution "Parente" sont divisées par plusieurs sous tâches à solution "Enfant".

Après la construire l'arbre à solution, on peut construire le diagramme de **Gantt** (**voir annexe 5**) afin de calculer et estimer le temps et le coût du développement des tâches à solution. De plus, il offre une vision globale de la progression du projet et une manière efficace à gérer des projets parallèles. Dans le diagramme de Gantt, on peut déterminer le parcours critique qui nous permet de savoir la durée minimale du projet. De plus, chaque tâche obtient un coût et un délai pour attirer l'attention des développeurs,pour augmenter l'efficacité de la gestions des projets parallèles.



Selon, le diagramme de Gantt, on pourrait éventuellement décider quelles tâches à solution seront terminées et pendant quelle période(**Sprint** pour la méthode Scrum). Sachant le tableau **Kanban** qui est inventé par la compagnie automobile Toyota comme une méthode très légère et effective afin de classifier les

tâches en différents états. Une histoire est comme une tâche “Parent” qui est composée un ou plusieurs tâches “Enfant”. La colonne “À faire” contient des tâches en état de préparation. De plus, La colonne “En cours” signifie les tâches courantes et la colonne “fini” contient des tâches qui sont terminées et produites.

De l'autre coté, on a des diagrammes statistiques concernant la quantité et la volume des tâches réalisées dans un Sprint. J'utilise normalement trois diagramme qui présentent différents types de statistiques pour bien contrôler l'avancement et l'efficacité à la production des tâches à réaliser.

Le première diagramme montre l'efficacité à la réalisation des tâches par semaine. Le deuxième diagramme montre l'efficacité à la réalisation des tâches par Sprint. Le troisièmement diagramme présente le statistique de l'efficacité à la réalisation des tâches par mois.

1.3 Web services

Modèle Général

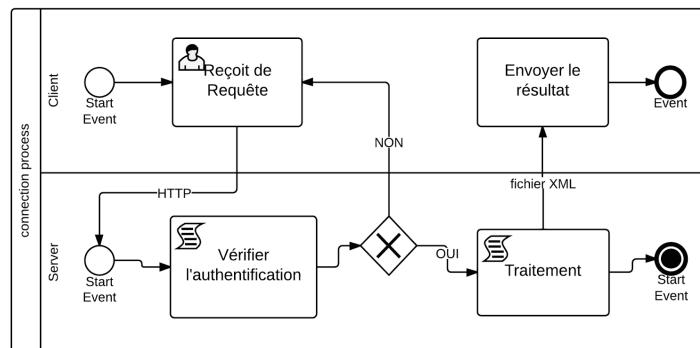


Figure 8 - le modèle général de web service

Pour mieux comprendre la procédure de web service, il faut dessiner un modèle **BPMN** ou un modèle de **Workflow** pour chaque web service. Mais, il existait un modèle général qui représente la procédure de workflow de web service.

Dans ce modèle, la procédure de web service peut décomposé en quatre tâches essentielles:

- La réception à demande de web service qui permet d'écouter et recevoir les demandes de web service
- La vérification des pré conditions de web service permet de vérifier les paramètres de web service
- Le traitement de web service en serveur.
- L'envoie de résultats en format JSON ou XML de web service qui permet de retourner les résultats de web service en forme standard qui peuvent être traiter par tous types d'utilisateurs.

1.3.1 Web services basiques

Selon la carte heuristique (voir annexe 6) à l'organisation des web service du Push4Apps.

1.3.1.1 Authentification

C'est un web service très important qui permet d'authentifier chaque demande de n'importe quel utilisateur. En fait, il est simple à réaliser avec un script MySQL avec une paire de paramètres (Login, Password). Il cherche dans la table User et retourne 1 si l'utilisateur est trouvé ou 0 sinon.

Si chaque demande de web service doit attendre la réponse après une recherche complète dans la base avec un million d'utilisateurs inscrits dans le Push4Apps cela prend beaucoup de temps.

Donc, j'ai trouvé une idée inspirée par un collègue il est un développeur du web en utilisant php et javascript. C'est beaucoup mieux de réaliser ce mécanisme qui permet une gestion des sessions des utilisateurs en utilisant un token à la place d'un login et password. Un token est un ticket qui est généré par le login, le password et le temps du moment où l'utilisateur se connecte au Push4Apps. De plus, comme tout ticket temporaire, il sera expiré dans un temps qui peut être décidé par l'utilisateur.

1.3.1.2 Autorisation

C'est un web service qui définit les groupes d'utilisateurs qui peuvent avoir des droits différents pour accéder à différents types de web service. Dans le cas normal, nous avons trois niveaux d'utilisateurs:

- Super-User (Administrateur qui peut utiliser tous les web services ainsi que tous les autres utilisateurs)
- Manager (il peut gérer les web services ainsi que les NormalUser)
- NormalUser (il peut seulement utiliser les web services fournis par le SuperUser ou le Manager)

1.3.1.3 E2F_DateFormatter

Ce web service permet de reformer les temps envoyés à partir des clients du Push4App. Par exemple, un utilisateur français envoie une date à terme "22-06-2014 22:22:22" au Push4Apps. Ensuite, il doit se transformer en forme anglaise avant d'entrer dans les web services du Push4Apps.

1.3.1.4 ReturnRes

Ce web service permet de reformer les résultats des web services en **JSON** ou **JSONArray**. Parce que JSON est inter-opérable par tous les types de client.

1.3.1.5 SignIN

Ce web service permet de se connecter et créer des sessions en utilisant le login et le password. Ainsi on n'aura plus besoin d'envoyer le login et le password à chaque fois qu'on veut utiliser les web services. L'utilisateur recevra un token de sécurité avec une durée valable après le login de l'utilisateur. Donc, l'utilisateur enverra son token de sécurité à la place de son login et son mot de passe.

Autrement, Si l'utilisateur entre le Push4Apps par le Back-Office. Le Push4Apps lui créera une session en notant toutes ses coordonnées au moment de la connexion, comme la latitude et la longitude pour la géolocalisation et le type d'utilisateur, etc. De plus, au niveau de la sécurité, on doit ajouter la date d'expiration de session aux web services à la gestion des clients. Parce qu'on doit défendre contre l'attaque **DDoS**.

1.3.1.6 SignOUT

Ce web service permet de sortir du Push4App en nettoyant la session d'utilisateur et son token de sécurité. Ainsi personne ne pourra utiliser Push4Apps avec son ancien token.¹⁸ De plus, le mécanisme à l'expiration de session appellera automatiquement cet web service à chaque deux heures à partir de la dernière connexion.

1.3.2 Web services pour la gestion des utilisateurs

1.3.2.1 ActiveUser

Ce web service permet d'activer quelques utilisateurs en changeant le statut de l'utilisateur comme "IsActive". Donc l'utilisateur ne peut pas se connecter au Push4Apps si son statut "IsActive" égale à 0, sinon, oui.

1.3.2.2 DeactiveUser

Ce web service permet de l'administrateur déactive un compte agence ou un compte utilisateur. Donc l'utilisateur qui es déactive par son agence ou l'administrateur ne peut pas se connecter la prochaine fois du Push4Apps.¹⁹ Les web services "ActiveUser" et "DeactiveUser" sont utilisés par l'adminstrateur ou les agences afin d'activer ou déactiver les comptes de leurs clients. De plus, ils peuvent aussi être utilisés par les programmes comme "système de consommation" qui permet d'activer ou de déactiver les comptes lorsqu'ils dépassent la limite de consommation.

1.3.2.3 AddUser

Ce web service permet d'ajouter d'inscrire au Push4Apps un utilisateur au Push4apps en donnant son login et son mot de passe etc.

1.3.2.4 CreditUser

Ce web service permet d'augmenter le crédit du compte à utilisateur du Push4Apps. Et le crédit est un sorti d'argent qui supporte la transaction d'envoi du push. Parce que l'utilisateur ne peut pas envoyer push si son crédit égale à 0.

1.3.2.5 DebitUser

Ce web service sert à débiter le compte à utilisateur du Push4Apps. Par conséquence, l'utilisateur ne peut pas envoyer un push lorsque son crédit du compte égale à 0.

1.3.2.6 GetUserById

Ce web service sert à retourner le profile d'utilisateur par son numéro d'identité qui contient tous les informations d'utilisateur. Il est normalement utilisé par les programmes du bureau d'arrière pour obtenir un JSONObject un structure des données concernant les informations de client.

¹⁸ UPDATE User SET IsConnected = NULL, SecuToken = NULL WHERE SecuToken = ?

¹⁹ UPDATE User SET IsActive = 0 WHERE IdUser = ?

1.3.2.7 GetUserCreditByAppli

Ce web service sert à retourner le crédit du propriétaire d'application en demandant le numéro d'identité d'applications. Il permet d'obtenir le crédit mensuel de l'application décidé par son propriétaire.

1.3.2.8 GetUsers

Ce web service sert à retourner une liste des utilisateurs concerné par son numéro d'identité comme ses agences et ses clients.

Ce web service permet que des agence ou l'administrateur peuvent obtenir tout l'information de ses clients. Il dépend au type d'utilisateur. On retourne tout l'utilisateur Si le demandeur est l'administrateur du Push4Apps. On retourne la liste des utilisateurs crée par son propriétaire.

1.3.2.9 GetUserType

Ce web service permet de savoir le type d'utilisateur en demandant son numéro d'identité. Le type d'utilisateur peut être détecté par son Id ainsi par son numéro d'agence. D'abord, si son numéro d'identité est égale à 1. Ensuite, si son numéro n'est pas 1 et il y a des numéros d'agence sont égal à son numéro d'identité.

1.3.2.10 ModifyUser

Ce web service permet de changer ou mise à jour le profile d'utilisateur. L'administrateur ou des agences peuvent changer le profile de son client.

1.3.2.11 VerifyMailUser

Ce web service sert à vérifier l'adresse mail de l'utilisateur. Il permet d'envoyer mail aux utilisateurs du Push4Apps afin d'attendre sa confirmation de réception pour vérifier son mail.

1.3.3 Web services pour la gestion des applications

1.3.3.1 AddApplication

Elle permet d'ajouter une application pour une agence ou un client. Il permet d'ajouter le profile d'application en spécifiant le nom d'applications, les autres informations selon les plate-formes mobiles etc.

1.3.3.2 ChangeEtatApplication

Ce web service permet d'activer ou de désactiver la capacité d'une application pour envoyer un push. Il permet d'activer ou de désactiver l'application par son propriétaire.

1.3.3.3 GetApplicationById

Ce web service retourne un objet d'application en utilisant son numéro d'identité IdAppli. On peut obtenir tout l'information de l'application par rapport son numéro.

1.3.3.4 GetApplications

Ce web service retourne tous les applications concernées qui contient tous les informations d'application. Il permet d'obtenir les applications selon le numéro de leur propriétaire.

1.3.3.5 ModifyApplication

Ce web service permet de modifier les informations d'application. Il nous permet de modifier l'application par son numéro.

1.3.4 Web services pour push

1.3.4.1 BookPush

Elle est une fonction ([voir annexe 7](#)) très importante qui nous permet d'envoyer push aux appareils Android et iOS. Cet web service permet de pouvoir réserver un push au temps décidé par les utilisateurs. J'ai implémenté un mécanisme temporaire, autrement un compteur du temps, qui permet de lancer un programme dans chaque 30 seconds.

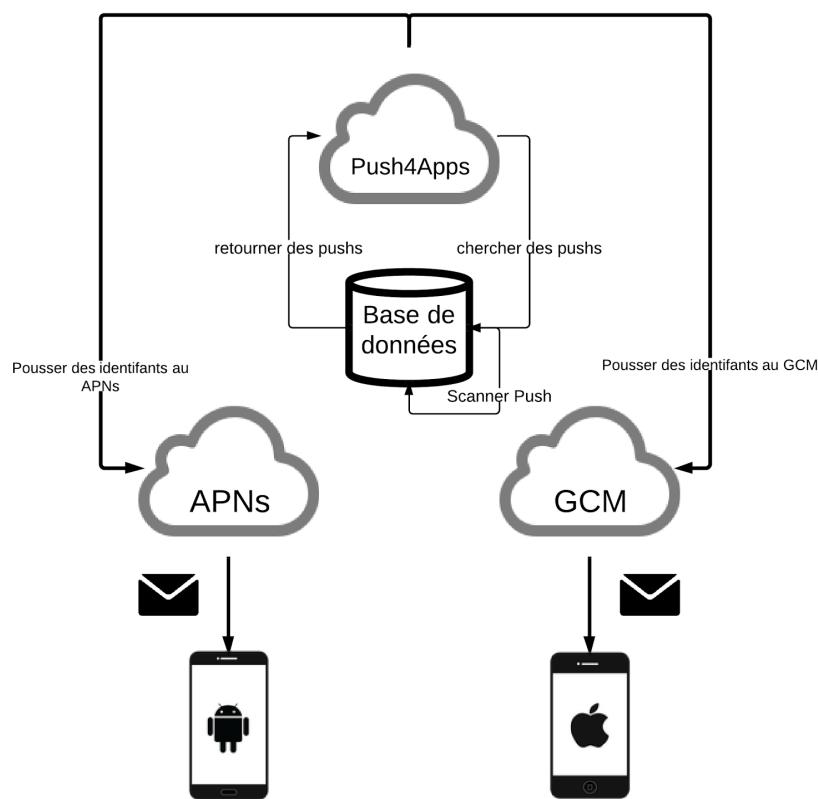


Figure 9 - le schéma du web service BookPush(réservation du push)

Dans la première phase, premièrement, il va scanner la base de données en cherchant tout le push qui ne sont pas encore envoyé dans une liste d'attente avec une liste des identifiants des appareilles associées à leurs parent (un push). Deuxièmement, chaque push possède un numéro d'application qui permet de trouver l'application associée. Ensuite, chaque appareil enregistré au Push4Apps possède un numéro d'application référençant l'application sur laquelle on enverra le push. De plus, le même appareil qui a installé différentes applications, il enregistre différentes identifiants au Push4Apps. Troisièmement, le serveur Push4Apps pousse l'objet push, il possède une liste d'identifiants des appareils, au **GCM** pour envoyer des notifications sur Android ou au **APNs** pour envoyer des notifications sur iOS.

