

République Tunisienne

Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique

Université de Carthage

École Nationale d'Ingénieurs de Carthage



# RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Informatique

Par

**Lina KSOURI**

---

## Conception et développement d'une solution de gestion de risque

---

Encadrant professionnel : **M. Mohamed MELITI** Manager

Encadrante académique : **Mme Olfa LAMOUCHI** Maître assistante

Réalisé au sein de Vneuron



Année Universitaire 2021 - 2022



République Tunisienne  
Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique  
Université de Carthage  
École Nationale d'Ingénieurs de Carthage

# RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du  
Diplôme National d'Ingénieur en Informatique

Par

**Lina KSOURI**

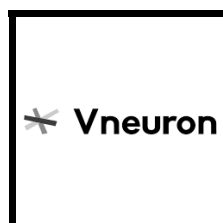
---

## Conception et développement d'une solution de gestion de risque

---

Encadrant professionnel :	<b>M. Mohamed MELITI</b>	Manager
Encadrante académique :	<b>Mme Olfa LAMOUCHI</b>	Maître Assistante

Réalisé au sein de Vneuron



Encadrant professionnel

**M. Mohamed MELITI**

**Signature et cachet**

Encadrant académique

**Mme Olfa LAMOUCHI**

**Signature**

# *Dédicaces*

*Je dédie ce travail,*

*A mes chers parents*

*pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études. Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours eu pour vous.*

*A mes chères sœurs Féten et Chahd*

*pour leurs encouragements permanents, et leur soutien moral, Je vous dis merci et je vous souhaite bonheur, réussite et prospérité.*

*A toute ma famille*

*pour leur soutien tout au long de mon parcours universitaire, Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible*

*À mes meilleurs amis*

*pour leur soutien constant et leurs amour qui est le meilleur des encouragements*

*À tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail*

*J'espère qu'ils trouveront dans ce travail toute ma reconnaissance.*

*Merci d'être toujours là pour moi.*

*Lina KSOURI*

# Remerciements

Au terme de ce projet, Je tiens à remercier tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Tout d'abord, Je voudrais adresser toute ma gratitude à Mohamed MELITI mon encadrant professionnel , pour sa confiance et les connaissances qu'il a su partager avec moi. Je le remercie aussi pour sa disponibilité au sein de l'entreprise et la qualité de son management.

Je tiens également à remercier Madame Olfa LAMOUCI pour sa disponibilité, ses précieux conseils, ses efforts et pour l'accueil chaleureux qu'elle nous a toujours réservé.

De même, j'exprime mes respects à toute l'équipe de la société Vneuron qui m'accueille parmi eux et qui m'a offert un excellent cadre de travail.

Avec beaucoup d'égards, je ne manquerais pas d'exprimer ma grande reconnaissance à tous les membres de jury pour avoir accepté de juger ce modeste travail.

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Cadre général du projet</b>	<b>3</b>
Introduction . . . . .	3
1.1 Organisme d'accueil . . . . .	3
1.1.1 Présentation de l'organisme d'accueil . . . . .	3
1.1.2 Business Unit . . . . .	3
1.1.3 Partenaires . . . . .	4
1.1.4 Secteur d'activité . . . . .	4
1.2 Cadre du projet . . . . .	5
1.2.1 Étude de l'existant . . . . .	5
1.2.2 Présentation ERM . . . . .	7
1.3 Méthode de travail . . . . .	9
1.3.1 Présentation de la méthode Scrum . . . . .	9
<b>2 Analyse et spécification des besoins</b>	<b>12</b>
2.1 Capture des besoins . . . . .	12
2.1.1 Identification des acteurs . . . . .	12
2.1.2 Besoins fonctionnels . . . . .	13
2.1.3 Besoins non fonctionnels . . . . .	14
2.2 Backlog du produit . . . . .	14
2.3 Diagramme de cas d'utilisation . . . . .	18
2.3.1 Diagramme de cas d'utilisation global . . . . .	18
2.3.2 Diagramme de cas d'utilisation détaillé : . . . . .	18
2.4 Diagramme de séquence système . . . . .	23
2.4.1 S'inscrire . . . . .	23
2.4.2 S'authentifier . . . . .	24
2.4.3 Ajouter entity . . . . .	24
2.4.4 Ajouter risque . . . . .	25

2.4.5	Analyser risque . . . . .	25
2.4.6	Modifier entity . . . . .	26
2.4.7	Supprimer employé . . . . .	26
<b>3</b>	<b>Architecture Et Conception</b>	<b>28</b>
3.1	Architecture du système . . . . .	28
3.1.1	Architecture matérielle . . . . .	28
3.1.2	Architecture logicielle . . . . .	29
3.2	Conception détaillé . . . . .	31
3.2.1	Diagramme de classes . . . . .	31
3.2.2	Diagramme de séquence objet . . . . .	33
3.2.3	Entrôpot de données . . . . .	39
<b>4</b>	<b>Réalisation</b>	<b>41</b>
4.1	Environnement de travail . . . . .	41
4.1.1	Environnement matériel . . . . .	41
4.1.2	Environnement logiciel . . . . .	41
4.1.3	Technologies utilisées . . . . .	42
4.2	Réalisation . . . . .	44
4.2.1	Interface d'authentification . . . . .	44
4.2.2	Interface d'inscription . . . . .	44
4.2.3	Interface de gestion des entités tiers . . . . .	45
4.2.4	Interface de gestion des entités . . . . .	47
4.2.5	Interface de gestion des risques . . . . .	49
4.2.6	Interface de gestion de contrôle . . . . .	51
4.2.7	Interface de gestion d'utilisateur . . . . .	53
	<b>Conclusion générale et perspectives</b>	<b>55</b>
	<b>Bibliographie</b>	<b>56</b>
	<b>Annexes</b>	<b>58</b>
	Annexe 1. Dictionnaire des données . . . . .	58
	Annexe 2. Matrice de risque . . . . .	60



---

Annexe 3. Exemple de JWT . . . . .	61
------------------------------------	----

# Table des figures

1.1	Partenaires Vneuron [3] . . . . .	4
1.2	Domaines d'activité [2] . . . . .	4
1.3	Les Quatre Étapes du Processus ERM . . . . .	8
1.4	Le fonctionnement de la méthode Scrum [7] . . . . .	10
2.1	Diagramme de contexte statique . . . . .	13
2.2	Diagramme de cas d'utilisation global . . . . .	18
2.3	Diagramme de cas d'utilisation : gérer des entités . . . . .	19
2.4	Diagramme de cas d'utilisation : gérer les risques . . . . .	21
2.5	Diagramme de cas d'utilisation : gérer les utilisateurs . . . . .	22
2.6	Diagramme de séquence système « S'inscrire » . . . . .	23
2.7	Diagramme de séquence système « S'authentifier » . . . . .	24
2.8	Diagramme de séquence système « Ajouter entity » . . . . .	24
2.9	Diagramme de séquence système « Ajouter risque » . . . . .	25
2.10	Diagramme de séquence système « Analyser risque » . . . . .	25
2.11	Diagramme de séquence système « Modifier entity » . . . . .	26
2.12	Diagramme de séquence système « Supprimer employé » . . . . .	27
3.1	Architecture 3-tiers . . . . .	29
3.2	Architecture MVC . . . . .	30
3.3	Architecture globale du projet[14] . . . . .	31
3.4	Diagramme de classe globale . . . . .	32
3.5	Diagramme de séquence objet « S'inscrire » . . . . .	34
3.6	Diagramme de séquence objet « S'authentifier » . . . . .	35
3.7	Diagramme de séquence objet « Ajouter entity » . . . . .	36
3.8	Diagramme de séquence objet « Ajouter risque » . . . . .	37
3.9	Diagramme de séquence objet « Modifier entity » . . . . .	38
3.10	Diagramme de séquence objet « Analyser risque » . . . . .	39
3.11	Table Risk analyse . . . . .	40
3.12	Processus d'analyse d'un risque . . . . .	40

4.1	Interface d'authentification . . . . .	44
4.2	Interface d'inscription . . . . .	45
4.3	Interface de lise des entités tiers . . . . .	46
4.4	Interface d'ajout d'une entité tier . . . . .	46
4.5	Interface de lise des entités . . . . .	47
4.6	Interface d'ajout d'une entité . . . . .	48
4.7	Interface d'affichage des détailles d'une entité . . . . .	48
4.8	Interface de lise des risques . . . . .	49
4.9	Interface d'ajout d'un risque . . . . .	50
4.10	Interface d'analyse d'un risque . . . . .	51
4.11	Interface de modification d'un contrôle . . . . .	52
4.12	Risque avant contrôle . . . . .	52
4.13	Risque après contrôle . . . . .	53
4.14	Interface 1 de la suppression . . . . .	53
4.15	Interface 2 de la suppression . . . . .	54
Annexe 2.1	Matrice de risque . . . . .	60
Annexe 3.1	Exemple de JWT . . . . .	61

# Liste des tableaux

1.1	Difference entre TRM et ERM[4]	7
2.1	Backlog du produit	15
2.2	Description textuelle « Ajouter entity »	19
2.3	Description textuelle « Consulter une entity »	20
2.4	Description textuelle « Modifier entity »	20
2.5	Description textuelle « Ajouter risque »	21
2.6	Description textuelle « Analyser un risque »	22
2.7	Description textuelle « Supprimer employé »	22
3.1	architecture 3-tiers dans ce projet	30
Annexe 1.1	Table entity_tier	58
Annexe 1.2	Table user	58
Annexe 1.2	Table Risque	58

# Liste des abréviations

**ERM** : Enterprise Risk Management.

**TIC** : Technologies de l'information et de la communication.

**TRM** : Traditionnal Ris Management.

**MVC** : Model View Controller.

**UML** : Unified Modeling Language.

**CU** : Cas Utilisation.

**API** : Application Protocol Interface.

**JWT** : Json Web Token.

**JSON** : JavaScript Object Notation.

**URL** : Uniform Resource Locattor.

**HTTP** : HyperText Transfert Protocol.

# Introduction générale

Les entreprises quelles que soient privées ou publiques font face à une forte diversification des risques qui sont un élément inévitable dans sa vie et qui peuvent menacer leurs existances à long terme, et cette exposition est liée au développement technologique, à la complexité des technologies utilisées et aussi l'absence de la culture de gestion de risque.

Par ailleurs, il existe de nombreuses classifications des risques, certaines sont dites financières parce qu'elles sont liées aux fluctuations des prix des matières premières, certaines sont dites juridiques parce qu'elles sont liées à l'évolution des textes légaux et réglementaires, et certaines sont dites opérationnels parce qu'ils proviennent de processus internes inadéquats ou défaillant de personnes et systèmes ou d'événements externes (fraudes et malveillances, OS, incendit et erreurs humaines...)

Dans ce contexte, plusieurs compagnies cherchent d'avantage à minimiser et à gérer ses risques et particulièrement les risques opérationnels qui deviennent une priorité. En effet, La gestion des risques a connu une évolution très significative au cours des dernières années et s'est développé à travers les pratiques et les besoins de l'entreprise, et d'autre part bénéficie de la recherche scientifique investie dans le domaine de la modernisation des modes de management.

Et pour bien gérer ses risques opérationnels, il y eu l'apparition d'une technique de management qui s'appelle ERM[1]. Cette dernière est une discipline qui aide à orienter la stratégie et la prise de décision, à réduire la probabilité et la gravité des événements à risque et à optimiser les ressources et le capital temps.

C'est dans ce cadre et pour l'obtention d'un diplôme national d'ingénieur en informatique, nous avons travaillé au sein de Vneuron pour concevoir et développer un outil de gestion de risque. Cette solution doit non seulement répondre à tous les objectifs des compagnies d'assurances ou de banques pour gérer leurs risques, mais aussi de les contrôler.

L'objectif de ce rapport est de présenter notre projet et de décrire ses différentes phases, ainsi que les résultats obtenus et les outils utilisés. Dans ce présent rapport nous allons aborder 4 chapitres qui détaillent les étapes effectuées durant la réalisation de notre projet.

Le premier chapitre est un chapitre introductif où nous présente le contexte général du projet, la société accueillante et décrit les grandes lignes du projet ainsi que la méthode adoptée pour la gestion du projet.

Le second chapitre traite la spécification des besoins fonctionnels et non fonctionnels, acteurs qui vont réagir avec notre solution et différents diagrammes de cas d'utilisation...

Le troisième Chapitre comporte l'analyse de notre solution permettant la description des modules et des entités à développer, ainsi que l'architecture logique que nous mettrons en œuvre.

Le quatrième chapitre sera consacré pour l'étude des technologies que nous avons utilisé ainsi que la réalisation du projet.

---

# CADRE GÉNÉRAL DU PROJET

---

## Introduction

Ce chapitre représente un premier aperçu du projet réalisé. En effet, nous commençons par une brève présentation de l'organisme d'accueil. Nous enchaînons ensuite par une description et une critique de l'existant, qui va nous guider vers la solution proposée et la méthodologie de conception adoptée.

## 1.1 Organisme d'accueil

### 1.1.1 Présentation de l'organisme d'accueil

Accompagner sa stratégie et soutenir ses ambitions en Tunisie et en Afrique, Ip-Tech Group, spécialisé dans les services informatiques, adopte le nom de marque Vneuron.

Fondé en 2007, Vneuron est un éditeur de logiciels et l'un des acteurs clés de la transformation numérique, de la gestion des risques et de la conformité réglementaire, et de l'excellence opérationnelle dans les banques, les compagnies d'assurance. Il a obtenu une position de leadership dans le domaine des solutions TIC au service des sociétés et institutions publiques et privées Tunisiennes et Africaines. Pour assurer plus de proximité, Vneuron compte plus de 200 clients répartis sur 3 continents à savoir l'Afrique, l'Europe et le Moyen Orient.[2].

### 1.1.2 Business Unit

Vneuron se décompose en 3 unités organisationnelles :

- **Digital Experience** : fournit des technologies pour gérer les documents d'expérience numérique et gérer les workflows d'intégration numérique.
- **Risk & Compliance** : un acteur clé dans la Lutte anti-Blanchiment de Capitaux



(LBC) et contre le Financement du Terrorisme (FT).

– **Operational Excellence** : fournisseur de solutions dédiées aux institutions financières pour la maîtrise des risques et le pilotage de l'activité.

### 1.1.3 Partenaires

Plusieurs principaux acteurs en Afrique et en Tunisie comptent parmi les partenaires de Vneuron parmi lesquelles on cite : Amen Capital, Aster Training, ISIE, GAT..



FIGURE 1.1 : Partenaires Vneuron [3]

### 1.1.4 Secteur d'activité

Vneuron est impliqué dans la création de propriété intellectuelle et le développement de logiciels dans les domaines suivants :



FIGURE 1.2 : Domaines d'activité [2]

## 1.2 Cadre du projet

Ce travail qui nous a été confié par l'entreprise était de développer une solution de gestion des risques qui nous permettrait de mieux identifier et analyser les risques pour optimiser les délais et les coûts.

Donc, dans cette partie, nous allons détailler l'étude de l'existant du projet puis présenter en général l'ERM.

### 1.2.1 Étude de l'existant

L'étude d'existant a été une étape importante et primordiale qui nous a permis d'identifier les problèmes et de reproduire la source des besoins réels, destinés à être pris en compte lors de la conception et de la mise en place de notre solution. Dans cette partie, nous allons faire une description de l'existence afin d'arriver à une solution décisive.

#### 1.2.1.1 Description de l'existant

Les compagnies d'assurance sont confrontées à un certain nombre de facteurs de risque qui peuvent avoir un impact très important et menacer leur survie à long terme. Donc, pour gérer ses risques, ils utilisent des méthodes traditionnelle et manuelle.

D'une part, le processus d'identification et d'évaluation des risques a été principalement géré à l'aide des feuilles Excel, de documents Word et de présentation PowerPoint.

D'autre part, ses sociétés d'assurance se concentrent essentiellement sur les risques de dangers c'est-à-dire les risques assurables comme les blessures causées par les chutes au travail ou les incendies et excluent les autres risques pour minimiser l'espace dans les fichiers Excel...

#### 1.2.1.2 Critique de l'existant

Avec l'évaluation continue des risques opérationnelles et suites aux gigantesques pertes causées par ce risques, nous avons constaté que cette technique utilisée manque de dynamisme et aussi il y a beaucoup de perte de temps au niveau de l'enregistrement des risques et à chaque mise à jour. En plus, nous risquons de perdre des données à cause de plusieurs problèmes. Parmi ces problèmes, on peut citer : problème de fraude ou piratage du système qui peut effacer tout les données existantes dans le fichier excel aussi le dommage de l'ordinateur qui cause le dommage de tout les fichiers. Et finalement, il y a manque de sécurité des données.

### 1.2.1.3 Solution proposée

Pour réduire les anomalies de cette technique et trouver une solution à ses problèmes, nous avons décidé de développer une solution pour gérer ses risques opérationnels, cette solution se base sur une technique de management qui s'appelle ERM.

Notre solution doit assurer principalement les exigences suivantes :

— **Un système de gestion des référentiels :**

Cette gestion nous permet de faire un registre de données nécessaires pour chaque société qui facilite la gestion des risques.

Ce registre se compose de 5 parties :

— *Gestion des entités tiers* : permet d'identifier les niveaux hiérarchiques des entités classes.

— *Gestion des entités classes* : permet d'identifier les classes qui constituent les entités types .

— *Gestion des entités types* : Ce système permet le regroupement des entités pour mieux faciliter le travail.

— *Gestion des entités* : Permet d'identifier les groupes qui constituent l'entreprise et qui offrent un ensemble d'activité.

— *Gestion des processus* : Cette gestion permet d'identifier les processus et les procédures afin d'identifier les risques dues à ses processus.

— **Un système de gestion des risques :**

Pour bien gérer les risques, ce système offre la possibilité d'identifier les risques existantes pour les évoluer.

— **Un système de gestion des contrôles :**

Cette étape de contrôle offre à l'utilisateur de l'assurance d'appliquer un contrôle interne à chaque risque.

— **Un système de gestion des utilisateurs** : Il permet au manager de gérer le personnel de la société.

### 1.2.2 Présentation ERM

La Gestion des risques de l'entreprise ERM est un processus structuré, actif et continu. Elle est appliquée à l'ensemble de l'organisation pour mieux comprendre tous les risques, leurs liens les uns avec les autres et leur impact cumulatif sur l'organisation. Il vise à ajouter de la valeur à l'organisation en minimisant les pertes et en maximisant les opportunités de croissance, aussi à permettre aux entreprises de gérer l'ensemble des risques auxquels elles sont confrontées, principalement les risques contingents, financiers, opérationnels et stratégiques.[1]

D'après les statistiques sur la gestion des risques d'entreprise, seulement 36% des sociétés qui ont un programmes pour la gestion des risques d'entreprise. Aussi, la crise générée par la Covid-19 prouve que l'ERM doit plus que jamais être une composante fondamentale de la stratégie et de la culture des organisations.

