

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Université Assane Seck de Ziguinchor  
UFR Sciences et Technologies  
Département Informatique



## Mémoire de fin d'études

Pour l'obtention du diplôme de Master  
Mention : Informatique ; Spécialité : Génie Logiciel  
Sujet :

**Conception et développement sous Salesforce d'un module de gestion de risques dans un projet**

Présenté par : M. Abdoulaye Diallo

Soutenance le 04/07/2020

Sous la direction de : Dr Youssou FAYE

Sous la supervision du - Pr. Salomon SAMBOU

### Membres du jury

Salomon SAMBOU	Professeur	Président	UASZ
Youssou FAYE	Maître de Conférence Titulaire	Encadreur	UASZ
Mouhamadou GAYE	Maître de Conférence	Rapporteur	UASZ
Papa Alioune CISSE	Maître de Conférence	Rapporteur	UASZ
Moussa Fofana	Ingénieur	Maître de Stage	TérangaCloud Solutions SA

# Résumé

Le risque dans un projet peut être défini comme un danger ou un problème possible qui peut impacter négativement sur les objectifs du projet. Le risque peut être estimé selon sa probabilité d'apparition et sa gravité.

Ce document est un travail réalisé au sein de l'entreprise Téranga Cloud Solutions. Il porte sur la réalisation d'un module de gestion des risques pour le compte de l'application de gestion de projet : Project Management.

Il sera développé sous la plateforme Salesforce et avec la norme de gestion des risques ISO<sup>1</sup> 31000 2009.

Ce module sera évolutif car il sera basé sur le modèle SOC<sup>2</sup> et sera intégré dans l'application Project Management.

Il doit permettre une meilleure gestion des risques durant toute la durée d'un projet. Pendant la durée de ce dernier le module de gestion des risques doit permettre au risk manager<sup>3</sup> de suivre l'état des risques, de pouvoir faire des évaluations et de prévoir des mesures futures en cas d'impact d'un risque sur un projet.

Enfin elle sera déployée sous Appex Change la boutique d'application de Salesforce.

Mot clés : Risque, Salesforce, Modèle SOC

---

<sup>1</sup> Iso: International Organization Standardization

<sup>2</sup> Separation of Concern: [https://en.wikipedia.org/wiki/Separation\\_of\\_concerns](https://en.wikipedia.org/wiki/Separation_of_concerns)

<sup>3</sup> Gestionnaire des risques : [https://fr.wikipedia.org/wiki/Gestion\\_des\\_risques](https://fr.wikipedia.org/wiki/Gestion_des_risques)

# Abstract

Risk in a project can be defined as a potential hazard or problem that can negatively impact the project objectives. The risk can be estimated according to its probability of occurrence and severity

This document is a work done within the company Téranga Cloud Solutions SA . It involves the development of a risk management module for the project management application: Project Management.

It will be developed under the Salesforce platform and with the ISO 31000 2009 risk management standard.

This module will be scalable as it will be based on the SOC (Separation of Concern) model and will be integrated into the Project Management application.

It must allow for better risk management throughout the duration of a project. During the duration of the latter, the risk management module must enable the risk manager to monitor the state of risks, to be able to carry out evaluations and to plan future measures in the event of the impact of a risk on a project.

Finally, it will be deployed under Apex Change the Salesforce application store.

Keywords: Risk, Salesforce, SOC.

# *Dédicaces*

Ce présent mémoire est dédié à toutes les personnes qui m'ont soutenu , encouragé et encadré tout le long de mon cursus.

A mon père Amadou Diallo à ma mère Aissatou Ba et à ma grande mère Halimatou Souaré.

A mes frères Mamadou Diallo Ousmane Baba Diallo Seydou Gackou Thierno Ibrahima Diallo Salif Diallo Soufour Diallo .

A mes sœurs Mariama Diallo Néné Gallé Diallo Ndeye Binta Diallo, Sophinatou Diallo Dieynaba Diallo.

A mon fils Papa Amadou Diallo et à mes neveux Babacar Fam Aziz Fall Major Mouhamed Gaye.

A monsieur Lat Grand Ndiaye pour m'avoir accompagné pendant tout mon cursus universitaire.

A ma sœur Mame Bousso Kane mes amis Moustapha Diouf, Khalidou Ibrahima Sow et Abdoulaye Diop.

A ma seconde famille, toute la famille Atchikity pour leur soutien inconditionnel et leur amour inestimable qu'ils m'ont apporté.

A notre défunte DRH Madame Diop qui nous a accompagné pendant toute la durée de notre stage au sein de l'entreprise Téranga Cloud Solutions SA.

A notre défunt camarade de classe Mamadou Petit Coulibaly, que le bon dieu l'accueille dans son paradis.

A mes sœurs, cousines, neveux, nièces oncles parents

Enfin tous ceux qui ont participé à ma réussite.

A vous tous je vous dédie ce travail et vous dit merci.

# Remerciements

<<C'est avec un grand plaisir que je réserve ces quelques lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce modeste travail. >>

Je tiens tout d'abord à exprimer mes plus chaleureux remerciements à mes encadrants :

**Monsieur Youssou FAYE**, Enseignant chercheur à l'Université Assane SECK de Ziguinchor

Un monsieur très ambitieux et généreux. Je vous remercie pour tout le temps que vous avez consacré pour nous accompagner dans ce travail depuis le début de notre stage.. Je ne saurais vous remercier. Merci pour tout votre soutien.

**Monsieur Mamadou Bousso** le Directeur Technique de Téranga Cloud Solutions

**Monsieur Moussa Fofana** notre Maitre de Stage

Je souhaite remercier les membres de mon jury de mémoire :

**Monsieur Salomon SAMBOU**, professeur à l'Université Assane SECK de Ziguinchor, pour le temps qu'il a bien voulu consacrer à l'évaluation de ce travail, mais aussi, de m'avoir fait l'honneur de présider le jury de ma soutenance.

Mes respectueux remerciements vont à mes rapporteurs :

**M. Mouhamadou GAYE** Enseignant chercheur à l'Université Assane SECK de Ziguinchor, pour m'avoir fait l'honneur de participer à mon jury de mémoire.

**M. Papa Alioune CISSE** Enseignant chercheur à l'Université Assane SECK de Ziguinchor. Je vous remercie de m'avoir fait l'honneur d'être rapporteur de mon mémoire.

Mes amicaux remerciements vont à tous les camarades de promotion, pour tous les beaux moments passés ensemble.

Enfin je ne manquerai pas d'exprimer ma grande reconnaissance à tous les professeurs de l'UASZ, particulièrement ceux qui ont participé de près ou de loin à notre formation de master.

# Table des Matières

Introduction générale :	1
1 Chapitre1 :Le Cloud.....	3
1.1 Définition :	3
1.2 L'Utilité du Cloud Computing :	3
1.3 Les différents types de Cloud :	4
1.4 Les modèle de service de cloud :	5
1.4.1 : Le Logiciel à la demande.....	5
1.4.2 La Platform as a Service (PaaS).....	6
1.4.3 IAAS : Puissance de calcul et stockage à la demande.....	7
1.5 Les avantages et les inconvénients de la sauvegarde en ligne :	8
1.5.1 Les avantages du Cloud :	8
1.5.2 Inconvénients du cloud .....	9
1.6 Le développement d'applications dans le Cloud :	9
1.6.1 Avantages du développement dans le cloud :	10
1.6.2 Inconvénients du développement dans le cloud :	10
2 Chapitre2 : Présentation de Salesforce et de Project Management :	13
2.1 Présentation de la structure d'accueil :	13
2.1.1 Présentation générale :	13
2.1.2 Présentation des différents services de l'entreprise .....	14
2.1.3 Organigramme :	16
2.2 Présentation de Salesforce :	16
2.2.1 Historique et présentation :	16
2.2.2 Présentation du CRM.....	17
2.2.3 Salesforce et le développement d'applications :	21
2.3 Présentation de Project Management :	29
2.3.1 Présentation :	29
2.3.2 Objectifs de Project Management.....	29
2.3.3 Objectifs de l'entreprise à travers le développement d'un module de gestion des risques :	29
2.3.4 Rôles des différents acteurs de l'entreprise dans la Gestion des risques :	29
2.3.5 Organisation du projet :	30
2.3.6 Les objectifs du stage :	30
3 La Gestion des risques :	33

3.1	Les normes de Gestion des risques : .....	33
	Gestion des risques des dispositifs médicaux .....	34
3.2	Justification du choix de la norme : .....	35
3.3	La norme de Gestion des risques ISO 31000 2009 .....	35
3.3.1	Historique et Présentation : .....	35
3.3.2	Le but de la Norme : .....	36
3.3.3	Les principes de la norme : .....	36
3.3.4	Certification de la norme : .....	38
3.3.5	Structure de la Norme : .....	38
3.3.6	L'évaluation des risques : .....	39
3.4	Etude de l'existant : .....	40
3.4.1	Présentation des applications de gestion des risques dans Salesforce : .....	40
3.4.2	Etude comparative entre les deux applications : .....	41
3.4.3	Objectifs à atteindre pour notre sujet : .....	43
4	Méthodologie et analyse des besoins : .....	46
4.1	Méthodologie : .....	46
4.2	Analyse des besoins : .....	47
4.2.1	Identification des besoins non fonctionnels : .....	47
4.2.2	Les différents acteurs du Projet et leurs cas d'utilisation : .....	48
4.2.3	Les diagrammes de séquence : .....	50
5	Conception et Proposition d'une Solution : .....	60
5.1	Conception : .....	60
5.1.1	L'Architecture du projet : .....	60
5.1.2	La modélisation : .....	62
5.1.3	Le design du projet : .....	68
5.1.4	Gestion des profils : .....	69
5.1.5	Le langage SOQL .....	70
5.1.6	La notion de governors limit.....	70
5.1.7	Les tests.....	71
5.1.8	La gestion de la sécurité dans le projet : .....	71
5.2	Présentation de la solution : .....	72
	Conclusion générale et perspectives : .....	81

## *Table des figures*

Figure 1 Le Cloud .....	3
Figure 2 Les modèles de service du Cloud Computing .....	5
Figure 3 Le SAAS.....	6
Figure 4 Le PAAS.....	6
Figure 5 Le IAAS.....	7
Figure 6 Différence entre les différents services de Cloud.....	7
Figure 7 Collaboration et partage dans le Cloud .....	8
Figure 8 Architecture de Salesforce.....	17
Figure 9 Les service de Salesforce.....	18
Figure 10 SalesCloud.....	18
Figure 11 Marketing Cloud.....	19
Figure 12 Analytics Cloud.....	20
Figure 13 Chatter .....	20
Figure 14 Apex Change .....	21
Figure 15 Le schéma Builder de Salesforce .....	22
Figure 16 La Dev Console .....	23
Figure 17 Les deux contrôleurs du Framework .....	24
Figure 18/Architecture du framework lightning .....	25
Figure 19 Structure d'un Composant .....	27
Figure 20 Norme ISO 31000 2009.....	38
Figure 21 Le ERM .....	39
Figure 22 Matrice des risques .....	40
Figure 23 Applications de Gestion des Risques dans ApexChange .....	41
Figure 24 Processus ALM .....	47
Figure 25 Cas d'utilisation de l'administrateur.....	48
Figure 26 Cas d'utilisation du Project Manager.....	49
Figure 27 Cas d'utilisation du développeur.....	49
Figure 28 Authentification des Utilisateurs .....	51
Figure 29 Création d'un Risque .....	53
Figure 30 Affichage détails risque.....	54
Figure 31 Modification d'un risque.....	56
Figure 32 Suppression d'un Risque.....	58
Figure 33 Les différentes Couches .....	61
Figure 34 Diagramme de Classes .....	64
Figure 35 Correspondance entre un Objet et une Classe .....	65
Figure 36 Création d'un Custom Label .....	65
Figure 37 Relation entre deux Objets Salesforce.....	66
Figure 38 Création d'un Custom Object .....	66
Figure 39 Création d'un Record Type.....	67
Figure 40 Modèle Final.....	68
Figure 41 SLDS .....	69

Figure 42	Exemple de Code SLDS .....	69
Figure 43	Liste des Profils .....	70
Figure 44	Les governors Limits .....	71
Figure 45	Page d'authentification .....	72
Figure 46	Formulaire de Création d'un Projet et d'un Program .....	73
Figure 47	List des Projets .....	74
Figure 48	Formulaire de création d'un risque .....	75
Figure 49	Liste des risques .....	76
Figure 50	code couleur des risques .....	76
Figure 51	Filtre des risques Par Projet .....	77
Figure 52	Détails des risques pour un projet .....	78
Figure 53	Tableau de Bord des risques par project1 .....	79
Figure 54	Tableau de bord des risques par Project2 .....	79
Figure 55	Tableau de bord des criticités des risques .....	80

## *Liste des Abréviations*

CRM	Customer relationship management
PAAS	Plateforme as a service
IAAS	Infrastructure as a service
SAAS	Software as a service
ALM	Application Lifecycle Management
SOC	Separation of concerns
SLDS	Salesforce lightning design system
ERP	Entreprise Resource Planning
AFNOR	Association Française de Normalisation
NF	Norme Française
ISO	International Organization Standardization
XML	Extensible Markup Language
ERM	Entreprise Risk Management
API	Application Programming Interface
SOAP	Simple Object Access Protocol
SOQL	Salesforce Object Query Language
IDE	Integrated development environment
IRAM	Instituto Argentino de Normalizacion
DIN	Deutsches Institut fur Normung

## Introduction générale :

Les organismes de tous types et de toutes dimensions confrontés à des facteurs et des influences internes et externes ignorent si et quand ils vont atteindre leurs objectifs. L'incidence de cette incertitude sur l'atteinte des objectifs d'un organisme constitue le «risque». Pour une réussite d'un projet un logiciel de gestion de projet s'impose. Il permettra de rendre plus facile l'exécution des tâches et une meilleure prévision des risques. Le risque est défini comme un danger éventuel plus ou moins prévisible inhérent à une activité.

Dans la gestion de projet le risque zéro n'existe pas. Il est fort probable qu'un projet ne se réalise pas à cause des différents risques auxquels il s'expose. Ceci nous pousse alors à anticiper et limiter le maximum de risques et de les suivre tout au long du projet. La gestion des risques apparaît ainsi comme un des aspects des plus intéressants dans le suivi d'un projet.

C'est dans ce contexte que s'inscrit notre sujet de mémoire dont l'objectif principal est de développer un module de gestion des risques qui est un des modules de Project Management pour le compte de Téranga Cloud Solutions. Pour ce faire nous avons utilisé Salesforce qui est un CRM entièrement basé dans le Cloud et la norme de gestion des risques ISO 31000 2009.

Ce document est structuré en cinq chapitres.

Dans le chapitre 1 nous allons présenter le cloud computing pour comprendre dans le chapitre 2 la plateforme Salesforce.

Le chapitre 2 est consacré à la présentation de la plateforme Salesforce que nous avons utilisé pour développer notre application.

Le chapitre 3 se focalise essentiellement sur la gestion des risques (les différentes normes et la présentation de la norme choisie).

Dans le chapitre 4 nous allons parler de la méthodologie et de l'analyse des besoins pour mieux comprendre les différents acteurs et leurs rôles respectifs dans l'application future.

Et nous terminerons par le chapitre 5 où nous allons présenter l'implémentation et la présentation de la solution à travers la présentation des différentes interfaces.

Enfin nous terminerons ce document par une conclusion générale et des perspectives.

## Les Généralités du Cloud Computing

### Introduction :

Dans ce premier chapitre nous allons présenter les généralités du cloud computing. Nous allons aussi présenter les avantages et les inconvénients du développement d'application dans le cloud. Ceci nous permettra de mieux comprendre la plateforme Salesforce dans le chapitre2.

---

Chapitre1 :Le Cloud.....	3
1.1 Définition : .....	3
1.2 L'Utilité du Cloud Computing : .....	3
1.3 Les différents types de Cloud : .....	4
1.4 Les modèle de service de cloud : .....	5
1.4.1 : Le Logiciel à la demande.....	5
1.4.2 La Platform as a Service (PaaS).....	6
1.4.3 IAAS : Puissance de calcul et stockage à la demande.....	7
1.5 Les avantages et les inconvénients de la sauvegarde en ligne : .....	8
1.5.1 Les avantages du Cloud : .....	8
1.5.2 Inconvénients du cloud .....	9
1.6 Le développement d'applications dans le Cloud : .....	9
1.6.1 Avantages du développement dans le cloud : .....	10
1.6.2 Inconvénients du développement dans le cloud : .....	10

---

## 1 Chapitre1 : Le Cloud

### 1.1 Définition :

Le cloud (ou cloud computing) est une technologie qui permet de déposer des données sur des serveurs localisés à distance et d'y avoir accès n'importe quand et depuis n'importe quel appareil connecté à internet.

Tous types de données peuvent y être conservées : mails, images, notes, logiciels, documents bureautiques, etc.



*Figure 1 Le Cloud*

### 1.2 L'Utilité du Cloud Computing :

Grâce au cloud, on peut déporter sur des serveurs distants des traitements informatiques traditionnellement localisés sur des serveurs locaux. Ainsi, au lieu de devoir installer des logiciels ou applications sur le disque dur de son ordinateur, ceux-ci sont disponibles sur internet. Cependant on distingue deux types d'utilisation de cloud :

- Le Stockage des données en ligne grâce à des sites internet spécialisés qui proposent un espace de stockage.
- Les Service en ligne : certains logiciels, applications et jeux vidéo sont accessibles directement sur l'internet sans installer de logiciel.

Le cloud computing est aussi utilisé pour des raisons suivantes :

- ✓ **Le partage** : sur le Cloud plusieurs personnes peuvent travailler sur le même document et les contenus peuvent être partagés et envoyés à n'importe qui et facilement.
- ✓ **La sauvegarde** : avec l'option de prévention d'éventuels accidents techniques ou matériels, le Cloud nous permet de planifier des sauvegardes régulières de nos fichiers.
- ✓ **L'accessibilité** : certains sites proposent des fonctionnalités adaptées au type de document stocké. Exemple : Google Music pour la musique.

- ✓ **La mobilité** : tous les documents sauvegardés sont accessibles depuis n'importe quel appareil connecté à internet et à n'importe quel moment.
- ✓ **Le manque d'installation de logiciel** : de plus en plus de logiciels jeux et applications ont leur équivalent en ligne. Ainsi on n'a pas besoin d'installation d'application sur un appareil.

### 1.3 Les différents types de Cloud :

Il existe quatre types de Cloud : le Cloud Privé, le Cloud Public, le Cloud Hybride et le Cloud Communautaire.

#### ➤ Le Cloud Privé :

Le Cloud Privé repose sur l'individualisation des ressources et des services. Chaque client bénéficie d'une infrastructure qui lui est dédiée. On distingue deux Cloud Privés : Interne et Externe. Le Cloud Privé Interne revient pour l'entreprise à investir dans une infrastructure qui lui est propre et de la mettre en libre-service auprès de ses utilisateurs et des services métiers. Le Cloud Privé Externe signifie que cette infrastructure est hébergée en dehors de l'entreprise par un fournisseur de service.

#### ➤ Le Cloud Public :

Le Cloud public permet d'accéder à tout moment et à la demande à un pool de ressources informatiques en réseau, flexible et évolutif mais aussi accessible à un grand nombre d'utilisateurs sur internet. Les logiciels, les infrastructures et les plateformes fournissent des services public Cloud moyennant des frais d'utilisation.

#### ➤ Le Cloud hybride

Un Cloud hybride est un environnement informatique Cloud qui fait appel à une combinaison de services en Cloud privé sur site et en Cloud public, avec une capacité d'orchestration entre les deux plateformes.

Le Cloud hybride permet ainsi de déplacer les charges de travail entre Cloud public et privé au gré de l'évolution des coûts et des besoins informatiques. Il confère alors à l'entreprise une flexibilité accrue et davantage de possibilités de déploiement de données.

#### ➤ Le Cloud Communautaire :

Le cloud communautaire est utilisé par plusieurs organisations qui ont des besoins communs et en même temps.

C'est la possibilité pour plusieurs entités ou membres d'organisations ayant les mêmes besoins d'utiliser une seule et unique solution Cloud. Dans ce cas, le Cloud est partagé au sein de plusieurs structures (sociétés, filiales, regroupement d'entreprises, groupes métiers etc.) et sa gestion est assurée soit en interne, soit en externe. Il peut alors être utilisé pour des applications génériques adaptées aux préoccupations du groupe, ou encore pour héberger des applications très spécialisées, mais communes à de très nombreuses entreprises qui souhaitent fédérer leurs efforts.

#### 1.4 Les modèles de service de cloud :

Le Cloud offre plusieurs modèles de service dont les plus utilisés sont : Le SAAS<sup>4</sup>, le PAAS<sup>5</sup> et le IAAS<sup>6</sup>.

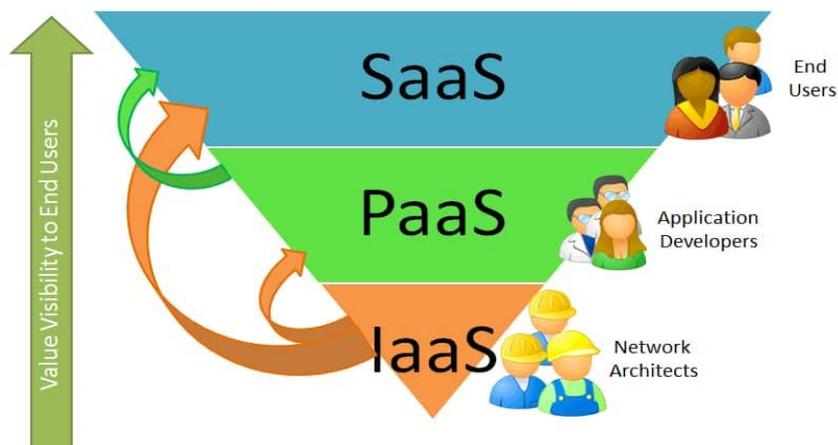


Figure 2 Les modèles de service du Cloud Computing

##### 1.4.1 : Le Logiciel à la demande

Le Software as a Service (SaaS) est accessible à toutes les entreprises et, il est facturé au nombre d'utilisateurs. L'entreprise loue les applications du fournisseur de services. Plus besoin d'acheter un logiciel.

Ces applications sont accessibles via différentes interfaces, navigateurs Web, clients légers...

De nombreux logiciels sont disponibles dans le Cloud, en mode SaaS :

- Messagerie : L'e-mail est l'application la plus utilisée en mode Cloud

<sup>4</sup> Software as a service : <https://www.lebigdata.fr/definition-saas>

<sup>5</sup> Plateforme as a service : <https://www.lemagit.fr/definition/Platform-as-a-Service-PaaS>

<sup>6</sup> Infrastructure as a service : <https://azure.microsoft.com/fr-fr/overview/what-is-iaas/>

- CRM : La gestion de la relation client est l'autre application phare du Cloud
- ERP<sup>7</sup> : Certains progiciels de gestion intégrée sont proposés en mode Saas.
- Collaboratif : Les outils de collaboration (partage de documents, réseaux sociaux...) se prêtent bien au mode Saas.



