

**Rapport de stage de fin d’études**

Présenté par

**Amine Darkaoui**

Pour obtenir le titre **d’ingénieur**

**En MATHÉMATIQUES APPLIQUÉES ET CALCUL SCIENTIFIQUE**

**------------------------------------------------------------**

**Développement d’une AppExchange sur Salesforce**

Touring Club Suisse

**------------------------------------------------------------**

Date de soutenance : xx xx 2015

***Jury***

|  |  |
| --- | --- |
| Monsieur | .. |
| Monsieur  Monsieur | ..  .. |

Stage de fin d’études effectué au sein de Capgemini

Sous la direction de M. Hichem KRICHEL, « CRM Project Manager »

Table des matières

[1. Remerciements 6](#_Toc408224669)

[2. Introduction 7](#_Toc408224670)

[3. Présentation de Capgemini 8](#_Toc408224671)

[3.1 Présentation de l’entreprise 8](#_Toc408224672)

[3.2 Historique de Capgemini 8](#_Toc408224673)

[3.3 Présence de Capgemini 9](#_Toc408224674)

[3.4 Les secteurs d’activités 10](#_Toc408224675)

[3.5 La BU Industrie & Distribution 11](#_Toc408224676)

[3.6 Présentation du TCS 11](#_Toc408224677)

[3.7 TCS au sein de Capgemini 12](#_Toc408224678)

[4. Contexte du stage 13](#_Toc408224679)

[4.1 Présentation du Cloud Computing 13](#_Toc408224680)

[4.2 Présentation des divers Cloud 14](#_Toc408224681)

[4.3 Présentation des technologies 17](#_Toc408224682)

[4.4 Contexte précis de travail 18](#_Toc408224683)

[4.5 Composition de l’équipe projet 19](#_Toc408224684)

[4.6 Rôles 19](#_Toc408224685)

[5. Aspects organisationnels 21](#_Toc408224686)

[5.1 Méthodologie du projet 21](#_Toc408224687)

[5.2 Planning de mon stage 21](#_Toc408224688)

[6. Travail réalisé 22](#_Toc408224689)

[6.1 Objectif 22](#_Toc408224690)

[6.2 Outils de développement 22](#_Toc408224691)

[6.3 Idée & but 22](#_Toc408224692)

[6.4 Mode basic 23](#_Toc408224693)

[6.5 Détails des objets en images 25](#_Toc408224694)

[6.6 Call2action 28](#_Toc408224695)

[7. Parcours de configuration 29](#_Toc408224696)

[7.1 Menu 360 View 29](#_Toc408224697)

[7.2 Configuration du menu 30](#_Toc408224698)

[7.3 Affichage du visuel 32](#_Toc408224699)

[7.4 Diagramme de la configuration 34](#_Toc408224700)

[8. Aspects techniques 35](#_Toc408224701)

[8.1 Bibliothèque de dessin 35](#_Toc408224702)

[8.2 Menu et Visualforce 37](#_Toc408224707)

[8.3 Requetes 38](#_Toc408224710)

[9. Mode étendu 40](#_Toc408224714)

[10. Charte graphique 41](#_Toc408224715)

[10.1 Logos 41](#_Toc408224716)

[11. Conclusion 42](#_Toc408224717)

[12. Références bibliographiques 43](#_Toc408224718)

[13. Glossaire 44](#_Toc408224719)

[14. Table des figures 45](#_Toc408224720)

[Annexe 46](#_Toc408224721)

[*15.* Annexe 1: Explication de la plateforme force.com 47](#_Toc408224722)

[16. Annexe 2: Méthode agile eXtreme Programming 51](#_Toc408224726)

[17. Annexe 3: Apercu des taches à faire 52](#_Toc408224727)

|  |  |  |  |
| --- | --- | --- | --- |
| **Information** | | | |
| **Référence** | **Nombre de page** | **Version** | **Du** |
| Rapport\_de\_stage\_Ingenieur\_Amine\_DARKAOUI |  | 1.0 | 20/12/2014 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Approbation du document** | | | |
| **Organisme ou entreprise** | **Nom (fonction)** | **Date** | **Visa** |
| Capgemini France | Hichem KRICHEL  (Chef de projet) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Diffusion** | | | | |
| **Destinataire** | **Organisme ou entreprise** | **Nombre** | **Pour action** | **Pour info** |
| Hichem KRICHEL |  | 1 |  | X |
| Service des Stages | Sup Galilée | 1 |  | X |
| Laurent Tournier | Tuteur | 1 | X | X |
| Emmanuel Audusse | Tuteur | 1 | X | X |
|  | Tuteur |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Mises à jour** | | | |
| **Version** | **Date** | **Auteur** | **Motifs** |
| 1.1 | xx/01/2015 | Amine DARKAOUI |  |

|  |
| --- |
| **Remarque** |
| Le présent document expose le rapport de stage effectué par Amine DARKAOUI au sein de Capgemini, dans l’entité Industries & Distribution. Le stagiaire a été impliqué sur le projet Dynamo pour le compte du Touring Club Suisse pendant une durée de six mois (du 01/07/2014 au 31/12/2014). |

## Remerciements

Au terme de cette expérience, je tiens à exprimer ma reconnaissance aux personnes qui m’ont aidé tout au long de mon stage.

Tout d’abord, je souhaite vivement remercier Fabien Bartolomucci qui m’a donné l’opportunité d’effectuer ce stage au sein de l’équipe TCS à Capgemini.

Je tiens également à remercier mes tuteurs Josselin Prost (Directeur du projet) Hichem Krichel (Chef de projet), Abdellah Boussaidi et Abdelhakim Irbah (Responsables Technique) pour m’avoir accueilli dans leur équipe, formé et encadré tout au long de mon stage. Ils m’ont été d’une grande aide et d’un soutien inestimable, et qui sans eux ma mission aurait été beaucoup plus complexe. Ils m’ont pris en charge, confiés des responsabilités et fait confiance pour que je mène à bien mon projet et que j’atteigne de façon autonome mes objectifs.

Je voudrais remercier les membres de l’équipe TCS : Stephane Kouadio, Amer Nasserdine, Amine Bendris, Mohamed Neti, Benjamin Lamotte, Alex Kamphet, Gauthier Becker. J’ai eu la chance d’intégrer une équipe jeune, dynamique et sympathique. Ils ont su être motivants, à disposition si j’avais des questions, encourageants et grâce à eux j’ai pu évoluer et m’épanouir au sein de cette équipe.

## Introduction

La troisième année du cycle ingénieur se conclut par un stage de fin d’étude de six mois dans une entreprise, cette étape va me permettre d’acquérir de véritables comportements d’ingénieurs, appliquer les notions apprises lors des trois années de formation et se confronter au milieu professionnel avec ses exigences et sa rigueur.

J’ai effectué mon stage de fin d’étude au sein de l’équipe CRM de l’entreprise Capgemini. L’aspect informatique et nouvelle technologie « Cloud computing » qui est géré par Salesforce m’intéressait et m’ouvrerait des perspectives nouvelles d’apprentissage dans un domaine en plein essor.

De plus le projet dans lequel j’ai été intégré présente l’opportunité de travailler avec l’assurance suisse TCS (Touring Club Suisse) ce qui m’a permis de voir le cycle de vie des produits proposés aux clients, le travail effectué par les différents offices (back, front) etc…

De nos jours, le Cloud Computing prend plus d’ampleur dans le domaine informatique. Ainsi plusieurs sociétés utilisent de telles applications pour rendre des services à leurs clients à travers le monde et comprendre, anticiper ses besoins pour lui proposer des offres qui lui conviennent. Les entreprises sont de plus en plus exigeantes puisqu’elles veulent accéder à l’information non seulement rapidement, mais facilement, à l'endroit et au moment qu'ils ont choisi.

TCS n’a pas échappé à cette exigence et a lancé un défi majeur pour revoir ses processus de vente, centraliser les données clients et refaire tout son système d’information pour répondre aux besoins de ses clients et surtout pour contrer la rude concurrence entre les différentes entreprises qui cherchent à fidéliser et conquérir de nouveaux clients et marchés.

L’objectif de mon stage à Capgemini est donc de participer aux étapes de l’implémentation d’une « solution CRM » pour le client TCS en utilisant la plateforme Salesforce. J’ai commencé mon stage par monter en compétence grâce à une formation en ligne. Ensuite, on m’a confié plus de responsabilités au fil de mon stage.

Dans un premier temps, je décrirai le cadre de ma mission, les nombreux outils utilisés dans le projet, l’organisation de celui ci et ensuite je passerai au sujet de mon stage en décrivant la partie fonctionnelle, technique et finir par une conclusion sur les apports personnels et techniques que le stage m’a apporté.

## Présentation de Capgemini

### fiche_Cap.png Présentation de l’entreprise

Capgemini est la première société de services en ingénierie informatique (SSII / SS2I/ ESN) en France dans le secteur du conseil en management, des services informatiques et de l’infogérance. Elle est la 13e SSII mondiale en 2013. Implanté dans 44 pays, le groupe offre un large éventail de compétences qu’il sait mettre en œuvre de manière cohérente.

En coopération avec ses clients, Capgemini contribue à l’élaboration de leur orientation stratégique, à sa mise en œuvre et les aide à tirer le meilleur parti de la technologie.

Pour eux, elle prend en charge la gestion de leurs processus opérationnels et de leurs infrastructures informatiques, et cela en associant ses compétences en matière de technologies et de gestion des opérations.

Figure : Fiche d’identité (source : Wikipedia)

### Historique de Capgemini

|  |  |  |  |
| --- | --- | --- | --- |
| 1967 | Création de SoGETI par Serge KAMPF à Grenoble | 1996 | adoption d’un nouveau nom et d’un nouveau logo:Cap**Gemini. Et l’**Ouverture de Capgemini Asia à Singapour. |
|  |  |  |  |
| 1973 | **CAP Gemini Sogeti** :Fusion de Sogeti, Computer Systems et CAP | 2000 | Acquisition de la branche Consulting d’Ernst & Young pour former Cap Gemini Ernst & Young |
|  |  |  |  |
| 1985 | Cotation de Cap Gemini Sogeti à la bourse de Paris | 2007 | Acquisition de **Kanbay International**, une SSII indienne et la création de **Capgemini Technology Services Maroc** |
|  |  |  |  |
| 1991 | Création de la **société Gemini Consulting**  après la fusion de cinq entreprises de conseil en gestion | 2010 | Capgemini TS Maroc lance son activité Infrastructure Management (activité Sogeti). |

### Présence de Capgemini

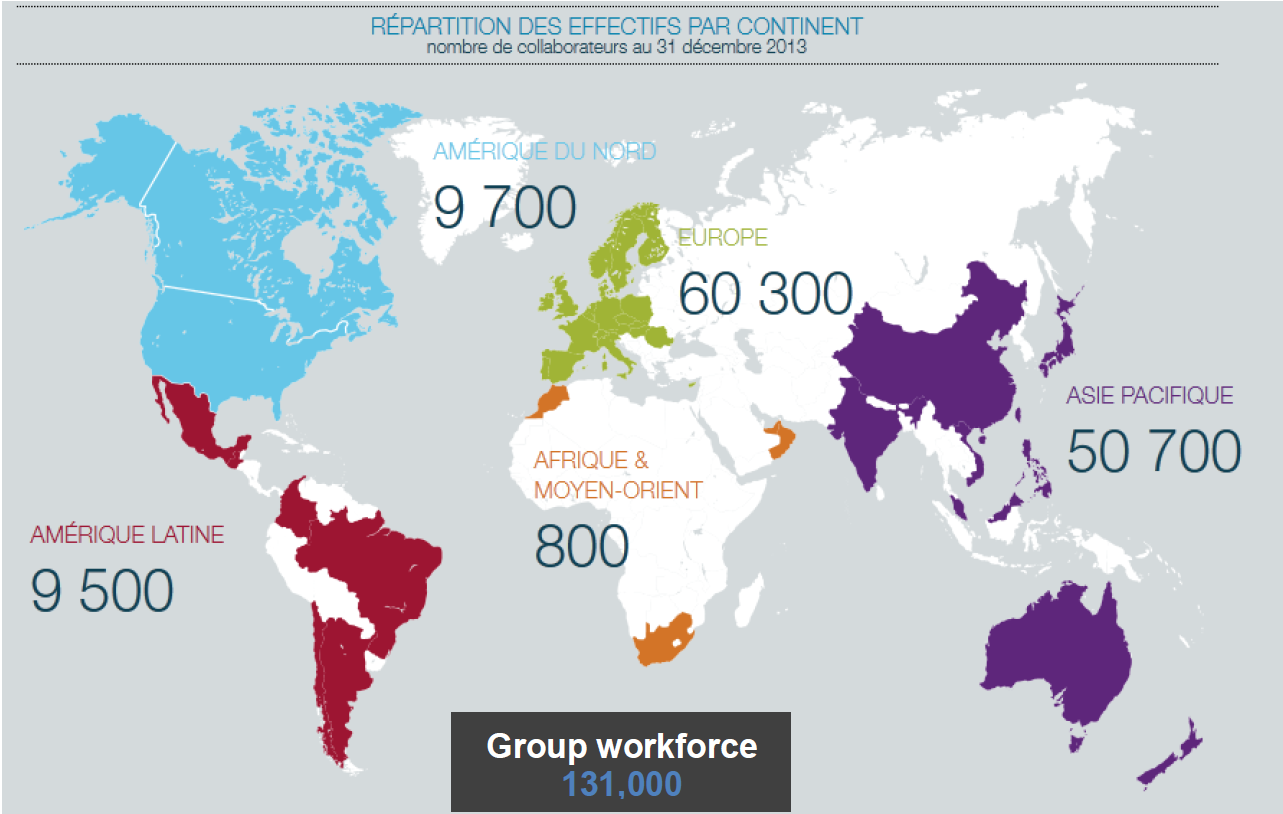


Figure : Implémantation de Capgemini dans le monde au 31 déc. 2013 (Source : Documentation Capgemini)

Au niveau mondial

Avec une présence dans plus de 30 pays, Capgemini dispose de ressources dans le monde entier.