#### ► Différence entre TRM et ERM :

Les pratiques traditionnelles de gestion des risques concernent principalement le risque de perte résultant de l'exposition aux dangers. Il se caractérise par la standardisation et la rétrospective mais plus difficile à adapter au profil de risque ou aux circonstances de l'entreprise et aussi moins capable de s'adapter aux changements. Ce tableau montre la différence entre TRM et ERM :

**TABEAU 1.1** : Difference entre TRM et ERM[4]

	TRM	ERM
Cloisonné VS holistique	Les diverses fonctions de risque au sein d'une organisation possèdent leur risque respectif et on tendance à fonctionner en silos.	Adapte une vision plus holistique, en examinant les relations entre les différents types de risque.
Réactif VS Proactif	Réactif - tend à répondre aux incidents qui se sont produits et à se concentrer sur la prévention de la répétition des incidents.	Adapte une approche proactive de la gestion des risques en utilisant une combinaison de personnes, de processus et de technologie.
Assurable VS Non assurable	Se concentre uniquement sur les risques qui peuvent être assurés, par exemple, si un membre du personnel fait une chute au travail et se blesse, ou si une inondation endommage une partie d'un bureau.	Tient compte des risques assurables ainsi que de tout autre risque auquel une organisation est confrontée et auquel aucune somme d'argent ne peut remédier, comme une cyber-violation qui entraîne la perte de données.
Adaptabilité	Approches normalisées et prescrites	Fluide, adaptable, agile
État d'esprit	Aversion au risque : se concentre sur l'atténuation	Tolérance au risque : adopte une culture du risque à l'échelle de l'entreprise
Lien	Suivre des normes de base et limitées qui peuvent bloquer les opérations et apporter une valeur minimale à une organisation.	Suit des normes modernes telles que le cadre COSO et la norme ISO 31000, qui complètent les compétences techniques et non techniques requises pour étendre la gestion des risques.

► **Avantages de ERM :**

Plusieurs sont les avantages de l'ERM, les plus importants sont :

- **Favoriser la prise de décision** grâce à une vision holistique du risque et de son impact global sur l'entreprise.
- **Éliminez les silos** et facilitez la communication et la collaboration au sein de votre organisation.
- **S'adapte aux conditions changeantes** avec un processus souple d'identification des risques, de hiérarchisation des actions et de mesure des résultats en termes de valeur créée pour l'entreprise.
- **Favoriser une culture de sensibilisation aux risques** dans l'ensemble de l'organisation en collaborant avec des disciplines extérieures à la fonction de gestion des risques.

► **Processus de l'ERM :**

La mise en œuvre d'une démarche de management global des risques (ERM) est essentielle pour atteindre l'excellence en matière de gestion des risques. Cependant, il stipule que le processus de gestion des risques doit se composer de quatre étapes :



FIGURE 1.3 : Les Quatre Étapes du Processus ERM

- **Étape 1** : Identification des risques existants sur le plan qualitatif.
- **Étape 2** : Évaluation des risques en se basant sur un plan quantitatif. Évaluer la gravité et le niveau de probabilité des risques identifiés pour obtenir les résultats des valeurs à calculer.
- **Étape 3** : Réponse aux risques ou contrôle de risque qui consiste à trouver les bonnes pratiques pour bien minimiser les dangers.
- **Étape 4** : Surveillance des risques régulièrement pour assurer que les mesures d’atténuation appropriées et opportunes sont prises.

## 1.3 Méthode de travail

Afin d’assurer le bon déroulement d’un projet informatique, il est devenu crucial de suivre une méthode de travail tout au long du cycle de vie du projet, qui doit être la plus adaptative et adéquate au projet afin d’atteindre un résultat défini. Il existe plusieurs méthodes sur le marché, chacune avec ses propres caractéristiques, ce qui la rend intéressante sous certaines conditions. Dans notre projet, nous avons adapté l’approche agile qui est une approche itérative de la gestion de projet et du développement de logiciels qui permet aux équipes de créer de la valeur plus rapidement et plus facilement[6]. Plusieurs sont les méthodes de l’approche agile, mais nous avons choisi la méthode Scrum.

Dans cette partie, nous allons présenter la méthode Scrum et explique le choix de cette méthode.

### 1.3.1 Présentation de la méthode Scrum

**Scrum** est une méthodologie de développement agile, qui sert à développer tout projet innovant et complexe. Il s’agit d’une approche itérative, incrémentale, flexible et efficace conçue pour apporter de la valeur au client tout au long du développement du projet. En Scrum, les membres de l’équipe se concentrent sur le même objectif afin de livrer un projet et répondre aux besoins du clients. [7]

Le choix de Scrum comme une méthode de pilotage pour ce projet était basé sur ses points forts. D’autres facteurs déterminent également ce choix :

- D’abord et avant tout, la méthode utilisée par Vneuron.
- Deuxièmement, la puissante capacité de Scrum à s’adapter aux changements grâce à de

courtes itérations.

Le schéma ci-dessous résume le processus scrum adopté par notre équipe.

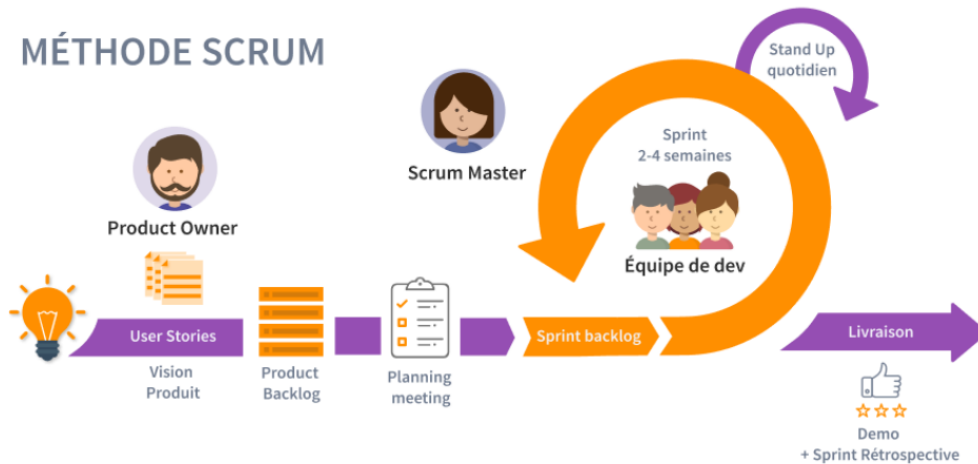


FIGURE 1.4 : Le fonctionnement de la méthode Scrum [7]

#### 1.3.1.1 Les rôles Scrum

SCRUM comporte trois rôles qui sont :

- **Le product owner** : représentant des clients et des utilisateurs, il est seul responsable de gérer le backlog et de s'assurer de la valeur du travail de l'équipe. De plus, il confirme ou infirme la version partiellement livrable présentée.
- **Le scrum master** : Expert du framework de méthodologie Scrum dans l'équipe. Il veille à la bonne exécution. Il guide l'équipe pour diriger le développement de produits vers le succès.[7]
- **L'équipe de développement** : Il contient de 2 à 5 développeurs. Leur responsabilité est de convertir le contenu du backlog produit en un sous-ensemble de fonctionnalités remplaçables à la fin de chaque Sprint.

#### 1.3.1.2 Les événements Scrum

On distingue cinq événements définis par Scrum :

- **Sprint**

Sa durée maximale est de 4 semaines. Il comprend la planification de sprint, le résumé quotidien, l'évaluation de sprint et la rétrospective de sprint. [8].

- **La planification du sprint**

La planification se fait par niveaux, chaque niveau correspond à un Sprint. Une réunion

de collaborateurs dure généralement 8 heures. l'équipe discute des objectifs du sprint et décide de ce qui sera accompli pendant cette période.[8].

- **La mêlée quotidienne**

Chaque jour, l'équipe doit se réunir au maximum 10 ou 15 minutes et répondre aux trois questions suivantes : •Qu'avez-vous fait hier ? •Qu'allez-vous faire aujourd'hui ? •Quels sont vos problèmes ?

- **La revue de sprint**

À la fin du sprint, une revue de sprint est effectuée qui comprend la vérification des incréments et l'ajustement du Product Backlog selon les besoins. Sa durée maximale est de 4 heures pour un sprint de 4 semaines. [8].

- **La rétrospective du sprint**

Le but de la réunion finale est de "Post mortem" du sprint et de voir si des erreurs ont été commises ou si certains points doivent être améliorés. Prend généralement 15 à 30 minutes pour faire un point et nous appliquons ici un principe Start/Stop/Continue.

## Conclusion

Dans ce premier chapitre, nous décrivons d'abord le contexte du projet en présentant l'organisme d'accueil. Puis nous présentons notre projet, en commençant par le problème puis le solution. Au final, nous avons décrire la méthodologie adapté dans notre projet.

Dans le chapitre suivant, nous avons analyser et spécifier les besoins du solution proposée.



---

# ANALYSE ET SPÉCIFICATION DES BESOINS

---

## Introduction

Une fois notre sujet cadré et avant le début du développement, il est important d'effectuer une analyse et une spécification des besoins, qui est une étape fondamentale et préliminaire pour assembler chaque projet et faciliter travail. Par conséquent, dans ce chapitre, nous identifions d'abord les acteurs qui spécifient les exigences fonctionnelles et non fonctionnelles. Ensuite, créez un backlog de produit pour votre projet. Enfin, nous présentons différents diagrammes de cas d'utilisation ainsi que diagrammes de séquence.

## 2.1 Capture des besoins

Dans cette partie, nous commençons par identifier les acteurs puis préciserons les exigences fonctionnelles et non fonctionnelles à satisfaire.

### 2.1.1 Identification des acteurs

Les acteurs sont les rôles joués par des entités externes qui interagissent avec le système et s'échangent d'informations d'entrée et/ou de sortie. Donc, les acteurs principaux de système sont :

- **Le manager** : Cet acteur a les droits les plus élevés. Dans un autre sens, il peut gérer n'importe quelle fonctionnalité fournie par le système tels que gestion des comptes, gestion d'un module référentiel qui contient la gestion des entités tiers, classes, types, entités et process gestion des risques et gestion des contrôles.
- **Employé** : Cette personne a les droits de gérer les fonctionnalités suivantes : gestion d'un module référentiel qui contient la gestion des entités tiers, classes, types, entités et process, gestion des risques et gestion des contrôles.

La figure 2.1 présente un diagramme de contexte statique montrant les relations entre divers acteurs avec le système.

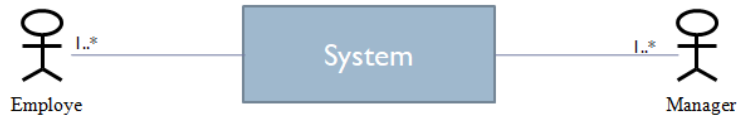


FIGURE 2.1 : Diagramme de contexte statique

### 2.1.2 Besoins fonctionnels

L'application résultante doit fournir un ensemble de fonctions qui doivent être liées à un ensemble d'exigences de l'utilisateur. Ceux-ci définissent les services que l'utilisateur souhaiterait voir fournis par cette application.

C'est pourquoi, dans cette partie, nous identifierons ses besoins :

- **L'inscription :**

Chaque utilisateur doit s'inscrire avec le rôle employé par défaut avant l'authentification et la gestion des autres fonctionnalités.

- **L'authentification :**

Une fois que chaque utilisateur a son propre compte, il peut s'identifier en saisissant ses données secrètes (username et mot de passe)

- **La gestion des utilisateurs :**

Seul le manager peut consulter liste des utilisateurs, supprimer un compte ou modifier les rôles des utilisateurs et créer un compte.

- **La gestion d'un module référentiel :**

Cette gestion permet d'identifier les parties nécessaires pour gérer les risques, elle est composée de 5 parties :

- Gestion des entités tiers : Chaque utilisateur peut créer un nouveau tier et consulter la liste des entités tiers.
- Gestion des entités classes : tous les employés de la société ont la possibilité d'ajouter un nouveau classe et consulter la liste des entités classes.
- Gestion des entités types : De même que les deux gestions précédentes, chacun peut ajouter et consulter liste des entités types.

— Gestion d’entity : Le manager ainsi que l’employé, peuvent créer, modifier et consulter les détails d’une entité.

— La gestion des processus : L’utilisateur peut créer, modifier, consulter les détails d’un process et la liste de tous les processus.

- **La gestion des risques :**

Le manager ainsi que l’employé ont la possibilité de créer un risque, le modifier ou consulter ses détails, consulter la liste des risques et finalement analyser et évoluer un risque.

- **Gestion de contrôle :**

Le manager, ainsi que l’utilisateur, peuvent consulter les contrôles existantes, affecter un contrôle pour un risque et ajouter un nouveau contrôle.

### 2.1.3 Besoins non fonctionnels

Ce sont les besoins qui sont indirectement liés au fonctionnement du système et assurent le bon fonctionnement des applications. L’ensemble des exigences non fonctionnelles de notre solution sont citées ci-dessous :

- **Modifiabilité et Réutilisabilité :** Il faut que Le code soit clair pour permettre une modification ou ajout de nouvelles fonctionnalités dans le futur et aussi réutiliser le code dans un autre système.

- **Ergonomie :** Cette solution doit fournir une interface simple et guidée pour faciliter son utilisation.

- Être **optimisée** et ne consomme pas beaucoup de mémoire et qu’elle soit rapide.

- **La fiabilité :** Les résultats apportés par la solution doivent être fiables et reflètent effectivement l’état de la base au moment de son interrogation, c’est-à-dire lors de la mise à jour des données.

- **Minimiser** le gaspillage du temps qui peut causer une perte de coût.

## 2.2 Backlog du produit

Le Product Backlog est défini par le Scrum Guide comme « une liste ordonnée et évolutive des choses qui doivent être faites pour améliorer le produit ». [9].

Le tableau 2.1 suivant résume le backlog du produit de notre solution.

**TABEAU 2.1** : Backlog du produit

<b>Id</b>	<b>Fonctionnalités</b>	<b>User stories</b>	<b>Acteurs</b>	<b>Priorité</b>
01	Inscription	<b>En tant qu'utilisateur, je veux</b> m'inscrire	Manager Employé	Elevée
02	Authentification	<b>En tant qu'utilisateur, je veux</b> m'authentifier	Manager Employé	Elevée
03	Gestion des utilisateurs	<b>En tant qu'utilisateur, je veux</b> consulter la liste des utilisateurs	Manger	Élevée
		<b>En tant qu'utilisateur, je veux</b> créer un employé	Manager	Élevée
		<b>En tant qu'utilisateur, je veux</b> modifier un employé	Manager	Élevée
		<b>En tant qu'utilisateur, je veux</b> supprimer un employé	Manager	Moyenne
04	Gestion des entités tiers	<b>En tant qu'utilisateur, je veux</b> consulter la liste des entités tiers	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter une nouvelle Entité Tier <b>afin</b> d'hierarchiser les entités de mon entreprises	Manager Employé	Élevée
05	Gestion des entités classes	<b>En tant qu'utilisateur, je veux</b> consulter la liste des entités classes	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter une nouvelle Entité Class	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de rattacher une Entity Class à une Entity Tier <b>afin</b> d'établir l'arborescence des entités.	Manager Employé	Élevée

06	Gestion des entités types	<b>En tant qu'utilisateur, je veux</b> consulter listes des entités types	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter une nouvelle Entité Type	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de rattacher une Entité Type à une Entité Class	Manager Employé	Élevée
07	Gestion des entités	<b>En tant qu'utilisateur, je veux</b> consulter listes des entités	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter une nouvelle Entité	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de rattacher une Entité à une Entité Type	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de consulter les détails d'une entité	Manager Employé	Faible
		<b>En tant qu'utilisateur, je veux</b> être capable de mettre à jour une Entité existante	Manager Employé	Moyenne
08	Gestion des processus	<b>En tant qu'utilisateur, je veux</b> consulter listes des processus	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter une nouvelle processus	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de rattacher une processus à une Entité	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de consulter les détails d'une processus	Manager Employé	Faible

		<b>En tant qu'utilisateur, je veux</b> être capable de mettre à jour une processus existante	Manager Employé	Moyenne
09	Gestion des risques	<b>En tant qu'utilisateur, je veux</b> consulter listes des risques	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter une nouveau risque	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de rattacher un risque à une processus	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de consulter les détails d'un risque	Manager Employé	Faible
		<b>En tant qu'utilisateur, je veux</b> être capable de mettre à jour un risque existant	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'analyser un risque	Manager Employé	Moyenne
10	Gestion des contrôles	<b>En tant qu'utilisateur, je veux</b> consulter listes des contrôle	Manager Employé	Moyenne
		<b>En tant qu'utilisateur, je veux</b> être capable d'ajouter un nouveau contrôle	Manager Employé	Faible
		<b>En tant qu'utilisateur, je veux</b> être capable de rattacher un contrôle à plusieurs risques	Manager Employé	Élevée
		<b>En tant qu'utilisateur, je veux</b> être capable de consulter les détails d'un contrôle	Manager Employé	Faible
		<b>En tant qu'utilisateur, je veux</b> être capable de mettre à jour un contrôle existant	Manager Employé	Moyenne

## 2.3 Diagramme de cas d'utilisation

Dans le langage UML, les diagrammes de cas d'utilisation permettent de modéliser le comportement d'un système et de capturer les exigences du système. Ils décrivent la fonctionnalité générale et la portée d'un système et identifier les interactions entre les acteurs et le système. [10].

### 2.3.1 Diagramme de cas d'utilisation global

La figure ci-dessous représente les différentes cas d'utilisation et les acteurs :

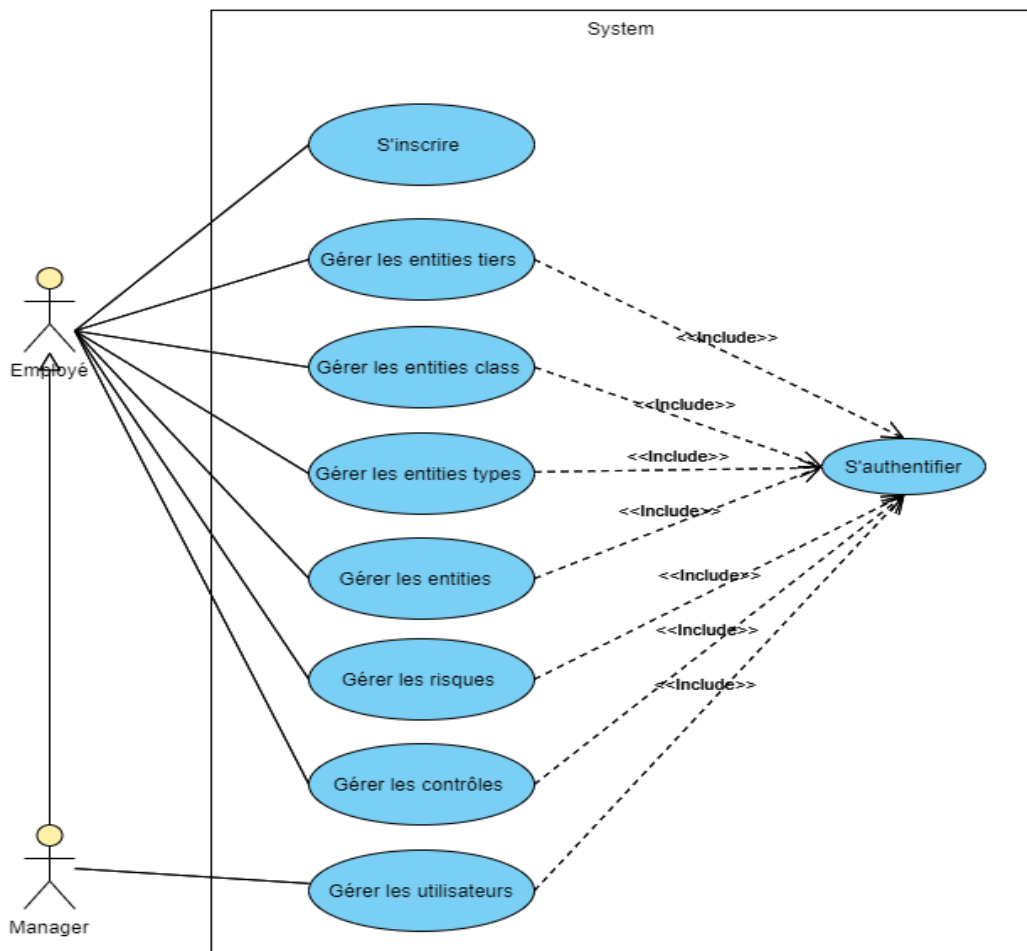


FIGURE 2.2 : Diagramme de cas d'utilisation global

### 2.3.2 Diagramme de cas d'utilisation détaillé :

Dans cette partie , on va détailler quelque cas d'utilisation en représentant des diagrammes de cas d'utilisation raffinés et une description textuelle.

