
DEDICATION

À la mémoire de ce qui a m'a quitté trop tôt, dont le sourire chaleureux et la tendresse infinie ont marqué nos vies. Ton amour reste gravé dans nos cœurs et continue de nous guider chaque jour. Tes derniers mots sont toujours dans ma mémoire...

À mon père Hichem, le rocher sur lequel je m'appuie, dont les valeurs, la patience et la sagesse ont été mes plus grandes leçons. Ton dévouement inébranlable a été ma lumière dans l'obscurité.

À ma mère Samira, mon inspiration, dont l'amour inconditionnel et le soutien sans faille m'ont porté à travers les hauts et les bas. Tes sacrifices et ta présence ont façonné mon existence.

À mes frères, Naceur et Taha Yassine, mes complices de jeux, mes alliés de confidences. Nos liens familiaux sont des trésors que je chéris profondément, et chaque moment partagé est une source de joie.

À mes amis, complices de fous rires, épaules solides et sources infinies de soutien. Votre amitié a été un cadeau précieux et une source de réconfort dans toutes les circonstances.

À vous tous, qui avez fait partie de mon voyage jusqu'à ce point, je vous suis profondément reconnaissant. Votre amour, vos encouragements et vos sourires ont été les étoiles qui ont illuminé mon chemin.

En dédiant ce travail à ces personnes spéciales, je souhaite rendre hommage à vos contributions qui ont enrichi ma vie et m'ont aidé à grandir. Vos influences continuent de résonner en moi, formant une partie essentielle de mon parcours.

REMERCIEMENT

Je tiens à exprimer ma profonde gratitude envers mon encadrante académique, Mme Ferchichi Imen, pour son engagement inlassable et son soutien durant tout le processus de ce projet. Votre expertise, vos conseils éclairés et votre patience ont été d'une importance cruciale pour la réussite de ce travail. Votre présence a été un moteur de motivation et d'excellence.

Mes remerciements s'adressent également à l'ensemble de l'équipe pédagogique qui a joué un rôle essentiel dans mon parcours académique. Vos enseignements enrichissants et vos retours constructifs ont contribué à mon épanouissement personnel et professionnel.

Un remerciement spécial à mon encadrante professionnelle au sein de la société ELCO , Mlle Aouissaoui Rania, pour sa disponibilité, ses conseils pratiques et son investissement dans ce projet. Votre expertise sur le terrain a été une source précieuse d'apprentissage et de développement de compétences.

Je tiens à adresser ma reconnaissance au gérant de la société, M. Jouini Helmi, pour avoir rendu cette collaboration possible. Votre appui et vos ressources ont grandement facilité la réalisation de ce projet et ont contribué à son succès.

Mes remerciements s'étendent également à tous ceux qui, de près ou de loin, ont contribué à cette aventure. Votre apport a été essentiel et a enrichi cette expérience de manière significative.

Je vous adresse toute ma gratitude pour votre engagement, vos conseils et votre soutien. Votre contribution a été précieuse dans la réalisation de ce projet de fin d'études.

TABLE DES MATIÈRES

Table des figures	6
Liste des tableaux	8
1 cadre général	11
1.1 Présentation de l'organisme d'accueil	11
1.1.1 les secteurs d'activités	12
1.2 Analyse :	13
1.3 Critique de l'existant :	13
1.4 Solutions existantes	14
1.5 Solution Proposée :	15
1.6 Méthodologie adoptée (de développement)	15
1.6.1 Comparaison des méthodologies	15
1.6.2 Choix de la méthodologie	15
1.6.3 Présentation de la méthodologie Agile SCRUM	15
1.6.4 Le processus Scrum	17
1.6.5 Pilotage du projet avec Scrum :	17
1.7 Diagramme de Gantt (Planification) :	17
2 Étude de l'Art	19
2.1 Concepts Fondamentaux de la Sécurité de l'Information	19
2.1.1 Définitions et Termes Clés	19
2.1.2 Classification des Actifs d'Information	19
2.1.3 Les Trois Piliers de la Sécurité de l'Information (CID)	20
2.2 Normes de Sécurité Pertinentes : ISO 27001	20
2.2.1 Présentation et Objectifs de l'ISO 27001	20

2.2.2	Comparaison avec d'Autres Normes et Cadres	21
2.3	TISAX : Norme de Sécurité pour l'Industrie Automobile	21
2.3.1	Vue d'Ensemble de TISAX	21
2.3.2	Exigences et Objectifs de TISAX	21
2.4	Avantages et Défis de l'Adoption d'ISO 27001 et TISAX	21
2.4.1	Avantages de l'Adoption	22
2.4.2	Défis de l'Adoption	22
2.5	Métriques et Indicateurs de performance (KPI)	24
3	Analyse et spécification des besoins	27
3.1	Identification des acteurs	27
3.2	Spécification des besoins :	27
3.2.1	Les besoins fonctionnels	28
3.2.2	Les besoins non fonctionnels :	28
3.3	Les fonctionnalités du backlog	28
3.4	conception	29
3.4.1	Langage de Modélisation Unifié (UML)	30
3.4.2	Diagramme de Cas d'utilisation (use case)	30
3.4.3	Diagramme conception de Cas d'utilisation global	30
3.4.4	Diagramme de Cas d'utilisation relatif au cas d'utilisation« gérer requirement».	31
3.4.5	Description textuelle du cas d'utilisation « Ajouter un requirement»	31
3.4.6	Description textuelle du cas d'utilisation « modifier un requirement»	32
3.4.7	Description textuelle du cas d'utilisation « supprimer une requirement »	33
3.4.8	Diagramme de cas d'utilisation relatif au cas d'utilisation 2(gérer kpi)	34
3.4.9	Description textuelle (consulter kpi)	34
3.4.10	description textuelle ajouter détails kpi	35
3.5	Diagrammes de séquences	36
3.5.1	Diagramme de séquence relatif au cas d'utilisation 1(s'authentifier)	37
3.5.2	Diagramme de séquence relatif au cas d'utilisation 2 (supprimer requirement)	38
3.6	Diagramme d'activité	39
4	Aperçu conceptuel	41
4.1	Architecture physique	41
4.2	Patrons de conception	42
4.2.1	Patron de conception MVC :	42
4.2.2	Patron de conception DAO :	43
4.2.3	Patron de conception DTO :	43