De plus il est intéressant de noter une présence exponentielle dans les pays offshore, notamment en Inde et au Brésil.

Au niveau national

Capgemini est présente principalement en ile de France avec 3117 collaborateurs, en deuxième position arrive le Sud Est avec 1487, en troisième l’Ouest avec 927, le Sud Ouest avec 872 et en dernier le Nord Est avec 564.

### Les secteurs d’activités



Figure : Secteurs d’activités de Capgemini

Telecom, Média, Entertainment :

Les avancées technologiques de ces secteurs ont entrainé une modification radicale du comportement des clients. C’est pourquoi Capgemini aide les entreprises à s’adapter à l’évolution rapide de leur environnement commercial.

Manufacturing :

Avec la diminution du pouvoir d’achat et l’augmentation du e-commerce, les clients recherchent davantage les prix les plus bas qu’auparavant et les conditions de ventes sont plus difficiles.

Capgemini propose des outils de ciblage des clients et de partage des informations, basés sur les dernières technologies, qui permettent aux entreprises du secteur d’être plus efficace et de prendre moins de risques.

Grande consommation, Distribution, Transport :

Comme pour le Manufacturing, la crise touche aussi ces secteurs. Le coût des matières premières augmente et le prix des produits varie trop souvent.

A nouveau, les innovations technologiques permettent de fidéliser le client en lui proposant des offres sur mesure et en lui facilitant l’accès aux marchandises.

Le transport est lui aussi optimisé grâce à des solutions de suivi logistique, de gestion des entrepôts, de synchronisation des étapes, etc.

Banque, Assurances, Services Financiers :

Les banques et les compagnies d’assurance doivent constamment évoluer pour se maintenir à un niveau de compétitivité élevé. Ces évolutions – qui vont de la mise à disposition de nouveaux supports d’accès (Smartphone, internet,...) ou de nouveaux moyens de paiement à la fusion d’établissements – nécessitent une importante phase de transformation des systèmes d’information. C’est à cette étape que Capgemini intervient.

Services publics :

Capgemini accompagne les services publics comme les impôts, la santé, etc. dans leur transformation grâce à ses nombreux domaines d’expertises.

Energie, Utilities, Chimie :

L’instabilité des prix des matières premières et de la demande pétrolière ainsi que la problématique du développement durable obligent ces secteurs à être très réactifs dans leur évolution. Capgemini travaille déjà avec les 10 premières sociétés d’Utilities du monde et emploie plus de 12 000 consultants spécialisés à travers le monde.

### La BU Industrie & Distribution

La BU Industrie & Distribution regroupe plusieurs secteurs d’activités (comme l’industrie, la chimie…).La répartition de cette BU se fait en fonction des compétences des solutions (SAP, Siebel, CRM…).

Cette nouvelle division donne naissance à des « Skill Centers » localisés par pays.

Voici ici la répartition des « Skill Centers » de la BU Industrie & Distribution France :

Figure : Skill Center de la BU Industrie & Distribution

### Présentation du TCS

TCS (Touring Club suisse) est une organisation Suisse proposant des services d’assistance aux personnes et aux véhicules. Ils gèrent une importante clientèle (1,6 millions de membres TCS soit 1 habitant sur 5).

Le domaine d’activité du TCS est varié comme l’assistance & sociétariat l’assistance aux véhicules, la protection juridique et les services financiers …

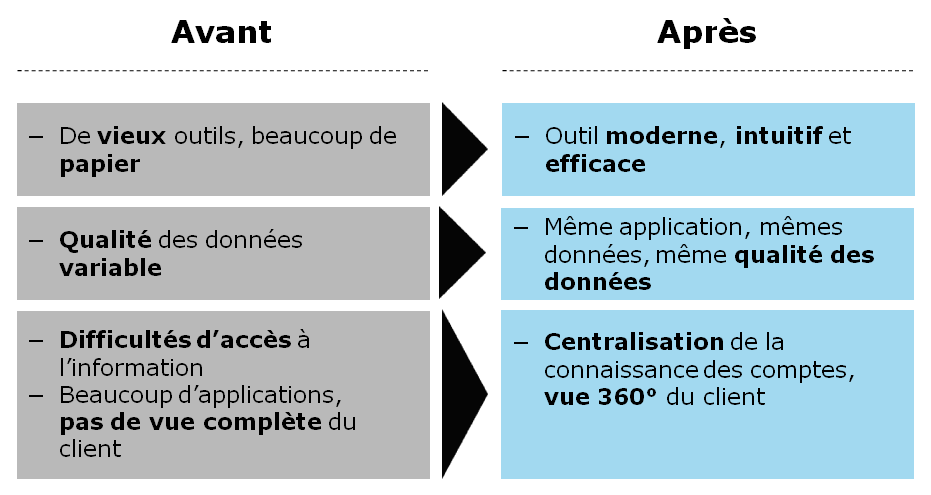
Le TCS souhaite développer sa stratégie de croissance et de modernisation afin d’augmenter le taux d’acquisition de nouveaux clients, de mettre en place des stratégies de vente ainsi que des programmes de fidélisation des clients dans un contexte économique où la concurrence se fait sentir au fil du temps. Pour atteindre ces objectifs, Capgemini a répondu à l’appel d’offre pour réaliser ce projet qui sera découpé en plusieurs parties, c’est ce que nous appelons des **Lots**.

Figure : Siège du TCS

Le projet TCS a débuté en 2011. Il y a eu le lot 1 et le lot 2. Actuellement nous sommes en période de **Tierce Maintenance Applicative**(TMA) du lot 2. Un lot 3 est en préparation, qui proposera notamment une évolution de la solution Cloud Marketing (campagnes marketing).

L’Objectif du projet est de mettre en place une solution centralisant toutes les données client, permettant l’optimisation du conseil, du marketing, de la vente et de la gestion des produits. Le défi majeur pour le TCS est de « Gérer au mieux la complexité de son organisation au profit de ses membres ».

Voici quelques évolutions que permet d’avoir les solutions cloud (Salesforce).



### TCS au sein de Capgemini

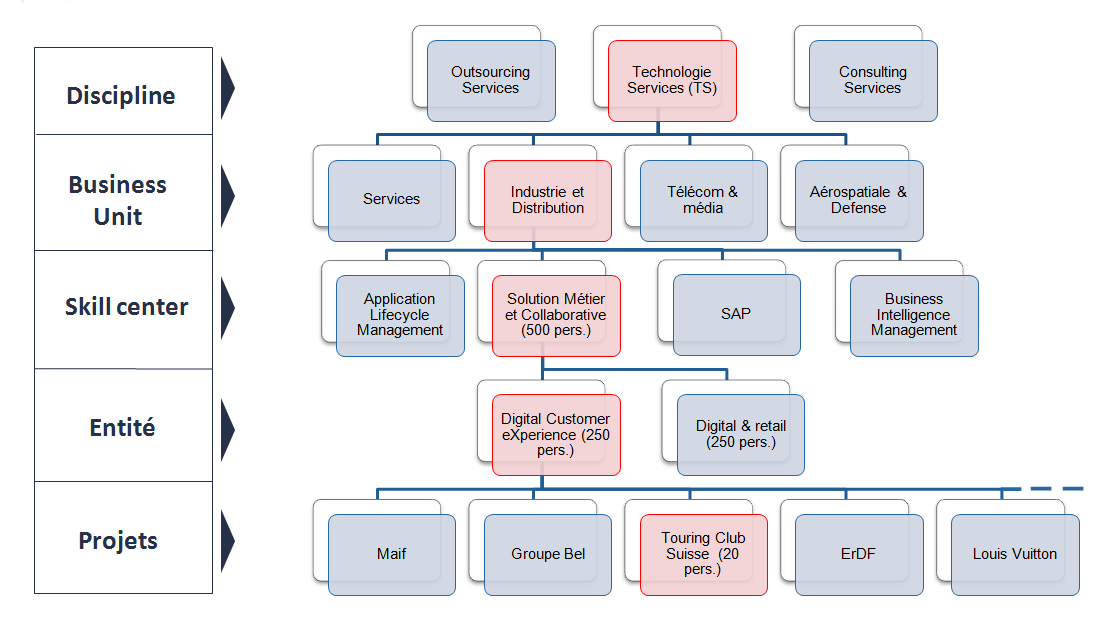


Figure : Diagramme Capgemini

## Contexte du stage

J’ai effectué mon stage dans le monde du Cloud Computing au sein d’une équipe CRM Salesforce dans un projet pour le Client TCS (Touring Club Suisse).

J’étais intégré à l’équipe en tant que consultant junior travaillant sur un sujet pour le skill.

### Présentation du Cloud Computing

Une entreprise qui souhaite héberger ses systèmes d’informations dans ses propres locaux doit faire face à un certain nombre d’inconvénients. L’entreprise doit posséder une infrastructure adaptée et souvent imposante, sans compter les outils informatiques indispensable pour la gestion et la sécurisation des données et les dispositifs matériels sécuritaires à mettre en place pour limiter les dégâts en cas de problèmes (incendie, tremblement de terre, vol, etc.). Elle doit investir également dans une variété de logiciels. De plus, cela nécessite une équipe d’experts spécialisés dans l’installation, la configuration, les tests, l’exécution, la sécurité, la mise à jour, ce qui augmente d’autant plus le coût et la complexité.

Le Cloud Computing ("Informatique dans le nuage") permet aux entreprises de s’en décharger pour se focaliser sur leur propre travail.

Le Cloud Computing est une technologie permettant de délocaliser les données et les applications sur des infrastructures dématérialisées accessibles depuis Internet.

Cette notion désigne un ensemble d’applications complémentaires déployé chez un fournisseur expérimenté – qui gère la maintenance, la sécurité, la mise à jour et le stockage des infrastructures – et accessible via Internet. Présentée sous forme de package minimal opérationnel, une solution Cloud est ensuite adaptée au besoin du client par ajout de fonctionnalités, augmentation des capacités de stockage, etc.

L’adoption du Cloud Computing en entreprises vient avec son lot d’avantages, et également avec son lot d’inconvénients. Il reste aux entreprises de trouver la solution idéale et la plus adéquate à leur besoin, en matière de fonctionnalités, coût, technologie.

**Avantages du Cloud Computing:**

* Effort financier limité au départ.
* Mutualisation du matériel permettant d'optimiser les coûts.
* Temps de déploiement de la solution rapide.
* Modularité (capacité à faire évoluer une installation).
* Maintenance et prise en compte permanente des évolutions logicielles.
* Facilement réversible sans conséquences financières lourdes.
* The Green Cloud (économie d’énergie et bilan carbone réduit).

**Inconvénient du Cloud Computing:**

* Besoin obligatoire d’un accès Internet.
* Dépendance à la qualité de réseau.
* Sécurité.
* Localisation des données (une fois dans le Cloud, on ne sait plus à quel endroit elles sont présentes physiquement).
* Isolement des données (dans le cas d’une concurrence d’accès par exemple).

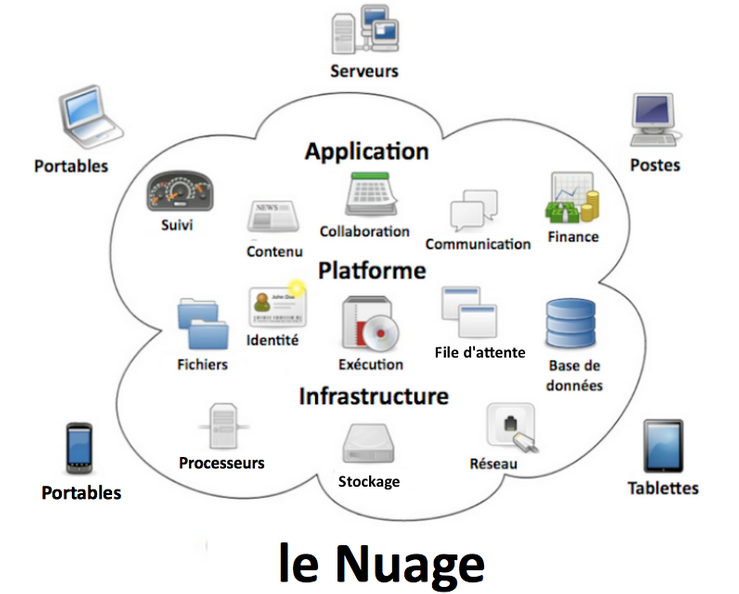


Figure : Principe du cloud computing

### 

### Présentation des divers Cloud

1. * 1. Salesforce

Salesforce est une entreprise de Cloud Computing proposant des plateformes d'applications selon quatre types d'offres :

* Sales Cloud, la solution la plus connue pour le marketing
* Service Cloud, un service client amélioré à coûts réduits
* Force.com, la solution la plus personnalisable proposant des outils de développement
* Chatter, la collaboration d'entreprise en temps réel avec la sécurité de Salesforce.