Figure 3 Le SAAS

#### 1.4.2 La Platform as a Service (PaaS)

Facturée à la consommation, le platform as a service est un environnement qui permet à l'entreprise de déployer ses propres applications en dehors de sa salle informatique. L'entreprise loue un environnement middleware à l'infrastructure masquée.

L'utilisateur gère, mais ne contrôle pas l'infrastructure Cloud (réseaux, serveurs, systèmes d'exploitation, stockage). Il a ainsi le contrôle sur les applications déployées et la possibilité de configurer l'environnement d'hébergement applicatif.



Figure 4 Le PAAS

<sup>7</sup> Enterprise Resource Planning : [https://fr.wikipedia.org/wiki/Progiciel\\_de\\_gestion\\_intégr%C3%A9](https://fr.wikipedia.org/wiki/Progiciel_de_gestion_intégr%C3%A9)

### 1.4.3 IAAS : Puissance de calcul et stockage à la demande.

L'Infrastructure as a Service (IaaS) est la mise à disposition par Internet de machines virtuelles aux ressources « facilement » modifiables (à la hausse ou baisse) et hautement disponible. L'entreprise loue ainsi des capacités de traitement, de stockage et autres ressources qu'elle peut structurer et gérer de façon autonome côté logiciel dès le système d'exploitation.

Pour des calculs complexes ou ponctuels (simulations budgétaires, calculs prédictifs...) ou encore du stockage (notamment dans le cadre de sauvegardes), les administrateurs de ressources informatiques trouvent dans le Cloud des environnements aux capacités quasi illimitées et dont la mise en œuvre est quasi instantanée.



Figure 5 Le IAAS

Cependant il existe une différence entre les différents services de cloud :

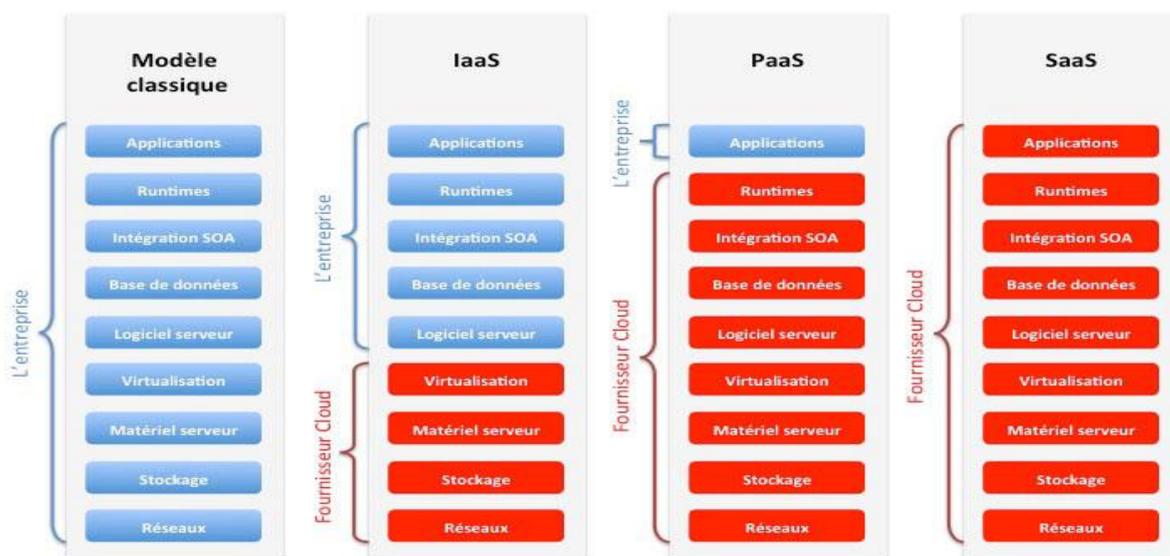


Figure 6 Différence entre les différents services de Cloud

## 1.5 Les avantages et les inconvénients de la sauvegarde en ligne :

### 1.5.1 Les avantages du Cloud :

Le cloud présente de nombreux avantages dont les plus intéressants sont :

- **L'Externalisation** : Les données et documents sont stockés sur un serveur distant. Le disque dur de votre ordinateur est allégé et vos documents importants conservés. Le fournisseur de service prévient d'éventuelles pannes et mis à jour .
- **La Mobilité et l'accessibilité** : l'accès au cloud est relativement aisé : une adresse email et un mot de passe suffisent. Les données conservées sont accessibles depuis n'importe quel appareil, depuis n'importe où et à n'importe quel moment. On aura plus besoin de puissants ordinateurs pour utiliser certains logiciels, il suffit seulement d'un accès à internet.
- **L'aspect économique** : On aura plus besoin de dépenses en matériel de stockage (clés USB, disques durs...) En cas de problème technique, on n'aura pas de frais pour la réparation du fait qu'on n'est pas responsable de celle-ci ou de la maintenance.
- **La Collaboration et le partage** : La plupart des services de stockage en ligne offrent la possibilité de partager des contenus, favorisant ainsi le travail de plusieurs personnes sur un même document. Le partage sur les réseaux sociaux est aussi très utilisé, notamment pour les photos.



*Figure 7 Collaboration et partage dans le Cloud*

### 1.5.2 Inconvénients du cloud

Les inconvénients du cloud les plus fréquents sont principalement l'aspect sécurité et l'aspect connexion.

#### ✚ **La Sécurité :**

D'habitude les données envoyées dans le cloud sont en sécurité. Quel que soit le prestataire, les serveurs qui conservent les données sont concentrés dans d'immenses parcs informatiques (Datacenter) surveillés en permanence afin d'éviter les attaques matérielles. Malgré tout, et bien qu'il soit faible, le risque d'intrusion de pirates informatiques dans ces serveurs est une réalité. En confiant nos données à un tiers, d'une certaine manière, l'aspect contrôle n'existera plus.

✚ **Nécessité d'une connexion internet :** Le cloud est très pratique. On peut accéder à ses propres documents n'importe quand et de n'importe où... Mais si l'utilisateur n'a pas de connexion internet ou une connexion insuffisante, il ne pourra pas y accéder.

De plus, selon nos besoins (accès à des jeux vidéo ou transfert de gros fichiers), il nous faudra une connexion très performante.

✚ **Pérennité :** Qu'arrive-t-il si le fournisseur décide de mettre fin au service ?

Exemple : Amazon, Hubic ou Dropbox précisent dans leurs conditions qu'ils ne peuvent fermer leur service n'importe quand et sans préavis. Donc le Cloud ne doit pas être comme unique moyen de sauvegarde.

### 1.6 Le développement d'applications dans le Cloud :

Le Cloud computing favorise de nouveaux modèles de développement principalement basés sur la rapidité, la fiabilité de déploiement, l'agilité, la flexibilité et la gestion du multi environnement.

Le développement sur le Cloud se démarque du développement sur plusieurs aspects :

- Lors du développement dans le Cloud, l'application est décomposée en petits éléments (composants) consécutifs.
- Le processus de test du code est rapide car les éléments sont petits ce qui signifie que les développeurs ont plus de facilité à corriger les bugs puisqu'ils ont déjà le code en mémoire.
- L'utilisation minimale des ressources informatiques.
- En utilisant les services du PAAS (Platform as a service) d'un prestataire cloud, les développeurs en charges du test ont plus de facilité à tester leur code.

- Il est possible de créer déployer et configurer des environnements de façon cohérente pour correspondre à divers environnements de production et à des fins de test et de développement.
- Les départements et équipes en charge du développement peuvent ainsi faire appel à des services de cloud spécifiques des développeurs, tels que des outils qui aident à accélérer et automatiser l'élaboration, les tests et le déploiement des applications.
- La possibilité de créer et de configurer des environnements rapidement permettra lors du développement d'une version mobile pour les entreprises et de recevoir quotidiennement les notifications de nouvelles versions.
- La rapidité, fiabilité et la gestion du multi environnement.

#### **1.6.1 Avantages du développement dans le cloud :**

Une architecture multi-tenant permet de mutualiser les ressources et les coûts nécessaires à l'exécution de l'application. En ce sens, cette architecture peut être vue comme une alternative à la virtualisation. L'agrégation de données et la fouille de données sont facilitées car une seule base de données stocke toutes les données.

#### **1.6.2 Inconvénients du développement dans le cloud :**

Le développement d'une application multi-tenant est plus difficile car il y a plus de possibilités de configuration et la gestion des données est complexifiée. La sécurité des données est essentielle afin qu'un utilisateur d'une entité ne puisse pas accéder aux données d'une autre entité. Afin d'y parvenir, le chargement des données doit se limiter à n'inclure que les données de l'entité courante, sauf à de rares exceptions telles que le traitement par lots. Le processus de livraison est plus risqué car si l'application contient un bug, il y aura un impact sur toutes les entités. De plus il peut devenir difficile de trouver des fenêtres pour la mise à jour de l'application lorsqu'une application multi-tenant est utilisée sur plusieurs fuseaux horaires. Enfin, les mécanismes de sauvegarde, et encore plus de restauration, posent aussi de vrais problèmes. En effet, alors que la sauvegarde peut s'effectuer à chaud, sur l'ensemble des données (toutes entités confondues), il est beaucoup plus difficile de réaliser une restauration des données d'une seule entité.

#### **Conclusion :**

Le cloud est un modèle omniprésent pratique et à la demande à un réseau partagé et à un ensemble de ressources informatiques configurables. Il offre différents modèles de service suivant le besoin.

Il donne aussi l'option de développement d'applications et met en œuvre certains aspects comme la rapidité et la fiabilité du déploiement et la gestion du multi environnement.

Il existe cependant certains inconvénients mais qui n'impactent pas trop le développement d'applications.

Cette présentation du cloud nous permet de mieux comprendre dans le chapitre deux la plateforme Salesforce et comment celle-ci fonctionne.

## Chapitre 2

### Présentation de Salesforce et de Project Management

#### Introduction :

Dans ce chapitre nous parlerons de la structure d'accueil et de la plateforme Salesforce ainsi que de notre application à développer ; c'est-à-dire les objectifs du projet et les objectifs de l'entreprise à travers le développement de cette application.

---

2	Chapitre2 : Présentation de Salesforce et de Project Management : .....	13
2.1	Présentation de la structure d'accueil : .....	13
2.1.1	Présentation générale : .....	13
2.1.2	Présentation des différents services de l'entreprise .....	14
2.1.3	Organigramme : .....	16
2.2	Présentation de Salesforce : .....	16
2.2.1	Historique et présentation : .....	16
2.2.2	Présentation du CRM .....	17
2.2.3	Salesforce et le développement d'applications : .....	21
2.3	Présentation de Project Management : .....	29
2.3.1	Présentation : .....	29
2.3.2	Objectifs de Project Management .....	29
2.3.3	Objectifs de l'entreprise à travers le développement d'un module de gestion des risques : .....	29
2.3.4	Rôles des différents acteurs de l'entreprise dans la Gestion des risques : .....	29
2.3.5	Organisation du projet : .....	30
2.3.6	Les objectifs du stage : .....	30

---

## 2 Présentation de Salesforce et de Project Management :

### 2.1 Présentation de la structure d'accueil :

#### 2.1.1 Présentation générale :

La Société Téranga Cloud Solutions est un partenaire intégrateur certifié de Salesforce, plateforme n° 1 de solution de gestion de la relation client (en anglais CRM<sup>8</sup>). Installé à Thiès, ville qui est située à 70 km de Dakar capitale du Sénégal, Téranga Cloud Solutions est le leader dans le domaine de l'intégration CRM en Afrique de l'ouest.

Elle propose aussi des services de développement Salesforce, d'intégration et de formation en passant par l'audit de votre CRM, la migration de données, l'administration et le support AM mais aussi des solutions BI<sup>9</sup> afin d'aider les entreprises à identifier, séduire et fidéliser de plus en plus de client.



*Figure 8 Emplacement Téranga Cloud Solutions*

Elle propose des services de développement Salesforce, d'intégration et de formation en passant par l'audit de votre CRM, la migration de données, l'administration et le support AM<sup>10</sup> mais aussi des solutions BI afin d'aider les entreprises à identifier, séduire et fidéliser de plus en plus de client.

---

<sup>8</sup> Customer Relationship Management

<sup>9</sup> Bussiness Intelligence

<sup>10</sup> AM:



*Figure 9 Logo Téranga Cloud Solutions*

### **2.1.2 Présentation des différents services de l'entreprise**

L'entreprise propose différents services à savoir :

- ✓ Un audit CRM et une réingénierie des processus :

Ce processus permet à une entreprise de gérer de façon efficace les données clients en permettant une accessibilité simple et automatique et une meilleure gestion de la communication. A travers un examen complet du code, de l'architecture applicative et de la plateforme, l'audit CRM permet : une identification des freins de performances qui pénalisent l'application, et de préconiser les solutions adaptées aux difficultés rencontrées .Elle permet aussi de remodeler votre application au niveau de son architecture et/ou de son code à moindre coût et dans des délais raisonnables.

- ✓ Développements d'applications :

Pour chaque projet une méthodologie rigoureuse est appliquée :

- Une aide au choix de la solution ;
- Un cadrage et définition du besoin ;
- Une gestion de la conduite du changement ;
- Une définition de l'architecture fonctionnelle ;
- Un pilotage de projet et assistance méthodologique ;
- Des solutions développées dans une « sandbox » ;
- Une validation accélérée en mettant en relation les développeurs et les équipes opérationnelles autour d'un démonstrateur ;
- Un déploiement fait par étapes sans interruption de services ;
- Une maintenance prévue dès la phase de conception pour réduire les coûts ;
- Une assurance d'un haut niveau d'adoption des utilisateurs grâce à la formation.

✓ Mise en œuvre et déploiement Salesforce :

Téranga Cloud Solutions vous permet de personnaliser votre CRM et de l'intégrer avec votre SI à travers :

- ✚ La Modélisation des processus / workflow de l'entreprise ;
- ✚ L'Intégration par connecteurs, mash-ups avec les autres applications de votre SI (Google apps, Microsoft Outlook, box...) ;
- ✚ L'export vers l'outil de production comptable (journal des ventes JDV, journal de situation JDS, ...) Sage comptabilité générale/analytique.

✓ Formations :

L'entreprise propose aussi des formations qui permettent d'acquérir des compétences dans le domaine Salesforce. Elle est surtout connue au niveau international pour ses solutions en gestion de la relation client.

Les sociétés peuvent notamment profiter des dernières innovations mobiles, cloud et issues des réseaux sociaux, pour mieux gérer leurs ventes, leur service client et leur marketing.

Téranga Cloud Solutions, vous propose ci-dessous les formations Salesforce les plus pertinentes de l'éditeur :

- Formation utilisateur commercial ou marketing ;
- Formation manager ;
- Formation administrateur ;
- Formation développeur.

Elle propose aussi d'autres services comme la mise en place d'une solution CRM Salesforce® performante et adaptée à votre métier ce qui permet :

- ❖ D'optimiser vos processus métiers et les temps de réponse de vos applications ;
- ❖ D'analyser et de mettre en place des solutions d'intégration de données ;
- ❖ De mettre en place des fonctionnalités complémentaires ;
- ❖ De refondre votre application pour l'adapter à vos besoins ;
- ❖ De définir les règles de sécurité (rôles et profils) de votre organisation ;
- ❖ D'implémenter des workflows pour automatiser vos processus, assigner des tâches, envoyer des relances par email ;
- ❖ De Mettre en place de formulaires web pour faire le lien entre votre site Internet et votre CRM ;
- ❖ De mettre en place des rapports et tableaux de bord Salesforce.

### 2.1.3 Organigramme :

L'entreprise Téranga cloud solutions comporte plusieurs services structurés de la façon suivante :

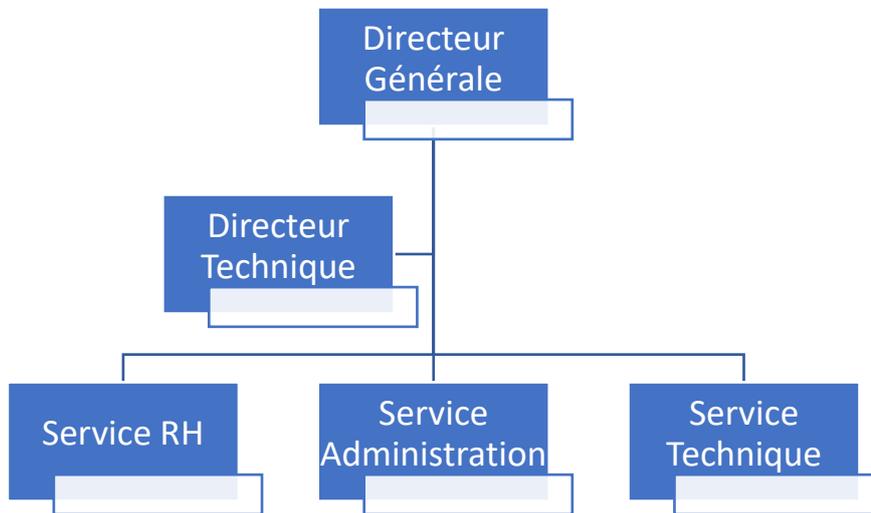


Figure 10 Organigramme de L'entreprise

## 2.2 Présentation de Salesforce :

### 2.2.1 Historique et présentation :

Salesforce a été créé en 1999 par Marc Bénéioff, Parker Harris, Dave Moellenhoff et Frank Dominguez et a fait ses débuts en tant que société spécialisée dans les logiciels SaaS (Software as a service).

Salesforce<sup>11</sup> est un éditeur de logiciels, basé à San Francisco aux États-Unis. Il distribue des logiciels de gestion basés sur Internet et héberge des applications d'entreprises. L'entreprise est surtout connue au niveau international pour ses solutions en gestion de la relation client.

Depuis sa création, Salesforce mène une politique active d'acquisitions, procédant au rachat d'une vingtaine de sociétés de toutes tailles, afin de renforcer sa présence sur certains marchés spécifiques comme les outils pour les directions marketing et conserver son leadership.

---

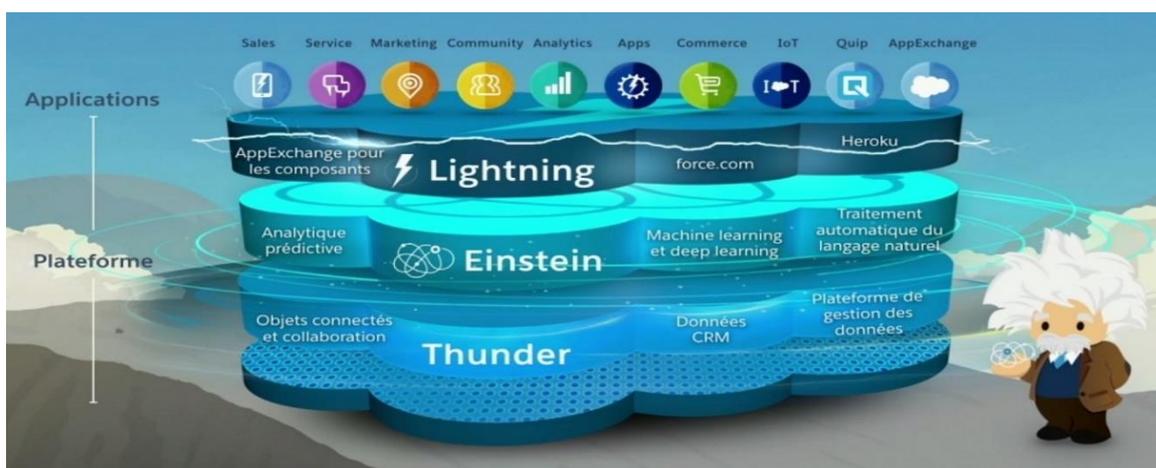
<sup>11</sup> [www.salesforce.com](http://www.salesforce.com)

## 2.2.2 Présentation du CRM

Salesforce est un CRM<sup>12</sup> entièrement basé sur le cloud. Un CRM est une application utilisée pour automatiser les ventes et les fonctions de marketing en utilisant un logiciel appelé CRM Software.

Exemple de quelques CRM : SAP CRM, Sage CRM Oracle CRM.

Salesforce est basé sur une architecture en couches superposées :



*Figure 11 Architecture de Salesforce*

Ce qu'on peut comprendre de la plateforme Salesforce c'est que tout ce qu'il offre se trouve dans le cloud.

La plateforme Salesforce offre un ensemble de services dont les plus connus sont : Sales Cloud, Services Cloud Appexchange, Chatter, Marketing cloud, Analytics Cloud...

---

<sup>12</sup> Customer Relationship Management



Figure 12 Les service de Salesforce

✓ Sales Cloud

SalesCloud est une plateforme CRM conçue pour assister les commerciaux et les services des ventes. Sales Cloud est un produit SaaS (software as a service) personnalisable qui rassemble toutes les informations concernant les clients dans une plateforme intégrée. Les données marketing, génération de leads, ventes, service client et analytique métier sont ainsi centralisées et accessibles. Cette plateforme Cloud est accessible via un navigateur. Une application mobile est également disponible. Un fil social en temps réel permet en outre aux utilisateurs de collaborer en partageant des informations ou en posant des questions à la communauté d'utilisateurs.



Figure 13 SalesCloud

✓ Marketing cloud :

Salesforce Marketing Cloud est une plateforme CRM en mode SaaS qui permet aux acteurs du marketing de créer et gérer les relations avec les clients et les campagnes marketing. Marketing Cloud intègre des solutions de gestion des parcours client ainsi que des fonctions de messagerie, d'itinérance, de réseautage social, de personnalisation Web, de publicité, de création et de gestion de contenu, et d'analyse de données. Des déclencheurs basés sur les événements peuvent lancer certaines actions. Ainsi, lorsqu'un client adhère à un programme de fidélité, Marketing Cloud peut automatiquement lui envoyer un message de bienvenue, tandis que d'autres événements peuvent déclencher une mise à jour des données dans l'enregistrement de contact du client.



*Figure 14 Marketing Cloud*

✓ Analytics Cloud :

Analytics Cloud est la première plateforme cloud analytique. Elle est accessible depuis un ordinateur, un smartphone ou une tablette. La plateforme permet aux entreprises d'analyser leurs données, de partager les analyses entre collaborateurs et d'en tirer les informations nécessaires, afin d'identifier des opportunités commerciales.