Dans la deuxième phase, la gestion de mémoire serait un problème critique en risquerait la robuste du serveur Push4Apps lorsqu'il essaie d'envoyer parallèlement mille pushes sur des appareils dans le monde entier. Donc, la meilleure solution serait de bien repartir moyennement les réservations de push aux différents temps d'envoi (max 1000 pushes / 10mins).

1.3.4.2 DeletePush

Ce web service permet de supprimer un push lorsque l'utilisateur voulait supprimer son push avant son départ. Cet web service permet de supprimer les pushes réservés par les utilisateurs. De plus, il faut qu'ils ne soient pas encore envoyés.

1.3.4.3 GetPushById

Ce web service sert à retourner les informations de push en utilisant le numéro d'identité. Cet web service permet d'obtenir les informations du push par son numéro.

1.3.4.4 ModifyPush

Ce web service sert à modifier un push. Cet web service permet de modifier le push par son numéro en remplissant le formulaire de push.

1.3.4.5 RecvReponse

Ce web service permet de recevoir les coordonnées de l'appareil client comme la Géo-localisation lorsqu'un client reçoit le push.

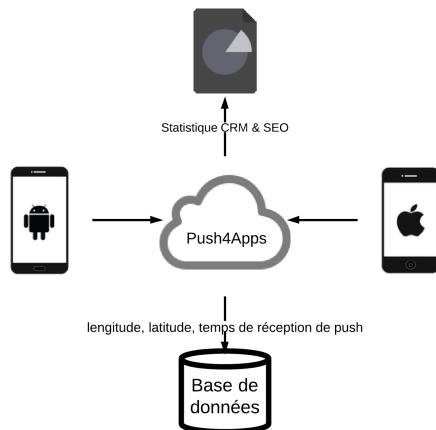


Figure 10 - la schéma du web service (RecvReponse)

En bref, Les appareils mobiles appellent cet web service lorsqu'ils auront reçu le push. De plus, ils retournent leurs longitude, leurs latitude, le nom de la ville, le nom du pays, la code postale à la géo-localisation, le temps GMT à la réception. Ensuite, le Push4Apps stocke les données dans la base de données. Enfin, des autres web services peuvent analyser des données afin de produire des statistiques au CRM et au SEO.

1.3.4.6 RecvReponseOpened

Ce web service permet de recevoir les coordonnées de l'appareil client comme la géo-localisation lors que client ouvert le push. Il est ensemble au service “RecvReponse” sauf qu'ils retournent la géo-localisation et le temps de l'ouverture de push.

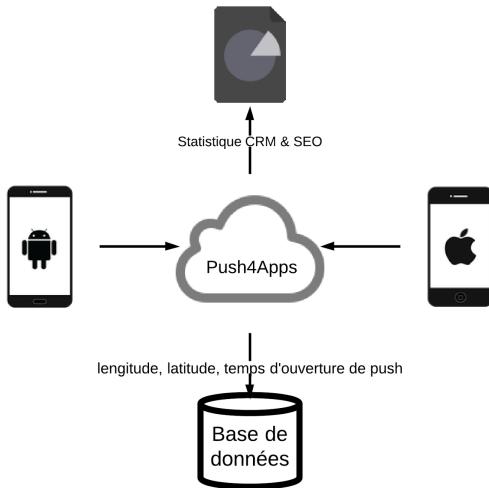


Figure 11 - la schéma du web service (RecvReponseOpened)

En bref, Les appareils mobiles appellent cet web service lorsqu'ils auront reçu le push. De plus, ils retournent leurs longitude, leurs latitude, le nom de la ville, le nom du pays, la code posta, le temps GMT à l'ouverture. Ensuite, le Push4Apps stocke les données dans la base de données.

1.3.5 Web services pour les statistiques

1.3.5.1 GetPushStatesByCountry

Ce web service sert à donner les informations statistiques classifiées par le nom du pays. Il permet de retourner les statistiques à la géolocalisation du push reçu envoyé en donnant le nom de la ville et le nom du pays.



Figure 12 - la schéma du web service (GetPushStatesByCountry)

D'abord, il cherche tout le données à l'ouverture du push. Ensuite, il sépare de les données par leurs noms de pays. Enfin, il envoie à la côté client un array des jsons nommé après le nom du pays qui contient des données.

1.3.5.2 GetPushStatesByCityCountry

Ce web service sert à recevoir les informations statistiques classifiées par le nom du pays et le nom de la ville. Il permet de retourner les statistiques à la géo-localisation du push envoyé en donnant le nom du pays et le nom de la ville. D'abord, il cherche tout le données à la réception du push. Ensuite, il sépare de les données par leurs noms de pays. Enfin, il envoie à la côté client un array des jsons nommé après le nom du pays et le nom de la ville qui contient des données.

1.3.5.3 GetPushStatesOpenedByCountry

Ce web service sert à retourner les informations statistiques de ldu push classifiées par le nom du pays. Il ensemble le web service “**GetPushStatesByCountry**” permet de retourner les statistiques à l'ouverture du push ouvert envoyé en donnant le nom du pays. D'abord, il cherche tout le données à l'ouverture du push. Ensuite, il sépare de les données par leurs noms de pays. Enfin, il envoie à la côté client un array des jsons nommé après le nom du pays qui contient des données.

1.3.5.4 GetPushStatesOpenedByCityCountry

Ce web service sert à retourner les informations statistiques de ldu push classifiées par le nom du pays et le nom de la ville. Il ensemble le web service “**GetPushStatesByCityCountry**” permet de retourner les statistiques à la géolocalisation du push ouvert envoyé en donnant le nom de la ville et le nom du pays.

1.5.5.5 GetPushStatesByHourDate

Ce web service sert à l'indiquer les informations statistiques classifiées par le nom du pays.²⁰ Il permet de retourner les statistiques au temps du push envoyé en donnant le temps de la date. Ce web service sert à retourner les informations statistiques de ldu push classifiés par l'heure et la date de réception. Il ensemble le web service “**GetPushStatesByCityCountry**” permet de retourner les statistiques à la géolocalisation du push ouvert envoyé en donnant l'heure et la date.

1.3.5.6 GetPushStatesOpenedByHourDate

Ce web service sert à retourner les informations statistiques de ldu push classifiées par l'heure et la date de la reception du push.²¹ Ce web service sert à l'indiquer les informations statistiques classifiées par le nom du pays.²² Il permet de retourner les statistiques au temps du push envoyé en donnant le temps de la date. Ce web service sert à retourner les informations statistiques de ldu push classifiés par l'heure et la date de réception. Il ensemble le web service “**GetPushStatesOpenedByCityCountry**” permet de retourner les statistiques à la géolocalisation du push ouvert envoyé en donnant l'heure et la date.

Ce web service permet de retourner les statistiques à la géolocalisation du push ouvert envoyé en donnant le temps et la date.

1.3.6 Web services pour les historiques

1.3.6.1 GetHisto

Ce web service sert à retourner une liste des histoires des push concernés à quelque application.²³Il permet de retourner les histoires du push envoyé concerné le numéro de l'application.

1.3.7 Web services pour les appareilles mobiles

1.3.7.1 AddDevice

Ce web service ajoute des informations d'appareille lorsque l'utilisateur se connecte au Push4Apps à parir de son appareille. Il permet d'ajouter les identifiant des appareilles mobiles dans la base de données du Push4Apps pour qu'on puisse y envoyer les pushes.

1.3.7.2 ChangeDeviceUdid

Ce web service permet de modifier et mise à jour le numéro de l'identification de l'appareille. Il permet de modifier l'identifiant des appareilles mobiles stocké dans la base de données.

²⁰ SELECT date(DeliveredTime) as DeliveredDate, hour(DeliveredTime) as DeliveredHour, count(*) as DeliveredNumber FROM StatePush WHERE IdPush = ? GROUP BY hour(DeliveredTime), date(DeliveredTime)

²¹ SELECT date(OpenedTime) as OpenedDate, hour(OpenedTime) as OpenedHour, count(*) as OpenedNumber FROM StatePushOpened WHERE IdPush = ? GROUP BY hour(OpenedTime), date(OpenedTime)

²² SELECT date(DeliveredTime) as DeliveredDate, hour(DeliveredTime) as DeliveredHour, count(*) as DeliveredNumber FROM StatePush WHERE IdPush = ? GROUP BY hour(DeliveredTime), date(DeliveredTime)

²³ SELECT push.* FROM Push push, Application app, User user WHERE user.IdUser = app.IdUser AND app.IdAppli = push.IdAppli AND user.IdUser = ? AND app.IdAppli = ? ORDER BY push.IdPush DESC

1.3.7.3 ChangeEtatDevice

Ce web service sert à activer ou déactiver la fonctionnalité pour envoyer ou recevoir la notification mobile.

1.3.7.4 VerifDevice

Ce web service sert à vérifier si l'appareil est déjà existant dans la base de données du Push4Apps. Il permet de vérifier si l'identifiant de l'appareil ajoutant est déjà dans la base de données.

1.3.8 Web service pour les Contacts

1.3.8.1 GetContact

Ce web service sert à retourner le contact d'agence ou les informations d'administrateur du Push4Apps. Il permet d'obtenir le contact d'utilisateur du Push4Apps selon son numéro d'identité.

1.3.9 Web service pour les SDK

1.3.9.1 AddSDK

Il permet d'ajouter les liens aux téléchargements des SDK au Push4Apps. SDK est un framework minimal pour simplifier leurs développements.

1.3.9.2 DeleteSDK

Il permet d'enlever ou supprimer un lien du sdk.

1.3.9.3 ModifySDK

Il permet d'enlever ou supprimer un lien du sdk.

1.4 Système à la consommation des pushes

Le système à la consommation mensuel des pushes sert à gérer et contrôler la nombre des pushes envoyés mensuel par des utilisateurs du Push4Apps. Il existe plusieurs règles qu'il fallait respecter en utilisant le Push4Apps:

- Crédit d'utilisateur est décidé par son agence.
- Crédit d'agence est décidé par l'administrateur du Push4apps.
- Crédit d'application est décidé par son agence.
- Utilisateur ne peut plus envoyer le push lorsque son crédit est terminé.
- Utilisateur ne peut plus envoyer le push lorsque le crédit de son agence est terminé.
- Des applications ont leurs dépassements qui permettent de dépasser leurs crédit mensuel. De plus, l'application sera bloquée lorsqu'elle sera dépassée sa limite.

- Des agences ont aussi leurs dépassements qui permettent de dépasser leur crédit mensuel. De plus, tous les applications seront bloquées lorsqu'il dépasse sa limite.
- Des utilisateurs ne peuvent pas envoyer plus de pushes que sa limite mensuelle ainsi la limite de son agence et la limite de l'application.
- Donc, avant d'envoyer les pushes, il fallait vérifier la consommation d'expéditeur, la consommation d'application et la consommation de son agence.

1.5 Système des alertes à la consommation du Push4Apps

Le système des alertes permet d'envoyer des mails concernant la consommation des pushes à l'utilisateur du Push4Apps pour qu'ils seront informés le dépassement de sa consommation mensuelle. De plus, la consommation change seulement après l'envoi des pushes. Et la consommation mensuel sera remise à 0 après la fin de chaque mois. L'utilisateur peut décider le nombre préféré des pushes et le temps à l'envoi des pushes.

Donc, il marche selon des règles suivantes:

- Client reçoit un mail qui indique que 80% de son crédit mensuel à l'application x est utilisé si il a utilisé 80% de son crédit mensuel à l'application x.
- Agence reçoit un mail qui indique que 100% de la consommation mensuelle de son client x est utilisée si son client x a utilisé 100% de son crédit mensuel.
- Agence reçoit un mail qui indique que la limite de la consommation mensuelle de l'application x est dépassée si son client x a utilisé 100% et plus son dépassement de son crédit mensuel à l'application x.
- Administrateur reçoit un mail qui indique que 100% du crédit mensuel de son client x est utilisé si son agence x a utilisé 100% de son crédit mensuel.
- Administrateur reçoit un mail qui indique que la limite du crédit mensuel de son client x est dépassée si son agence x a utilisé 100% de son crédit et plus son dépassement à la consommation mensuelle.

1.6 Statistiques à la consommation des clients du Push4Apps

D'abord, si la nombre des pushes préférés par l'utilisateur égale à 0. Le système à la consommation va choisir la nombre maximale des pushes. Ensuite, si la nombre des pushes préférés par l'utilisateur supérieur à le nombre des appareils enregistrés au Push4Apps. Le mécanisme du système à la consommation va choisir la nombre des appareils enregistrés. Enfin, SI la nombre des pushes préférés par l'utilisateur supérieur à son crédit mensuel restant. Le mécanisme du système à la consommation va choisir son crédit mensuel restant.

Partie II Back office du Push4Apps

2.1 Template des pages du Bureau d'arrière

2.1.1 Objective

L'objectif du bureau d'arrière est de faciliter la gestion du Push4Apps. De plus, il permet aussi de visualiser les statistiques aux histoires des notifications mobiles (push).

2.1.2 Framework Php

On a choisi le "Fatfree" Php framework comme l'infrastructure du web site. De plus, j'ai aussi choisi la combinaison entre le framework **JQuery 1.9** et la framework **Bootstrap** de Javascript qui permet de réaliser tous les effets dynamiques des pages.

2.1.3 Template

Tout le page du bureau d'arrière du Push4Apps contient trois parties:

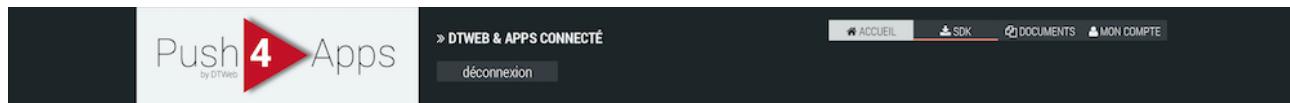


Figure 13 - la tête de la template Push4Apps

- Header:** l'en-tête de page qui permet d'afficher le logo du Push4Apps et le nom d'utilisateur. De plus, il permet aussi de partir du page actuel et accéder à l'autre page ainsi la déconnexion du Push4Apps.