La solution la plus utilisée sur le projet est Force.com, puisqu'elle permet d'adapter la plateforme par rapport aux besoins du client et de développeur des applications grâce à un langage de programmation que la plateforme propose.

* + 1. Force.com

Cette solution intègre un mode de conception à la fois visuel par glisser-déposer mais aussi des API pour faire communiquer les applications entre elles.

Force.com est une solution idéale pour les applications qui :

* Stockent des données relationnelles structurées ;
* Implémentent des règles métiers ;
* Peuvent être affichées depuis un mobile ;
* Requièrent l'intégration d'un autre système ;
* Nécessitent des temps de réponses cours ;
* Doivent être sécurisées ;
* Ont des résultats qui doivent être analysés, vérifiés et conformes.

Les données sont stockées sous forme d’objet. Il existe des objets dits "standards" que sont par exemple les Comptes, les Contacts... et les objets "personnalisés" qui sont créés de zéro suivant les besoins.

Ces objets contiennent plusieurs informations :

* Des champs standards et personnalisés, comme le nom ou le prénom d'un Contact
* Des déclencheurs, ou plutôt les actions qui doivent être réalisées après certaines manipulation comme une insertion, une mise à jour
* Des présentations de pages qui sont les différentes pages à montrer à l'utilisateur lorsque celui-ci se rend sur une instance de l'objet en question
* etc.

Du point de vue sécurité, il n'est normalement pas possible d'accéder à des webservices externes. Pour le rendre possible, il faut spécifier spécifiquement les informations de connexion au site distant.

Pour un développeur Force.com, tout peut se faire à travers le site web.

En effet, celui-ci propose un panneau latéral de configuration ainsi qu'une console du développeur.

Le panneau permet de tout gérer, que ce soit les classes, les objets standards et personnalisés, les accès aux sites distants, les constantes, les déploiements, les batchs, ... La liste est plutôt longue.

En plus de cela, il nous est mis à disposition ce qu'ils ont appelé la "Console du développeur". Il s'agit d'une fenêtre web dans lequel il est possible de faire plus ou moins ce qu'on veut du point de vue développement.

On peut par exemple exécuter des requêtes, lancer des batchs ou tout simplement tester du code Apex, notion que nous verrons par la suite.

Cependant, pour tout ce qui concerne les accès à la base de données, il existe un outil web plus complet et pratique configuré pour Salesforce.

* + 1. Workbench

Workbench est un outil web pour les administrateurs et développeurs voulant interagir avec Salesforce.com.

Il facilite beaucoup les opérations CRUD (Create, Read, Update, Delete) : l'écriture de requêtes mais aussi les opérations de mise à jour, d'insertion et de suppression en masse sans passer par du code.

En effet, pour l'écriture des requêtes, il suffit de sélectionner un objet pour que tous les champs qu'il contienne soit visible.

Il est bien sûr possible de modifier le contenu de la clause WHERE suivant différents critères, de classer les résultats ou encore de mettre une limite au nombre de résultats retournés.

De plus, en requêtant l'ID unique de l'objet, il est possible de se rendre directement sur sa page sur Salesforce.com via un simple clic.

Il est aussi possible de demander les résultats des requêtes sous forme d'un CSV, ce qui permet ensuite de faire des mises à jour ou suppression en masse via l'interface fournie par Workbench.

* + 1. Zuora

Zuora est une société qui fournit des logiciels d'entreprise sur le cloud et qui se concentre sur tout se qui touche au « billing » (paiement). Cette solution facilite le traitement des factures et fournit une API permettant de récupérer les informations rattachées à celle-ci.

L'API propose toute une panoplie d'objets servant spécifiquement à tout ce qui est facturation : compte, souscription, facture, ligne de facture, produit, etc.

Zuora donne donc la possibilité de :

* Gérer les comptes de facturation
* Consulter le catalogue produit
* Gérer les souscriptions
* Effectuer des opérations sur les factures
* Administrer
* Synchroniser des données avec la plateforme avec Salesforce, par des appels web service
* Comme avec Salesforce, effectuer des requêtes dans le langage ZOQL (Zuora Object Query Language).
  + 1. Boomi

Boomi est une plateforme d'intégration multicloud qui permet le développement de processus utilisant des fonctionnalités de différents clouds par un système de glisser-déposé.

Il permet ainsi d'aller requérir des informations via des webservices proposés par les divers clouds afin de les traiter ensemble et d'en produire un document contenant diverses informations provenant de deux clouds différents par exemple.

Dans le cadre du projet Dynamo, Boomi va être le point central des échanges entre les Salesforce, Zuora, Caméléon et Aprimo.

Personnellement, je n’ai pas été amené à utiliser Boomi, mais j’ai pu voir la façon dont les processus ont été réalisés.

Ces processus sont un ensemble d'éléments, ou nœuds, reliés les uns aux autres. Un élément non relié à un autre ne fait pas parti du processus.

À l'entrée d'un nœud, il y a en général un "document" qui est transmis par le nœud précédent. Un "document" peut être de différents formats, ou profil, à savoir XML, "flat file" (CSV par exemple).

Les nœuds peuvent être de diverses natures :

* Un connecteur vers un Cloud
* Une condition si... alors... sinon...
* Une branche, i.e. une entrée puis différentes sorties
* Un appel à un autre processus
* Une mise en cache
* Une opération spécifique (fusion, division, script)
* Un point d'arrêt

Le placement de ces nœuds et des liens se fait entièrement à la souris par glisser-déposer.

Le déploiement se fait simplement sur Boomi. Il suffit d'aller sur la page prévue pour cela, sélectionner son processus et déployer.

La différence avec, par exemple, Salesforce.com est qu'il n'est pas possible de modifier la version déployée puisqu'il n'est pas possible de modifier la version déployée sans passer par un nouveau déploiement.

Boomi permet aussi d'avoir une vision simple des différents "atomes" ou environnement à disposition, ainsi que les processus qui y sont installés.

Il est aussi possible d'exécuter manuellement un processus déployé et de surveiller son état.

### Présentation des technologies

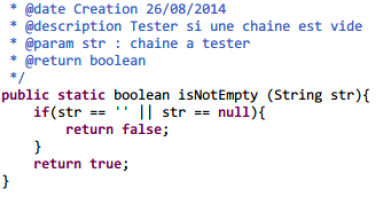
* 1. 1. APEX

Apex est un langage propriétaire ressemblant fortement à Java pour la plateforme Force.com.   
Ce langage permet d'ajouter des fonctionnalités que Force.com ne possède pas de base, comme par exemple écrire des déclencheurs de base de données pour actionner une tache, des contrôleurs...

Il est fortement intégré avec la base de données et les langages de requêtes (SOQL) et possède un bon support aux web services.

Sa sauvegarde se passe uniquement du côté du serveur et n'est pas stockée sur la machine locale (sauf, bien entendu, si le code a d'abord été écrit sur la machine).

Les classes de tests (classes pour couvrir le code et vérifier si les méthodes retournent le résultat attendu) sont écrites avec le langage Apex. Le projet ne pourra pas passer en production tant que le taux de couverture des tests n'est pas supérieur à 75%. Ceci indique l'importance de faire des tests sur les classes produites.



* + 1. VisualForce

Est Framework qui permet aux développeurs de construire des pages web (simples ou complexes), qui s’exécutent nativement sur la plate-forme Force.com. Ce Framework intègre un langage de balisage similaire au HTML. Dans le langage Visualforce, chaque balise correspond à un composant d’interface (bouton, colonne, image…).

Le comportement des composants peut être contrôlé par la même logique qui est utilisée dans les pages standards, où les développeurs peuvent y associer leur propre logique avec une classe contrôleur écrite en Apex ou à travers un script en Javascript. On peut utiliser les balises HTML déjà existantes, car le rendu est toujours en HTML, les balises Visualforce sont traduites dans une phase supplémentaire en balise HTML.

* + 1. Requêtes SOQL et ZOQL

SOQL signifie Salesforce Object Query Language et ZOQL signifie Zuora Object Query Language.

Ces requêtes permettent d'aller chercher des informations dans les objets présents sur Salesforce ou Zuora.

Leur syntaxe ressemble aux requêtes SQL, avec les clauses SELECT, FROM et WHERE.

Exemple d’une requête SOQL :

**SELECT Id FROM Account WHERE Name = ‘Amine’**

* + 1. Sublime

L'environnement de développement utilisé pour tout ce qui touche à Apex, Visualforce et les autres langages de programmation que j’ai utilisé.

Il existe un plugin qui vient se greffer à Sublime afin de pouvoir accéder aux fichiers sauvegardés sur l’environnement de travail.   
Il permet notamment l'ajout des menus suivants :

* Sauvegarde vers le serveur distant,
* Comparaison de code source avec le serveur,
* Exécution de tests, ...

Cependant, comme indiqué plus haut, le code Apex est gardé sur le serveur, et nécessite donc pour sa compilation de passer par le serveur.

### Contexte précis de travail

En arrivant en stage le 1 Juillet 2014 chez Capgemini Industries et Distribution, j’ai participé à une demi-journée d’accueil des stagiaires. Durant cette demi-journée, j’ai assisté à la présentation du groupe et j’ai fait la rencontre de l’équipe dirigeante et de l’équipe des ressources humaines.

Comme tout nouveau collaborateur Capgemini, j’ai reçu toutes les ressources nécessaires pour intégrer mon projet immédiatement. Cela comprend :

**Ressources matérielles :** badge d’accès aux locaux Capgemini Paris, ordinateur portable (avec souris, écran supplémentaire, câble de sécurité), bureau dans l’espace de travail avec l’équipe projet, téléphone de bureau et casier.

**Ressources logicielles :** Identifiant me permettant à d’accéder au différentes SandBox pour pouvoir développer sous Salesforce.

**Documentation :** accès au répertoire de partage de documents de l’équipe projet via l’outil Collabnet-Team forge. Cet outil est accessible depuis l’intranet de Capgemini. L’ensemble de la documentation du projet est accessible et à disposition des membres de l’équipe.

Par ailleurs, l’intranet de Capgemini permet l’accès à toutes les informations concernant l’organisation du groupe, les différents services (support, documentation utilisateur) et téléchargement d’outils logiciels.

### Composition de l’équipe projet

L’équipe avec laquelle j’allais travailler était composée de 2 stagiaires et 14 consultants répartis comme suit : un directeur de projet, un chef de projet, trois consultants fonctionnels, six consultants techniques, un architecte, et deux responsable de data patch. A la fin du stage deux nouveaux consultants et un stagiaire ont rejoint le projet.

Figure : Répartition des consultants sur le projet Dynamo (TCS)

**Remarque :**

Le schéma ci-dessus illustre la composition de l’équipe au moment où j’ai intégré le projet. Le schéma n’a pas de signification hiérarchique.

### Rôles

|  |  |
| --- | --- |
| Rôle | Tâches et responsabilités |
| Chef de projet | * Suivi du projet * Affectation d’activités aux membres de l’équipe * Communication avec le client * Animation des réunions mensuelles. |
| Responsable technique | * Coordination du travail de l’équipe technique * Vérification de la qualité du travail technique * Responsable de la résolution des anomalies techniques * Développement (Salesforce, Zuora, Cameleon, Boomi, SQL) * Préparation des packages de livraison au client (Code source, documentation technique, documentation d’administration) * Rédaction de documents d’administration de l’application |
| Consultant technique | * Développement Salesforce, Zuora, Caméleon, Boomi, base de données (SQL pour l’équipe datapatch) * Résolution d’anomalies techniques * Rédaction de documents de Spécifications Techniques Détaillées (STD) * Rédaction de documents de Conception Technique Détaillée (CTD) * Exécution de tests * Déploiement des développements. |
| Responsable fonctionnel | * Animation des ateliers de cadrage avec le client et rédaction des comptes rendus * Coordination du travail de l’équipe fonctionnelle * Responsable de la qualité des Spécifications Fonctionnelles Détaillées (SFD) et des cas de tests * Rédaction des SFD * Rédaction des plans de test et cas de test * Rédaction des documents de synthèse des tests * Exécution de tests |
| Consultant fonctionnel | * Rédaction des SFD * Rédaction des plans de test et cas de test * Exécution de tests * Rédaction des comptes rendus d’ateliers de cadrage |
| Architecte | * Conception de l’architecture de l’application * Préparation des environnements (développement, test, recette) * Rédaction de documents (Dossier d’Architecture Technique, document d’exploitation de l’application |
| Responsable DataPatch | * Assurer la cohérence des données dans la base. * Développement des datapatch dans le but de corriger les anomalies de données. |

## Aspects organisationnels

### Méthodologie du projet

Le *Scrum* est une méthode agile dédiée à la gestion de projets.