*Figure 15 Analytics Cloud*

✓ Chatter et Community Cloud :

Chatter est une plateforme de collaboration de type réseau social pour les entreprises éditée par Salesforce, fournisseur de solutions CRM dans le Cloud. Chatter peut servir d'intranet pour l'entreprise ou de répertoire pour les employés. Chaque employé possède une page de profil où figurent sa photo ainsi que des informations à caractère professionnel indiquant son rôle dans l'entreprise, son supérieur hiérarchique, son emplacement géographique et ses coordonnées. Les employés peuvent « suivre » aussi bien des personnes que des documents en vue de collaborer sur des opportunités commerciales, des demandes de service, des campagnes, des projets et des tâches. Tout comme Facebook et LinkedIn, Chatter permet aux utilisateurs de gérer leurs flux et de contrôler les paramètres de réception des notifications. Les développeurs peuvent incorporer les caractéristiques et fonctionnalités de Chatter dans des applications créées sur la plateforme Force.com.



*Figure 16 Chatter*

✓ AppExchange :

Salesforce AppExchange est la place de marché Cloud de Salesforce.com qui permet de consulter, de télécharger et d'installer des applications logicielles. AppExchange offre une combinaison d'applications gratuites et payantes regroupées au sein de collections, de catégories et de solutions sectorielles englobant une variété de domaines, comme les petites entreprises, le service client, l'enseignement, la fabrication et l'immobilier.



*Figure 17 Appex Change*

### **2.2.3 Salesforce et le développement d'applications :**

Salesforce permet le développement d'applications métiers entièrement basé sur le cloud.

Avec Salesforce on a la possibilité de développer des applications de deux manières : en faisant du point and click (sans code) ou du développement avec du code.

Avec le point and click, Salesforce offre des outils de développement déclaratifs. Ces outils ne nécessitent pas de connaissances sur les principes de développement.

Exemple : Le générateur d'application Lightning et le process Builder.

Ce modèle de développement piloté par les métadonnées est l'une des principales différences entre le développement sur Salesforce et le développement hors de la plateforme. Les métadonnées formeront la structure de notre application ; elles contiennent en effet la configuration. La plateforme restitue ensuite les métadonnées de notre application sur l'interface utilisateur avec les données associées. Pour le développement avec du code Salesforce donne la possibilité d'utiliser un ensemble d'outils :

### 2.2.3.1 La base de données Force.com

La base de données Force.com est une application conçue pour simplifier le développement d'applications et de site web dans le cloud. La base de données Force.com est utilisée pour personnaliser notre modèle de données, notre logique métier et notre interface utilisateur.

Elle permet de créer des objets, de créer des relations entre les objets, d'assurer l'intégrité des données et d'auditer facilement le changement de la base.

Elle utilise le schéma builder pour gérer la création des objets et leurs relations.

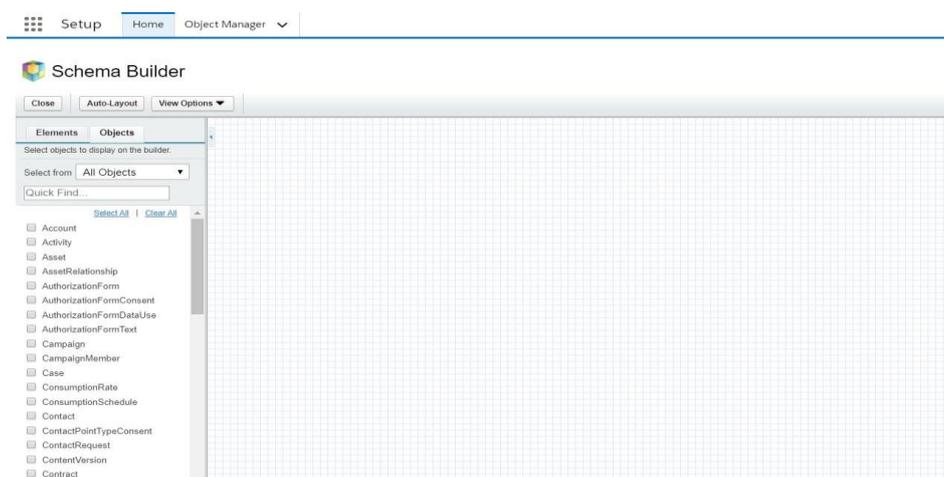


Figure 18 Le schéma Builder de Salesforce

### 2.2.3.2 La Console du développeur (Dev Console):

Le développeur console est un environnement de développement intégré qui nous permet de créer, déboguer ou tester les différentes applications de notre organisation.

Sous Salesforce on peut y accéder à partir du menu configuration.

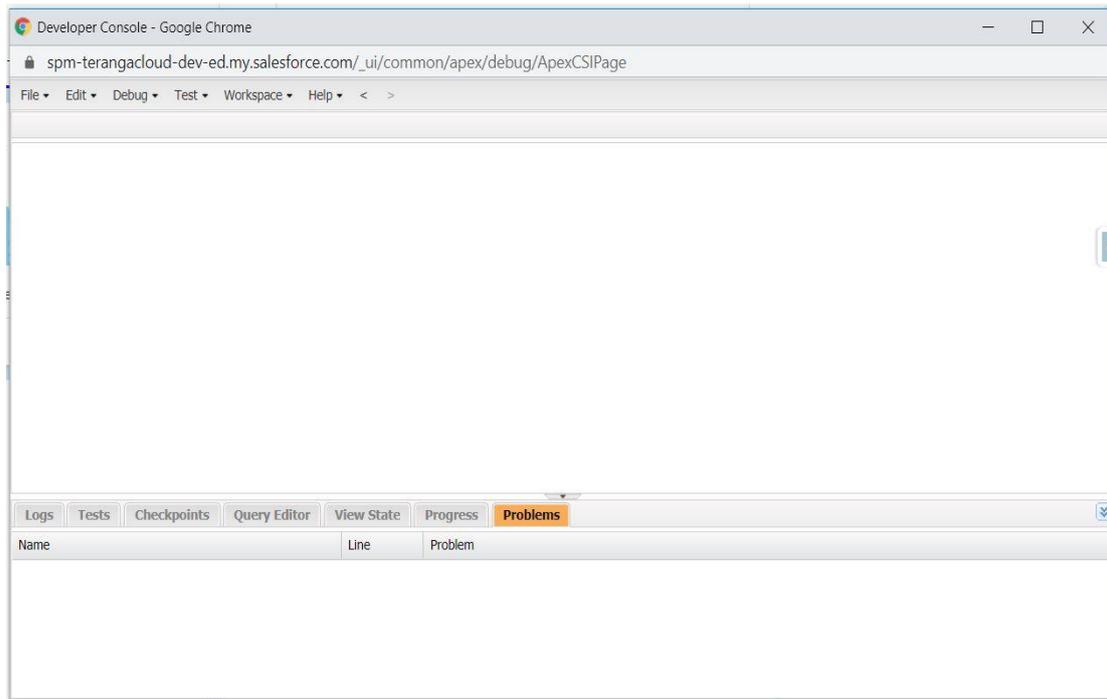


Figure 19 La Dev Console

### 2.2.3.3 L'IDE Force.com:

Force.com permet aux développeurs de créer et de déployer rapidement des applications cloud fiables, solides, sécurisées et évolutives sans avoir à se soucier du dimensionnement du matériel ou des piles d'applications. Pour vous aider à aller plus vite, Force.com offre des outils et des services prêts à l'emploi pour automatiser tous nos processus d'affaires, pour intégrer des applications externes et pour offrir aussi des expériences mobiles.

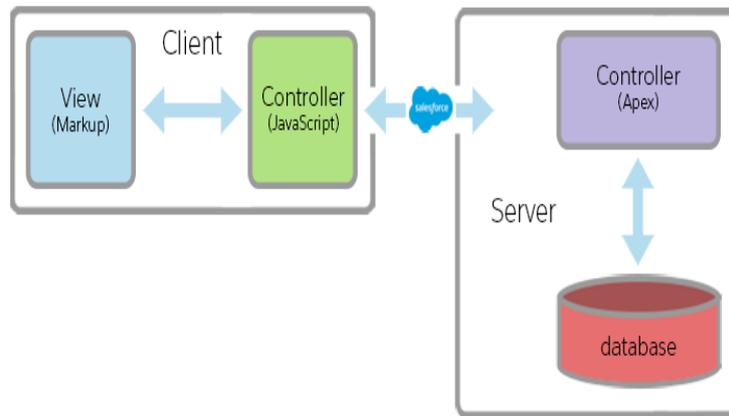
### 2.2.3.4 Le Framework Lightning:

Le Framework Lightning Component est un Framework d'interface utilisateur pour le développement d'applications à page unique pour les appareils mobiles et de bureau. Lightning n'est pas un Framework MVC<sup>13</sup> mais un Framework basé sur les composants. Elle offre deux modèles de programmation : le modèle Lightning Web Components et de nos jours le modèle Aura Components.

---

<sup>13</sup> Modèle Vue Controlleur

Aura Framework est un Framework open-source créé par Salesforce pour créer des applications Web dynamiques pour mobiles et postes de travail. Salesforce utilise Aura pour développer des applications. Salesforce Lightning est créé à l'aide du Framework Aura. Il repose sur du JavaScript coté client et du Apex coté serveur.



*Figure 20 Les deux contrôleurs du Framework*

Le Framework Lightning présente plusieurs avantages de développement :

- Il aide à accélérer le développement et à améliorer les performances.
- De plus, on n'a pas besoin de consacrer notre temps à la compatibilité, mais l'application Lightning utilise un design réactif et offre une excellente expérience utilisateur. Le Framework Lightning Component prend également en charge les dernières technologies de navigation, telles que HTML5, CSS3 et les événements tactiles.
- Il s'agit d'une architecture événementielle pour un meilleur découplage entre les composants.
- Il utilise aussi une architecture de serveurs qui s'appuie sur JavaScript du côté client pour gérer les métadonnées et les données d'application des composants graphiques. Le client n'appelle le serveur que lorsqu'il est absolument nécessaire ; par exemple pour obtenir plus de métadonnées ou de données. Le serveur envoie uniquement les données nécessaires à l'utilisateur pour maximiser l'efficacité. Le Framework utilise JSON pour échanger des données entre le serveur et le client. Il utilise intelligemment notre serveur, notre navigateur, nos périphériques et notre réseau afin que nous puissions nous concentrer sur la logique et les interactions de nos applications.
- Les composants sont encapsulés et leurs composants internes restent privés, tandis que leur forme publique est visible par les consommateurs du composant. Cette forte

séparation donne aux auteurs des composants la liberté de modifier les détails internes de l'implémentation et isole les consommateurs des composants de ces changements.

- Le framework est compatible avec les périphériques et les navigateurs croisés.
- Les applications utilisent une conception réactive et prennent en charge les dernières technologies de navigation telles que HTML5, CSS3 et les événements tactiles.

#### 🚦 Architecture

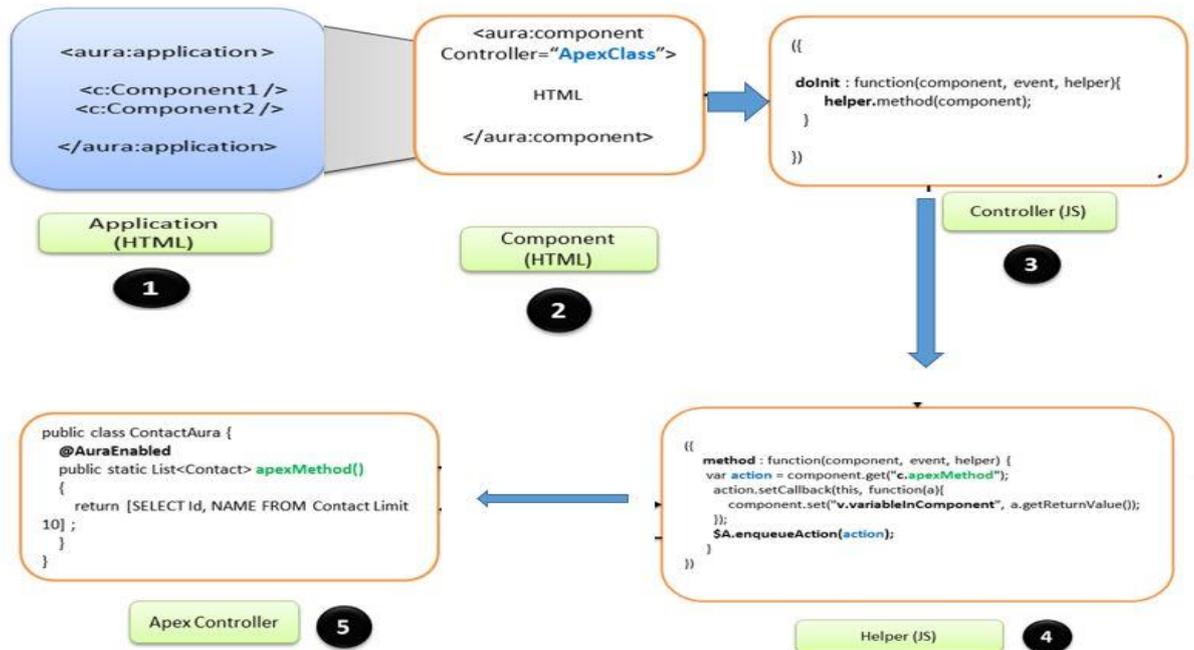


Figure 21/Architecture du framework lightning

#### 1 Application :

L'application est utilisée pour exécuter le composant Lightning. On peut inclure dans l'application les composants à tester. Il s'agit du composant de niveau supérieur qui contient tous les autres composants.

#### 2 Component :

Elle permet de construire une interface utilisateur en utilisant des composants ainsi qu'une collection de composants d'entrée Lightning pour collecter les données utilisateur. Une fois les composants créés, il y a plusieurs options que nous pouvons voir dans la barre latérale et quelques-unes de ces options comme suit :

### 3 Controller (JS) :

Le contrôleur est un fichier qui contient les méthodes de contrôleur JavaScript côté client pour gérer les événements déclenchés et traités par les composants.

### 4 Helper

Ce fichier contient les fonctions JavaScript qui peuvent être appelées à partir de n'importe quel code JavaScript d'un paquet de composants. Il s'agira d'une classe Helper pour le contrôleur. Il sera utile lorsque nous aurons besoin de réutiliser la même logique ailleurs.

### 5 Apex Controller :

Le contrôleur serveur représente l'ensemble des classes écrites dans le langage apex.

#### Le langage apex :

Le langage apex est un langage de programmation objet et fortement typé qui permet aux développeurs d'exécuter des instructions de contrôle de flux et de transaction sur le serveur de la plateforme Force.com. Sa syntaxe est très proche du langage java. Apex permet aux développeurs d'ajouter une logique applicative à la plupart des événements système, y compris aux clics de bouton, aux mises à jour d'enregistrements associés. Le code Apex peut être initialisé par des demandes émanant de services Web et de déclencheurs d'objets. C'est un langage facile à tester. Le langage Apex fournit une prise en charge intégrée de la création et de l'exécution de tests unitaires, qui comprennent des résultats de tests indiquant la quantité de code couverte et les parties de votre code qui peuvent être améliorées.

#### Autres particularités du Framework.

Le Framework Lightning permet de créer des événements qui seront déclenchés en cas d'une action. Ils existent deux types d'évènements : les évènements de type composant et les évènements de type application.

Un Événement de type composant ne permet de communiquer qu'avec le composant parent et lui-même alors que l'évènement de type application est utilisé pour diffuser à d'autres composants et pas exclusivement des ancêtres. Les événements applicatifs peuvent communiquer avec plusieurs composantes qui sont intéressées par l'évènement c'est-à-dire peuvent communiquer avec toute l'application.

Le Framework utilise aussi tous les aspects de la programmation orientée objet comme : Extensible, Abstract, Extends, Implements.

Un composant créé peut avoir les attributs suivants :

- Access (Type String) : Cet attribut indique si le composant peut être utilisé en dehors du composant lui-même.
- Contrôler (Type String) : La classe contrôler coté serveur apex.
- Description (Type String) : La description du composant.
- Extends (Type Composant) : Le composant à étendre.
- Extensible (Type Booléen) : Initialisé à true si le composant peut être étendu. Sa valeur par défaut est false.
- Implements (Type string) : Une liste d'interface séparée par une virgule que le composant peut implémenter.
- Is Template (Type Booléen) : Cet attribut est défini à true si le composant est un Template. La valeur par défaut est false. Un Template doit avoir l'attribut « Is Template= True » défini dans sa balise aura component.
- Template (Type Composant) : Le Template démarre dès le chargement du Framework et de l'application. Le Template par défaut est AURA Template. On peut personnaliser le Template en créant notre propre composant qui pourra étend le Template par défaut.

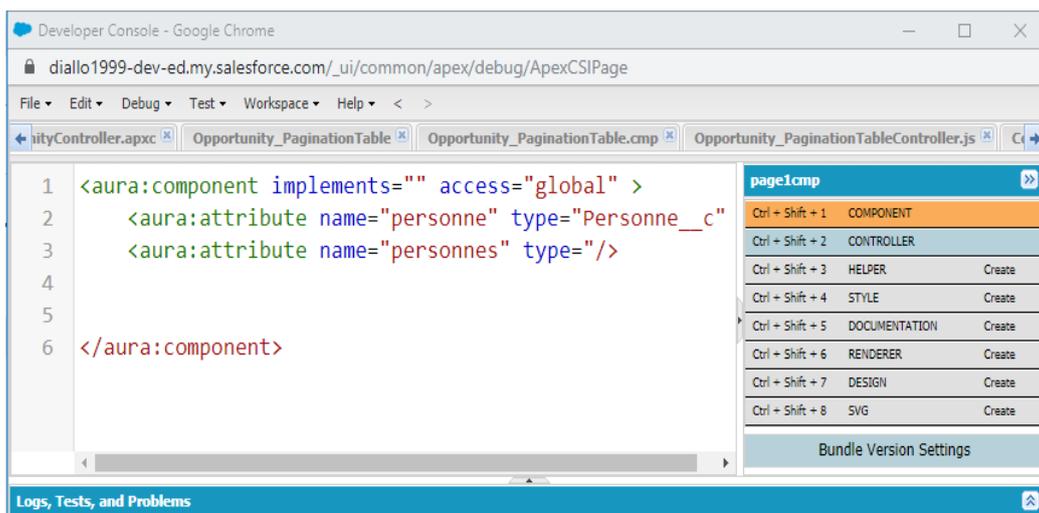


Figure 22 Structure d'un Composant

### 2.2.3.5 Notion environnement dans Salesforce:

Le déploiement dans Salesforce peut se faire suivant différentes organisations ou environnement. Sous Salesforce un environnement ou une organisation est une instance de la plateforme Force.com. Un environnement Salesforce a les caractéristiques suivantes :

- Il peut être utilisé pour le développement, les tests et / ou la production.
- Il contient des données (enregistrements) et des personnalisations (objets et champs de base de données personnalisés, code Apex, Visual force, flux de travail, etc.)
- Chaque environnement est basé sur une édition, qui contient des fonctionnalités spécifiques, des objets, du stockage et des limites.
- Par défaut, les environnements ne sont pas provisionnés avec certaines fonctionnalités avancées.

Exemple La technologie multidevise ou de prévisualisation de développeur.

- Nous pouvons demander l'activation de fonctionnalités avancées en contactant le service clientèle de salesforce.com.
- Tous les environnements peuvent être accessibles via un navigateur Web, mais certains peuvent également être accessibles depuis l'IDE Force.com, l'API SOAP et l'API Métadata.

Exemple : Lorsqu'on s'inscrit à Gmail, on obtient un nom d'utilisateur unique et un accès à notre compte de messagerie privé dans le cloud. De même, lorsqu'on s'inscrit à une organisation Force.com, on obtient un nom d'utilisateur unique et un accès privé à notre environnement de cloud computing Force.com.

De manière générale, il existe trois types d'environnements :

- ✓ Les environnements de production et environnement Salesforce.com qui ont des utilisateurs payants actifs accédant aux données critiques de l'entreprise.
- ✓ Les environnements de développement où nous pouvons étendre, intégrer et développer sur Force.com sans affecter nos environnements de production.
- ✓ Les environnements de test. Il peut s'agir d'environnements de production ou de développement spécifiquement utilisés pour tester les fonctionnalités des applications avant de les déployer en production ou de les mettre à la disposition de nos clients.

## **2.3 Présentation de Project Management :**

### **2.3.1 Présentation :**

Project Management est une application développée pour le compte de l'entreprise Téranga Cloud Solutions. Elle est une application de gestion de projet qui permet de gérer un projet depuis sa création jusqu'à sa date d'échéance à travers différents modules.

### **2.3.2 Objectifs de Project Management**

L'application Project management a pour objectif de :

- ✓ Permettre à l'entreprise de faire de la gestion de projet de façon dynamique
- ✓ Permettre à l'entreprise d'avoir une vue globale sur tous les projets
- ✓ Permettre à l'entreprise de gérer tous les risques auxquels elle peut être confrontée

Project Management comporte plusieurs modules :

- Gestion du planning et des collaborations
- Gestion des activités et des TimeSheet
- Gestion des dépenses et des finances
- Gestion du Reporting
- Gestion des risques

### **2.3.3 Objectifs de l'entreprise à travers le développement d'un module de gestion des risques :**

A travers le développement d'un module de gestion des risques l'entreprise Téranga Cloud Solutions doit pouvoir manager et contrôler les risques mais aussi faire des évaluations des risques. Ce module doit permettre aussi de suivre l'évolution des risques que l'entreprise peut rencontrer pendant le déroulement d'un projet.

### **2.3.4 Rôles des différents acteurs de l'entreprise dans la Gestion des risques :**

Au sein de l'entreprise Téranga Cloud Solutions il existe plusieurs acteurs qui interviennent dans des projets. Il y'a le système administrator<sup>14</sup> qui peut créer et visualiser des risques, le project Manager<sup>15</sup> (qui fait office de risk manager<sup>16</sup>). Celui-ci est l'acteur principale de la

---

<sup>14</sup> Super Utilisateur

<sup>15</sup> Manageur de projet

<sup>16</sup> Manageur des risques

gestion des risques dans un projet. Il crée un risque et définit les paramètres sur lesquels on quantifie le risque. Il a accès aux différents tableaux de bords et aux rapports sur les risques.

### 2.3.5 Organisation du projet :

Notre module, gestion des risques est une partie de Project Management.

Il a été développé au sein de l'entreprise par plusieurs personnes :

Acteurs	Module à Développer
Dr Mamadou Bousso	Responsable technique Chargé du suivi du projet
Moussa Fofana	Project Manager chargé de la définition des tâches
Abdoulaye Diallo	Gestion des risques
Moustapha Diouf	Reporting
Dieynaba Diallo	Gestion du planning et des collaboration
Stephane Ouédraogo	Gestion des activités et du TimeSheet
Youssouph Diédhiou	Gestion de la Finance et des dépenses

### 2.3.6 Les objectifs du stage :

L'objectif principal de notre stage c'est de développer un module de gestion des risques pour le compte de l'entreprise Téranga Cloud Solutions. Ce module sera intégré dans l'application Project Management. L'entreprise à travers ces différents projets rencontre plusieurs risques. De ce fait ce module lui permettra de gérer les risques auxquels elle sera confrontée.

