# Dédicaces

Je dédie le fruit de mes efforts à ma famille et toutes les personnes chères à mon cœur, qui par leur soutien, amour et encouragement ont contribué à l’aboutissement de ce travail.

À MES CHERS PARENTS MOHAMED ET SAMIA

Que ce travail soit l’expression de ma reconnaissance pour vos sacrifices consentis, votre Soutien moral et matériel que vous n’avez cessé de prodiguer. Vous avez tout fait pour Mon bonheur et ma réussite. Que Dieu vous préserve en bonne santé et vous protéger pour moi et mes fréres

Merci pour tout, Que Dieu vous préserve en bonne santé et vous accorde une longue vie

À MES PROFESSEURS ET MES ENSEIGNANTS

Qui m’ont formé durant ma carrière scolaire et universitaire, je vous remercie pour le bon encadrement que j’ai eu tout au long de ce projet.

Vous étiez toujours présents pour m’aider et m’encourager. Sachez que vous êtes toujours dans mon cœur.

À TOUS MES AMIS...

À tous mes amis qui n’ont cessé de m’encourager et de me soutenir À tous ceux que j’aime Et à tous ceux qui m’aiment

# Remerciements

Au terme de ce rapport, je tiens à exprimer mes profonds respects et gratitudes à l’École Polytechnique de Sousse qui m’a offert l’occasion d’effectuer ce stage au sein de la société " PEAK Technologies ", aussi bien, pour enrichir mes connaissances dans la vie professionnelle et de découvrir l’environnement de la société.

Je tiens à exprimer mes remerciements avec un grand plaisir et un grand respect à mon encadrant académique Mr. Rouatbi Adnen pour tous les conseils et les informations qu’il ma dispensés afin que ce stage se passe dans les meilleures conditions, pour son soutient, son sérieux, sa gentillesse et surtout pour son aide précieuse tout au long de l’élaboration de ce travail. .Je présente mes sincères remerciements et mes profondes gratitudes à mon encadrant professionnel Mr. Hasni Mehdi

son assistance, son soutien, sa disponibilité et ses précieux conseils durant la période de ce stage.

J’adresse aussi ma plus vive reconnaissance à tous mes enseignants de l’École Polytechnique de Sousse pour la formation qu’ils mont donnée ainsi qu’aux membres du jury qui ont accepté de juger ce travail.

Finalement, je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

TABLE DES MATIÈRES

**Introduction générale 1**

1. **Cadre général du projet 3**
   1. **Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   2. Étude Préalable : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
      1. Problématique : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
      2. Étude de l’existant : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
      3. Critique de l’existant : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
      4. Solution proposé : . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   3. Présentation de l’organisme d’accueil : . . . . . . . . . . . . . . . . . . . . . . . . 6
      1. Présentation de l’entreprise . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      2. Spécialisations de «L2T» . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
      3. Fiche Technique de «L2T» : . . . . . . . . . . . . . . . . . . . . . . . . . . 7
      4. Logo de L2T . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   4. **Conclusion** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
2. **Analyse et Spécification des besoins 9**
   1. **Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2. Spécification des besoins . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      1. Identification des acteurs . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      2. Les besoins fonctionnels . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
      3. Besoins non fonctionnels . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
   3. Étude de méthodologie de développement . . . . . . . . . . . . . . . . . . . . . . 11
      1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
      2. Choix de la méthodologie . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
      3. Méthodologie choisie (En v) . . . . . . . . . . . . . . . . . . . . . . . . . . 14
      4. Les principales caractéristiques de la méthode cycle en V . . . . . . . . . . . 14
   4. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
3. **Etude Conceptuelle 16**
   1. **Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
   2. Définition UML . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
   3. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      1. Diagramme des cas d’utilisation . . . . . . . . . . . . . . . . . . . . . . . . 17
      2. Diagramme de classes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
      3. Diagramme de séquence . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
      4. diagramme de déploiement . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
   4. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
4. **Réalisation 34**
   1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
   2. Architecture de la solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
      1. Modèle MVC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
   3. Environnement du projet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
      1. Environnement matériel . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
      2. Environnement logiciel . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
   4. Présentation de la solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
      1. Interface d’accueil . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
      2. Interface registre . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
      3. Interface login . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
      4. Dashboard admin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
      5. Liste annonces prestataire admin . . . . . . . . . . . . . . . . . . . . . . . . 44
      6. Liste profiles demandeur admin . . . . . . . . . . . . . . . . . . . . . . . . 44
      7. Choix type annonce prestataire . . . . . . . . . . . . . . . . . . . . . . . . . 45
      8. Créer annonce prestataire baby-sitter . . . . . . . . . . . . . . . . . . . . . . 46
      9. Créer annonce prestataire auxiliaire de vie . . . . . . . . . . . . . . . . . . . 47
      10. Créer annonce prestataire garde animaux . . . . . . . . . . . . . . . . . . . 48
      11. Liste annonces prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
      12. Détails annonce prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
      13. Chat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
      14. Profil prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
      15. Choix type annonce demandeur . . . . . . . . . . . . . . . . . . . . . . . . 51
      16. Créer annonce demandeur baby-sitter . . . . . . . . . . . . . . . . . . . . . 52
      17. Créer annonce demandeur auxiliaire de vie . . . . . . . . . . . . . . . . . . 53
      18. Créer annonce demandeur garde animaux . . . . . . . . . . . . . . . . . . . 54
      19. Edit annonce prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
      20. Supprimer annonce Demandeur . . . . . . . . . . . . . . . . . . . . . . . . 56
   5. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

**Conclusion Générale 57**

**Bibliographie 58**

TABLE DES FIGURES

* 1. Logo officiel de Bambinosit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
  2. Etude de l’application Yoopies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
  3. Logo officiel de L2T . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.1 Le cycle en v . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

* 1. UML . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
  2. Diagramme des cas d’utilisation globale . . . . . . . . . . . . . . . . . . . . . . . . 18
  3. Diagramme des cas d’utilisation «Valider utilisateur» . . . . . . . . . . . . . . . . . 19
  4. Diagramme des cas d’utilisation «Valider annonce» . . . . . . . . . . . . . . . . . . 20
  5. Diagramme des cas d’utilisation «Valider commentaire» . . . . . . . . . . . . . . . 21
  6. Diagramme des cas d’utilisation «gérer annonce» . . . . . . . . . . . . . . . . . . . 22
  7. Diagramme des cas d’utilisation «gérer profilr» . . . . . . . . . . . . . . . . . . . . 22
  8. Diagramme des cas d’utilisation «consulter annonce demandeur» . . . . . . . . . . . 23
  9. Diagramme des cas d’utilisation «consulter annonce prestataire» . . . . . . . . . . . 23
  10. Diagramme de classes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
  11. Diagramme de séquence «s’inscrire» . . . . . . . . . . . . . . . . . . . . . . . . . . 25
  12. Diagramme de séquence«s’authentifier» . . . . . . . . . . . . . . . . . . . . . . . . 26
  13. Diagramme de séquence«valider compte uilisateur» . . . . . . . . . . . . . . . . . . 26
  14. Diagramme de séquence«créer annonce» . . . . . . . . . . . . . . . . . . . . . . . . 27
  15. Diagramme de séquence «modifier annonce» . . . . . . . . . . . . . . . . . . . . . 28
  16. Diagramme de séquence «supprimer annonce» . . . . . . . . . . . . . . . . . . . . 29
  17. Diagramme de séquence «détail annonce» . . . . . . . . . . . . . . . . . . . . . . . 29
  18. Diagramme de séquence «ecrire commentaire» . . . . . . . . . . . . . . . . . . . . 30
  19. Diagramme de séquence «gérer message» . . . . . . . . . . . . . . . . . . . . . . . 31
  20. Diagramme de séquence «consulter notification» . . . . . . . . . . . . . . . . . . . 32
  21. Diagramme de séquence «diagramme de déploiement» . . . . . . . . . . . . . . . . 33
  22. Modèlé MVC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
  23. Interface d’accueil 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
  24. Choix du Type d’Utilisateur . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
  25. Formulaire d’Inscription . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
  26. Interface login . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
  27. Dashboard admin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
  28. Liste annonces prestataire admin . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
  29. Liste profiles demandeur admin . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
  30. Choix type annonce prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
  31. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
  32. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
  33. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
  34. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
  35. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
  36. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
  37. Liste annonces prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
  38. Détails annonce . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
  39. Chat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
  40. Profil prestataire . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
  41. Choix type annonce demandeur . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
  42. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
  43. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
  44. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
  45. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
  46. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
  47. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
  48. Step 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
  49. Step 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
  50. Supprimer annonce Demandeur . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

LISTE DES TABLEAUX

1.2 Etude de l’application Bambinosit . . . . . . . . . . . . . . . . . . . . . . . . . . . 5 1.3 Etude de l’application Yoopies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.4 Fiche signalétique de la société L2T . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2.1 Comparaison des méthodes de développement . . . . . . . . . . . . . . . . . . . . 14

* 1. Environnement matériel . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
  2. Technologies et languages utilisées . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

***Introduction générale***

Dans un contexte économique et réglementaire de plus en plus contraignant, la maîtrise des risques et la conformité aux normes constituent des enjeux majeurs pour les organisations. La digitalisation croissante des processus expose en effet les entreprises à de nouveaux risques – opérationnels, financiers, juridiques ou liés à la sécurité des données – qu’il convient d’identifier, d’évaluer puis de piloter de façon continue. C’est dans cette optique que s’inscrit notre Projet de Fin d’Études (PFE), dont l’objectif est de concevoir et de réaliser une plateforme logicielle innovante, baptisée **SentinelRisk**, dédiée à la gestion intégrée des risques et à la conformité réglementaire.

Le projet a été mené au sein de **PEAK Consulting GmbH – PEAK Technologies SARL**, cabinet de conseil spécialisé en Identity & Access Management (IAM) et en sécurité des systèmes d’information, implanté à Mannheim (Allemagne) et à Sousse (Tunisie). Fondée en 2016, PEAK-CS accompagne de grandes organisations dans la mise en place de solutions IAM, le développement d’applications sur mesure et la prévention de la fraude, s’appuyant sur une expertise attestée par un effectif de 11 – 50 collaborateurs et un portefeuille de références internationales.

Le sujet de notre projet de fin d’etudes porte sur le développement complet de SentinelRisk :

* **Étude et spécification des besoins** (identification des acteurs, besoins fonctionnels et non fonctionnels),
* **Conception conceptuelle** (modélisation UML des cas d’utilisation, des classes et des séquences),
* **Implémentation technique** (backend Spring Boot, frontend Angular, base PostgreSQL, authentification Keycloak),
* **Déploiement** (Docker Compose, Kubernetes).

Les principaux objectifs sont de proposer une solution robuste et évolutive permettant :

1. d’agréger et de catégoriser les risques (opérationnels, financiers, sécurité des données…),
2. de calculer automatiquement un score de risque et de piloter des plans de mitigation,
3. d’assurer le suivi de la conformité réglementaire via des cadres (ISO 27001, NIST, GDPR, SOC 2),
4. de gérer les profils utilisateurs et les droits d’accès grâce à une intégration transparente avec Keycloak.

Pour guider la rédaction de ce rapport, la structure retenue est la suivante :

* **Chapitre 1 : Cadre général du projet**

Resitue le contexte métier, présente l’organisme d’accueil, expose la problématique, justifie le choix de la méthodologie en cycle en V et illustre la planification par un diagramme de Gantt.

* **Chapitre 2 : État de l’art**

Dresse une revue des solutions et normes existantes, en abordant notamment le protocole SNMP, la gouvernance et les cadres réglementaires (ISO 27001, NIST, GDPR, SOC 2).

* **Chapitre 3 : Analyse et spécification des besoins et architecture**

Identifie les acteurs et leurs cas d’usage, détaille les exigences fonctionnelles et non-fonctionnelles, puis décrit l’architecture technique retenue.

* **Chapitre 4 : Étude conceptuelle**

Modélise le système à l’aide de diagrammes UML (cas d’utilisation, classes, séquences et déploiement).

* **Chapitre 5 : Réalisation**

présente l’environnement de développement et détaille l’application à travers ses interfaces et ses flux métiers.

* **Conclusion générale et perspectives**

Bilan des objectifs atteints, apports personnels et pistes d’évolution

Cette trame nous permettra de présenter de manière claire et méthodique l’ensemble des travaux réalisés dans le cadre de notre PFE, de l’analyse initiale à la mise en œuvre concrète de la plateforme SentinelRisk.

CHAPITRE 1

CADRE GÉNÉRAL DU PROJET

**1.1 Introduction**

Avant de démarrer tout projet, il est essentiel d’analyser le contexte et l’existant afin de bien définir les objectifs et les besoins. Cette étape permet de mieux comprendre le cadre général du travail, d’identifier les points à améliorer et d’orienter les choix techniques.   
Dans ce chapitre, nous présentons l’organisme d’accueil, identifions la problématique, analysons et critiquons l’existant, puis proposons une solution adaptée. Nous détaillons ensuite la méthodologie de développement adoptée et la planification du projet à travers un diagramme de Gantt.

**1.3 Présentation de l’organisme d’accueil :**

**1.3.1 Présentation de l’entreprise**

PEAK Consulting GmbH (Allemagne) et sa filiale PEAK Technologies SARL (Tunisie) constituent un cabinet de conseil fondé en 2016, spécialisé dans l’Identity & Access Management (IAM) et le développement de solutions sur mesure. Depuis sa création, PEAK-CS a mené de nombreux projets pour des clients internationaux de premier plan, bénéficiant d’un fort taux de satisfaction et d’un positionnement reconnu sur le marché de la sécurité des systèmes d’information

**1.3.2 Spécialisations de «PEAK Technologies SARL»**

• IAM Consulting : intégration et exploitation de solutions IAM adaptées aux besoins métiers.

• Développement : réalisation de solutions logicielles web et mobiles, incluant des applications IAM, .NET, Android, iOS…

• Graphic Design : création d’identités visuelles, brochures et supports de communication.

• Services pour startups : recrutement de talents, umbrella services (portage salarial) et outsourcing de développement

**1.3.3 Fiche Technique de «PEAK Technologies SARL» :**

|  |  |
| --- | --- |
| **Nom de**  **l’entreprise :** | PEAK Technologies SARL |
| **Nom du gérant :** | Mr Finch John |
| **Site web :** | https ://www.l2t.io/ |
| **Adresse :** | Imm. Espace Sousse, Rue de Khartoum, Sousse 4000. |
| **Téléphone :** | +(216) 22 419 820 |
| **Émail :** | info@peak-cs.com |

Table 1.4 – Fiche signalétique de la société PEAK Technologies SARL

A blue text on a black background

AI-generated content may be incorrect.

Figure 1.3 – Logo officiel de PEAK Technologies SARL

**1.4 Problématique :**

Dans le contexte actuel, les organisations sont exposées à une multitude de risques – opérationnels, financiers, juridiques, technologiques et liés à la sécurité des données – dont la nature et l’impact évoluent constamment. La multiplication des référentiels normatifs (ISO 27001, NIST, GDPR, SOC 2, etc.) et l’hétérogénéité des outils de gestion (tableurs, solutions ponctuelles de reporting, ERP, solutions IAM isolées) conduisent souvent à une vision fragmentée et peu dynamique du pilotage des risques et de la conformité.

