

Développement d'applications

Alter Frame

Clément Hémidy

DUT Informatique

IUT Paris Descartes

Tuteur de Stage

Gérant d'Alter Frame

Thibault Montjean

Stage du 30 septembre au 13 décembre 2013



alter frame

Editeur de Solutions

Remerciements

Je remercie avant tout **Thibault Montjean** pour m'avoir accueilli au sein de son équipe et pour m'avoir fait confiance en me donnant des responsabilités sur différents projets et **Jean Zanazaka** pour m'avoir suivi lors de ce stage.

Je remercie également **toute l'équipe d'Alter Frame** et plus spécialement **Clément Fleury** pour son aide précieuse lors du développement des deux projets présentés dans ce rapport et pour avoir participé à mon apprentissage lors de ce stage qui a été pour moi une expérience enrichissante durant laquelle j'ai pu renforcer mes compétences dans le domaine du développement et découvrir ce qu'est le travail en équipe au sein d'une entreprise de service.

Table des matières

Remerciements	2
1. Introduction	4
2. Présentation de la société	5
2.1 Alter Groupe	5
2.2. Qu'est-ce qu'Alter Frame ?	7
2.3. Alter Frame, c'est tout d'abord une société éditrice de solutions	7
2.3.1. Solutions Full-web	7
2.3.2. Solutions plus spécifiques	8
2.3.3. Ma position dans Alter Frame	8
3. Présentation des projets	9
3.1. Projet 1 : Dtells.....	10
3.1.1. Le besoin exprimé par le client	10
3.1.2. La solution apportée par Alter Frame	10
3.1.2.1. Le serveur et les Web Services	11
3.1.2.2. Les Applications Mobiles.....	12
3.1.3. La réalisation du projet.....	13
3.1.3.1. Utilisation du framework Yii.....	13
3.1.3.2. Les Web Services (Modèles).....	14
3.1.3.3. Les Web Services (Contrôleur et Vues)	16
3.1.3.4. L'application Android	18
3.1.3.5. Le résultat final de l'application Android	20
3.1.3.6. Les apports de ce projet.....	21
3.2. Projet 2 : Schneider Electric, Help center (STIE).....	22
3.2.1. Le besoin exprimé par le client	22
3.2.2. La solution apportée par Alter Frame	22
3.2.3. La réalisation du projet.....	23
3.2.3.1 Le back office	23
3.2.3.2. Le résultat final du back office	24
3.2.3.3. Les web services SOAP.....	25
3.2.3.4. Les apports de ce projet.....	26
3.2.3.5. La comparaison personnelle entre REST et SOAP.....	26
4. Conclusion.....	27
5. Annexes	28
1. Classe « user » abrégée du projet php de « Dtells »	28
2. Classe « user » de l'application Android.....	31

1. Introduction

Ce rapport rend compte du Stage de fin d'études que j'ai réalisé dans le cadre de mon DUT Informatique au sein de l'IUT Paris Descartes. L'accomplissement de ce stage a lieu au cours de la deuxième année et permet la validation de ce Diplôme.

J'effectue ce stage dans la société « Alter-Frame » depuis le 30 Septembre et j'ai fait le choix de le prolonger jusqu'en Mars 2014.

Qu'elle soit positive ou non, l'expérience que pouvait m'apporter ce stage était pour moi une opportunité importante à saisir.

Pour ce stage j'ai préféré choisir une entreprise de taille moyenne comme « Alter Frame » permettant d'acquérir certaines responsabilités que je n'aurais pu avoir aussi rapidement dans un grand groupe.

L'intérêt de ce stage était donc de renforcer mes compétences en informatique mais aussi de découvrir le travail professionnel au sein d'une société de service.

Afin de rapporter au mieux mon expérience, je vais dans un premier temps présenter l'entreprise et le cadre dans lequel j'ai travaillé, puis je décrirai les différents projets dans lesquels j'ai pu apporter ma contribution lors de ces 10 semaines de stage. Pour terminer ce rapport, je ferai part de mon ressenti général et des enseignements que j'ai pu en tirer.

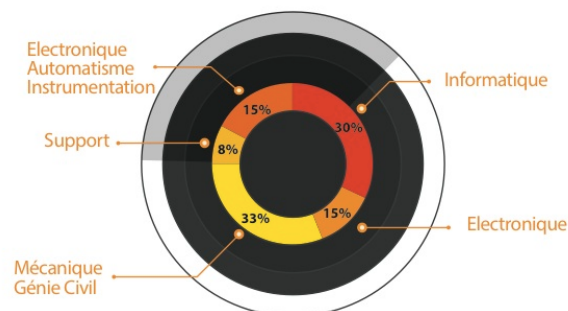
2. Présentation de la société

2.1 Alter Groupe

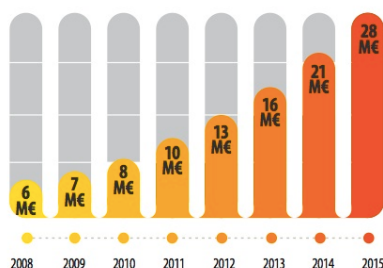
ALTER GROUPE est une jeune PME créée en 2005. Cette société est spécialisée dans le conseil en ingénierie et dans l'édition de logiciels, principalement sous la forme de développement au forfait.

Créé par les associés **Maxime Lacour** et **Louis Vachette**, **ALTER GROUPE** est au départ l'association de deux entités :

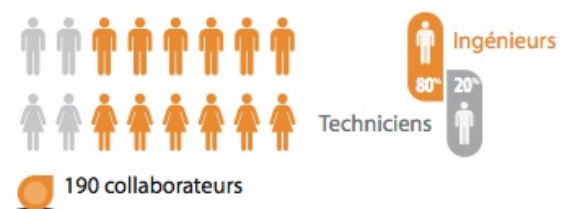
- **ALTER SOLUTIONS**, proposant des expertises à la pointe des nouvelles technologies dans les secteurs du transport, de l'énergie et des systèmes d'information ;
- **ALTER DÉFENSE**, proposant d'accompagner et de répondre aux besoins de sous-traitance des industriels des secteurs de l'aéronautique, de l'aérospatial et de la défense.
-



« Le groupe propose des expertises métiers variées ... »



13 M€, c'est le chiffre d'affaire d'ALTER GROUPE en 2012 représentant une croissance annuel de 30%



190, c'est le nombre de collaborateurs au sein d'ALTER GROUPE

Dans un second temps, est intervenue la création de la filiale **ALTER FRAME** opérée par **Thibault Montjean** et ses associés.