#### Conclusion :

Salesforce est un CRM entièrement basé dans le cloud et permet de développer des applications métiers en utilisant soit le point and click soit le framework Aura Lightning Component. Notre application (Project Management) comprend plusieurs modules dont la gestion des risques.

L'entreprise à travers ce projet va développer son propre module de gestion des risques et pourra gérer les différents risques dont elle sera confrontée.

Cette présentation nous a permis de comprendre la technologie qu'on a utilisé pour développer notre module mais aussi l'objectif principal de l'entreprise à travers le développement de cette application.

## Chapitre 3

### La Gestion des risques

#### Introduction :

Dans ce chapitre nous parlerons de la gestion des risques à savoir les normes et les différentes applications de gestion des risques dans Salesforce à travers un tableau comparatif.

Enfin nous parlerons des objectifs à atteindre pour notre sujet.

---

Chapitre 3 :La Gestion des risques : .....	33
3.1 Les normes de Gestion des risques : .....	33
Gestion des risques des dispositifs médicaux .....	34
3.2 Justification du choix de la norme : .....	35
3.3 La norme de Gestion des risques ISO 31000 2009 .....	35
3.3.1 Historique et Présentation : .....	35
3.3.2 Le but de la Norme : .....	36
3.3.3 Les principes de la norme : .....	36
3.3.4 Certification de la norme : .....	38
3.3.5 Structure de la Norme : .....	38
3.3.6 L'évaluation des risques : .....	39
3.4 Etude de l'existant : .....	40
3.4.1 Présentation des applications de gestion des risques dans Salesforce : .....	40
3.4.2 Etude comparative entre les deux applications : .....	41
3.4.3 Objectifs à atteindre pour notre sujet : .....	43

---

### 3 La Gestion des risques :

Le risque est un danger, un inconvénient plus ou moins probable auquel on est exposé. Le risque est dans tous les domaines. Dans chaque domaine il existe une famille de normes qui permettent de faire de la gestion de risque.

#### 3.1 Les normes de Gestion des risques :

- Justification du choix de la norme Iso :

L'ISO est le plus grand organisme de normalisation au monde. Elle est composée de représentants d'organisations nationales de normalisation de 165 pays.

Exemple :

Pays	Organisme de normalisation
France	AFNOR <sup>17</sup>
Argentine	IRAM <sup>18</sup>
Allemagne	DIN <sup>19</sup>

- Comparaison entre les différentes normes ISO :

Pour cette multitude de normes il existe des caractéristiques spécifiques qui permettent de les classer soit par leur domaine d'application soit par la manière dont elles gèrent le risque.

Norme	Domaine d'application	But
ISO 14000	Pour le management des risques environnementaux	Cette norme donne des outils pratiques aux entreprises et aux organisations de tous types qui souhaitent maîtriser leurs responsabilités environnementales
ISO 27000	Pour le management des risques liés à la sécurité Informatique	La famille de normes ISO 27000 aide les organisations à assurer la

<sup>17</sup> Association Française de Normalisation

<sup>18</sup> Instituto Argentino de normalization

<sup>19</sup> Deutsches Institut fur Normung

		sécurité de leurs informations
ISO 14971	Gestion des risques des dispositifs médicaux	À mettre en place une gestion efficace des risques induits par un dispositif médical, une activité imposée par la réglementation Européenne.
ISO 9000	Management des risques de qualité	Elles offrent des lignes directrices et des outils aux entreprises et aux organismes qui veulent que leurs produits et services soient constamment en phase avec ce que leurs clients demandent et que la qualité ne cesse de s'améliorer.
ISO 31000	Management du risque et lignes directrices	Elle s'applique à tout type de risque et à tout type d'entreprise
ISO 17666 :2003	Management des risques	Elle présente les éléments nécessaires à la mise en œuvre d'une politique de management des risques intégrée par tout acteur du projet
ISO 31010	Gestion des risques et techniques d'évaluation des risques	Elle traite des concepts de l'évaluation des risques, des processus et de la sélection des techniques d'évaluation des risques auxquels sont confrontés l'entreprise

## 3.2 Justification du choix de la norme :

Parmi ces normes certaines sont appliquées à un domaine bien spécifique. Cependant la norme ISO 31000 2009 n'a pas de limites elle s'applique à tout type de risque et à tout type d'entreprise. C'est une norme qui coiffe les autres normes de gestion des risques et peut être appliquée dans plusieurs domaines de la gestion des risques.

La mise en œuvre d'ISO 31000 permet, par exemple, à une entreprise :

- ✚ D'augmenter la probabilité que les objectifs de l'entreprise seront atteints
- ✚ De prendre conscience de la nécessité d'identifier et de traiter le risque à travers tout l'organisme
- ✚ D'améliorer l'identification des opportunités et des menaces
- ✚ De se conformer aux obligations légales et réglementaires, ainsi qu'aux normes internationales
- ✚ D'améliorer la gouvernance
- ✚ D'établir une base fiable pour la prise de décision et la planification
- ✚ D'améliorer les contrôles
- ✚ D'allouer et d'utiliser efficacement les ressources pour le traitement du risque
- ✚ D'améliorer l'efficacité et l'efficience opérationnelle
- ✚ D'améliorer la prévention des pertes et le management des incidents
- ✚ De minimiser les pertes
- ✚ D'améliorer l'apprentissage organisationnel

## 3.3 La norme de Gestion des risques ISO 31000 2009

### 3.3.1 Historique et Présentation :

En novembre 2009, la nouvelle norme internationale ISO 31000 Management des risques fut publiée avant d'être adoptée par l'AFNOR<sup>20</sup> sous NF<sup>21</sup> 31000-2009.

Cette norme propose différents principes et des lignes directrices du management des risques ainsi que les processus de mise en œuvre au niveau stratégique et opérationnel.

---

<sup>20</sup> Association Française de normalisation :

[https://fr.wikipedia.org/wiki/Association\\_francaise\\_de\\_normalisation](https://fr.wikipedia.org/wiki/Association_francaise_de_normalisation)

<sup>21</sup> Norme Française:

Selon la norme Iso 31000 2009 le risque représente l'effet de l'incertitude sur l'atteinte des objectifs.

### **3.3.2 Le but de la Norme :**

Le but de la norme ISO 31000 2009 est de s'appliquer et de s'adopter à tout public, toute entreprise publique ou privée, toute collectivité toute association tout groupe d'individus.

Il ne s'agit pas d'uniformiser les pratiques, ni de créer un système de management parallèle. Au contraire la norme ISO 31000 2009 propose un référentiel unique pour les organisations de tout secteur et de toute taille.

Elle est adaptable et suffisamment flexible pour harmoniser les processus de management de tous les types de risques faisant peser une incertitude sur l'atteinte des objectifs de l'entreprise.

L'approche proposée par la norme ISO 31000 permet de formaliser les pratiques de management des risques tout en permettant aux entreprises de mettre en place un cadre ERM.

### **3.3.3 Les principes de la norme :**

Le management du risque selon la norme ISO 31000 s'articule autour de 11 principes :

- ✓ Le management des risques crée de la valeur et la préserve : Le management des risques contribue de façon tangible à l'atteinte des objectifs et à l'amélioration des performances de l'organisation à travers la révision de son système et ses processus
- ✓ Le management des risques est intégré aux processus d'organisation : Le management des risques doit être intégré dans le système de management existant tant au niveau stratégique qu'au niveau opérationnel.
- ✓ Le management des risques est intégré aux processus de décision : Le management des risques est une aide à la décision pour faire des choix augmentés pour définir des priorités et pour sélectionner les actions les plus appropriées.
- ✓ Le management des risques traite explicitement de l'incertitude : En identifiant les risques potentiels, l'organisation peut mettre en place des outils de réduction et de financement des risques dans le but de maximiser les chances de succès et de minimiser les possibilités de perte.
- ✓ Le management des risques est systématiquement structuré et utilisé en temps utile : Les processus du management des risques devraient être cohérents à travers l'organisation afin d'assurer l'efficacité, la pertinence, la cohérence et la fiabilité des résultats.

- ✓ Le management des risques s'appuie sur la meilleure information disponible : Pour un management des risques efficace, il est important de considérer, de comprendre toutes les informations disponibles et pertinentes pour une activité tout en reconnaissant les limites des données et des modèles utilisés.
- ✓ Le management du risque intègre des facteurs humains et culturel : Le management du risque doit reconnaître la contribution des personnes et des facteurs culturels à la réalisation des objectifs de l'organisation.
- ✓ Le management des risques est transparent et participatif : En impliquant les parties prenantes, internes et externes lors des processus de management des risques, l'organisation reconnaît l'importance de la communication et de la consultation lors des étapes de l'identification, d'évaluation et de traitement des risques.
- ✓ Le management des risques est dynamique et itératif et réactif au changement : Le management des risques doit être flexible. L'environnement concurrentiel oblige l'organisation à s'adapter au contexte interne et externe, spécialement lorsque de nouveaux risques apparaissent, lorsque certains risques sont modifiés tandis que d'autres disparaissent.
- ✓ Le management des risques facilite l'amélioration continue de l'organisation : Les organisations possédant une maturité en matière de management des risques sont celles qui investissent à long terme et qui démontrent la réalisation régulière de ses objectifs.

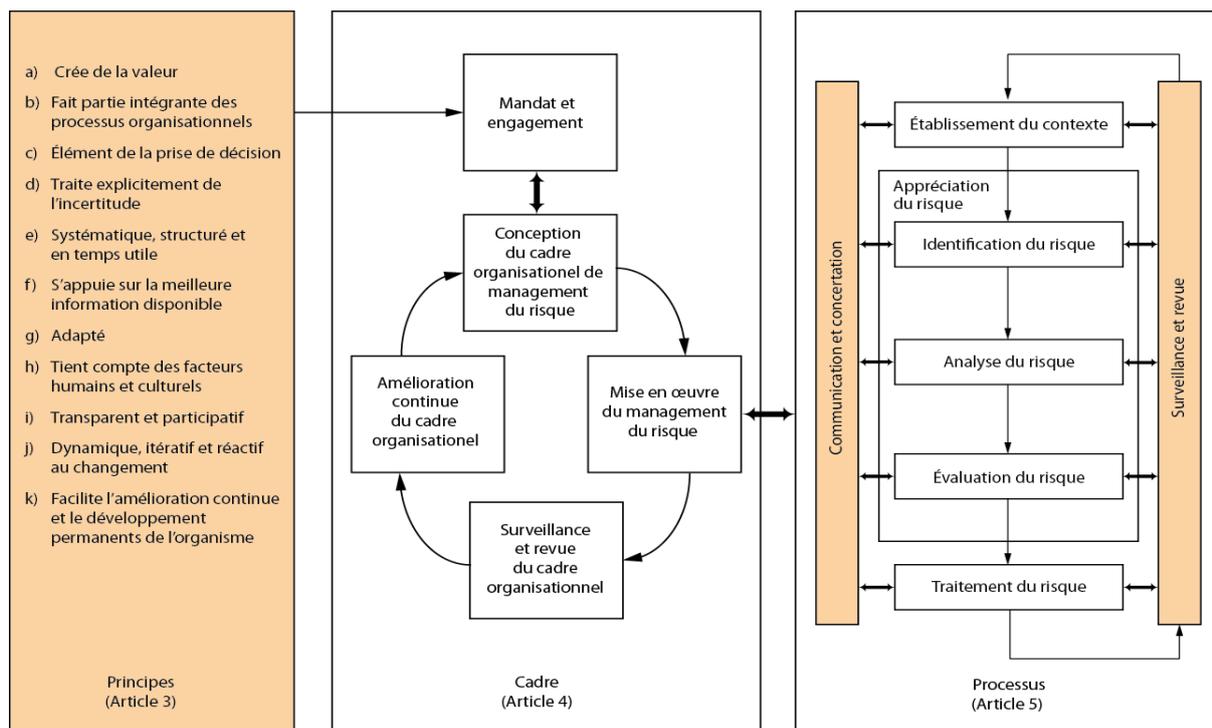


Figure 23 Norme ISO 31000 2009

### 3.3.4 Certification de la norme :

La norme ISO 31000 2009 ne fait que définir des lignes directrices, et non des exigences ce qui fait qu'elle n'est pas destinée à la certification.

### 3.3.5 Structure de la Norme :

La norme ISO 31000 2009 présente une structure tripartite : Les principes, le cadre organisationnel et le processus de management.

- Les principes :

Les principes répondent à la question pourquoi fait-on du management des risques. Le processus d'intégration de ces principes se fait ensuite à deux niveaux : le niveau décisionnel et le niveau opérationnel

- Le cadre organisationnel :

Le cadre d'organisation explique comment intégrer, via le processus itératif de la roue de Deming (Plan-Do-Check-Act), le management des risques dans la stratégie de l'organisation (conduite stratégique).

- Le principe de management :

Le processus de management précise comment intégrer le management des risques au niveau opérationnel.

La norme ISO 31000 2009 propose aussi la notion de ERM (Entreprise Risk Management) :

L'entreprise risk management (ERM) ou gestion intégrée des risques est une technique de management des risques destinée à aider les entreprises à gérer tout risque auquel elles doivent faire face, principalement aléatoire, financier, opérationnel et stratégique.



Figure 24 Le ERM

### 3.3.6 L'évaluation des risques :

Le risque est évalué selon trois critères : la fréquence, la gravité et la maîtrise.

- La fréquence d'apparition : La fréquence d'apparition du risque pendant le déroulement du projet ; Comprise entre 1 et 4

Niveau de fréquence	Interprétation
1	Très improbable
2	Improbable (rare)
3	Probable (occasionnel)
4	Très probable (fréquent)

- La gravité du risque : Comprise entre 1 et 4

Niveau de gravité	Interprétation
1	Faible
2	Moyenne
3	Grave
4	Très Grave

- La maîtrise comprise entre 0.5 et 1

Maitrise	Interprétation
0.5	Niveau 1
1	Niveau 2
2	Niveau 3

En fin la criticité d'un risque sera calculée sous la formule :

$$C = (F + G) * M$$

Avec C la criticité, F la fréquence, G la gravité et M la maîtrise.

La criticité sera représentée dans une matrice appelée matrice de criticité d'un risque :



Figure 25 Matrice des risques

La couleur verte sera utilisée quand la criticité est comprise entre 1 et 4

La couleur jaune si la criticité est comprise entre 4 et 8

La couleur orange si la criticité est comprise entre 8 et 12

Enfin la couleur rouge si la criticité est comprise supérieure ou égale à 12.

### 3.4 Etude de l'existant :

Dans le cadre général, il existe plusieurs applications de gestion des risques.

#### 3.4.1 Présentation des applications de gestion des risques dans Salesforce :

Etant un CRM Salesforce voit la gestion des risques de façon globale .D'abord elle s'intéresse à la gestions des risques au niveau de sa plateforme et aux différents risques externes que les

développeurs Salesforce rencontrent dans le développement des projets .Pour la gestion des risques Salesforce propose des applications de gestion des risques qu'on peut intégrer dans notre environnement .

Dans la boutique d'applications de Salesforce (AppexChange), il existe des applications de gestions des risques déjà développées.

Exemple : Xactium Risk<sup>22</sup>, Riskconnect<sup>23</sup> etc.

Toutes ces deux applications permettent de faire de la gestion des risques pour une entreprise quelconque. Xactium est l'un des systèmes de gestion des risques d'entreprise les plus flexibles et adaptables disponibles aujourd'hui et est reconnu comme représentant l'avenir pour les applications de gestion des risques.



Figure 26 Applications de Gestion des Risques dans AppexChange

### 3.4.2 Etude comparative entre les deux applications :

	Xactium Risk	Riskconnect
Type d'application	Payante 20 Euro par Utilisateur et par mois	Payante 6.000 \$ Us par mois
Domaines d'application	S'applique à tout type de risque suivant sa catégorie	S'applique à tout type de risque qui peut empecher l'entreprise d'atteindre ses objectifs

<sup>22</sup> <https://riskconnect.com/solutions/project-risk/>

<sup>23</sup> <https://www.xactium.com/>

Norme Utilisée	ISO 31000 et 3LOAD <sup>24</sup>	S'adapte à la Norme ISO 31000
Caractéristiques	<ul style="list-style-type: none"> <li>- Une plate-forme facile à utiliser, collaborative et configurable pour faciliter l'adoption et l'alignement des processus.</li> <li>- Reporting et partage de données flexibles pour une visibilité en temps réel des informations et tendances clés en matière de risques</li> <li>- Automatisation et intelligence artificielle intégrées pour améliorer radicalement l'efficacité des processus.</li> <li>- Agrégation des risques et aperçu prédictif pour une meilleure prise de décision.</li> </ul>	<ul style="list-style-type: none"> <li>- Un référentiel de données unique, incluant les flux d'autres systèmes</li> <li>- Profils de risque utilisant des filtres sélectionnés par l'utilisateur</li> <li>- Rapports de scénarios liés aux simulations</li> <li>- Fonctionnalité de workflow dans toute l'entreprise</li> <li>- Des tableaux de bord uniques pour tous les utilisateurs</li> <li>- Validation et escalade des risques émergents</li> <li>- Optimisation du contrôle</li> <li>- Gestion du comité des risques, y compris l'information destinée à l'examen des organismes de réglementation</li> <li>- Cotation quantitative ou qualitative des risques</li> <li>- Registre des risques avec les mesures d'atténuation associées</li> </ul>

<sup>24</sup> Three lines of defense: <https://www.risk.net/definition/three-lines-of-defence-3lod>

		- Corrélation entre les risques pour effectuer des simulations - Alertes par courriel
Cibles dans l'entreprise	Equipe de gestion des risques	Les dirigeants de l'entreprise
Outils Intégrés	Rapports et Tableaux de bord	Rapports et tableau de bord
Certification	Certifiée selon les normes de sécurité de l'information ISO 27001	Riskconnect a été certifiée par un organisme indépendant (SOC Type2 <sup>25</sup> )
Accessibilité	Accessible via AppexChange	Accessible via AppexChange

### 3.4.3 Objectifs à atteindre pour notre sujet :

A travers notre sujet, nous comptons développer un module de gestion des risques qui doit permettre de créer des risques suivant les types qui existent (risque technique, risque financier ...), évaluer les risques (suivant sa fréquence d'apparition, la gravité et la maîtrise), créer une matrice des risques pour les différents risques d'un projet, créer enfin des tableaux de bords et des rapports qui permettrons de suivre l'évolution des risques durant toute la durée du projet.

#### Conclusion :

Il existe différentes normes de gestion des risques suivant un domaine bien défini. La norme de gestion des risques ISO 31000 2009 ne se limite pas à un domaine bien spécifique et peut s'appliquer à tout type de risque.

Salesforce quant à elle propose des applications de gestion des risques payantes qui se trouvent au niveau de AppexChange.C'est pourquoi l'entreprise propose de développer son propre module de gestion des risques.

<sup>25</sup> SOC Type2: <https://www.evariant.com/faq/what-is-soc-2-type-II-certification>

Ce chapitre nous a permis de faire la comparaison entre les différentes normes de gestion des risques et de définir les paramètres avec lesquels on va quantifier les risques.

## Chapitre 4

### *Méthodologie et Analyse des Besoins*

#### Introduction :

Dans ce chapitre nous parlerons de la méthodologie utilisée dans le cadre du projet mais aussi nous ferons une analyse des besoins et présenterons les différents acteurs de notre application future à travers des diagrammes de cas d'utilisations et de séquence.

---

Chapitre 4 : Méthodologie et analyse des besoins : .....	46
4.1 Méthodologie : .....	46
4.2 Analyse des besoins : .....	47
4.2.1 Identification des besoins non fonctionnels : .....	47
4.2.2 Les différents acteurs du Projet et leurs cas d'utilisation : .....	48
4.2.3 Les diagrammes de séquence : .....	50

---

## 4 Méthodologie et analyse des besoins :

### 4.1 Méthodologie :

Dans le développement traditionnel d'applications il existe différentes méthodes ou cycle de vie qui nous permettent de gérer le développement d'un projet en différentes phases séquentielles jusqu'à la livraison du produit.

Exemple de méthodes traditionnelles méthodes en cascade ou modèle en V<sup>26</sup>.

Il existe aussi d'autres méthodes récentes comme les méthodes agiles.

Salesforce quant à lui propose comme processus de développement Application Life cycle Management. C'est ce processus qui a été utilisé tout le long de notre projet.

ALM<sup>27</sup> définit le processus de gestion du développement d'une application de la conception à la version finale. ALM crée également un cadre permettant de corriger les différents bugs d'application et d'améliorer les fonctionnalités au fil du temps. Nous distinguons six étapes au niveau du processus ALM :

- Planification de la version

On commence tout d'abord à planifier le projet, ce qui consiste à collecter des exigences et de les analyser. On doit aussi déterminer les divers environnements de développement et de tests dont l'équipe a besoins au fur et à mesure que le projet avance dans le cycle ALM.

- Développement

On doit réaliser le travail en respectant les spécifications de conception, et ce, dans un environnement contenant une copie des métadonnées de l'organisation de production, mais dépourvu des données de production.

- Test

On teste les modifications apportées pour vérifier qu'elles fonctionnent comme prévu avant de les intégrer dans l'environnement de production.

- Compilation de la version

On regroupe tous les actifs qu'on a créés ou modifiés au cours de la phase de développement en un seul artefact, un ensemble logique de personnalisation qu'on a déployé en production.

---

<sup>26</sup> <https://cyclededeveloppementdunlogiciel.wordpress.com/le-modele-en-v/>

<sup>27</sup> Application Life cycle Management

➤ Test de la version

On teste dans un environnement de transfert qui produit autant que possible la production de ce qu'on a réellement déployé pour plus de sécurité.

➤ Publication

Lorsque les tests sont terminés et qu'on atteint les critères de qualité, on peut déployer le projet en production.