#### ► Raffinement cas d'utilisation : Gérer les entités

La figure ci-dessous présente le diagramme de cas d'utilisation Gérer Entities

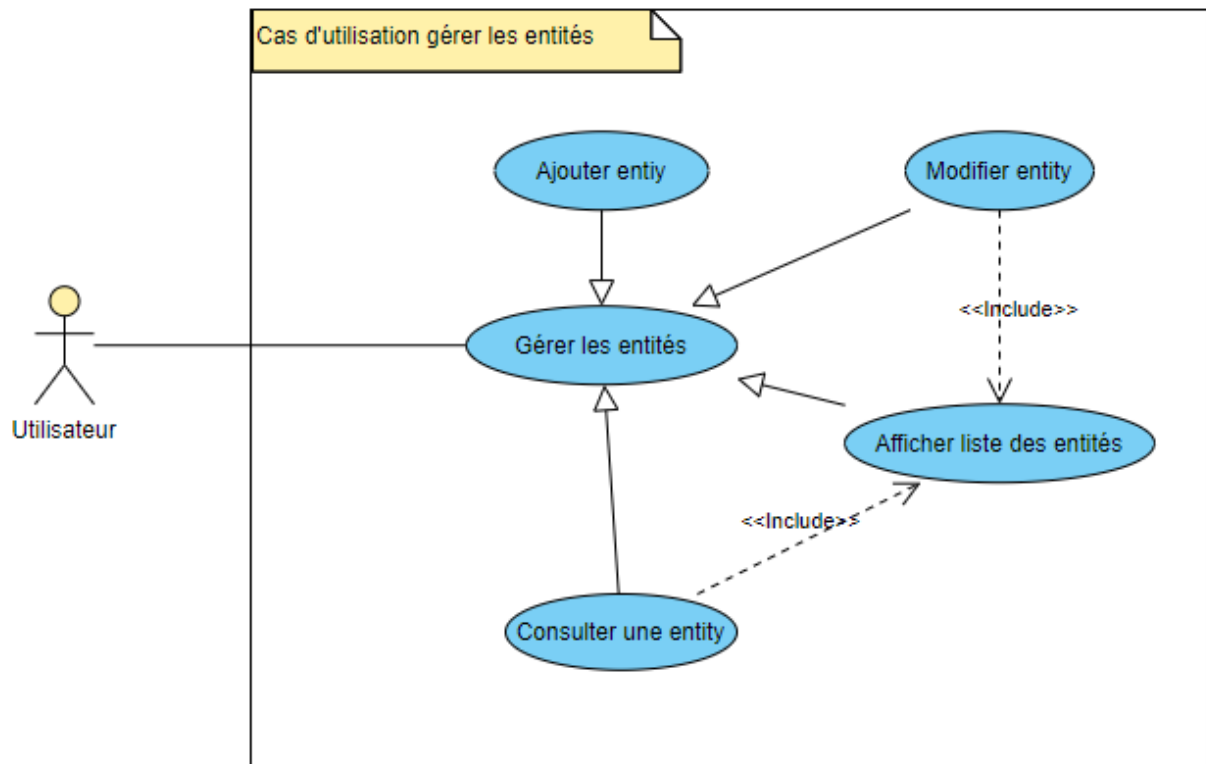


FIGURE 2.3 : Diagramme de cas d'utilisation : gérer des entities

Description textuelle du cas d'utilisation Ajouter entity :

TABLEAU 2.2 : Description textuelle « Ajouter entity »

<b>Cas d'utilisation</b>	Ajouter entity
<b>Acteur(s)</b>	Manager, Employé
<b>Pré condition(s)</b>	Authentification
<b>Post condition</b>	Entity ajoutée
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le manager/L'employé demande le formulaire d'ajout d'un entity</li> <li>2. Le système affiche une interface d'ajout .</li> <li>3. Le manager/L'employé remplit les champs et confirme l'ajout</li> <li>4. Le système vérifie les données saisies.</li> <li>5. Le système enregistre la nouvelle entity</li> </ol>
<b>Scénario alternatif</b>	4.A. des données manquantes ou non valide



**Description textuelle** du cas d'utilisation Consulter une entity :

**TABLEAU 2.3** : Description textuelle « Consulter une entity »

<b>Cas d'utilisation</b>	Consulter une entityy
<b>Acteur(s)</b>	Manager, Employé
<b>Pré condition(s)</b>	Authentification
<b>Post condition</b>	Détailles de l'entity choisi sont affiché
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le manager/L'employé consulte la liste des entities.</li> <li>2. Le manager/L'employé choisit l'entity parmi la liste pour consulter ses détails .</li> <li>3. Le système affiche les détails de l'entity .</li> </ol>

**Description textuelle** du cas d'utilisation Modifier entity :

**TABLEAU 2.4** : Description textuelle « Modifier entity »

<b>Cas d'utilisation</b>	Modifier entity
<b>Acteur(s)</b>	Manager, Employé
<b>Pré condition(s)</b>	Authentification
<b>Post condition</b>	Entity Modifiée
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le manager/L'employé consulte la liste des entities.</li> <li>2. Le manager/L'employé choisit l'entity à modifier .</li> <li>3. Le système affiche le formulaire de modification .</li> <li>4. Le manager/L'employé saisit les changements qu'il souhaite faire.</li> <li>5. Le système vérifie les nouvelles données saisies.</li> <li>6. Le système enregistre les modifications.</li> </ol>
<b>Scénario alternatif</b>	5.A. des données manquantes ou non valide

### ► Raffinement cas d'utilisation : Gérer les risques

La figure ci-dessous présente le diagramme de cas d'utilisation Gérer les risques

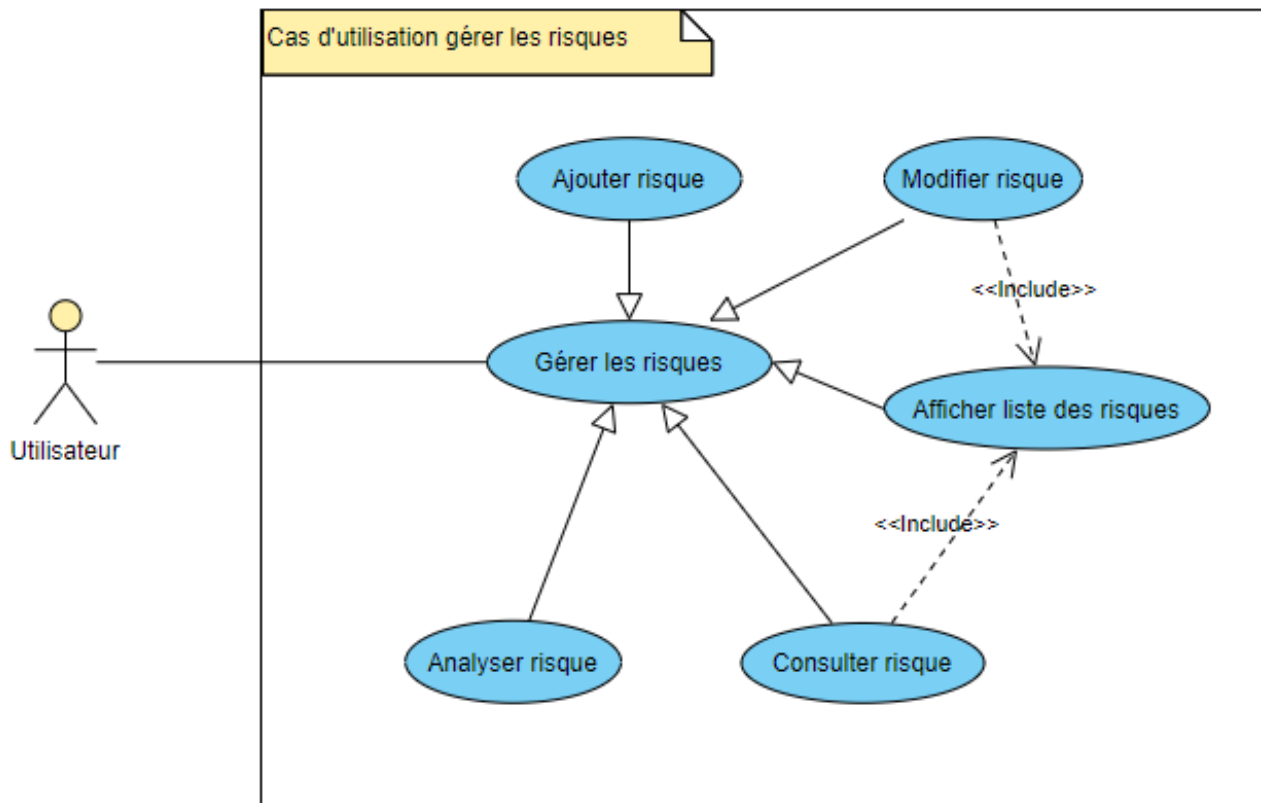


FIGURE 2.4 : Diagramme de cas d'utilisation : gérer les risques

Description textuelle du cas d'utilisation Ajouter un risque :

TABLEAU 2.5 : Description textuelle « Ajouter risque »

<b>Cas d'utilisation</b>	Ajouter risque
<b>Acteur(s)</b>	Manager, Employé
<b>Pré condition(s)</b>	Authentification
<b>Post condition</b>	Risque ajouté
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le manager/L'employé demande le formulaire d'ajout d'un risque</li> <li>2. Le système affiche une interface d'ajout .</li> <li>3. Le manager/L'employé remplit les champs et confirme l'ajout</li> <li>4. Le système vérifie les données saisies et calcule le risque brut.</li> <li>5. Le système enregistre le nouveau risque.</li> </ol>
<b>Scénario alternatif</b>	4.A. des données manquantes ou non valide

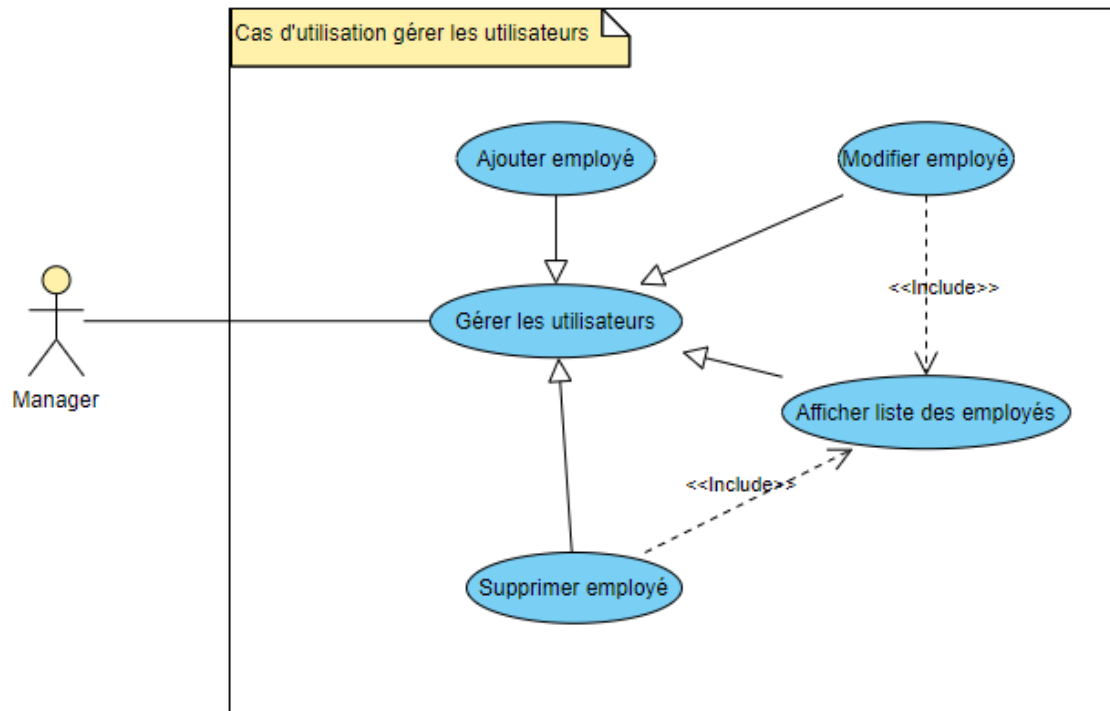
**Description textuelle** du cas d'utilisation Analyser un risque :

**TABLEAU 2.6 :** Description textuelle « Analyser un risque »

<b>Cas d'utilisation</b>	Analyser un risque
<b>Acteur(s)</b>	Manager, Employé
<b>Pré condition(s)</b>	Authentification
<b>Post condition</b>	un graphique est affiché
<b>Scénario nominal</b>	1. Le manager/L'employé choisir le nom du risque à analyser. 2. Le système affiche la valeur du risque brut et un graphique .

► **Raffinement cas d'utilisation : Gérer les utilisateurs**

La figure ci-dessous présente le diagramme de cas d'utilisation Gérer les utilisateurs



**FIGURE 2.5 :** Diagramme de cas d'utilisation : gérer les utilisateurs

**Description textuelle** du cas d'utilisation supprimer un employé :

**TABLEAU 2.7 :** Description textuelle « Supprimer employé »

<b>Cas d'utilisation</b>	Supprimer employé
<b>Acteur(s)</b>	Manager
<b>Pré condition(s)</b>	Authentification

<b>Post condition</b>	Employé supprimé
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le manager/L'employé consulte la liste des employés</li> <li>2. Le manager/L'employé choisit employé à supprimer</li> <li>3. Employé choisi supprimer .</li> </ol>

## 2.4 Diagramme de séquence système

Un diagramme de séquence est un diagramme qui représente les messages échangés entre les objets( système,utilisateurs...) lors de leur interaction.[11].

Dans cette partie, on va faire un analyse des cas d'utilisation en considérant le système comme une boîte noire.

### 2.4.1 S'inscrire

La figure 2.6 représente l'interaction entre le manager/l'employé et le système lors de l'inscription.

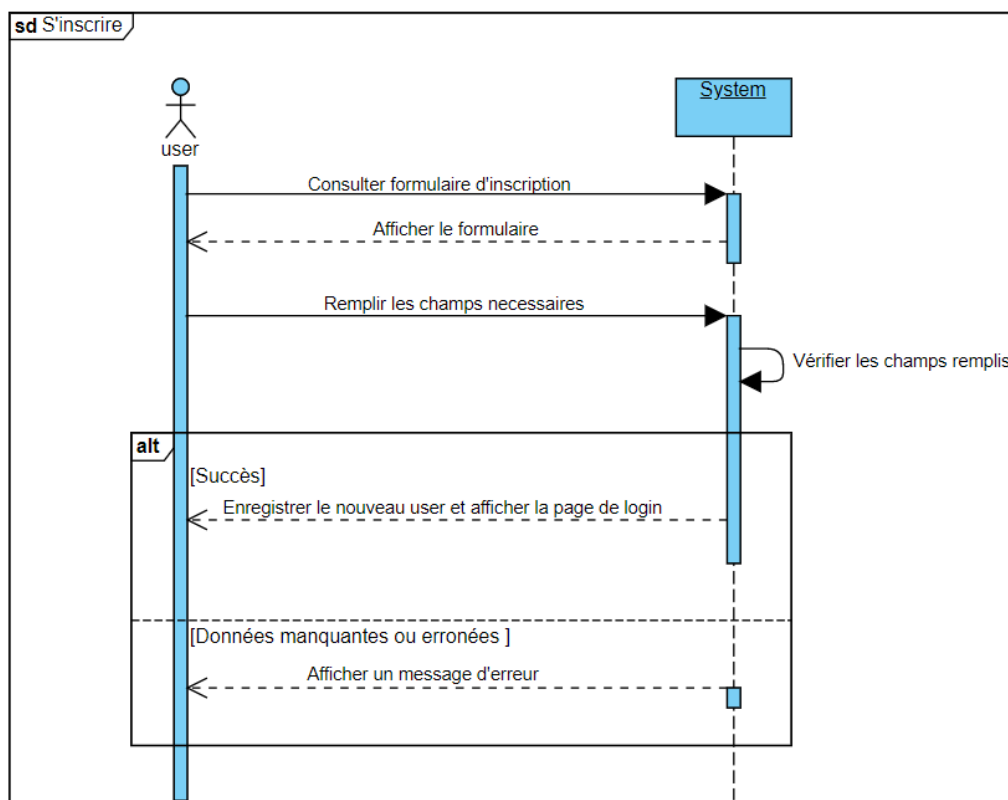


FIGURE 2.6 : Diagramme de séquence système « S'inscrire »

### 2.4.2 S'authentifier

La figure 2.7 illustre le diagramme de séquence système du CU « S'authentifier » .

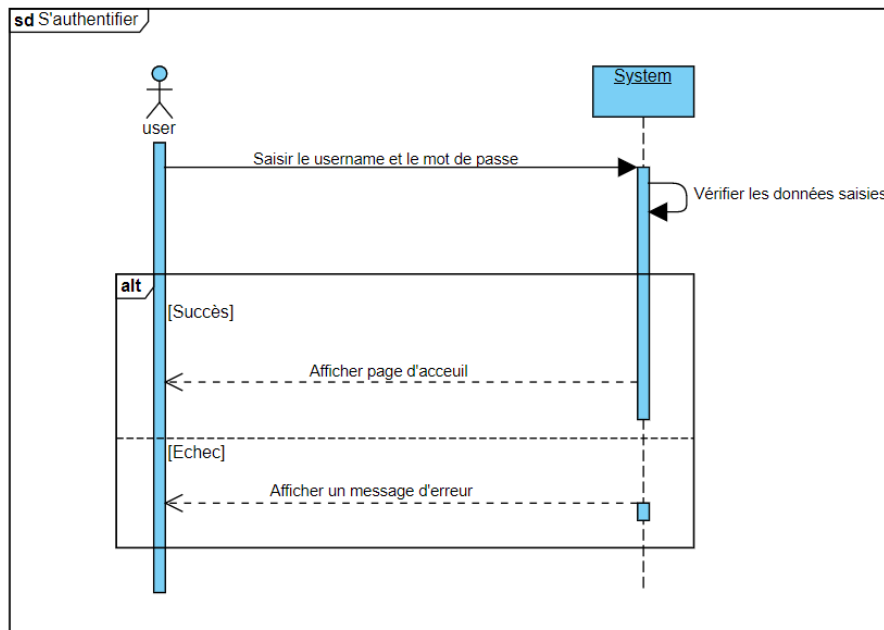


FIGURE 2.7 : Diagramme de séquence système « S'authentifier »

### 2.4.3 Ajouter entity

Pour mieux comprendre le déroulement du CU « Ajouter entity » , son diagramme de séquence système est illustré dans cette figure

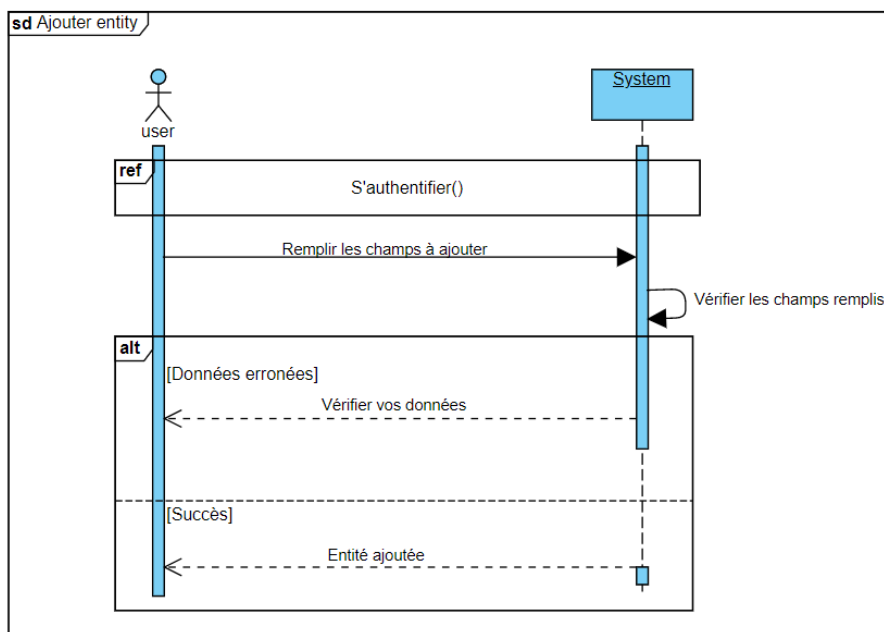


FIGURE 2.8 : Diagramme de séquence système « Ajouter entity »

### 2.4.4 Ajouter risque

La figure 2.9 représente le diagramme de séquence de l'Ajout d'un risque.