Créée pour répondre à un besoin de développement de logiciels sur mesure (applications techniques et scientifiques, logiciels de gestion et applications mobiles), elle est à présent une filiale bien implantée dans le groupe, d'autant qu'**ALTER FRAME** a ses bureaux situés au sein du siège social d'**ALTER GROUPE**.

2.2. Qu'est-ce qu'Alter Frame ?

Alter Frame, c'est **600 000** euros de chiffre d'affaires et **10** employés répartis sur plusieurs secteurs.

Alter Frame traite avec divers clients qui ont des demandes, elles aussi, variées. Nous pouvons y trouver des clients internationaux, par exemple, « Lafarge », « Chryso », « GDF-Suez » et des clients de proximité : « La Ferme de Gally », « L'établissement public du Château de Versailles » et « La mairie de Versailles ».

2.3. Alter Frame, c'est tout d'abord une société éditrice de solutions

Les activités d'Alter Frame se divise en deux branches :

- Des solutions Full-web
- Des solutions plus spécifiques

2.3.1. Solutions Full-web

Les solutions Full-web d'Alter Frame sont le plus souvent des IHM couplés avec des SGBD pour réaliser des intranets, des outils de gestion dédiés, etc. Ces solutions sont le plus souvent écrites à l'aide des langages PHP, HTML, CSS, XML et .NET.

Pour remplir les objectifs à bien, tout en gardant une qualité de code exemplaire, le Framework Yii est utilisé. Il permet ainsi de simplifier les étapes de développement des projets tout en réduisant les temps de développement.

Le SGBD MySQL est le plus souvent utilisé, avec le langage PHP (Yii) ; le SGBD Oracle est aussi utilisé, mais plus rarement.

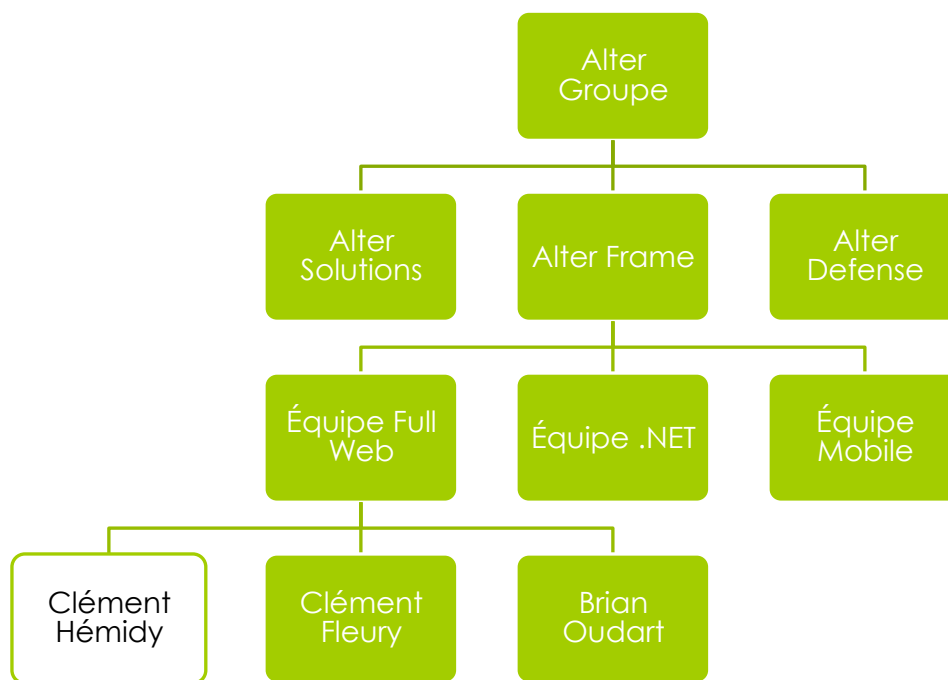
Le SGBD SQL Server est aussi utilisé pour les projets développés sous .NET, compte tenu de son intégration presque parfaite dans ce *framework*.

2.3.2. Solutions plus spécifiques

Des solutions plus spécifiques peuvent être réalisées lors de projets plus complexes, avec des besoins particuliers exprimés par le client.

Alter Frame va par exemple utiliser cette voie de développement pour des projets qui nécessitent du calcul, avec des traitements complexes de données. Le développement de ces projets passe alors par l'utilisation d'outils de « bas niveau » sous les langages C, C++, ADA et Fortran par exemple.

2.3.3. Ma position dans Alter Frame



La hiérarchie d'Alter Frame

Bien que les missions principales qui m'ont été confiées se soient principalement réalisées au sein de l'équipe « Full Web », j'ai pu aussi apporter ma contribution dans l'équipe « Mobile » pour le projet « Dtells » et son application sous Android.

3. Présentation des projets

Au sein d'Alter Frame, j'ai tout de suite été plongé dans le travail en continu, au sein du service, comme un employé normal. Mon tuteur de stage m'a donc proposé quelques projets disponibles et mon choix s'est arrêté sur un sujet faisant à la fois appel à la programmation serveur et à la programmation mobile. Ce projet me convenait parfaitement, pour plusieurs raisons :

- L'opportunité de consolider mes bases relativement solides en programmation PHP, tout en apprenant à utiliser le Framework Yii en faisant appel aux personnes expertes dans ce domaine et proches de moi ;
- L'occasion de faire évoluer mes compétences en Java (SDK Android) lors du développement d'un projet d'application Android.