NOM	CONSOMMATION	DÉPASSEMENT	ÉTAT	ACTION	PUSH
Agence 1	1 500 / 10 000	100 %			
> Client 1					
> Client 2					
App 1	150 / 800	non			
App 2	250 / 500	illimité			
App 3	550 / 500	10 %			
> Client 3					
> Client 4					
Agence 2	5 500 / 7 500	non			
> Agence 3	12 000 / 100 000	illimité			

Figure 13 - le corps de la template Push4Apps

- Corps:** le corps de page permet d'afficher le tableau des utilisateurs, leurs applications et leurs statistiques des pushes.



Figure 14 - le pied de la template Push4Apps

- **Footer:** le pied de page qui permet d'afficher l'illustration des buttons et le contact d'agence.

2.2 Login

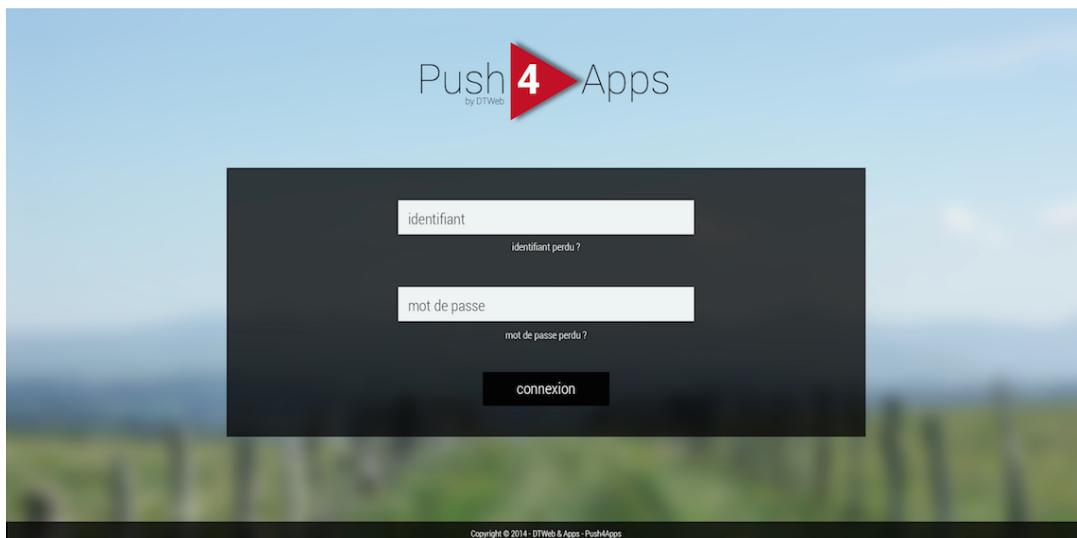


Figure 15 - la page de login du Push4Apps V1.0 après l'intégration visuelle

2.2.1 Analyse de la fonctionnalité

Il permet de se connecter au bureau d'arrière du Push4Apps en entrant son login et son mot de passe. De plus, on donnerait aussi la raison à refuse de connexion.

2.2.2 Analyse et estime du développement

Au niveau visuel, il faut d'abord changer le logo du Push4Apps. En suite, on devrait changer la forme et la position de footer(pied de page). De plus, on devrait aussi changer le couleur et la forme du dialogue de connexion. Enfin, on doit adapter l'image du fond de page aux écrans en différents tailles. Les point difficiles sont de changer de couleurs et de remettre le footer du page en bas du page.

Au niveau fonctionnel, il faut d'abord valider le login et le mot de passe en demandant le web service de connexion "SiginIN" au Push4Apps. De plus, il entrain si le login et le mot de passe sont correct. Sinon, on donnera la raison:

- Le login or le mot de passe n'est pas correct.
- L'agence a déactivé son compte.

Enfin, on devrait aussi réaliser la mécanisme à la perte d'identité et à la perte de mot de passe. Donc, le Push4Apps devrait aussi être capable de régénérer le login et le mot de passe au cas ou l'utilisateur oubliait son login ou son mot de passe. J'ai utilisé le Bcrypt. L'utilisateur recevra un message du Push4Apps lorsqu'il aura déclarer la perte de son mot de passe.

2.3 Page Accueil

The screenshot shows the Push4Apps administrator home page. At the top, there's a header with the logo "Push4Apps by DTWeb", a navigation bar with "ACCUEIL", "SDK", "DOCUMENTS", and "MON COMPTE", and a "déconnexion" button. Below the header is a search bar with "rechercher" and a filter button with "filtrer". There are also buttons for "ajouter une agence" and "ajouter un client". A dropdown menu shows filtering options: "Tout" (selected), "Agences", "Clients", "Applications", "Tous" (selected), "Actif", and "Inactif". The main content is a table with columns: NOM, CONSUMMATION, DÉPASSEMENT, ÉTAT, ACTION, and PUSH. The table data is as follows:

NOM	CONSUMMATION	DÉPASSEMENT	ÉTAT	ACTION	PUSH
Agence 1	1 500 / 10 000	100 %	✓ ■	✓ User icon	✓ Mobile icon
> Client 1			✓ ■	✓ User icon	✓ Mobile icon
> Client 2			✓ ■	✓ User icon	✓ Mobile icon
App 1 *	150 / 800	non	✓ ■	✓ User icon	✓ Mobile icon
App 2 *	250 / 500	illimité	✓ ■	✓ User icon	✓ Mobile icon
App 3 *	550 / 500	10 %	■ X	■ X	✓ Mobile icon
> Client 3			✓ ■	✓ User icon	✓ Mobile icon
> Client 4			■ X	■ X	✓ User icon
> Agence 2	5 500 / 7500	non	✓ ■	✓ User icon	✓ Mobile icon
> Agence 3	12 000 / 100 000	illimité	✓ ■	✓ User icon	✓ Mobile icon

At the bottom of the page, there are footer links: "Modifier", "Ajouter un client", "Ajouter une application", "Nouveau Push", and "Statistiques de l'application".

Figure 16 - la page d'accueil pour l'administrateur du Push4Apps après la modification

2.3.1 Analyse et estime du développement

Cet page permet d'administrer les agences et les utilisateurs du Push4Apps ainsi leurs applications. Donc il contient un table qui présente relativement les relations entre les utilisateurs et les applications en forme de la pyramide. Autrement, dans la liste des agences du Push4apps, on se trouvera la liste de leurs clients. De plus, dans la liste des clients de l'agence, on trouvera la liste des applications qui sont créées par l'agence et gérées par le client.

2.3.2 Changement visuel

Selon le template du Push4Apps, il contient trois parties majeures:

- Le header compose des éléments comme le logo, le nom d'utilisateur, le butons pour le page d'accueil, le butons de SDK, le documentation et le butons du profile d'utilisateur.
- Le corps contient la liste des agences et des utilisateurs qui permet de gérer leurs applications. De plus, l'hierarchie des relations entre les utilisateurs du Push4Apps se présente à la forme de la pyramide.
- Le footer présente l'illustration des buttons aux fonctionnalités et le contact d'utilisateur.

2.3.3 Changement fonctionnel

La liste des utilisateur est une table qui contient des sub tables. Autrement, chaque ligne présente un agence du Push4Apps qui contient un ou plusieurs clients. De plus, chaque clients contient aussi une liste des applications. Donc, on devrait aussi ajouter des butons qui servent à la gestion du Push4Apps. Pour chaque client et chaque application, il leur permet des comportements comme: ajouter, éditer et supprimer.

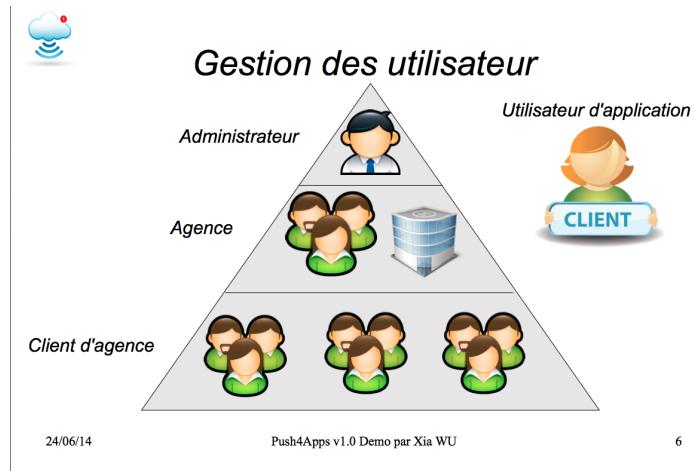


Figure 17 - la pyramide à la gestion des utilisateurs du Push4Apps

Selon la pyramide à l'hiérarchie pour la gestion des utilisateurs du Push4Apps, l'administrateur du système se trouve en haut. Donc, l'administrateur peut gérer tout le monde. De plus, il peut choisir à ajouter un agence ou un client. Ensuite, un agence peut ajouter seulement son client. Enfin, un client peut observer le statistique de ses client final des applications.

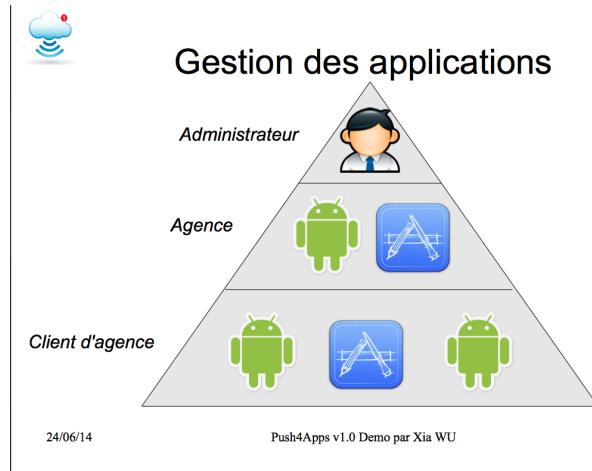


Figure 18 - la pyramide à la gestion des applications du Push4Apps

Selon la pyramide à l'hiérarchie pour la gestion des applications du Push4Apps, l'administrateur du système se trouve en haut. Donc, l'administrateur peut gérer toutes les applications. De plus, un agence peut ajouter sa propre application ou une application de son client.

Pour réaliser le tableau hiérarchisé qui contient trois niveaux: D'abord, l'administrateur du Push4Apps a des agences mobile. Ensuite, chaque agence a ses clients associés qui demande son agence mobile à développer ses applications. Enfin, chaque client possède des applications développées par son agences.

Donc, j'ai implémenté un web service général "**GetUsers**²⁴"(voir annexe 8) pour tout le type des utilisateur. Autrement, selon la hiérarchise des utilisateurs du Push4Apps, chaque table du résultat d'utilisateur au niveau dessous est une partie du résultat du web service de l'utilisateur au niveau dessus.

Donc, selon la vision du table à partie d'accueil à l'administrateur:

NOM	CONSOMMATION	DÉPASSEMENT	ÉTAT	ACTION	PUSH
Agence 1	1 500 / 10 000	100 %			
> Client 1					
> Client 2					
App 1	150 / 800	non			
App 2	250 / 500	illimité			
App 3	550 / 500	10 %			
> Client 3					
> Client 4					
> Agence 2	5 500 / 7 500	non			
> Agence 3	12 000 / 100 000	illimité			

Figure 19 - le table à la gestion pour l'administrateur du Push4Apps

On obtient le résultat du web service "GetUsers" en format JSONArray:

```
[
    // niveau d'administrateur
    {
        // niveau d'agence
        ...
        "IdAgence": "1",
        ...
        "Clients": [
            {
                // niveau de client
                "IdUser": "6",
                "UserApps": [
                    {
                        // niveau d'application
                        ...
                        "IdAppli": "21",
                        ...
                        // niveau d'application
                    }
                ],
                // niveau de client
                ...
            }
        ],
        // niveau d'agence
        ...
    }
    // niveau d'administrateur
]
```

Figure 20 - le structure de JSON reçue du web service GetUsers pour l'administrateur

selon la vision du table à partie d'accueil à un agence:

²⁴ un web service général récursif avec la détection du type d'utilisateur qui puisse parcourir effectivement la hiérarchie des utilisateurs.

NOM	CONSOMMATION	DÉPASSEMENT	ÉTAT	ACTION	PUSH
> Client 1					
Client 2					
App 1	150 / 800	non			
App 2	250 / 500	illimité			
App 3	550 / 500	10 %			
> Client 3					
> Client 4					

Figure 21 - le tableau à la gestion pour un agence du Push4Apps

On obtient le résultat du web service “GetUsers” en format JSONArray:

```
[
    // niveau d'agence

    "Clients": [
        {
            // niveau de client
            ...
            "IdUser": "6",
            ...
            "UserApps": [
                {
                    // niveau d'application
                    ...
                    "IdAppli": "21",
                    ...
                    // niveau d'application
                }
            ],
            // niveau de client
            ...
        },
        // niveau d'agence
    ]
]
```

Figure 22 - la structure de JSON reçue du web service GetUsers pour un agence

NOM	CONSOMMATION	DÉPASSEMENT	ÉTAT	ACTION	PUSH
App 1	150 / 800	non			
App 2	250 / 500	illimité			
App 3	550 / 500	10 %			

Figure 23 - le tableau à la gestion pour un client du Push4Apps

Selon la vision du tableau à partie d'accueil à un client d'agence, on obtient le résultat du web service “GetUsers” en format JSONArray:

```
[
    // niveau de client
]
```

```
{
    // niveau d'application
    ...
    "IdAppli": "21",
    ...
    // niveau d'application
}
// niveau de client
]
```

Figure 24 - la structure de JSON réçu du web service GetUsers pour un client

Enfin, l'implémentation du web service en RESTeasy déployée en JBOSS EAP 6.

2.4 Page Formulaire pour l'ajoute d'agence

Figure 25 - la page de formulaire pour l'ajoute d'agence du Push4Apps après la modification

2.4.1 Analyse et estime du développement

Dans la page de la formulaire pour l'agence, on doit réaliser la formulaire qui permet d'ajouter un agence à l'administrateur du Push4Apps.