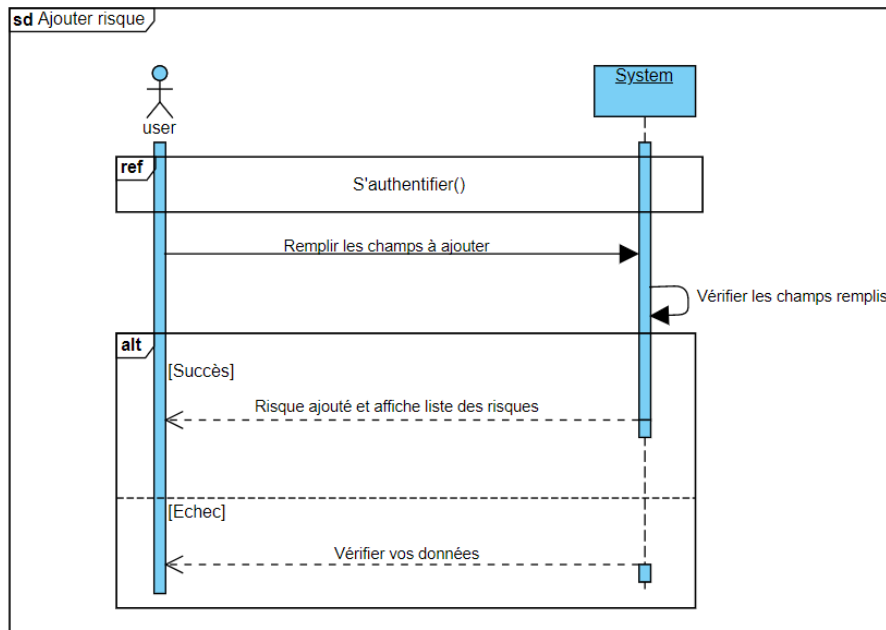


FIGURE 2.9 : Diagramme de séquence système « Ajouter risque »

### 2.4.5 Analyser risque

La figure 2.10 représente le déroulement de la cas d'utilisation de l'analyse d'un risque .

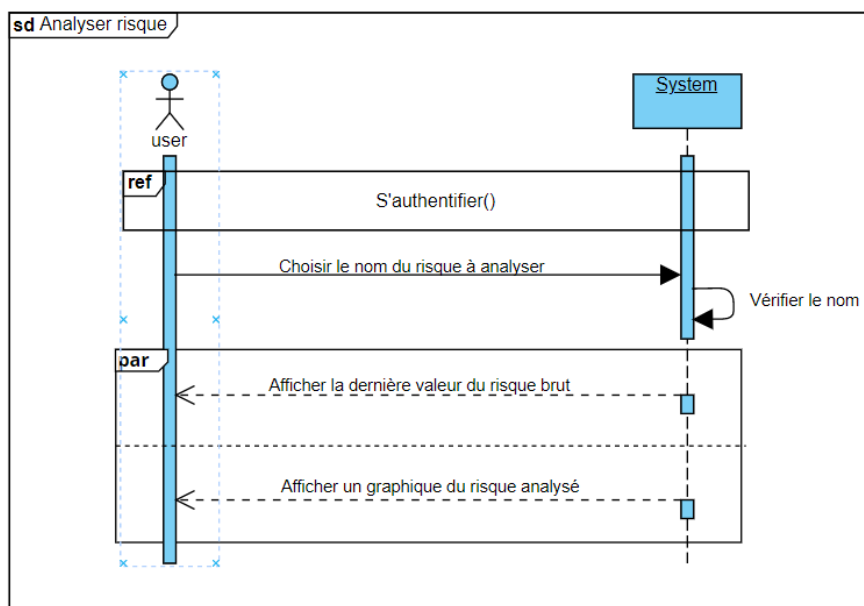


FIGURE 2.10 : Diagramme de séquence système « Analyser risque »

### 2.4.6 Modifier entity

Après avoir représenté le CU de modification d'une entity, on va représenter son diagramme de séquence pour bien comprendre son déroulement.

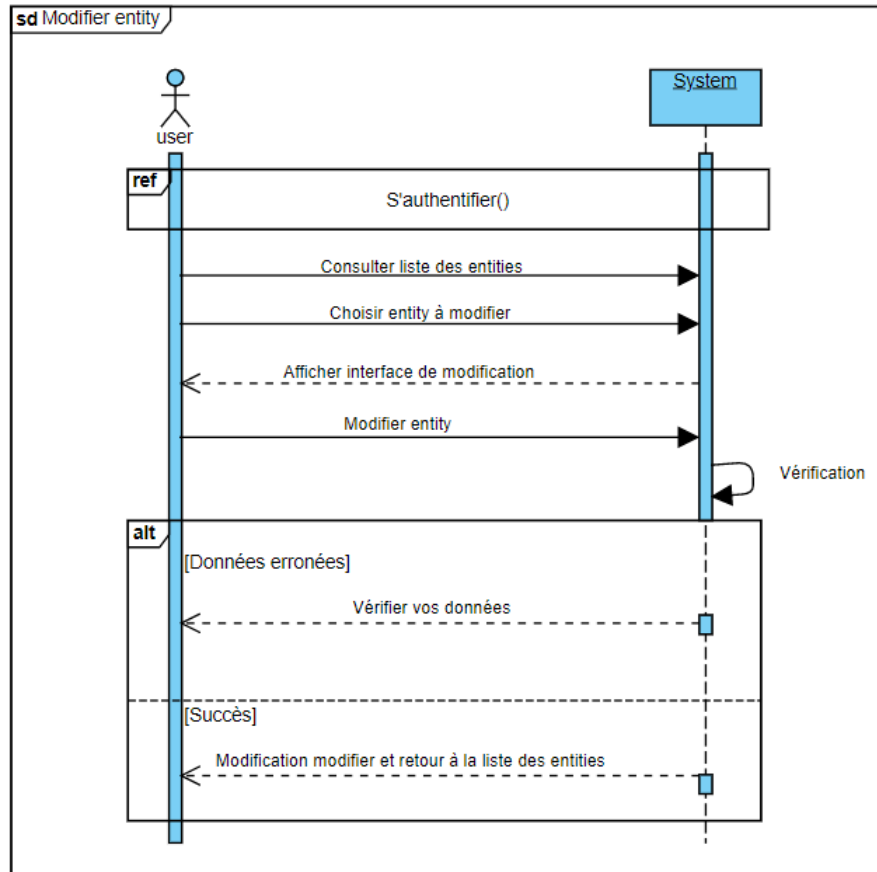


FIGURE 2.11 : Diagramme de séquence système « Modifier entity »

### 2.4.7 Supprimer employé

La figure ci-dessous montre le déroulement de la cas d'utilisation de suppression d'un employé .

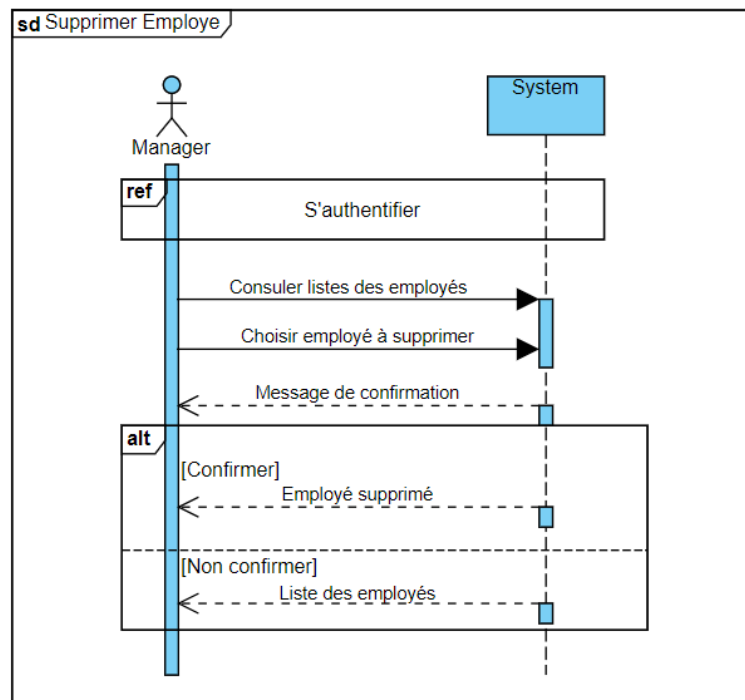


FIGURE 2.12 : Diagramme de séquence système « Supprimer employé »

## Conclusion

Dans ce chapitre, nous avons d'abord présenté les différents acteurs et documenté leurs exigences fonctionnelles et non fonctionnelles. Nous avons ensuite créé un backlog de produit et enfin présenté les diagrammes de cas d'utilisation et les diagrammes de séquence système pour nous aider à mieux comprendre le fonctionnement de la solution.

Le chapitre suivant se concentrera sur l'architecture et la conception de la solution



---

# ARCHITECTURE ET CONCEPTION

---

## Introduction

Dans le chapitre précédent, nous avons détaillé les besoins de notre solution et les analysé dans des diagrammes de cas d'utilisation et de séquence système.

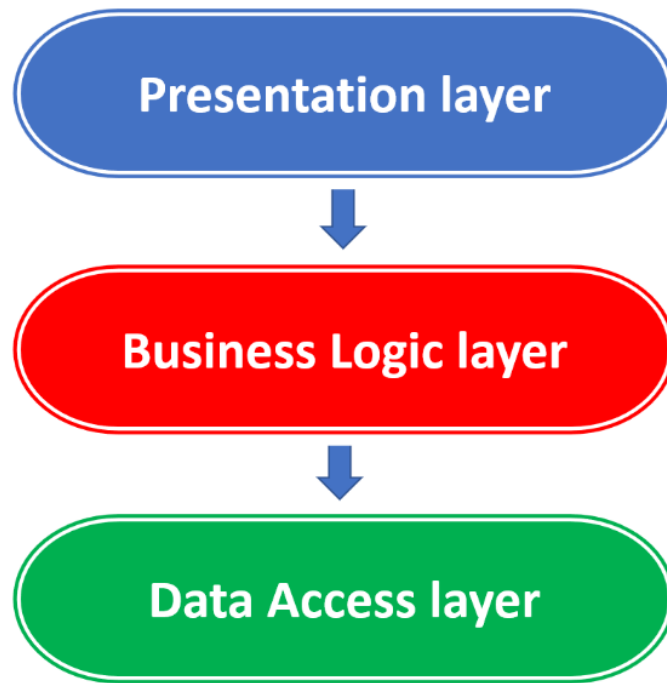
Maintenant, nous allons présenter d'abord l'architecture détaillée de la solution puis passer à la phase de conception.

## 3.1 Architecture du système

L'architecture du système est le modèle de base pour une mise en œuvre réussie d'une application Web. C'est avant tout l'organisation et la structure de ce dernier. Cela permet aux développeurs ou à toute personne intéressée par la conception et la construction de tels logiciels de bien comprendre comment cela fonctionne.[\[12\]](#).

### 3.1.1 Architecture matérielle

Pour notre solution, l'architecture choisie est l'architecture 3-tier qu'on appelle aussi architecture à 3 couches qui consiste à diviser le code en trois niveaux différentes. La figure 3.1 représente les trois niveaux de notre architecture :



**FIGURE 3.1 :** Architecture 3-tiers

**Couche de présentation :** C'est l'interface utilisateur, il présente toutes les fonctionnalités du système.[12]

**Couche métier :** Cette deuxième couche contient logique de métier, elle dirige les fonctionnalités du système comme le traitement des données échangées entre les deux autres couches.[12]

**Couche de données :** Au niveau de cette couche, on distingue l'interaction avec les bases de données pour stocker les données.[12]

### 3.1.2 Architecture logicielle

Le choix de MVC comme architecture logicielle pour notre projet dépend de plusieurs arguments. D'une part, cette architecture cherche toujours à séparer les trois couches présentation, métier et données. D'autre part, elle est facile à entretenir et nous permet de réutiliser les différents composants.

La figure ci-dessous décrit cette architecture :

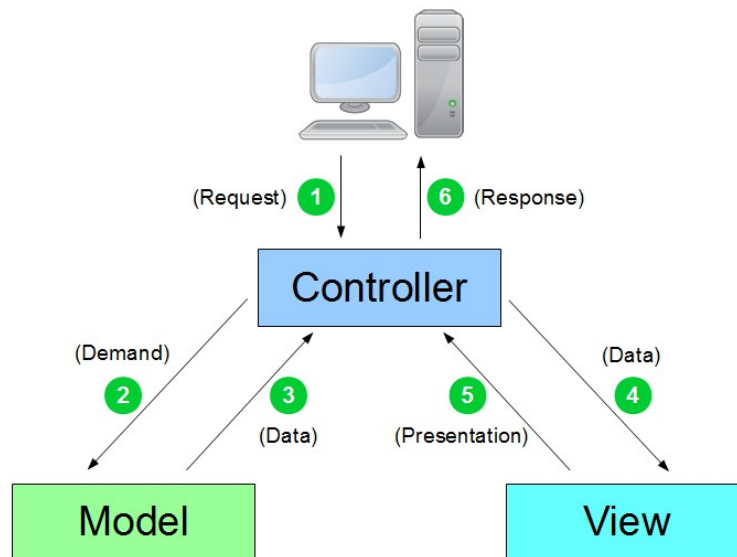


FIGURE 3.2 : Architecture MVC

**Model :** Le modèle est responsable de la maintenance des données, de la communication bidirectionnelle avec la base de données, puis de la transmission des données requises au contrôleur.[13]

**View :** Les vues sont créées par les données collectées des modèles, mais ces données ne sont pas collectées directement, mais via des contrôleurs, de sorte que les vues ne communiquent qu'avec les contrôleurs.[13]

**Controller :** C'est l'intermédiaire entre le modèle et la vue, il permet l'échange des données entre eux.[13]

En effet, dans le tableau suivant, nous avons identifier chaque couche pour la partie **Backend** et la partie **Frontend** :

TABLEAU 3.1 : architecture 3-tiers dans ce projet

	Backend	Frontend
<b>Couche présentation</b>	MVC	Component + Router
<b>Couche métier</b>	Les services	Angular services
<b>Couche données</b>	Repositories	HTTP calls

La figure ci-contre montre l'architecture globale de notre projet :

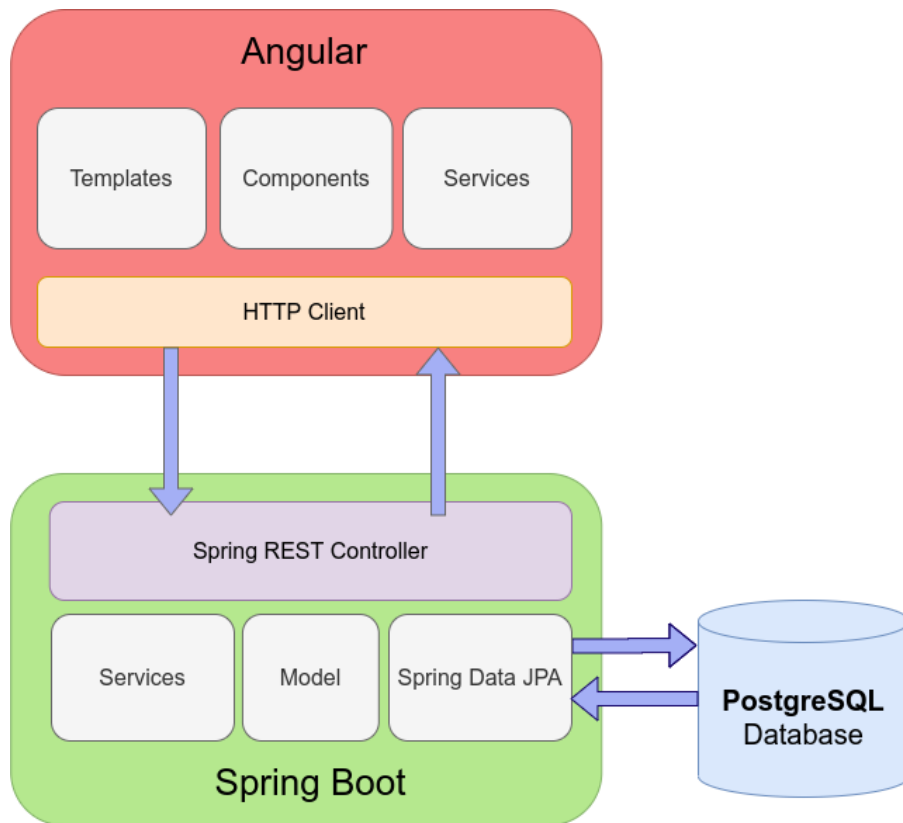


FIGURE 3.3 : Architecture globale du projet[14]