3.1. Projet 1 : Dtells

3.1.1. Le besoin exprimé par le client

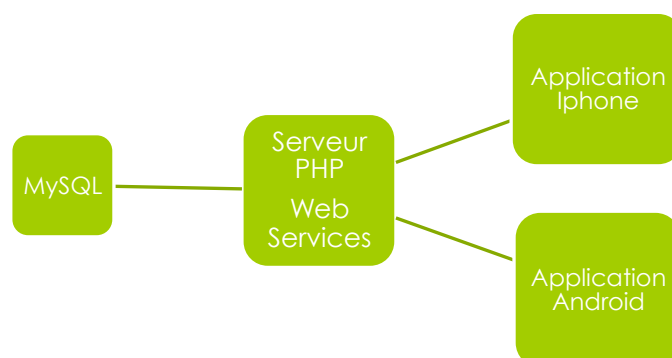
Outil de gestion centralisée de contacts professionnels et personnels, Dtells a pour but de rassembler la totalité de ses contacts dans le Cloud. L'idée est venue initialement du besoin personnel de son créateur. Elle a essentiellement l'avantage de garantir la pérennité de son carnet d'adresses. Un changement de téléphone ? Il suffit de 5 minutes et tous ses contacts sont restaurés. Vous changez de société et voulez mettre à jour vos coordonnées professionnelles en les diffusant à votre cercle de contacts : Dtells vous le permettra. Avec la quasi-présence des Smartphones dans la société actuelle, la décision de développer une réponse pratique à ce besoin sur mobile (IOS et Android) n'a pas été délicate à prendre.

Le développement devra conduire à deux applications mobiles (sous IOS et sous Android) qui permettront de demander la création d'un compte Dtells, de remplir deux types de cartes de visite, professionnelle et particulier, avec leurs informations respectives et enfin d'ajouter d'autres personnes inscrites sur la plateforme pour partager ses cartes de visite.

Le besoin fonctionnel étant ainsi exprimé, une solution technique a dû être trouvée pour y répondre.

3.1.2. La solution apportée par Alter Frame

Le projet se compose donc de deux applications mobiles et d'un serveur permettant la communication et la sauvegarde des données.



Architecture du projet

Dans la réalisation de ce projet, il m'a été confié plus particulièrement de prendre en charge la partie liée au serveur avec les Web Services et l'application Android.

3.1.2.1. Le serveur et les Web Services

Pour le serveur, j'ai pris la décision de développer cette partie sous PHP avec le Framework Yii. Utiliser le Framework Yii était tout nouveau pour moi car je ne l'avais pas utilisé auparavant mais un collègue ayant une grande connaissance de ce Framework a pu me donner l'aide nécessaire pour mener à bien le projet.

Pour homogénéiser les communications et permettre une même utilisation du serveur tant du côté de l'application iPhone que de l'application Android, j'ai pris la décision d'utiliser l'architecture Client-Serveur REST (**RE**presentational **S**tate **T**ransfer).

Mais qu'est-ce que « REST » ?

REST est tout d'abord un très bon moyen pour ouvrir un accès vers un SGBD tout en permettant de contrôler l'interaction que le client aura avec ce même SGBD. Grâce à l'architecture REST, la table du modèle de données peut être ouverte au client en étant caractérisée par une **ressource**. Pour pouvoir manipuler cette ressource, le protocole HTTP est composé de plusieurs types de requêtes ou **verb**.

- GET : récupération du contenu d'une ressource
- POST : création d'une ressource
- PUT : modification d'une ressource
- DELETE : suppression d'une ressource

En jouant sur l'utilisation de ces verbs, il est possible de manipuler la totalité du modèle de données.

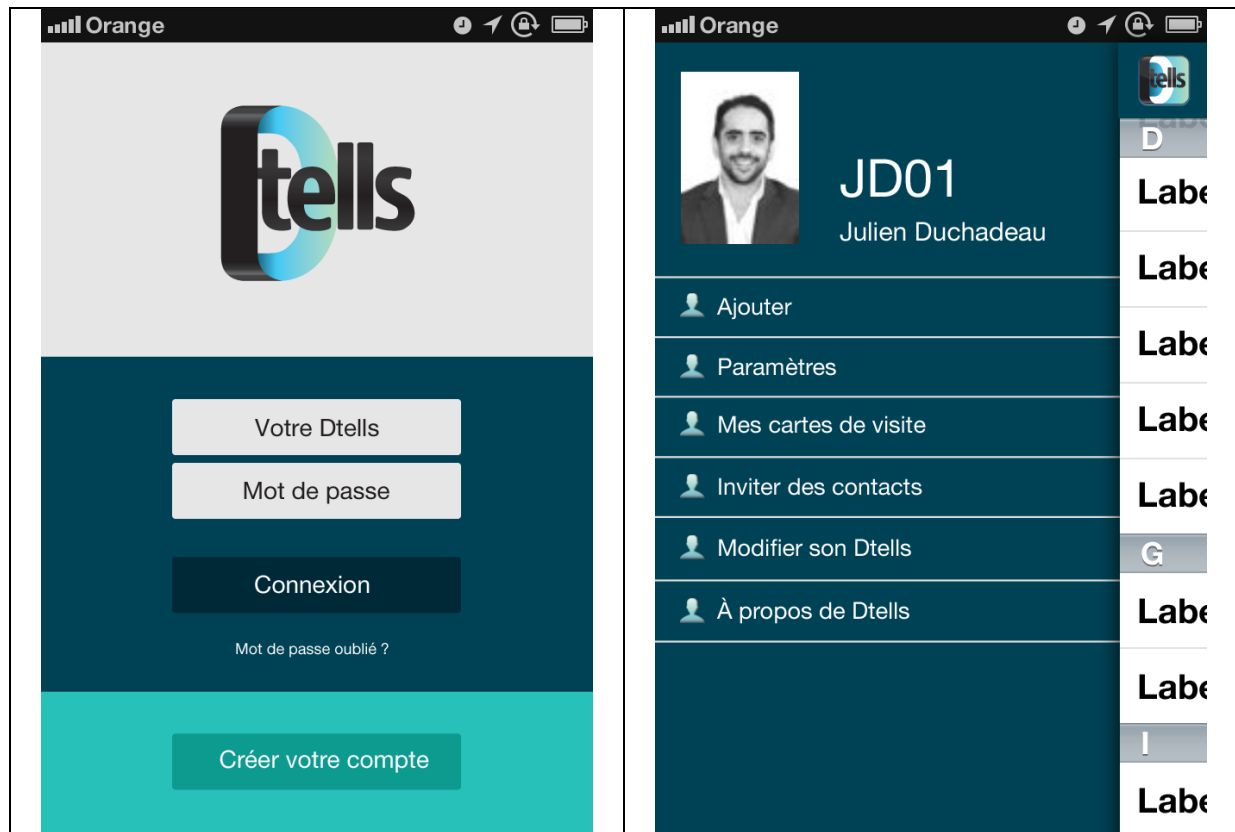
Exemple : la requête POST sur « /api/user » permettra la création d'un utilisateur et la requête DELETE sur « /api/user » permettra la suppression d'un utilisateur.