2.4.2 Changement visuel

Pour réaliser cette page, je doit utiliser "CSS 3"²⁵ et la framework "Fatfree" pour implémenter des blocs du formulaire. De plus, j'ai utilisé le logiciel "Dreamwaver CS6"²⁶ afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments.

2.4.3 Changement fonctionnel

Il faut essayer d'adapter le nouveau design du formulaire en gardant bien les fonctionnalités d'avant. De plus, on devrait aussi ajouter certains nouveaux champs dans le nouveau formulaire. Par exemple, le dépassement du crédit mensuel et le numéro de téléphone.

2.5 Page Formulaire pour l'ajoute de client

The screenshot shows the Push4Apps interface with a dark header bar. The header includes the Push4Apps logo, the text '» AGENCE XCXCXCONNECTÉ', a 'déconnexion' button, and navigation links for 'ACCUEIL', 'SDK', 'DOCUMENTS', and 'MON COMPTE'. Below the header is a dark horizontal bar with the text 'AJOUT / MODIFICATION D'UN CLIENT'. The main content area contains two input sections. The first section, labeled 'identification', has three input fields: 'identifiant' (with placeholder 'identifiant'), 'mot de passe' (with placeholder 'mot de passe'), and 'répétez le mot de passe' (with placeholder 'répétez le mot de passe'). The second section, labeled 'contact', has four input fields: 'client' (with placeholder 'client'), 'nom du contact' (with placeholder 'nom du contact'), 'prénom du contact' (with placeholder 'prénom du contact'), and 'email' (with placeholder 'email'). At the bottom left is a 'créer/modifier le compte' button.

Figure 26 - la page de formulaire pour l'ajoute de client du Push4Apps après la modification

2.5.1 Analyse et estime du développement

Dans la page de la formulaire pour le client de l'agence, on doit réaliser la formulaire qui permet d'ajouter un client à son agence.

2.5.2 Changement visuel

²⁵ Les feuilles de style en cascade, généralement appelées CSS de l'[anglais Cascading Style Sheets](#), forment un [langage informatique](#) qui décrit la présentation des documents [HTML](#) et [XML](#). (http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade)

²⁶ Dreamweaver fut l'un des premiers éditeurs [HTML](#) de type « tel affichage, tel résultat », mais également l'un des premiers à intégrer un gestionnaire de site. (http://fr.wikipedia.org/wiki/Adobe_Dreamweaver)

J'ai utilisé le même design du formulaire "agence" en cachant certains éléments par html balise. De plus, j'ai changé les noms des champs afin d'adapter le web service "AddUser" du Push4Apps qui permet de créer un nouveau utilisateur.

2.5.3 Changement fonctionnel

Il faut essayer d'adapter le nouveau design du formulaire en gardant bien les fonctionnalités d'avant. De plus, on devrait aussi ajouter certains nouveaux champs dans le nouveau formulaire. Par exemple, le dépassement du crédit mensuel et le numéro de téléphone.

2.6 Page Formulaire pour l'ajoute d'application

The screenshot shows the Push4Apps interface for adding a new application. At the top, there's a navigation bar with the logo 'Push4Apps by DTWeb', the text 'AGENCE XCXCXCONNECTÉ', and links for 'ACCUEIL', 'SDK', 'DOCUMENTS', and 'MON COMPTE'. Below the navigation is a 'déconnexion' button. The main content area has a title 'AJOUT / MODIFICATION D'UNE APPLICATION'. It contains a form with the following fields:

- application pour le client XCXCXCONNECTÉ**: A text input field.
- nom**: An input field.
- crédit push mensuel**: An input field.
- dépassement mensuel**: A dropdown menu set to 10%.
- plateforme**: A section with radio buttons for 'Android' and 'iOS'.
- google api key**: An input field.
- certificat de développement** and **certificat de production**: Each with a 'parcourir' (choose) button.
- logo de l'application**: A section with a 'parcourir' button and a preview thumbnail.
- Créer/modifier l'application**: A large blue 'submit' button at the bottom of the form.

Figure 27 - la page de formulaire pour l'ajoute d'application du Push4Apps avant la modification

2.6.1 Analyse et estime du développement

Dans la page de la formulaire pour ajouter une nouvelle application, on doit réaliser la formulaire qui permet d'ajouter une application au Push4Apps.

2.6.2 Changement visuel

Pour réaliser cette page, je doit utiliser "CSS 3" et la framework "Fatfree" pour implémenter des blocs du formulaire. De plus, j'ai utilisé le logiciel "Dreamwaver CS6" afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments.

2.6.3 Changement fonctionnel

Il faut essayer d'adapter le nouveau design du formulaire en gardant bien les fonctionnalités d'avant. De plus, on devrait aussi ajouter certains nouveaux champs dans le nouveau formulaire. Par exemple, le dépassement du crédit mensuel et le logo d'application.

2.7 Page Formulaire de push

Figure 28 - la page de formulaire pour réserver après la modification

2.7.1 Analyse et estime du développement

Dans la page de la formulaire pour réserver un nouveau push (notification mobile). Donc on doit changer et ajouter des blocs des éléments pour réaliser la formulaire. De plus, j'ai ajouté une nouvelle fonctionnalité "push géo-localisé" plus que le concurrent Apps Panel au Push4Apps. Le push géo-localisé permet de géolocaliser tous les appareils mobiles ou des applications sont bien installées dessous. Ensuite, l'utilisateur du Push4apps donne un point zéro avec le longitude, le latitude et plus un rayon de la cercle de géo-localisation. Du coup, après la livraison du push, tout l'appareil mobile installé l'application va recevoir la notification mobile.

2.7.2 Changement visuel

Pour réaliser cette page, je doit utiliser “CSS 3” et la framework **Fatfree** pour implémenter des blocs du formulaire. De plus, j’ai utilisé le logiciel **Dreamwaver CS6** afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments. De plus, on devrait aussi ajouter une carte du monde sur la page qui permet de choisir le point zéro et décider le rayon pour la géo-localisation du push.

2.7.3 Changement fonctionnel

Il faut essayer d’adapter le nouveau design du formulaire en gardant bien les fonctionnalités d’avant. De plus, on devrait aussi ajouter certains nouveaux champs dans le nouveau formulaire. Ensuite, j’ai utilisé la librairie “Fatfree”, le Php et le Javascript ensemble pour réaliser des inter-actions du web page sans actualiser le page. L’API du googlemap est **Google Géocoder V3**.

2.8 Page Statistique pour l’application

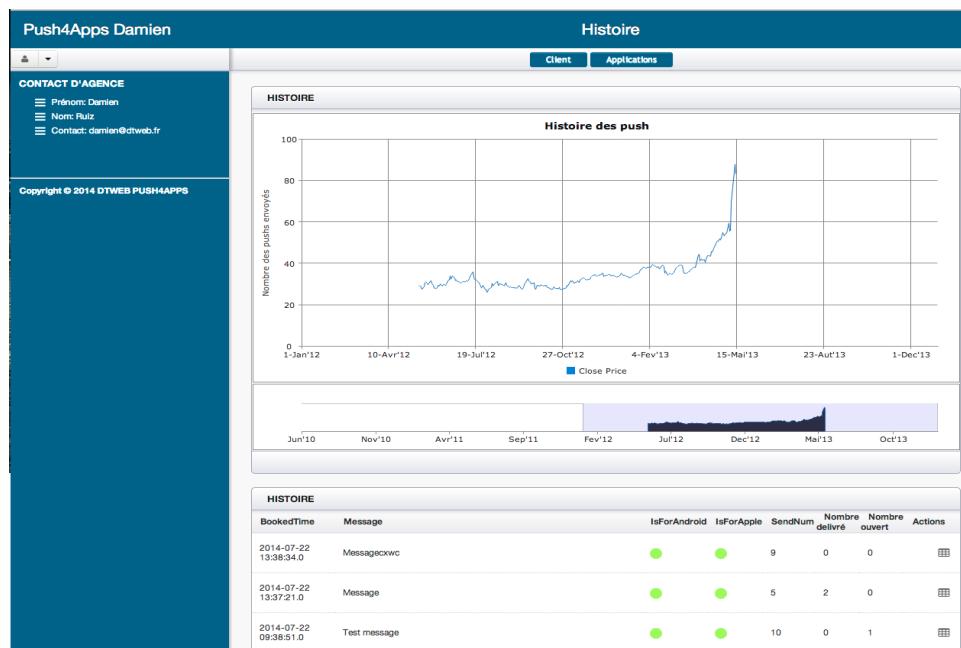


Figure 29 - la page statistique pour l’application de client du Push4Apps avant la modification

2.8.1 Analyse et estime du développement

Dans la page de l’histoire des push pour une application, on doit réaliser une liste des pushes qui permet d’avoir une vision globale ainsi qu’il puisse probablement prévenir le prochain push.

2.8.2 Changement visuel

Pour réaliser cette page, je doit utiliser “CSS 3” et la framework “Fatfree” pour implémenter des blocs du formulaire. De plus, j’ai utilisé le logiciel “Dreamwaver CS6” afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments.

2.8.3 Changement fonctionnel

Selon le design du page de l'histoire des push à l'application du Push4Apps, on devrait ajouter un diagramme du statistique à l'histoire des pushes envoyés. Dans ce diagramme, il y a un curseur bleu qui montait les nombres des pushes envoyés selon le temps. De plus, chaque point contient trois indices: la nombre rouge des pushes envoyés, la nombre bleue des pushes délivrés et la nombre jaune des pushes ouverts.

Dans la deuxième partie, on avait une liste des pushes envoyés, chaque ligne contient un button qui permet d'accéder à l'histoire spécifique à quelque push.

2.9 Page Statistique pour les histoires de notification mobile

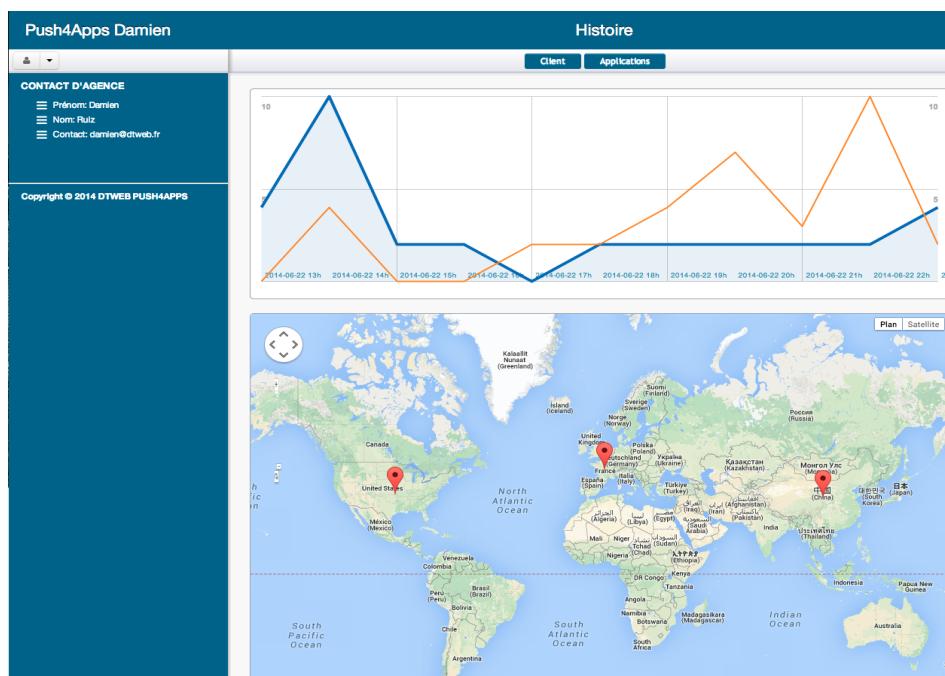


Figure 30 - la page statistique pour les histoires de push mobile du Push4Apps avant la modification

2.9.1 Analyse et estime du développement

Dans la page à l'histoire spécifique de quelque push, on avait deux statistiques importants. D'abord, le première diagramme aux statistiques. Ensuite, la carte du monde monte le géo-statistique du pushes qui monte le statistique des pushes envoyés par pays ou par villes .

2.9.2 Changement visuel

Pour réaliser cette page, je doit utiliser **CSS 3** et la framework **Fatfree** pour implémenter des blocs du formulaire. De plus, j'ai utilisé le logiciel **Dreamwaver CS6** afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments.

2.9.3 Changement fonctionnel

D'abord, j'utilise la librairie graphique Javascript **Amcharts** pour le diagramme du statistique au temps de livraison et au temps d'ouverture. Donc, on devrait appeler le web service GetPushState et le web service "GetPushStateOpened" pour obtenir les données des statistiques. Dans le diagramme au statistique du temps, on avait une courbe bleue qui donne la nombre de livraisons par heure et une autre courbe orange qui donne la nombre des ouvertures par heure.

Ensuite, la carte du monde donne le statistique aux géo-localisation des livraisons de push ainsi le statistique aux géo-localisation des ouvertures. Donc, on existe deux niveaux de statistique:

- Statistique au niveau des pays
- Statistique au niveau des villes

2.10 Page GéoLive pour géo-localiser les appareilles mobiles

Dans la page de Géolive([voir annexe 9](#)), on doit réaliser la formulaire qui permet d'afficher la géo-localisation d'utilisateur du Push4Apps. Pour réaliser cette page, je doit utiliser "CSS 3" et la framework "Fatfree" pour implémenter des blocs du formulaire. De plus, j'ai utilisé le logiciel "Dreamwaver CS6" afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments. Il faut essayer d'adapter le nouveau design du formulaire en gardant bien les fonctionnalités d'avant. De plus, on devrait aussi ajouter certains nouveaux champs dans le nouveau formulaire. Par exemple, le dépassement du crédit mensuel et le logo d'application.

2.11 Page Push2Persp pour notifier les appareilles mobiles spécifiques

Dans la page de Push2Perso([voir annexe 10](#)), on doit réaliser la formulaire qui permet d'envoyer des pushes sur quelque appareil spécifique. Pour réaliser cette page, je doit utiliser "CSS 3" et la framework "Fatfree" pour implémenter des blocs du formulaire. De plus, j'ai utilisé le logiciel "Dreamwaver CS6" afin de faciliter le design du page en laissant le logiciel calculer les positions des éléments. Il faut essayer d'adapter le nouveau design du formulaire en gardant bien les fonctionnalités d'avant. De plus, on devrait aussi ajouter certains nouveaux champs dans le nouveau formulaire.