Elle se base sur un cycle itératif de développement. Le projet est ainsi découpé en « sprint », chacune d'une durée d'environ deux semaines.

Cependant, ce n'est pas le seul aspect inhérent à la méthodologie *Scrum*, il y a aussi les réunions journalières et après chaque sprint pour faire le point sur ce qui a été réalisé/doit être réalisé.

Il y a aussi un  *backlog*  ou carnet du produit. Il s'agit d'une liste des fonctionnalités à réaliser suivant les besoins du client. Dans le cadre du projet que j'ai intégré, le *backlog* avait été décidé avant mon arrivée et nous avons fait des réunions journalières d'une quinzaine de minutes durant lesquelles on décrivait brièvement ce qu'on avait fait la veille, sur quoi on va travailler durant la journée et les points bloquants.

Ces réunions se basaient sur des tâches qui nous ont été assignées par le leader technique. Ces tâches étaient écrites sur des post-it sur lesquels figuraient le nom de la tâche, la durée estimée et la durée réalisée.

J’avais aussi des réunions bimensuelles avec le responsable du projet pour définir les nouveaux besoins, valider l’état d’avancement et faire des demos de l’application que j’ai eu à développer.

### Planning de mon stage

Tout au long de mon stage, j’ai mis en place une organisation personnelle pour mener à bien mes tâches et assurer la qualité de mon travail. Pour cela, j’ai mis en place une liste des tâches avec des priorités, une estimation des charges et une date limite pour chaque tâche. Cette liste des tâches a été mise à jour régulièrement en fonction des priorités sur le projet et des nouvelles tâches qui m’étaient affectées. Parallèlement à cela, j’ai défini un planning de ces tâches pour avoir une vision plus générale des tâches et contrôler mon avancement sur mon travail. Le diagramme de Gantt Dans l’annexe permet de visionner les différentes étapes du stage. Voir **annexe 3**

## Travail réalisé

### Objectif

L’objectif de cette partie est de présenter l’application AppExchange qui sera développée sur la plateforme.

La suite de ce document présente :

* L’idée et la motivation du projet.
* Les objectifs.
* Les premiers aperçus.
* Les étapes à réaliser, step by step, pour la réalisation de l’appli.
* Manuel d’utilisation de l’application et quelques démonstrations.

### Outils de développement

L’application a été réalisée à l’aide des langages HTML, CSS et JavaScript / jQuery pour la partie graphique. J’ai utilisé ensuite Apex,Visualforce, SOQL pour l’intégration à la plateforme Salesforce et la requête des données.

360 View est compatible avec tous les navigateurs.

### Idée & but

|  |
| --- |
| BUT |
| Créer une application hébergée sur le web qui ne nécessite aucun logiciels ni matériels afin de répondre à un des 5 axes de développement de Capgemini: Proposer des offres innovantes et assurer à Capgemini une présence et une visibilité sur le marché. |
| La plateforme offre des applications avec un choix de mode:   * « Basic » gratuit et très attractif avec des fonctions standards * « Etendu » ou le client fait appel à Capgemini pour entendre les fonctionnalités. |
| L’Appexchange 360 View va donc offrir ces deux modes qui seront présentés dans le document. |

* L’objectif est multiple :
  + Donner un avantage compétitif à Capgemini autour de Salesforce avec un produit utilisable dans n’importe quelle démo ou avant vente.
  + Donner de la visibilité supplémentaire à Capgemini sur le marché.
  + Générer des nouveaux revenus de licence.
  + Générer des nouveaux revenus de prestation (nouveau prospects, service pour étendre la licence basic).

La solution 360View est une application qui sera présente sur le store « Appechange » de Salesforce.

360View est une application innovante permettant depuis une plateforme Salesforce de consulter et de gérer de manière rapide, intuitive et unifiée l’ensemble des objets standards (client, compte, opportunités, activités…) contenues dans le système d’information des entreprises (CRM, BI…).

Puissante et très efficace, l’application 360 View permet d’accroitre la productivité des forces de vente sur le terrain et centraliser des informations qui nécessitent beaucoup de temps à consulter séparément. Adaptée à tous les métiers (Commercial, RH, …), 360View permet d’accéder à la base de données en temps réel

***« Accroissez votre efficacité grâce à la puissance et la simplicité de 360View sur votre organisation. »***

### Mode basic

Dans la licence basic, 360 va se concentrer sur les objets « horizontaux de Salesforce » à savoir :

* Account
* Contact
* Case
* Opportunities



Chaque vue sera complètement paramétrable via un **menu** pour que l’application colle au mieux aux attentes de l’utilisateur. Au sein de chaque objet les champs affichés sont au choix parmi un ensemble qu’on va présenter.

Les objets (bulle) de l’application sont divisés en deux blocks :



* La partie bleue contient le logo de l’objet et son titre.
  + Type du logo : Image.
  + Type du titre : Texte.
* La partie blanche contient les informations relatives à un contact et configuré par l’utilisateur à travers un menu.

L’objet « Contact » est un objet à part puisqu’il présente plus de fonctionnalités à l’utilisateur à travers plusieurs boutons.

**Contact** : Divisé en trois parties

* La première contient le logo du customer et le titre de la bulle.
* La deuxième contient les champs de l’objet qui ont été choisis lors de la configuration.
* La troisième contient des boutons « call2action » qu’on va détailler plus tard dans ce document.

1. * 1. Présentation du mode

L’application proposera un navigateur visuel 360° d’un objet, pour l’exemple qu’on va présenter par la suite, le Contact/Customer sera au centre de la vue, les objets autour sont choisis pour illustrer les informations reliés au contact.

Dans Salesforce, il existe différents types de relation entre les objets. Leurs différences résident dans leur mode de gestion de la suppression des données, du partage et des champs obligatoires dans les présentations de page.

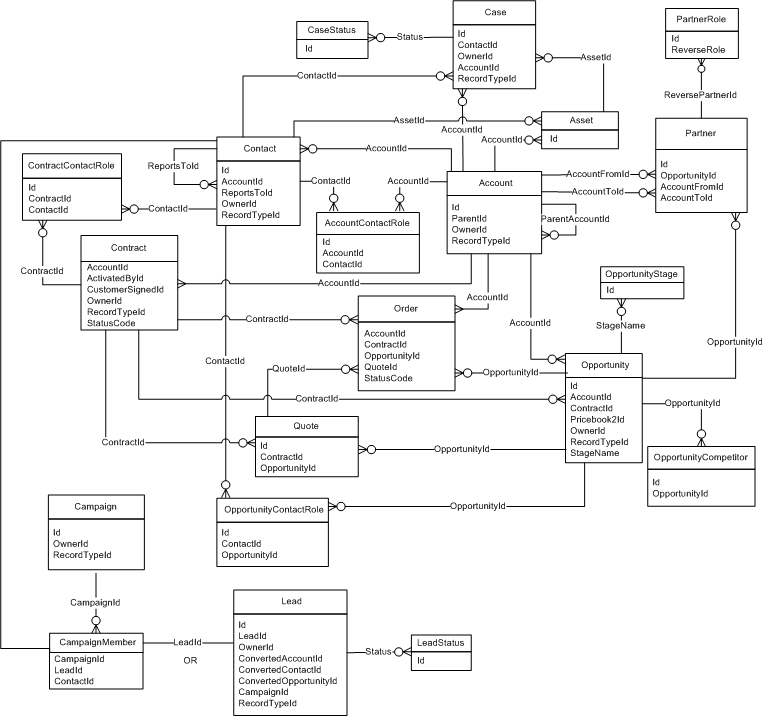
* + 1. Relation entre les objets

Dans notre modèle nous allons utiliser la relation *Lookup*. Ce type de relation relie étroitement deux objets de sorte que l'enregistrement principal contrôle certains comportements des enregistrements de détails et de sous-détails. Par exemple, vous pouvez définir une relation *Lookup* entre deux objets, telle que Compte-Rapport sur les dépenses, et élargir la relation à des enregistrements de détail secondaires, tels que Compte-Rapport sur les dépenses-Élément de ligne de dépense. Vous pouvez ensuite appliquer des opérations à la relation principal-détails-sous-détails.

Les comportements de relations *Lookup* comprennent :

* La suppression d'un enregistrement *Lookup* le déplace vers la Corbeille et laisse l'enregistrement principal inchangé. La suppression d'un enregistrement *Lookup* supprime également les enregistrements de détails et de sous-détails associés. L'annulation de la suppression d'un enregistrement de détail entraîne sa restauration. L'annulation de la suppression d'un enregistrement principal annule également la suppression des enregistrements de détail et de sous-détail associés. Cependant, si vous supprimez un enregistrement de détail, puis ultérieurement et séparément, son enregistrement principal, vous ne pouvez pas annuler la suppression de l'enregistrement de détail, car n'a plus d'enregistrement principal associé.
* Par défaut, les enregistrements ne peuvent pas être réapparentés dans des relations principal-détails. Pour des enregistrements enfant de relations principal-détails dans des objets personnalisés, les administrateurs peuvent toutefois autoriser une nouvelle relation parente avec d'autres enregistrements, en sélectionnant l'option Autoriser une nouvelle relation parente dans la définition de la relation principal-détails.
* Les paramètres de sécurité définis pour l'enregistrement principal s'appliquent aux enregistrements de détails et de sous-détails.
* Le champ de relation *Lookup* (qui correspond au champ reliant les deux objets) doit figurer dans la présentation de page des enregistrements de détails et de sous-détails.
* L'objet principal peut être un objet standard, par exemple un Compte ou une Opportunité, ou un objet personnalisé.

Voici un aperçu des relations et les champs qui pointent vers chaque objet pour le model standard.



**Figure 6 : Relations objets standards Salesforce**

Le schéma représente les objets standards et quelques uns de leurs champs. Un objet peut contenir plusieurs fils et un fils et obligatoirement reliés à un parent. Ces règles sont très importantes quand il s’agit de faire des requêtes et des sous-requêtes pour récupérer les objets fils qui sont autour de la bulle centrale.

### Détails des objets en images

* + 1. Account

Le contact est relié à l’objet « Account » de type compte B2B ou B2C. Dans l’exemple que je vais présenter, j’ai choisi des champs qui illustrent des informations pertinentes qu’on peut requêter.



Nombre de contacts. Le visuel affiche dans un premier temps leur nombre pour y accéder il faut cliquer dessus. L’objet est sous forme de lien

Type du contact

Nom de l’entreprise à laquelle le compte est rattaché

* + 1. Notes



Nombre de notes enregistrés sur la base.

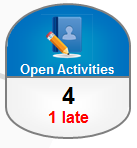
* + 1. Opportunities



Montant des opportunités

Statut des opportunités en cours :

* Ouverte/Gagnée/Perdue



Nombre des activités du contact en cours avec un détail sur leurs statuts

Statuts :

* *Late*: s’affichera en rouge si l’activité a dépassé sa date d’exécution.
* *Last closed*: Le nombre de jours de la dernière activité close (permet de garder un œil sur l’historique des activités).

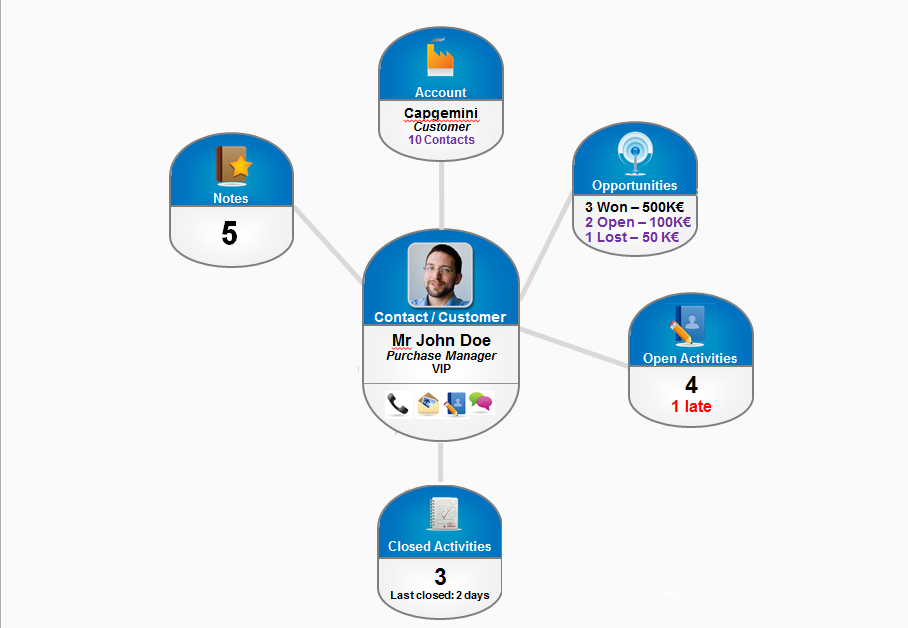


Figure : Vue du visuel

L’exemple ci-dessus montre une version basic, on peut voir le client au centre avec les principales informations relatives à son compte. Les informations sont réparties sur 5 bulles.