Pour la mise en œuvre de cette architecture, j'ai utilisé l'extension « RestFullYii » du Framework Yii qui a l'avantage de faciliter le développement de Web Services de type REST.

Pour stocker les données du client, le choix d'un serveur de type MySQL a été fait.

3.1.2.2. Les Applications Mobiles

L'objectif principal des deux applications mobiles étant précisé, il suffisait alors de passer à la programmation. De plus il nous a été fourni des maquettes d'écran représentant précisément le résultat final souhaité par le client.



Maquettes proposées par le client

3.1.3. La réalisation du projet

3.1.3.1. Utilisation du framework Yii

Yii permet la création d'application Full-Web avec l'architecture MVC. Dans cette architecture nous retrouverons donc les **M**odèles, les **C**ontrôleurs et enfin les **V**ues.

M. Ces modèles sont l'image d'une table relationnelle du serveur MySQL en PHP. Cette gestion des modèles intégrés dans Yii permet une utilisation simplifiée du SGBD. On pourra par exemple instancier automatiquement un modèle contenant les données de la base de données ou enregistrer les modifications dans la base.

V. La vue est la partie qui s'occupera uniquement de l'affichage – sans aucun traitement – demandé par le contrôleur. Très souvent on retrouve principalement du code HTML. *Dans notre cas il s'agira de l'affichage en « JSON » du résultat de la requête REST qui sera renvoyé au téléphone.*

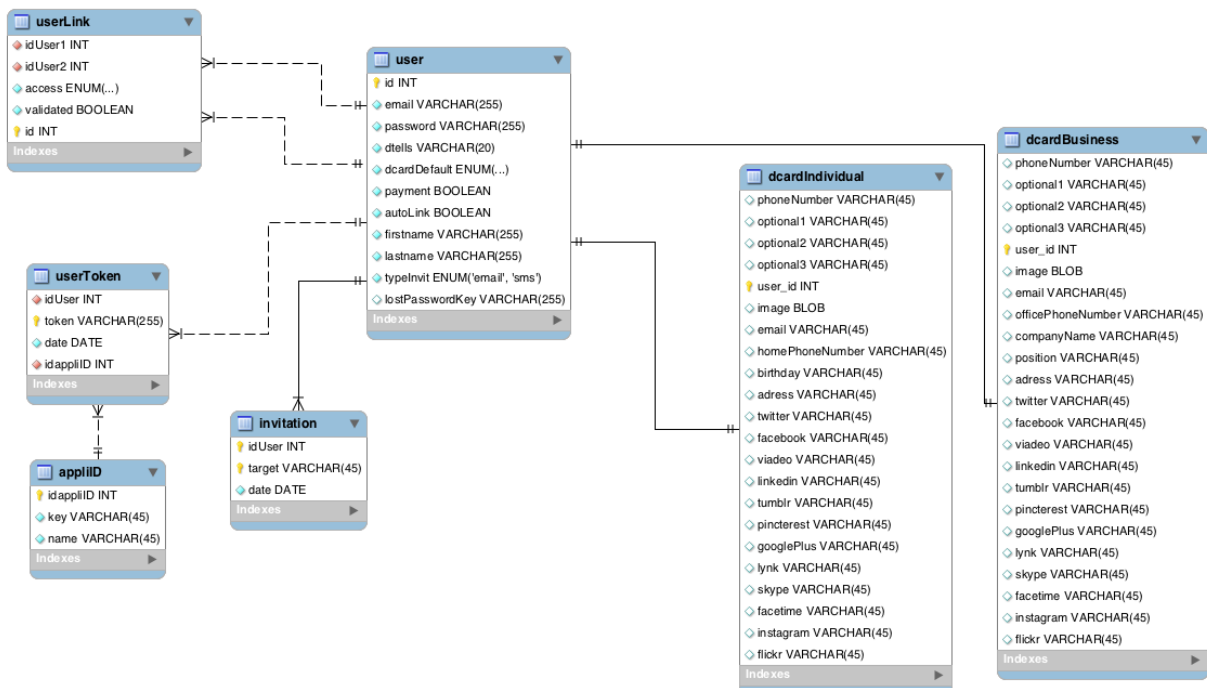
C. Enfin nous avons les contrôleurs permettant d'effectuer l'ensemble du traitement d'une requête cliente, en utilisant les modèles et en affichant le résultat dans une vue. Le contrôleur sert alors d'aiguillage.

3.1.3.2. Les Web Services (Modèles)

La première étape d'un projet consiste donc à expliciter le modèle de données utilisé. Cette étape est **cruciale** car grâce à elle, tous les modèles de notre application PHP pourront être construits.

Une bonne application PHP commence par la qualité de son modèle...

J'ai donc construit ce modèle avec les spécificités fonctionnelles requises et le résultat final est le suivant :



Notre modèle de données composé de 7 tables.

- **user** : table contenant les informations principales d'un compte utilisateur
- **userLink** : table permettant la création de liens entre les utilisateurs pour former les carnets d'adresses
- **userToken** : table contenant le système de jetons des Web Services REST
- **appliID** : table recensant toutes les applications utilisant les Web Services
- **invitation** : table permettant la création de comptes afin de savoir si le futur utilisateur s'est fait parrainer
- **dcardIndividual** & **dcardBusiness** : tables enregistrant les cartes de visites de l'utilisateur

Choix spécifique: les tables dcardIndividual et dcardBusiness ont été tout simplement dupliquées, sans factorisation, rendant l'utilisation des Web Services associés plus simples.

Dans la continuité, j'ai réalisé les modèles en PHP. Cette étape est très importante tout en s'avérant très longue.