Partie III MBaaS du Push4Apps

3.1 DTWeb Manager avec Push4Apps

J'ai participé un autre projet R&D en réalisant le développement de DTWeb manager. L'objectif de ce projet est d'aider les clients de DTWeb à bien suivre l'avancement de leurs projets développé.

En bref, DTWeb Manager est un système distribué en temps réel coupé en trois parties essentielles avec une base de données centralisée:

- La première partie du système est la gestion des statistiques d'avancement des projets en cours liés aux clients de DTWeb.
- La seconde partie du site web est le back-office de DTWeb Manager qui permet à l'administrateur d'observer et organiser les projets selon leur progression de développement.
- La dernière partie destinée aux supports mobiles permet de suivre la gestion des projets. Par conséquent, les clients de DTWeb peuvent facilement observer et suivre en temps réel les statistiques de production de leurs applications.

3.1.1 À quoi sert le MBaaS de Push4Apps?

Un MBaaS (Mobile Back-end As A Service) est une mélange entre Push4Apps et DTWebManager. Il permet à un client de suivre et gérer complètement la cycle de vie de son projet via un support mobile tel que iOS ou Android. Une intégration de Push4Apps a été faite dans DTWeb Manager. De plus, il offre un service CRM ou un service SEO au projet. Par exemple, après la recette du projet, un client peut envoyer des pushes via Push4Apps. Ensuite, il observe directement sur l'application DTWeb Manager les statistiques géographiques de réception de pushes par les clients.

3.2 Statistiques d'application

Push4Apps offre aussi plusieurs services de statistiques liés aux comportements de clients mobiles:

- L'historique des notifications envoyées en temps réels.
- Les statistiques pour les réceptions et les ouvertures de certains pushes spécifique à des pays, le tout visualisé sur la carte du monde.
- Les statistiques pour les réceptions et les ouvertures de certains pushes spécifique à des villes, le tout visualisé sur la carte du monde.

3.2.1 L'historique des notifications mobiles

Les statistiques de l'historique des pushes sont basées sur trois attributs « SendNum », « DeliveredNum » et « OpenedNum » de la table Push (**voir annexe 11**) de la base de données ainsi que sur le web service « GetHisto » qui nous permet de visualiser rapidement chaque notification²⁷.

²⁷ GetHisto mysql script: SELECT push.IdPush, push.IdAppli, push.Message, push.Rayon, push.BookedTime, push.FinishedTime, push.SendNum, push.DeliveredNum, push.OpenedNum, push.IsForAndroid, push.IsForApple FROM Push push, Application app, User user WHERE user.IdUser = app.IdUser AND app.IdAppli = push.IdAppli AND user.IdUser = ? AND app.AppliName = ? ORDER BY push.IdPush DESC



Figure 31 - la page statistique pour les histoires des notifications mobiles

Selon la figure 31, on peut voir une ligne bleue qui présente la nombre des pushs envoyés dans le temps. Ce diagramme de statistiques montre aussi à l'aide du nombre en vert le nombre de pushs reçus sur des appareils mobiles et le nombre des pushs ouverts par des utilisateurs en orange.

Ces statistiques permettent de voir globalement la consommation du service de notifications mobiles dans le temps. De plus, on peut prédire la consommation dans le futur.

3.2.2 Statistiques de push par la géolocalisation des pushs envoyés

Push4Apps offre des statistiques de géolocations des notifications mobiles. Il affiche deux niveau de statistiques : par villes ou par pays.

Les statistique de réceptions des pushs ([voir annexe 12](#)) est basé sur deux attributs « **RepLongLoc** », « **RepLatLoc** » de la table « **SendPush** » et deux attributs “**RepLongLoc**, **RepLatLoc**” de la table **StatePushOpened** de la base de données.

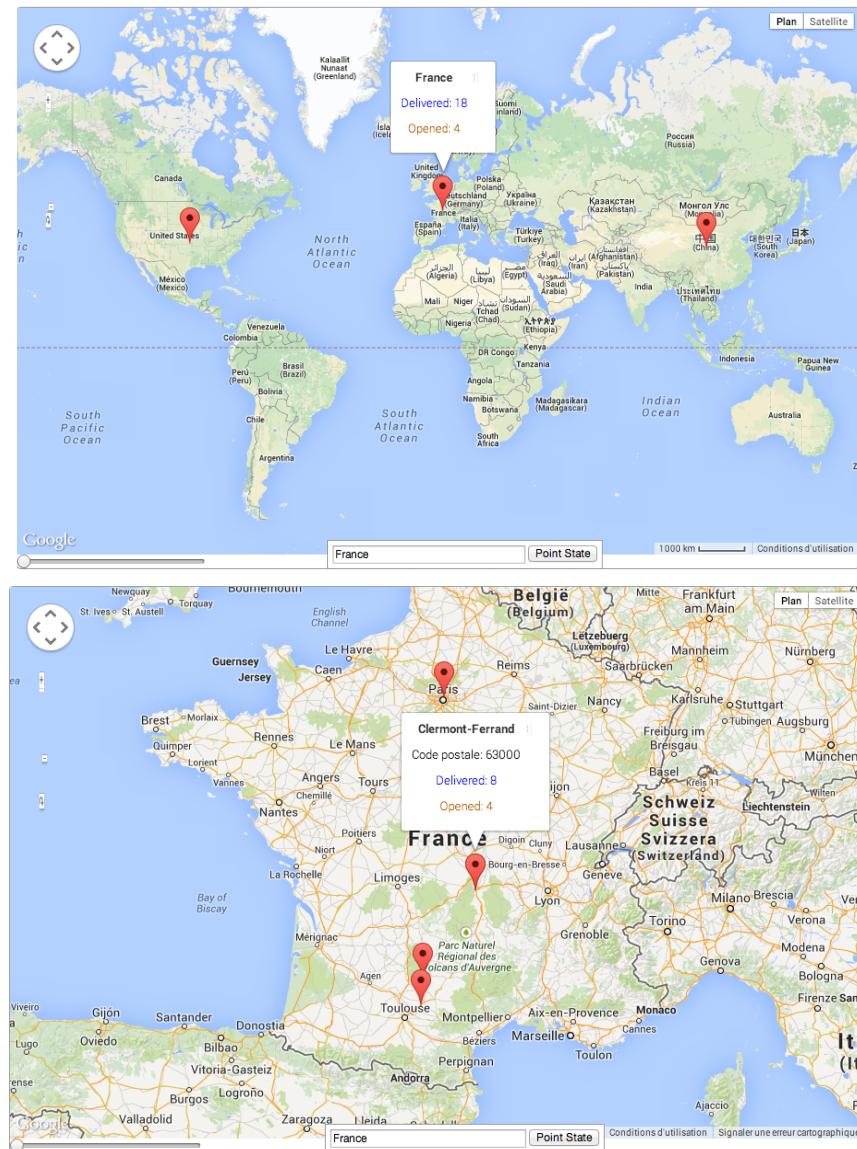


Figure 32 - la page statistique à la géolocalisation des notifications mobiles

Selon la première partie de la figure ci-dessus (voir figure 32), on affiche en premier les pays qui ont reçu le push. On peut voir qu'il a eu 18 réceptions et 4 ouverture de push en France. De plus, on peut descendre au niveau des villes en cliquant l'icone rouge de géolocalisation. Les statistiques sont basé sur le web service **GetPushStatesByCountry** et le web service **GetPushStatesOpenedByCountry**.

Selon la deuxième partie de cette figure, on observe qu'il a eu 8 réceptions de push et 4 ouverture de push à Clermont-Ferrand. De plus, ces statistiques sont basé sur les web service **GetPushStatesByCityCountry** et **GetPushStatesOpenedByCityCountry**.

3.2.3 Statistique de push par une région spécifique

On trouve qu'il a eu 8 réceptions de push et 4 ouverture de push à Clermont-Ferrand. De plus, le statistique est basé sur le web service **GetPushStatesByPoint**(voir annexe 13) qui nous permet de résumer

rapidement chaque notification par une **une calculatrice de distance GPS(voir annexe 14)** décidée par utilisateur.

3.3 Prédiction du push

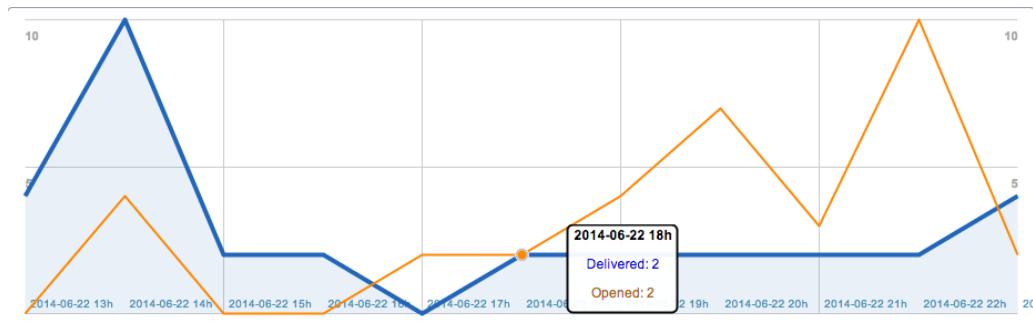


Figure 33 - la page statistique pour une région spécifique

Selon la figure 33, les statistiques de pushs récus et de pushs ouverts par rapport des heures, il est possible de prédire la meilleure heure pour envoyer des pushs. Le 22.06.2014 à 13h est l'heure où il y a eu le plus nombre de réception. Le 22.06.2014 à 21h est la meilleure heure d'ouverture de push.

Sachant que l'ouverture de push est le comportement qui intéressent le propriétaire de l'application, le premier critère de prédiction est l'heure où il y a eu la plus nombre d'ouverture. Mais, il y a eu toujours un écart de temps entre le moment où il y a la plus nombre de réceptions et le moment où il y a la plus grand nombre d'ouverture. Enfin, on pourrait prédire que le meilleur moment pour envoyer le prochain push seraiut probablement 20h.

Partie IV SSL du Push4Apps

4.1 Attaques typiques au serveur

On existe plusieurs façon d'attaquer un serveur de web services. La façon la plus effective et la plus dangereuse est **DDoS** qui permet d'envoyer massivement des demandes de web services jusqu'à la surcharge totale du serveur.

4.2 Authentification

L'authentification est le processus permettant de prouver l'identité de la personne faisant appel aux web services. L'identification est un facteur important dans Push4Apps. Les demandes de connexion sont acceptés ou rejetés en fonction des informations d'identification. Par exemple, le droit de créer des web services est réservé aux développeurs enregistrés et au propriétaire de l'application en question. En tant que développeur, ils envoient des demandes qui invoquent ces web services, ils auront donc besoin de prouver leurs identités via le système d'authentification du serveur.

4.3 Autorisation

Push4Apps permet de définir la liste des utilisateurs ayant accès à tel ou tel web service. L'autorisation peut être mis à disposition via **HTTPS** et le gestionnaire de sécurité permettra de valider les informations d'identification. L'autorisation et l'authentification sont des aspects clés de la gestion de l'identification dans les web services du Push4Apps.

4.4 Session client et token sécurisé

La méthode la plus simple de l'autorisation est d'obtenir un token de session lors de la connexion ou l'inscription et passer cette élément sur chaque demande. Donc, ce n'est évidemment pas suffisamment en sécurité pour être transmis sur autre chose que d'une connexion SSL. Mais, il peut être utile pour tester les services rapidement et facilement sans la surcharge de la signature de la demande à chaque fois.

4.5 Réalisation des mécanismes à la sécurité par OAuth2

4.5.1 OAuth2 et la gestion centralisée des identités

Quand un système grandit, le nombre demande d'accès augmente, les données des comptes des utilisateurs sont en croissance rapide. Par conséquent, les approches simples pour la sécurité vont commencer à devenir un problème pour la maintenance. Il faut donc mettre en place une approche centralisée de la gestion d'identité, et un service pour l'ensemble du système qui peut fournir et gérer les données de manière sécurisée. C'est là que les normes comme OAuth2 entrent en jeu. Ils introduisent une nouvelle couche dans l'architecture,

et une certaine complexité supplémentaire qui peut être difficile à prendre en main au début. Cependant les avantages seront en vaut la peine dans de nombreux cas.

4.5.2 Présentation rapide d'OAuth2

OAuth2 est un protocole permettant à une application client, souvent d'une application web, pour agir au nom d'un utilisateur, mais avec la permission de l'utilisateur. Les actions d'un client est autorisé à effectuer sont effectuées par un serveur de ressources (une autre application Web ou un service Web), et l'utilisateur approuve les mesures prises par dire à un serveur d'autorisation qu'il a confiance dans le client de faire ce qu'il demande. Les clients peuvent également agir comme eux-mêmes (pas pour le compte d'un utilisateur) s'ils sont autorisés à le faire par le serveur d'autorisation.

La façon la plus commune pour un client de se présenter à un serveur de ressources utilise un token de porteur, comme prévu dans les spécifications de base de OAuth2. Le token est obtenue à partir du serveur d'autorisation, avec l'approbation de l'utilisateur, si nécessaire, et conservé par le Client. Puis, quand il a besoin d'accéder à un serveur de ressources le client envoie un en-tête HTTP spéciale sous la forme JSONArray.

La valeur du token est opaque à un client, mais peut être décodé par un serveur de ressources pour qu'il puisse vérifier que le client et l'utilisateur ont la permission d'accéder à la ressource demandée. Des exemples courants de serveurs d'autorisation sur Internet sont Facebook, Google et Cloud Foundry tous qui fournissent également des serveurs de ressources (l'API Graph dans le cas de Facebook, les API Google dans le cas de Google, et le Cloud Controller dans le cas de cloud Foundry).

Partie V Optimisation du Push4Apps

5.1 Work-bench de performance du Push4Apps

SOAPUI permet de tester la qualité des web services déployés sur le Push4Apps. De plus, il peut aussi faire des tests unitaire, des tests fonctionnelles et des tests de robustesse afin de savoir si on pourra passer à la grande échelle.