Les principaux points de blocage identifiés sont :

* **Dispersion des données et des processus** : absence d’un référentiel unique pour consolider les incidents, les évaluations et les plans d’action, ce qui engendre des redondances et des écarts d’information.
* **Calcul de risque manuel ou semi-automatique** : les méthodes artisanales (feuilles Excel, scripts isolés) ne garantissent ni la traçabilité ni la fiabilité des scores, et sont difficiles à maintenir face à un contexte réglementaire évolutif.
* **Manque d’intégration IAM & gouvernance** : la gestion des droits d’accès et des profils utilisateurs, cruciale pour la conformité, n’est pas toujours reliée aux workflows de mitigation des risques.
* **Faible capacité d’analyse transverse** : difficulté à obtenir des indicateurs consolidés et des tableaux de bord dynamiques qui permettent de prioriser rapidement les actions selon le niveau de criticité.
* **Absence de surveillance automatisée des infrastructures** : les ressources réseau (serveurs, routeurs, équipements IoT, postes de travail) ne sont pas suffisamment supervisées en continu, ce qui retarde la détection des vulnérabilités et des anomalies opérationnelles.

Ainsi, la **problématique** centrale de ce projet consiste à **concevoir et implémenter une plateforme intégrée** qui :

1. **Centralise** toutes les informations relatives aux risques et à la conformité,
2. **Automatise** le calcul des scores de risque et la génération de plans de mitigation,
3. **Intègre** la gestion des identités et des accès via Keycloak pour assurer un contrôle granulaire des droits,
4. **Superviser en continu** l’état des actifs réseau grâce au protocole SNMP (collecte d’OID, détection d’anomalies, remontée automatique de risques),
5. **Fournit** des tableaux de bord interactifs et des indicateurs clés pour un pilotage en temps réel.

Répondre à cette problématique permettra de doter les entreprises d’un outil unique, évolutif et sécurisé, capable de répondre aux exigences réglementaires tout en facilitant la prise de décision et la réactivité face aux nouveaux risques.

* + 1. **Étude de l’existant :**

La conception et la réalisation de tout projet doivent être précédées par une étude de l’existant qui détermine les points faibles et les points forts des systèmes actuels, tenant compte des objectifs à atteindre. La recherche que nous avons menée sur les solutions de Gouvernance et de Monitoring existantes dans le marché tunisien et dans le marché international, nous a permis d’exposer dans la suite quelques applications qui s’y intéressent de près :

* + - 1. **Étude de l’application SimpleRisk**

**SimpleRisk,** vise à centraliser la gestion des risques au sein des entreprises, en proposant une plateforme intuitive, complète et adaptée aux exigences de gouvernance et de conformité des organisations modernes . *D’une part*, cette plateforme va permettre aux entreprises d’identifier, d’évaluer, de suivre et d’atténuer l’ensemble de leurs risques sur une interface unique , leur offrant une vue globale pour prioriser les actions, anticiper les problèmes et assurer la conformité réglementaire. *D’autre part*, elle permettra aux équipes chargées des risques et de la conformité de collaborer plus facilement grâce à des workflows clairs et automatisés, d’accélérer leurs processus de suivi et de validation, et de se concentrer sur la résolution des problèmes plutôt que sur des tâches administratives lourdes. Nous mettons à la disposition des entreprises un registre centralisé des risques, des workflows intégrés pour la gestion et le suivi des actions, un reporting clair et complet, des outils pour assurer la conformité réglementaire, etc., afin que chaque utilisateur – du dirigeant au collaborateur – puisse bénéficier d’un pilotage des risques simplifié et efficace.



Figure 1.1 – Logo officiel de SimpleRisk

|  |  |
| --- | --- |
| **Avantages :** | **inconvénients :** |
| - Open-source  -Facilité d’Utilisation  - Personnalisation par développeurs  - Communauté active  - Extensibilité | - Fonctionnalités limitées  - Interface utilisateur vieillissante  - Absence de module de conformité avancée  - Manque de tableaux de bord dynamiques  - Gestion des accès et audit peu robuste |

Table 1.2 – Étude ade l’application SimpleRisk

* + - 1. **Étude de l’application Eramba**

**Eramba,** est une plateforme open-source de **Gouvernance, Risque & Comformité (GRC)** destinée à offrir aux organisations un outil unique pour organiser, surveiller et documenter l’ensemble de leurs processus de gouvernance, de gestion des risques et de conformité. *D’une part*, elle permet de définir et de maintenir un référentiel normatif (ISO 27001, SOC 2, PCI-DSS…) tout en traçant incidents, évaluations, plans d’action et audits dans un registre centralisé. *D’autre part*, elle fournit des workflows automatisés pour la gestion des politiques de sécurité, la collecte d’évidences, la planification des revues et la génération de rapports, facilitant la collaboration entre les équipes de sécurité, de conformité et d’audit. Avec ses tableaux de bord prêts à l’emploi, son moteur de reporting et sa modularité via plugins, Eramba s’adresse aussi bien aux petits organismes qu’aux grandes entreprises souhaitant piloter efficacement leurs enjeux GRC.[2]

A green logo with a white background

AI-generated content may be incorrect.

Figure 1.1 – Logo officiel de Eramba

|  |  |
| --- | --- |
| **Avantages :** | **inconvénients :** |
| - Open-source et sans licence  - Référentiels normatifs intégrés  - Workflows GRC complets  - Extensible via plugins et API  - Reporting et tableaux de bord prêts à l’emploi | - Automatisation limitée  - Absence de supervision réseau  - Intégration IAM basique  - Analyse transverse limitée |

Table 1.2 – Etude de l’application Eramba

**1.2.3 Critique de l’existant :**

Cette partie consiste à comprendre et analyser les solutions existantes et à déterminer leurs points

Faibles et leurs points forts pour pouvoir dégager les besoins du projet, et de les prendre en considération

|  |  |
| --- | --- |
| **Avantages :** | **inconvénients :** |
| - Plateformes open-source, sans coûts de licence  - Référentiels normatifs (ISO 27001, NIST, SOC 2, GDPR…) intégrés  - Workflows GRC complets (risque, contrôle, conformité, plans d’action)  - Extensible via plugins et API  - Communauté active et documentation pour pilotes (configurations, guides, retours d’expérience) | - Automatisation limitée (calculs semi-manuels, pas de supervision réseau)  - Supervision SNMP / collecte d’actifs absente  - IAM & gouvernance basiques (intégration Keycloak/accès peu granulaires)  - Tableaux de bord statiques ou vieillissants (manque d’indicateurs temps réel et de filtres avancés)  -UX/UI perfectible (ergonomie, responsivité, cohérence graphique) |

Table 1.2 – Etude de l’application Eramba

**1.2.4 Solution proposée :**

Notre solution, SentinelRisk, se présente comme une plateforme web modulaire et centralisée, conçue pour couvrir l’ensemble du cycle de vie de la gestion des risques et de la conformité réglementaire. Elle repose sur une architecture à trois niveaux : un backend robuste développé en Spring Boot, exposant des API REST sécurisées par Keycloak ,un frontend moderne en Angular Material garantissant une expérience utilisateur fluide et responsive et une base de données PostgreSQL, assurant performance et fiabilité du stockage des données. SentinelRisk intègre nativement la gestion des identités et des accès (IAM) via Keycloak pour relier automatiquement les profils utilisateurs aux workflows de mitigation et aux processus d’audit. Les principaux modules – catégories de risque, identification et évaluation des risques, contrôle et plan de mitigation, évaluations périodiques et mapping vers les cadres ISO 27001, NIST, GDPR, SOC 2 – communiquent entre eux afin d’offrir un référentiel unique et cohérent. L’automatisation du calcul des scores de risque, la génération dynamique de tableaux de bord et d’alertes, ainsi que le déploiement simplifié en conteneurs Docker garantissent une solution évolutive, facile à maintenir et adaptée aux besoins des PME comme des grands groupes.

## **Étude de méthodologie de développement**

* + 1. **Introduction**

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d’une application. C’est la manière particulière avec laquelle les tâches dans ce projet sont organisées. Une méthodologie de développement permet : de produire un produit de meilleur qualité (mieux documenté, acceptable par l’utilisateur, facile à maintenir, plus homogène), aide à assurer que les spécifications des utilisateurs sont suivies, aide le manager du projet à contrôler le projet, réduit les coûts de développement, encourage la communication entre les personnes.

* + 1. **Choix de la méthodologie**

Devant le nombre de méthodes disponibles, le choix parmi elles devient difficile. Nous étudions à ce propos quelques méthodes de développement objet. Suite à cette étude, nous choisissons la méthodologie la plus correspondante.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Description** | **Points Forts** | **Point Faibles** |
|  | - Le développement |  |  |
|  | reprend les différentes |  |  |
|  | étapes du cycle en V. |  | -Adapté qu’aux projets |
|  | Par l’implémentation | - Très adaptatif : si | suffisamment gros. Ce |
|  | de versions | chaque prototype | type de cycle |
|  | successives, le cycle | apporte des | dedéveloppement est |
|  | recommence en | fonctionnalités | complexe, entre les |
|  | proposant un produit | indépendantes, il est | étapes prévues en |
|  | de plus en plus complet | possible de changer | théorie et celles mises |
| Spirale | et robuste c’est la | l’ordre de livraison des | en pratique il y a une |
|  | notion de prototype. | versions. | grande différence. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | - Rarement utilisé tel |
|  |  |  | quel. - le V est bien |
|  |  | - Validations | souvent déséquilibré, |
|  |  | intermédiaires. - Facile | tantôt côté analyse, |
|  | - Ce modèle se | de prévoir les tests à | tantôt côté recette et la |
|  | découpe en 9 phases | réaliser au moment où | marge d’erreur est bien |
|  | qui se répondent 2 à 2 : | l’on conçoit une | souvent |
|  | à chaque étape de | fonctionnalité ou une | proportionnelle à la |
|  | conception correspond | interface, le travail | marge de liberté prise |
|  | une phase de test ou de | s’enchaîne donc de | par rapport au modèle |
| En v | validation. | façon assez naturelle. | théorique. . |
|  |  | -Itératif. -Spécifie le |  |
|  | - -Promu par Rational | dialogue entre les |  |
|  | -Le RUP est à la fois | différents intervenants |  |
|  | une méthodologie et un | du projet : les livrables, |  |
|  | outil prêt à l’emploi | les plannings, les | -Coûteux à |
|  | (documents types | prototypes. . . -Propose | personnaliser. -Très axé |
| RUP | partagés dans un | des modèles de | processus, au détriment |
| (Rational | référentiel web). -Cible | documents, et des | du développement : |
| Unified | des projets de plus de | canevas pour des | peu de place pour le |
| Process) | 10 personnes. | projets types. | code et la technologie. |
|  |  |  | - Ne couvre pas les |
|  |  | -Itératif. -Simple à | phases en amont et en |
|  |  | mettre en œuvre. -Fait | aval au |
|  |  | une large place aux | développement : |
| XP | -Ensemble de « Bests | aspects techniques : | capture des besoins, |
| (Extreme | Practices » de | prototypes, règles de | support, maintenance, |
| Program- | développement (travail | développement, | tests d’intégration. . . |
| ming) | en équipes, transfert de | tests. . . -Innovant : | -Élude la phase |
| (Two | compétences. . .). | programmation en duo, | d’analyse, si bien qu’on |
| Tracks) | -Cible des projets de | kick-off matinal | dépenser son énergie à |
|  | moins de 10 personnes. | debout. | faire et défaire. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | -Plutôt superficiel sur |
|  |  |  | les phases situées en |
|  |  | Itératif. -Fait une large | amont et aval du |
|  |  | place à la technologie | développement : |
| 2TUP | -S’articule autour de | et à la gestion du | capture des besoins, |
| (Two | l’architecture. -Cible | risque. -Définit les | support, |
| Tracks | des projets de toutes | profils des | maintenance,gestion du |
| Unified | les tailles. -Propose un | intervenants, les | changement. -Ne |
| Process) | cycle de | livrables, les plannings, | propose pas de |
|  | développement en Y. | les prototypes | documents types. |

Table 2.1 – Comparaison des méthodes de développement

* + 1. **Méthodologie choisie (En v)**

Le cycle en V est une méthode de gestion de projet linéaire utilisée principalement pour le développement de logiciels. Chaque phase de conception est accompagnée d’une phase de test. Bien que ce modèle soit déjà un peu ancien, il s’agit d’une méthode idéale pour les projets simples et de petite taille.

A diagram of a v

AI-generated content may be incorrect.

Figure 2.1 – Le cycle en v

* + 1. **Les principales caractéristiques de la méthode cycle en V**

La méthode du cycle en V implique toutes les étapes du cycle de mise en œuvre d’un produit ou d’un logiciel. Elle permet de définir le processus d’un projet en neuf étapes regroupées en trois phases : la conception, la réalisation et la validation.

* + - * **La conception :**

Dans un premier temps, la phase de conception permet de recueillir les besoins et spécifications

nécessaires pour programmer de manière exacte l’ensemble du projet. Il s’agit également de compiler à l’avance les éléments indispensables au développement du projet. La conception d’un logiciel selon la méthode du cycle en V implique quatre étapes : l’analyse des besoins, les spécifications, la conception générale et la conception détaillée.

* + - * **La réalisation :**

Après la conception vient l’étape de réalisation qui permet de créer et d’assembler tous les com- posants nécessaires pour arriver au produit final. Il peut s’agir de la fabrication d’un équipement ou du codage d’un logiciel. Dans ce dernier cas, l’équipe choisit le langage de programmation en fonction des attentes fonctionnelles et architecturales établies lors de la phase de conception. L’étape de réalisation doit ensuite suivre les directives et normes de codage en vigueur.

* + - * **La validation :**

Conception du produit. Ces mesures d’assurance qualité prennent la forme de tests unitaires, de tests d’intégration, de tests de validation et d’une recette fonctionnelle. Les tests unitaires correspondent à l’étape de conception détaillée. Réalisés au niveau du code par le développeur, ils permettent de détecter et de corriger d’éventuelles erreurs. L’équipe contrôle ainsi le bon fonctionnement de chaque brique logicielle.[3]

* + 1. **Planification du projet**

Le diagramme de Gantt ci-dessous présente la planification de notre projet répartie sur cinq mois. Chaque barre correspond à une phase majeure du cycle en V, positionnée sur l’axe temporel mensuel :

A graph with different colored rectangles

AI-generated content may be incorrect.

Figure 2.1 – Diagramme de Gantt

* + Analyse des besoins : 10 février – 10 mars
  + Spécifications système : 10 février – 10 avril
  + Conception détaillée : 10 mars – 10 avril
  + Implémentation / Développement : 10 avril – 10 juin
  + Tests unitaires : 10 mai – 31 mai
  + Tests d’intégration : 1 juin – 7 juin
  + Tests de validation (recette) : 8 juin – 10 juin

**1.4 Conclusion**

Ce premier chapitre a plongé dans le contexte métier de la gestion des risques en présentant l’organisme d’accueil et ses enjeux, exposé la problématique d’une vision fragmentée et manuelle des workflows GRC, justifié le recours à une méthodologie en cycle en V pour garantir rigueur et traçabilité tout au long du projet, et illustré notre calendrier de travail à travers un diagramme de Gantt. Ces éléments servent de préambule à l’état de l’art présenté au chapitre suivant.