En effet ces modèles PHP contiennent en plus de la description de ce qu'il y sera représenté beaucoup de configuration. On va devoir définir les règles de contrôle pour éviter toute donnée non.

Par exemple pour un champ contenant une adresse mail, nous allons retrouver une règle de contrôle vérifiant que la valeur renseignée dans l'instance de ce modèle répond bien à cela.

Toutes ces règles nous permettrons d'éviter toute erreur SQL lors d'un enregistrement.

L'annexe 1 est le modèle PHP de la table « user » décrite ci-dessus.

Nous pourrons remarquer entre autre les validateurs représentés dans la fonction « rules ».

Nous retrouverons de même les relations inter modèle permettant l'accès aux éléments d'une table reliée par une clé étrangère.

3.1.3.3. Les Web Services (Contrôleur et Vues)

Contrôleurs

Pour réaliser les Web Services avec « RestFullyii », chaque ressource proposée au client est représentée par un contrôleur.

J'ai donc réalisé 5 contrôleurs :

- **UserController** : le plus important car il gère les inscriptions, les connexions, les profils...
- **UserLinkController** : ce contrôleur permet la création, l'édition et le visionnage des liens entre membres.
- **InvitationController** : il permet la gestion des parrainages.
- **DcardIndividualController** & **DcardBusinessController** : ils permettent l'édition des cartes de visites.

Mais une difficulté supplémentaire s'est ajoutée au projet. En plus d'avoir la possibilité d'accès et de manipulation des ressources, des actions spécifiques sur ces ressources ont été nécessaires.

Nous avons par exemple introduit dans le contrôleur UserController une action spécifique « login » qui va permettre grâce au Dtells et au mot de passe de générer un jeton d'accès à ces Web Services.

Vues

Les « Vues » sous RestFullyii sont générées automatiquement sous le format JSON. Elles représentent une réponse que le client (téléphone mobile) devra interpréter. Elles peuvent signifier des erreurs, des résultats de traitements spécifiques ou de simples restitutions de données.

Exemple de résultat renvoyé par les Web Services

Quand l'url « /api/user » (UserController) sera appelée avec le « **verb** GET », le serveur renverra une réponse sous cette forme :

```
{
  "data": {
    "totalCount": "1",
    "user": {
      "dtells": "test",
      "lastName": "My last name",
      "firstName": "My first name",
      "id": "12",
      "dcardBusiness": null,
      "email": "test@test.com",
      "dcardIndividual": null,
      "dcardShopping": null,
      "dcardDefault": "individual"
    }
  },
  "success": "true",
  "message": "Record find"
}
```

Cette réponse sera ensuite traitée côté client.

3.1.3.4. L'application Android

L'application Android a tout d'abord été développée sous une architecture MVC.

Nous retrouvons ici, à l'image du Framework Yii présenté ci-dessus, des modèles, qui sont tout simplement des classes Java composées d'attributs. Ces modèles vont permettre d'instancier les résultats des Web Services et de les stocker en local pour dans un second temps procéder à de la gestion de cache.

La gestion du cache permettra par exemple l'ouverture de l'application sans connexion internet.

Dans l'application Dtells j'ai donc créé une classe « user » composée des champs de la table « user ». Cette classe représentera un utilisateur.

Annexe 2 : modèle user de l'application Dtells.

Dans l'annexe nous pourrions remarquer un constructeur adapté pour construire une instance en fonction du résultat d'un Web Service.

Globalement, l'application est composée de « fenêtres » qui constituent ni plus ni moins l'ensemble des pages de l'application Android. Ces différentes fenêtres sont appelées des activités. Dans ces activités, nous allons donc placer la partie de code qui exécutera les traitements, les événements qui s'activeront lors du touché de l'utilisateur sur un bouton par exemple. Cette même activité aura en charge le chargement de la vue.

Nous avons enfin les vues ou « Layout ». Ces vues sont écrites en XML et permettent de générer les affichages sous forme d'objets comme des boutons, des champs d'édition, et plein d'autres éléments mis à disposition par le SDK Android.

Réalisation

Pour réaliser cette application j'ai suivi plusieurs étapes :

- Étape 1 : j'ai réalisé toutes les vues (XML) et l'interface
- Étape 2 : j'ai procédé à la création des modèles (Java)
- Étape 3 : j'ai mis en place la communication entre les Web Services et mon application
- Étape 4 : j'ai créé tout les composants nécessaires à l'interaction utilisateur permettant l'utilisation de l'interface et donc de l'application

Disposant des maquettes des pages souhaitées par le client, j'ai pu travailler très rapidement sur les vues en faisant abstraction des traitements.

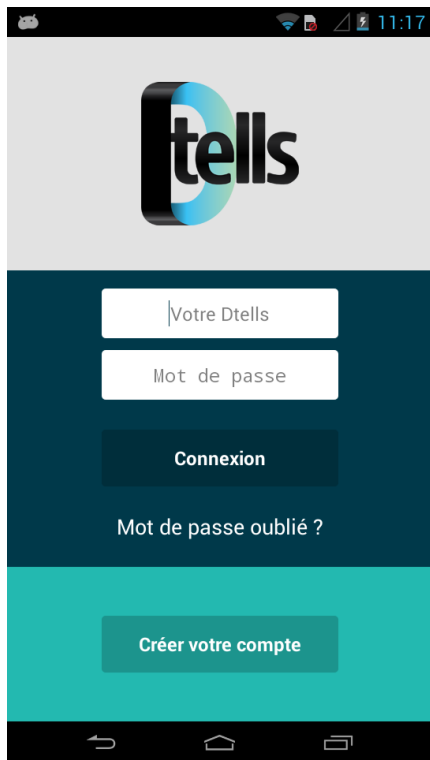
La préparation des vues terminée, je me suis ensuite attardé sur la création des modèles locaux. Pour réaliser ces modèles il m'a suffit de créer des objets Java, à l'image de ceux que nous pouvons retrouver sur le serveur PHP.

Ces modèles m'ont permis de stocker localement le contenu qui est présent sur le serveur grâce aux Web Services.