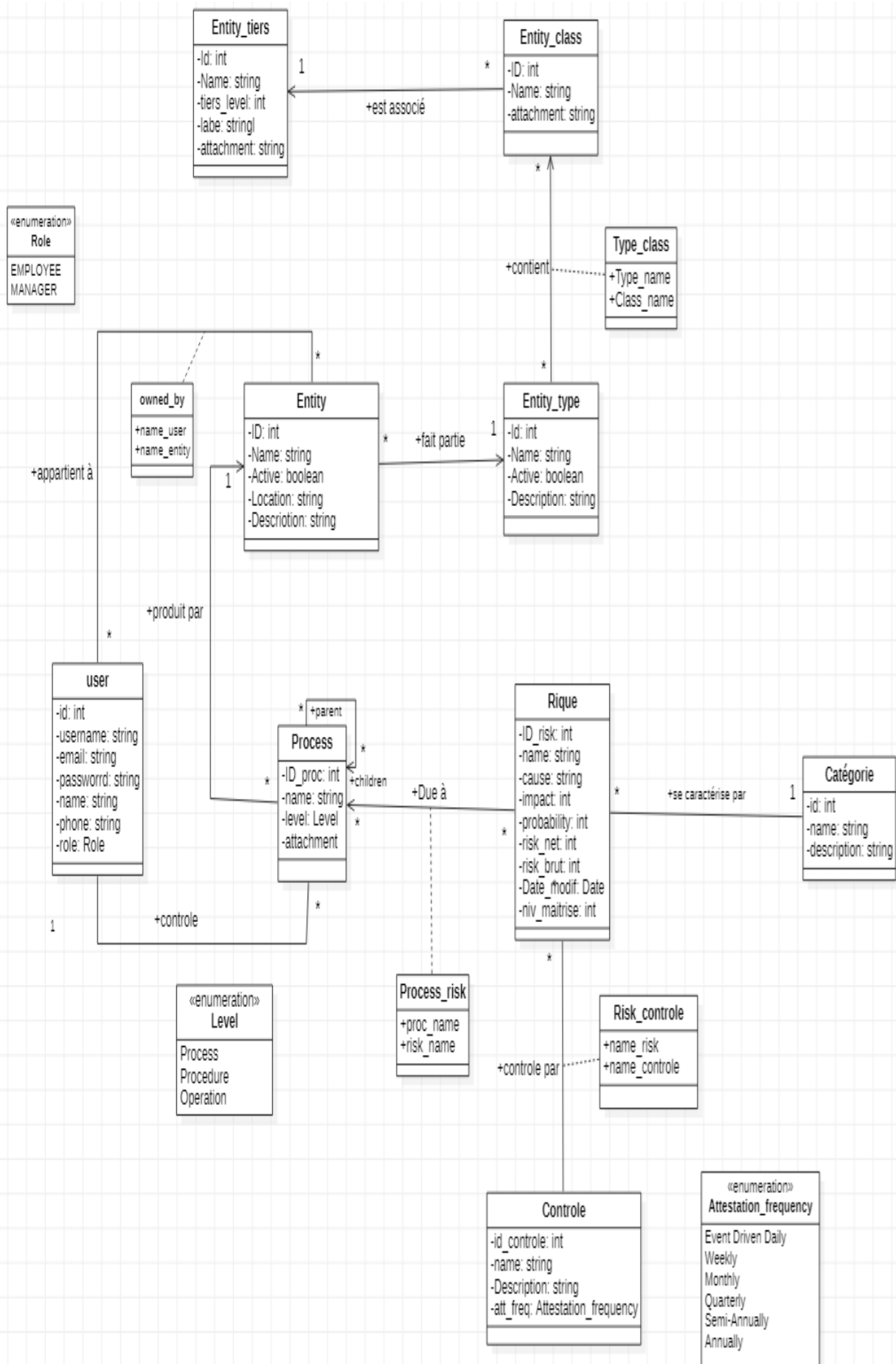
## 3.2 Conception détaillé

Le but de ce chapitre est de bien détaillé nos besoins pour permettre sa réalisation physique. Donc, pour atteindre ce but, nous avons présenté les diagrammes de séquences objet et le diagramme de classes détaillé.

### 3.2.1 Diagramme de classes

Le diagramme de classe est parmi les diagrammes UML les plus utiles et important dans la réalisation de n'importe quels projets parcequ' il décrit clairement la structure d'un système particulier, modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.[15]

La figure ci-dessous représente le diagramme de classes de notre projet :



**FIGURE 3.4 :** Diagramme de classe globale

Après avoir présenter le diagramme de classe globale, nous allons se concentrer sur la description et la définition des tables existants pour mieux le comprendre :

- **Process** : C'est l'ensemble des activités offertes et produites par les entités de l'entreprise.

*Exemples* : gestion des contrat, Reporting annuel...

- **Entity** : Un groupe dont l'objectif et la finalité sont d'offrir des services et d'activités de nature différents.

*Exemples* : Ressources humaines, Finance...

- **Entity\_type** : nous permet de regrouper les entités de même nature pour faciliter le travail.

- **Entity\_class** : un moyen de baliser des entités sur de nombreuses entity types.

- **Entity\_tiers** : C'est un niveau de hiérarchie que nous donnons aux classes pour donner une idée sur la hiérarchisation.

- **Risque** : Elle représente les risques opérationnels provenant de processus internes.

- **Catégorie** : Facilite la classification des risques opérationnels.

- **Contrôle** : Représente les contrôles appliqués aux risques pour minimiser les pertes.

### 3.2.2 Diagramme de séquence objet

Cette étape présente différents scénarios d'échange de messages entre acteurs et différentes classes du système. La partie suivante décrit ces interactions à l'aide de diagrammes de séquence.

#### 3.2.2.1 S'inscrire

La figure 3.5 représente le diagramme de séquence correspond à l'opération d'inscription de l'utilisateur.

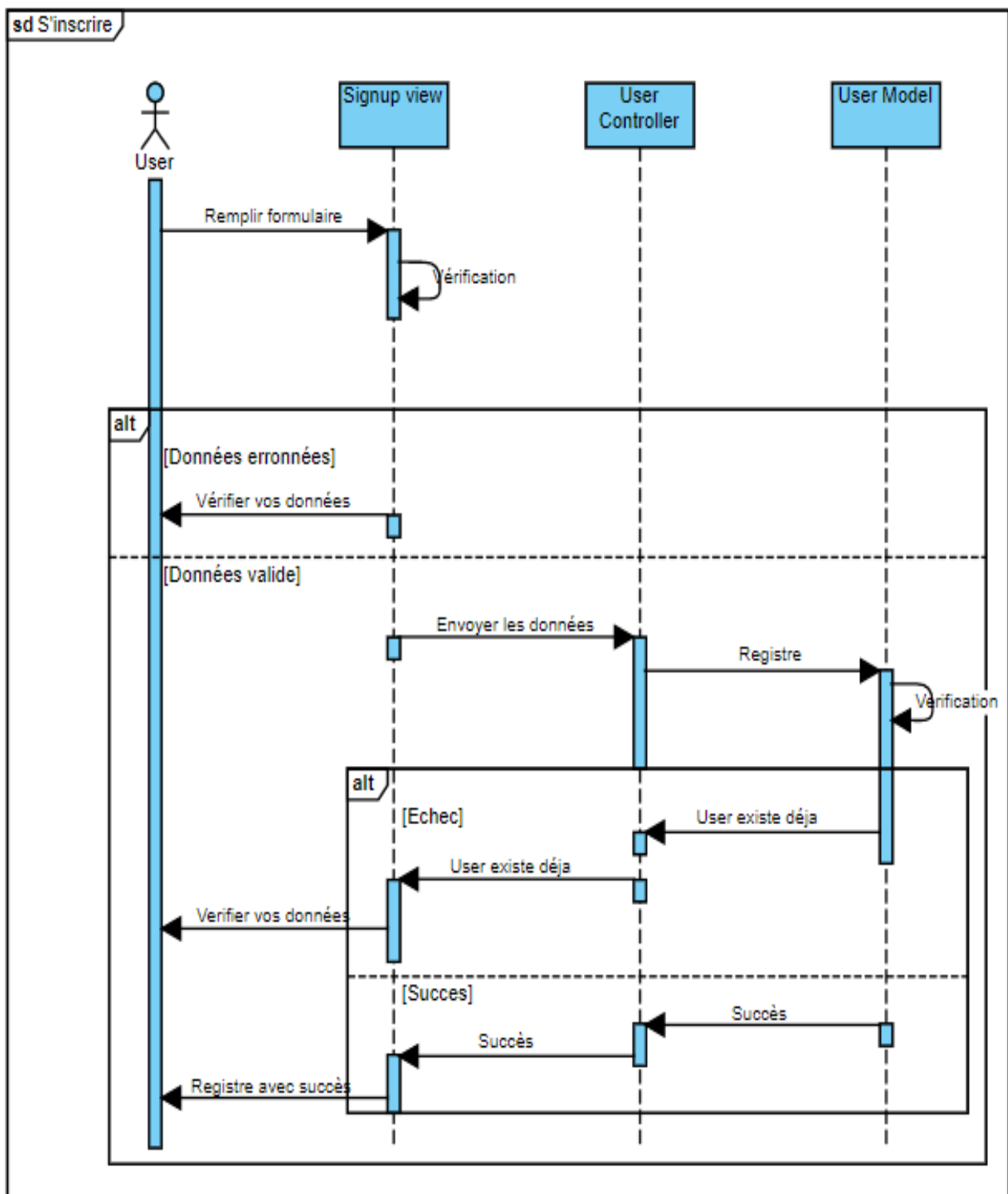


FIGURE 3.5 : Diagramme de séquence objet « S'inscrire »

### 3.2.2.2 S'authentifier

La figure 3.6 illustre le diagramme de séquence objet du CU « S'authentifier » .

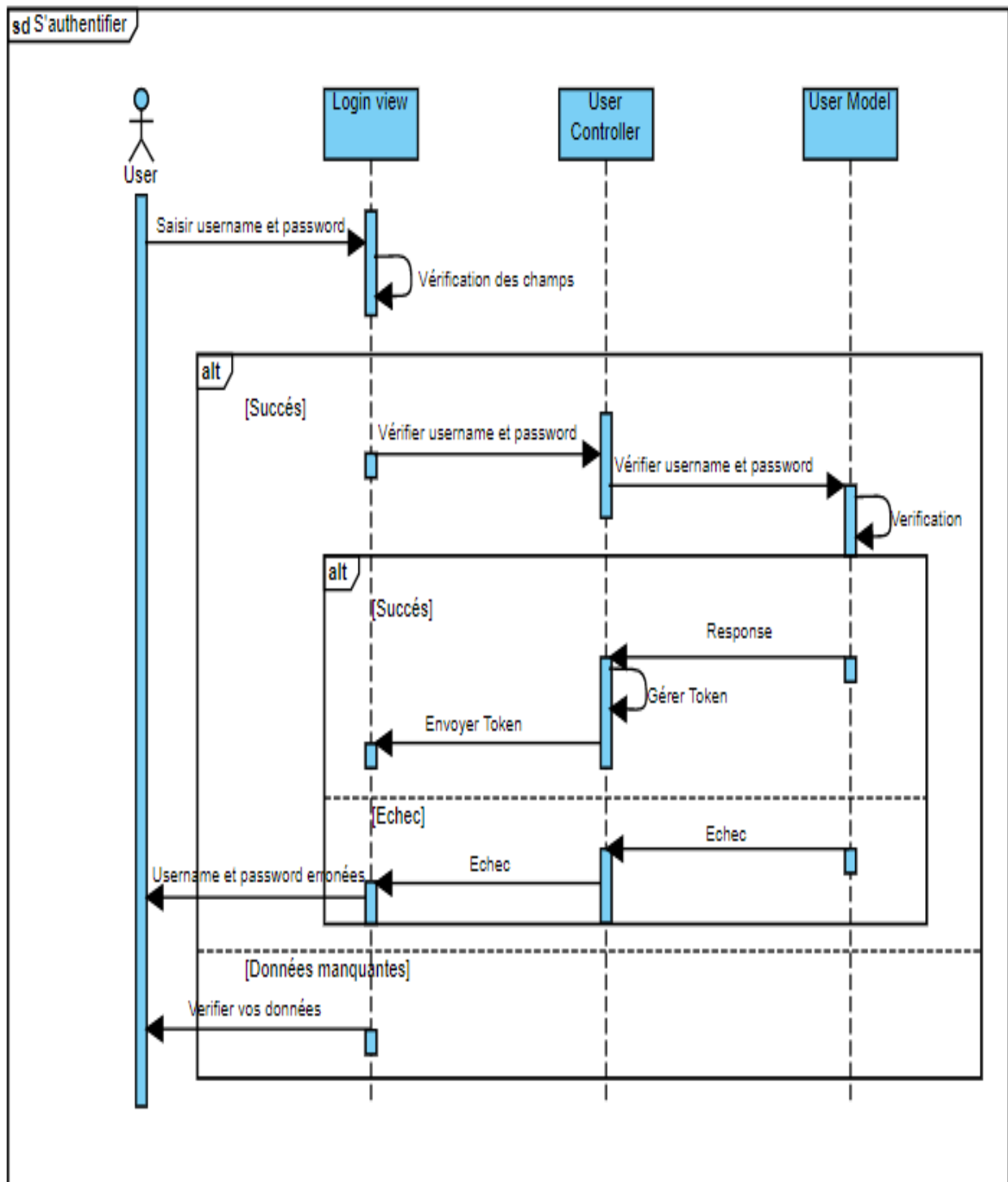


FIGURE 3.6 : Diagramme de séquence objet « S'authentifier »

### 3.2.2.3 Ajouter entity

La figure ci-dessous illustre le diagramme de séquence correspond à l'opération d'ajout d'une entity.



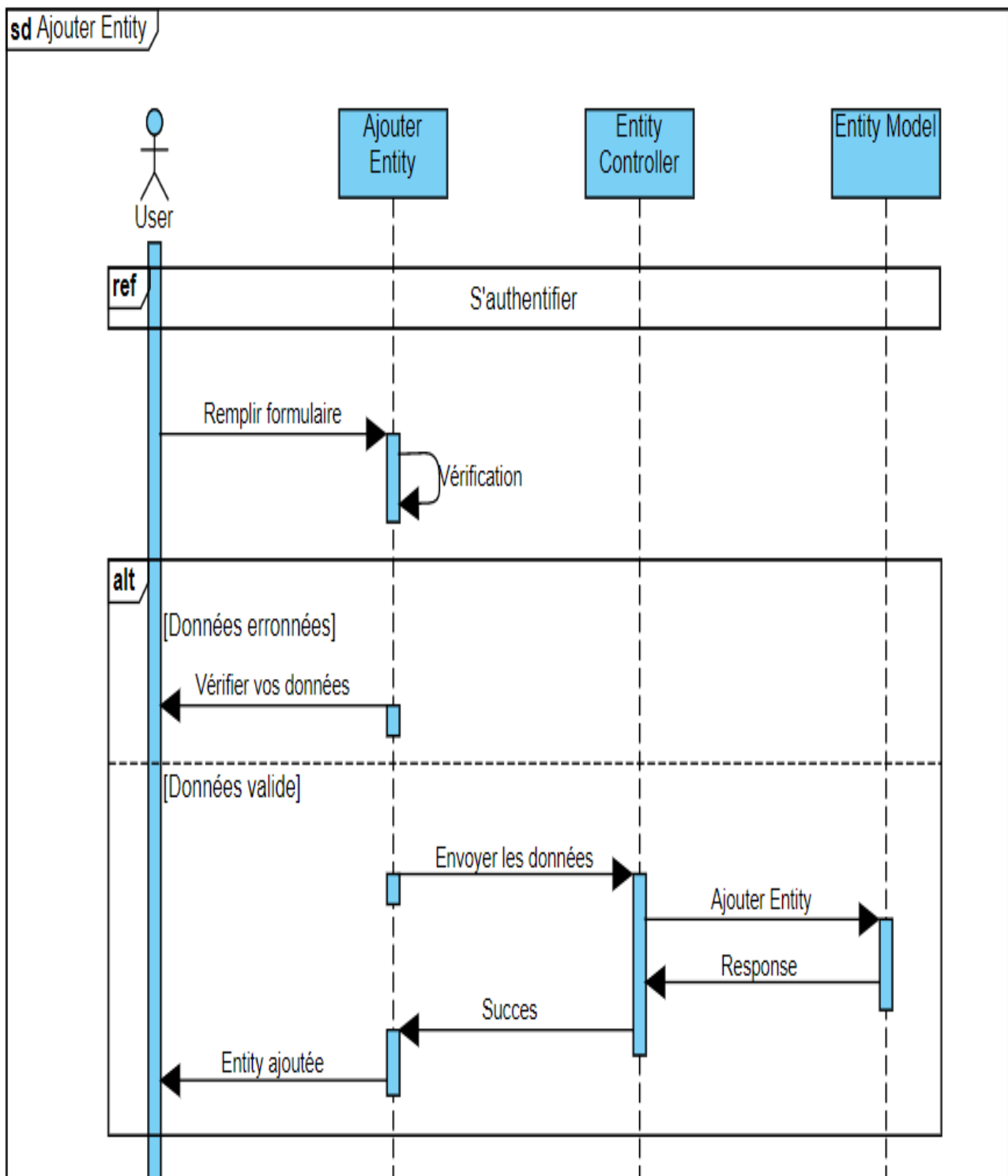


FIGURE 3.7 : Diagramme de séquence objet « Ajouter entity »

### 3.2.2.4 Ajouter risque

La figure 3.8 représente le diagramme de séquence objet de l'Ajout d'un risque.

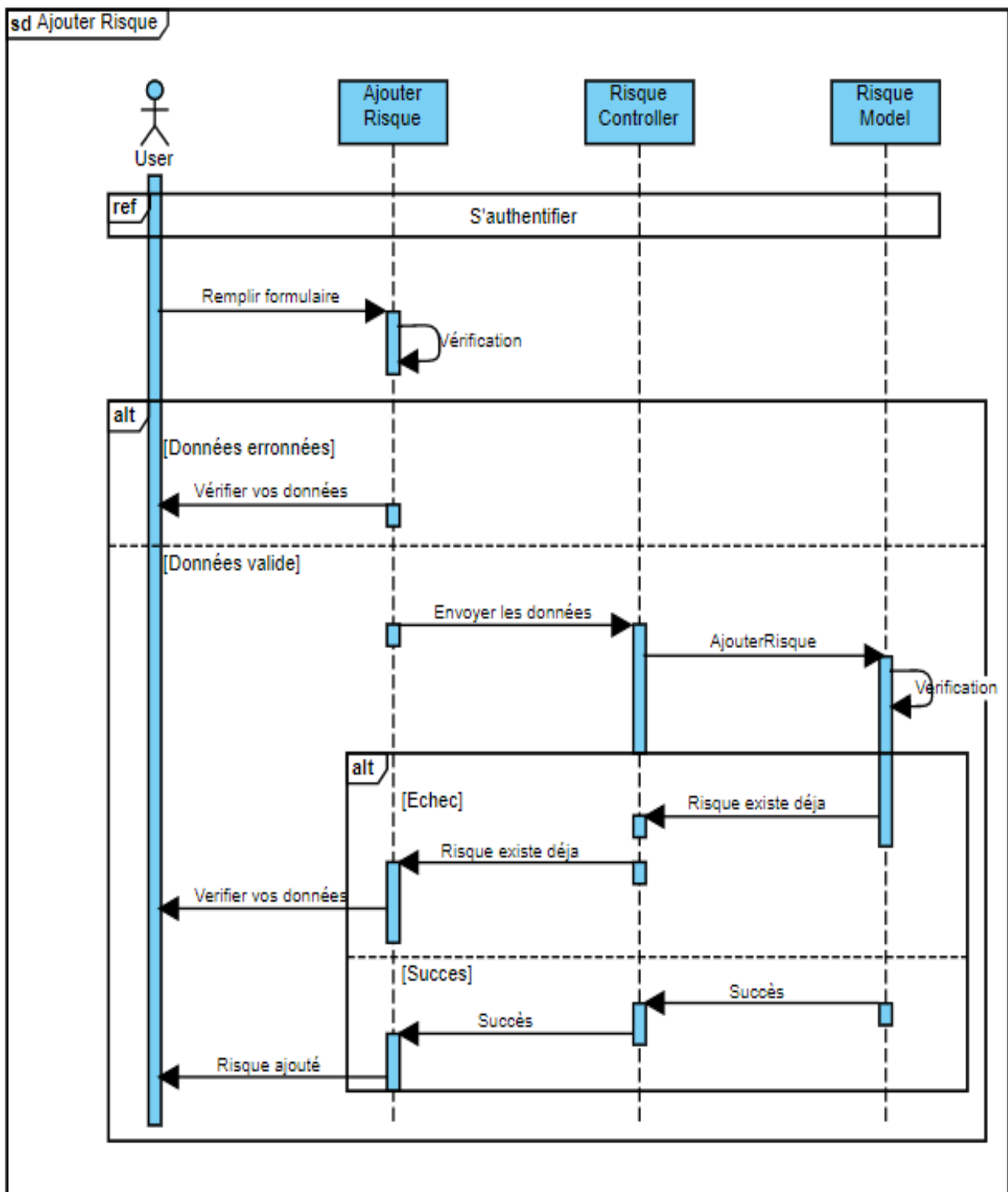


FIGURE 3.8 : Diagramme de séquence objet « Ajouter risque »

### 3.2.2.5 Modifier entity

La figure ci-dessous illustre le diagramme de séquence correspond à l'opération de modification d'une entity.