CHAPITRE 2

État de l’art

1. **Introduction**

Au sein de l’environnement sans cesse renouvelé des infrastructures informatiques, la supervision réseau est aujourd’hui un pilier indispensable pour garantir performance, fiabilité et sécurité. Confrontés à l’expansion et à la complexité croissantes des architectures, les outils de monitoring doivent évoluer pour fournir des données à la fois précises et en temps réel, tout en offrant simplicité d’usage, capacité de montée en charge et intégration fluide. Le chapitre « État de l’art » propose une étude approfondie des solutions de surveillance réseau les plus répandues, examinant leurs fonctionnalités, architectures et limites. En comparant ces outils selon leur ergonomie, leur évolutivité, leurs performances et leur coût, nous mettons en lumière les lacunes persistantes. Cette analyse révèle tant les atouts que les faiblesses de plateformes établies comme Nagios, Zabbix ou PRTG, et justifie la nécessité d’une solution plus moderne et adaptable. Comprendre l’état actuel du domaine est une étape essentielle pour concevoir un système à la fois innovant et en phase avec les besoins opérationnels du secteur.

1. **Aperçu sur SNMP**

Le protocole SNMP (Simple Network Management Protocol) est couramment employé pour administrer les équipements sur les réseaux IP. Il permet de superviser et de contrôler des matériels tels que routeurs, commutateurs, serveurs ou imprimantes. SNMP repose sur une architecture client-serveur : le gestionnaire SNMP (client) échange avec les agents SNMP (serveurs) installés sur les dispositifs. Cette communication donne aux administrateurs la possibilité de recueillir des indicateurs clés de performance, de configurer les équipements et de recevoir des notifications en cas d’événements réseau.

SNMP a connu plusieurs évolutions : SNMPv1 a vu le jour à la fin des années 1980, suivi de SNMPv2 au début des années 1990, puis de SNMPv3 à la fin des années 1990, cette dernière version apportant des renforcements significatifs en matière de sécurité et de gestion des données. Grâce à ces avancées successives, SNMP demeure un outil fiable et complet pour la gestion des infrastructures réseau.[1]

1. **Importance de la gestion réseau**

Une gestion efficace du réseau est essentielle au maintien de la santé et des performances des infrastructures numériques modernes. Face à la complexité croissante des réseaux et au nombre croissant d'appareils connectés, les entreprises sont confrontées à des défis majeurs pour garantir la fiabilité des communications et des transferts de données. Une gestion réseau efficace contribue à :

* **Surveillance des Performance** : Suivi Continue de réseau performance métrique permet d'identifier en temps opportun les goulots d'étranglement et les opportunités d'optimisation.
* **Gestion des pannes** : SNMP permet une détection rapide des défaillances, ce qui permet aux administrateurs de réagir rapidement pour minimiser les temps d’arrêt.
* **Gestion de la configuration** : Les changements de configuration automatisés peuvent être mis en œuvre à l’aide du SNMP, ce qui améliore l’efficacité opérationnelle et la cohérence entre les appareils.
* **Planification de la capacité**: En analysant les tendances d’utilisation, les organisations peuvent mieux planifier la croissance future et l’affectation des ressources.

L’utilisation efficace du SNMP contribue à l’efficacité organisationnelle globale, réduisant les coûts opérationnels et améliorant la prestation des services.

1. **Architecture SNMP**

L’architecture Simple Network Management Protocol (SNMP) est conçue pour permettre une surveillance et une gestion efficaces des périphériques réseau. Il fonctionne sur un modèle client-serveur, composé de deux composants principaux : les gestionnaires SNMP et les agents SNMP. Cette architecture permet aux administrateurs réseau de recueillir des informations, de configurer les périphériques et de répondre efficacement aux événements sur divers éléments du réseau.

Le SNMP architecture comprend :

* **SNMP Directeur** : Centralisé logiciel que supervise le fonctionnement du réseau.
* **SNMP Agent** : Logiciel en cours d'exécution sur réseau appareils, collectionner et stockage des données de gestion.
* **Base d'informations de gestion (MIB)** : Base de données définissant la structure et les types de données disponibles pour la gestion.
  1. **Gestion Information Base (MIB)**

Le La base d'informations de gestion (MIB) est un composant crucial de SNMP Architecture. Il s'agit d'un référentiel structuré décrivant les objets gérés au sein d'un périphérique réseau. Les MIB sont organisées hiérarchiquement et utilisent des identifiants d'objet (OID) pour identifier chaque objet de manière unique.

Clé caractéristiques de Les MIB incluent :

* + - **Structure hiérarchique** : les objets sont organisés sous forme d'arborescence, facilitant ainsi la navigation et l'accès.
    - **Standard et Coutume Objets** : Contient standard objets défini par IETF comme Bien en tant qu'objets spécifiques au fournisseur qui fournissent des fonctionnalités supplémentaires.
    - **Types de données** : définit différents types de données pour les objets, tels qu'un entier, une chaîne ou une jauge, permettant une représentation précise des données.
  1. **SNMP agents**

SNMP agents sont logiciel applications installé sur réseau appareils que collecter et gérer l'information. Leurs principales responsabilités comprennent :

* + - **Données Collection** : Surveillance appareil paramètres comme processeur usage, mémoire, trafic réseau, etc. Ces données sont stockées dans la base de données MIB.
    - **Réponse aux demandes** : lorsqu'un gestionnaire SNMP envoie une demande, l'agent récupère les données pertinentes du MIB et répond en conséquence.
    - **Envoi de notifications** : les agents peuvent générer des messages non sollicités appelés interruptions pour alerter le gestionnaire SNMP des événements importants, tels que des pannes ou des dépassements de seuil.
  1. **SNMP gestionnaires et leur rôles**

Les gestionnaires SNMP, souvent intégrés aux systèmes de gestion de réseau (NMS), jouent un rôle essentiel dans la gestion des opérations réseau. Leurs principales fonctions incluent :

* + - **Information Récupération** : Envoi demandes à SNMP agents à collecter données depuis le MIB, permettant la surveillance en temps réel des périphériques réseau.
    - **Configuration Gestion** : Faciliter changements à appareil configurations en envoyant des commandes aux agents, garantissant ainsi des performances optimales de l'appareil.
    - **Surveillance des événements** : analyse continue des données des agents et réponse aux pièges pour résoudre les problèmes rapidement.
    - **Rapports et Analyse** : Agrégation données sur temps pour s'orienter analyse, rapports et planification des capacités, aidant les administrateurs à prendre des décisions éclairées
  1. **Communication Modèle (Protocole Données Unités)**

La communication dans SNMP est principalement réalisé en utilisant Unités de données de protocole (PDU). PDU sont le format pour messages échangés entre SNMP gestionnaires et agents. Le Les principaux types de PDU dans SNMP incluent :

* **GET** : Utilisé par le gestionnaire SNMP pour demander des informations à un agent. L'agent répond avec les données demandées.
* **ENSEMBLE** : Permet le SNMP gestionnaire de modifier le valeurs de spécifique objets dans un MIB de l'agent. Utilisé pour les modifications de configuration.
* **GETNEXT** : Facilite la récupération de l'objet suivant dans la hiérarchie MIB, permettant au gestionnaire de parcourir la structure MIB.
* **PIÈGE** : Un non sollicité message envoyé par le agent à le directeur, alerte il à événements importants ou changements de statut.
* **INFORMER** : Similaire à PIÈGE, mais nécessite reconnaissance depuis le gestionnaire, assurant une communication fiable.

Ces PDU activer robuste et efficace communication dans le SNMP cadre permettant une gestion efficace du réseau.[2]

* 1. **Flux de travail de SNMP opérations**

Le flux de travail des opérations SNMP implique plusieurs étapes clés, en particulier pour les opérations GET, SET et TRAP :

* + - * + **OBTENIR Opération :**

1. Le gestionnaire SNMP envoie une requête GET PDU à l'agent SNMP, spécifiant l'OID des données souhaitées.
2. Le agent processus le demande, récupère le données depuis le MIB, et renvoie une réponse PDU contenant les informations demandées.
   * + - * **Opération SET** :
3. Le gestionnaire SNMP envoie une requête SET PDU à l'agent, incluant l'OID et la nouvelle valeur à définir.
4. Le agent mises à jour le correspondant objet dans le MIB et répond avec confirmation indiquant le succès ou l'échec de l'opération.
   * + - * **PIÈGE Opération** :
         1. Lorsqu'un événement important se produit (par exemple, une panne de périphérique ou un dépassement de seuil), l'agent SNMP génère un PDU TRAP.
         2. L’agent envoie le PIÈGE à le SNMP directeur, lequel journaux le événement et peut déclencher des alertes ou des réponses automatisées.

Ce flux de travail structuré facilite une communication efficace entre les gestionnaires et les agents, permettant des réponses rapides aux conditions du réseau.

1. **Applications de SNMP**

Ce chapitre examine les applications de SNMP dans divers secteurs, mettant en valeur son rôle essentiel dans la gestion du réseau et la surveillance des performances.

* 1. **Utiliser cas dans divers industries**

Le protocole SNMP (Simple Network Management Protocol) est largement utilisé dans de nombreux secteurs en raison de son efficacité pour la surveillance et la gestion des réseaux. Parmi les principaux cas d'utilisation, on peut citer :

* **Télécommunications** : SNMP est essentiel à la surveillance de l'infrastructure réseau, notamment des routeurs, des commutateurs et des stations de base. Il permet aux fournisseurs de garantir la qualité de service, de gérer la bande passante et de réagir rapidement aux pannes ou aux problèmes de performances.
* **Technologies de l'information (TI)** : Dans les environnements informatiques, SNMP est utilisé pour gérer les serveurs, les postes de travail et les périphériques réseau. Il permet aux administrateurs de surveiller l'état du système, de suivre l'utilisation des ressources et d'appliquer des modifications de configuration à distance.
* **Internet des objets (IoT)** : Face à la prolifération des objets connectés, SNMP joue un rôle essentiel dans leur surveillance et leur gestion. Il permet une gestion centralisée de divers objets connectés, garantissant leur fonctionnement efficace et sécurisé.
  1. **Utilisé réseau surveillance outil**

Nombreux applications et outils effet de levier SNMP pour efficace réseau gestion.

Notable exemples inclure

Dans notre cas, nous utilisons Zabbix, une solution de surveillance réseau open source robuste qui fournit des informations détaillées sur les performances et la santé des infrastructures informatiques. Conçue pour surveiller divers composants, notamment les serveurs, les périphériques réseau, les applications et les ressources cloud, elle offre une grande polyvalence pour différents environnements. Voici les principales fonctionnalités de Zabbix [3].

* **Surveillance en temps réel** : Zabbix offre des fonctionnalités de surveillance en direct, permettant aux administrateurs de suivre les performances du réseau et l'état des appareils en temps réel, ce qui est crucial pour la détection immédiate des problèmes.
* **Méthodes de collecte des données** : il prend en charge plusieurs méthodes de collecte de données, notamment SNMP, IPMI, JMX et la surveillance basée sur des agents, offrant une flexibilité dans la manière dont les données sont collectées.
* **Tableaux de bord personnalisables** : les utilisateurs peuvent créer des tableaux de bord personnalisés adaptés à leurs besoins de surveillance spécifiques, affichant des indicateurs clés et des tendances pour une analyse rapide.
* **Alerte et Notifications** : Zabbix dispose d'un puissant système d'alerte qui peut informer les utilisateurs de problèmes potentiels par e-mail, SMS ou autres méthodes de communication, garantissant des réponses rapides aux événements critiques.
* **Évolutivité** : Il est conçu pour évoluer des petits réseaux aux grands environnements d'entreprise, capable de surveiller des milliers d'appareils sans compromettre les performances.
* **Données historiques et rapports** : Zabbix conserve les données historiques, permettant l'analyse des tendances et les rapports de performances, ce qui facilite la planification des capacités et la gestion des ressources.

1. **9 Gouvernance et conformité**

La gouvernance et la conformité sont devenues des piliers incontournables pour toute organisation souhaitant maîtriser ses risques et répondre aux exigences réglementaires. Là où la supervision technique (SNMP, logs, SIEM) couvre la santé et la performance des infrastructures, la gouvernance s’intéresse aux processus, rôles et responsabilités, et la conformité aux référentiels normatifs (ISO 27001, NIST, SOC 2, GDPR…).

**9.1Définitions clés**

* **Gouvernance** : ensemble des règles, processus et structures qui orientent, pilotent et contrôlent l’organisation pour atteindre ses objectifs tout en respectant les réglementations et bonnes pratiques.
* **Conformité** : état d’un système qui satisfait aux exigences d’un référentiel normatif ou réglementaire, via la mise en place de politiques, procédures et contrôles adaptés.

**9.2Référentiels normatifs**

* **ISO 27001** : standard international pour un Système de Management de la Sécurité de l’Information (SMSI), fondé sur l’analyse de risques et le choix de mesures de contrôle.
* **GDPR** : règlement européen sur la protection des données personnelles, obligeant à documenter les traitements, obtenir des consentements et répondre aux droits des personnes.
* **NIST Cybersecurity Framework** : guide américain décrivant cinq fonctions (Identify, Protect, Detect, Respond, Recover) pour structurer la gestion du risque cyber.
  + **SOC 2** : rapport de conformité américain centré sur la sécurité, la disponibilité, l’intégrité du traitement, la confidentialité et la vie privée.

**9.3Outils et pratiques du marché**

Plusieurs plateformes GRC (Governance, Risk & Compliance) open-source ou commerciales — Eramba, SimpleRisk, RSA Archer, OpenPages — proposent :

* + - Un **référentiel central** des politiques et exigences.
    - Des **workflows** d’audit, de gestion des non-conformités et de plans d’action.
    - Des **tableaux de bord** et des rapports standards.

Cependant, ces solutions montrent souvent des limites : interfaces datées, intégrations disparates (IAM, supervision technique), et automatisation partielle des calculs de risque et des escalades.

**9.4Limites identifiées**

* + Fragmentation des données : risques métier, incidents techniques et obligations réglementaires gérés dans des silos.
  + Traçabilité et audit : difficultés à relier chaque action utilisateur à un événement de conformité.

Reporting statique : dashboards peu interactifs, indicateurs généralement mis à jour manuellement.

**10 Cycle de gestion du risque**

Le cycle de gestion du risque est la démarche systématique que suivent les organisations pour identifier, évaluer, traiter et surveiller leurs risques.

Il se compose de cinq étapes clés :

1. **Identification du Risque**

Recueil des menaces et vulnérabilités potentielles via des méthodes variées : ateliers internes, entretiens avec les métiers, audits, sondes techniques (SNMP, journaux d’événements, SIEM). Cette phase permet de constituer un inventaire exhaustif des risques.

1. **Analyse du risque**

Estimation de la **probabilité** d’occurrence et de l’**impact** sur les actifs ou processus. On distingue deux approaches :

* **Qualitative** (échelles faibles/moyen/élevé, matrices de criticité)
* **Quantitative** (calcul de scores, valeur monétaire du risque)

1. **Évaluation du risque**

Priorisation des risques selon leur score global (impact × probabilité), en tenant compte du contexte métier et des exigences réglementaires. Seuls les risques dépassant un certain seuil seront traités en priorité.

1. **Traitement du risque**

Mise en œuvre des **mesures de contrôle** :

* **Préventives** pour éviter le risque (ex. mise en place de sauvegardes).
* **Détectives** pour identifier la survenue (ex. scripts SNMP de monitoring).
* **Correctives** pour réduire les conséquences (ex. bascule automatique).

Planification et assignation de **plans d’action** SLA-driven pour structurer la remédiation.

1. **Surveillance et revue**

Suivi continu des indicateurs (KPIs, tableaux de bord, résultats SNMP) et réévaluation périodique des risques (évaluations formelles, audits). Permet d’ajuster les contrôles et de vérifier l’efficacité des plans d’action.