La suite logique du déroulement de ce projet a été de préparer tous mes échanges avec les Web Services, les actions dont l'application avait besoin (récupérer la liste de ses contacts, modifier les informations de son compte, modifier ses cartes de visites, etc).

Disposant de tous les composants principaux désormais prêts à l'emploi, il m'a donc suffit d'associer le tout pour rendre l'application fonctionnelle.

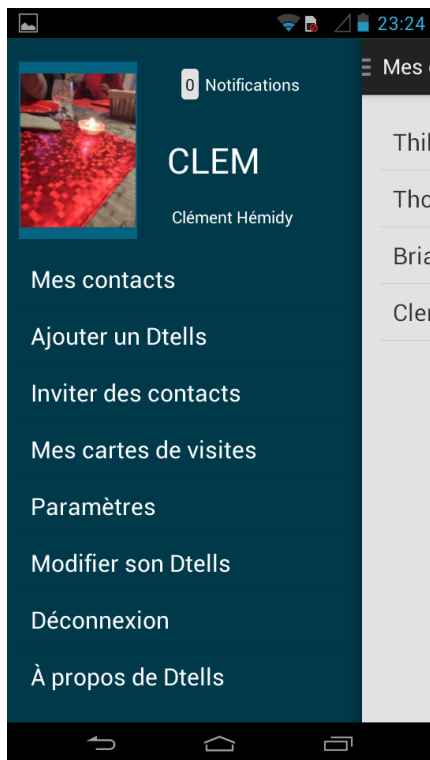
3.1.3.5. Le résultat final de l'application Android



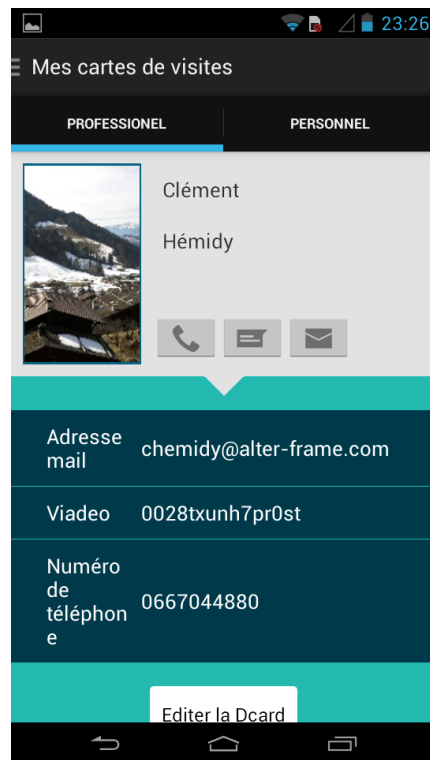
Connexion



Inscription



Menu de l'application



Affichage d'une carte de visite

3.1.3.6. Les apports de ce projet

Lors de la réalisation de ce projet, j'ai peu à peu découvert la complexité de la relation client.

Ayant participé à une réunion avec le client et aux différentes livraisons intermédiaires, je commence à avoir une bonne vision de ce qu'est la réalisation d'un projet au sein d'une SSII.

Cependant, ayant déjà longuement fait l'apprentissage du travail en équipe, la coopération avec mon collègue qui prenait en charge la réalisation de l'application Iphone s'est avérée peu compliquée.

3.2. Projet 2 : Schneider Electric, Help center (STIE)

3.2.1. Le besoin exprimé par le client

Schneider Electric, premier constructeur Français de matériel de gestion électrique, dispose de plusieurs centres de support téléphonique. Pour simplifier la vie des personnes y travaillant et permettre une meilleure qualité du support technique, Schneider Electric a comme ambition d'équiper d'iPad tout le personnel de ces centres.

Sur ces iPad il est envisagé d'installer une application destinée à aider les hôtes dans leur démarche de support, sous forme d'une série de questions/réponses. Les hôtes devront pouvoir initialiser une nouvelle session de support et ainsi engager une série de questions/réponses conduisant au final à une solution.

Schneider a ainsi fait appel à Alter Frame pour réaliser ce projet.

3.2.2. La solution apportée par Alter Frame

Nous avons proposé de structurer la plateforme comme ci-dessous.

- PHP
 - Un back-office permettant l'administration des données
 - Questions
 - Réponses
 - Solutions
 - Des Web Services en SOAP permettant à l'application iPad de se synchroniser sur le back-office
- Une application iPad

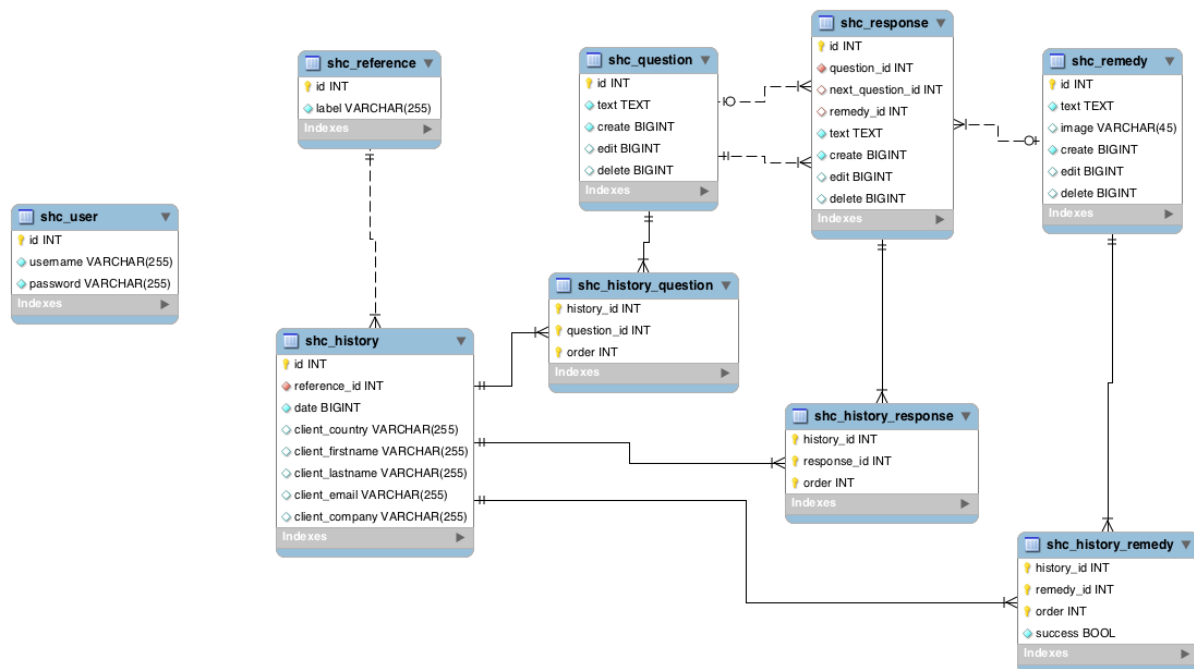
Dans ce projet, il m'a été demandé de m'occuper du back office couplé aux web services.