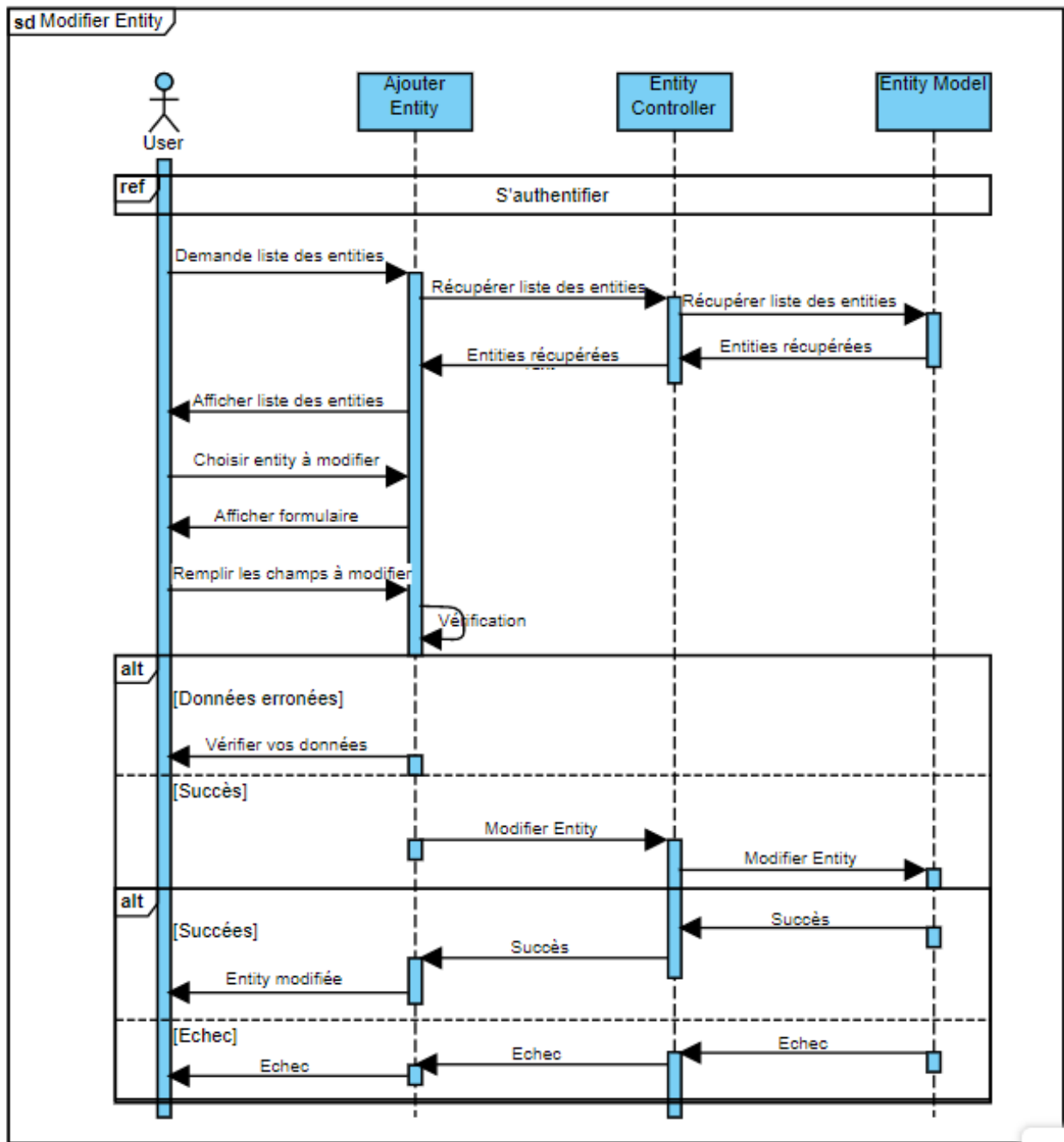


FIGURE 3.9 : Diagramme de séquence objet « Modifier entity »

### 3.2.2.6 Analyser risque

La figure 3.10 représente le diagramme de séquence objet de l'analyse d'un risque .

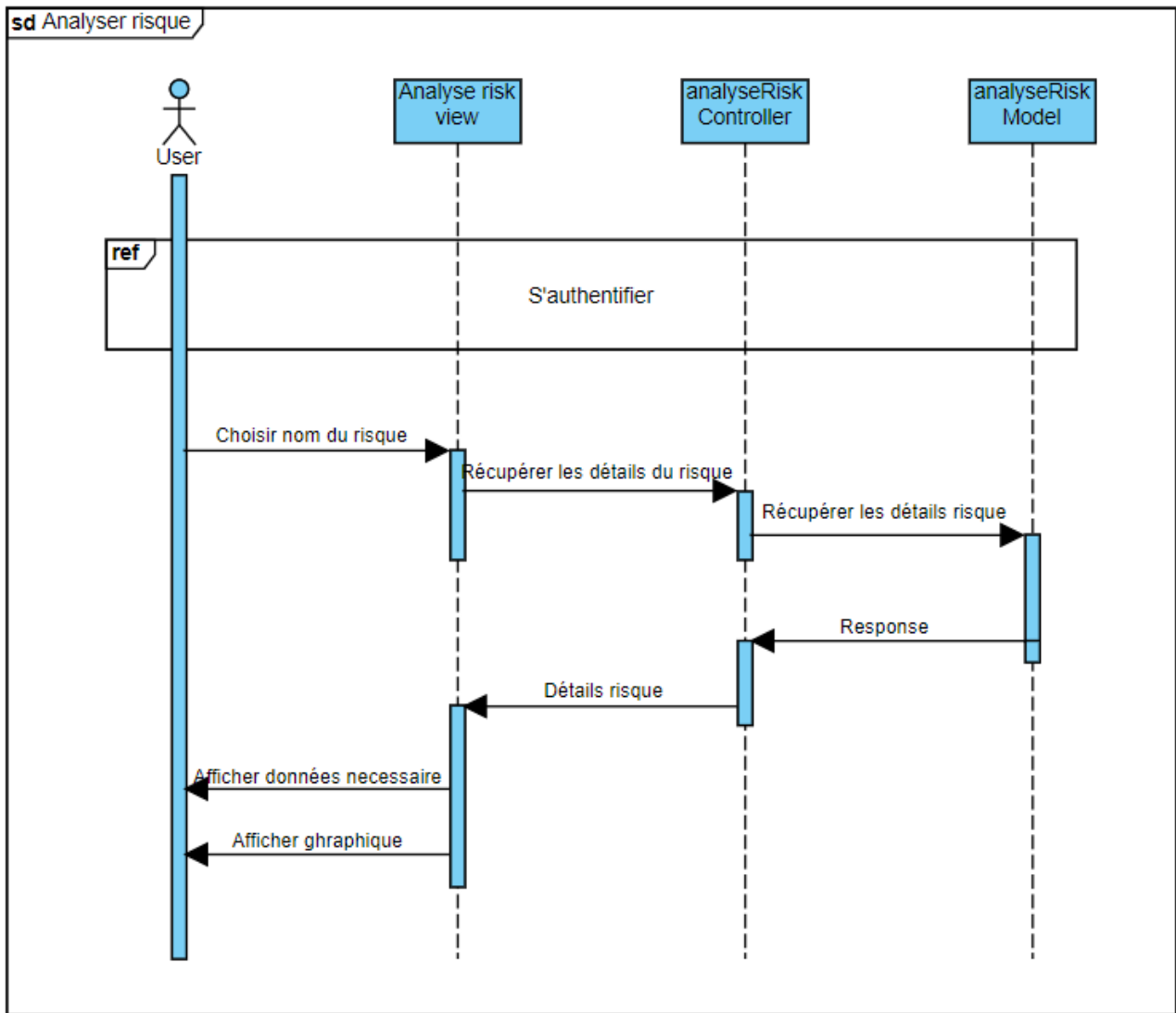


FIGURE 3.10 : Diagramme de séquence objet « Analyser risque »

### 3.2.3 Entrôpot de données

Un entrepôt de données est un référentiel central pour stocker des données structurées (tables de base de données, feuilles de calcul Excel) et des données semi-structurées (fichiers XML, pages Web) à des fins de reporting et d'analyse.

Dans notre cas, pour faciliter l'analyse des risques, nous avons stocké les données nécessaires provenant de la table Risque dans une autre table qui nous permet d'accéder facilement à une multitude de données historiques des risques, qui peuvent être utilisés pour l'exploration de données, la visualisation des données.

Donc, la figure ci-dessous présente la table extraite à partir de la table risque :

Risk analyse
- Name : String
- Risk_brut : Int
- Last_update : Date

FIGURE 3.11 : Table Risk analyse

### Méthode d'alimentation de Notre table :

L'extraction des données de la table de risque se fait en temps réel c'est-à-dire lors de l'opération d'ajout et de modification d'un risque, les données citées dans la table ci-dessus seront ajoutées automatiquement. Après cette étape, on choisit le nom du risque à analyser et on visualiser les données dans un diagramme.

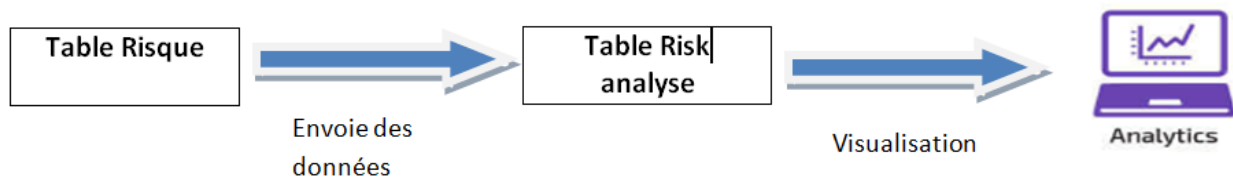


FIGURE 3.12 : Processus d'analyse d'un risque

## Conclusion

Dans ce chapitre, nous avons essayé de présenter l'architecture de l'application. De plus, avec différents schémas, le fonctionnement souhaité du système a été décrit de manière exhaustive et détaillée pour faciliter la mise en œuvre et la maintenance. Dans le chapitre suivant, nous allons présenter la partie de réalisation de notre application web.

---

# RÉALISATION

---

## Introduction

Dans ce dernier chapitre, on va clôturer notre projet en présentant l'environnement matérielles et logicielles qui ont permis son développement, ainsi que quelques captures d'écran qui montrent la réalisation du projet.

### 4.1 Environnement de travail

#### 4.1.1 Environnement matériel

Pour réaliser ce travail, l'ordinateur utilisé possède les caractéristiques suivantes :

- **Marque** : Lenovo
- **Processeur** : Intel(R) Core(TM) i5-8250U CPU
- **RAM** : 8,00 Go
- **Système d'exploitation** : Windows 10 Professionnel 64bits

#### 4.1.2 Environnement logiciel

- **Visual Studio Code**

Visual Studio Code ou VS Code est un éditeur de code open source développé par Microsoft en 2015 et disponible sur les systèmes d'exploitation Windows, Linux et Mac. Il est léger mais puissant, facile à installer, à comprendre et rapide aussi il offre un support intégré pour JavaScript, TypeScript et Node.js, et dispose d'un riche écosystème d'extensions pour d'autres langages et runtimes (C++, C#, Java, Python, PHP, Go, .NET, etc.).[16]

- **IntelliJ IDEA**

IntelliJ IDEA est un IDE intelligent et sensible au contexte pour travailler avec Java et

d'autres langages JVM comme Kotlin, Scala et Groovy sur toutes sortes d'applications. Il est développé par JetBrains en 2001 et disponible pour deux éditions : Community Edition et l'Ultimate Edition.[\[17\]](#)

- **Postman**

Postman est l'un des outils les plus populaires utilisés dans les tests d'API. Il a commencé en 2012 pour simplifier le flux de travail des APIs dans les tests et le développement.

- **PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle orienté objet hautes performances, open source, capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes.[\[18\]](#)

- **UML**

Pour l'analyse et la conception de notre projet, nous avons utilisé deux outils : Visual Paradigm Online et StarUML.

### 4.1.3 Technologies utilisées

- **Spring Boot** [\[19\]](#)

Spring Boot est un framework de développement Java. Il s'agit essentiellement d'une variante du framework Spring classique qui vous permet de créer des API Web (les API sont des programmes qui permettent à deux applications distinctes de communiquer entre elles et d'échanger des données). Selon votre complexité et votre configuration, Spring Boot peut être très utile pour réduire la complexité de la configuration des ressorts.

Spring boot nous propose plusieurs fonctionnalités :

- Dépendances de type starter pour simplifier la création et la configuration d'applications
- Serveur intégré pour éviter la complexité lors du déploiement des applications
- Métriques, vérification et configuration externalisée
- L'auto-configuration

- **Spring Security**[\[20\]](#)

Spring Security est un framework axé sur l'authentification et l'autorisation pour les applications Java. Comme pour tous les projets Spring, la véritable puissance de Spring

Security réside dans la facilité avec laquelle il peut être étendu pour répondre à des besoins personnalisés.

Avec spring security et à l'aide de JWT que nous avons assuré la sécurité de notre solution.

### Présentation de JWT [\[21\]](#)

JWT est une norme(RFC 7519) qui définit une solution compacte et autonome pour transférer en toute sécurité des informations entre les applications sous forme d'objets structurés au format JSON. Il présente une petite taille et peut être envoyé via des URL, des paramètres POST et des en-têtes HTTP, ce qui facilite sa transmission et la rendre rapide.

### Structure de JWT

JWT est constitué de 3 parties importantes séparés par un "." :

- *Header* : l'en-tête se compose de deux parties le type de jeton qui est JWT et l'algorithme utilisé pour le hashage dans notre cas c'est HS512.
- *Payload* : La partie charge utile du JWT contient les données réelles envoyées avec le jeton. Cette partie est également appelée la partie "claims" du jeton JWT. Les claims peuvent être enregistrer, publique ou privée.
- *Signature* : La partie signature du JWT est utilisée pour vérifier que le message n'a pas été modifié en transit.

- **Angular13** [\[22\]](#)

Angular est une plate-forme et un framework permettant de créer des applications clientes d'une seule page à l'aide de HTML et de TypeScript. Angular est écrit en TypeScript. Implémente les fonctionnalités de base et facultatives sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans votre application. Il offre de nombreuses commodités modernes telles que la liaison de données, une intégration tierce facile et globalement moins de codage pour les développeurs.

Pour faire une visualisation de données dans un graphique, Angular propose un composant angular open source qui s'appelle **HighChart Angular** qui fournit une visualisation Highcharts élégante et riche en fonctionnalités dans une application Angular et peut être utilisée de manière transparente avec les composants Angular.



## 4.2 Réalisation

Dans cette partie, nous avons présenté quelques interfaces de notre application.

### 4.2.1 Interface d'authentification

Cette figure représente la première fenêtre de notre solution qui se charge par défaut. Donc, L'utilisateur essaye de se connecter en saisissant son username et son mot de passe. Si, les données entrées sont incorrectes un message d'erreur s'affiche sinon, l'utilisateur sera rediriger vers la page précédente.

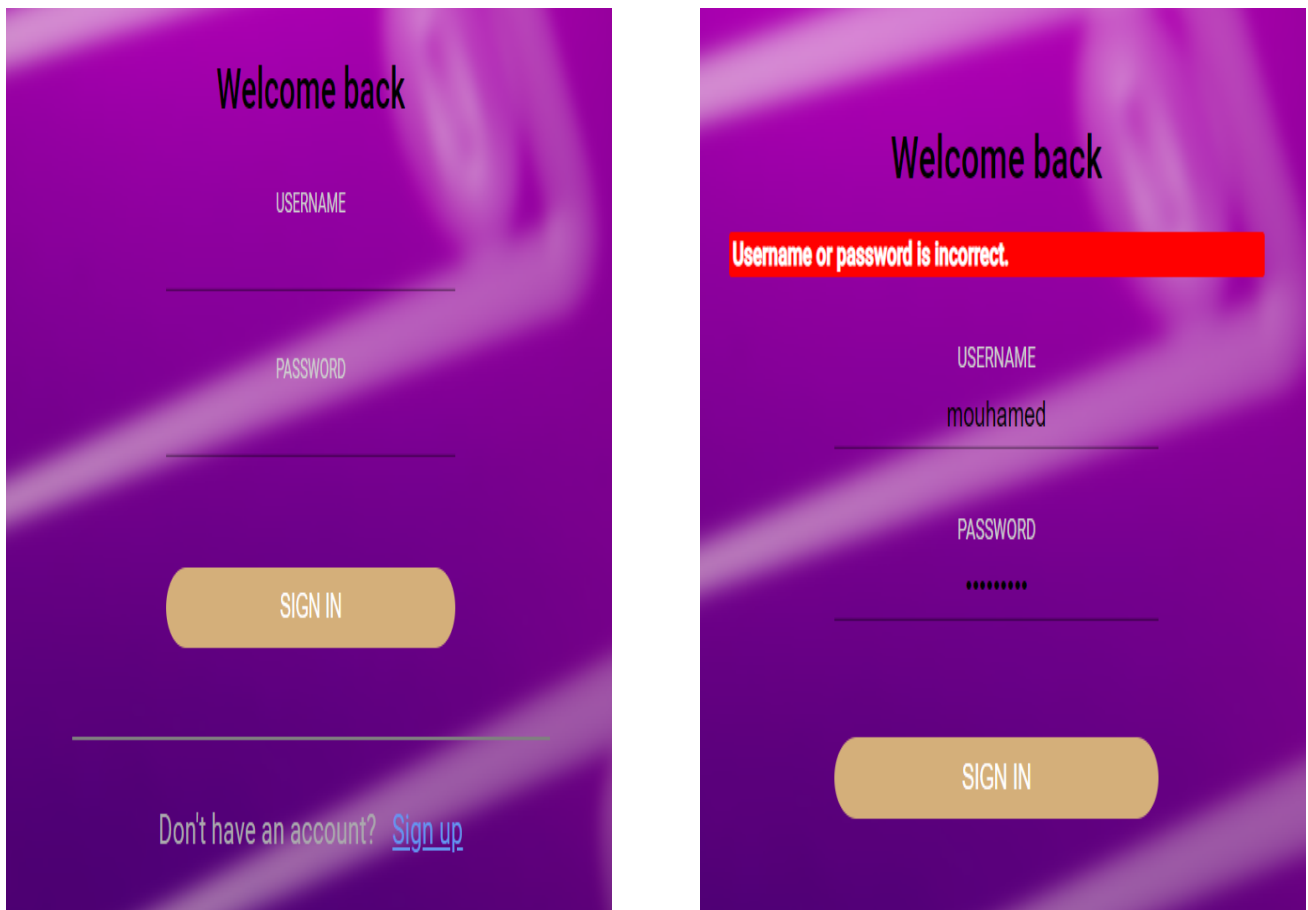


FIGURE 4.1 : Interface d'authentification

### 4.2.2 Interface d'inscription

Si un utilisateur veut se connecter mais il n'a pas encore un compte donc il se dirige vers la page de Sign Up. Dans cette page, chaque utilisateur doit saisir son nom et prénom dans le champs name, son username, son email, son numéro de téléphone, son mot de passe ainsi

qu'un code donnant par l'administration. Ce code nous permet de connaître rôle de chaque utilisateur. Si, un des données est incorrecte ou manquant un message d'erreur s'affiche sinon il revient à la page de Login.