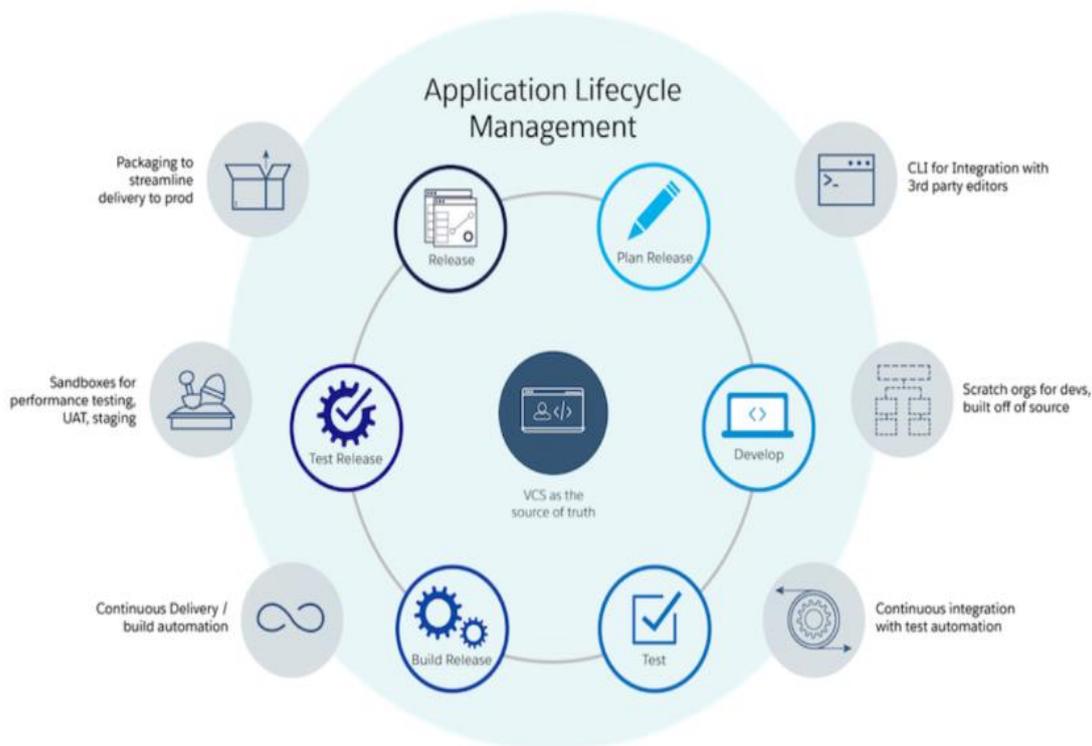


Figure 27 Processus ALM

## 4.2 Analyse des besoins :

### 4.2.1 Identification des besoins non fonctionnels :

- ✚ Contrainte ergonomique : L'application doit avoir une interface graphique simple et facile à utiliser pour permettre une manipulation aisée et compréhensible aux acteurs qui gèrent les risques
- ✚ Contrainte de performance : L'application doit assurer que le temps de réponse à une requête de l'utilisateur soit optimisé rapide et acceptable

- ✚ La sécurité : La solution proposée devra permettre à l'utilisateur une navigation sécurisée. Elle ne doit être accessible qu'après une authentification.

#### 4.2.2 Les différents acteurs du Projet et leurs cas d'utilisation :

L'application Project Management dispose de plusieurs acteurs. Un acteur est un utilisateur qui a toujours le même comportement vis-à-vis de l'application. Dans le cadre de notre module de gestion des risques on a trois acteurs qui sont :

- L'administrateur du système (system administrator) :  
L'administrateur dispose de tous les droits sur le système et sur les objets à manipuler il est en fait le super utilisateur du système.

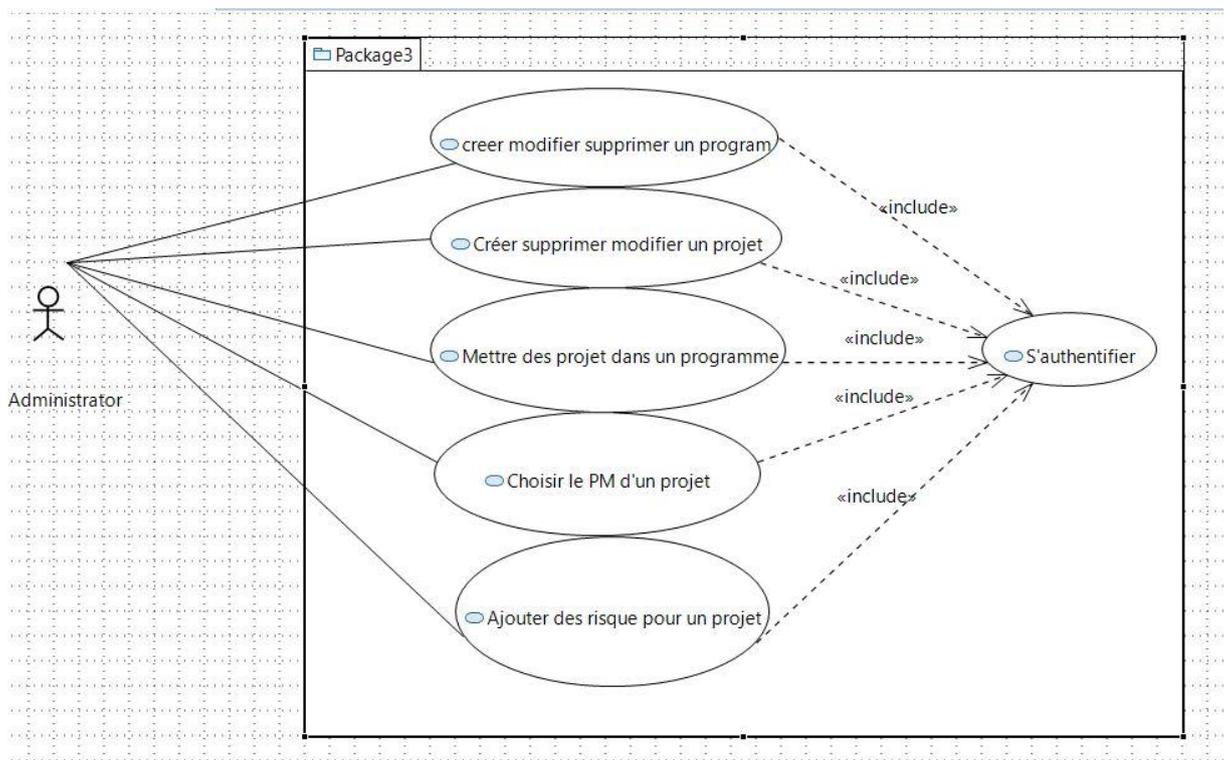


Figure 28 Cas d'utilisation de l'administrateur

- Le PM ou Project Manager :  
Le Project manager est le responsable d'un projet quelconque. Ses droits se limitent à la gestion d'un projet choisit par le l'administrateur même du système.  
Il est PM pendant la durée d'un projet et peut être changé à tout moment suivant le besoin.

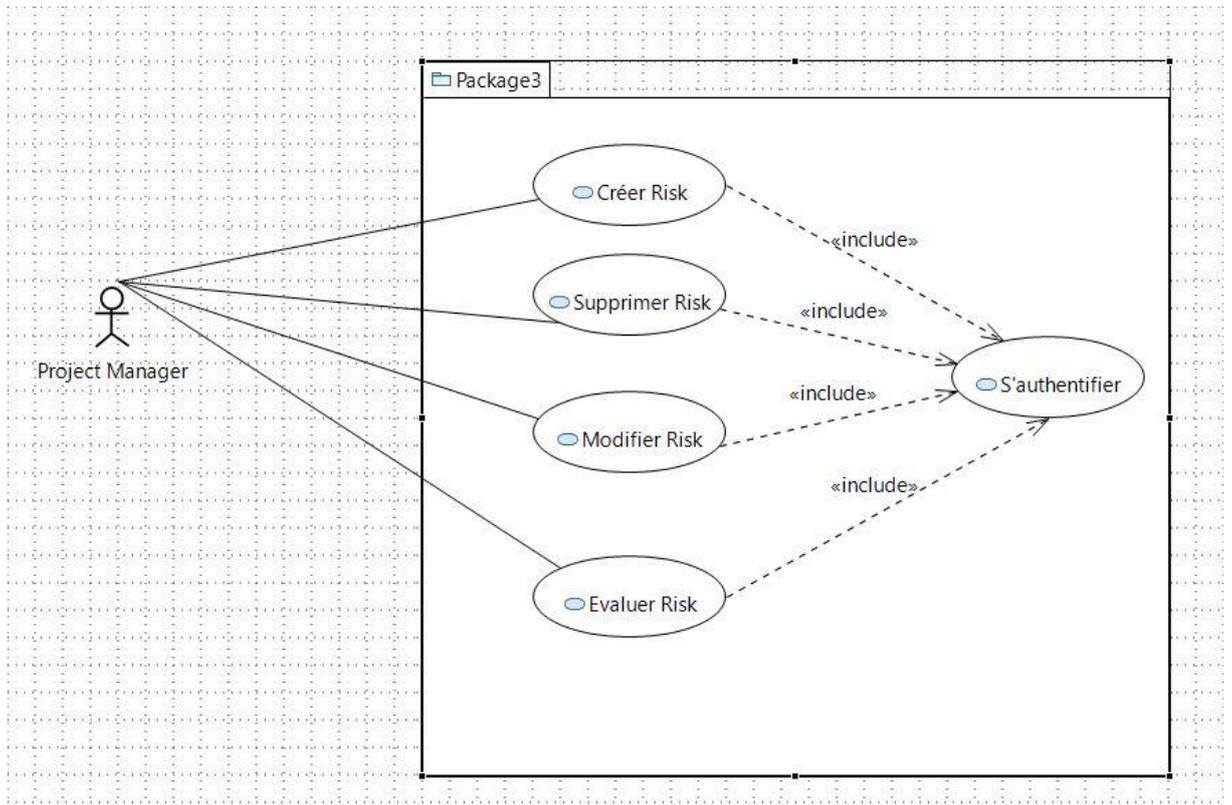


Figure 29 Cas d'utilisation du Project Manager

- Le développeur :  
Le développeur quant a lui a besoin juste d'accéder aux différents objets qu'il manipule.

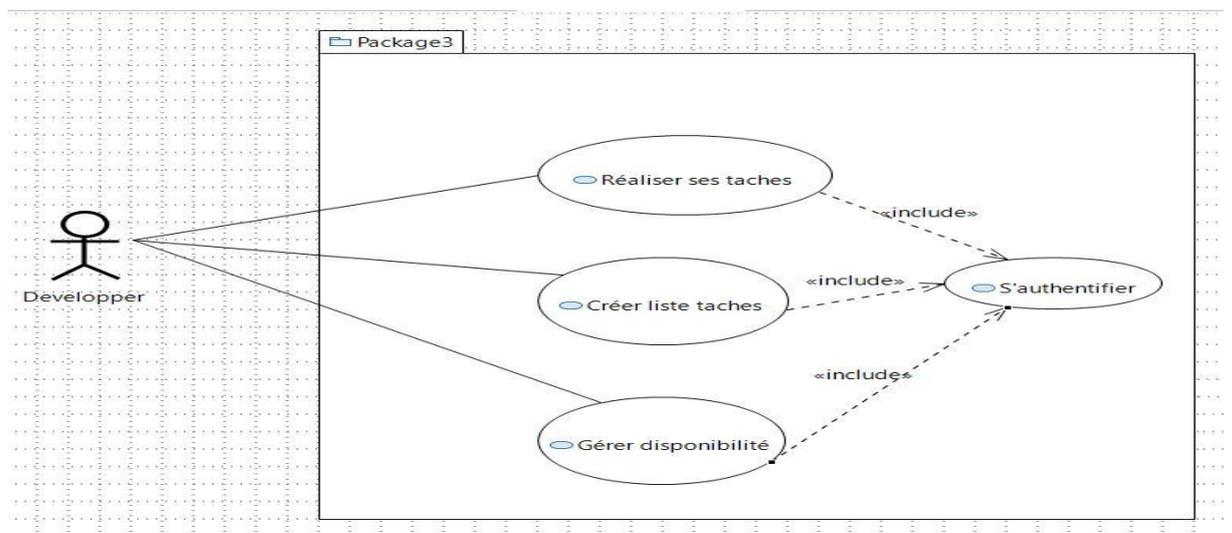


Figure 30 Cas d'utilisation du developpeur

### 4.2.3 Les diagrammes de séquence :

Les différents scénarios qu'on peut avoir seront présentés sous forme de diagramme de séquence.

✚ Authentification :

Authentification	
Acteur(s)	Administrateur, projectManager, développeur
Description	Ce cas permet à un utilisateur de se connecter.
Auteur	Abdoulaye Diallo
Pré-condition(s)	L'utilisateur doit être enregistré dans la base
Postconditions(s)	On se trouve à la page d'accueil de connexion
Démarrage	
Scénario nominal	
<b>1.</b> Le système retourne la page de connexion de l'application.	
Scénario alternatif	
<b>1.</b> Le système retourne à la page de connexion	

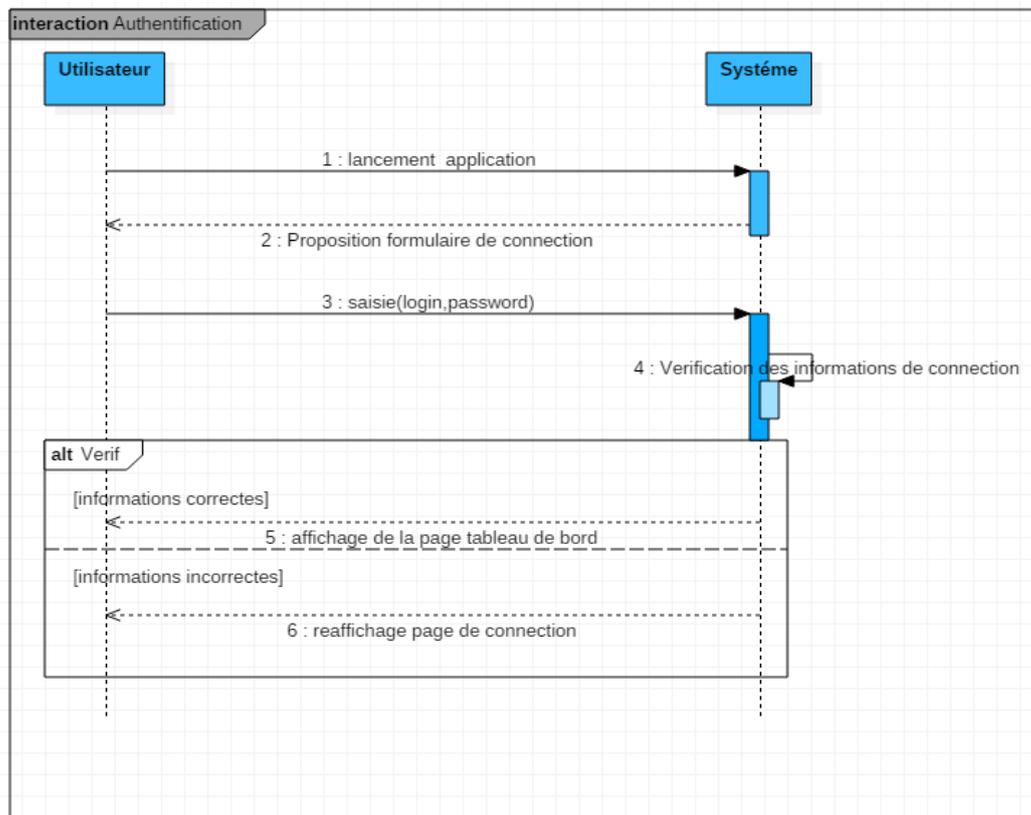


Figure 31 Authentification des Utilisateurs

- Diagramme de séquence de création d'un risque :

Création d'un risque	
Acteur(s)	Administrateur ou Project Manager
Description	Ce cas permet à l'administrateur et au PM de créer un nouveau risque
Auteur	Abdoulaye Diallo
Préconditions	Être Authentifié

Post-condition(s)	
Démarrage	Un risque doit être ajouté dans la base
Scénario nominal	
<ol style="list-style-type: none"> <li>1) Le système affiche le formulaire de saisie</li> <li>2) L'utilisateur saisit les informations pour le risque</li> <li>3) L'utilisateur valide la saisie</li> <li>4) Le système vérifie si les informations entrées sont correctes</li> <li>5) Le système demande une confirmation</li> <li>6) L'utilisateur confirme</li> </ol>	
Scénario alternatif	
<ol style="list-style-type: none"> <li>1) L'utilisateur annule la saisie.</li> <li>2) L'utilisateur ne valide pas le formulaire de saisie</li> <li>3) L'utilisateur ferme l'application</li> </ol>	
Scénario D'exception	
<ol style="list-style-type: none"> <li>1. Les informations saisies ne sont pas correctes</li> <li>2. Espace insuffisant pour enregistrer la saisie</li> </ol>	

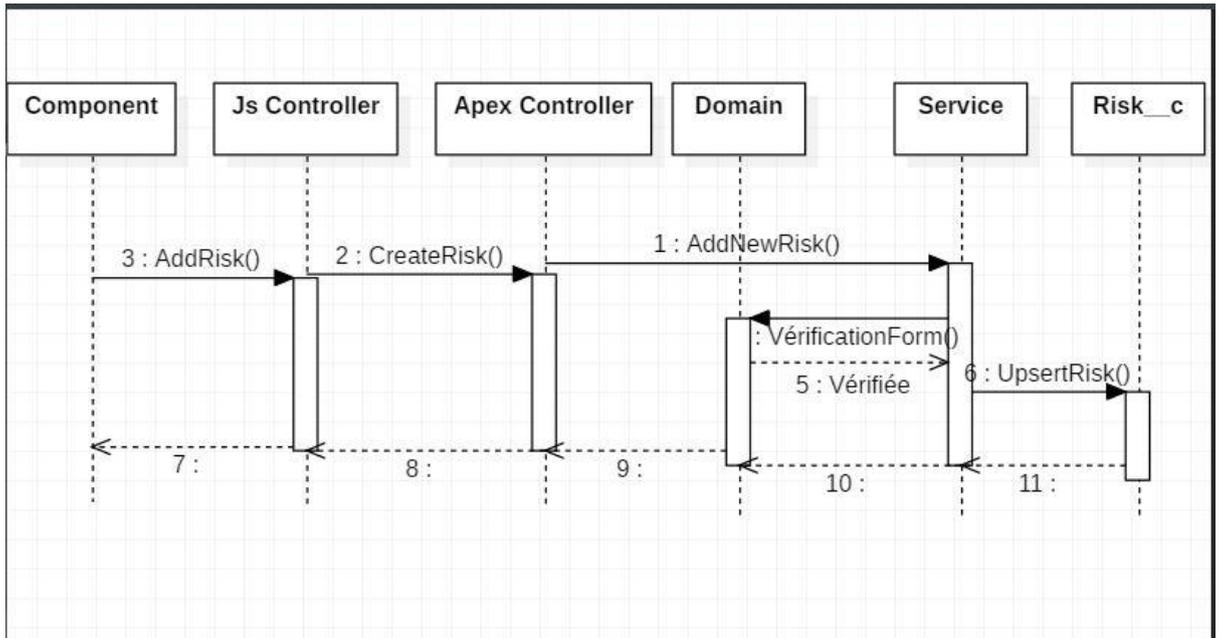


Figure 32 Création d'un Risque

- Diagramme d'affichage des détails d'un risque :

Affichage détails d'un Risque	
Acteur(s)	Administrateur ou Project Manager
Description	Ce cas permet à l'administrateur et au PM de visualiser les détails d'un risque
Auteur	Abdoulaye Diallo
Préconditions	Être Authentifié
Post-condition(s)	
Démarrage	Les détails d'un risque doivent être visualisés.

Scénario nominal
<p>1) L'utilisateur demande la liste de ses risques.</p> <p>2) Le système affiche la liste de tous les risques</p> <p>3) L'utilisateur sélectionne un risque</p> <p>4) Le système vérifie si l'ID du risque existe</p>
Scénario alternatif
<p>1) L'utilisateur fait un retour au niveau de la liste.</p> <p>2) L'utilisateur ferme l'application</p>
Scénario D'exception
<p>1) Les informations du risque sont vides</p>

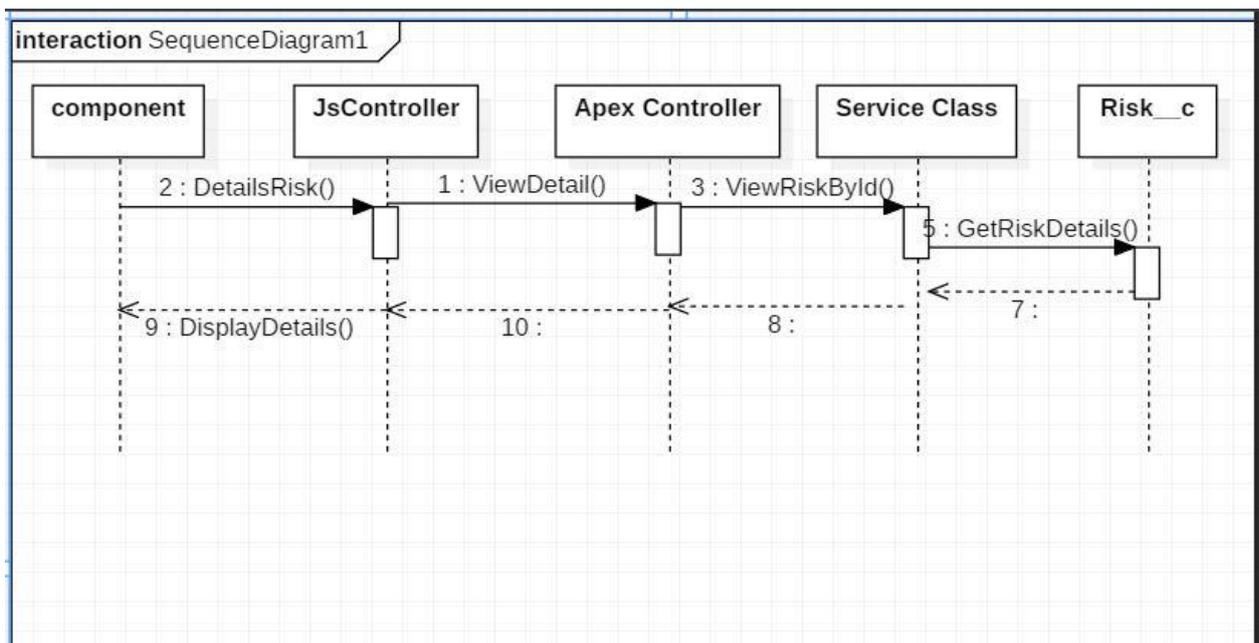


Figure 33 Affichage détails risque

- Diagramme de séquence de modification d'un risque :