3.2.3. La réalisation du projet

3.2.3.1 Le back office

J'ai développé le back office en PHP avec le Framework Yii. Ayant depuis le projet « Dtells » plus de facilité sur ce Framework, la réalisation a été plutôt aisée.

Dans ce projet, un collègue a eu en charge la création du modèle. C'est pourquoi j'ai eu comme unique responsabilité de créer les modèles en PHP.



Modèle de données pour le projet STIE

La différence importante avec le projet « Dtells » réside dans le fait que j'ai dû créer une application web et non des web services. J'ai utilisé le Framework CSS/HTML « Bootstrap » pour créer rapidement une interface web.

La forte utilité de Yii s'est révélée lors de cette étape, avec l'existence de l'extension « YiiBooster » permettant la création d'élément HTML Bootstrap sous Yii.

Le back office ainsi réalisé comporte au final 3 sections (3 contrôleurs) permettant la création, l'édition et la suppression des questions, réponses et solutions de la base de données.

3.2.3.2. Le résultat final du back office

STIE - Help Center

Issues

Remedies

Logout

Home / Issues

Issues

New Issue

Displaying 1-3 of 3 results.

#	Text	
10	Drives does not start, no code displayed	
11	Check the environment, is it ok ?	
12	Drive/motor connection status	

Version : 0.0.0-staging
© 2013 Propriété de Alter Frame
Tous droits réservés.

STIE - Help Center

Issues

Remedies

Logout

Home / Remedies

Remedies

New Remedy

Displaying 1-2 of 2 results.

#	Text	
2	Replace the drive	
31	Restart the drive	

Version : 0.0.0-staging
© 2013 Propriété de Alter Frame
Tous droits réservés.

Les questions et solutions affichées sous forme de tableaux

STIE - Help Center

Issues

Remedies

Logout

Home / Issues / Issue #10

Issue #10 : Drives does not start, no code displayed

Responses

Displaying 1-4 of 4 results.

#	Text	Next Question	Next Remedy	
13	BLF	12-Drive/motor connection status		
14	EEF	11-Check the environment, is it ok ?		
15	CrF		2-Replace the drive	
16	IF 1		31-Restart the drive	

New response

Label *

☒ Your response has a question for next step.
☐ Your response has a remedy for next step.

Next question :

Create

Version : 0.0.0-staging
© 2013 Propriété de Alter Frame
Tous droits réservés.

L'édition de questions incluant la création et l'édition de réponses

3.2.3.3. Les web services SOAP

Sur ce projet nous avons fait un choix différent. Le protocole SOAP a été utilisé pour les Web Services. En effet, « Dtells » et « STIE » étant les deux premiers projets Mobile d'Alter Frame ayant besoin de Web Services, la stratégie a été de tester les deux principales technologies du domaine pour faire un choix judicieux lors du développement des prochains projets similaires.

L'avantage du protocole SOAP a été son intégration native dans le framework Yii permettant ainsi un développement aisé.

Le protocole SOAP ayant la même utilité que REST, sa différence se situe au niveau de l'échange. Sur SOAP les données transitant entre le serveur et le client sont typées, permettant l'échange d'objets.

L'intégration de SOAP sur Yii est une liste d'actions accessibles. Chaque action est décrite avec les données attendues en entrées et les données renvoyées au client.

Il suffit au final de programmer le traitement effectué par l'action.

Lors de ce projet j'ai donc réalisé deux actions :

- **updateDatabase** prenant un entier en entrée et renvoyant un tableau en sortie
 - Cette action permet à l'iPad de récupérer les éléments créés, modifiés ou supprimés depuis la dernière synchronisation.
- **sendHistory** prenant un Objet représentant l'historique d'une session de support et renvoyant *null* si l'insertion en base se passe bien

3.2.3.4. Les apports de ce projet

Sur ce projet j'ai pu découvrir l'intégration de Bootstrap au sein de Yii et ainsi mieux appréhender une autre facette de l'utilisation de Yii.

J'ai aussi eu l'avantage de pouvoir participer au projet, du début jusqu'à la fin, me permettant de voir la première livraison client.

Ce projet m'a aussi permis de pouvoir découvrir le protocole SOAP, et ainsi de permettre de me faire une idée sur les deux principales solutions proposées pour la création de Web Services.

3.2.3.5. La comparaison personnelle entre REST et SOAP

Je souhaite ici exprimer mon ressenti final sur le développement d'un Web Services en PHP.

REST offre la possibilité d'être plus flexible mais nécessite en contrepartie un temps de développement plus important que pour SOAP.

SOAP ne restant qu'un simple protocole, on pourra utiliser uniquement ce qui est prévu par celui-ci.

SOAP nécessite par contre un temps de développement et une certaine difficulté à l'utilisation sur les clients (téléphone dans notre cas).

Personnellement, je conseillerais donc l'utilisation de Web Service REST car ils demeurent tout de même plus flexibles et faciles d'utilisation sur les clients. Par contre une qualité de réalisation est nécessaire pour avoir une utilisation optimale de ceux-ci.

4. Conclusion

Ce début de stage a été pour moi une expérience positive, enrichissante et synonyme de nombreux enseignements.

J'ai en effet eu la chance de pouvoir découvrir le fonctionnement d'une société de service structurée permettant un apprentissage dans la vie professionnelle de qualité. Au sein d'Alter Frame j'ai pu apprécier l'atmosphère très « professionnelle » qui réside au sein de l'équipe et sa façon de mener à bien les projets. Autant de modèles à garder en mémoire pour la suite de mon parcours professionnel.