**11 Conclusion**

Cette synthèse montre comment SentinelRisk couvre l’ensemble du cycle de gestion du risque : de l’identification et de l’analyse des menaces (y compris la collecte automatique via SNMP), à l’évaluation avec calcul de score, au traitement par contrôles et plans d’action SLA-driven, puis à la surveillance et à la revue en temps réel et périodique via des tableaux de bord interactifs. Cette approche intégrée garantit un pilotage continu et proactif des risques et de la conformité.

CHAPITRE 3

ANALYSE ET SPÉCIFICATION DES BESOINS

**2.1 Introduction**

L’objectif de ce chapitre est de définir précisément ce que la plateforme SentinelRisk doit accomplir, en identifiant les acteurs, les cas d’usage, et les exigences qui vont guider sa conception et son implémentation. Nous y présenterons d’abord la liste des parties prenantes (acteurs), puis nous formulerons les besoins fonctionnels (ce que doit faire le système) et non‐fonctionnels (contraintes de qualité). Enfin, nous exposerons la méthodologie de développement retenue pour mener à bien le projet.

**2.2 Spécification des besoins**

La spécification des besoins est la première étape dans le processus de développe- ment d’une application informatique. Ainsi, nous présenterons dans cette section les besoins fonctionnels et non fonctionnels de notre application.

**2.2.1 Identification des acteurs**

**Administrateur système (admin) :**

L’administrateur système (admin) est responsable de la gestion des comptes et des rôles au sein de Keycloak. Il assure également la configuration des référentiels normatifs ainsi que des critères de scoring, garantissant ainsi un cadre sécurisé et conforme pour l’ensemble du système.

**Gestionnaire des risques (risk\_manager):**

Le gestionnaire des risques (risk\_manager) est chargé de créer et de modifier les fiches de risque. Il a également pour mission de lancer les évaluations associées et de suivre l’avancement des plans de mitigation afin de réduire l’exposition aux menaces identifiées.

**Responsable conformité (compliance\_officer)**

Le responsable conformité (compliance\_officer) définit les mappings vers les cadres réglementaires tels qu’ISO 27001 ou le RGPD. Il supervise également les contrôles et les audits de conformité afin de garantir le respect des exigences légales et normatives.

**Auditeur (auditor)**

L’auditeur (auditor) consulte les rapports, exporte les tableaux de bord et vérifie la traçabilité des actions et des changements. Son rôle est essentiel pour assurer une surveillance indépendante et garantir l’intégrité du système.

**Utilisateur standard (user)**

L’utilisateur standard (user) peut saisir des incidents et consulter son propre plan de mitigation ainsi que les actions qui le concernent. Il participe ainsi activement à la gestion des risques en signalant les problèmes et en suivant les mesures correctives mises en place.

**2.2.2 Les besoins fonctionnels**

Les besoins fonctionnels se sont les fonctionnalités qui seront accomplies par l’application en faveur de ces utilisateurs. Ce sont les services offerts par notre système à l’ensemble de ses acteurs primaires. Notre application offrira les fonctionnalités suivantes :

**Gestion des utilisateurs et des rôles**

L’application doit intègre une console de gestion des comptes construite sur Keycloak : les administrateurs peuvent créer, modifier ou désactiver des profils utilisateurs, réinitialiser les mots de passe et consulter l’historique des connexions. Chaque compte est associé à un ou plusieurs rôles (ADMIN, RISK\_MANAGER, COMPLIANCE\_OFFICER, AUDITOR, USER), qui déterminent l’accès aux différentes sections de l’application.

**Catégorisation des risques**

Les utilisateurs habilités peuvent structurer le référentiel de risques en créant, renommant ou supprimant des catégories et des sous-catégories hiérarchiques. Cette arborescence facilite le classement des risques selon leur nature (opérationnel, sécurité, conformité, financier) et permet de visualiser d’un coup d’œil la répartition des risques par domaine, grâce à un affichage en arbre dynamique.

**Identification et évaluation des risques**

L’application doit permettre l’ajout de nouveaux risques en saisissant les informations essentielles (nom, description, catégorie, impact, probabilité). Le score doit être calculé automatiquement pour aider à la priorisation.

**Plan de mitigation et contrôles associés**

Pour chaque risque, la plateforme doit suggère des types de contrôles — préventifs, détectifs ou correctifs — et propose de les créer ou de les lier au risque. Un module de suivi affiche l’état d’avancement de chaque contrôle (planned, in\_progress, implemented) et permet d’assigner un responsable, de fixer des dates d’implémentation et d’enregistrer des preuves ou notes de test, assurant ainsi le pilotage précis du plan de mitigation.

**Mapping conformité**

Le système doit embarquer les référentiels normatifs (ISO 27001, NIST, GDPR, SOC 2) et leurs exigences, et offrire une interface pour associer chaque risque et contrôle aux exigences correspondantes. Une analyse d’écarts doit automatiser l’identification des non-conformités, génèrer des rapports de conformité détaillés (PDF, CSV) et proposer des actions correctives lorsqu’un écart est détecté.

**Tableaux de bord et alertes**

Le Dashboard interactif doit regrouper des KPI et graphiques (nombre de risques par statut, taux de conformité, avancement des plans, santé SNMP) avec des filtres dynamiques (période, framework, rôle). Des notifications push et des alertes par email informent les utilisateurs dès qu’un seuil critique est franchi (risque à score élevé, échéance de contrôle dépassée, actif SNMP en anomalie), garantissant une surveillance continue et proactive.

**2.2.3 Besoins non fonctionnels**

Les besoins non fonctionnels sont des critères très importants pour assurer la qualité des services assurés par notre plateforme et garantir la satisfaction de tous les utilisateurs. Ils sont présentés dans les points suivants :

**Sécurité**

* Authentification et autorisation via Keycloak
* Protection des APIs (HTTPS, CORS, CSRF)

**Performance & Scalabilité**

* Temps de réponse < 300 ms sur requêtes courantes
* Support de plusieurs centaines d’utilisateurs concurrents

**Maintenabilité**

* Code modularisé (Spring services, Angular modules)
* Tests unitaires (> 70 % de couverture)

**Interopérabilité**

* APIs REST conformes aux standards JSON
* Possibilité d’intégration future (webhooks, SIEM)

**Ergonomie & Accessibilité**

* Interface responsive (desktop, tablette, mobile)
* Conformité WCAG 2.1 niveau AA
  1. **Architecture de la solution**

Devant la diversité des architectures logicielles existantes, le choix de l’une d’entre elles peut s’avérer complexe. Il est donc essentiel d’analyser les principales approches architecturales proposées dans le développement d’applications, telles que l’architecture MVC (Model-View-Controller), MVVM (Model-View-ViewModel), ou encore l’architecture hexagonale. Cette étude comparative nous permettra d’identifier l’architecture la plus adaptée à notre solution, en tenant compte de ses exigences techniques, de sa maintenabilité et de son évolutivité.

|  |  |  |  |
| --- | --- | --- | --- |
| **Architecture** |  |  |  |
|  | **Description** | **Points forts** | **Point faibles** |
| MVC | L’architecture MVC sépare une application en trois couches :  Modèle : gère les données et la logique métier  Vue : affiche les données à l’utilisateur  Contrôleur : traite les actions de l’utilisateur et met à jour le modèle ou la vue  Cette séparation permet de rendre le code plus clair, plus maintenable et facilite le travail en équipe. | **Séparation claire des responsabilités**  **Facilité de test unitaire**  **Flexibilité dans l’évolution de l’interface**  **Travail en équipe facilité**  **Bonne structure pour les applications web** | **Complexité accrue pour les petites applications**  **Lien étroit entre le contrôleur et la vue**  **Courbe d’apprentissage pour les débutants**  **Multiplication des fichier** |
| MVVM | L’architecture MVVM organise une application en trois couches :  Modèle : contient les données et la logique métier  Vue:interface utilisateur  ViewModel : fait le lien entre la vue et le modèle, en exposant les données de manière adaptée à l’interface, souvent via des mécanismes de liaison automatique (data binding)  MVVM est très utilisé dans les applications avec interfaces réactives (comme Flutter, Android, WPF…). | **Séparation claire**  **Data binding**  **Facilite les tests unitaires**  **Très adapté aux interfaces interactives ou réactives** | **Complexité accrue pour les petits projets**  **Courbe d’apprentissage plus élevée (surtout pour la gestion du binding)**  **Peut entraîner une surcharge de code dans le ViewModel si mal structuré** |

**4.2.1 Architecture choisie**

On a eu recours au modèle MVC, ce qui signifie (modèle, vue et contrôleur). Ce modèle représente une architecture à trois couches utilisées pour la programmation client/serveur et d’interface graphique. C’est un modèle architectural très puissant qui intervient dans la réalisation d’une application. Il tire sa puissance de son concept de base qui est la séparation des données (modèle), de l’affichage(vue) et des actions (contrôleur). C’est trois couches sont décrites comme suit :

**4.2.1.1 Modèle**

Il correspond aux données stockées généralement dans une base de données. Dans un langage orientée objet ces données sont exploitées sous forme de classes. Le modèle peut aussi agir sur la vue en mettant à jour ses données.

**4.2.1.2 Vue**

Ne contenant que les informations liées à l’affichage, la vue se contente d’afficher le contenu qu’elle reçoit sans avoir connaissance des données. En bref, c’est l’interface homme machine de l’application.

**4.2.1.3 Contrôleur**

Le contrôleur sert de base à récupérer les informations, de les traiter en fonction des paramètres demandés par la vue (par l’utilisateur), puis de renvoyer à la vue les données afin d’être affichées. C’est donc l’élément qui va utiliser les données pour les envoyer à la vue.

L’interaction entre ces trois couches est décrite à l’aide de la figure suivante :

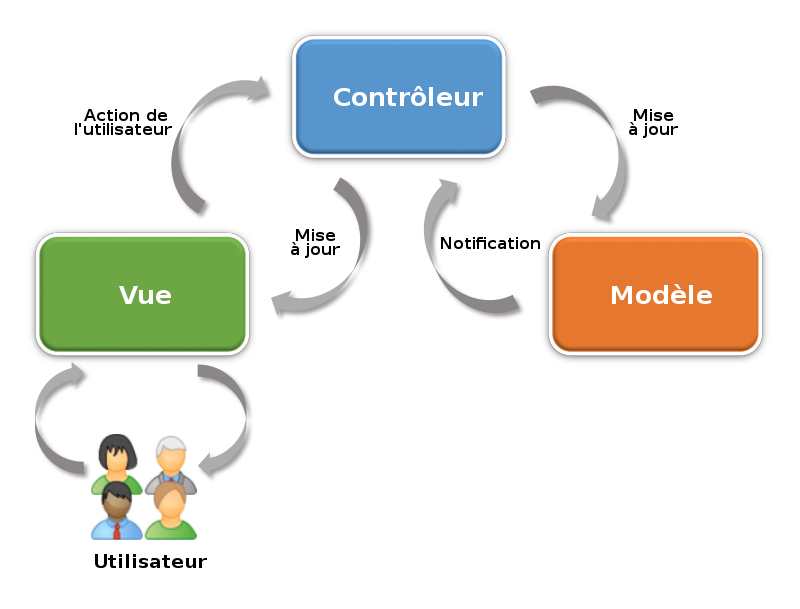


Figure 4.1 – Modèlé MVC

Les avantages apportés par l’architecture MVC sont :

— La séparation des données de la vue et du contrôleur (ce qui permet une conception claire et efficace de l’application).

— Une indépendance des données, de l’affichage et des actions (ce qui donne plus de souplesse pour la maintenabilité et l’évolutivité du système).

— Un gain de temps de maintenance et d’évolution de l’application.[5]

**2.4 Conclusion**

Dans ce chapitre, nous avons d’abord identifié les différents acteurs interagissant avec le système ainsi que les besoins fonctionnels et non fonctionnels de notre application. Nous avons ensuite retenu l’architecture la plus adaptée pour répondre à ces exigences.

Dans le chapitre suivant, nous entrerons dans la phase de conception UML de SentinelRisk, en réalisant les diagrammes de cas d’utilisation, de classes, de séquence et de déploiement.

CHAPITRE 3

ETUDE CONCEPTUELLE

**3.1 Introduction**

Dans le chapitre précédent, nous avons identifié les acteurs et défini les caractéristiques clés de la solution. Nous entrons désormais dans la phase de conception, essentielle au développement de notre application. Nous montrerons des diagrammes de cas d’utilisation généraux et détaillés, des diagrammes de classes et des diagrammes de séquence. Ces outils traduiront les exigences des utilisateurs en structures et interactions concrètes, jetant ainsi les bases du développement de nos solutions.

**3.2 Définition UML**

UML a été choisi comme principal langage de modélisation car il est bien adapté aux projets orientés objet. Ce choix présentait plusieurs avantages : il facilitait grandement la compréhension et la communication autour des concepts orientés objet, et il est aujourd’hui devenu un standard largement adopté par les acteurs majeurs de l’industrie du logiciel.

Nous avons choisi trois types de base de diagrammes UML pour modéliser notre système :

— **Les diagrammes de cas d’utilisation**

— **Les diagrammes de classes**

— **Les diagrammes de séquence**

— **diagramme de déploiement**

Dans la section suivante, nous présentons les diagrammes de cas d’utilisation en détail, décrivant brièvement chaque cas. Ensuite, nous discutons du diagramme de classes pour avoir un aperçu du système. Enfin, nous explorons le diagramme de séquence en présentant des scénarios associés. Pour conclure, nous incluons le diagramme de déploiement, illustrant l’architecture physique du système et la distribution des composants logiciels sur les différents nœuds matériels.[4]

A logo for a company

AI-generated content may be incorrect.

Figure 3.1 – UML

**3.3 Conception**

Dans cette phase de développement, nous passons de la phase de spécification à la conception concrète du système. Cette étape est cruciale car elle crée les bases sur lesquelles repose l’ensemble du projet. En regardant les besoins spécifiques des utilisateurs et en considérant les limitations techniques, nous pouvons développer une architecture robuste et fonctionnelle qui répond aux attentes et aux exigences du projet.

**3.3.1 Diagramme des cas d’utilisation**

Un diagramme de cas d’utilisation décrit les besoins et les interactions des utilisateurs avec le système. Il donne un aperçu des fonctions que le système doit fournir, en se concentrant sur les activités et les objectifs de l’utilisateur. Il s’agit d’un outil essentiel pour comprendre et visualiser les fonctions de base du système du point de vue de l’utilisateur

A diagram of a diagram

AI-generated content may be incorrect.

Figure 3.2 – Diagramme des cas d’utilisation globale

**3.3.1.1 Diagramme des cas d’utilisation «Gérer utilisateur»**

Le diagramme des cas d’utilisation «Gérer utilisateur» présente les différentes interactions entre l’administrateur du système et les fonctionnalités associées à la gestion des utilisateurs. L’administrateur peut créer, modifier supprimer des comptes utilisateurs, assurant ainsi que seuls les utilisateurs autorisés ont accès au système. Ce diagramme permet de comprendre les rôles et les actions spécifiques que l’administrateur peut effectuer pour maintenir et contrôler la base des utilisateurs.