Modification d'un risque	
Acteur(s)	Administrateur ou project Manager
Description	Ce cas permet à l'utilisateur de modifier un risque
Auteur	Abdoulaye Diallo
Pré-Conditions	Etre Authentifié
Post-condition(s)	
Démarrage	Un risque doit être modifié
Scénario nominal	
<ol style="list-style-type: none"> <li>1. Le système affiche le formulaire de modification</li> <li>2. L'utilisateur change les informations relatives au risque</li> <li>3. L'utilisateur valide la saisie</li> <li>4. Le système vérifie si les informations entrées sont correctes</li> <li>5. Le système demande une confirmation</li> <li>6. L'utilisateur confirme</li> </ol>	
Scénario alternatif	
<ol style="list-style-type: none"> <li>1. L'utilisateur annule la saisie.</li> <li>2. L'utilisateur ne valide pas le formulaire de sais</li> <li>3. L'administrateur ferme l'application</li> </ol>	
Scénario D'exception	

1. Les informations saisies ne sont pas correctes
2. Le risque ne peut pas être modifié

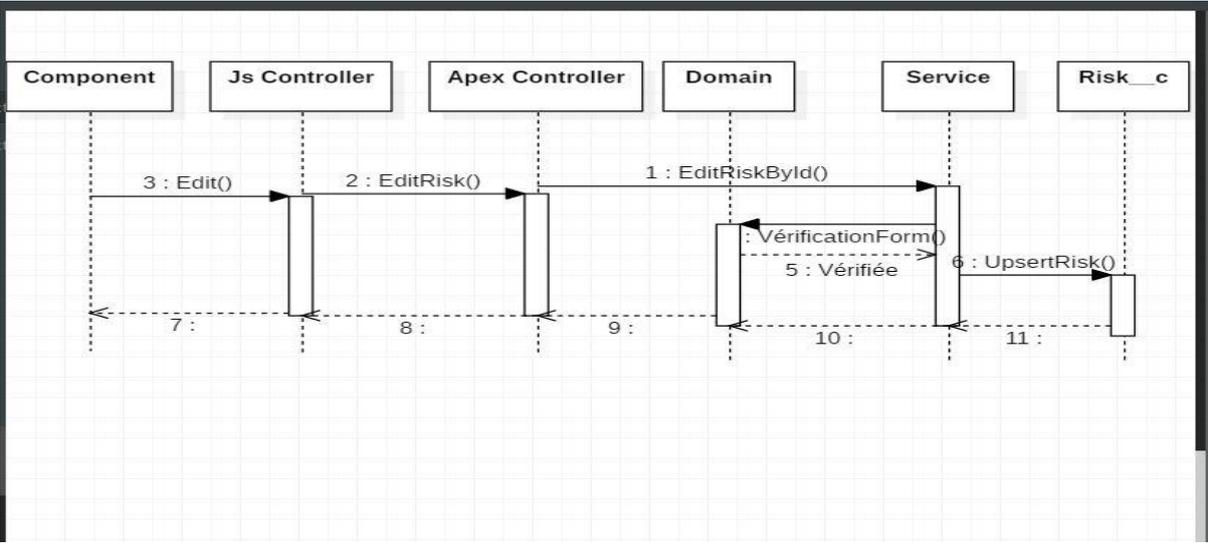
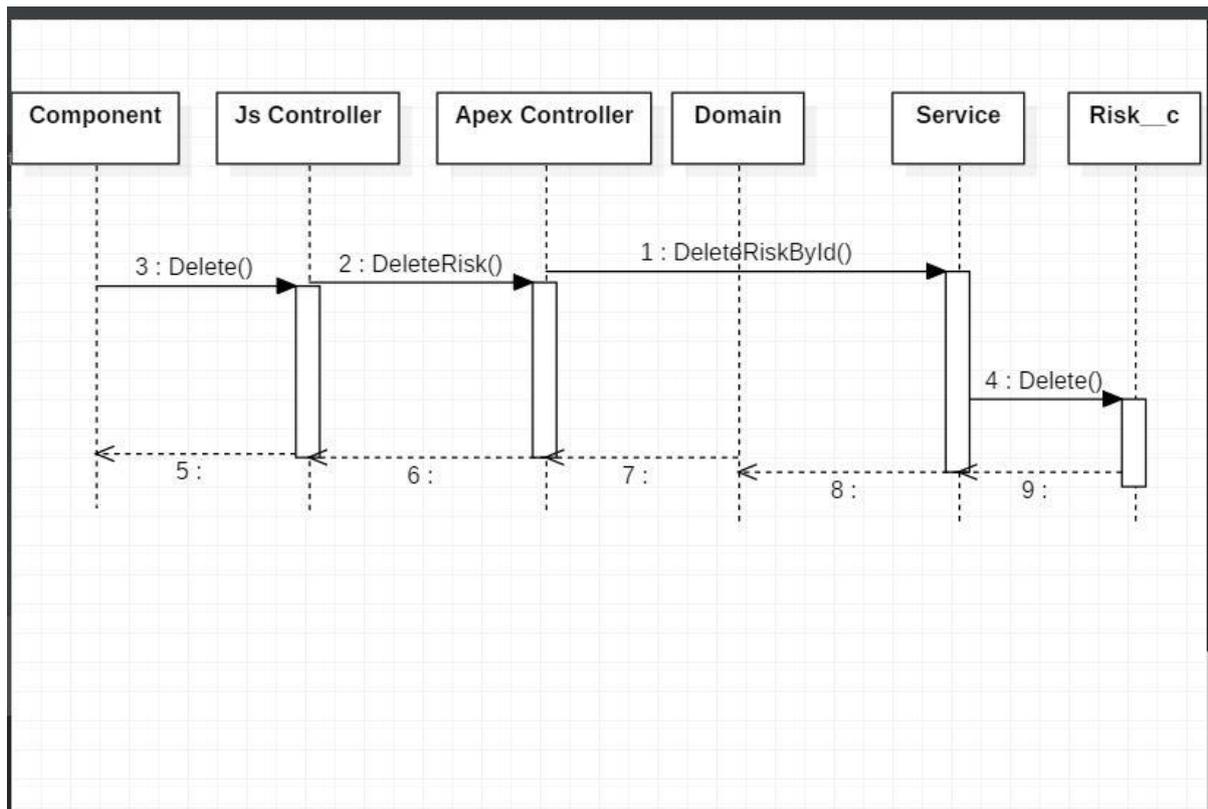


Figure 34 Modification d'un risque

- Diagramme de suppression d'un risque :

Suppression d'un risque	
Acteur(s)	Administrateur ou Project Manager
Description	Ce cas permet à utilisateur de supprimer un risque
Auteur	Abdoulaye Diallo
Pré_Condition	Être Authentifié
Post-condition(s)	

Démarrage	Un risque doit être supprimé
Scénario nominal	
<ol style="list-style-type: none"> <li>1. Le système affiche la liste des risques</li> <li>2. L'utilisateur sélectionne le risque à supprimer</li> <li>3. Il appuie sur le bouton supprimer de la liste</li> <li>4. Le système demande une confirmation</li> <li>5. Il confirme la suppression</li> <li>6. Le système libère de l'espace</li> </ol>	
Scénario alternatif	
<ol style="list-style-type: none"> <li>1. L'utilisateur souhaite annuler la suppression</li> <li>2. L'utilisateur ne confirme pas la validation de la suppression</li> </ol>	
Scénario D'exception	
Le risque ne peut pas être supprimée pour le moment (elle est soumise à une approbation ou elle est déjà validée).	



*Figure 35 Suppression d'un Risque*

**Conclusion :**

Notre module comprend différents acteurs qui interagissent avec le système. Chaque acteur a un rôle bien défini dans le système. La suppression, l'ajout ou la modification se font suivant des séquences bien définies avec des validations à chaque étape. Ce chapitre nous a permis de définir les différents acteurs et leurs cas d'utilisation. Ceci va nous permettre de comprendre les rôles des différents acteurs dans la présentation de la solution.

## Chapitre 5

### *Conception et Proposition d'une Solution*

#### Introduction :

Dans ce chapitre nous parlerons de la conception de notre application ainsi que la présentation de la solution à travers les différentes interfaces obtenues.

---

Chapitre 5 : Conception et Proposition d'une Solution : .....	60
5.1 Conception : .....	60
5.1.1 L'Architecture du projet : .....	60
5.1.2 La modélisation : .....	62
5.1.3 Le design du projet : .....	68
5.1.4 Gestion des profils : .....	69
5.1.5 Le langage SOQL .....	70
5.1.6 La notion de governors limit.....	70
5.1.7 Les tests.....	71
5.1.8 La gestion de la sécurité dans le projet : .....	71
5.2 Présentation de la solution : .....	72
Conclusion générale et perspectives : .....	81

---

## 5 Conception et Proposition d'une Solution :

### 5.1 Conception :

#### 5.1.1 L'Architecture du projet :

L'architecture utilisée pour le développement du projet est l'architecture SOC<sup>28</sup>. Le choix de cette architecture repose sur le fait qu'elle est évolutive et permet de segmenter le code en différentes parties.

##### 5.1.1.1 L'architecture SOC en relation avec notre projet :

L'architecture Separation Of Concern est la séparation des diverses préoccupations en différents couches afin de permettre la simplification du code à parcourir et à entretenir. Lorsque des modifications sont réalisées, les impacts et les régressions sur d'autres zones sont réduits et des programmes peuvent évoluer facilement. Notre projet est basé sur cette architecture. Ainsi l'architecture de notre projet présente différentes couches.

##### 5.1.1.2 Les différentes couches utilisées :

Notre projet a été développée en utilisant l'architecture en couches.

Différentes couches ont été utilisées (Domain, Service, Selector). Chaque couche a un rôle spécifique et pourra faire appel à une autre en cas de besoin.

###### La couche Domaine :

La couche domaine fournit un niveau plus précis d'encapsulation et de réutilisation de code au sein de notre application telle que la validation complexe, le retour aux valeurs par défaut ainsi que d'autres logiques liées au calcul et à la manipulation complexe.

C'est au niveau de cette couche que la vérification de la sémantique doit se faire. Après la vérification la couche domaine fait appel à la couche service pour une autre opération.

###### La couche Service :

La couche service quant à elle est un moyen d'encapsuler les processus métier de notre application.

Exemple de processus métier : Click d'un bouton Save pour la création d'un risque.

---

<sup>28</sup> Separation of concern: [https://en.wikipedia.org/wiki/Separation\\_of\\_concerns](https://en.wikipedia.org/wiki/Separation_of_concerns)

La couche service nous permet de réutiliser plusieurs fois le code relatif à un objet quelconque. La couche service en cas de besoin peut faire aussi appel à la couche Selector. Exemple quand on clique sur le bouton modifier la classe service fait appel à la classe selector pour aller chercher les informations relatives à l'objet à modifier au niveau de la base.

✚ La couche Selector :

Elle contient la logique chargée de demander des informations à nos objets personnalisés. Elle peut être utilisée par n'importe quelle autre classe (domaine, service contrôleur apex ...).

Au niveau de la couche Selector on y stocke tout ce qu'il y'a comme code qui permet de faire des requêtes de sélection au niveau de la base.

Exemple : Récupérer un risque à partir de son identifiant(Id).

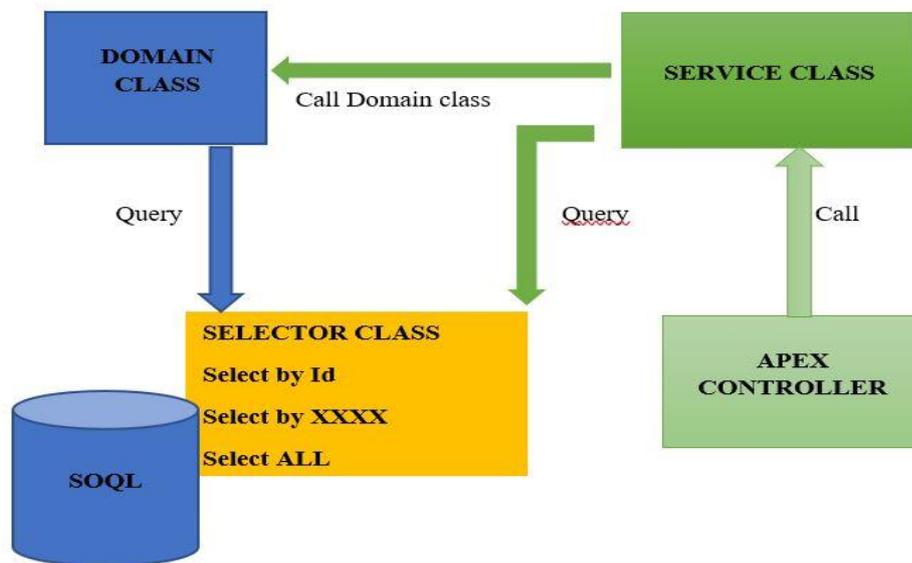


Figure 36 Les différentes Couches

### 5.1.1.3 Liste des différentes classes domaine et service :

Dans notre projet nous aurons une classe domaine et une classe service avec pour chaque classe des méthodes et des rôles spécifiques.

#### ❖ Domain Risk :

On a une classe domaine appelée Risk Domain ou on va vérifier la sémantique au niveau de notre formulaire. C'est au niveau de la classe Domain tout ce qui concerne une validation finale des données présentes dans le formulaire. Dans cette classe domain on a comme exemple les méthodes : CheckForm() ;

#### ❖ Service Risk :

La classe service doit nous permettre après la validation par le Domain d'insérer le nouveau risque au niveau de notre base. C'est à ce niveau ou on implémentera les requêtes qui nous permettrons de l'insérer au niveau de la base.

- ✓ SaveRisk() ;
- ✓ EditRisk() ;
- ✓ DeleteRisk() ;

#### ❖ Domain Project :

Au niveau de la classe Domain on va vérifier les données en entrée pour savoir le projet auquel on veut associer les risques n'a pas atteint sa date limite mais aussi on va vérifier si le risque qu'on veut associer à ce projet est valide.

#### ❖ Service Project :

La classe service de notre application va gérer tout ce qui est interaction avec la base de données. C'est au niveau de la classe domaine qu'on va implémenter les méthodes qui vont nous permettre de mettre à jour d'ajouter ou de supprimer un risque.

- SaveProject() ;
- EditProject() ;
- DeleteProject() ;

### **5.1.2 La modélisation :**

#### **5.1.2.1.1 La modélisation du projet sous Salesforce :**

Sous Salesforce tout est métadonnée. Un objet crée est un fichier XML.

Dans la modélisation traditionnelle on utilise souvent des outils pour créer nos différentes classes et aussi créer des relations entre les différentes classes.

Cependant sous Salesforce il existe des objets créés au préalable ou des objets qu'on peut créer selon le besoin. Sous Salesforce on a deux types d'objets : les objets standards et les objets personnalisés. C'est sur ces objets qu'on se base pour pouvoir modéliser.

✓ Les objets standards :

Les objets standards (standards objects) sont les objets inclus avec Salesforce (on les appelle aussi les objets métiers).

Exemple : Account, Contact, Opportunity ,etc.

Au niveau de ces objets on peut ajouter de nouveaux champs en cas de besoin.

✓ Les objets personnalisés :

Un objet personnalisé (Custom object) est un objet qui n'est pas inclus dans Salesforce. C'est un objet qu'on crée et qui est spécifique à un secteur d'activité. Comme par exemple on crée l'objet Risk pour stocker un risque.

#### **5.1.2.1.2 Processus de modélisation :**

Dans le cadre de notre projet différentes étapes ont été utilisées pour partir du modèle de base jusqu'à la stabilisation d'un modèle.

Les différentes étapes de la modélisation sous Salesforce :

- ✓ Présentation du modèle de base : Notre modèle de base comprend plusieurs classes

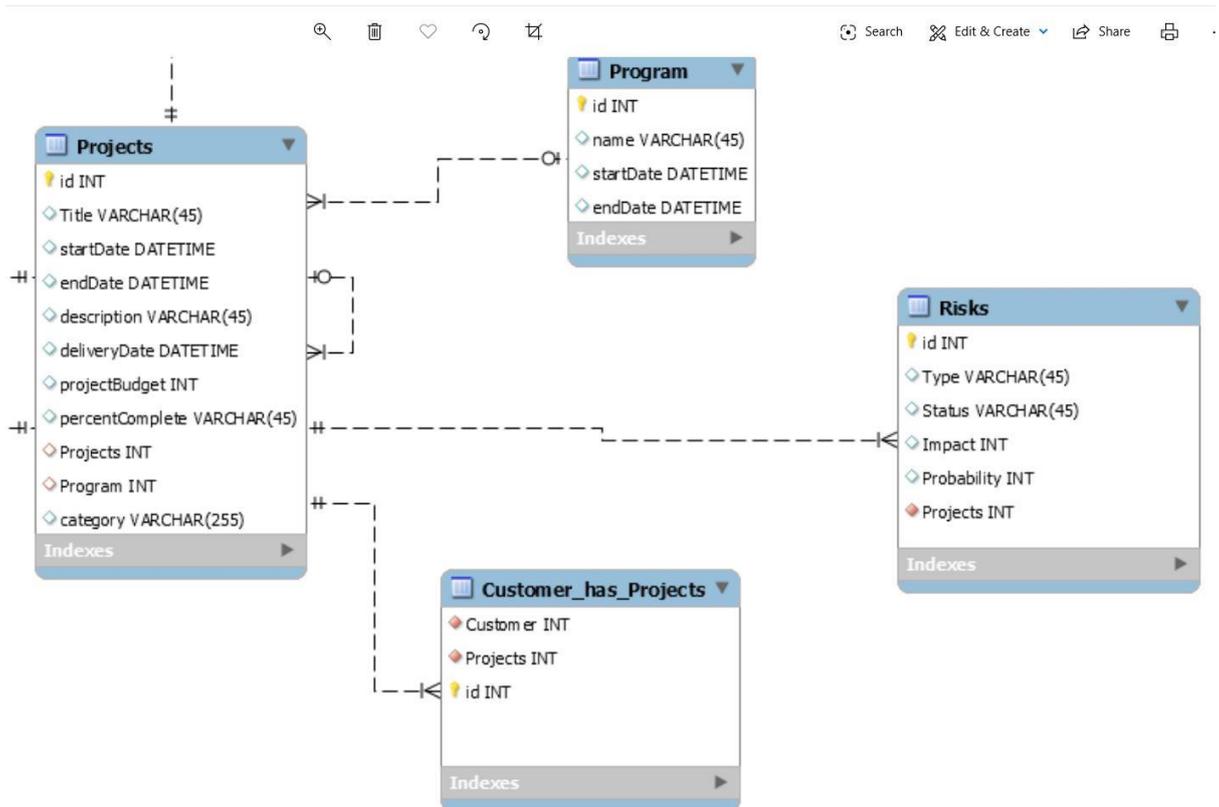


Figure 37 Diagramme de Classes

- ✓ Correspondance entre les objets Salesforce et les classes du modèle de base :  
 Dans cette étape on fait une comparaison entre une classe du modèle de base et un objet Salesforce. On vérifie s'il existe une correspondance des types de champs entre la classe et l'objet Salesforce. Si cette correspondance est supérieure ou égale 75% cet objet Salesforce peut être utilisé pour représenter la classe de notre modèle de base. Pour les champs manquants on complète par des champs personnalisés.

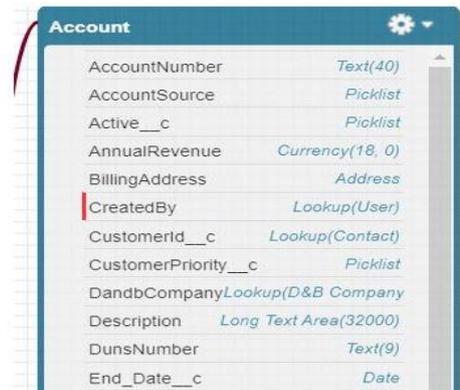
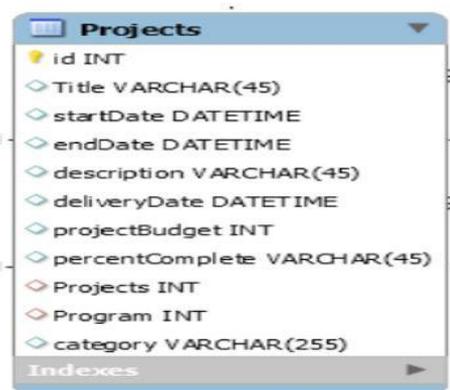


Figure 38 Correspondance entre un Objet et une Classe

**Create Date/Time Field (Object: Account)**

Field Label: Start Date

Field Name: Start\_Date

Description: Date de début du projet

Help Text:   
This text displays on detail and edit pages when users hover over the Info icon next to this field.

Default Value:

Required:  Always require a value in this field in order to save a record

Save Cancel

Figure 39 Création d'un Custom Label

- ✓ Vérification des différentes relations entre les objets :

La deuxième étape consiste à recenser les autres Subjects<sup>29</sup> qui sont liés au Subject choisi premièrement. Le choix se fera suivant deux critères. D'abord on regarde le type de relation qui existe entre l'objet choisit et les autres auxquels il est lié. Ensuite on regarde les types des champs de ces Objets.

Si le premier critère est satisfait on utilise cet objet et on crée la relation entre ces deux objets. Si les deux critères sont satisfaits cet objet est aussi choisi pour représenter une autre classe du modèle de base.

<sup>29</sup> [https://trailhead.salesforce.com/fr/content/learn/modules/apex\\_database/apex\\_database\\_subjects](https://trailhead.salesforce.com/fr/content/learn/modules/apex_database/apex_database_subjects)

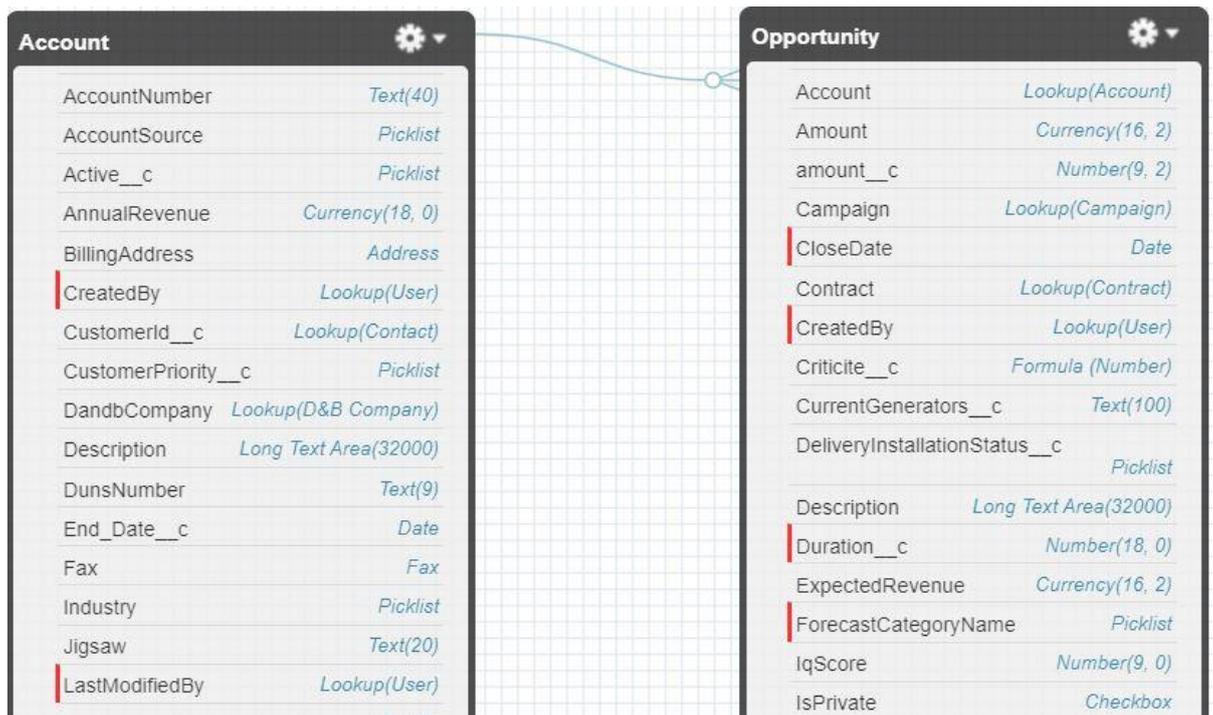


Figure 40 Relation entre deux Objets Salesforce

- ✓ Si les deux premières étapes ne sont pas satisfaites on crée un champ personnalisé.