Les bulles autour sont à choisir par l’utilisateur. Cette option peut être paramétrable via un menu qui prendra proposera un choix de champs et d’objets fils ainsi que des filtres sur des champs pour affiner les affichages.

La bulle ne pourra pas contenir toutes les informations de la configuration (le détail des opportunités si elles sont nombreuses, les descriptions des notes si elles sont au format *rich text* etc…). Salesforce propose des options de navigation que nous allons expliquer dans la partie suivante.

* Une mini fenêtre « pop up » (cf. exemple ci-dessous).
  + La page comportera tout les détails avec des liens qui redirigerons l’utilisateur vers l’opportunité, le compte relié à celle-ci etc… (en ce qui concerne cet exemple).

La page sera placée en dessous de l’appli de manière compacte (maximum 10 lignes) pour un meilleur rendu.

**L’accès aux comptes deviendra à la fois simple et très rapide.**

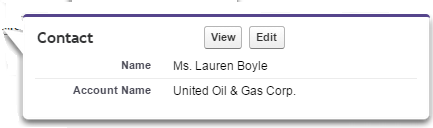


Figure : Popup lors du survol des bulles

* Le navigateur offre au maximum trois niveaux de visionnage.
  + Le niveau 0 qui affiche le contact comme sur l’exemple ci-dessus
  + Le niveau 1 est composé des principaux objets salesforce (Opportunities, account,...).

### Call2action

Les « call2action » sont des boutons illustrés qui permettent de rapidement créer des records ou lancer des fonctions (appeler, envoyer un email, prendre rdv, chatter, …)

Nous choisirons des « call2action » qui seront positionnés sur la l’objet central « Contact ». La bulle sera donc divisée en trois parties comme cela a été mentionné plus haut.

Nous mettrons donc quatre boutons qui nous aiderons donc à naviguer plus rapidement.

* Bouton pour avoir le numéro du contact et l’appeler ou organiser un rendez-vous téléphonique.
* Bouton pour envoyer un mail.
* Bouton pour prendre un rendez-vous.
* Bouton pour chatter si le contact est en ligne.

L’exemple ci-dessous montre à quoi cette partie ressemble.

lookup.png

Figure : Boutons Call2action

## Parcours de configuration

Pour afficher un visuel donné (comme l’exemple présenté ci-dessus), il faut passer par une phase de paramétrage qui proposera à l’utilisateur les différents objets Salesforce, leurs fils ainsi que leurs champs. Chaque configuration donnera lieu à un résultat différent. Nous allons dans cette partie présenter dans le détail le menu de l’application.

### Menu 360 View

* + 1. Controller Salesforce

Un contrôleur Visualforce est un ensemble d'instructions qui spécifient ce qui arrive quand un utilisateur interagit avec les composants spécifiés dans le balisage Visualforce, comme quand un utilisateur clique sur un bouton ou un lien. Les contrôleurs donnent également accès à des données qui doivent être affiché dans une page (les données de la base des clients, une liste des opportunités etc…), et peuvent modifier le comportement d’un composant.

La plate-forme Force.com fournit un certain nombre de contrôleurs classiques qui contiennent les mêmes fonctionnalités et logique qui sont utilisés pour les pages Salesforce standard. Par exemple, si vous utilisez le contrôleur des comptes standards, en cliquant sur un bouton Enregistrer dans une page Visualforce le résultat est le même que quand on clique sur « Enregistrer » sur une page compte d'édition standard.

Il existe un contrôleur standard pour chaque objet Salesforce qui peut être interrogé en utilisant l'API Force.com.

L’intégration d’un contrôleur standard se fait comme suit :



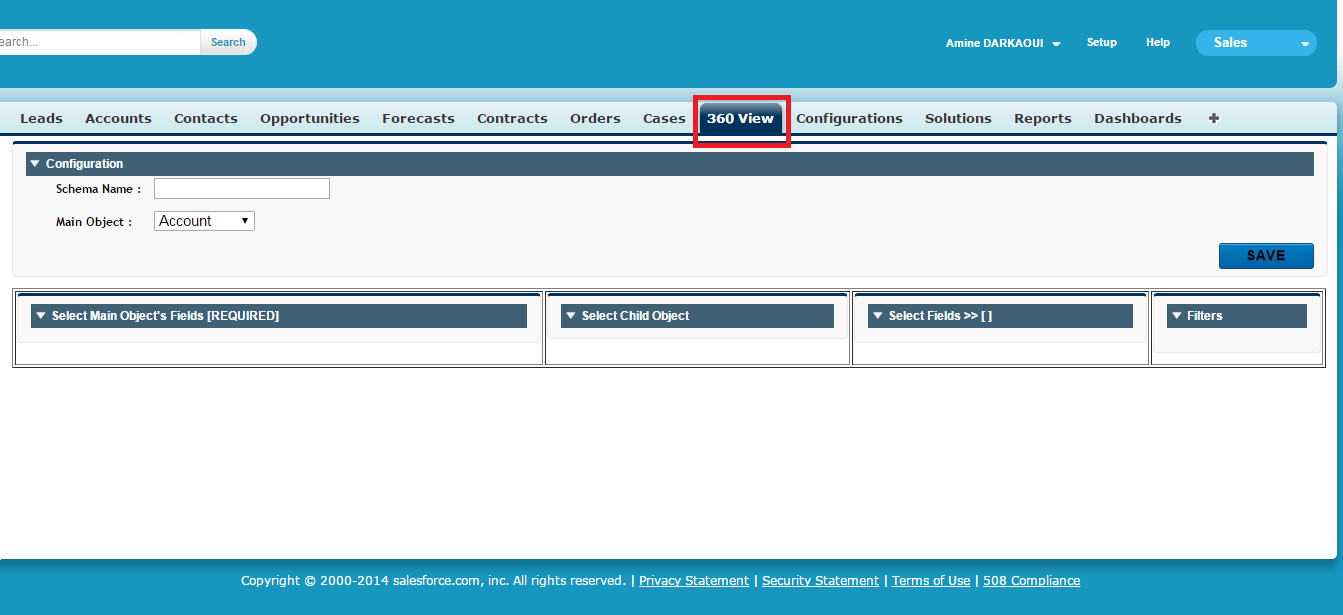
Dans le cas du menu de l’application, je n’ai pas eu recours à un contrôleur standard mais un à plusieurs que j’ai écrit pour exécuter les différentes taches coté serveur dont j’avais besoin.

### Configuration du menu

* + 1. Step 1 : Configuration dans le menu 360 VIEW :

Le menu est sous forme d’une page Visualforce accessible via un bouton « 360 View » sur la page d’accueil. Il est composé de plusieurs parties comme le présente la capture d’écran ci-dessus.

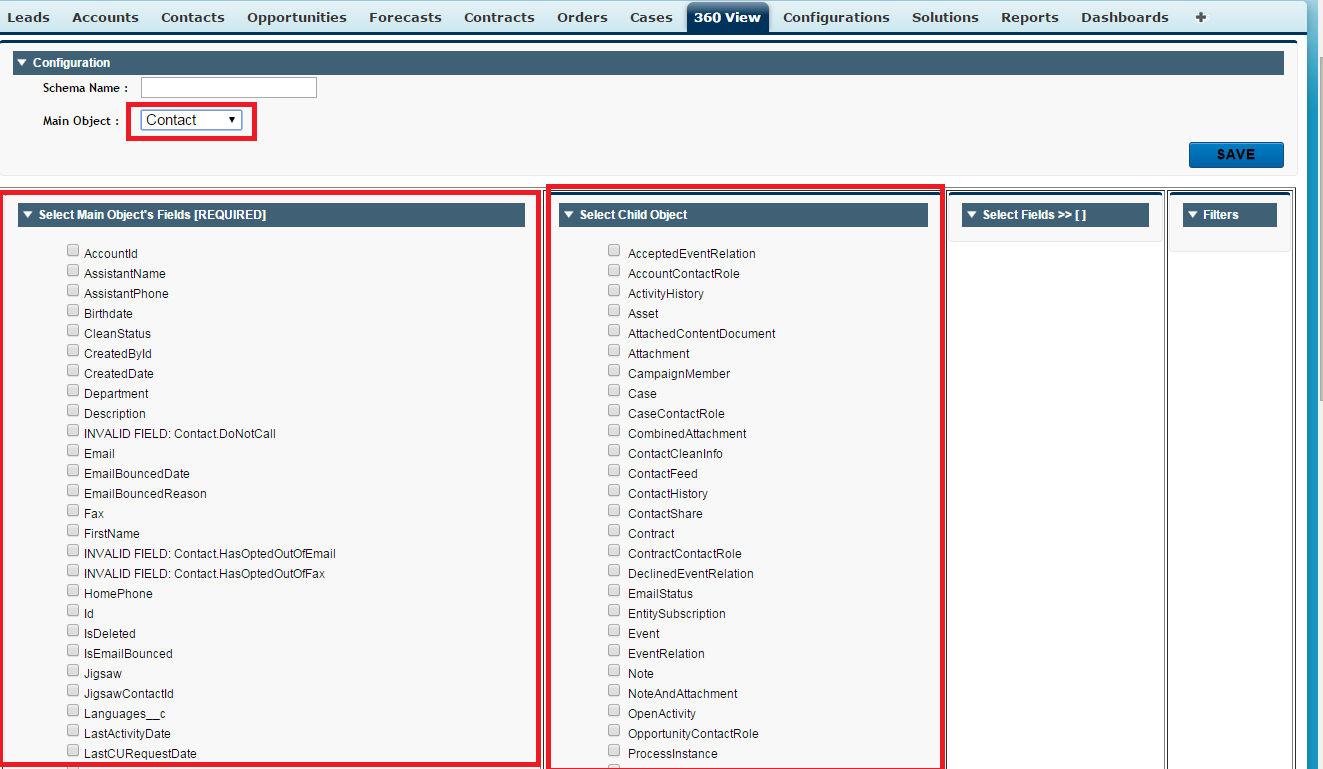
1. Nom du schéma : L’utilisateur doit entrer un nom pour sa configuration, il devra être unique. Si l’utilisateur ne saisit pas un nom pour le schéma un message sous forme de fenetre en JavaScript s’affichera pour rappeler cela et bloquer la sauvegarde des données.
2. L’objet central : Sous forme d’une *select list* qui est composée d’un des objets standards de Salesforce. Nous lui avons ajouté un évenement (onchange) en JavaScript qui se déclenche si sa valeur d’entrée change.

Cet événement conditionne l’affichage des blocks 1 et 2 via des méthodes qui sont écrite dans le controlleur et qui pour un objet donné, retournent sa description compléte qui contient les champs, les objets fils et plusieurs propriétés encore.

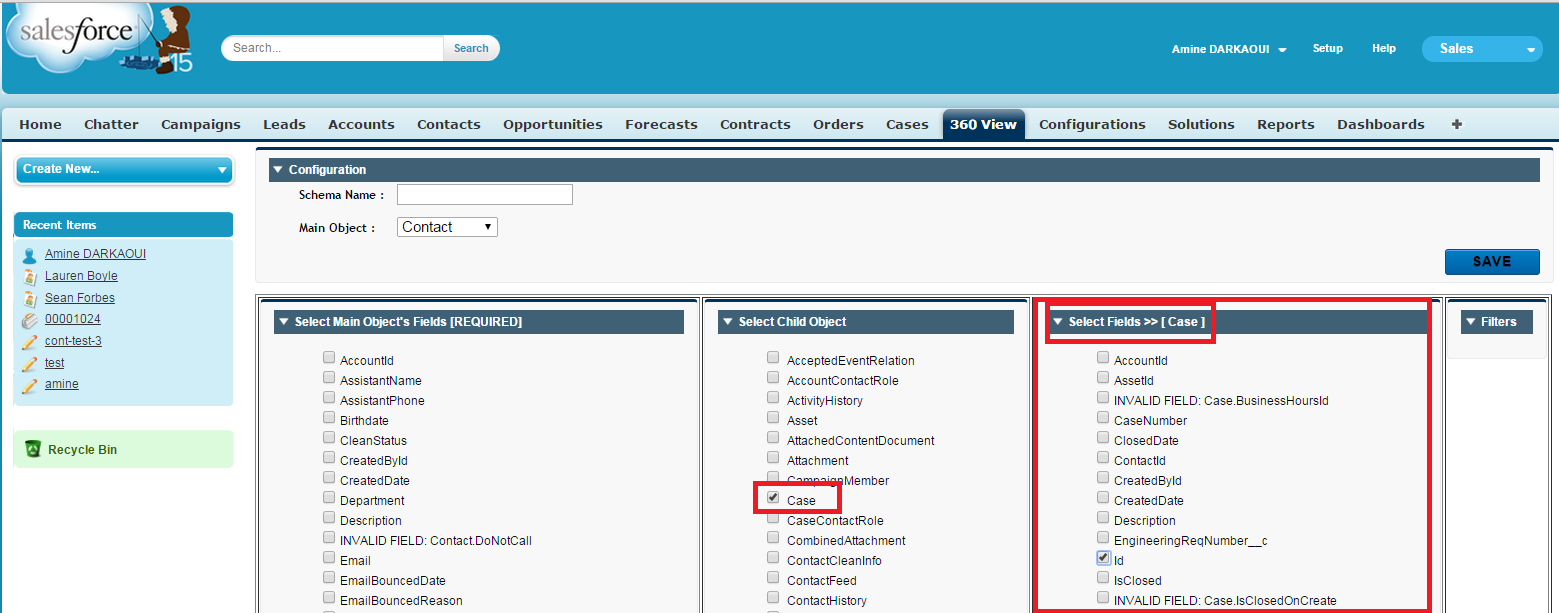
* + 1. Step 2 : Choix de l’objet central :