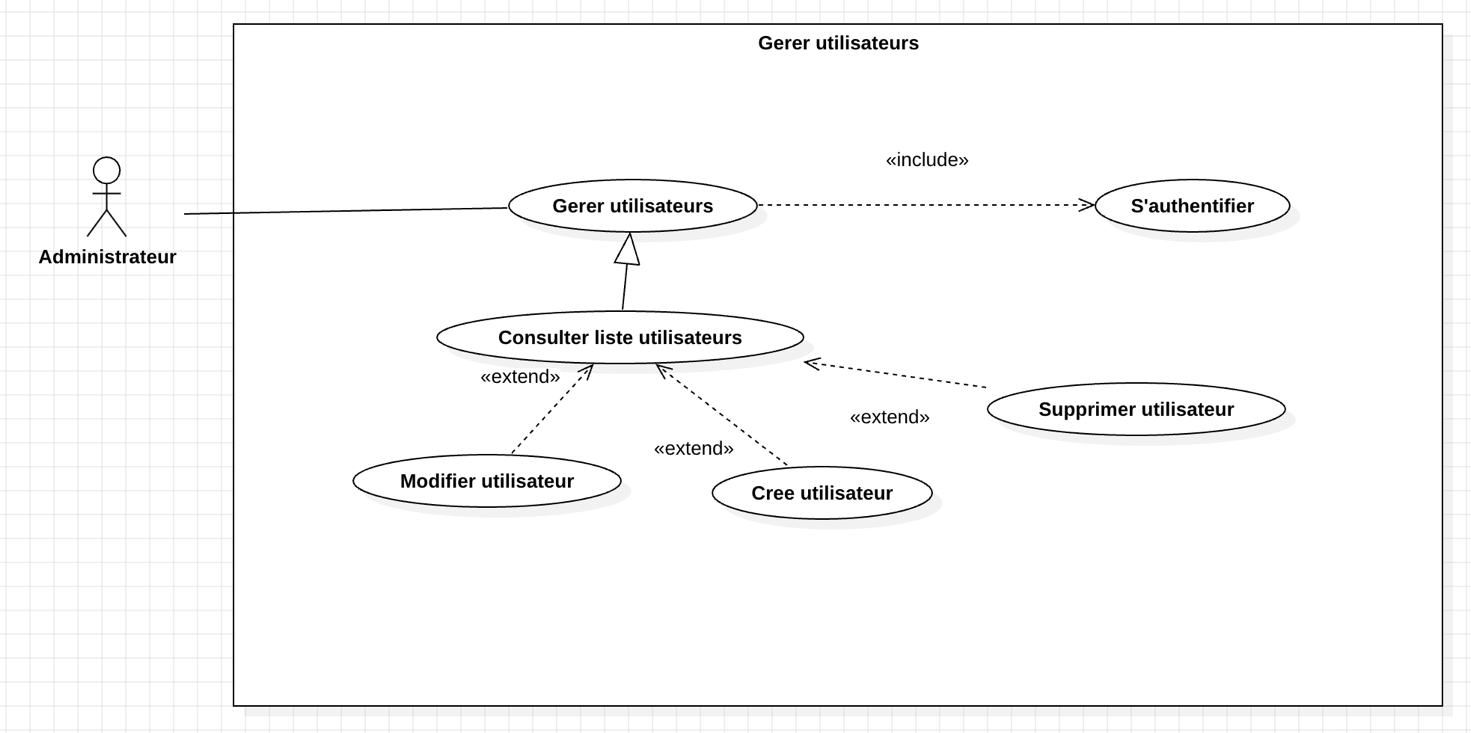


Figure 3.3 – Diagramme des cas d’utilisation «Gérer utilisateur»

**3.3.1.2 Diagramme des cas d’utilisation «Gérer risques»**

Le diagramme des cas d’utilisation « Gérer risques » décrit comment le gestionnaire de risques interagit avec le système pour administrer l’intégralité du référentiel de risques :

* Créer, modifier et supprimer un risque
* Lister tous les risques existants
* Importer en masse les risques à partir d’un fichier Excel préformaté
* Télécharger le modèle Excel afin de respecter les critères et listes de sélection obligatoires

Chacune de ces opérations inclut le cas d’usage « S’authentifier », garantissant que seules des personnes dûment identifiées et autorisées peuvent accéder à ces fonctionnalités.

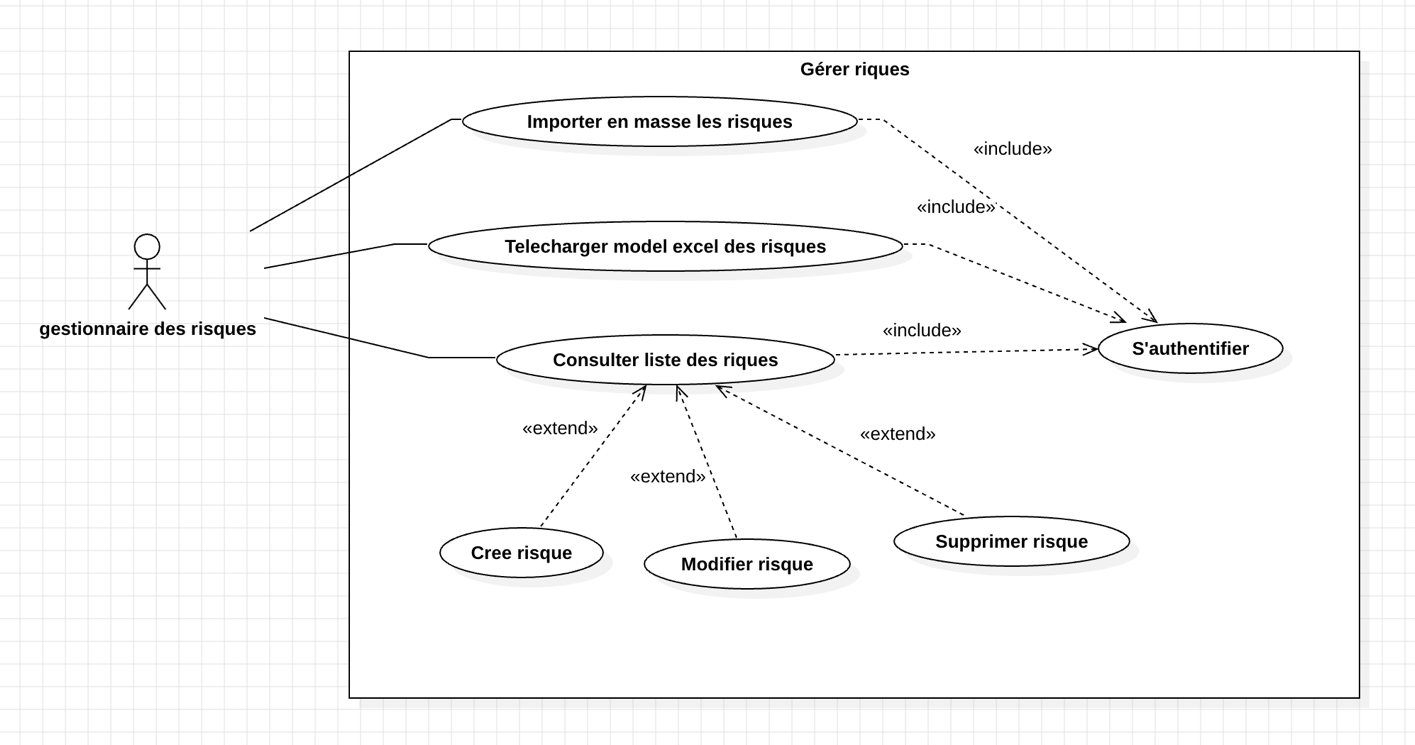


Figure 3.4 – Diagramme des cas d’utilisation «Gérer risques»

**3.3.1.3 Diagramme des cas d’utilisation «**Gérer l’agent SNMP**»**

Le diagramme de cas d’utilisation **«** **Gérer l’agent SNMP »** illustre l’ensemble des interactions entre le Gestionnaire des risques et le module SNMP de SentinelRisk. Il regroupe quatre opérations principales :

* **Gérer les actifs SNMP** (ajout, modification, suppression et listing des équipements à scanner)
* **Configurer l’agent SNMP** (définition des OID, des seuils et de la périodicité des analyses)
* **Lancer un scan SNMP** (exécution manuelle d’une session de mesure et remontée des résultats en temps réel)
* **Consulter l’historique des scans** (visualisation, détail d’un scan, création de risque à partir des données)

Chacune de ces actions inclut le cas d’usage « S’authentifier », garantissant que seules les personnes autorisées peuvent piloter et exploiter l’agent SNMP.

A diagram of a diagram

AI-generated content may be incorrect.

Figure 3.5 – Diagramme des cas d’utilisation « Gérer l'agent SNMP »

**3.3.1.4 Diagramme des cas d’utilisation «** **Gérer les conformités »**

Le cas d’utilisation **« Gérer les conformités »** regroupe l’ensemble des activités dont est chargée la **Responsable conformité** au sein de SentinelRisk. Ce workflow commence toujours par **S’authentifier** (via Keycloak), puis permet à l’acteur de :

* + **Gérer les frameworks normatifs**
  + Créer, modifier, supprimer ou consulter les référentiels (ISO 27001, NIST, SOC 2…) dans lesquels les exigences de conformité sont définies.
  + **Gérer les exigences**
  + Ajouter, mettre à jour ou retirer les exigences propres à chacun de ces frameworks (par exemple des contrôles ou règles à appliquer).
  + **Réaliser une analyse d’écarts**
  + Lancer la comparaison entre les contrôles existants et les exigences normatives pour identifier automatiquement les points de non-conformité.
  + **Gérer les mappings risques ↔ exigences**
  + Associer chaque risque identifié aux exigences correspondantes, préciser leur statut (Conforme, Non conforme, Partiellement conforme) et joindre les preuves justificatives.
  + **Consulter le dashboard conformité**
  + **Générer des rapports de conformité**
  + Exporter les résultats de l’analyse (tables, graphiques, synthèses) au format PDF ou CSV pour les audits et les revues de gouvernance.

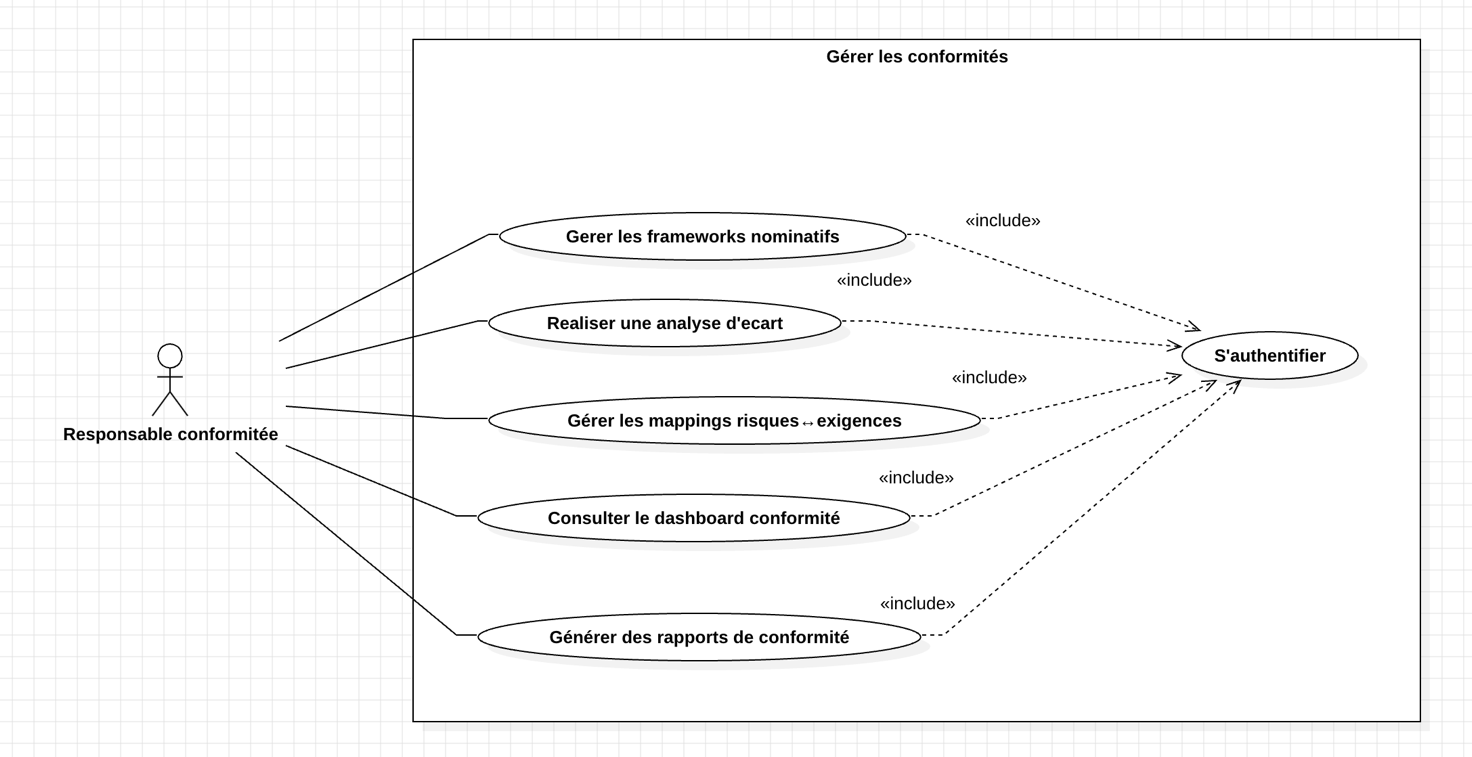


Figure 3.6 – Diagramme des cas d’utilisation « Gérer les conformités »

**3.3.1.5 Diagramme des cas d’utilisation « Générer des rapports »**

Le cas d’utilisation **« Générer des rapports »** décrit comment l’**Auditeur**, une fois **authentifié** (cas inclus « S’authentifier »), construit et exporte ses rapports :

1. **Générer rapports**

– Point d’entrée unique auquel l’Auditeur se connecte pour lancer la création d’un document.

1. **Étapes optionnelles** (symbolisées par « extend »)

* **Sélectionner le type de rapport** : choisir parmi conformité, remédiation, évaluation de risques, SNMP, ou rapport global.
* **Définir la période** : préciser une plage de dates ou une période prédéfinie (aujourd’hui, semaine, mois).
* **Appliquer filtres & critères** : affiner par référentiel, statut, responsable ou autres dimensions métier.
* **Personnaliser le contenu** : inclure/exclure sections, choisir le niveau de détail et le format (PDF, CSV, Excel).
* **Aperçu avant export** : visualiser un rendu préliminaire pour valider la mise en page et les données.
* **Exporter & télécharger** : générer le fichier final et proposer son téléchargement.

Chacune de ces extensions enrichit le flux principal **« Générer rapports »**, tout en restant optionnelle dans l’enchaînement métier.

A diagram of a gender report

AI-generated content may be incorrect.

Figure 3.7 – Diagramme des cas d’utilisation «Générer des rapports»

**3.3.1.6 Diagramme des cas d’utilisation «Gérer les plans d’action»**

Le diagramme des cas d’utilisation **« Gérer les plans d’action »** décrit les différentes interactions et fonctionnalités offertes au **Gestionnaire des risques** pour administrer ses plans d’action : consultation de la liste et des détails, création, modification, suppression et mise à jour du statut, le tout après authentification.