4.2.4	Patron de conception IOC	44
4.3	Conception détaillée	45
4.3.1	Diagramme de classe	45
5	Réalisation	48
5.1	Diagramme de déploiement	48
5.2	Le style architectural	49
5.2.1	Partie front-end	49
5.2.2	Partie back-end	50
5.3	Environnement de développement	52
5.3.1	Technologie de programmation	52
5.3.2	Langage de programmation	53
5.3.3	Base de données	53
5.3.4	Outils	54
5.4	Développement des interfaces	56
5.4.1	L'interface authentification	56
5.4.2	L'interface Dashboard	57
5.4.3	L'interface requirements	60
	Webographie	63

TABLE DES FIGURES

1.1	Logo Elco Solutions	11
1.2	le domaine automobile	12
1.3	carte arduino	12
1.4	dashboard monitoring tools	13
1.5	logo CyberDay	14
1.6	logo Hyperlex	14
1.7	Le processus Scrum	16
1.8	Diagramme de Gantt	18
2.1	Process d'un SMSI	20
2.2	Logo iso 27001	21
2.3	Logo certification TISAX	21
2.4	Planification SMSI	22
2.5	Cycle de traitement des risques	23
2.6	Stratégie SMSI	24
3.1	Logo UML	30
3.2	le diagramme de cas d'utilisation global	31
3.3	le diagramme de cas d'utilisation « gérer requirements »	31
3.4	le diagramme de cas d'utilisation « gérer Kpi»	34
3.5	Diagramme de séquence relatif au cas d'utilisation s'authentifier	37
3.6	Diagramme de séquence relatif au cas d'utilisation supprimer requirement	38
3.7	diagramme d'activité pour l'authentification	39
3.8	diagramme d'activité pour Gérer KPI	40
4.1	architecture MVC	43

4.2	principe du patron de conception DAO	43
4.3	principe du patron de conception DTO	44
4.4	exemple dto de l'application	44
4.5	Diagramme de classe	46
5.1	Diagramme de déploiement	49
5.2	Architecture MVVM	50
5.3	L'architecture Microservices	51
5.4	L'architecture Microservices	52
5.5	Logo node.js	52
5.6	Logo React	52
5.7	Logo Typescript	53
5.8	Logo Nestjs	53
5.9	Logo Sql server	54
5.10	Logo Azur devops	54
5.11	Logo Visual studio code	54
5.12	Logo Azur devops	55
5.13	Logo Draw.io	55
5.14	Logo Overleaf	55
5.15	Logo Github	56
5.16	L'interface authentification	57
5.17	L'interface Dashboard	58
5.18	L'interface results requiremets	58
5.19	L'interface results Kpi sensibilisation	59
5.20	L'interface d'ajouter et de modifier de requirements	60
5.21	L'interface liste de requirements	61

LISTE DES TABLEAUX

1.1	Comparaison entre méthodes agiles et méthodes classiques	16
1.2	Équipe et rôle	17
3.1	Les fonctionnalités du backlog	29
3.2	Description textuelle « ajouter un requirement »	32
3.3	Description textuelle « Modifier un requirement »	33
3.4	Description textuelle « supprimer une requirement »	34
3.5	Description textuelle « consulter kpi »	35
3.6	Description textuelle « ajouter détails kpi »	35

INTRODUCTION GÉNÉRALE

La sécurité de l'information est devenue un enjeu critique pour les entreprises de toutes tailles, dans tous les secteurs d'activité, et la mise en place d'une plateforme de gestion de la sécurité peut aider à renforcer la sécurité des données, à prévenir les cyberattaques et à répondre rapidement aux incidents de sécurité

La norme ISO 27001 fournit un cadre de référence pour la gestion de la sécurité de l'information, en établissant des exigences pour la mise en place d'un système de management de la sécurité de l'information (SMSI) efficace.

En utilisant cette norme, notre plateforme de gestion de sécurité assure une approche centrée sur la sécurité pour répondre aux besoins de sécurité de l'entreprise.

Dans ce rapport, nous allons présenter les différentes étapes de la conception et de l'implémentation de notre plateforme de gestion de sécurité, en utilisant les exigences de la norme ISO 27001 comme base.

Nous allons examiner les différentes exigences de sécurité de l'entreprise et les intégrer dans notre plateforme de gestion de sécurité, en utilisant des technologies de sécurité robustes et en suivant les meilleures pratiques de sécurité recommandées par la norme ISO 27001.

Nous allons également présenter les fonctionnalités clés de notre plateforme de gestion de sécurité, y compris la gestion des identités et des accès, la détection et la réponse aux incidents de sécurité, la surveillance continue de la sécurité et la gestion des vulnérabilités.

Nous allons également présenter les résultats de nos tests et de nos évaluations de sécurité Pour démontrer l'efficacité de notre plateforme de gestion de sécurité basée sur la norme ISO 27001.

La mise en place d'une plateforme de gestion de sécurité basée sur la norme ISO 27001 est essentielle pour protéger les informations critiques de l'entreprise et se conformer aux exigences réglementaires en matière de sécurité de l'information.

Le présent rapport fournira des informations utiles pour les entreprises qui cherchent à mettre en place une plateforme de gestion de sécurité basée sur la norme ISO 27001 ou à améliorer leur sécurité de l'information existante.

Il s'articule autour de quatre chapitres :

- Le premier chapitre contient une présentation de l'entreprise d'accueil de mon stage, son système d'information et il présente une vue globale du service informatique de la société en matière de ressources existantes. Ainsi, il décrit l'étude de l'existant et la solution proposée qui répond aux besoins de l'entreprise.
- Le deuxième chapitre présente des notions de base et quelques termes primordiaux sur la gestion du système de management du système d'information ainsi qu'une étude de la norme de certification ISO 27001 et Tisax et ses exigences qui se termine par la justification du choix de la solution à adapter, ensuite, il présente une description détaillée de la solution choisie, son architecture, ses services et son mode de fonctionnement pour finir avec un choix technique de la solution à développer.
- Le troisième chapitre consacré à l'analyse et la spécification des besoins de l'application de gestion de système de management du système d'information.
- Le quatrième chapitre présente l'implémentation et la mise en place de la solution.

Finalement, ce rapport s'achève par une conclusion générale qui récapitule les résultats obtenus et expose les perspectives des travaux à venir.

CHAPITRE 1

CADRE GÉNÉRAL

Introduction

Ce premier chapitre est dédié à la présentation du cadre général du projet. Il est Subdivisé en deux sections. La première est une description de l'organisme d'accueil et du Service auquel nous étions affectées. La deuxième section contient une présentation du Projet, une étude, analyse et critique de l'existant

1.1 Présentation de l'organisme d'accueil

Elco-Solutions[1] est une entreprise 100% privée Créé en 2016 avec l'accent sur le développement de logiciels embarqués et les solutions de connectivité.