The figure displays two states of a registration interface titled "Join Us!".

**Left Screenshot (Empty Form):**

- Fields: FULL NAME, USERNAME, EMAIL, PHONE, PASSWORD, CODE.
- Buttons: SIGN UP.
- Footer: NB\*: Default value of code : EMPLOYEE. If you already has an account just [Sign in!](#)

**Right Screenshot (Form with Data and Error):**

- Fields: FULL NAME (Mouhamed), USERNAME (mouhamed), EMAIL (empty), PHONE (23787323), PASSWORD (masked with dots), CODE (masked with dots).
- Buttons: SIGN UP.
- Error Message: An valid email is required. (highlighted in red).

FIGURE 4.2 : Interface d'inscription

### 4.2.3 Interface de gestion des entités tiers

Cette partie contient les deux interfaces d'ajout et la liste des entités tiers :

- **Interface Liste des entités tiers :**

Après l'authentification, chaque utilisateur a le droit de consulter la liste des des entités tiers en cliquant sur Réfrentiel puis entity puis entity tier et cette figure ci-dessous sera affichée :

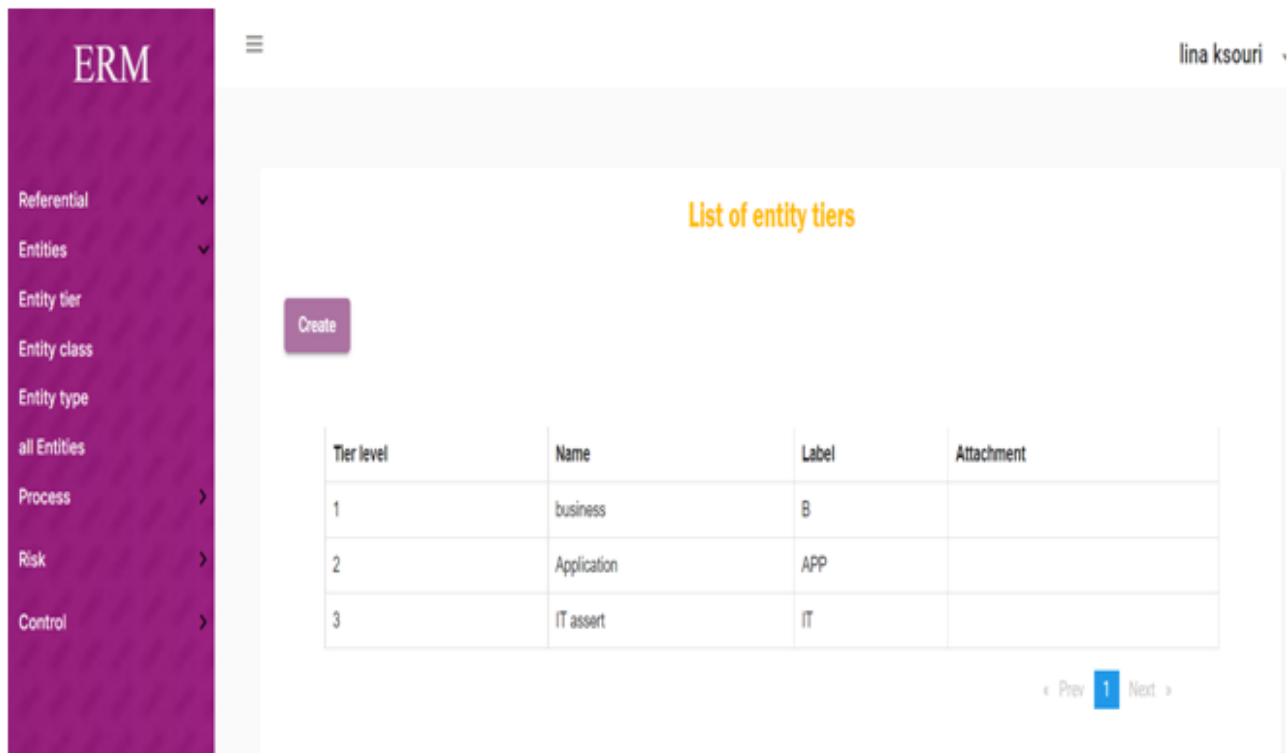


FIGURE 4.3 : Interface de lise des entités tiers

- **Interface d'ajout d'une nouvelle entité tier :**

Pour hiérarchiser les compagnies, l'utilisateur peut ajouter une nouvelle entité tier en remplissant le formulaire par le nom, libellé et le niveau d'hérarchisation de chaque entité.

Name

Label

Tiers Level

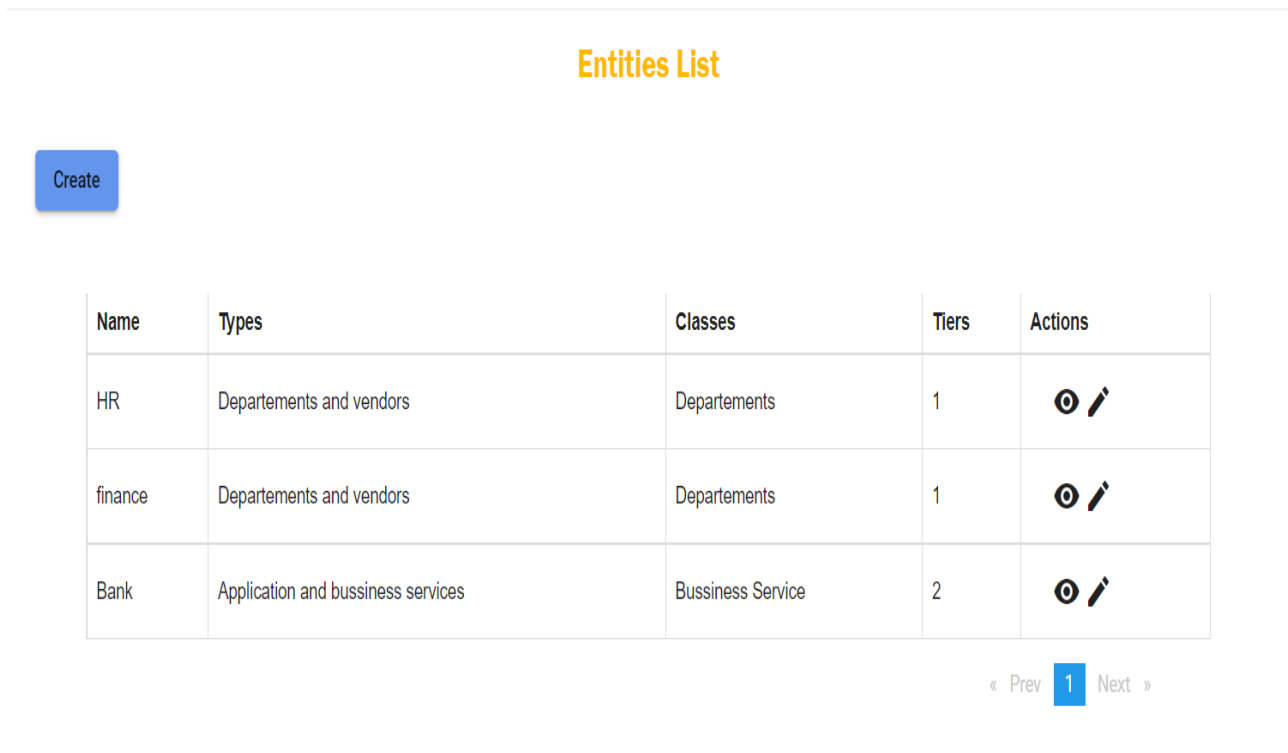
Attachment  Aucun fichier choisi







FIGURE 4.4 : Interface d'ajout d'une entité tier

### 4.2.4 Interface de gestion des entités

Dans cette partie, chaque utilisateur peut consulter liste des entité, ajouter, modifier et consulter détailles de chaque entité.

- **Interface Liste des entités :**



Name	Types	Classes	Tiers	Actions
HR	Departements and vendors	Departements	1	 
finance	Departements and vendors	Departements	1	 
Bank	Application and bussiness services	Bussiness Service	2	 

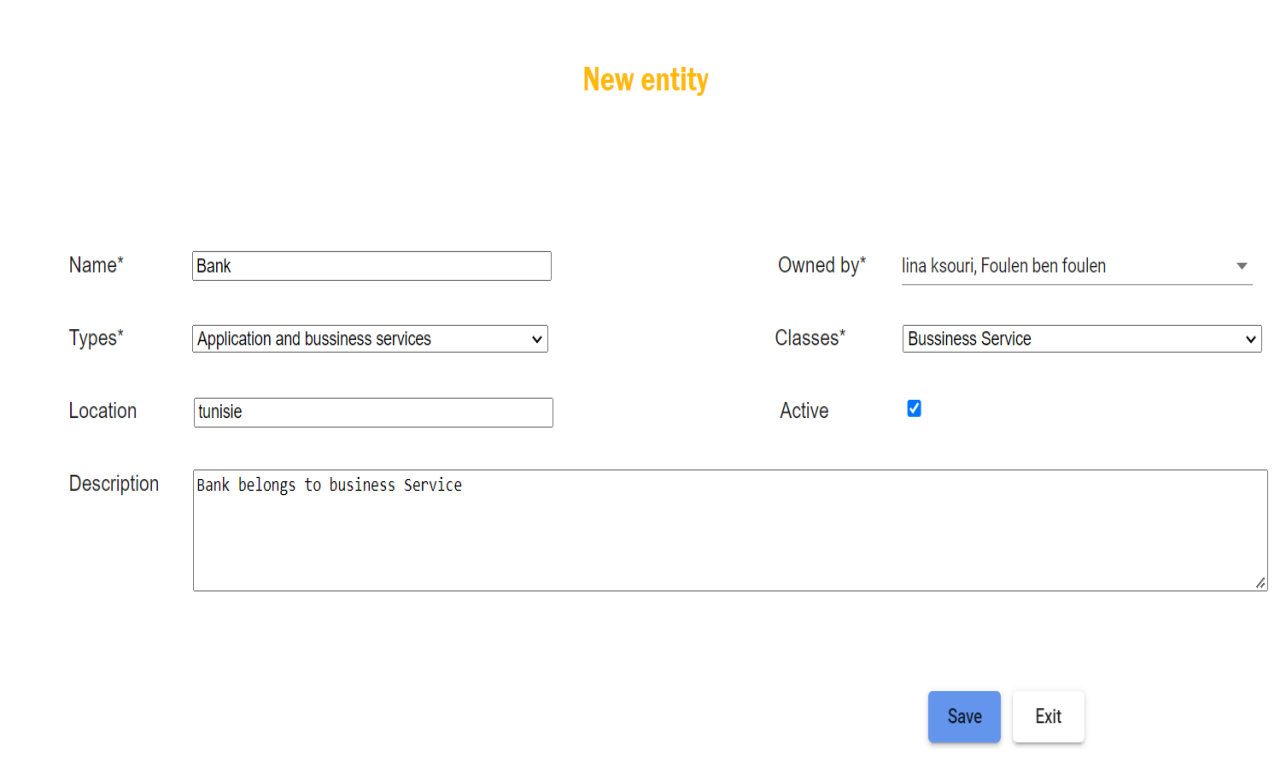
« Prev 1 Next »

**FIGURE 4.5 :** Interface de lise des entités

- **Interface d'ajout d'une nouvelle entité :**

Chaque entité se caractérise par son nom, le type qui regroupe les entités de même nature, son classe où elle appartient et d'autres champs.

Donc, la figure ci-dessous présente le formulaire d'ajout d'un entité :



**New entity**

Name\*

Owned by\*

Types\*

Classes\*

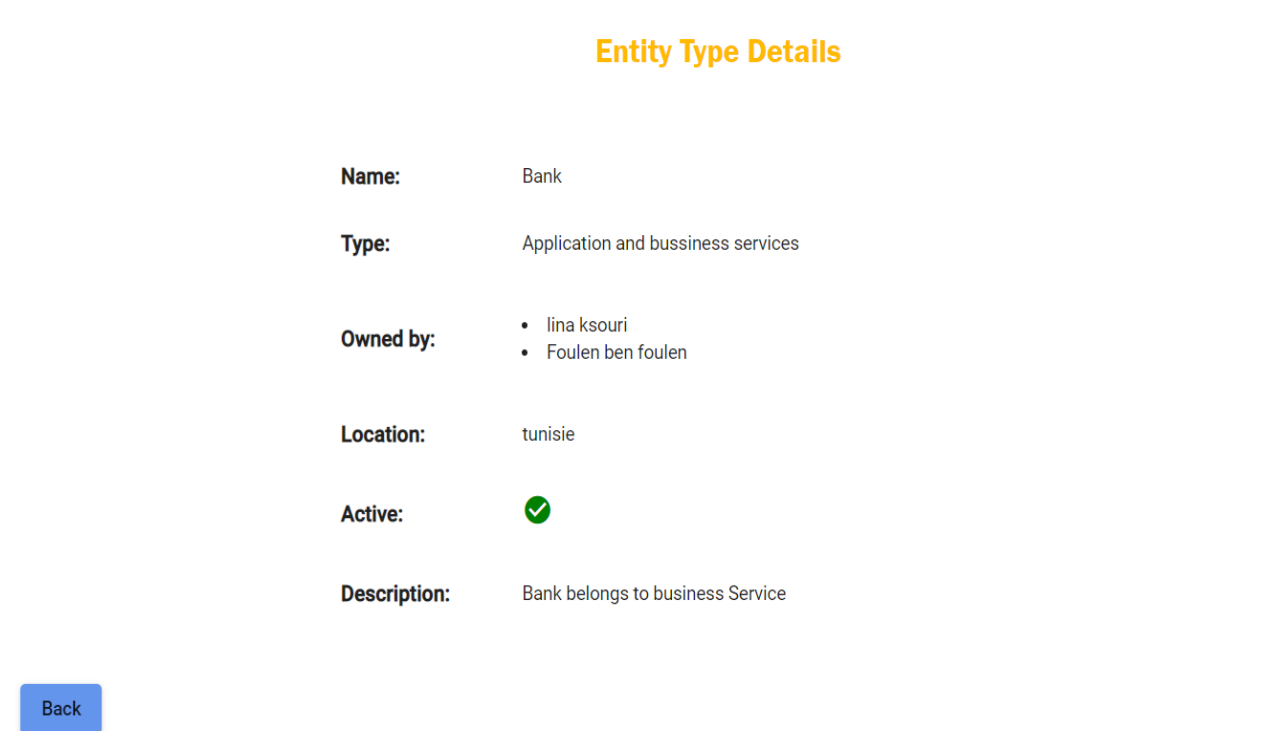
Location

Active ☒

Description

FIGURE 4.6 : Interface d'ajout d'une entité

- Interface d'affichage des détaillés d'une entité :



**Entity Type Details**

Name: Bank

Type: Application and bussiness services

Owned by:

- lina ksouri
- Foulen ben foulen

Location: tunisie

Active: ☒

Description: Bank belongs to business Service

FIGURE 4.7 : Interface d'affichage des détaillés d'une entité

### 4.2.5 Interface de gestion des risques

Après avoir fait le registre des données qui caractérisent chaque compagnie d'assurance ou de banque et les activités de chaque entité existantes, nous passons à l'étape de gestion de risque opérationnel qui nous permet d'identifier et d'analyser chaque risque.

- **Interface Liste des risques :**

Name	Cause	Category	Risk brut	Actions
inexacte information	Les informations sont incorrectes	Fraude interne	9	
risk1		Clients/tiers, produits et pratiques commerciales	20	

« Prev 1 Next »

**FIGURE 4.8 :** Interface de lise des risques

- **Interface d'ajout d'un nouveau risque :**

Pour ajouter un nouveau risque, l'utilisateur doit remplir le formulaire suivant en choisissant son nom, sa catégorie, les processus qui peuvent produire ce risque ainsi son impact et sa probabilité qui nous aident à calculer le risque brut aussi il est important d'ajouter la date de la dernière modification.

Cet ajout permet l'ajout automatique d'une ligne dans la table risk\_analyse pour faciliter l'analyse.

### New risk

Name*	<input type="text" value="inexacte information"/>	Category*	<input type="text" value="Fraude interne"/>
Severity*	<input type="text" value="Moderate"/>	Likelihood*	<input type="text" value="Likely"/>
Process*	<input type="text" value="facturation client"/>	Last update*	<input type="text" value="22/09/2022"/>
Cause	<input type="text" value="Les informations sont incorrectes"/>		

SaveExit

FIGURE 4.9 : Interface d'ajout d'un risque

- Interface d'analyse d'un risque :

Après l'opération d'ajout des risques, nous avons passé à l'étape d'analyse et d'évaluation d'un risque pour nous permettre de prendre une décision.