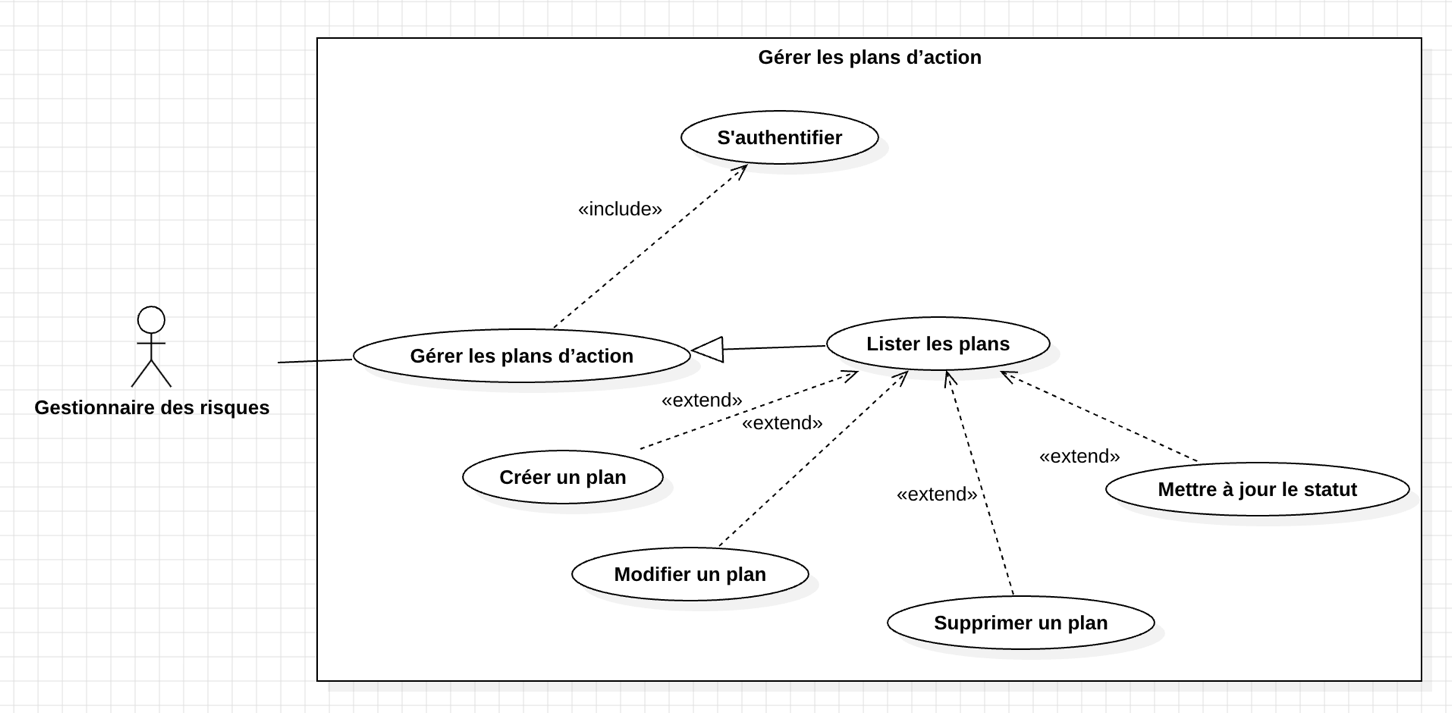


Figure 3.8 – Diagramme des cas d’utilisation « Gérer les plans d’action »

**3.3.1.7 Diagramme des cas d’utilisation « Consulter le Dashboard »**

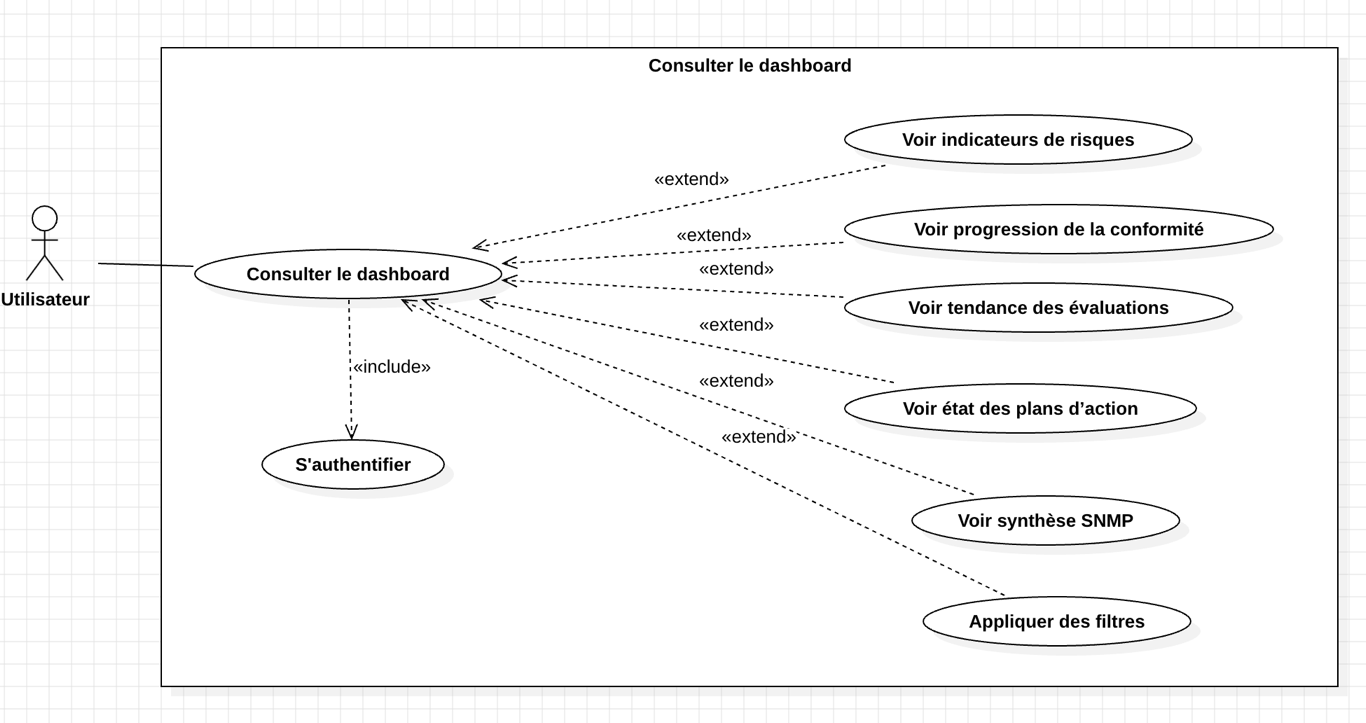
Le diagramme des cas d’utilisation « Consulter le Dashboard » montre les interactions spécifiques permettant aux utilisateurs (Gestionnaire des risques, Responsable conformité, Auditeur) d’accéder aux indicateurs clés de risques, à la progression de la conformité, aux tendances d’évaluation, à l’état des plans d’action et à la synthèse SNMP, le tout après authentification.

Figure 3.9 – Diagramme des cas d’utilisation «Consulter le Dashboard»

**3.3.2 Diagramme de classes**

Ce diagramme est souvent considéré comme central dans la modélisation objet. Il présente une vue statique des composants du système, y compris leurs propriétés et méthodes, ainsi que les interactions et relations entre ces éléments.

A diagram of a computer

AI-generated content may be incorrect.

Figure 3.10 – Diagramme de classes

**3.3.3 Diagramme de séquence**

Le digramme de séquence permet de modéliser la dynamique de l’application et d’identifier les objets, les messages entre ces objets et leurs ordres. En fait, focalise sur l’enchaînement du message. Pour bien comprendre la dynamique de notre système nous proposons les digrammes de séquence illustrés dans les figures suivantes.

**3.3.3.1 Diagramme de séquence «s’authentifier»**

Le diagramme de séquence «S’authentifier» illustre le processus d’authentification des utilisateurs, montrant les interactions entre l’utilisateur , le système et les éventuels services externes .Ce diagramme permet de comprendre les étapes nécessaires pour vérifier l’identité d’un utilisateur et accéder au système de manière sécurisée.

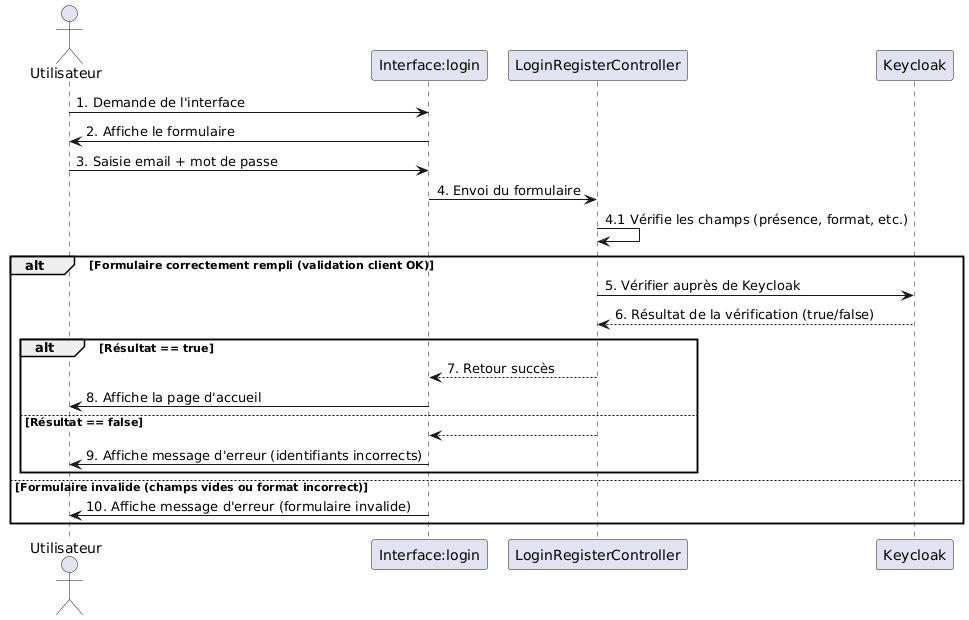


Figure 3.12 – Diagramme de séquence «s’authentifier» action verifier etpa 4

**3.3.3.2 Diagramme de séquence «Créer utilisateur»**

Le diagramme de séquence «Créer utilisateur» présente les interactions et les étapes impliquées dans la validation d’un compte utilisateur pour la connexion.

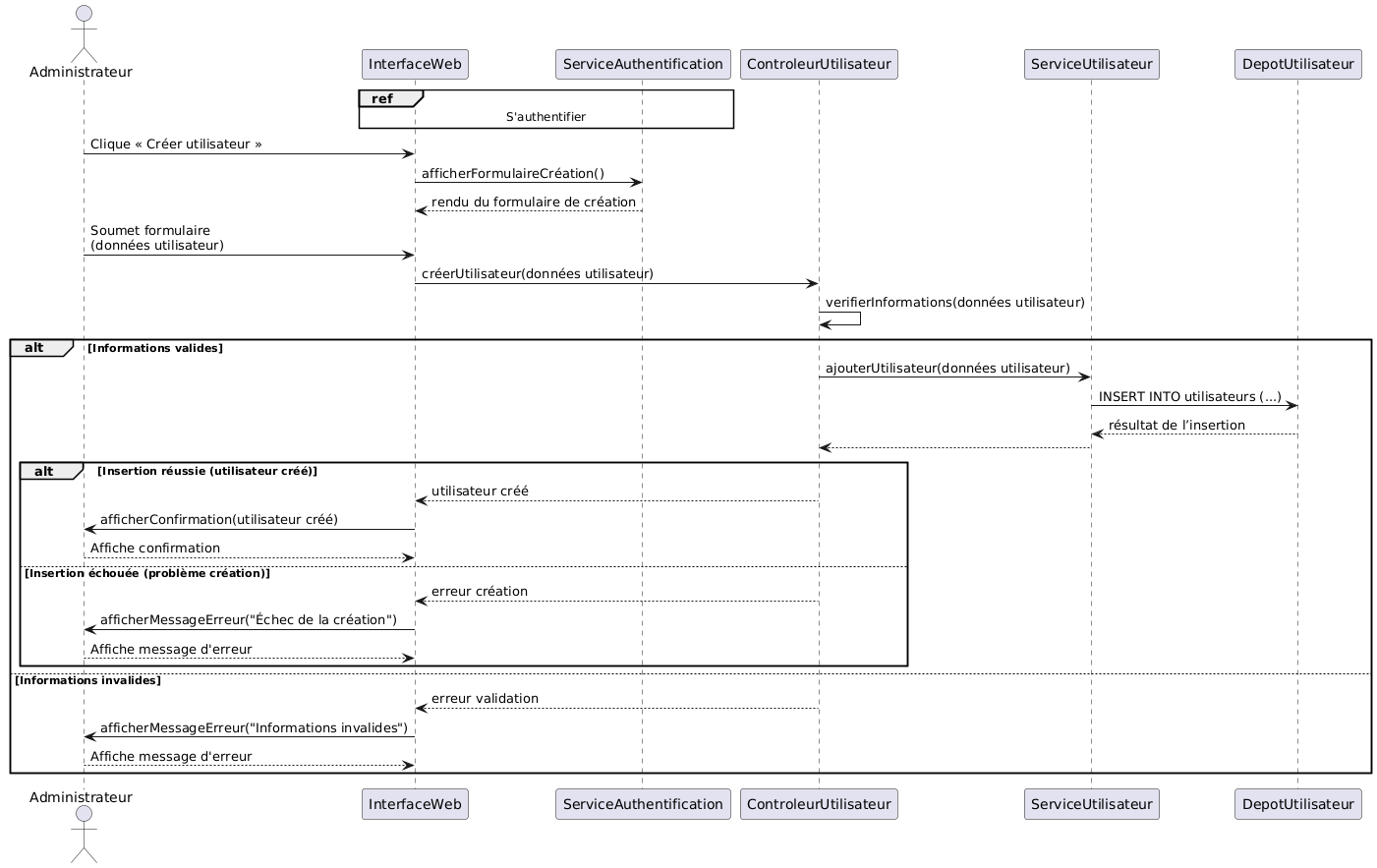


Figure 3.13 – Diagramme de séquence «Créer utilisateur»

**3.3.3.3 Diagramme de séquence «Modifier un utilisateur»**

Le diagramme de séquence «Modifier un utilisateur» décrit le processus de modification des information d’un utilisateur .Offrant une vue claire des étapes impliquées et des interactions entre l’utilisateur et le système.