Elle aide les grandes entreprises internationales à accroître leur compétitivité en tirant parti de leur expertise en développement de logiciels, en IoT et en connectivité.

Elco-solutions est une entreprise certifiée Iso 9001, Iso 27001 et Tisax.



FIGURE 1.1 – Logo Elco Solutions

1.1.1 les secteurs d'activités

Automobile :

Elco propose une gamme étendue de services dans le secteur de l'automobile. Ils se spécialisent dans l'accompagnement des projets clients liés à divers domaines, Voitures connectées - Contrôle moteur - Processeur de véhicule - Recharge sans fil,

Travailler avec plusieurs équipementiers automobiles de niveau 1 pour développer des technologies de pointe. Elco couvre un large éventail de disciplines technologiques et accompagne ses clients tout au long du cycle en V.

Elco Solutions Software est déjà intégrée dans les voitures allemandes et françaises produites en série.



FIGURE 1.2 – le domaine automobile

Industrie :

L'équipe d'Elco Solutions met à disposition ses services pour une large gamme de besoins dans le domaine de l'automobile. Leurs domaines d'expertise incluent le développement de Linux embarqué, les réseaux et les bus de terrain, l'internet des objets, ainsi que le développement d'applications industrielles.



FIGURE 1.3 – carte arduino

Outil de surveillance et de supervision industriel :

ElcoMES 4.0 : Connectivité - Décision basée sur les données - Supervision et contrôle

Aider les organisations manufacturières à améliorer la productivité et à réduire le temps et les coûts de production en tirant parti des capacités de mesure des indicateurs de performance clés permettant la prise de décision basée sur les données et le contrôle statistique des processus.



FIGURE 1.4 – dashboard monitoring tools

1.2 Analyse :

La gestion de la sécurité du système d'information est un travail pénible, laborieux, d'une part vue la diversité des procédures ainsi que le volume important des données analyser au sein d'une entreprise, ce qui entraine la complexité de suivre toutes les activités liées à la sécurité d'information en établissant les exigences de la norme ISO 27001 et Tisax.

Dans ce contexte, Elco-solutions cherche une solution pour éliminer complètement la pratique Fastidieuse de mettre en place des processus manuellement.

1.3 Critique de l'existant :

Vu le grand nombre des procédures, à gérer, en établissant les exigences de la norme ISO 27001 et Tisax, Le comité de sécurité du système d'information d'Elco-Solutions est Incapable de suivre et de documenter toutes les activités liées à la sécurité de l'information, Telles que les audits et les évaluations des risques.), la communication entre les Différents Services et collaborateurs de l'entreprise peut être inefficace, ce qui peut entraîner des problèmes de coordination et de collaboration, l'entreprise est plus exposée aux risques liés à la sécurité de l'information, tels que les pertes de données et les failles de Sécurité.

1.4 Solutions existantes

L'étude de l'existant est une phase importante dans la phase d'analyse d'un projet. Dans ce qui suit, nous allons nous intéresser dans une première partie à la description des solutions existantes.

❖ *CyberDay*

Cyberday est une solution payante. Elle permet de diviser les cadres choisis (par exemple ISO 27001, NIST CSF, ISO 27701) en tâches de sécurité prioritaires et vous guide dans leur mise en œuvre directement sur Microsoft Teams.

Cyberday offre la possibilité de choisir les cadres de cybersécurité les plus importants comme cibles pour le travail. À partir de ceux-ci, une liste unique de tâches de sécurité est formée pour le système de gestion.



FIGURE 1.5 – logo CyberDay

❖ *Hyperlex*

En parlant des normes ISO, Hyperlex s'est équipée de moyens de protection et de protocoles à travers un smsi qui décerne la fameuse certification et son extension.

Elle a pu développer une intelligence artificielle qui fait aujourd'hui leur force. Cette IA va permettre de ne plus manquer d'échéances, d'éléments clés ou de clauses importantes et donc de prévenir les risques contractuels.



FIGURE 1.6 – logo Hyperlex

1.5 Solution Proposée :

Résoudre les problèmes et obstacles évoqués par Elco-solutions, satisfaire au maximum

Les utilisateurs en matière de qualité et de continuité d'activités, et alléger la complexité

De gérer le système d'information. Elco-Solutions propose ce projet :

Qui comprend la mise en place d'une solution autonome, complète de gestion centralisée du Système d'information.

Il s'agit donc et sans doute de :

Développer une Plateforme pour automatiser les processus ISMS (Information Security Management System) pour la conformité aux normes de sécurité de l'information telles que ISO27001 et TISAX.

1.6 Méthodologie adoptée (de développement)

Avant de réaliser un projet informatique, il est primordial de sélectionner la méthode de travail permettant le développement de l'application pour répondre aux besoins du client.

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d'une application. C'est le fait de modéliser un système avant sa réalisation pour bien comprendre son fonctionnement et assurer sa cohérence. Un modèle est ainsi un facteur de réduction des coûts et des délais. Il est donc indispensable pour assurer un bon niveau de qualité de produit.

1.6.1 Comparaison des méthodologies

Afin de bien mener notre choix, nous avons commencé par une étude comparative entre les méthodes Agiles et classiques. Nous avons choisi une méthode agile, car elle garantit un maximum de contrôle sur le produit final et une meilleure qualité de communication avec les utilisateurs. Elles intègrent également le concept de travail d'équipe. Dans les méthodes agiles, on peut se référer au framework "SCRUM", qui servira à bien gérer les différentes étapes de montage du projet.

1.6.2 Choix de la méthodologie

Après la comparaison effectuée ci-dessus, nous avons constaté que la méthode agile[2] serait la plus adéquate pour réaliser notre projet.

1.6.3 Présentation de la méthodologie Agile SCRUM

Scrum est une structure Agile qui facilite la collaboration au sein des équipes et les aide à réaliser des tâches à haute valeur ajoutée. Elle propose un schéma de valeurs, rôles et directives pour leur permettre de se concentrer sur chaque itération et de s'améliorer en continu.