Figure 41 Création d'un Custom Object

Quand un objet Salesforce correspond à deux classes du modèle de base on crée un record type pour différencier les différentes classes. Ceci permet d'utiliser un même objet pour deux classes.

**Step 1. Enter the details**

Enter a name and description for the new record type. The new record type will include all the picklist values from the existing record type, you will be able to customize the picklist values.

**Record Type**

Existing Record Type: --Master--

Record Type Label: Invoice

Record Type Name: Invoice i

Description:

Active:

**Step 2. Assign page layouts**

Group Record Type	Invoice
Record Type Name	Invoice
Description	

Select the page layout that users with this profile see for records with this record type. After saving, click the **Page Layouts** link to view the page layouts.

Apply one layout to all profiles Group Layout

Apply a different layout for each profile

Profile:	Page Layout
Analytics Cloud Integration User	Group Layout
Analytics Cloud Security User	Group Layout
Authenticated Website	Group Layout
Authenticated Website	Group Layout

*Figure 42 Création d'un Record Type*

Ces différentes étapes ont permis d'arriver au modèle suivant pour notre module :

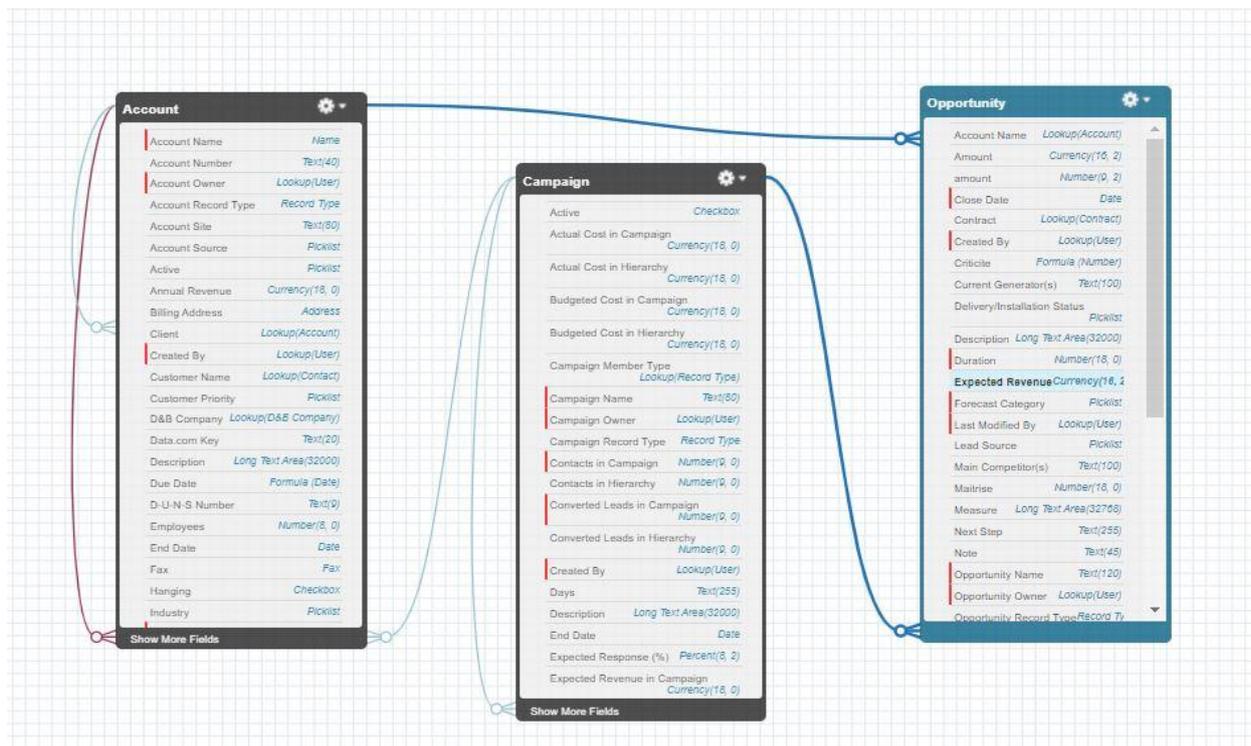


Figure 43 Modèle Final

### 5.1.3 Le design du projet :

Dans le cadre de la réalisation de notre projet on a utilisé l'outil SLDS<sup>30</sup>.

Le système SLDS<sup>31</sup> nous aide à créer des applications avec l'aspect et la convivialité de Lightning Expérience sans avoir à écrire une seule ligne CSS. SLDS est un Framework CSS qui nous donne accès aux icônes, palettes de couleurs et polices que nos développeurs utilisent pour créer Lightning Expérience.

SLDS est un ensemble de modèles de conception reproductibles et de bout de code réutilisables, nommés composants. Les développeurs associent ses composants de différentes manières pour créer des applications avec du code JavaScript. Ses caractéristiques sont :

- **Evolutivité** : Les développeurs utilisent les éléments fondamentaux dans un grand nombre de produits et de fonctionnalités.
- **Efficacité** : Les développeurs se concentrent sur l'ergonomie et l'utilité.
- **Cohésion visuelle** : permet de créer des produits et fonctionnalités visuellement cohérents pour une expérience utilisateur homogène.

<sup>30</sup> Salesforce Lightning Design System

- Partage : permet aux développeurs de récupérer les parties d'un projet et de s'appuyer sur le travail de leurs collègues.

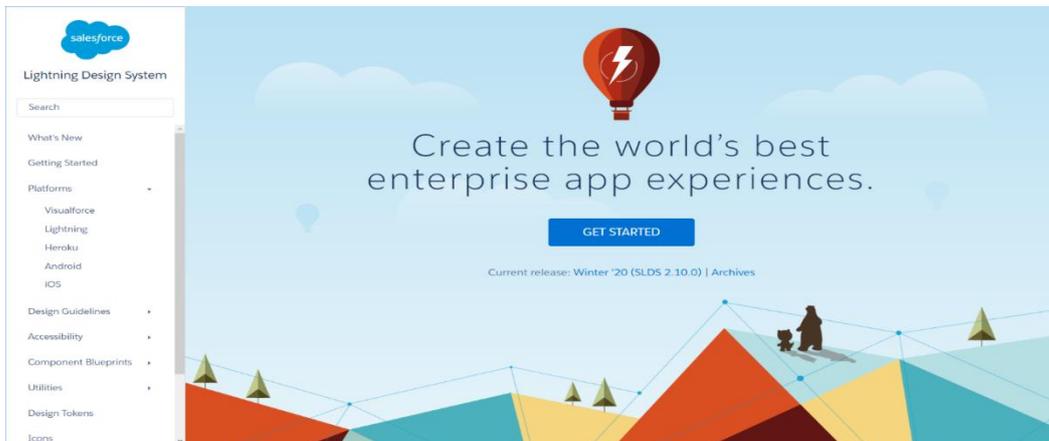


Figure 44 SLDS

SLDS aide Salesforce à concevoir une expérience produit cohérente de manière efficace et à grande échelle.

Exemple de code SLDS prêt à être utilisé :

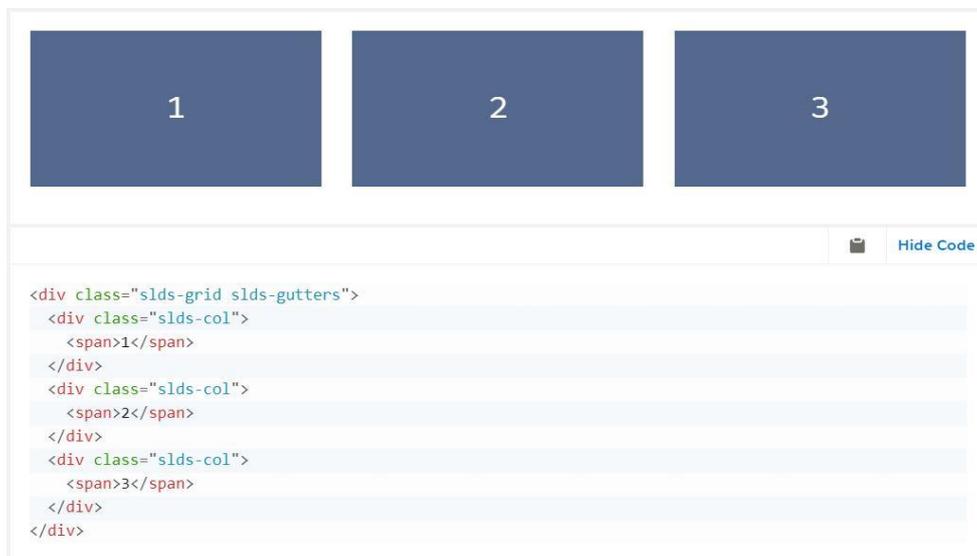


Figure 45 Exemple de Code SLDS

#### 5.1.4 Gestion des profils :

Sous Salesforce il existe plusieurs profils définis par défaut.

Les profils donnent des accès de lecture, d'écriture et de modification sur les objets existants.

Dans le cadre du développement de notre module, les profils de Salesforce ont été utilisés pour garantir un niveau d'utilisation et d'accès à l'application selon les différents acteurs.

Ces différents profils permettent de limiter les opérations d'exécutions des utilisateurs dans l'application.

Dans le cadre de notre travail des profils différents ont été utilisés :

1	Users	Profiles
2	Administrator	System administrator
3	Project manager	Standard User
4	Developer	Contract Manager

Figure 46 Liste des Profiles

### 5.1.5 Le langage SOQL

Pour lire un enregistrement dans Salesforce on a besoin d'un langage de requête.

Le langage SOQL<sup>32</sup> est celui par défaut utilisé par Salesforce. Il permet de rechercher des informations spécifiques dans les données Salesforce.

Le langage SOQL nous permet de lire les informations sauvegardées. Il est très similaire au langage SQL. On peut l'incorporer directement dans le code Apex.

### 5.1.6 La notion de governors limit

Comme Apex est exécuté dans un domaine mutualisé, le moteur d'exécution de Apex applique une limitation stricte pour empêcher qu'un emballement de code Apex ne monopolise des ressources partagées. Ces limitations, ou gouverneurs, surveillent et applique les statistiques présentées dans le tableau suivant. Si un code Apex dépasse une limite, le gouverneur associé génère une exception à l'exécution qui ne peut pas être gérée.

Exemples de governor limit :

---

<sup>32</sup> Salesforce Object Query Language

Description	Synchronous Limit	Asynchronous Limit
Total number of SOQL queries issued <sup>1</sup>	100	200
Total number of records retrieved by SOQL queries	50,000	
Total number of records retrieved by Database.getQueryLocator	10,000	
Total number of SOSL queries issued	20	
Total number of records retrieved by a single SOSL query	2,000	
Total number of DML statements issued <sup>2</sup>	150	

*Figure 47 Les governors Limits*

### 5.1.7 Les tests

Les méthodes des tests sont écrites en Apex et annotées avec le mot clé `testMethod`. Il est important de comprendre que les tests unitaires sont un prérequis obligatoire au déploiement de code dans un environnement de production. Ils sont aussi obligatoires si le code doit être packagé pour être mis à disposition de la communauté Salesforce via l'AppExchange.

Avant le déploiement dans AppExchange les méthodes de test doivent couvrir au minimum 75% de la totalité du code qui doit être déployé. Donc on peut considérer les méthodes de test comme <<du code apex qui teste du code apex>>.

### 5.1.8 La gestion de la sécurité dans le projet :

Dans le cloud l'aspect sécurité est un paramètre primordial. Dans le cadre de notre projet nous avons utilisé la gestion de la sécurité de Salesforce.

Dans Salesforce on peut contrôler les utilisateurs qui ont accès aux données de l'organisation, à un objet spécifique, à un champ ou à un enregistrement individuel.

- Organisation : on peut gérer la liste des utilisateurs autorisés, définir des stratégies de mot de passe et limiter les connexions à certains horaires et emplacements.
- Objet : l'accès aux données de l'objet est l'aspect le plus facile à contrôler. En définissant des autorisations pour un type d'objet particulier, il est possible d'empêcher un groupe d'utilisateurs de créer, voir, modifier ou supprimer des enregistrements de cet objet.

- Champs : on peut limiter l'accès à certains champs, même si un utilisateur a accès à l'objet.
- Enregistrement : on peut autoriser des utilisateurs spécifiques à contrôler un objet, mais limiter l'accès aux enregistrements d'objet qu'ils consultent.

## 5.2 Présentation de la solution :

Dans cette partie nous allons présenter les différentes interfaces réalisées au niveau de notre application. Mais aussi les différentes contraintes rencontrées lors du développement sous Salesforce.

Notre module comporte deux grandes parties :

La création d'un programme et d'un projet :

La création et la gestion des risques :

On accède d'abord à la page d'accueil qui est par défaut la page de connexion de Salesforce.

Cette page n'est accessible que pour l'administrateur.



*Figure 48* Page d'authentification

Après connexion l'administrateur a la possibilité de créer un projet et d'un programme en même temps. En créant aussi un projet on lui associe un project manager qui fera office de

gestionnaire des risques. Précisons aussi qu'un projet appartient à un seul programme et qu'un programme peut contenir beaucoup de projets.

ADD NEW PROJECT

\* Name  
name

\* Category  
Select a type of the project

\* Start Date

\* Delivery Date

\* Project Budget

Percent Complete  
Percent

Program  
Select a project

Project parent  
Select a parent project

\* Project Manager  
Select a project manager

\* Project Status  
Select a status of the project

Customer  
Select a customer

Salesforce Sans 12 B I U

Type something interesting

Cancel Create

Figure 49 Formulaire de Création d'un Projet et d'un Program

Après la création d'un projet on le met directement au niveau de la liste des projets ou on peut avoir les options de modifications et de suppression.

Au niveau de cette page l'administrateur (le seul à y avoir accès) peut voir la liste de tous les projets de l'entreprise et leur budgets relatifs. Au niveau de cette liste on peut savoir les projets en cours les projets stoppés et les projets qui n'ont pas encore débuté.

NAME	CATEGORY	BUDGET
Compass	event projects	100000000
MabySambaProject	IP	20000000
Project Management	IT projects	5500000000
Project Sales Cloud	IT projects	5000000000
Ten	IT projects	1000000000
Sales Cloude	IT projects	2200000

Figure 50 List des Projets

On peut aussi visualiser les différents détails relatifs à un projet en cliquant sur détails .

MEMBER	ROLE	START DATE	END DATE	STATUS
Maby Gaye	Project Management	2020-01-16	2020-01-17	Active
Dieynaba Diallo	Developer	2020-02-08	2020-02-29	Active
Youssef Diedhiou	Developer	2020-02-11	2020-02-26	Inactive
Abdoulaye Diallo	Team Leader	2020-02-21	2020-02-28	Active
Arona Ndiaye	Team Leader	2020-02-13	2020-02-21	Inactive
Samba Wade	Freelance	2020-02-13	2020-02-13	Inactive

TITLE	IMPACT	TYPE	STATUS	CRITICITY
unknow risk 2	98	Conflict risk	Closed	8
unknown risk	4	Financial risk	Closed	8
risk de coupure de courant	600	Financial risk	In progress	2
risk de manque de connexion	89000	Financial risk	Closed	12

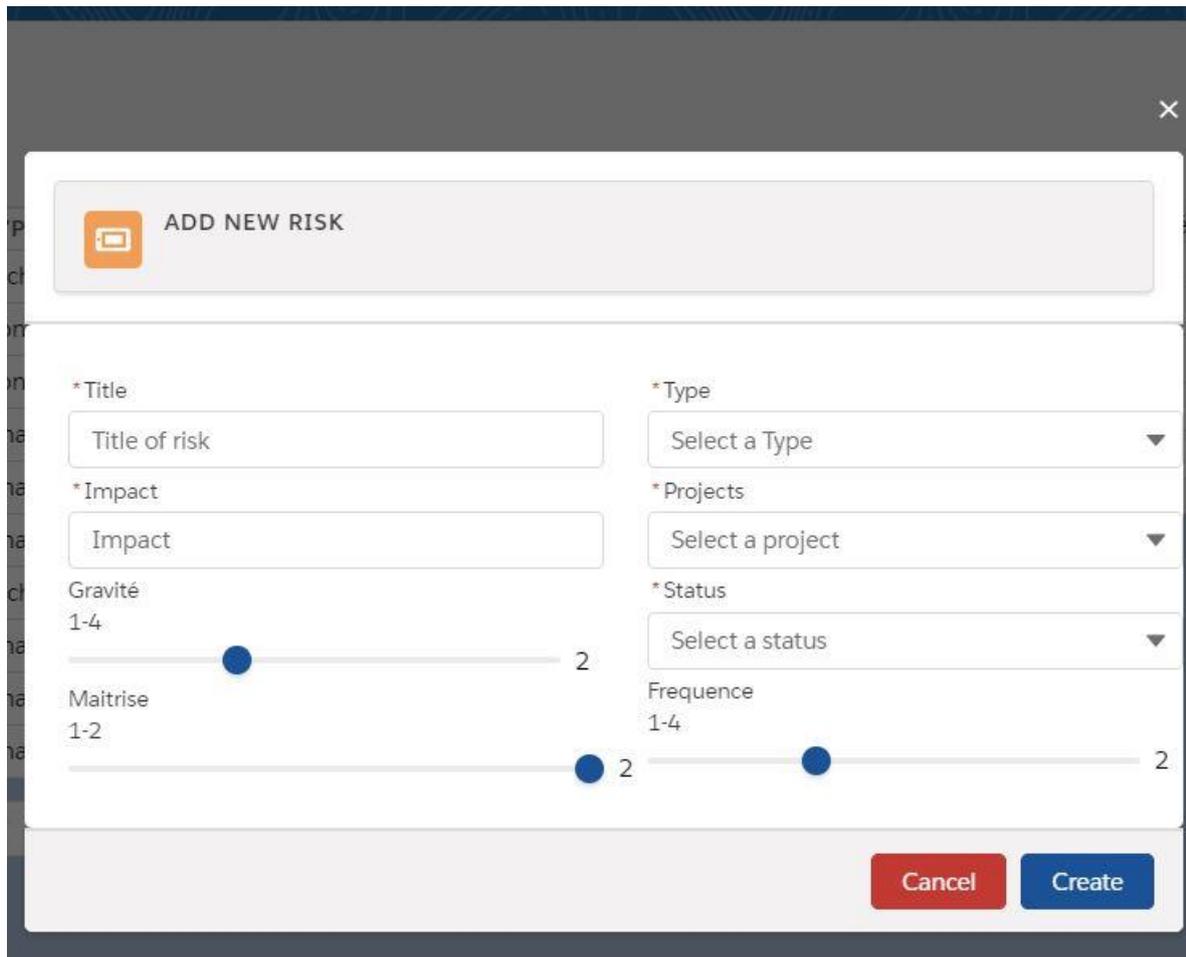
Fréquence	1	2	3	4
4				
3				
2				
1				
	1	2	3	4

Le project manager une fois choisi dispose de la même interface de connexion que l'administrateur

Après la création d'un projet et le choix du PM<sup>33</sup> on a les différentes interfaces pour la gestion des risques.

<sup>33</sup> Project Manager

La première page accessible pour le PM qui fait office de Risk Manager est la page de création d'un risque.



The image shows a software interface for creating a new risk. At the top, there is a header bar with an orange icon and the text "ADD NEW RISK". Below this, the form is organized into several sections:

- \*Title:** A text input field containing "Title of risk".
- \*Impact:** A text input field containing "Impact".
- Gravité:** A slider control with a range from 1 to 4. The current value is set to 2.
- Maitrise:** A slider control with a range from 1 to 2. The current value is set to 2.
- \*Type:** A dropdown menu with the text "Select a Type".
- \*Projects:** A dropdown menu with the text "Select a project".
- \*Status:** A dropdown menu with the text "Select a status".
- Frequence:** A slider control with a range from 1 to 4. The current value is set to 2.

At the bottom right of the form, there are two buttons: a red "Cancel" button and a blue "Create" button.

*Figure 51 Formulaire de création d'un risque*

Après la création d'un risque relatif à un projet on a accès à la liste des différents risques d'un projet avec un nouveau champ criticité qui est calculé automatiquement quand on crée un risque. C'est ce champ même qui va permettre de savoir la nature même du risque c'est-à-dire savoir s'il est nuisible ou non au projet.

Risks  
List of Risks  
10 items · Updated 13 minutes ago

TITLE	IMPACT	TYPE	PROJECT	PROJECT STATUS	CRITICITY DEGREE
taf	222	Financial risk	Compass	Closed	8
risk diop	63000	Human Risk	Sales Cloude	Opened	2
risk de changement de serveur	8900	Financial risk	MabySambaProject	Closed	14
risque de conflit avec le client	95000	Financial risk	Project Sales Cloud	In progress	12
risque de coupure de courant	500000	Environmental Risk	Ten	Opened	12
risque de saisie de cocaine	85000	Financial risk	Compass	Closed	12
risque de perte de code2	8500	Financial risk	Sales Cloude	Opened	4
risque de changement de PM	890	Technological risk	Project Management	Stopped	16
risque de non respect des délais	60000	Technological risk	Sales Cloude	In progress	6
risk compass	50	Financial risk	Compass	Closed	4

< 1 2 >

Figure 52 Liste des risques

New

Search this list...

CRITICITY DEGREE	GRAVITY CODE
8	
2	
14	
12	
12	
12	
4	
16	
6	
4	

Figure 53 code couleur des risques

Au niveau de la liste chaque risque est évalué suivant sa criticité.