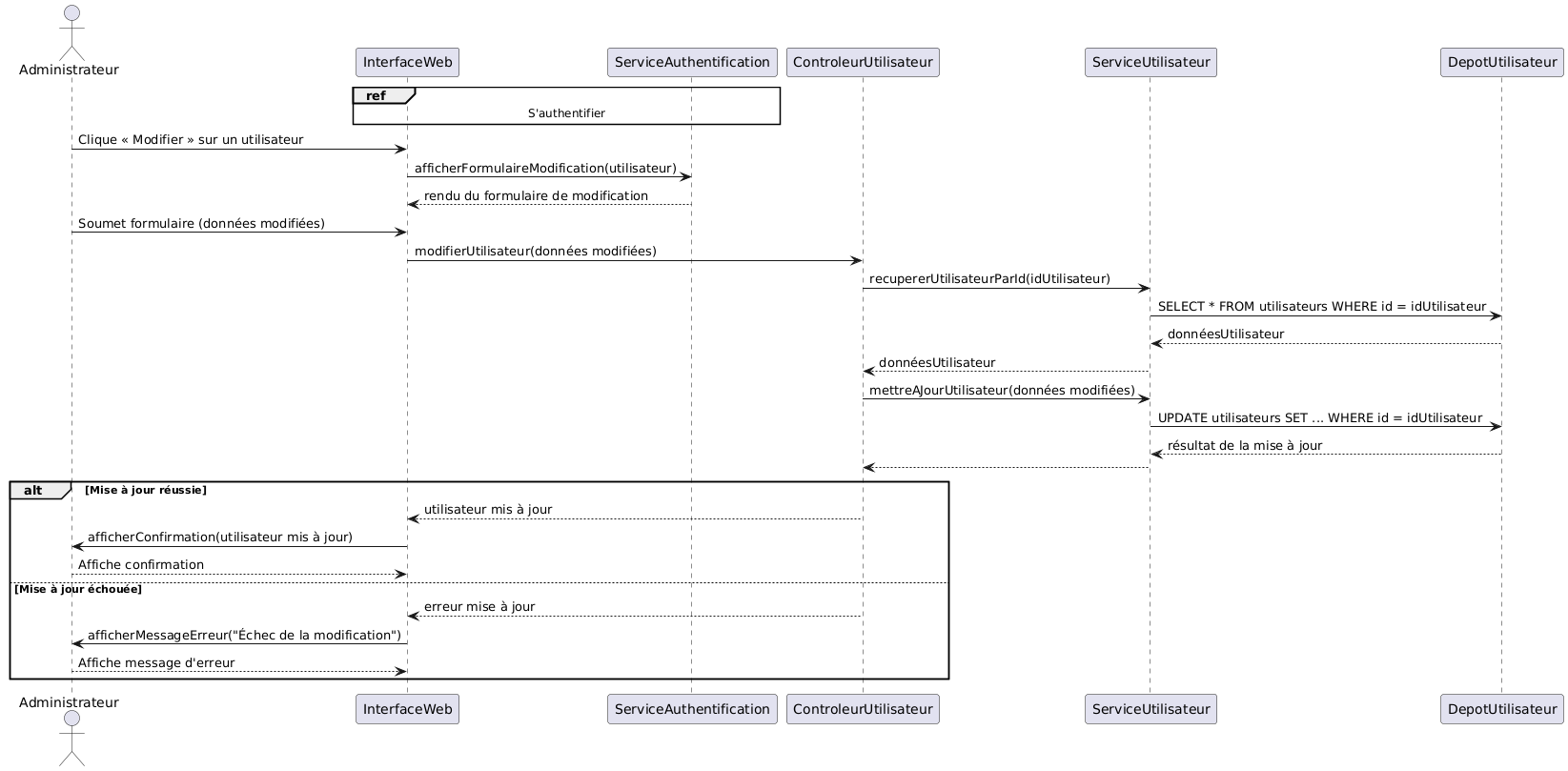


Figure 3.14 – Diagramme de séquence «Modifier un utilisateur»

**3.3.3.4 Diagramme de séquence «Suppression d’un utilisateur»**

Le diagramme de séquence «Suppression d’un utilisateur» illustre le flux d’interactions nécessaires pour qu’un utilisateur modifie son annonce existante. Il présente les étapes et les messages échangés entre le système et l’utilisateur pour effectuer des modifications sur une annonce, en assurant que les changements sont correctement enregistrés et mis à jour.

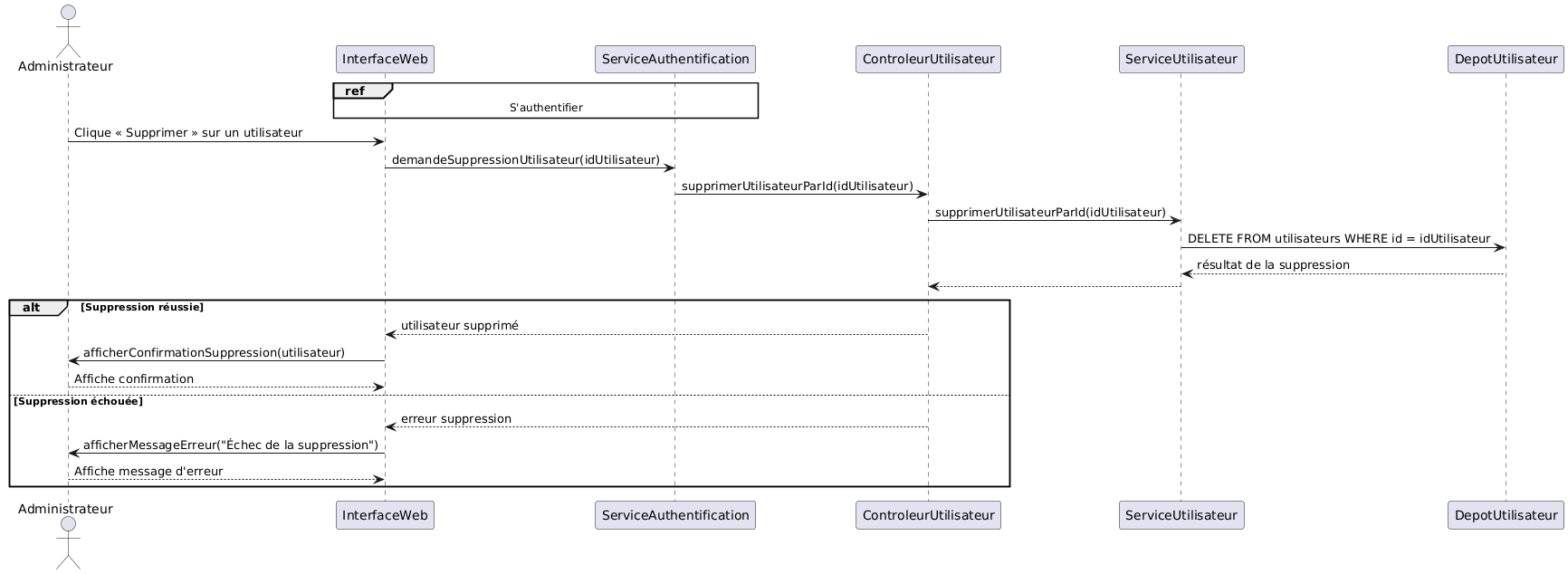


Figure 3.15 – Diagramme de séquence «modifier annonce»

**3.3.3.5 Diagramme de séquence « Lancer scan SNMP »**

Le diagramme de séquence « Lancer scan SNMP » décrit, de manière succincte, comment l’administrateur s’authentifie, déclenche un scan SNMP via l’interface web, et récupère ensuite les résultats affichés à l’écran.

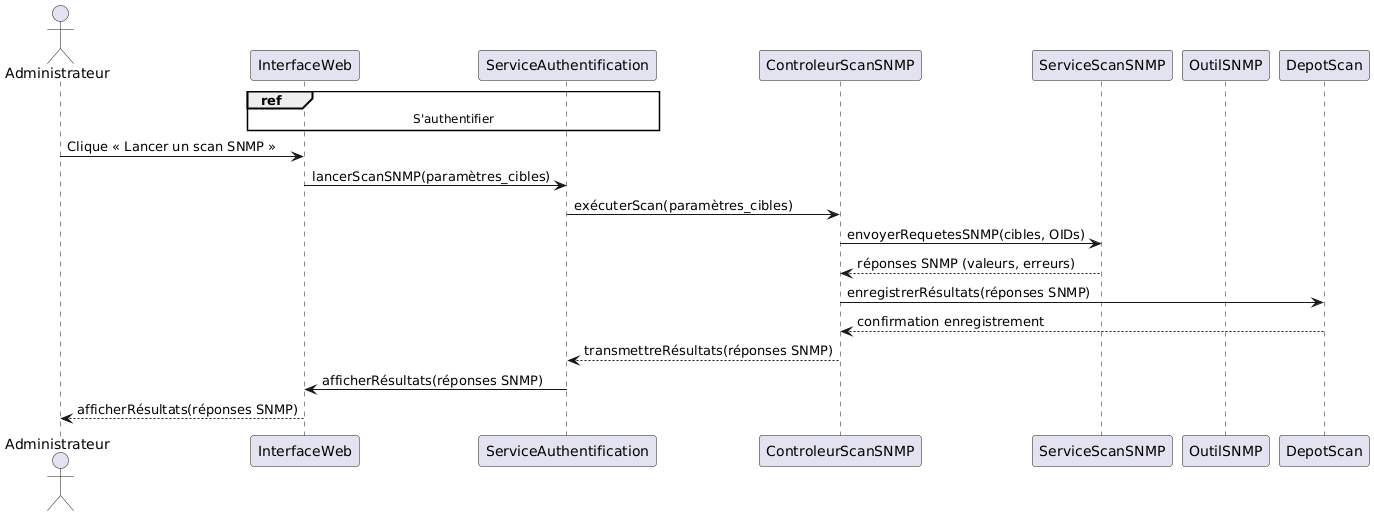


Figure 3.16 – Diagramme de séquence « Lancer scan SNMP »

**3.3.3.6 Diagramme de séquence « Consulter Dashboard »**

Le diagramme de séquence « Consulter Dashboard » montre brièvement comment, après authentification, l’administrateur ouvre la page du tableau de bord et obtient en retour les données agrégées : nombre d’utilisateurs, nombre de risques, résultats du dernier scan SNMP, données de mapping et plans d'action affichés à l'écran.

 Figure 3.17 – Diagramme de séquence « Consulter Dashboard »

**3.3.3.7 Diagramme de séquence « générer rapport»**

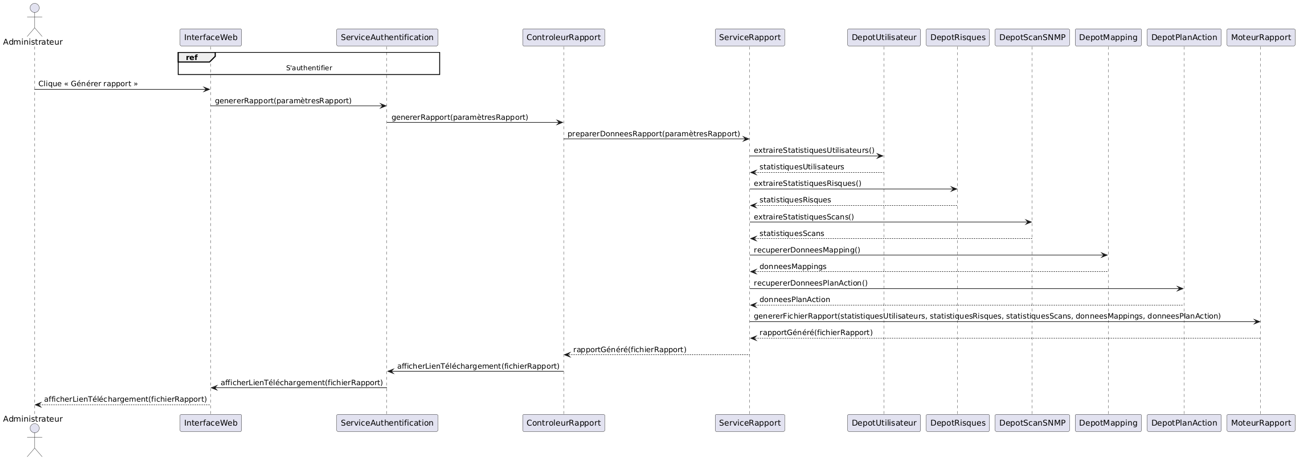
Le diagramme de séquence « générer rapport » montre comment l’administrateur, après authentification, déclenche la création d’un rapport. On y voit le contrôleur demander au service les données (statistiques utilisateurs, risques, résultats de scan, données de mapping et plans d’action), puis le service invoquer le moteur pour générer le fichier. Enfin, le lien de téléchargement du rapport est retourné à l’interface pour l’affichage à l’administrateur.

Figure 3.18 – Diagramme de séquence « générer rapport »

**3.3.3.8 Diagramme de séquence « crée plan d’action »**

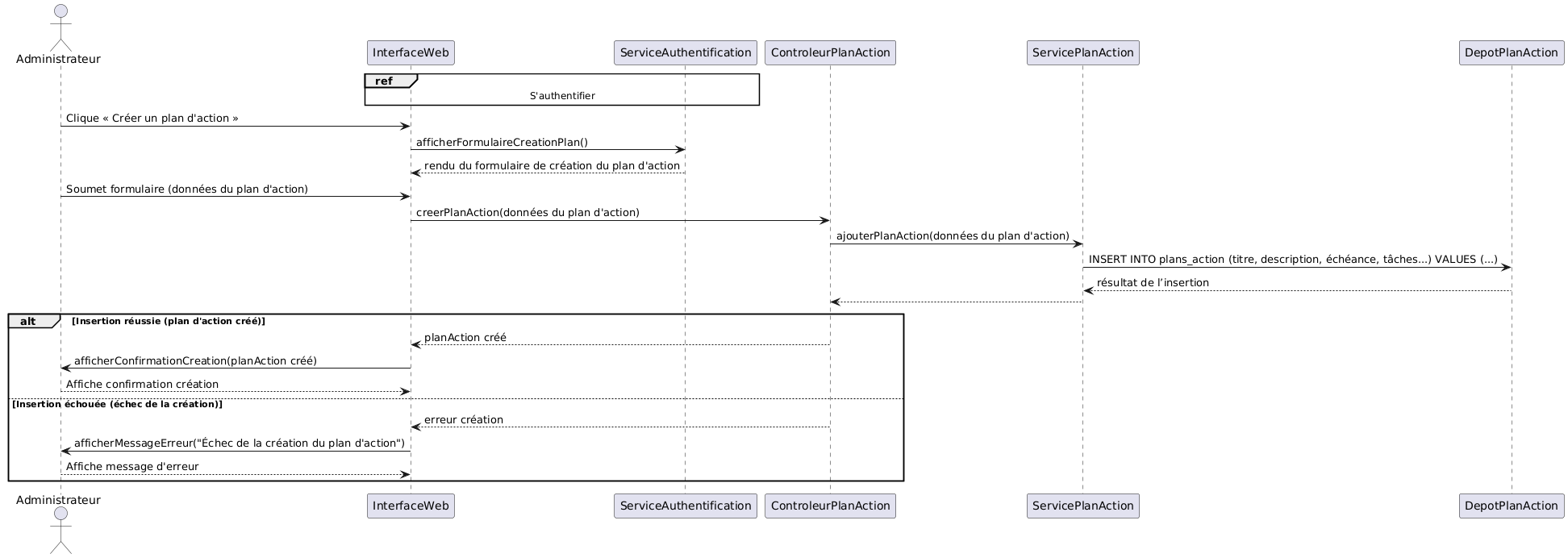
Le diagramme de séquence « Créer plan d’action » montre les échanges lorsqu’un administrateur authentifié initialise la création d’un nouveau plan d’action. On y voit l’administrateur cliquer pour ouvrir le formulaire, la référence d’authentification se déclencher, le formulaire s’afficher, puis l’administrateur soumettre les données du plan. Le contrôleur transmet ces données au service, qui tente d’insérer le plan dans le dépôt. Enfin, selon que l’insertion réussisse ou non, l’interface affiche soit une confirmation de création, soit un message d’erreur.

Figure 3.19 – Diagramme de séquence « Crée plan d’action »

**3.3.4 diagramme de déploiement**

Le diagramme de déploiement décrit l’architecture physique du système en montrant comment les composants logiciels sont déployés sur le matériel. Il met en évidence les nœuds matériels et les artefacts logiciels ainsi que les connexions entre eux. Ce diagramme est crucial pour comprendre l’infrastructure technique nécessaire au bon fonctionnement du système et pour planifier la répartition des ressources et des responsabilités.

A diagram of a computer process

AI-generated content may be incorrect.