La méthode Scrum fonctionne sur le principe des sprints : des cycles de travail de deux semaines à l'issue desquels un livrable est attendu. Outre le sprint, il existe deux autres événements Scrum : les réunions debout

TABLEAU 1.1 – Comparaison entre méthodes agiles et méthodes classiques

	Méthodes traditionnelles	Méthodes agiles
Complétude de la documentation technique, spécifications fonctionnelles et techniques	Cette démarche nécessite un cahier des charges complet (spécification fonctionnelle et technique détaillée)	Les méthodes agiles sont davantage axées sur la phase de développement et la documentation est souvent incomplète.
Temps passé	Implication du client dans la phase de conception et lors de la livraison de l'application	L'équipe de développement et le client sont en communication constante, discutant des projets et planifiant les sprints.
Projet innovant	Cette approche convient aux projets complexes et à grande échelle où il n'y a pas d'opportunités.	Cette approche est idéale pour les projets incertains ou innovants
Adéquation avec la structure organisationnelle de l'entreprise	La structure organisationnelle de cette approche est similaire à celle de la plupart des entreprises. Plus adapté aux grandes entreprises avec des projets bien définis	Les méthodes agiles peuvent parfois prêter à confusion et nécessitent des équipes familiarisées ou formées à la méthode. Cette approche est recommandée pour les startups.
Flexibilité	Les méthodes traditionnelles laissent peu de place aux changements inattendus et de dernière minute. (Recommandation : prévoyez un temps de maintenance pour toute modification ou mise à niveau)	La flexibilité est l'un des piliers du "Manifeste pour le développement agile de logiciels".

quotidiennes et les rétrospectives de sprint. Les réunions debout quotidiennes ont lieu tous les jours, comme leur nom le suggère. En 15 minutes, elles permettent à l'équipe Scrum d'interagir et de coordonner ses tâches pour la journée. Quant aux rétrospectives de sprint, elles sont organisées par le Scrum master à chaque fin de sprint. C'est l'occasion pour l'équipe de faire le point sur son travail et de mettre en place des changements pour les prochains sprints.

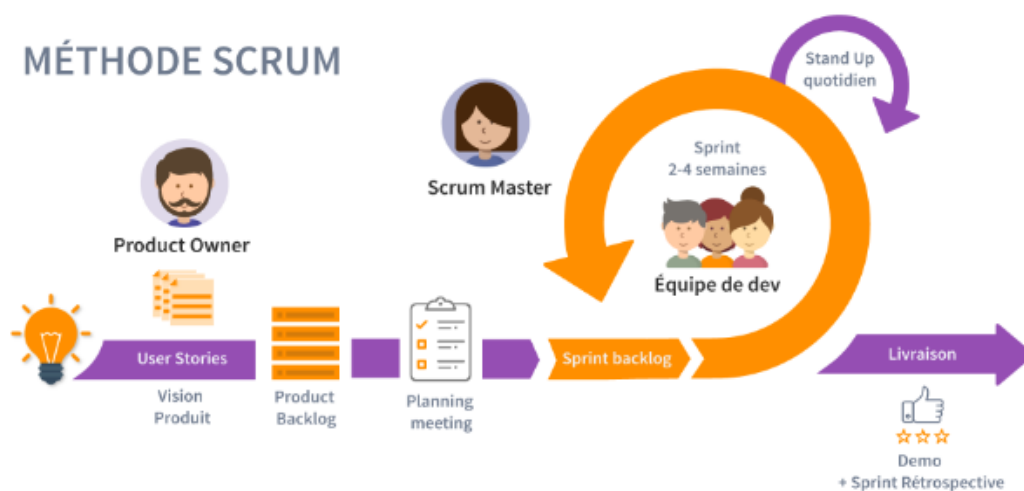


FIGURE 1.7 – Le processus Scrum

1.6.4 Le processus Scrum

Scrum Master : est un professionnel qui s'assure que les processus Scrum soient correctement appliqués (c'est-à-dire que Scrum soit compris et adopté) et veille à ce que l'équipe Scrum adhère à la théorie, aux pratiques et aux règles de Scrum.

Product Owner : est le professionnel responsable de maximiser la valeur du produit résultant du travail de l'équipe de développement ou, en d'autres termes, de maximiser la valeur pour le projet. Il est responsable de l'articulation des exigences du client et de la justification commerciale tout au long du projet. Nous pouvons dire que ce rôle incarne la voix du client.

L'équipe de développement : est constituée de professionnels qui livrent à chaque sprint un incrément « terminé » et potentiellement livrable du produit. Seuls les membres de l'Équipe de Développement créent l'incrément.

1.6.5 Pilotage du projet avec Scrum :

Les différents acteurs participants au déroulement des différentes phases de notre projet et leur associé

TABLEAU 1.2 – Équipe et rôle

Rôle SCRUM	Nom et prénom
Product Owner	Helmi jouini
Scrum Master	Rania aouissaoui
Team	Iheb chebil

1.7 Diagramme de Gantt (Planification) :

Dans cette partie, nous allons présenter la planification de notre travail durant la période de stage. La planification est cruciale pour un bon déroulement du projet, car elle permet d'identifier et à organiser les différentes tâches d'une manière séquentielle.

Elle fournit une meilleure estimation du temps nécessaire pour effectuer les différentes tâches du projet, peut également être estimé et définir la date approximative d'achèvement de chaque tâche. Pour cela, nous avons utilisé le diagramme de Gantt qui facilitera le suivi de l'avancement de notre projet. La figure suivante illustre notre diagramme de Gantt.

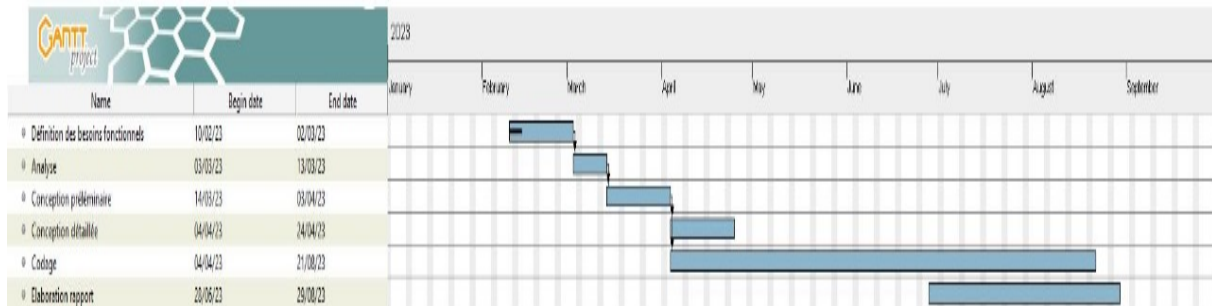


FIGURE 1.8 – Diagramme de Gantt

Conclusion

Ce chapitre a abordé une description de l'organisme d'accueil, le contexte général du projet et la méthodologie adoptée pour assurer la réalisation du travail dans les délais prévus. Le chapitre suivant approfondira davantage les fonctionnalités de notre projet en décrivant les différents acteurs ainsi que les différents cas d'utilisation.