Personnellement j'ai surtout apprécié la confiance qu'a pu avoir Thibaut Montjean dans sa décision de me laisser travailler sur des projets d'envergure, en me donnant des responsabilités et en me permettant d'acquérir une expérience non négligeable.

J'ai très naturellement réussi à m'intégrer dans l'équipe d'Alter Frame, rendant l'expérience d'autant plus intéressante vu l'importance d'une bonne communication au sein d'une équipe comme celle-ci.

En conclusion, cette première partie de stage, très positive, m'aura permis de me conforter dans mon idée de le prolonger jusqu'en mars 2014. De même, je me suis renseigné auprès de mon supérieur pour éventuellement poursuivre mes études en alternance au sein d'Alter Frame.

Ce stage aura ainsi été une opportunité pour rencontrer des personnes de tous horizons, me permettant d'envisager avec sérénité les suites possibles de ma formation en informatique.

5. Annexes

1. Classe « user » abrégée du projet php de « Dtells »

<?php

```
/**
 * Modèle de base des Users
 *
 * @property integer $id
 * @property string $email
 * @property string $password
 * @property string $dtells
 * @property string $dcardDefault
 * @property integer $payment
 * @property integer $autoLink
 * @property string $firstname
 * @property string $lastname
 * @property string $typeInvit
 * @property string $lostPasswordKey
 *
 * @property DcardBusiness $dcardBusiness
 * @property DcardIndividual $dcardIndividual
 * @property Invitation[] $invitations
 * @property UserLink[] $userLinks
 * @property UserLink[] $userLinks1
 * @property UserToken[] $userTokens
 */
abstract class BaseUser extends GxActiveRecord {

    /**
     * Constructeur du modèle
     *
     * @param string $className nom de la classe du modèle
     * @return CActiveRecord modèle
     */
    public static function model($className=__CLASS__) {
        return parent::model($className);
    }

    /**
     * Nom de la table correspondante
     * Renvoie le nom de la table du modèle en base de données.
     *
     * @return string nom de la table
     */
    public function tableName() {
        return 'user';
    }
}
```

```

/**
 * Règles de validation des attributs du modèle
 *
 * @return array règles de validation
 */
public function rules() {
    return array(
        array('email, password, dtells, dcardDefault, payment,
autoLink, firstname, lastname', 'required'),
        array('payment, autoLink', 'numerical',
'integerOnly'=>true),
        array('email, password, firstname, lastname,
lostPasswordKey', 'length', 'max'=>255),
        array('dtells', 'length', 'max'=>20),
        array('dcardDefault', 'length', 'max'=>10),
        array('typeInvit', 'length', 'max'=>5),
        array('typeInvit, lostPasswordKey', 'default',
'setOnEmpty' => true, 'value' => null),
        array('id, email, password, dtells, dcardDefault,
payment, autoLink, firstname, lastname, typeInvit, lostPasswordKey',
'safe', 'on'=>'search'),
    );
}

/**
 * Relations avec les autres modèles
 *
 * @return array modèles et caractéristiques des relations
 */
public function relations() {
    return array(
        'dcardBusiness' => array(self::HAS_ONE, 'DcardBusiness',
'user_id'),
        'dcardIndividual' => array(self::HAS_ONE,
'DcardIndividual', 'user_id'),
        'invitations' => array(self::HAS_MANY, 'Invitation',
'idUser'),
        'userLinks' => array(self::HAS_MANY, 'UserLink',
'idUser1'),
        'userLinks1' => array(self::HAS_MANY, 'UserLink',
'idUser2'),
        'userTokens' => array(self::HAS_MANY, 'UserToken',
'idUser'),
    );
}

```

```

/**
 * Fonction de recherche
 *
 * @return CActiveDataProvider données pour widget
 */
public function search() {
    $criteria = new CDbCriteria;

    $criteria->compare('id', $this->id);
    $criteria->compare('email', $this->email, true);
    $criteria->compare('password', $this->password, true);
    $criteria->compare('dtells', $this->dtells, true);
    $criteria->compare('dcardDefault', $this->dcardDefault, true);
    $criteria->compare('payment', $this->payment);
    $criteria->compare('autoLink', $this->autoLink);
    $criteria->compare('firstname', $this->firstname, true);
    $criteria->compare('lastname', $this->lastname, true);
    $criteria->compare('typeInvit', $this->typeInvit, true);
    $criteria->compare('lostPasswordKey', $this->lostPasswordKey,
true);

    return new CActiveDataProvider($this, array(
        'criteria' => $criteria,
    ));
}
}

```

2. Classe « user » de l'application Android

```
public class User {

    public User(LinkedHashMap<String, Object> user) {
        super();
        for (String key : user.keySet()) {
            Object val = user.get(key);

            if(key.equals("id"))
                this.setId((String) val);

            if(key.equals("email"))
                this.setEmail((String) val);

            if(key.equals("dtells"))
                this.setDtells((String) val);

            if(key.equals("dcardDefault"))
                this.setDcardDefault((String) val);

            if(key.equals("firstname"))
                this.setFirstname((String) val);

            if(key.equals("lastname"))
                this.setLastname((String) val);

            if(key.equals("typeInvit"))
                this.setTypeInvit((String) val);

            if(key.equals("autoLink"))
                this.setAutoLink((String) val);

            if(key.equals("dcardBusiness") && val != null)
                this.setDcardBusiness(new
DcardBusiness((LinkedHashMap<String, Object>) val));

            if(key.equals("dcardIndividual") && val != null)
                this.setDcardIndividual(new
DcardIndividual((LinkedHashMap<String, Object>) val));

            if(key.equals("dcardShopping") && val != null)
                this.setDcardShopping(new
DcardShopping((LinkedHashMap<String, Object>) val));
        }
    }
}
```

```
private String id;
private String email;
private String dtells;
private String dcardDefault;
private String firstname;
private String lastname;
private String typeInvit;
private String autoLink;
private DcardBusiness dcardBusiness;
private DcardIndividual dcardIndividual;
private DcardShopping dcardShopping;

public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

//Suite des getter et setter ...
}
```