Figure 3.21 – Diagramme de séquence « diagramme de déploiement »

**3.4 Conclusion**

Dans la phase de conception, la transition de la spécification vers la réalisation concrète du système est fondamentale pour établir des bases solides. Les diagrammes de cas d’utilisation, de classe, de séquence et de déploiement se révèlent des outils incontournables dans cette démarche, offrant une vision détaillée des interactions entre utilisateurs et système, de la structure interne des composants, ainsi que des séquences d’actions. Dans le chapitre suivant, nous aborderons la phase de réalisation, où nous passerons de la conception théorique à l’implémentation pratique du projet.

CHAPITRE 4

RÉALISATION

**4.1 Introduction**

Dans cette nouvelle phase du projet, nous entrons dans l’étape décisive du développement. Pour en assurer le succès, il convient de définir précisément l’environnement de travail, les méthodes et les outils qui seront mobilisés. Ce chapitre débute par la présentation de l’environnement du projet (infrastructures, langages, frameworks et outils de gestion), puis offre un aperçu des principales fonctions et des modules qui composeront le système en cours de développement.

**4.3 Environnement du projet**

**4.3.1 Environnement matériel**

Nous allons utiliser un ordinateur portable dont les caractéristiques sont présentées dans le Tableau

.

|  |  |
| --- | --- |
| **Ordinateur :** | **Macbook pro:** |
| Propriétaires | Omar Msaad |
| Processeur | Apple silicone m1 pro |
| RAM | 16,0Go |
| Disque Dur | 1000GB SSD |
| Système d’exploitation | Macos |
| Carte graphique | Apple silicone m1 pro |

Table 4.1 – Environnement matériel

**4.3.2 Environnement logiciel**

Nous allons utiliser les frameworks et langages de dévelopement présentées par les tableaux suivantes.

|  |  |  |
| --- | --- | --- |
| **Environnement** | **Présentation** | **Logo** |
| Star uml | StarUML est un outil de génie  logiciel dédié à la modélisation  UML et édité par la société coréenne MKLabs. Il est  multiplateforme et fonctionne sous  Windows, Linux et MacOS.[6] | A colorful star with a white background  AI-generated content may be incorrect. |
| Visual Paradigm | Visual Paradigm est un logiciel de diagrammes et de modélisation utilisé pour la conception de systèmes et de bases de données, ainsi que pour la gestion de projets et la gestion de la qualité.[7] | A black background with white text  AI-generated content may be incorrect. |
| GitHub | GitHub Desktop est une application gratuite et open source qui permet de gérer facilement ses dépôts GitHub depuis une interface graphique, sans utiliser la ligne de commande. Elle simplifie les opérations comme les commits, les branches, les fusions (merge) et les pull requests, ce qui la rend idéale pour les débutants comme pour les développeurs expérimentés..[8] | A purple cat in a circle  AI-generated content may be incorrect. |

|  |  |  |
| --- | --- | --- |
| Visual Studio | Visual Studio est un outil de développement puissant qui permet  d’effectuer l’ensemble du cycle de développement au même  endroit.[9] | A blue ribbon with a cross  AI-generated content may be incorrect. |
| PostgreSQL | PostgreSQL est un système de gestion de base de données  relationnelle et objet. C’est un outil  libre disponible selon les termes d’une licence de type BSD. Ce  système est comparable à d’autres systèmes de gestion de base de  données, qu’ils soient libres, ou  propriétaires.[10] | A blue elephant with white outline  AI-generated content may be incorrect. |
| Microsoft Word | Microsoft Word est un logiciel de traitement de texte développé par Microsoft, permettant de créer, modifier, mettre en forme et partager des documents professionnels.[11] |  |

|  |  |  |
| --- | --- | --- |
| HTML 5 | Il est utilisé afin de créer et de représenter le contenu d’une page web et sa structure.[12] | A logo with a shield and a letter s  AI-generated content may be incorrect. |
| CSS | Les feuilles de style en cascade, généralement appelées CSS de  l’anglais Cascading Style Sheets,  forment un langage informatique qui décrit la présentation des documents HTML et XML.[13] | A blue and white logo  AI-generated content may be incorrect. |
| JavaScript | JavaScript est un langage de programmation de scripts  principalement employé dans les  pages web interactives et à ce titre est une partie essentielle des applications web.[14] | A yellow and white logo  AI-generated content may be incorrect. |
| Java | Java est un langage de programmation de haut niveau orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy, présenté officiellement le 23 mai 1995 au SunWorld..[15] | Java (technique) — Wikipédia |
| TypeScript | TypeScript est un langage de programmation libre développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). [16] | A blue square with white letters  AI-generated content may be incorrect. |
| SQL | Le langage SQL (Structured Query  Language) est un langage informatique utilisé pour exploiter  des bases de données. Il permet de façon générale la définition, la  manipulation et le contrôle de sécurité de données.[17] | A blue logo with a blue cylinder  AI-generated content may be incorrect. |
| Spring Boot | Framework Java permettant de créer rapidement des applications backend autonomes. Il intègre un serveur web embarqué, la configuration automatique et de nombreux starters  .[18] | What is Spring Boot and use cases of Spring Boot? - DevOpsSchool.com |
| Spring Security + Keycloak | Spring Security gère l’authentification et l’autorisation côté backend, tandis que Keycloak fournit un serveur d’identité OIDC/JWT pour la gestion centralisée des utilisateurs et rôles. | Connaissez-vous Keycloak, la solution open source qui sécurise votre  système d'information? |
| Angular | Framework JavaScript/TypeScript pour construire des applications web monopages (SPA). Offre un système de composants, un routeur et une liaison de données bidirectionnelle. | Angular — Wikipédia |
| **Docker & Docker Compose** | Outil de conteneurisation permettant d’empaqueter chaque composant (base, Keycloak, backend, frontend, agent SNMP) dans des conteneurs isolés. Docker Compose orchestre leur exécution locale. | Docker-Compose | Wiki-Tech |
| **Maven** | Maven est un outil open source développé par la fondation Apache pour gérer et automatiser la compilation, la gestion des dépendances et l'intégration continue des projets Java. Il est couramment utilisé dans le développement logiciel, notamment pour les applications Java et Java EE. | Apache Maven — Wikipédia |
| **SNMP4J** | Aperçu IA  SNMP4J est une implémentation open source et professionnelle du protocole SNMP (Simple Network Management Protocol) pour Java |  |

Table 4.2 – Technologies et languages utilisées

**4.4 Présentation de la solution**

Dans cette section, nous présenterons certaines des interfaces utilisateur de notre application Web, tout en essayant de présenter les principales fonctionnalités de notre travail.

**4.4.1 Interface d’accueil**

L’interface d’accueil de notre application est conçue pour offrir une expérience utilisateur intuitive et agréable. Elle présente notre mission et explique qui nous sommes, donnant aux utilisateurs une compréhension claire de nos valeurs et de notre engagement. Les utilisateurs ont la possibilité de choisir leur rôle lors de l’inscription, en sélectionnant soit "Prestataire" pour ceux qui fournissent des services, soit "Demandeur" pour ceux qui recherchent des services. Cette interface simplifiée permet une navigation fluide et un accès rapide à nos services.

A website with a couple of people

AI-generated content may be incorrect.

Figure 4.2 – Interface d’accueil 1

**4.4.2 Interface registre**

L’interface de la plateforme est divisée en deux parties principales pour faciliter la navigation et l’utilisation. La première partie consiste à choisir le type d’utilisateur, tandis que la deuxième partie présente les formulaires adaptés en fonction du choix effectué.

**4.4.2.1 Choix du Type d’Utilisateur**

A group of people using a tablet

AI-generated content may be incorrect.

Figure 4.3 – Choix du Type d’Utilisateur

**4.4.2.2 Formulaire d’Inscription**

A screenshot of a login form

AI-generated content may be incorrect.

Figure 4.4 – Formulaire d’Inscription

**4.4.3 Interface login**

L’interface de connexion permet aux utilisateurs de se connecter facilement à leur compte existant.

A screenshot of a login form

AI-generated content may be incorrect.

Figure 4.5 – Interface login

**4.4.4 Dashboard admin**

Le dashboard admin est une interface centrale qui permet aux administrateurs de gérer efficacement les différentes fonctionnalités et utilisateurs du système. Il est divisé en plusieurs sections clés, chacune fournissant des outils et des informations spécifiques pour l’administration.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.6 – Dashboard admin

**4.4.5 Liste des risques**

Affiche la liste complète des risques enregistrés dans le système. Chaque ligne présente les informations clés du risque : le nom, la catégorie à laquelle il appartient, son niveau d’impact et de probabilité, son score calculé, son statut actuel (Identifié, En cours d’évaluation, …), ainsi que la date de création. Des boutons d’action (consultation, modification, suppression) permettent de consulter le détail d’un risque, de mettre à jour ses informations ou de le supprimer directement depuis cette interface.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.7 – Liste des risques

**4.4.6 Liste des utilisateurs de l’application**

Affiche la liste des comptes utilisateurs avec, pour chaque entrée, le nom d’utilisateur, l’email, le nom complet, le département, le rôle et le statut, ainsi que la date de création. Un bouton « + Nouvel utilisateur » en haut à droite permet d’accéder au formulaire d’ajout. Depuis cette page, on peut rapidement voir qui est administrateur, gestionnaire, etc., et activer ou désactiver des comptes via les icônes de modification/suppression disponibles dans la colonne « Actions ».A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.8 – Liste des utilisateurs

**4.4.7 Choix type annonce prestataire**

Cette page liste tous les contrôles enregistrés dans le système, en affichant pour chacun son nom, son type (préventif, détectif, correctif, compensatoire), son statut (planifié, en cours, implémenté), la fréquence d’exécution, l’efficacité mesurée (%), et la date de création. Elle permet également de créer un nouveau contrôle via le bouton « + Nouveau contrôle » et d’accéder à chaque contrôle pour le consulter ou le modifier grâce aux icônes « œil » et « crayon ».A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.9 – Choix type annonce prestataire

**4.4.8 Créer annonce prestataire baby-sitter**

Cette section permet aux prestataires de services spécialisés dans le baby-sitting de créer une annonce détaillée présentant leurs services, attirant ainsi des clients potentiels.

**4.4.8.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.10 – Step 1

**4.4.8.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.11

**4.4.9 Créer annonce prestataire auxiliaire de vie**

Cette section permet aux prestataires de services offrant des services d’auxiliaire de vie de créer une annonce détaillée.

**4.4.9.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.12 – Step 1

**4.4.9.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.13

**4.4.10 Créer annonce prestataire garde animaux**

Cette section permet aux prestataires de services offrant des services de garde d’animaux de créer une annonce détaillée présentant leurs offres.

**4.4.10.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.14 – Step 1

**4.4.10.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.15

**4.4.11 Liste annonces prestataire**

Cette section affiche une liste complète de les annonces du type service choisie créées par les prestataires de services sur la plateforme.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.16 – Liste annonces prestataire

**4.4.12 Détails annonce prestataire**

Cette section fournit des informations détaillées sur une annonce spécifique créée par un prestataire de services.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.17 – Détails annonce

**4.4.13 Chat**

Cette section facilite la communication entre les prestataires de services et les demandeurs via une interface de chat.

A screenshot of a chat

AI-generated content may be incorrect.

Figure 4.18 – Chat

**4.4.14 Profil prestataire**

Cette section affiche le profil d’un prestataire de services, y compris ses informations personnelles et professionnelles.

A screenshot of a web page

AI-generated content may be incorrect.

Figure 4.19 – Profil prestataire

**4.4.15 Choix type annonce demandeur**

Cette section permet aux demandeurs de services de choisir le type de service dont ils ont besoin, les guidant vers le formulaire approprié pour créer leur annonce.

A screenshot of a phone

AI-generated content may be incorrect.

Figure 4.20 – Choix type annonce demandeur

**4.4.16 Créer annonce demandeur baby-sitter**

Cette section permet aux demandeurs de services cherchant des services de baby-sitting de créer une annonce détaillée.

**4.4.16.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.21 – Step 1

**4.4.16.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.22

**4.4.17 Créer annonce demandeur auxiliaire de vie**

Cette section permet aux demandeurs de services cherchant des services d’auxiliaire de vie de créer une annonce détaillée.

**4.4.17.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.23 – Step 1

**4.4.17.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.24

**4.4.18 Créer annonce demandeur garde animaux**

Cette section permet aux demandeurs de services cherchant des services de garde d’animaux de créer une annonce détaillée présentant leurs besoins.

**4.4.18.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.25 – Step 1

**4.4.18.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.26

**4.4.19 Edit annonce prestataire**

Cette section permet à un prestataire de services de modifier une annonce existante qu’il a créée.

**4.4.19.1 Step 1**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.27 – Step 1

**4.4.19.2 Step 2**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.28 – Step 2

**4.4.20 Supprimer annonce Demandeur**

Cette section permet à un demandeur de services de supprimer une annonce existante qu’il a créée.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.29 – Supprimer annonce Demandeur

**4.5 Conclusion**

ans ce chapitre nous avons présenté les travaux réalisés suivant les principales fonctionnalités de notre solution. Nous avons présenté l’environnement de développement (Matériel et Logiciel). Ensuite, nous avons listé les différentes étapes de la réalisation de l’application en mettant l’accent sur les services réalisés.

***Conclusion Générale***

En conclusion, la plateforme "Maminou" représente une solution efficace et innovante pour la mise en relation des prestataires de services et des demandeurs de services dans le domaine de la garde d’enfants, de l’aide aux personnes âgées et de la garde d’animaux domestiques. Grâce à ses fonctionnalités conviviales et intuitives, elle offre une expérience utilisateur optimale, facilitant ainsi la recherche et la fourniture de services de qualité.

La conception et le développement de "Maminou" ont été guidés par une compréhension approfondie des besoins des utilisateurs et une attention particulière à l’ergonomie et à la facilité d’utilisation. Cette approche a permis de créer une plateforme qui répond efficacement aux attentes des utilisateurs, tout en offrant un environnement sécurisé et fiable pour la gestion des transactions et des communications.

Dans un contexte où la demande de services de garde et d’assistance est en constante évolution, "Maminou" se positionne comme un outil précieux pour les prestataires et les demandeurs, leur permettant de trouver des solutions adaptées à leurs besoins spécifiques. Avec un engagement continu envers l’innovation et l’amélioration, "Maminou" est appelé à jouer un rôle de premier plan dans le secteur des services à la personne, contribuant ainsi au bien-être et au confort des familles et des individus.

***Bibliographie***

1. Titre, https ://bambinosit.com, consulte le date
2. https ://yoopies.fr
3. https ://blog.hubspot.fr/marketing/cycle-en-v
4. https ://www.lucidchart.com/pages/fr/langage-uml
5. https ://fr.wikipedia.org/wiki/Mod
6. https ://fr.wikipedia.org/wiki/StarUML
7. https ://www.blogdumoderateur.com/tools/figma
8. https ://www.techtarget.com
9. https ://learn.microsoft.com
10. https ://fr.wikipedia.org/wiki/PostgreSQL
11. https ://paris-sorbonne.libguides.com
12. https ://developer.mozilla.org/fr/docs/Web/HTML
13. https ://fr.wikipedia.org/wiki/Feuilles-de-style-en-cascade
14. https ://fr.wikipedia.org/wiki/JavaScript
15. https ://fr.wikipedia.org/wiki/Bootstrap-(framework)
16. https ://fr.wikipedia.org/wiki/Laravel
17. https ://fr.wikipedia.org/wiki/Structured-Query-Language
18. https ://pusher.com/