CHAPITRE 2

ÉTUDE DE L'ART

Dans un monde interconnecté et numérisé, la sécurité de l'information est devenue une préoccupation cruciale pour les entreprises. La protection des données sensibles, la prévention des attaques cybernétiques et le maintien de la conformité réglementaire sont des enjeux essentiels. Ce chapitre vise à approfondir la compréhension de la sécurité de l'information en se concentrant sur deux normes clés : ISO 27001 et TISAX.

2.1 Concepts Fondamentaux de la Sécurité de l'Information

Dans cette partie, nous allons présenter les différents concepts fondamentaux de la sécurité de l'information

2.1.1 Définitions et Termes Clés

La sécurité de l'information repose sur des concepts fondamentaux tels que la confidentialité, l'intégrité et la disponibilité. La confidentialité garantit que les données sont accessibles uniquement aux parties autorisées, l'intégrité assure que les informations restent exactes et non altérées, tandis que la disponibilité assure que les données sont disponibles lorsque nécessaire.

SMIS[3] :

Un Système de Management de la Sécurité de l'Information (Information Security Management System en anglais) permet de gérer la sécurité de l'information.

Il désigne donc l'ensemble des politiques concernant la gestion de la sécurité des informations confidentielles. Un SMIS doit être efficace à long terme et donc doit s'adapter aux futurs changements internes et externes d'une entreprise.

2.1.2 Classification des Actifs d'Information

La classification des actifs d'information en fonction de leur valeur et de leur sensibilité est cruciale pour diriger les efforts de sécurité. Cela permet aux entreprises de concentrer leurs ressources sur la protection

des actifs les plus critiques et de réduire les risques.

2.1.3 Les Trois Piliers de la Sécurité de l'Information (CID)

La sécurité de l'information repose sur les trois piliers de la confidentialité, de l'intégrité et de la disponibilité. La confidentialité protège contre les accès non autorisés, l'intégrité assure que les données ne sont pas altérées, et la disponibilité garantit que les informations sont accessibles en temps voulu. Ces piliers fournissent un cadre solide pour élaborer des stratégies de sécurité.

2.2 Normes de Sécurité Pertinentes : ISO 27001

Différents éléments peuvent inciter une entreprise à sécuriser son système d'information autour de trois axes (sécurité de l'information, disponibilité de l'information et intégrité de l'information). Cette décision peut provenir d'une réglementation (Règlement Général sur la Protection des Données), d'un choix interne ou d'une exigence cliente (avec exigence d'audit ou de certification). Le macro-processus réalisé à partir des exigences de l'ISO 27001 permet d'établir, mettre en œuvre, évaluer et améliorer le SMSI d'un organisme pour obtenir la satisfaction des parties prenantes avec un SI sécurisé et performant, et ainsi le préparer à la certification.

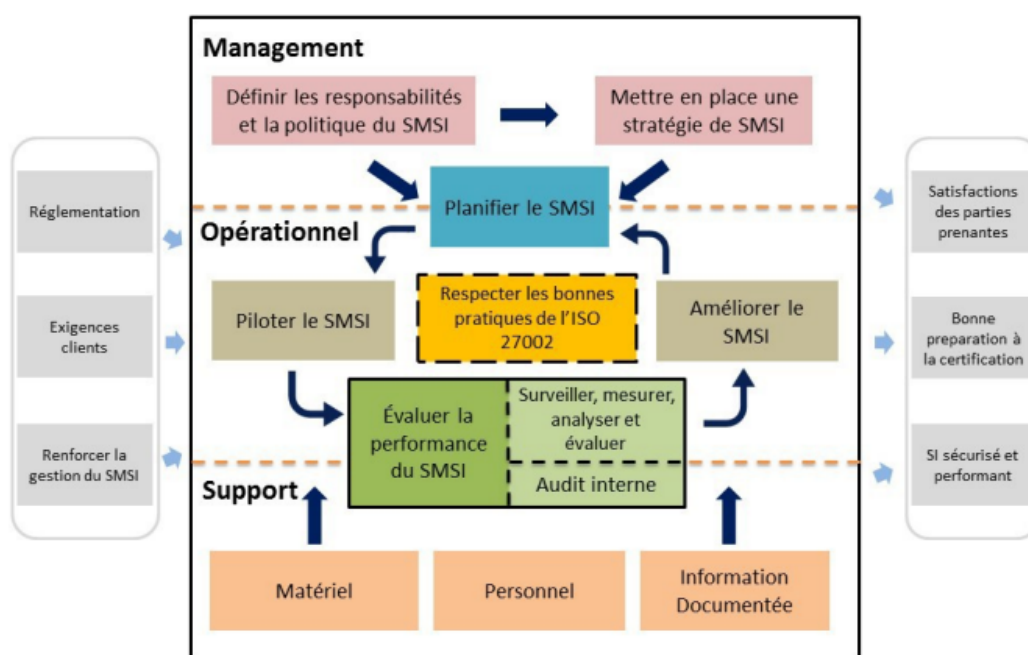


FIGURE 2.1 – Process d'un SMSI

2.2.1 Présentation et Objectifs de l'ISO 27001

L'ISO 27001[4], une norme internationale pour la gestion de la sécurité de l'information, fournit un cadre méthodique pour établir, mettre en œuvre, exploiter, surveiller, réviser, maintenir et améliorer un Système de Management de la Sécurité de l'Information (ISMS). Son objectif est de protéger les informations sensibles contre un large éventail de menaces.



FIGURE 2.2 – Logo iso 27001

2.2.2 Comparaison avec d'Autres Normes et Cadres

L'ISO 27001 se distingue d'autres normes telles que l'ISO 27002, NIST SP 800-53 et COBIT. Alors que la certification 27001 définit le processus global de gestion de la sécurité de l'information, 27002 offre des directives spécifiques pour les contrôles de sécurité. NIST SP 800-53 se concentre sur la sécurité des systèmes d'information, tandis que COBIT se penche sur la gouvernance de l'IT.

2.3 TISAX : Norme de Sécurité pour l'Industrie Automobile

Dans cette partie, nous allons présenter la norme TISAX ainsi que ses objectifs



FIGURE 2.3 – Logo certification TISAX

2.3.1 Vue d'Ensemble de TISAX

Trusted Information Security Assessment Exchange (TISAX) est un cadre de sécurité développé spécifiquement pour l'industrie automobile. Basé sur l'ISO 27001, TISAX met l'accent sur les échanges sécurisés de données et la protection des informations dans toute la chaîne d'approvisionnement.

2.3.2 Exigences et Objectifs de TISAX

TISAX[5] exige que les fournisseurs de l'industrie automobile prouvent leur conformité aux normes de sécurité. Les entreprises doivent passer par un processus d'audit et de validation pour démontrer leur engagement envers la protection des données et la sécurité de l'information.