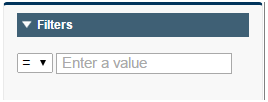
Le choix de l’objet central actualise deux sections :

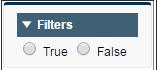
* La première pour choisir les champs à afficher pour l’objet central (**au moins un champ à cocher**)
  + Possibilité de cliquer sur le nom du champ ou la checkbox pour effectuer un choix.
* La deuxième section liste les objets fils de l’objet central choisi
  + Cliquer sur le nom des fils pour en choisir un. (**clique sur un nom d’un des fils n’actionne pas la checkbox**)

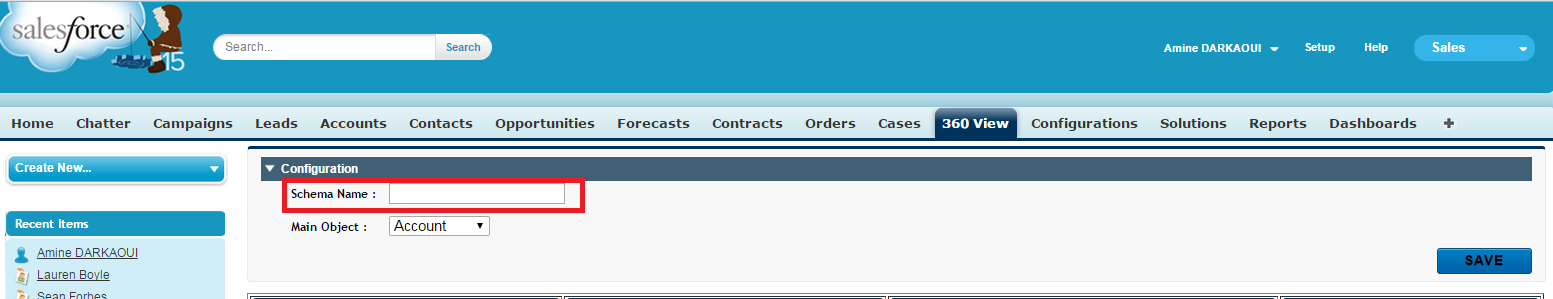


* + 1. Step 3 : Champs des objets fils
* Cliquer sur le nom de l’objet fils rafraîchit la troisième section qui affiche les champs de chaque fils.
* Le nom de l’objet fils est affiché dynamiquement dans le titre de la section des champs.
* Quand on clique sur un champ de l’objet fils, la checkbox du champ s’active et avec le nom du fils. Quand il n y a plus de champ coché, l’objet fils se décoche.
* Cliquer sur un champ de l’objet fils peut aussi rafraichir la quatrième section qui affiche des filtres en fonction du type du champ.
  + Exemple : Champ de type date va donner lieu à un filtre de type date et pareil pour Currency, Boolean, picklist (afficher les valeurs de la picklist).



* + 1. Step 4 : Paramétrage des filtres
* Une fois le champ de l’objet fils est coché, on peut d’affiner le résultat en cochant une des cases des filtres ou saisir une valeur, en fonction du type de filtre proposé. Les trois filtres ci-dessus sont un exemple du choix proposé ; le premier est sur les valeurs du type *Boolean* le second pour les *picklist* et le troisième sur les *Integer/Decimal*.

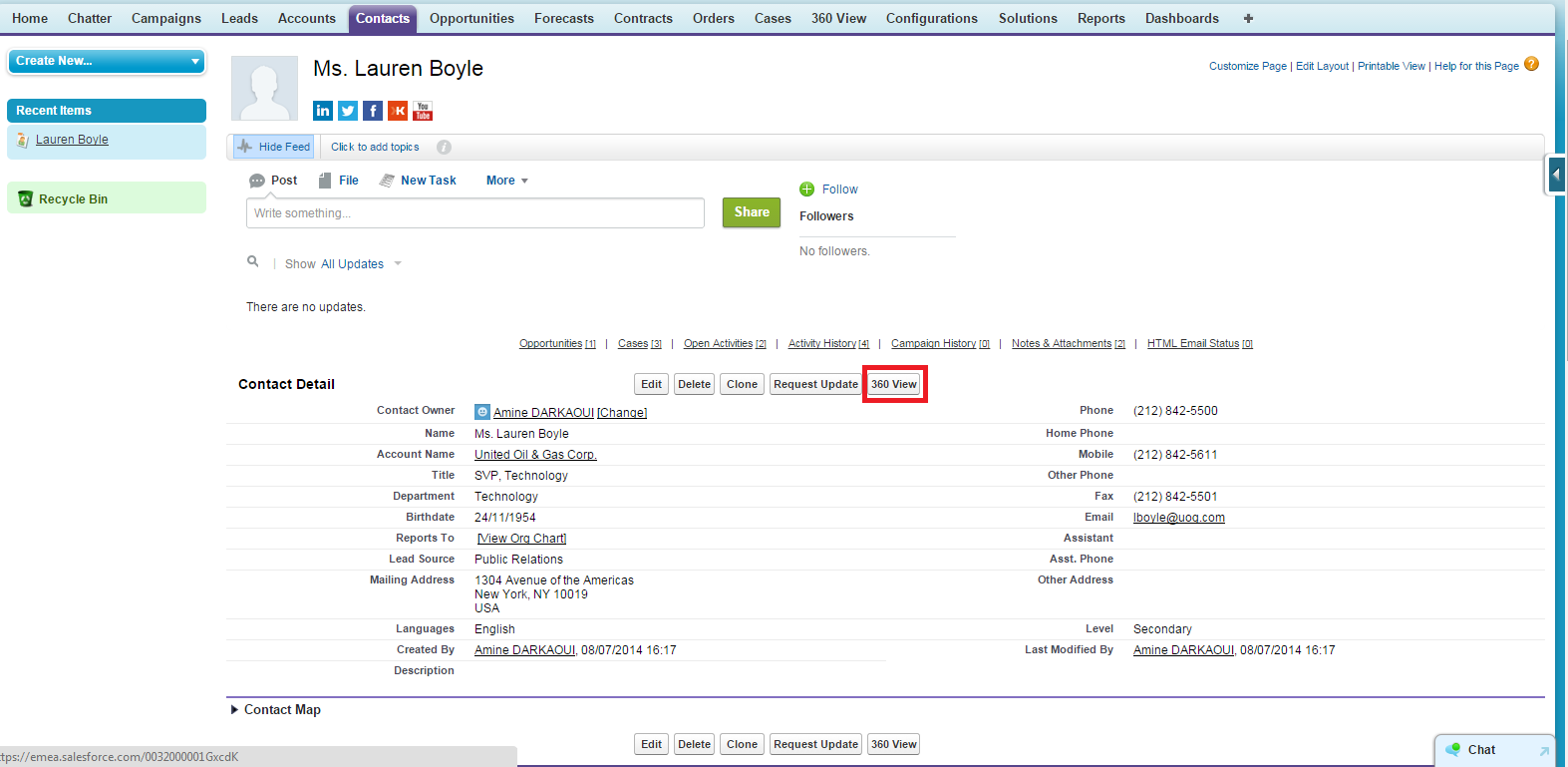
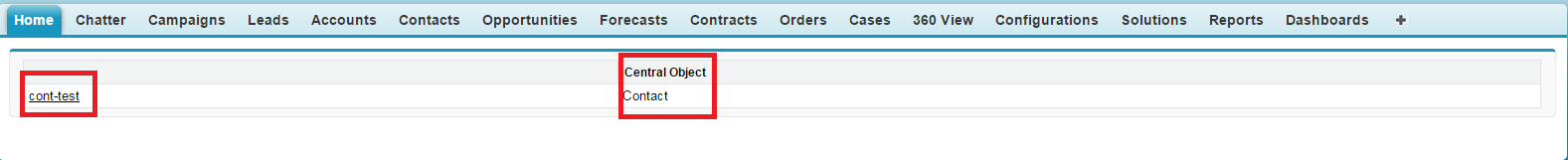


* + 1. Step 5 : Nom du schéma
* Saisir le nom du schéma pour qu’il soit enregistré dans l’objet **Configurations**.

### Affichage du visuel

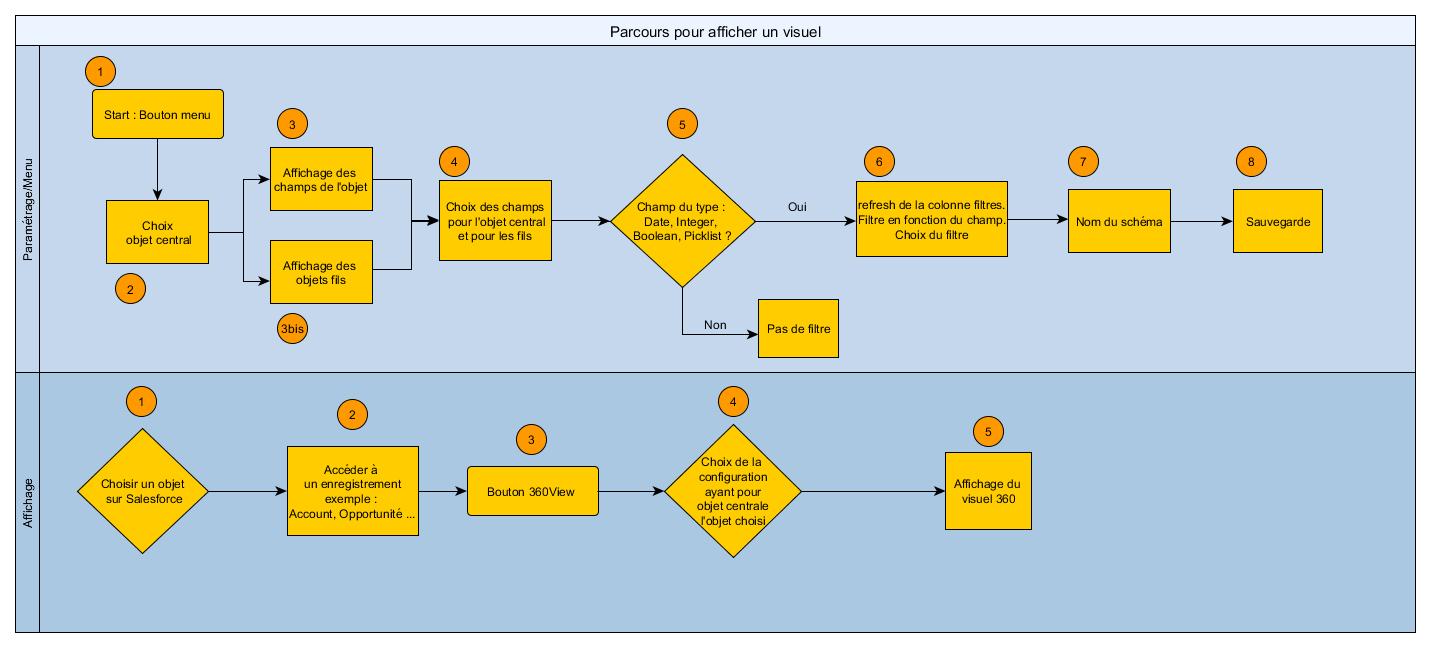
Une fois que la configuration à travers le menu est faite, le visuel est stocké nous pouvons l’afficher en choisissant un objet central.

Si par exemple on a configuré un visuel qui a pour objet central l’objet « Contact »



* Il faut aller dans un contact pour lequel on veut afficher les informations.
* Cliquer sur le lien 360 View.
* Le lien va rediriger vers une page contenant des liens de configurations qui ont pour objet central « Contact ».
* Cliquer sur la configuration désirée.