5.2 Stratégies à l'optimisation des web services

5.2.1 Parallélisme des tâches aux web services

Le parallélisme des tâches est une forme de la parallélisation du code informatique sur plusieurs processeurs dans des environnements de calcul parallèle. Parallélisme Groupe se concentre sur la distribution de processus d'exécution (threads) dans les différents nœuds de calcul parallèle. Il oppose à parallélisme de données comme une autre forme de parallélisme.

Dans un système multiprocesseur, la tâche parallélisme est obtenu lorsque chaque processeur exécute un autre fil (ou procédé) sur les mêmes ou différentes données. Les fils peuvent exécuter la même ou différente de code. Dans le cas général, les différents fils d'exécution communiquent les uns avec les autres comme ils travaillent. Communication se déroule généralement par le passage de données d'un fil à l'autre dans le cadre d'un workflow.

5.2.2 Tâches distribuées aux web services

La distribution des tâches est un domaine de la science informatique qui étudie les systèmes distribués. Un système distribué est un système logiciel dans lequel les composants situés sur des ordinateurs en réseau de communiquer et de coordonner leurs actions en passant des messages. Les composants interagissent les uns avec les autres pour atteindre un objectif commun. Trois caractéristiques importantes des systèmes distribués sont: la simultanéité des composants, le manque d'une horloge mondiale, et l'échec indépendant de composants. Des exemples de systèmes distribués varient de systèmes RESTful ou SOA pour les jeux massivement multi-joueurs en ligne de pair-à-pair applications.

Un programme informatique qui s'exécute dans un système distribué est appelé un programme distribué, et la programmation distribuée est le processus d'écriture de ces programmes. Il existe de nombreuses alternatives pour le mécanisme de transmission de messages, y compris les connecteurs de RPC comme et les files de messages. Un objectif important et le défi des systèmes distribués est la transparence de localisation.

L'informatique distribuée se réfère également à l'utilisation de systèmes distribués pour résoudre des problèmes de calcul. Dans le calcul distribué, un problème est divisé en de nombreuses tâches dont chacune est résolu par un ou plusieurs ordinateurs, qui communiquent entre elles par passage de messages.

5.3 Parallélisme et Répartition des charges dynamiques

La technologie Web des services est devenu le tissu de capacité de connexion omniprésente parmi les domaines d'activité divers et camps techniques. Dans le même temps, le calcul parallèle distribué devient l'architecture de facto pour la gestion de la performance des programmes de calcul intensif, de longue durée.

Est-ce qu'on aura besoin de considérer les services Web lors de la poursuite de l'amélioration des performances, les applications en cours d'exécution à long calcul intensif? En fait, Il peut sembler de cette façon, mais, plus étonnamment, les services Web jouent un rôle critique pas dans un, mais dans deux domaines de l'informatique haute performance (HPC) et le calcul parallèle distribué:

- Communications / déploiement
- Classifications / services de découverte de ressources

En d'autres termes, les services Web jouent un rôle dans la couche d'adaptation d'application et d'infrastructure, respectivement. Assez sensiblement, les services Web offrent de nouveau sur la promesse de collaboration universelle sémantique et syntaxique et une large acceptation dans une autre technologie de passage à l'âge.

Conclusion

Ce stage a été une expérience enrichissante. Tout d'abord, il m'a permis de connaître les avantages du métier de développeur web au sein d'une entreprise. J'ai également acquis de nouvelles connaissances et compétences en découvrant le développement des web services en RESTful.

Mon premier objectif était de créer un ensemble de web services de Push4Apps qui sert à envoyer des notifications push aux applications mobiles associées à DTWeb. De plus, j'ai ajouté la fonctionnalité de la géo-localisation qui permet d'avoir une position(latitude, longitude) ainsi qu'un rayon autour des coordonnées GPS reçus d'un appareil.

Les objectifs de ce projet ont été remplis et le résultat obtenu était très satisfaisant. Les web services du Push4Apps sont exportables et compatibles avec plusieurs serveurs d'applications comme JBoss, Tomcat, Glassfish etc.

Au sein de la société DTWeb, j'étais considéré comme un vrai collaborateur à part entière et non comme un simple stagiaire. J'ai ainsi pu découvrir les responsabilités et difficultés que peuvent rencontrer les développeurs informatiques quotidiennement. De plus, j'ai appris à faire face à certains problèmes rencontrés et à les résoudre simplement, afin de ne pas perdre du temps. En effet, les entreprises ont des délais à respecter et des contraintes à prendre en considération même si elles sont négligeables. Il a fallut pour cela suivre les tâches planifiées du diagramme de Gantt.

La mise en place du Back-Office et l'intégration des nouveaux visuels est complexe à implémenter car il faut améliorer de temps en temps les web service selon les demandes des clients. Cependant, en analysant les statistiques de réceptions et d'ouvertures de push. J'ai réussi à implémenter ces web services en répondant aux besoins du cahier de charges.

Durant ce stage, j'ai utilisé Fatfree, JQuery, CSS 3 et Ajax afin de faciliter l'intégration visuelle du Push4Apps.

Ce projet a permis de répondre à des besoins concrets, à savoir l'étude de nouvelles technologies et optimisation des charges du serveur. L'entreprise a dorénavant à disposition un outil prêt à l'emploi.

Au final, je pense avoir beaucoup appris, tant au niveau technique qu'au niveau fonctionnel et personnel. J'espère en apprendre encore davantage sur les technologies du web. Je pense que cette expérience m'a beaucoup apporté en maturité et me sera très utile lors de mon entrée sur le marché du travail.

Glossaire

Adobe Dreamwaver CS6 est un des premiers éditeurs **HTML** de type « tel affichage, tel résultat », mais également l'un des premiers à intégrer un gestionnaire de site

Amcharts est une librairie à la visualisation des charts. (<http://www.amcharts.com/>)

APNs: Apple Push Notification service
(<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html>)

B2B: Affaire à affaire (http://fr.wikipedia.org/wiki/Business_to_business)

Bcrypt est un plugin sécurisé et une librairie à l'hachage de password
(http://fatfreeframework.com/bcrypt#needs_rehash)

BPM: Business process modeling (BPM) in systems engineering is the activity of representing processes of an enterprise, so that the current process may be analyzed or improved
(http://en.wikipedia.org/wiki/Business_process_modeling)

BPMN: Business Process Model and Notation est une notation graphique standardisée¹ pour modéliser des procédures d'entreprise ou des processus métier. La première version était connue sous le nom de *Business Process Modeling Notation*.(<http://fr.wikipedia.org/wiki/BPMN>)

Bootstrap, a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. (<http://getbootstrap.com/getting-started/>)

CRM: La gestion de la relation client (GRC ou CRM), connue aussi sous son nom français correct de gestion des relations avec les clients (GRC), et sous sa traduction anglaise Customer Relationship Management (La gestion de la relation client (GRC)) (http://fr.wikipedia.org/wiki/Gestion_de_la_relation_client)

Carte heuristique ou carte cognitive, carte mentale, carte des idées, est un schéma, supposé refléter le fonctionnement de la pensée, qui permet de représenter visuellement et de suivre le cheminement associatif de la pensée.(http://fr.wikipedia.org/wiki/Carte_heuristique)

CSS 3: Les **feuilles de style en cascade** (Cascading Style Sheets en anglais), forment un langage informatique qui décrit la présentation des documents HTML et XML.
(http://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade)

DaaS: Data as a Service, ou DaaS, est une technique consistant à faire payer un abonnement pour l'accès à un dépôt de données via une interface fournie par le fournisseur.
(http://fr.wikipedia.org/wiki/Data_as_a_Service)

DDoS: L'algorithme MD5, pour Message Digest 5, est une fonction de hachagecryptographique qui permet d'obtenir l'empreinte numérique d'un fichier (<http://en.wikipedia.org/wiki/MD5>)

Fatfree est un framework MVC en Php qui consiste à faciliter le développement du web site en php.(<http://fatfreeframework.com/home>)

GCM: Google Cloud Messaging for Android (<http://developer.android.com/google/gcm/index.html>)

Géocodage consiste à affecter des coordonnées géographiques (longitude/latitude) à une adresse. (<http://fr.wikipedia.org/wiki/G%C3%A9ocodage>)

Géolocalisation: La géolocalisation ou géoréférencement est un procédé permettant de positionner un objet (une personne, etc) sur un plan ou une carte à l'aide de ses coordonnées géographiques.

Gantt: Le diagramme de Gantt (*Harmonogram Adamieckiego*) est un outil utilisé (souvent en complément d'un réseau PERT) en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. (http://fr.wikipedia.org/wiki/Diagramme_de_Gantt)

HTTPS: L'*HyperText Transfer Protocol Secure* (http://fr.wikipedia.org/wiki/HyperText_Transfer_Protocol_Secure)

JSON: JavaScript Object Notation est un format de données textuelles, générique, dérivé de la notation des **objets** du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. (http://fr.wikipedia.org/wiki/JavaScript_Object_Notation)

jQuery est une bibliothèque JavaScript **libre** et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web (<http://jquery.com/>)

Kanban: est une simple fiche cartonnée que l'on fixe sur les bacs ou les conteneurs de pièces dans une ligne d'assemblage ou une zone de stockage. (<http://fr.wikipedia.org/wiki/Kanban>)

Lucidchart est un service d'informatique en nuage qui permet de travailler collaborativement afin de créer des (<https://www.lucidchart.com/>)

Méthode Agile sont des groupes de pratiques de projets de développement en informatique (conception de logiciel), pouvant s'appliquer à divers types de projets. (http://fr.wikipedia.org/wiki/M%C3%A9thode_agile)

Macros: En programmation informatique, une **macro-définition** ou simplement **macro** est l'association d'un texte de remplacement à un identificateur, tel que l'identificateur est remplacé par le texte dans tout usage ultérieur. (<http://fr.wikipedia.org/wiki/Macro-d%C3%A9finition>)

MBaaS: Bureau d'arrière mobile comme un service (Mobile Back-end as a Service) (http://en.wikipedia.org/wiki/Backend_as_a_service)

OAuth est un API libre pour sécuriser facilement des web services , créé par Blaine Cook et Chris Messina. Il permet d'autoriser un **site web** à utiliser l'API sécurisée d'un autre site web pour le compte d'un utilisateur. OAuth *n'est pas* un protocole d'authentification. (<http://fr.wikipedia.org/wiki/OAuth>)

Push serveur: Le Server Push est un mode de communication client-serveur dans lequel le dialogue est lancé par le serveur. (http://fr.wikipedia.org/wiki/Server_push)

Parallélisme consiste à couper une tâche en plusieurs sous tâches en les exécutant simultanément sur plusieurs processeurs. (http://en.wikipedia.org/wiki/Task_parallelism)

RESTful: REST (REpresentational State Transfer) est un style d'**architecture** pour les systèmes hypermédia distribués (http://fr.wikipedia.org/wiki/Representational_State_Transfer)

RESTeasy: un projet de JBoss qui offrit des frameworks afin d'aider le développement des web services RESTful en Java. (<http://resteasy.jboss.org>)

SEO: optimisation pour les moteurs de recherche
(http://fr.wikipedia.org/wiki/Optimisation_pour_les_moteurs_de_recherche)

SEM: Search Engine Marketing (http://fr.wikipedia.org/wiki/Search_Engine_Marketing)

SOAP: ancien acronyme de Simple Object Access Protocol) est un protocole deRPC orienté objet bâti sur XML. Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP.
(<http://fr.wikipedia.org/wiki/SOAP>)

SSL: Transport Layer Security (TLS), et son prédecesseur Secure Sockets Layer (SSL), sont des protocoles de sécurisation des échanges sur Internet(http://fr.wikipedia.org/wiki/Transport_Layer_Security)

SHA256: SHA-2 (*Secure Hash Algorithm*) est une famille de fonctions de hachage.
(<http://fr.wikipedia.org/wiki/SHA-2>)

Scrum est un framework d'organisation de développement de produits complexes.
([http://fr.wikipedia.org/wiki/Scrum_\(m%C3%A9thode\)](http://fr.wikipedia.org/wiki/Scrum_(m%C3%A9thode)))