2.4 Avantages et Défis de l'Adoption d'ISO 27001 et TISAX

Dans cette partie, nous allons présenter les avantages et l'importance d'utilisation des deux normes susmentionnées dans notre domaine d'activité.

2.4.1 Avantages de l'Adoption

L'adoption d'ISO 27001 et de TISAX apporte des avantages considérables, tels qu'une meilleure gestion des risques, une amélioration de la confiance des clients et des partenaires commerciaux, ainsi que la conformité aux réglementations en constante évolution.

2.4.2 Défis de l'Adoption

Cependant, l'adoption de ces normes n'est pas sans défis. La mise en place d'un ISMS nécessite un engagement organisationnel fort et peut exiger des ressources significatives en termes de temps et de budget. Les exigences strictes de conformité peuvent également représenter un défi pour certaines entreprises.

Planification du SMSI : Les risques liés aux enjeux de l'organisme sont identifiés et les mesures nécessaires pour y remédier (l'annexe A) mises en application. Des objectifs pertinents et cohérents avec la politique de sécurité de l'information de l'organisme sont fixés et sont documentés, communiqués à l'ensemble des acteurs et mesurables afin de pouvoir évaluer la performance dans l'atteinte de ces objectifs.

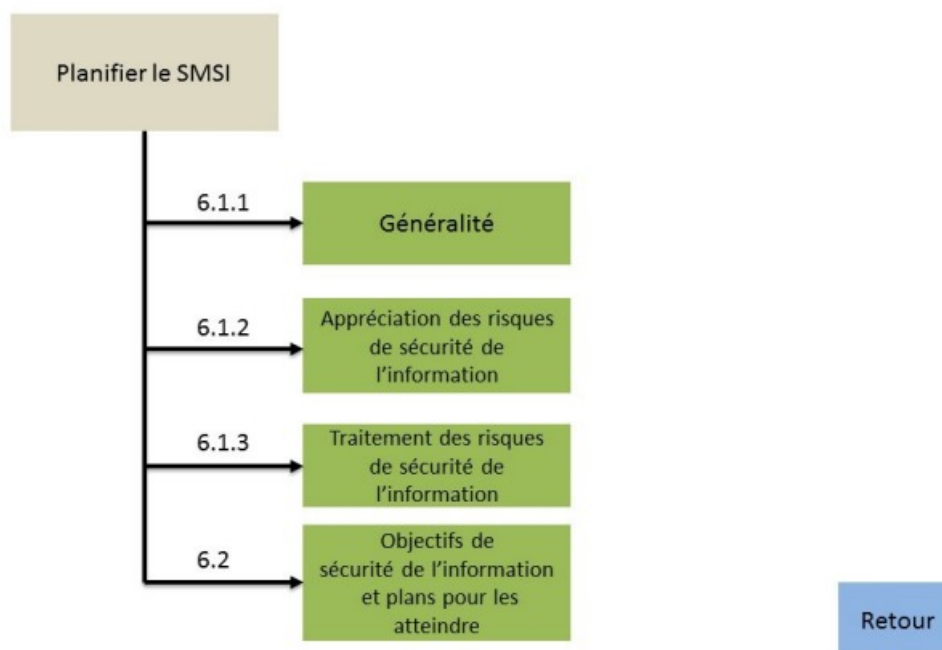


FIGURE 2.4 – Planification SMSI

Actions :

Responsable : Assurer que les objectifs sont réalisables :

S'attaquer aux effets indésirables

Déterminer : les ressources nécessaires telles que :

- les responsabilités
- les échéances

- Le moyen de les évaluer

Traitement des risques de sécurité de l'information

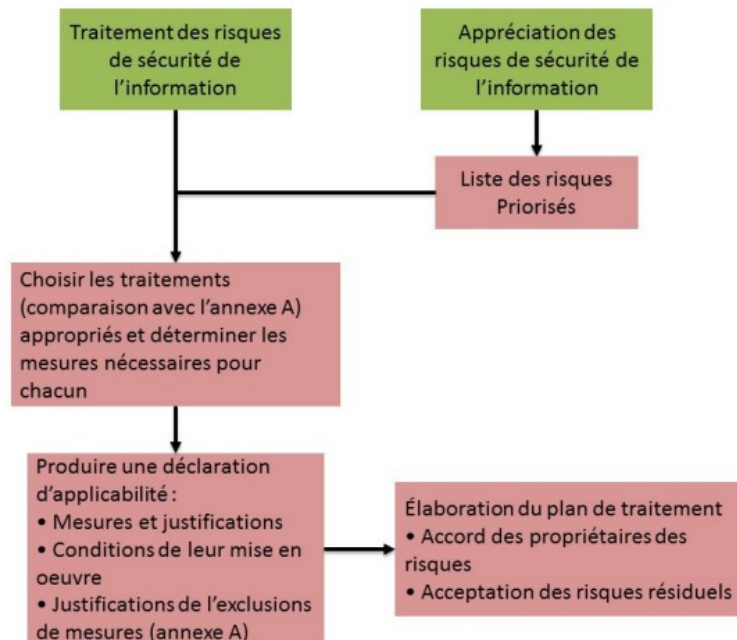


FIGURE 2.5 – Cycle de traitement des risques

Mettre en place une stratégie de SMSI

L'organisme détermine les enjeux externes et internes liés au contexte socio-économique dans lequel il se situe. De plus, les parties intéressées sont identifiées ainsi que leurs attentes et exigences. Ces dernières sont listées et revues périodiquement. Les champs d'application du système de management de la qualité (SMSI) sont fixés, ainsi que l'ensemble des processus nécessaires à la mise en œuvre de ce système.