Avec la liste on peut filtrer les risques par projet avec un filtre. Au niveau du filtre on peut voir la liste de tous les projets de l'entreprise.

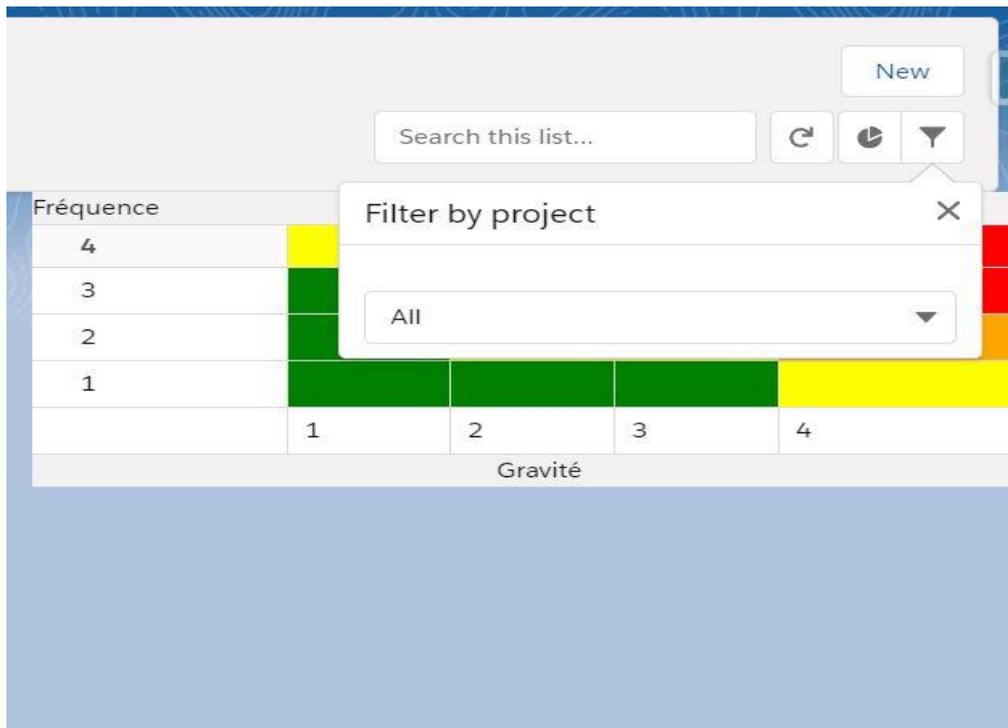
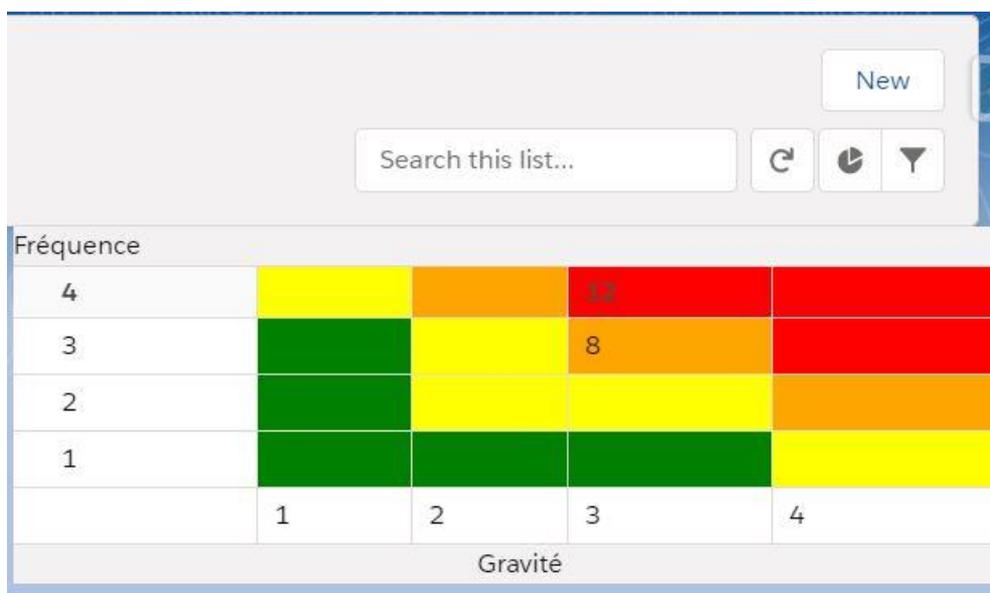


Figure 54 Filtre des risques Par Projet

Après la liste on a accès à la matrice des risques ou on peut avoir une vision sur tous les risques d'un projet quelconque avec les différents codes couleurs et les criticités des différents risques dans les cages correspondantes.



 Risks <b>List of Risks</b> 2 items · Updated 13 minutes ago			
TITLE	IMPACT	TYPE	STATUS
risk de perdre de code	10	Technological risk	Closed
risk	6300000	Financial risk	Closed

*Figure 55 Détails des risques pour un projet*

Cette page permettra au risque manager d'avoir une vue globale sur les risques du projet. La dernière page accessible par le Risk manager est la page de suivi des risques durant l'évolution du projet.

Au niveau de cette page tous les risques relatifs à un projet sont représentés dans le tableau suivant son indice de criticité. Ainsi on peut voir les projets les plus risqués et trouver des moyens de préventions on peut savoir aussi les risques qui impactent le plus sur le projet.

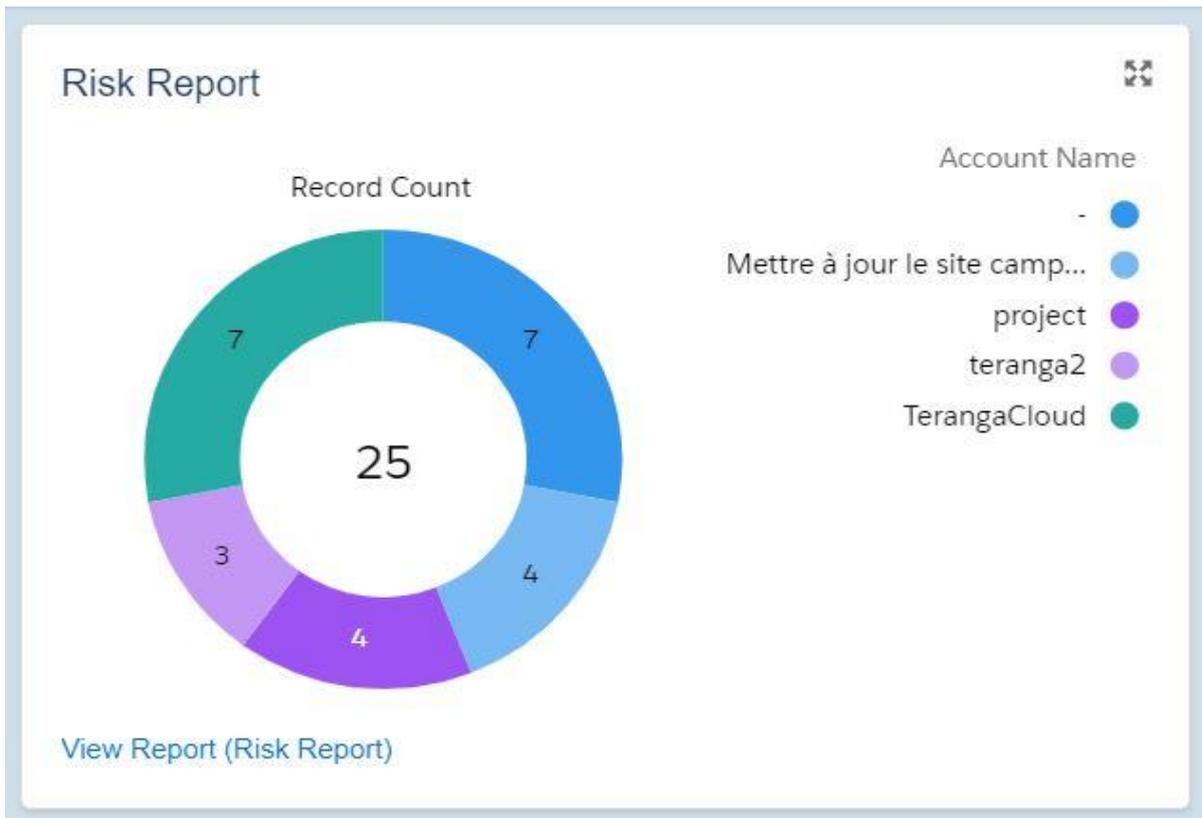


Figure 56 Tableau de Bord des risques par project1

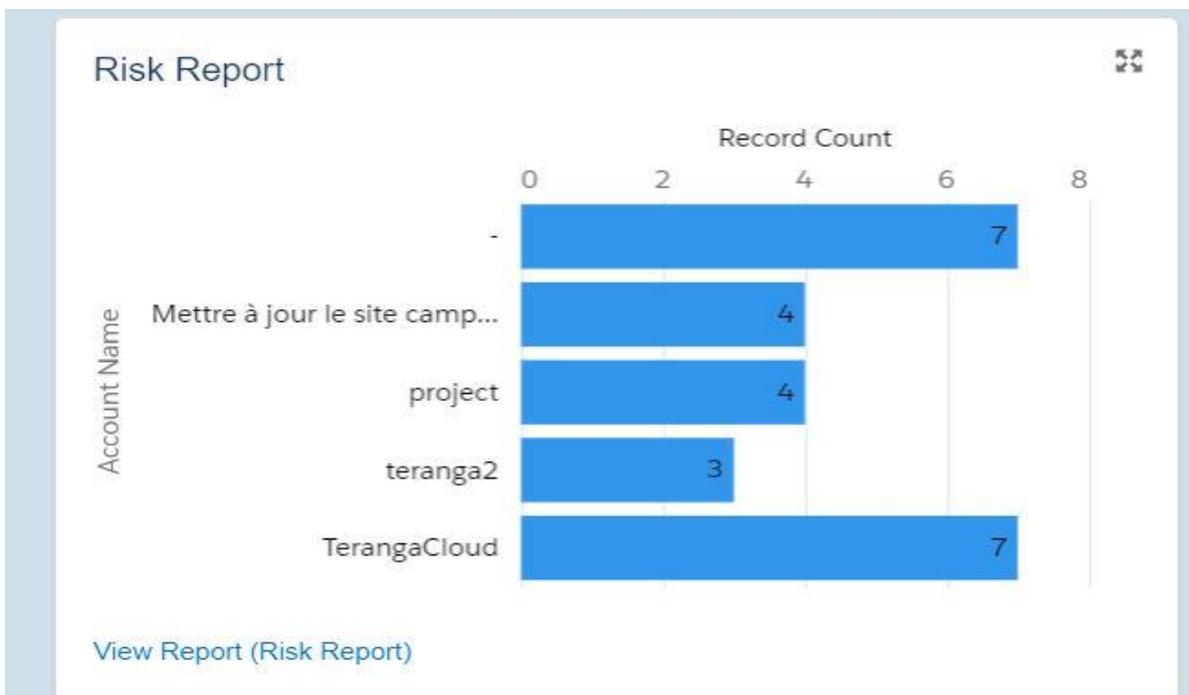


Figure 57 Tableau de bord des risques par Project2

Enfin on a un tableau de bords où on a la liste des risques les plus critiques et que le project Manager doit suivre tout le long du projet. On a ici les risques qui ont une criticité supérieure ou égale à douze.



Figure 58 Tableau de bord des criticités des risques

#### Conclusion :

L'architecture SOC est une méthode qui permet de séparer notre code en différentes parties pour lui permettre d'évoluer en cas de besoin. Pour générer un modèle de données Salesforce propose le schéma builder qui permet de créer des objets ou d'utiliser ses objets natifs et de créer les différents liens. Pour gérer l'accès aux différents pages Salesforce propose aussi des profils qui permettent de limiter ces accès.

Ainsi la solution finale permet de créer un risque et de le quantifier suivant certains paramètres et de définir le degré de criticité de ce risque. Nous avons proposé aussi des tableaux de bords qui permettent de savoir les projets les plus exposés aux risques.

### Conclusion générale et perspectives :

La gestion d'un risque dans un projet est indispensable dans l'exécution d'un projet. Une bonne gestion des risques permet de mener à bien un projet vu l'impact qu'un risque peut avoir sur un projet. Ce qui nous a poussé à développer un module de gestion des risques. Ainsi nous avons utilisé les technologies de force.com et le Framework Lightning de Salesforce pour la réalisation de notre module. Mais aussi dans le cadre notre projet nous avons utilisé utilisant la norme ISO 31000 2009 de gestion des risques qui nous permet de gérer les différents risques d'un projet en calculant la criticité qui détermine le comportement et la gravité du risque.

En perspectives nous comptons utiliser des alertes emails à tous les membres d'un projet pour la mise en garde sur certains risques. Nous projetons également de créer des tableaux de bords qui définissent l'état et la gravité d'un risque tout au long du projet. Ces tableaux de bords permettront de réduire la criticité d'un risque au fur et à mesure que le projet évolue.

Références :

[1] Salesforce module trailhead.

<https://trailhead.salesforce.com/en/modules>

[2] Salesforce Architecture.

[https://trailhead.salesforce.com/en/content/learn/modules/starting\\_force\\_com/starting\\_unders\\_tanding\\_arch](https://trailhead.salesforce.com/en/content/learn/modules/starting_force_com/starting_unders_tanding_arch)

[3] Multi-Tenant architecture.

[https://developer.salesforce.com/page/Multi\\_Tenant\\_Architecture](https://developer.salesforce.com/page/Multi_Tenant_Architecture)

[4] Understand Separation of Concern.

[https://trailhead.salesforce.com/en/content/learn/modules/apex\\_patterns\\_sl/apex\\_patterns\\_sl\\_soc](https://trailhead.salesforce.com/en/content/learn/modules/apex_patterns_sl/apex_patterns_sl_soc)

[5] Apprendre les principes de la couche service.

[https://trailhead.salesforce.com/fr/content/learn/modules/apex\\_patterns\\_sl/apex\\_patterns\\_sl\\_learn\\_sl\\_principles](https://trailhead.salesforce.com/fr/content/learn/modules/apex_patterns_sl/apex_patterns_sl_learn_sl_principles)

[6] Comprendre les principes de l'unité de travail.

[https://trailhead.salesforce.com/fr/content/learn/modules/apex\\_patterns\\_sl/apex\\_patterns\\_sl\\_learn\\_uow\\_principles](https://trailhead.salesforce.com/fr/content/learn/modules/apex_patterns_sl/apex_patterns_sl_learn_uow_principles)

[7] Learn Domain Layer Principles.

[https://trailhead.salesforce.com/en/content/learn/modules/apex\\_patterns\\_dsl/apex\\_patterns\\_dsl\\_learn\\_dl\\_principles](https://trailhead.salesforce.com/en/content/learn/modules/apex_patterns_dsl/apex_patterns_dsl_learn_dl_principles)

[8] Learn Selector Layer Principles.

[https://trailhead.salesforce.com/en/content/learn/modules/apex\\_patterns\\_dsl/apex\\_patterns\\_dsl\\_learn\\_selector\\_l\\_principles](https://trailhead.salesforce.com/en/content/learn/modules/apex_patterns_dsl/apex_patterns_dsl_learn_selector_l_principles)

[9] Cloud Computing : <https://www.lebigdata.fr/definition-cloud-computing>

[10] Salesforce Developer guide

<https://developer.salesforce.com>

[11] Application Lifecycle and Development Models.

<https://trailhead.salesforce.com/en/content/learn/modules/application-lifecycle-and-development-models/understand-what-application-lifecycle-management-is>

[12] Lightning Flows Salesforce :

[https://trailhead.salesforce.com/fr/content/learn/modules/business\\_process\\_automation/flow](https://trailhead.salesforce.com/fr/content/learn/modules/business_process_automation/flow)

[12] Concepts Apex et base de données.

[https://trailhead.salesforce.com/fr/content/learn/modules/apex\\_database/apex\\_database\\_soql](https://trailhead.salesforce.com/fr/content/learn/modules/apex_database/apex_database_soql)

[13] Séparation of Concern.

[https://en.wikipedia.org/wiki/Separation\\_of\\_concerns](https://en.wikipedia.org/wiki/Separation_of_concerns)

[14] Martin Fowler Separation of Concern.

<https://martinfowler.com/eaCatalog/>

[15] Qu'est ce que le langage apex.

[https://www.developerforce.com/guides/fr/apex\\_fr/Content/apex\\_intro\\_what\\_is\\_apex.htm](https://www.developerforce.com/guides/fr/apex_fr/Content/apex_intro_what_is_apex.htm)

[16] Modules|Salesforce Trailhead.

<https://trailhead.salesforce.com/fr/modules>

[17] Iso 31.000 2009 Principes et lignes directrices :

<https://www.iso.org/fr/standard/43170.html>

[18] Management du risque .

<https://www.pqb.fr/norms-2018-iso-31000-management-du-risque.php>

[19] Présentation de la norme Iso 31000 .

<https://www.pimido.com/business-comptabilite-gestion-management/management-et-organisation/fiche/norme-iso-31000-2008-ppt-423182.html>

[20] Gestion de projet Définition du risque.

[http://ressources.auneg.fr/nuxeo/site/esupversions/abf767af-234b-48ff-b2ec-2488500bc4ef/co/definition\\_1.html](http://ressources.auneg.fr/nuxeo/site/esupversions/abf767af-234b-48ff-b2ec-2488500bc4ef/co/definition_1.html)

[21] L'évaluation d'un risque

[https://www.cchst.ca/oshanswers/hsprograms/risk\\_assessment.html](https://www.cchst.ca/oshanswers/hsprograms/risk_assessment.html)

[22] Wikipedia Norme Iso 31000.

[https://fr.wikipedia.org/wiki/ISO\\_31000](https://fr.wikipedia.org/wiki/ISO_31000)

[23] Xactuim Risk Management

<https://www.xactium.com>

[24] Compliance Risk Management

<http://www.enterprisefeatures.com/what-is-compliance-risk-management/>

[25] Salesforce CRM

<https://www.salesforce.com/fr/>

[26] Wikipedia Salesforce

<https://fr.wikipedia.org/wiki/Salesforce.com>

[27] Présentation de Salesforce .

<http://www.comparatif-crm.com/salesforce-presentation-solution>

## Annexes:

### ➤ Correspondance avec les Objets Salesforce

Tables	Objet salesforce correspondant	champs manquants	Reutilisabilité			
Projects	Account	2 datatimes start date end date				
Tasks	Asset	1 number/				
Activities	Case					
Timesheets	Contract		on crée la relation entre lui et compaign de work unit			
Risks	Opportunity	1 Text long area measures				
Users	User					
Members	Contact					
ActivityLog	Campaign Members					
Reports	Contract					
OrganisationUnit	SPM_Organisation					
Deliverable	Order					
Expenses	case	1 text note	on cré la relation avec contract			
Absences	Campaign Members	1 number duration champ required				
InvoiceLines	opportunity	1 number units				
Invoices	contract					
Customers	contact					
Programs	compaign		on crée la relation entre lui et account			
TimesheetStatus	SPM_Organisation					
WorkUnit	compaign					
Members_has_project	contract					
Timesheet_has_timesheet	classe d'association					
Customer_has_project	Classe d'association					
Documents	Document de Salesforce					

- Correspondance entre les attributs des classes et les attributs des objets Salesforce

Objet du Modèle	Attributs de la Classe	Attributs de l'objet	Etat	Objet Salesforce
<b>Project</b>	Title	account name	<b>Done</b>	<b>Account</b>
	start date	start date		
	end date	End date		
	Description	Description		
	Delivry date	Sla expiration date		
	Project budget	Project buget		
	Percent complete	NAICS Description		
	Catégorie status	Rating		
	type			
<b>Task</b>	Description	Description	<b>Done</b>	<b>Asset</b>
	Start Date	StartDate__c		
	End date	EndDate__c		
	Link	Product code		
	Estimate Hours	Quantity		
	bilable hours	Asset Level		
	Logged Hours	Logged hours		
	Status	SPM_StatusCs		
	Work Unit			
	Name	Asset Name		

<b>Activites</b>	Name	Subject	<b>Done</b>	<b>Case</b>
	Cost	Cost__c		
	Short Name	SuppliedName		

<b>Timesheet</b>	Hrs total	Contract term (months)	<b>Done</b>	<b>Contract</b>
<b>Risk</b>	Title	Name	<b>Done</b>	<b>Oppportunity</b>
	Type	Type		
	Status	Delivry installation status		
	Impact	Next step		
	measure	measure		
	Probability	Probability(%)		
	Description	description		
<b>User</b>	FirstName	FirstName	<b>Done</b>	<b>User</b>
	LastName	LastName		
	Username	Username		
	Password	Name		
	User_role	CompanyName		
	User_profile	Division		
<b>Members</b>	Name	Name	<b>Done</b>	<b>Contact</b>
	Gender	LeadSource		
	Start date	Birthdate		
	Contract type	Level__c		
	Location	Title		
	End date	EndDate__c		
	Organisation Unit	IndividualId		
	Start activity	created date		

<b>Activity Log</b>	End Date	First responded date	<b>Done</b>	<b>Campaign Members</b>
	Activity type	Last Name		
<b>Organisation</b>	Name	Consumer credit Score Provider name	<b>Done</b>	<b>SPM_Organi</b>
<b>Delivarable</b>	Description	Description	<b>Done</b>	<b>Order</b>
	Short Name	Order Name		
	dead line	Contract end date		
	status	Status		
<b>Expenses</b>	Expense date	CreatedDate	<b>Done</b>	<b>Case</b>
	Amount	amount__c		
	expense type	Type		
	external company	SuppliedCompany		
	payment type	Origin		
	billable	SuppliedName		
	remboursable	SuppliedPhone		
	note	SPM_Description__c		
	Members_has_projects	Member_has_project__c		
<b>Absences</b>	type	SuppliedPhone	<b>Done</b>	<b>Case</b>
	Valid for	SuppliedName		
	Status	SuppliedCompany		
	Start Date	CreatedDate		
	End date	EndDate__c		
	duration	Cost__c		

<b>Timesheet status Note</b>		occupation	<b>Done</b>	<b>Individual</b>
<b>Work Unit</b>	Monday	Contact in campaign	<b>Done</b>	<b>Campaign</b>
	Tuesday	Contact in hierarchy		
	Wednesday	converted leads in campaign		
	Tuesday	converted leads in hierarchy		
	Friday	Lead in campaign		
	Saturday	Lead in hierarchy		
	Sunday	Num sent in campaign		
	Hrs Total	Num sent in hierarchy		
<b>Members has project</b>	role	Type	<b>Done</b>	<b>Contract</b>
	start Date	StartDate		
	End Date	End_Date__c		
	Status	Status_Membre__c		