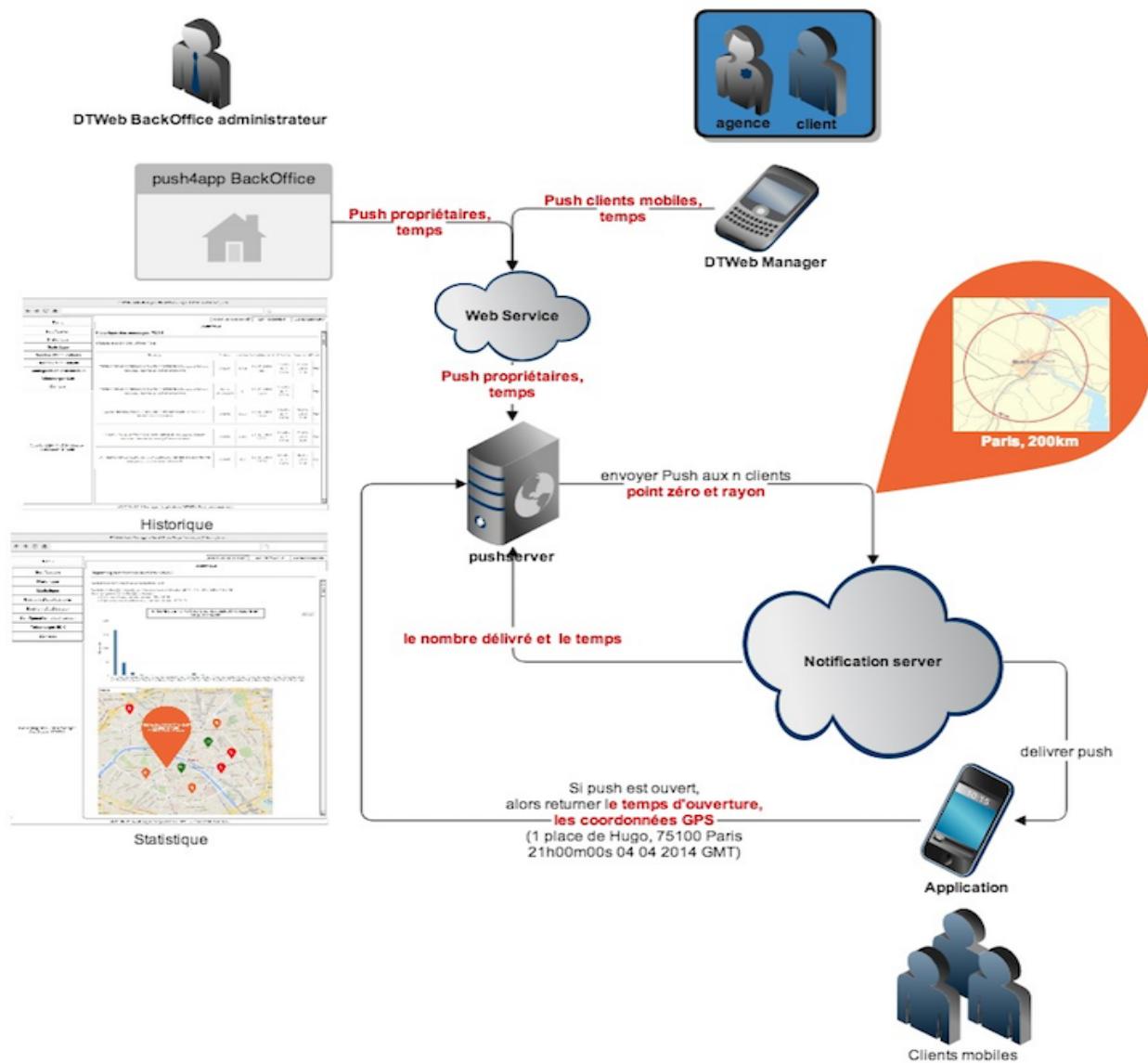
Sprint: Le sprint est une période d'un mois au maximum, au bout de laquelle l'équipe délivre un incrément du produit, potentiellement livrable. ([http://fr.wikipedia.org/wiki/Scrum_\(m%C3%A9thode\)#Le_sprint](http://fr.wikipedia.org/wiki/Scrum_(m%C3%A9thode)#Le_sprint))

SEO: L'optimisation pour les moteurs de recherche (en [anglais](#), *Search engine optimization : SEO*)
(<http://fr.wikipedia.org/wiki/SEO>)

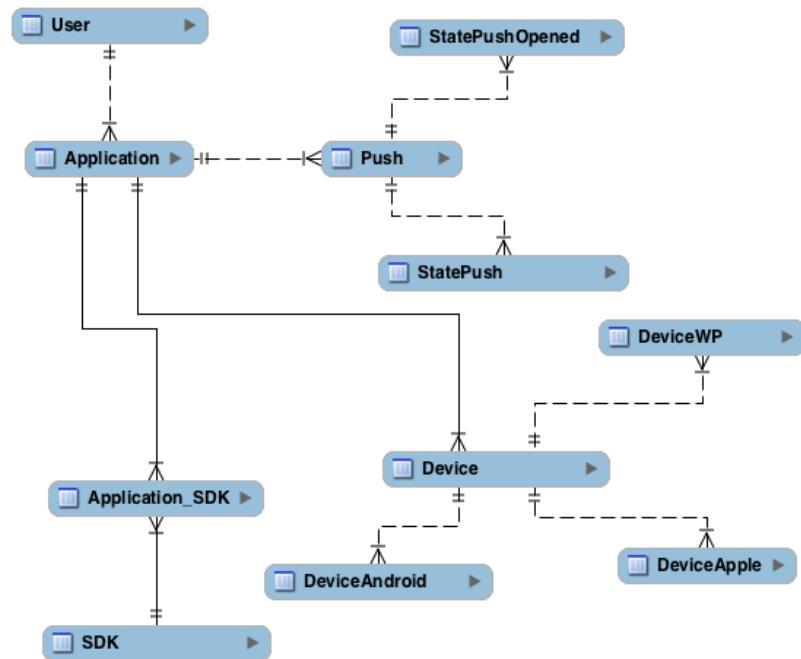
WSDL: Web Services Description Language est une grammaire XML permettant de décrire un service web. (<http://fr.wikipedia.org/wiki/WSDL>)

Workflow: anglicisme pour **flux de travaux**, est la représentation d'une suite de tâches ou opérations effectuées par une personne, un groupe de personnes, un organisme, etc. Le terme *flow* (flux) renvoie au passage du produit, du document, de l'information, etc., d'une étape à l'autre.
(<http://fr.wikipedia.org/wiki/Workflow>)