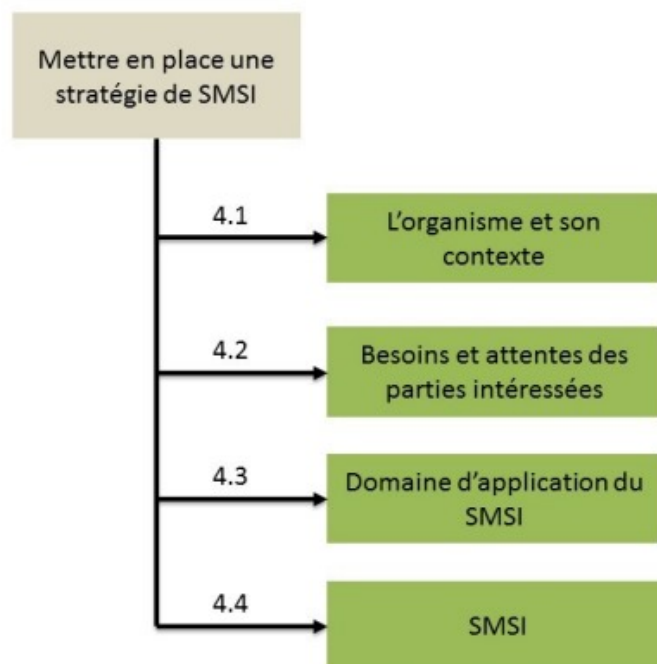


FIGURE 2.6 – Stratégie SMSI

Responsable : Directeur des systèmes d'information

- Établir le domaine d'application du SMSI[4] ainsi que les limites de son applicabilité en prenant en compte les interfaces et les dépendances existant entre les activités réalisées par l'organisation et celles réalisées par d'autres organisations.
- Établir, mettre en œuvre, tenir à jour et améliorer en continu le SMSI

2.5 Métriques et Indicateurs de performance (KPI)

Dans cette section, nous allons aborder en détail quelques indicateurs clés de performance (KPI) que nous avons utilisées pour évaluer de manière quantitative les résultats et l'efficacité de notre projet. Les KPI sont des mesures objectives qui nous permettent de suivre et de quantifier les progrès réalisés par rapport à nos objectifs préalablement fixés.

❖ Outils :

Pour calculer les différents KPI, la société prépare des fichiers VDA qui comprennent toutes les métriques à suivre et les formules de calcul tel que le fichier suivant :

VDA ISA : il fournit la base à une auto-évaluation pour déterminer l'état de la sécurité de l'information dans l'entreprise, les audits effectués par les services internes (par exemple, audit interne, sécurité de l'information) et l'examen conformément à TISAX (Trusted Information Security Échange d'évaluation, <http://enx.com/tisax/>).

❖ **Process :**

Pour mener à bien notre élaboration de KPI, nous devons suivre les étapes suivantes qui constituent le process de la métrique utilisée :

- **Collecte des Données :** Le fichier VDA contient des informations sur les vulnérabilités détectées dans notre système. Ce fichier peut être extrait à partir d'outils de gestion de vulnérabilités ou de rapports de scan de sécurité.
- **Analyse et Classification :** nous avons examiné le fichier VDA pour classer les vulnérabilités en fonction de leur gravité, de leur impact potentiel et du risque associé. Certains outils de gestion de vulnérabilités fournissent déjà des classifications de vulnérabilités.
- **Sélection des KPI :** Sur la base des vulnérabilités identifiées et classées, nous avons choisi les indicateurs clés de performance (KPI) pertinents à calculer.

❖ **Exemples d'indicateurs de performance KPI**

→ **Indicateur KPI 1 :** Taux d'Identification des Vulnérabilités

Définition : Le taux d'identification des vulnérabilités mesure le pourcentage de vulnérabilités détectées par rapport au nombre total de vulnérabilités potentielles dans le système.

Méthode de calcul : $(\text{Nombre de vulnérabilités détectées} / \text{Nombre total de vulnérabilités potentielles}) * 100$

Interprétation : Un taux plus élevé indique une meilleure capacité à détecter et à identifier les vulnérabilités potentielles dans le système.

Résultats obtenus : Le taux d'identification des vulnérabilités est de [X]%, ce qui témoigne de notre efficacité dans la détection proactive des vulnérabilités et notre engagement envers la sécurité de l'information.

Synthèse des résultats : Le taux d'identification des vulnérabilités est un indicateur essentiel pour évaluer notre capacité à repérer les failles potentielles dans le système. Plus ce taux est élevé, plus notre vigilance en matière de sécurité est renforcée. En continuant à surveiller et à améliorer ce taux, nous démontrons notre engagement envers la protection de nos actifs informatiques et la réduction des risques de sécurité.

En conclusion, l'évaluation du taux d'identification des vulnérabilités nous permet de quantifier notre efficacité dans la détection proactive des vulnérabilités. Cela contribue à une meilleure gestion des risques de sécurité et à la mise en œuvre de mesures préventives pour protéger nos actifs et données critiques.

→ **Indicateur KPI 2 :** Taux de Formation et de Sensibilisation

Définition : Le taux de formation et de sensibilisation mesure le pourcentage d'employés ayant suivi des sessions de formation en sécurité de l'information par rapport au nombre total d'employés de l'organisation.

Méthode de calcul : (Nombre d'employés ayant suivi une formation en sécurité / Nombre total d'employés) * 100

Interprétation : Un taux élevé indique une meilleure adhésion à la formation en sécurité et une sensibilisation accrue à la protection de l'information.

Résultats obtenus : Le taux de formation et de sensibilisation est de [X]%, ce qui montre notre engagement envers l'éducation en sécurité de l'information de nos employés.

Synthèse des résultats : Le taux de formation et de sensibilisation est un indicateur crucial pour évaluer notre succès dans la sensibilisation de notre personnel à la sécurité de l'information. Un taux élevé reflète un environnement de travail où la sécurité est prise au sérieux et où les employés sont conscients des meilleures pratiques en matière de sécurité.

En somme, l'évaluation du taux de formation et de sensibilisation démontre notre engagement envers la protection de l'information et l'éducation de notre personnel. Ce KPI contribue à renforcer la culture de sécurité au sein de l'organisation, réduisant ainsi les risques liés à la méconnaissance des menaces et des bonnes pratiques en sécurité.

Les KPI sont essentiels pour évaluer la réussite du projet de manière mesurable et concrète. Ils fournissent une base solide pour les décisions futures, les ajustements stratégiques et les leçons apprises pour les projets à venir. En somme, les indicateurs clés de performance sont un outil puissant pour évaluer l'impact et la valeur ajoutée de notre projet.

Conclusion

L'ISO 27001 et TISAX représentent des normes cruciales dans le domaine de la sécurité de l'information. Alors que l'ISO 27001 offre un cadre global pour la gestion de la sécurité de l'information, TISAX répond aux besoins spécifiques de l'industrie automobile. L'adoption de ces normes comporte des avantages substantiels, mais également des défis à surmonter. En ayant une compréhension approfondie de ces normes, notre projet de mise en place d'un ISMS sera mieux guidé, offrant ainsi une protection renforcée pour nos données et notre entreprise.

CHAPITRE 3

ANALYSE ET SPÉCIFICATION DES BESOINS

Introduction

Dans ce chapitre, nous listons les besoins fonctionnels et non fonctionnels de notre système et nous détaillons le travail par la méthodologie choisie dans le chapitre précédent. Par la suite, un bref aperçu sur le matériel de base, les technologies et les langages de programmation utilisés sont donnés pour la mise en place de l'environnement de travail.