Donc, d'abord, l'employé choisit un nom parmi la liste des noms des risques affichée puis le système affiche la dernière valeur du risque brut et un graphe qui étudie l'évaluation de ce risque durant cette année. La figure ci-dessous montre le résultat affiché par notre système :

### Analyse and Evaluation of risk

Name:

Evaluate

Your gross risk : 9 ==> Moderate

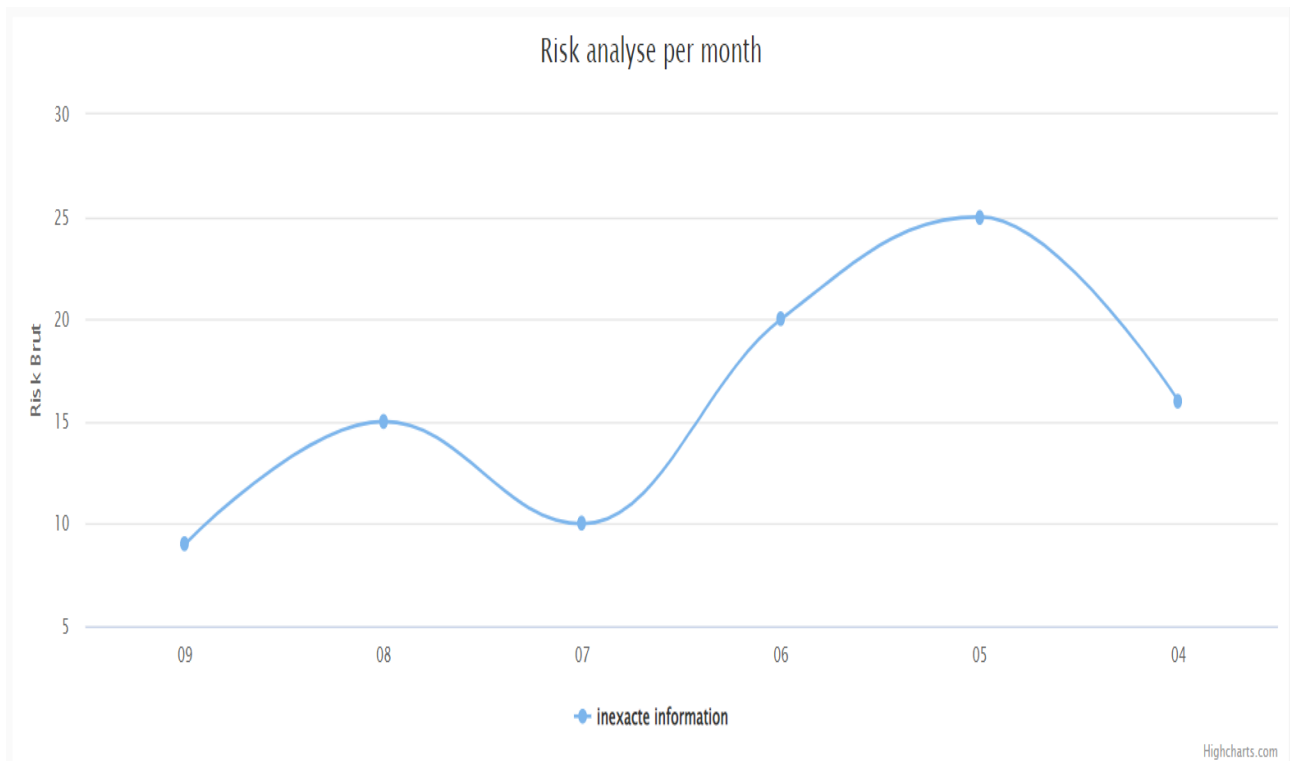


FIGURE 4.10 : Interface d'analyse d'un risque

#### 4.2.6 Interface de gestion de contrôle

Après l'identification et l'analyse des risques, maintenant c'est le temps de choisir le type de contrôle pour chaque risque.

- **Interface de modification d'un contrôle :**

Chaque utilisateur consulte la liste des contrôles et choisit le contrôle à modifier. Cette opération engendre la modification du risque à contrôler puisque l'utilisateur choisit le niveau de maîtrise de ce contrôle pour ce risque et parfois il modifie l'impact et la probabilité qui implique le recalcul de risque brut et le calcul de risque net.



The screenshot shows the 'Update Control' interface in the ERM application. On the left is a purple sidebar with 'ERM' at the top and a menu with 'Referential', 'Risk', and 'Control'. The main area has a title 'Update Control' in orange. Below it is a form with two columns of fields:

- Name\***: Risk Reduction
- Risk name**: inexacte information
- User name**: Foulen ben foulen
- Attestation frequency**: Semi-Annually
- Severity\***: 2
- Likelihood\***: 1
- Mastery level**: Standardized
- Last update\***: 23/09/2022

At the bottom right are 'Submit' and 'Exit' buttons.

FIGURE 4.11 : Interface de modification d'un contrôle

Les deux figures 4.12 et 4.13 montrent le risque avant et après le contrôle :

The screenshot shows a REST client interface with the following details:

- URL**: localhost:8090/risk/inexacte information
- Method**: GET
- Authorization**: Bearer Token (Token: eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJsaWJsaW9u...)
- Status**: 200 OK, Time: 1517 ms, Size: 708 B
- Response Body (JSON)**:
 

```

      {
        "id_risk": 1,
        "name": "inexacte information",
        "cause": "Les informations sont incorrectes",
        "impact": 3,
        "probability": 2,
        "niv_maitrise": 0,
        "risk_net": 0,
        "risk_brut": 6,
        "Last_update": "2022-09-22",
        "process": [
          "facturation client"
        ],
        "category": "Fraude interne",
        "last_update": "2022-09-22"
      }
      
```

FIGURE 4.12 : Risque avant contrôle

#### 4.2.7 Interface de gestion d'utilisateur

- Interface de supprimer un employé :

The screenshot displays a web application interface. At the top, a purple button labeled "Create" is visible. Below it, a table lists employees. The table has columns for "Name", "Phone", and "Actions". The first three rows of the table are partially obscured by a white confirmation dialog box. The dialog box contains an orange exclamation mark icon, the text "Are you sure?", the text "You won't be able to revert this!", and two buttons: "Yes, delete it!" (blue) and "Cancel" (red). The fourth row of the table is visible below the dialog box, showing the employee "Foulen ben foulén" with phone number "foulen@gmail.com" and role "EMPLOYEE". The "Actions" column for this row shows edit and delete icons. At the bottom right of the page, there are navigation links: "« Prev", "1" (highlighted in a blue box), "Next »".

Name	Phone	Actions
lina ksouri	23787323	
linaksouri	23787323	
Mouhamed	23787323	
Foulen ben foulén	foulen@gmail.com	EMPLOYEE

53

Si le Manager choisit de confirmer cette opération, l'employé sera supprimé et n'est plus affiché dans la liste des personnels.

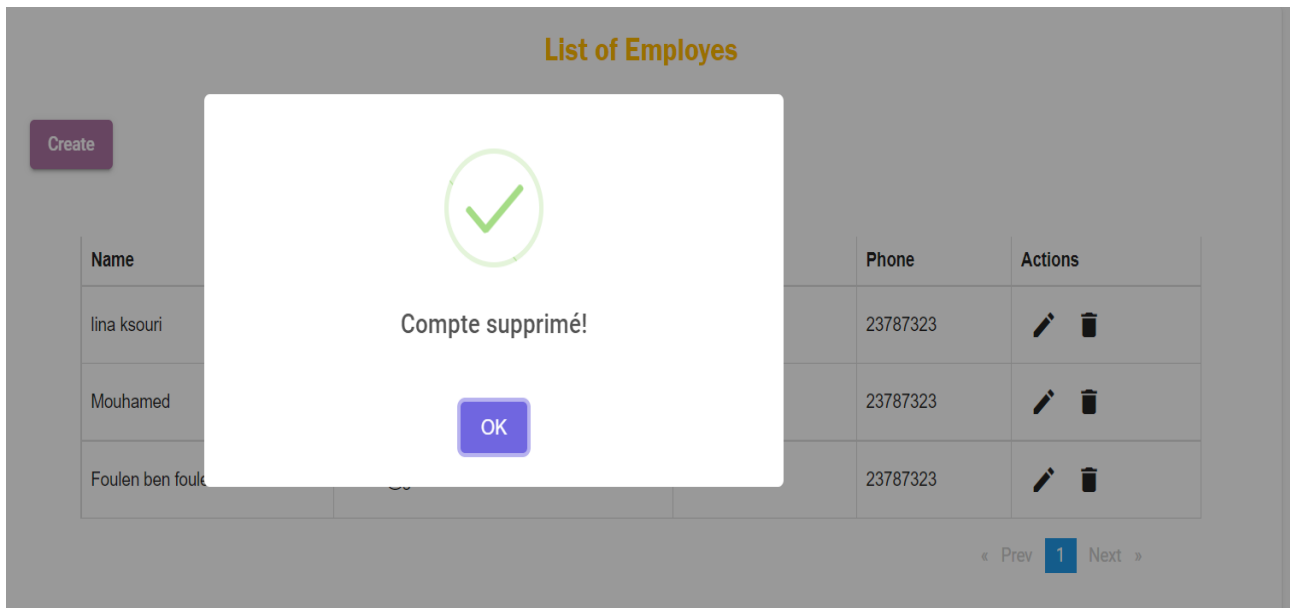


FIGURE 4.15 : Interface 2 de la suppression

## Conclusion

Dans ce chapitre, nous avons exploré tous les outils et environnements de travail pour la réalisation l'application et également nous avons présenté l'application elle-même avec une sélection d'interfaces développées.

# Conclusion générale et perspectives

Notre projet de fin d'études réalisé chez Vneuron a conduit à la mise en place d'une solution de gestion de risque. Cette solution consiste à développer une application web pour présenter une cartographie des risques opérationnels qui permettent de définir le démarche d'identification,évaluation et contrôle .

Pour arriver à ce résultat, nous avons détaillé toutes les phases qui composent le cycle de vie d'un logiciel. Dans un premier temps, nous avons commencé à collecter les données nécessaires pour comprendre le sujet et examiner le problème pour trouver une solution. Après cela, nous avons effectué une analyse des besoins et une conception détaillée à l'aide de diagrammes UML. Enfin, pour le reste du travail, nous avons implémenté le projet en utilisant le framework SPRING BOOT côté serveur et Angular 13 côté client.

Toute au long de ce projet, j'ai eu l'opportunité d'appliquer et d'améliorer mes connaissances théoriques acquises dans mon cursus académique. C'est aussi l'occasion de découvrir le monde professionnel. En effet, c'est une expérience enrichissante dans les domaines de développement et de management des risques.

Certe ce travail répond à des objectifs précédemment établis, mais notre vision est d'enrichir la solution pour couvrir de nouvelles fonctionnalités telles que la gestion des alertes et la création des rapports d'analyse et d'évaluation.

# Bibliographie

- [1] ERM. [Consulté Août 2022]. URL : <https://riskconnect.com/fr/resources/erm-enterprise-risk-management-guide/>.
- [2] VNEURON. *Site officiel*. [Consulté Juillet 2022]. URL : <https://www.vneuron.com/compliance/company/>.
- [3] PARTENAIRES. [Consulté Juillet 2022]. URL : <https://opex.vneuron.com/>.
- [4] ERM vs TRM. [consulté Septembre 2022]. URL : <https://www.ideagen.com/thought-leadership/blog/traditional-risk-management-vs-enterprise-risk-management-which-approach-is-best>.
- [5] ERM PROCESS. [Consulté Août 2022]. URL : <https://www.nexigroup.com/en/sustainability/enterprise-risk-management/ERM-process/>.
- [6] AGILE. [Consulté Septembre 2022]. URL : <https://www.atlassian.com/fr/agile>.
- [7] Méthodologie SCRUM. [Consulté Septembre 2022]. URL : <https://www.bocasay.com/fr/methode-scrum-benefices-developpements-web/>.
- [8] Les événements SCRUM. [Consulté Septembre 2022]. URL : <https://www.unow.fr/blog/le-coin-des-experts/les-cinq-evenements-de-scrum/>.
- [9] Backlog PRODUIT. [Consulté Septembre 2022]. URL : <https://www.qrpinternational.fr/blog/glossaire/quest-ce-quun-backlog-definition-etapes-caracteristiques-et-outils/>.
- [10] Diagramme de cas D'UTILISATION. [Consulté Août 2022]. URL : <https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case>.
- [11] Diagramme de SÉQUENCES. [Consulté Août 2022]. URL : <https://www.ibm.com/docs/fr/rsm/7.5.0?topic=uml-sequence-diagrams>.
- [12] Architecture SYSTÈME. [Consulté Septembre 2022]. URL : <https://sokeo.fr/architecture-application-web/>.
- [13] MVC. [Consulté Septembre 2022]. URL : <https://www.javaguides.net/2020/07/three-tier-three-layer-architecture-in-spring-mvc-web-application.html>.
- [14] Architecture globale de PROJET. [Consulté Septembre 2022]. URL : <https://frontbackend.com/spring-boot/angular-11-spring-boot-2-postgresql>.

- [15] Diagramme de CLASSE. [Consulté Septembre 2022]. URL : <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml>.
- [16] VS CODE. [Consulté Septembre 2022]. URL : <https://code.visualstudio.com/docs>.
- [17] INTELLIJ. [Consulté Septembre 2022]. URL : <https://www.jetbrains.com/idea/features/>.
- [18] POSTGRESQL. [Consulté Septembre 2022]. URL : <https://www.oracle.com/fr/database/definition-postgresql.html>.
- [19] Spring BOOT. [Consulté Septembre 2022]. URL : <https://waytolearnx.com/2019/04/difference-entre-spring-et-spring-boot.html>.
- [20] Spring SECURITY. [Consulté Septembre 2022]. URL : <https://spring.io/projects/spring-security>.
- [21] JWT. [Consulté Septembre 2022]. URL : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/json-web-token-jwt/>.
- [22] ANGULAR. [Consulté Septembre 2022]. URL : <https://angular.io/guide/architecture>.

# Annexes

## Annexe 1. Dictionnaire des données

Le tableau annexe 1.1 présente la définition des attributs de la table entity\_tiers.

Field	Description
ID	Identifiant de l'entité tier
Name	Le nom du niveau hiérarchie
Label	Libellé du niveau d'entité
Tier_level	Le niveau hiérarchie
Attachement	Document

Tableau annexe 1.1 : Table entity\_tier

Le tableau annexe 1.2 présente la description des attributs de la table user

Field	Description
ID	Identifiant de l'utilisateur
Username	Nom de l'utilisateur
Email	Email de l'utilisateur
Name	Nom et prénom de l'utilisateur
Password	Mot de passe de l'utilisateur
Phone	Numéro de téléphone
Role	Rôle de l'utilisateur qui peut être soit Manager, Employé

Tableau annexe 1.2 : Table user

Le tableau annexe 1.3 présente la description des attributs de la table risque

Field	Description
Id_risk	Identifiant du risque
Name	Nom du risque
Cause	Cause du risque
Impact	Impact de risque
Probability	La probabilité qu'un risque sera <u>produit</u>
Risk_net	Risque net après le contrôle (à calculer)
Risk_brut	Risque brut sans contrôle (à calculer)
Date_modif	Date de la dernière modification du risque
Niv_maitrise	Niveau de maitrise

Tableau annexe 1.2 : Table risque

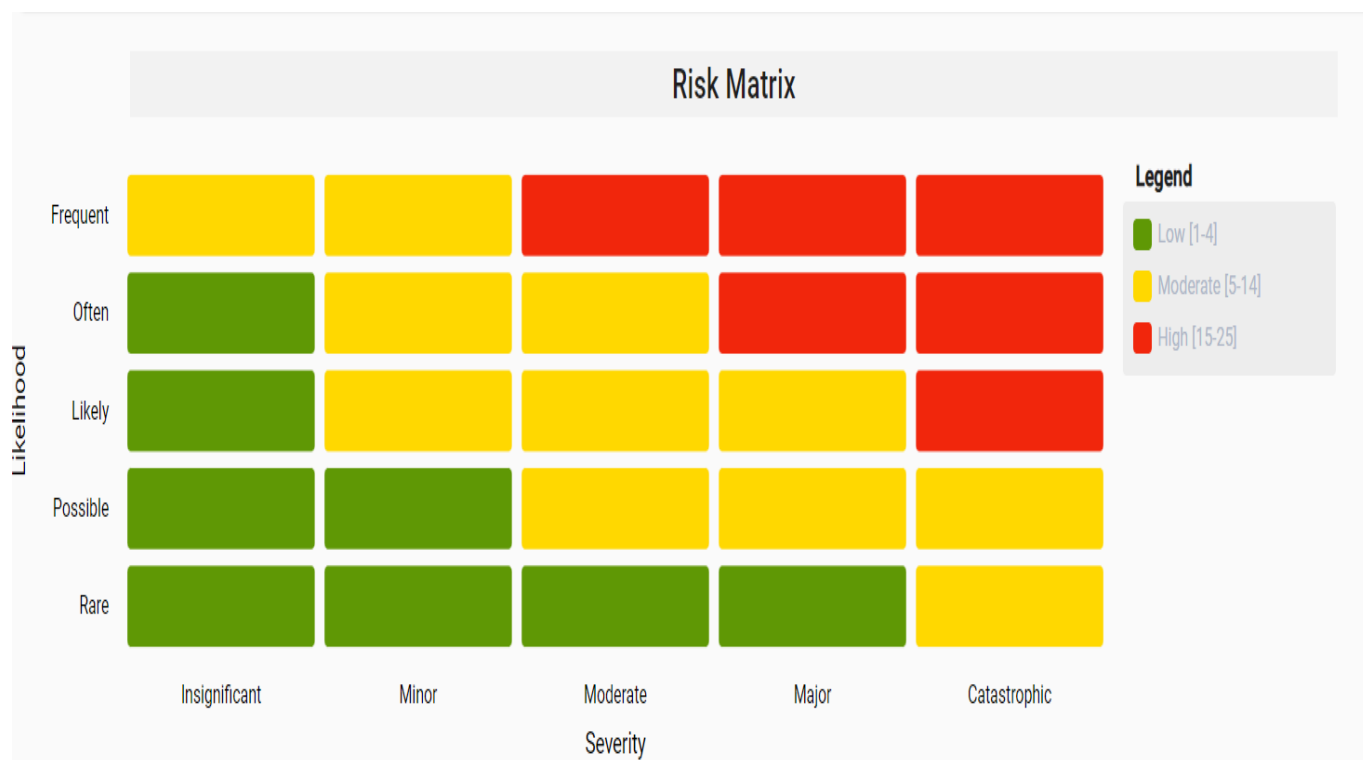
**Remarques :**

- Les valeurs de l'impact et de probabilité sont comprises entre 1 et 5. Et pour l'estimer, les assurances font recours à des experts.
- Les formules pour calculer risque net et le risque brut :
  - **Risk brut** = impact \* probability
  - **Risk net** = risk brut \* niv\_maitrise



## Annexe 2. Matrice de risque

La figure annexe 2.1 présente la matrice de risque :



**Figure annexe 1.2 :** Table risque

Cette matrise est un outil pour analyser et évoluer risque en amont la probabilité et la gravité(impact) des risques.

## Annexe 3. Exemple de JWT

La figure ci-dessous montre un exemple de JWT de notre solution :

```
"id": 1,  
"name": "lina ksouri",  
"email": "ksouri.lina36@gmail.com",  
"username": "lina",  
"password": "$2a$10$FzwxN/G3SrczQRA42QcAr07NB235.V5SisZLLLqXv9hvaqj1FmQPC",  
"phone": "23787323",  
"role": "MANAGER",  
"token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJsaW5hIiwicm9sZXMiOiJNQU5BR0VSIiwiaXhwIjoxNjYzNzYzNzIzZfQ.  
4KgD-4k37b-Q1v8NFI9bG-cgHLtmSQ2JfmQAXvIAR1LWV7TmKjb54sQzmJIkQa6l0MwCNGCfZ5RqvEi85_vcbw"
```

Figure annexe 1.3 : Exemple de JWT

# Résumé

Ce projet intitulé "**Conception et réalisation d'une solution de gestion de risque**", réalisé au sein de la société Vneuron et qui s'inscrit dans le cadre d'un projet de fin d'études à l'École Nationale d'Ingénieurs de Carthage. Ce travail consiste à développer une solution qui aide les compagnies d'assurance et bancaire d'établir un registre des entités et des processus et aussi d'identifier, analyser et contrôler les risques opérationnels en se basant sur une technique de management qui s'appelle ERM.

**Mots clés :** gestion de risque, assurance, bancaire, risque opérationnel, ERM.

# Abstract

This project entitled "**Design and implementation of a risk management solution**", carried out within the company Vneuron and which is part of an end of studies project at the National School of National Engineering School of Carthage. This work consists in developing a solution that helps insurance and banking companies to establish a register of entities and processes and also identify, analyze and control operational risks based on a management technique called ERM.

**Keywords :** risk management, insurance, banking, operational risk, ERM.