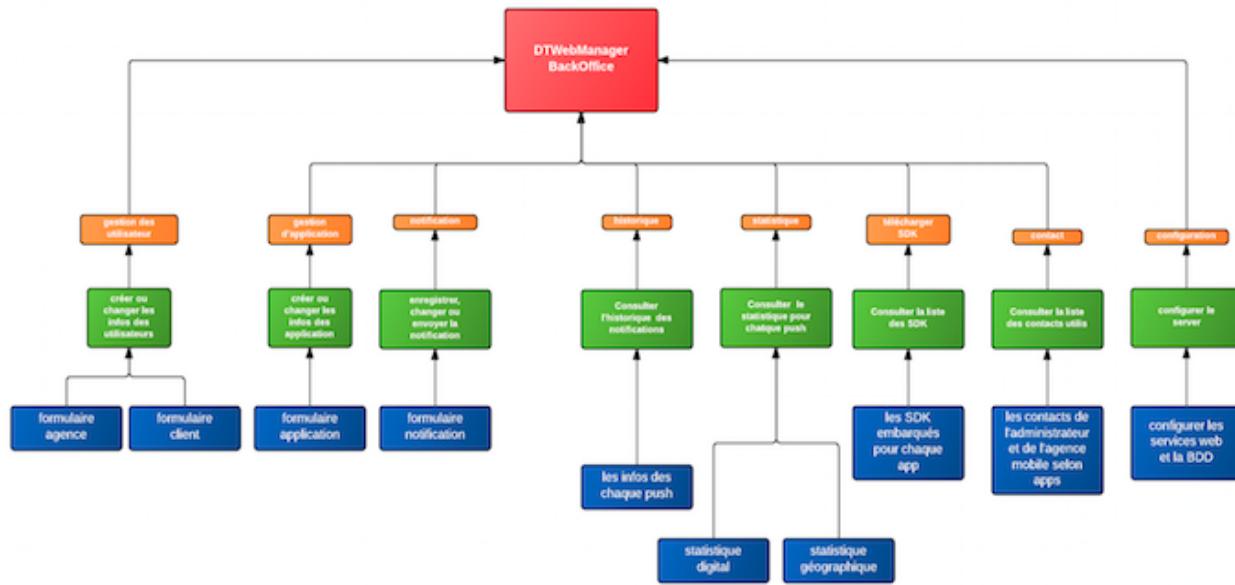
Annexe



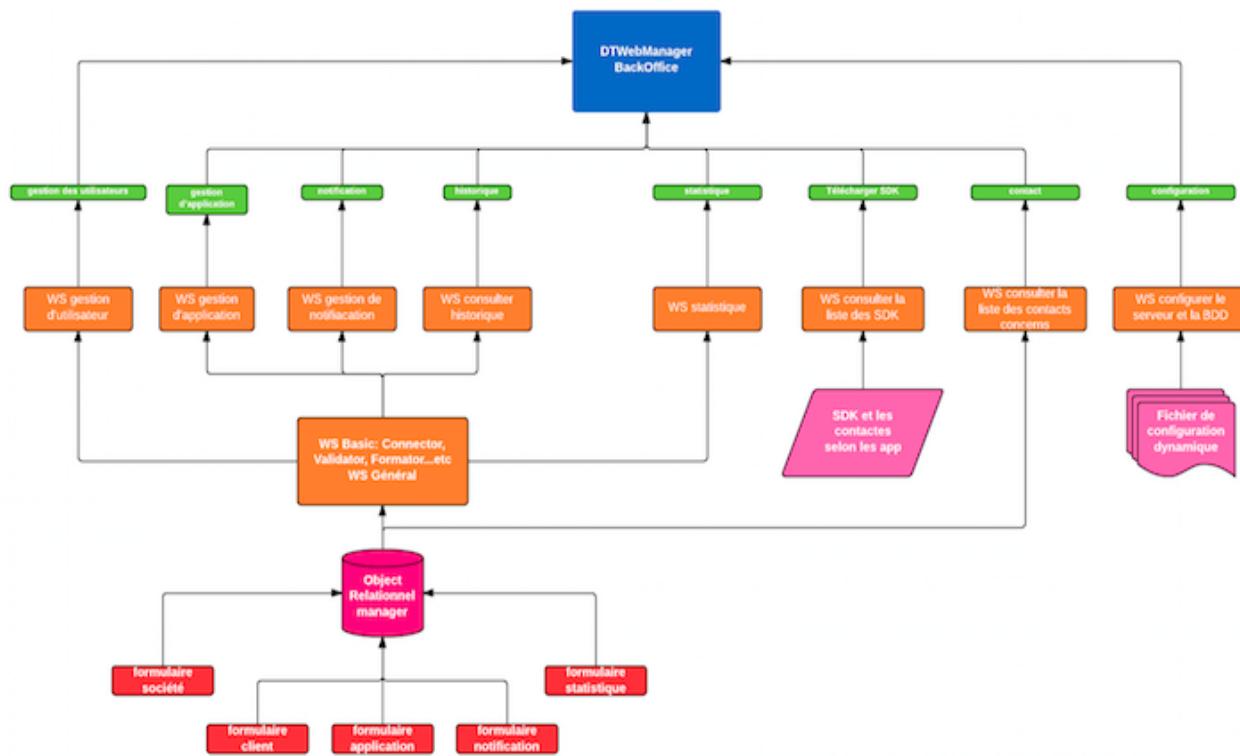
Annexe 1 Schéma du projet



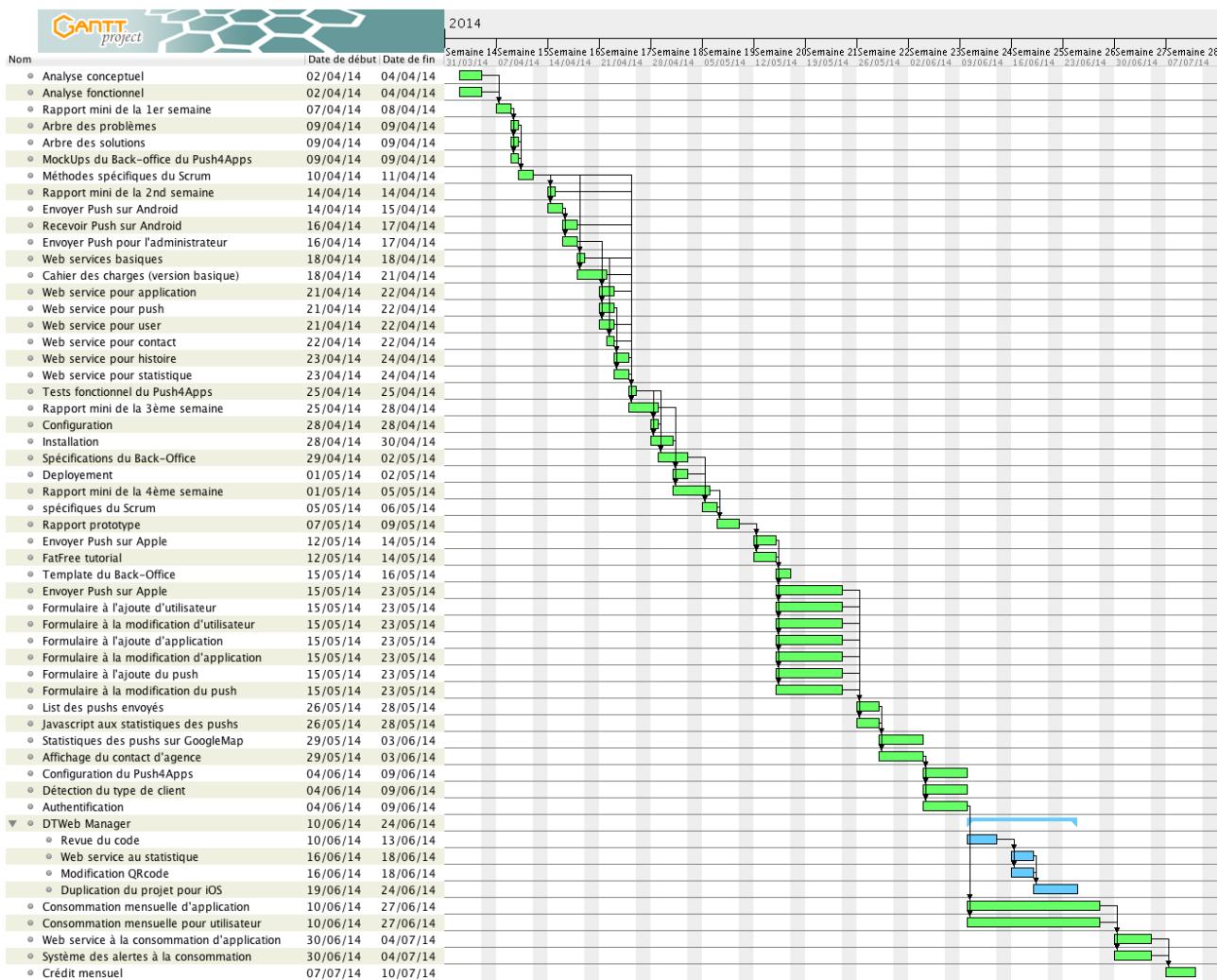
Annexe 2 Schéma de base de données



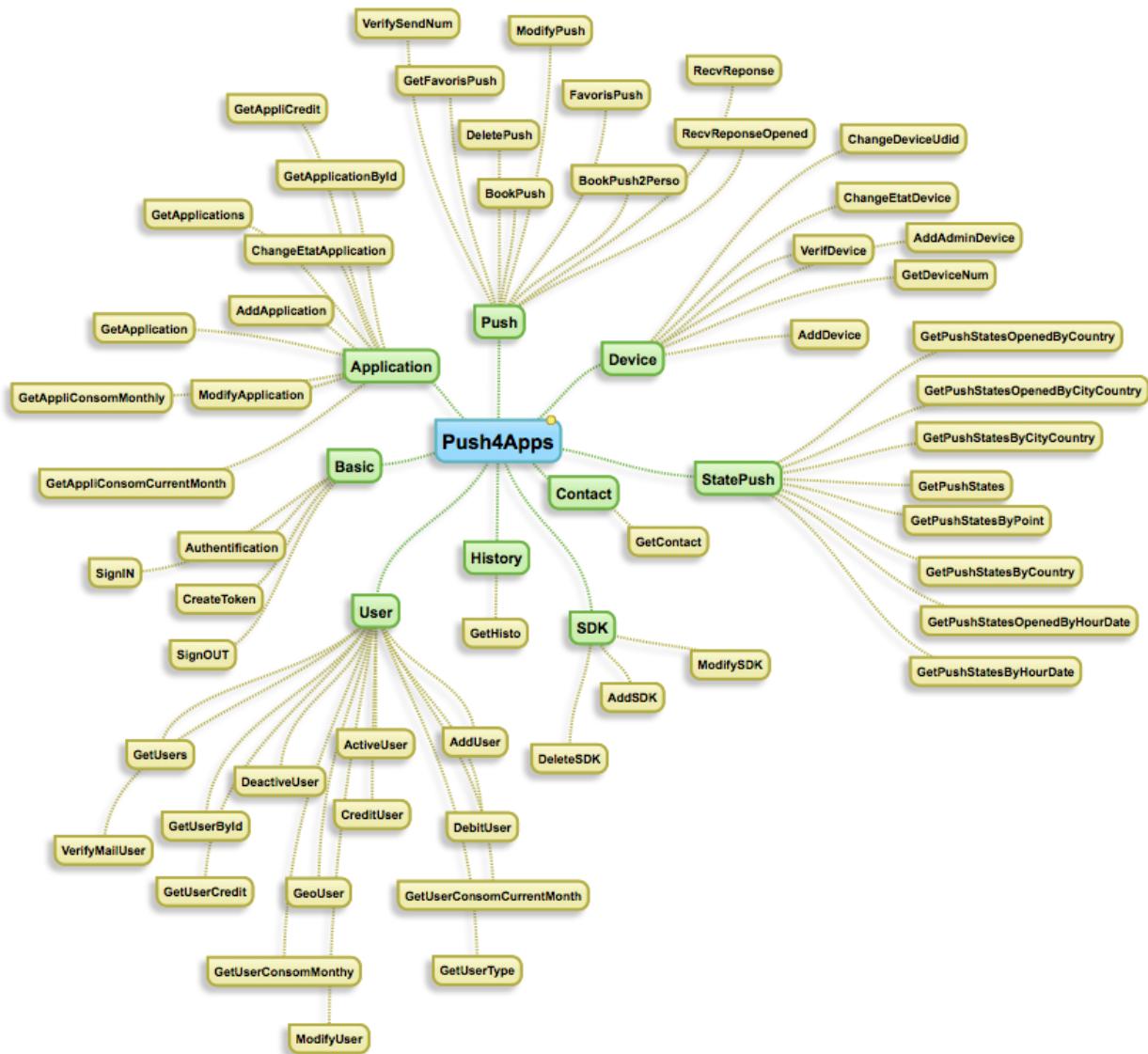
Annexe 3 L'arbre des problèmes techniques



Annexe 4 L'arbre des solutions aux problèmes techniques



Annexe 5 Diagramme de Gantt réel du Push4Apps



Annexe 6 la carte heuristique à l'organisation des web service du Push4Apps

// Fonction à la recherche de notification mobile
DEBUT FONCTION SearchPush

SQL script à la recherche de pushes non-envoyés

SI on a trouvé un plus qui n'est pas encore envoyé **ALORS**

TANT QUE la liste de pushes à envoyer n'est pas encore vide

Créer un objet à envoie de push

Décider le point zéro et la rayon de push

SI le push sera envoyé sur les smartphone Android **ALORS**

TANT QUE il y a des appareilles iOS hors de la liste de Anroid de push

Ajouter le UDID de l'appareille iOS dans la liste de Android de push

FIN TANT QUE

FIN SI

SI le push sera envoyé sur les smartphone iOS **ALORS**

TANT QUE il y a des appareilles iOS hors de la liste de Anroid de push

Ajouter le UDID de l'appareille iOS dans la liste de Android de push

FIN TANT QUE

FIN SI

FIN TANT QUE

TANT QUE il reste un push non envoyé

Envoyer push;

Écrire le temps d'envoie dans la base de données;

FIN TANT QUE

Si NON

Retourner "Non push réservé";

FIN SI

FIN FONCTION

Annexe 7 programme pour envoyer le push et la classe Push

```
// fonction récursive à la gestion de Push4Apps
public static JSONArray GetUsers (@QueryParam("idUser")int IdUser ){

    JSONArray res = null;
    int type = 0;

    //Initialisation de res et de type

    //Administrateur du Push4Apps
    if(type == 1) {

        //Chercher les agences de l'administrateur

        if(res.size() == 0){
            return res;
        }else{
            for(int i=0; i < res.size(); i++){

                //Chercher les clients de l'agence par GetUsers

            }
        }
    }

    //Agence
    else if(type == 2) {

        //Chercher les clients

        if(res.size() != 0) {
            for(int i=0; i < res.size(); i++){

                //Chercher les applications du client par GetUsers

            }
        }
    }

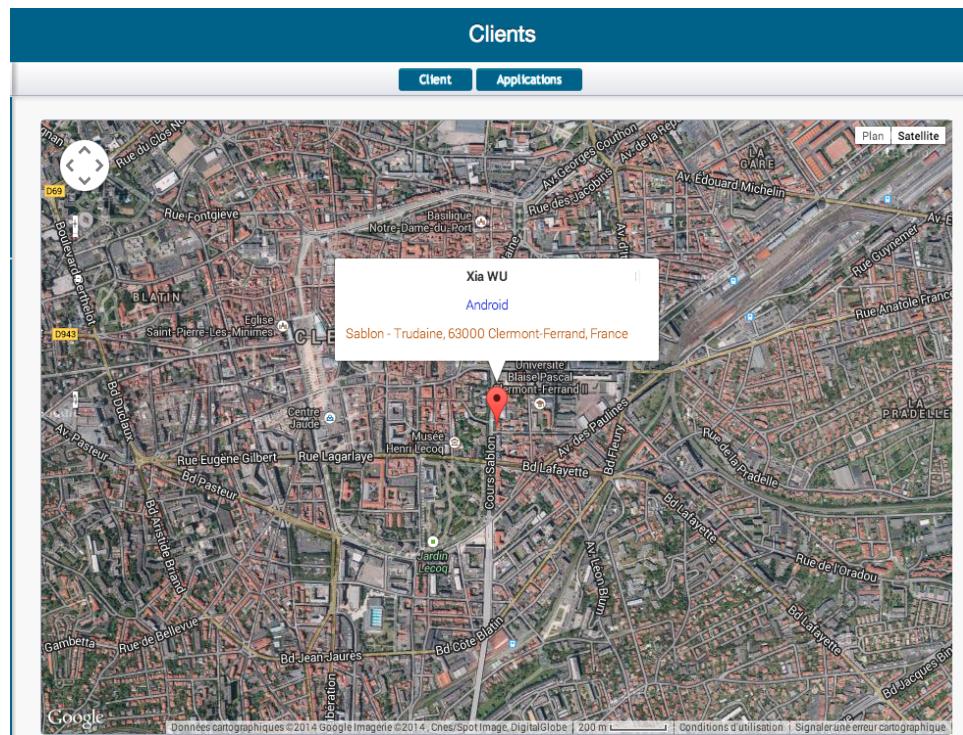
    //Client
    else if(type == 3) {

        //Chercher les applications par IdUser

    }

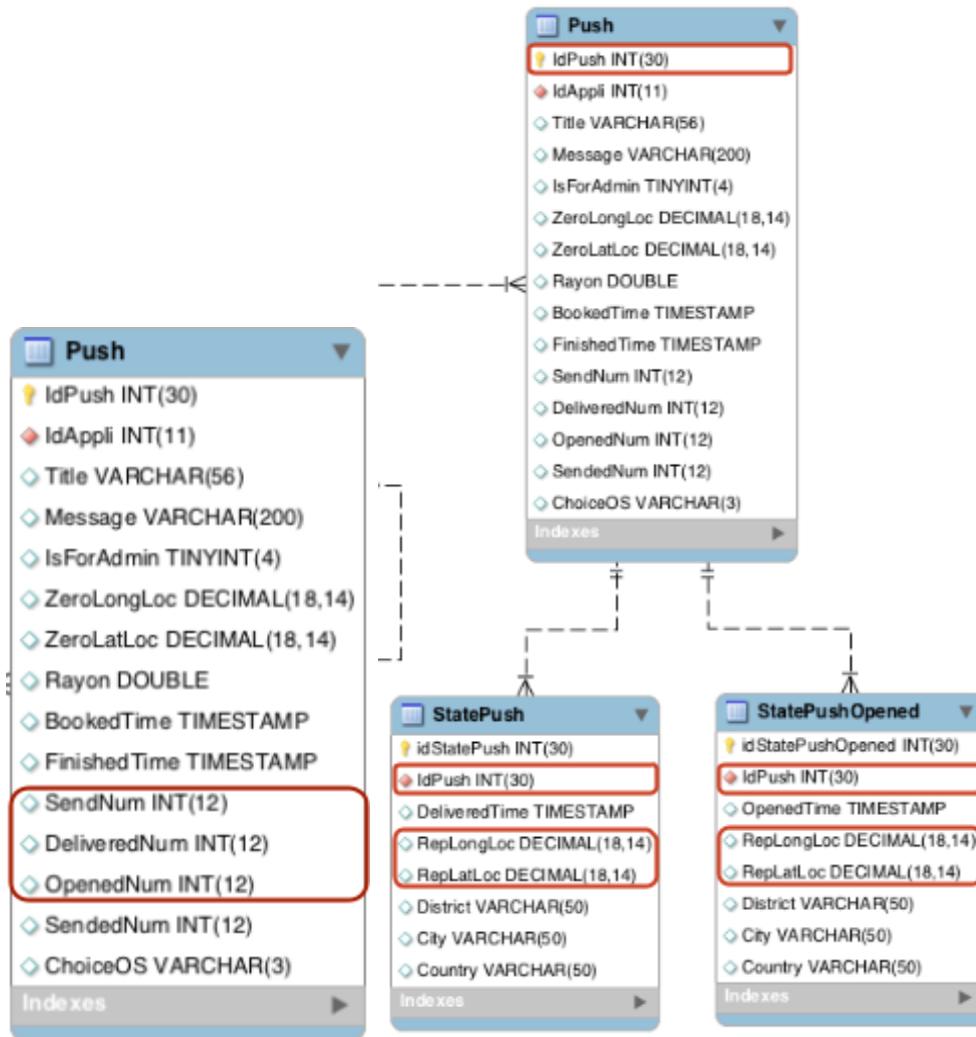
    return res;
}
```

Annexe 8 fonction récursive pour la table des utilisateurs



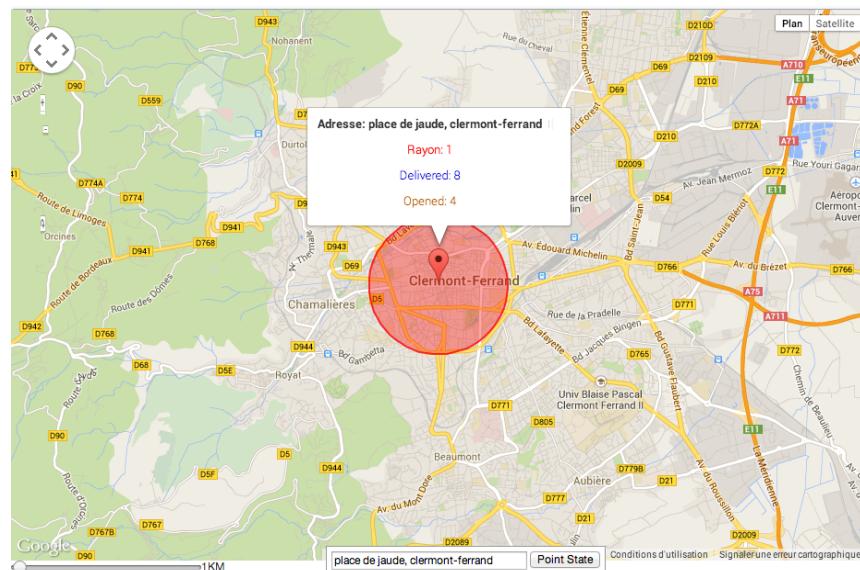
Annexe 9 la page de Géolive

Annexe 10 la page de Push2Perso



Annexe 11 la table Push

Annexe 12 la table StatePush et la table StatePushOpened



Annexe 13 Statistique par point spécifique

```

public class CalDistance {
    // calculer la distance globale entre deux points GPS
    public double distance(double lat1, double lon1, double lat2, double lon2, String unit) {
        double theta = lon1 - lon2;
        double dist = Math.sin(deg2rad(lat1)) * Math.sin(deg2rad(lat2))
                    + Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) * Math.cos(deg2rad(theta));
        dist = Math.acos(dist);
        dist = rad2deg(dist);
        dist = dist * 60 * 1.1515;
        if (unit == "K") {
            dist = dist * 1.609344;
        } else if (unit == "N") {
            dist = dist * 0.8684;
        }
        return (dist);
    }

    // calculer la distance horizontale
    public double deg2rad(double deg) {
        return (deg * Math.PI / 180.0);
    }

    // calculer la distance verticale
    public double rad2deg(double rad) {
        return (rad * 180.0 / Math.PI);
    }
}

```

Annexe 14 Calculatrice de la distance entre deux points GPS