### Diagramme de la configuration



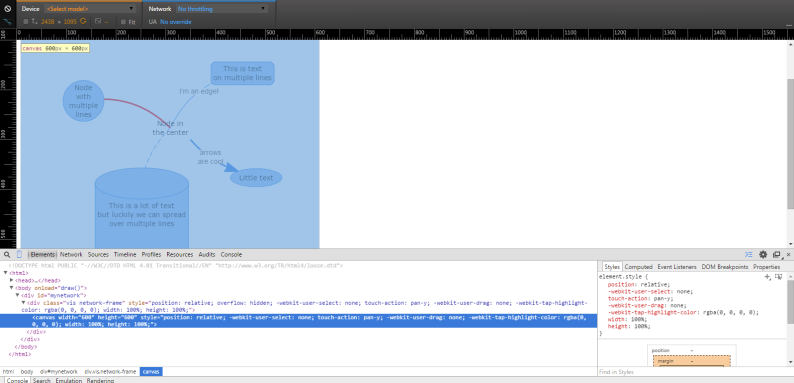
## Aspects techniques

La conception de l’Appexchange décrites précédemment a fait intervenir plusieurs langages de programmation, nous allons décrire brièvement le travail technique sous jacent à chaque étape.

### Bibliothèque de dessin

La première partie du projet a été de trouver une bibliothèque web permettant de faire les dessins des bulles, flèches et qui peut contenir des informations passée à l’aide de fonctions en JavaScript ou directement dans le DOM. J’ai passé plusieurs semaines à chercher, le nombre de bibliothèques de dessin disponibles sur internet est gigantesque, il fallait faire le tri pour trouver celle qui permettait d’avoir le plus de contrôle sur chaque élément du visuel. Un exemple pour illustrer ce point se trouve ci-dessous.

Le visuel de la figure xx est le résultat de bibliothèque « vis.js », cette dernière permet d’avoir des nœuds, bulles et les liaisons qui vont avec contenant du texte mais pas d’images (premier inconvenant) mais le principal problème est que le dessin généré est sous un format spécial *« canvas »* qui est un composant de HTML qui permet d’effectuer des rendus dynamiques d’images via des scripts.

Comme c’est indiqué dans la figure xx tout le visuel est comprimé dans une seule balise de type canvas ce qui restreint considérablement les modifications qui peuvent être faites. Ainsi pour accéder à une bulle pour y insérer

des informations sur un objet une fois que la requête ait renvoyé un résultat est impossible ou par exemple y ajouter des liens hypertexte etc…

Beaucoup de bibliothèques proposent des fonctionnalités de dessin très développées (diagrammes, UML, graphes en 2d…) mais qui n’étaient pas adaptés au travail que je devais faire, on peut citer :

# Collapsible Force Layout.

# Gojs

# Cytoscape

# J’ai finalement utilisé la bibliothèque «[html5-node-diagram](https://github.com/boundary/html5-node-diagram) ». La bibliothèque est facile d’utilisation, elle est écrite en JavaScript et permet de définir des nœuds qui contiennent n’importe quelle sorte balise en HTML. La définition d’un nœud se fait dans un fichier JavaScript qui sera téléchargé (upload) en tant que pièce jointe du projet et va être interprété par la plateforme Force.com.

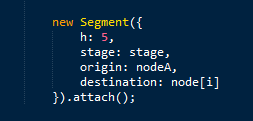
La plateforme Force.com peut interpréter beaucoup de langages de programmation, on peut citer des exemples comme Javascript, jQuery, CSS, Java…

Cette fonctionnalité est très utile pour développer des projets complexes en utilisant des bibliothèques annexes ou des scripts écrits dans des langages plus adaptés.

Dans ce qui suit, nous allons présenter des bouts de codes dans les différents langages utilisés et nous essayerons de les commenter pour voir ce qu’il y a derrière chaque partie présenté precedemment.

# 

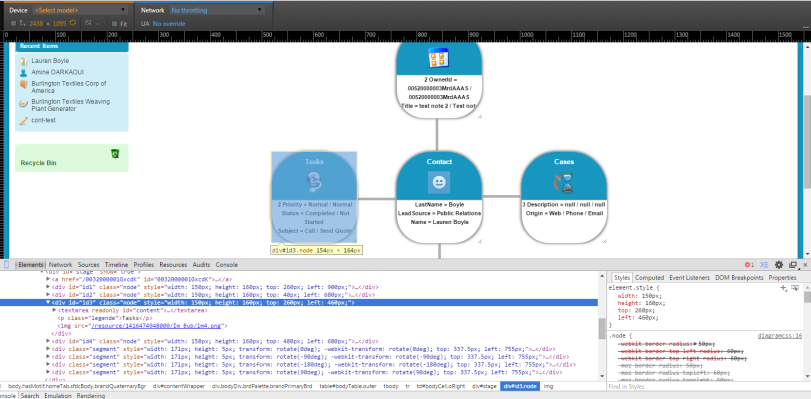
* On définit comme variables globales les dimensions des NODE.
* On créé une variable nodeA qui est une instance de l’objet Nœud et on remplit les champs de l’objet.
  + Titre : Pas nécessaire dans notre cas
  + Stage : La balise qui va contenir tout les nœuds et fleches de liaisons
  + Les dimensions
  + X,Y les coordonnées de la bulle sur la page web
  + Events : événements liés à la bulle (click, double clique, survol…).



Une fois les bulles définies on peu les lier entre elles à travers l’objet segment qui prend en plusieurs arguments :

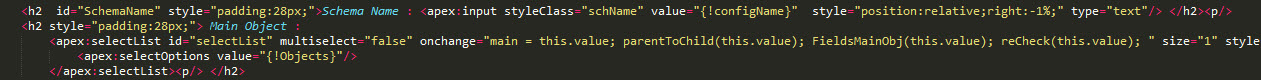
* Stage : cité ci-dessus
* Origin et la destination : les deux nodes à relier**.**

Le résultat est sous forme de bulles reliées entre elles comme présenté ci-dessous, complètement paramétrables auxquelles on peut ajouter des balises HTML.



### Menu et Visualforce

Afin de réaliser le menu qui affiche les champs des objets sur Salesforce et les objets fils de l’objet central ainsi que les filtres en fonctions du type du champ, j’ai du recourir à plusieurs fonctions de descriptions des objets sur salesforce. Le processus d’affichage suit les points suivants :

1. Choix de l’objet central par l’utilisateur je le sauvegarde grâce à une fonction JavaScript (onchange).
2. Transmettre au controlleur l’objet central.

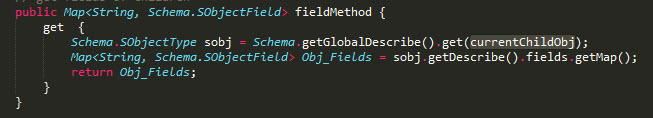
Assigner l’objet à cette variable coté controlleur (passage client-serveur)

Détecter le changement dans la liste et sauvegarder l’objet dans la variable main

Appelle à la fonction qui renvoie les champs d’un objet

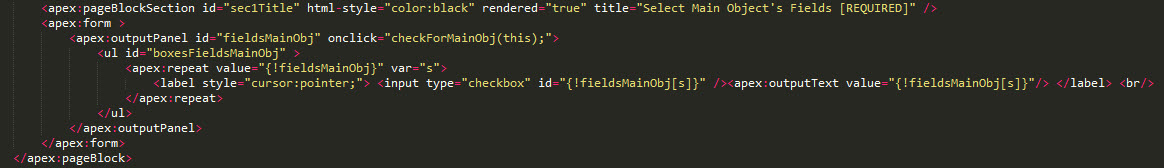
code2.jpg

1. Grâce à deux fonctions de description (voir ci-dessous) j’obtiens les champs de l’objet passé en paramètre et tous les objets qui ont pour relation parent-enfant.



Fonctions de description sur Saleforce

Objet saisi par l’utilisateur et sauvegardé

1. Une fois que les variables des « champs » et « objet fils » remplis je les récupère coté page vfp et je les affiche grâce à une balise boucle (*apex:repeat*).

* **d**

Checkbox

Variable de boucle

Fonctions de description sur Saleforce

### Requetes

### Une les différents champs et objets fils sont affichés, il faut récupérer en temps réel le choix de l’utilisateur, le stocker sous forme de chaine de caractère pour avoir ensuite la liste des configurations et pouvoir faire des requêtes en SOQL et ainsi afficher le visuel.

### La récupération de cases cochées se fait en JavaScript, les éléments cochés sont stockés dans deux map.  Une map est une collection qui fonctionne avec un couple clé – valeur. C’est une sorte de tableau 2 dimensions sauf que la clé est unique et de taille 1 alors que la valeur peut être un tableau. Le schéma de stockage est donc le suivant :

* Pour l’objet central et ses fils, le stockage se fait sur deux map « mainObjMap » et « childObjMap ». La clé prend le nom de l’objet et chaque clé correspond une valeur qui est les noms des champs.

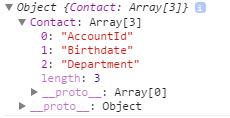


Figure : Map pour objet "Contact"

* Toutes les informations saisies par l’utilisateur sont stockées dans un objet créé spécialement pour l’application qui s’appelle « Configurations\_\_c » qui contient des champs qui sont :
  + Configuration Name : Nom de la configuration
  + central\_object : Objet central
  + query : La map de l’objet central pour effectuer la requête.
  + children\_query : La map des objets fils pour effectuer la requête.
  + where\_clauses : les conditions à ajouter à la requêtes si l’utilisateur choisi des filtres.

Une fois toutes les informations renseignées, il reste plus qu’à construire les requêtes en bouclant sur les objets champs… et récupérer le résultat pour le traiter (vérifier s’il n’est pas nul, le mettre en forme pour l’afficher en format HTML).



Figure : Script pour construire une requete

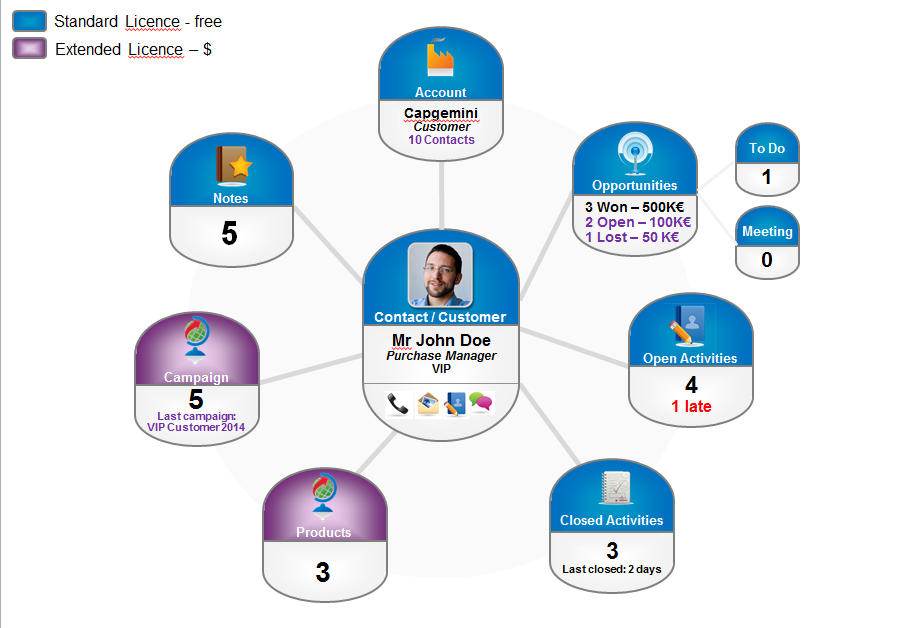
* d
* r

### P

## Mode étendu

Nous avons vu précédemment les principales caractéristiques du mode « Basic » qui va concentrer les objets « horizontaux de salesforce » (Account, Contact, Case, Opportunity, Activities).

Dans une licence étendue, soit à travers une configuration (consultant cap ou client), soit à travers du développement spécifique du « manage package », on pourrait ajouter les customs objects du projet client et rajouter (comme pour opportunité ci dessus dans l’illustration, 1er 2ème niveau de visualisation directement).



* Les fenêtres en mauve sont propres au client et au projet en cours, elles sont paramétrables sur demande en fonction des besoins d’utilisation.

## Charte graphique

### Logos

Le logo de l’application est le suivant :



Les bulles (version basique) seront sous cette forme :



Les bulles (version étendue) seront sous cette forme :



Les connexions entre les fenêtres :

fleche.png

## Conclusion

Acteur majeur sur le secteur du CRM en France, plusieurs ss2i travaillant dans ce domaine consolident leur expansion en s’associant avec toujours plus de grandes sociétés du secteur industriel, énergétique, des transports ou de la finance. Dans ce contexte d'expansion, ce secteur cherche à recruter des jeunes diplômés ayant une forte ouverture d'esprit, car ils doivent posséder un bagage technique, une bonne adaptabilité, une bonne expression orale et une bonne aptitude à comprendre le métier de ses clients.

Les applications CRM utilisées par les clients sont intégrées dans des Systèmes d’Informations complexes avec une forte interaction entre les modules du SI. Les ss2i ont la capacité de gérer cette complexité et à proposer de nouveaux concepts technologiques et métier pour une gestion plus efficiente du SI. Capgemini fait partie de ces ss2i ayant un portefeuille d’offres assez large pour accompagner ses clients depuis la phase de cadrage des besoins jusqu’à l’exploitation du parc informatique.

Pour toutes ces raisons, j’ai choisi d’intégrer Capgemini en stage pour avoir une première expérience significative dans le domaine des Systèmes d’Information auprès de spécialistes reconnus. Les compétences acquises durant mon parcours d'ingénieur à Sup Galilée me permettent aujourd’hui de trouver pleinement ma place dans une entreprise leader dans son domaine comme Capgemini.

L’intégration de stagiaires sur les projets est une politique appliquée par Capgemini depuis longtemps. Elle a réussi à former des collaborateurs compétents et polyvalents. Associé à des sessions de formation internes ou externes, le placement d’un stagiaire sur un projet est la meilleure opportunité d’apprentissage que Capgemini puisse leur offrir. L’encadrement sur les projets est régulier avec les chefs de projet et les consultants seniors dotés d’une expérience significative et d’un fort esprit d’entraide. De plus, le stagiaire est considéré comme un « consultant junior », et donc assume dès le début, des responsabilités pour chaque tâche proposée.

Lors du stage que j’ai mené sur le projet TCS, j’ai réussi à me placer au cœur de deux branches majeures de l’entreprise : La branche fonctionnelle et la branche technique qui sont toutes aujourd’hui assez distinctes chez Capgemini. Grâce à cette place, j’ai pu montrer mes capacités de compréhension et d’assimilation du métier du client et d’en comprendre le fonctionnement.

Pour conclure, ce stage m'a vraiment intéressé, notamment la partie technico-fonctionnelle, puisqu'il m'a permis de mener de a à z. Ainsi, ce stage riche en expériences et en enseignements m’a donné l’opportunité de travailler pour une entreprise leader dans le secteur de l'assurance Suisse, en lui délivrant et en maintenant une application cloud à forte valeur ajoutée. Enfin ce stage m’a permis de commencer une carrière dans le domaine des Systèmes d’Information, puisque j’ai eu des propositions d’embauches en CDI comme consultant technico-fonctionnel en CRM.

## Références bibliographiques

[http://www.capgemini.com](http://atos.net/en-us/Newsroom/fr-fr/Communiques_presse/2005/2005_09_30_01.htm)&talent.capgemini (Site interne contenant de la documentation sur Capgemini pour les associés de l'entreprise) & Les documentations sous format papier de Capgemini.

* Historique de Capgemini
* Présence de Capgemini
* Les secteurs d'activités
* Ma position dans Capgemini
* Projet Dynamo

http://laurence.mollard.free.fr/MEMOIREwebCRM.htm

* Le CRM

http://www.tcs.ch/fr/

* Le TCS

http://www.salesforce.com

* Le CRM
* Salesforce
* Le Cloud Computing

http://fr.wikipedia.org:

* Fiche d'identité de Capgemini
* Salesforce

[1] [http://fr.wikipedia.org/wiki/Gestion\_de\_la\_relation\_client](http://atos.net/en-us/Newsroom/fr-fr/Communiques_presse/2005/2005_09_30_01.htm)

[2] http://fr.wikipedia.org/wiki/Extreme\_programming

[3] http://fr.wikipedia.org/wiki/TeamForge

## Glossaire

|  |  |
| --- | --- |
| API | Application Programming Interface. |
| BU | Business Unit. |
| CC | Cloud Computing. |
| CRM | Customer Relationship Management. |
| ESN | Entreprise de Services du Numérique |
| GED | Gestion Electronique des Documents. |
| GRC | Gestion de la Relation Client. |
| HTML | Hyper Text Markup Language. |
| ID | Identifiant |
| Lot | Lotissement. |
| SAP | Systems, Applications and Products. |
| SFD | Spécifications Fonctionnelles Détaillées |
| SFDC | Salesforce.com. |
| SMC | Solution Métier & Collaboratives |
| SQL | Structured Query Language |
| SSII | Société de Services en Ingénierie Informatique. |
| STD | Spécifications Techniques Détaillées |
| TCS | Touring Club Suisse. |
| Vfp | VisualForce Page |
| WSDL | Web Services Description Language. |
| XML | Extensible Markup Language. |
| XP | eXtreme Programming. |

## Table des figures

[Figure 1 : Fiche d’identité (source : Wikipedia) 8](#_Toc408222895)

[Figure 2 : Implémantation de Capgemini dans le monde au 31 déc. 2013 (Source : Documentation Capgemini) 9](#_Toc408222896)

[Figure 3 : Secteurs d’activités de Capgemini 10](#_Toc408222897)

[Figure 4 : Skill Center de la BU Industrie & Distribution 11](#_Toc408222898)

[Figure 5 : Siège du TCS 11](file:///C:\Users\amine\Desktop\rapport.docx#_Toc408222899)

[Figure 6 : Diagramme Capgemini 12](file:///C:\Users\amine\Desktop\rapport.docx#_Toc408222900)

[Figure 7 : Principe du cloud computing 14](file:///C:\Users\amine\Desktop\rapport.docx#_Toc408222901)

[Figure 8 : Répartition des consultants sur le projet Dynamo (TCS) 19](#_Toc408222902)

[Figure 9 : Vue du visuel 27](file:///C:\Users\amine\Desktop\rapport.docx#_Toc408222903)

[Figure 10 : Popup lors du survol des bulles 28](file:///C:\Users\amine\Desktop\rapport.docx#_Toc408222904)

[Figure 11 : Boutons Call2action 28](file:///C:\Users\amine\Desktop\rapport.docx#_Toc408222905)

[Figure 12 : Map pour objet "Contact" 38](#_Toc408222906)

[Figure 13 : Script pour construire une requete 39](#_Toc408222907)

[Figure 14 : Page d’accueil de l’environnement recprjlot2 du TCS 47](#_Toc408222908)

[Figure 15 : Page de configuration Salesforce 48](#_Toc408222909)

[Figure 16 : Cycle itératif d’une méthode AGILE 51](#_Toc408222910)

# 

# Annexe

## Annexe 1: Explication de la plateforme force.com

Nous allons apporter un niveau de détail supplémentaire sur la plate-forme Force.com et Salesforce.com, afin de comprendre un peu mieux l’environnement technique dans lequel s’est déroulé mon stage, et comprendre les choix techniques. Ensuite, nous développerons la méthode agile eXtreme Programming, et enfin nous présenterons un extrait du PowerPoint du POC qui compare Adobe Marketing Cloud et ExactTarget Marketing Cloud sur différents critères.

##### Modèle de données

##### Pour comprendre au mieux le fonctionnement de la plate-forme Force.com, nous devons impérativement passer par son modèle de données. La plate-forme Force.com propose un modèle de données standard très riche. Le schéma suivant représente le modèle de données natif de la partie vente de la plate-forme Force.com.

##### 

Figure : Page d’accueil de l’environnement recprjlot2 du TCS

Les principaux objets de ventes sont :

* Activités
* Accueil
* Campagnes
* Pistes
* Comptes
* Contacts
* Opportunités
* Prévisions
* Requêtes
* Contrats
* Solutions
* Produits
* Partenaires
* Utilisateurs

Ce modèle est la partie standard du modèle données de Force.com, elle s’enrichira en créant des :

* Objets personnalisés
* Champs personnalisés
* Relations parents-fils
* Relations de recherche
* Objets de jonction

##### Approche de configuration

Le schéma suivant présente une partie de l’environnement de configuration de Salesforce.com.

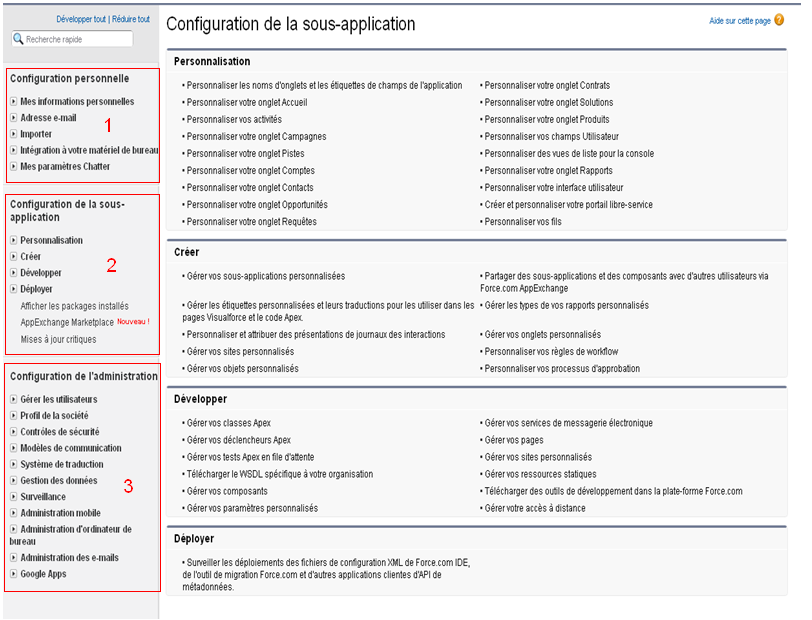


Figure 15 : Page de configuration Salesforce

1. La partie **Configuration personnelle :**

Dans cette partie on peut :

* Modifier les données, la langue, le fuseau horaire, le quota ou l'équipe de vente.
* Personnaliser les onglets et les listes associées.
* Modifier le mot de passe.
* Accorder un accès à la connexion au support client salesforce.com.
* Réinitialiser le jeton de sécurité.
* Gérer la visibilité du calendrier pour d'autres utilisateurs.
* Créer ou modifier un groupe privé.
* Modifier les paramètres de rappel.
* Importer les données Outlook, etc.
* Modifier les paramètres des messages sortants.
* Créer ou modifier un modèle de message.
* Télécharger Chatter Desktop.
* Supprimer des enregistrements du porte-documents Force.com Connect Offline.
* Définir les paramètres du porte-documents Force.com Connect Offline.
* Télécharger Force.com Connect Offline.

1. La partie **Configuration de la sous-application:**

Dans cette partie on peut :

* Personnaliser des vues de liste pour la console.
* Personnaliser l'interface utilisateur.
* Personnaliser les onglets.
* Créer et personnaliser le portail libre-service.
* Gérer les sous-applications personnalisées.
* Partager des sous-applications et des composants avec d'autres utilisateurs via Force.com AppExchange.
* Gérer les étiquettes personnalisées et leurs traductions pour les utiliser dans les pages Visualforce et le code Apex.
* Gérer les types de vos rapports personnalisés.
* Personnaliser et attribuer des présentations de journaux des interactions.
* Gérer les onglets personnalisés.
* Gérer les sites personnalisés.
* Personnaliser les règles de workflows.
* Gérer les objets personnalisés.
* Personnaliser les processus d'approbation.

Cette partie nous donne également accès au developpement spécifique en offrant les fonctionnalités suivantes:

* Gérer les classes Apex .
* Gérer les services de messagerie électronique.
* Gérer les déclencheurs Apex.
* Gérer les pages.
* Gérer les tests Apex en file d'attente.
* Gérer les sites personnalisés.
* Télécharger le WSDL spécifique à l'organisation.
* Gérer les ressources statiques.
* Gérer les composants.
* Télécharger des outils de développement dans la plate-forme Force.com.
* Gérer les paramètres personnalisés.
* Gérer l'accès à distance.

1. La partie **Configuration de l'administration:**

Cette partie nous offre la possibilité de contrôller l'administration du système. Notamment l'accès, la sécurité, l'intégration, l'interaction avec les systèmes distants, les services web, et la getion des utilisateurs.

* Profil de la société.
* Contrôles de sécurité.
* Modèles de communication.
* Gestion des données.
* Surveillance.
* Administration mobile.
* Administration d'ordinateur de bureau.
* Administration des e-mails.
* Intégration avec Google Apps.

## Annexe 2: Méthode agile eXtreme Programming

L’équipe de Capgemini se base sur la méthode AGILE eXtreme Programming. Voici une brève description de cette méthode.

« En [informatique](http://fr.wikipedia.org/wiki/Informatique) et plus particulièrement en génie logiciel, l'Extreme Programming (XP) est une [méthode agile](http://fr.wikipedia.org/wiki/M%C3%A9thode_agile) orientée sur l'aspect réalisation d'une application, sans pour autant négliger l'aspect [gestion de projet](http://fr.wikipedia.org/wiki/Gestion_de_projet). XP est adapté aux équipes réduites avec des besoins changeants. XP pousse à l'extrême des principes simples » [2].



Figure 16 : Cycle itératif d’une méthode AGILE

L’eXtreme Programming est défini de cette manière :

* Un planning construit avec le client et focalisé sur les buts principaux à atteindre.
* Des livrables livrés fréquemment : le projet est découpé en phases (définition des besoins, construction, déploiement) et la phase de construction est elle-même divisée en multiples itérations (spécifications courtes, développement, tests, retour du client). Cela permet de réduire les incompréhensions.
* Une mise en production rapide d’un système minimal fonctionnel, puis des ajouts de fonctionnalités régulièrement dans un temps de cycle très court.
* Aller droit à l’essentiel.
* Chaque développeur s'attribue des tâches et [les réalise avec un binôme](http://fr.wikipedia.org/wiki/Programmation_en_bin%C3%B4me).

## spec.jpgAnnexe 3: Apercu des taches à faire