3.1 Identification des acteurs

Un acteur représente un rôle joué par une entité externe qui interagit directement par le système étudié, il peut être un être humain, un matériel, un autre système... L'acteur est représenté par son rôle qui décrit les besoins et les capacités de ce dernier. L'activité du système a pour objectif de satisfaire les besoins des acteurs. Les acteurs de notre application sont les suivants :

ISMS Manager : c'est un acteur authentifié à son compte créé une fois validé. En se connectant, il a le droit de diriger la gestion et l'exploitation du système de gestion de la sécurité de l'information (ISMS).

3.2 Spécification des besoins :

L'analyse de notre application a permis d'identifier les fonctionnalités nécessaires pour différents consommateurs. Ainsi, dans cette section, les exigences fonctionnelles et non fonctionnelles sont décrites tout en identifiant les acteurs en dialogue avec le système.

Les exigences fonctionnelles et non fonctionnelles sont définies. En effet, les exigences fonctionnelles sont caractéristiques des taux d'entrée-sortie du système, tandis que les exigences non fonctionnelles sont des exigences de performance.

3.2.1 Les besoins fonctionnels

Les besoins fonctionnels représentent les attentes de chaque acteur de l'application à développer. Dans le cadre de notre projet, nous avons identifié les besoins fonctionnels suivants :

- **S'authentifier** : l'utilisateur peut accéder à l'application en saisissant ses identifiants.
- **Gérer requirements** : consiste à ajouter, modifier ou supprimer une requirement
- **Gérer les KPIs** : l'utilisateur peut consulter la liste des KPI, ajouter détail KPI, modifier détail KPI et supprimer détail KPI
- **Gérer tasks** consiste à ajouter, modifier ou supprimer une tâche.

3.2.2 Les besoins non fonctionnels :

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour assurer la qualité et le bon fonctionnement de cette solution adéquate aux attentes des utilisateurs. Notre solution permet de répondre aux exigences sélectionnées suivantes :

Évolutivité : L'application doit être capable de gérer une augmentation du nombre d'utilisateurs et de s'adapter à des nouvelles fonctionnalités et des exigences à mesure que l'entreprise évolue.

Performance : L'application doit garantir la sécurité des données des utilisateurs en mettant en place des mesures de protection telles que l'authentification sécurisée, le chiffrement des données et la conformité aux normes de sécurité

Disponibilité : L'application doit être disponible en tout temps, avec une disponibilité élevée et une résilience en cas de pannes ou de problèmes techniques.

Audit : L'audit joue un rôle crucial dans le développement et la maintenance d'une application, il permet de garantir la sécurité des données en identifiant la vulnérabilité, les erreurs de configurations et les failles de sécurité potentielles dans l'application.

3.3 Les fonctionnalités du backlog

L'élaboration du Product Backlog est une tâche effectuée par le « Product Owner » suite à des réunions avec le client. élaborer l'ensemble des caractéristiques fonctionnelles qui constituent le produit souhaité dans un ordre de priorité. Les caractéristiques fonctionnelles sont appelées des histoires utilisateurs (user story). Chaque histoire utilisateur est caractérisée par :

- **Un identifiant** : il détermine un identifiant unique pour l'histoire en question.
- **Une description** : elle décrit le besoin de l'acteur
- **Une priorité** : degré de priorité.

Pour prioriser nos user stories, nous avons pris en compte les critères suivants :

- La valeur apportée (Business Value).

- La fréquence d'utilisation
- La réduction des risques
- L'incertitude sur des besoins des utilisateurs qu'un user story permettra de diminuer
- La contribution à la qualité. Les travaux visant à garantir la qualité du produit devraient être prioritaires
- Les dépendances entre stories

Dans le tableau suivant, nous présentons la liste des user stories de notre projet ainsi que leur ordre de priorité.

TABLEAU 3.1 – Les fonctionnalités du backlog

US-ID	User story	Priorité
1	En tant qu'utilisateur, je dois pouvoir m'inscrire.	Forte
2	En tant qu'utilisateur, je dois pouvoir me connecter à mon compte en saisissant mes données personnelles.	Forte
3	En tant qu'utilisateur, je dois accéder à une page dédiée aux KPI à partir du tableau de bord.	Forte
4	En tant qu'utilisateur, je veux consulter la liste des KPI afin de pouvoir sélectionner une.	Moyenne
5	En tant qu'utilisateur, je veux consulter un KPI (Indicateurs Clés de Performance) pour obtenir une vision précise et informative de la performance du système.	Forte
6	En tant qu'utilisateur, je veux consulter un KPI et voir le graphe correspondant pour voir les variations au fil du temps.	Forte
7	En tant qu'utilisateur, je veux pouvoir rechercher un KPI suivant mon besoin	Moyenne
8	En tant qu'utilisateur, je veux pouvoir mettre à jour un KPI afin de refléter les données les plus récentes.	Forte
9	En tant qu'utilisateur, je veux pouvoir consulter la liste des tâches.	Moyenne
10	En tant qu'utilisateur, je veux pouvoir ajouter, modifier et supprimer une tâche.	Forte
11	En tant qu'utilisateur, je veux pouvoir consulter la liste des requirements.	Moyenne
12	En tant qu'utilisateur, je veux pouvoir ajouter, modifier et supprimer un requirement.	Forte

3.4 conception

La conception est la phase créative d'un projet d'ingénierie. Le but premier de la conception est de permettre de créer un système ou un processus répondant à un besoin en tenant compte des contraintes. Le système doit être suffisamment défini pour pouvoir être installé, fabriqué, construit et être fonctionnel, et pour répondre aux besoins du client. Avant de pouvoir passer à la phase de l'implémentation, on va consacrer ce chapitre à l'étude conceptuelle de notre application en appliquant la méthodologie UML.

3.4.1 Langage de Modélisation Unifié (UML)

UML[6] s'impose comme un standard de modélisation sur le marché. De plus, elle permet de vulgariser les aspects liés à la conception et à l'architecture propres au logiciel et aux utilisateurs. En effet, cette méthode apporte une compréhension rapide du programme à d'autres développeurs externes en cas de reprise du logiciel et facilite sa maintenance.



FIGURE 3.1 – Logo UML

3.4.2 Diagramme de Cas d'utilisation (use case)

Un cas d'utilisation (Use Case) est un diagramme qui modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Il permet de définir les acteurs externes du système sous forme des petits hommes et les activités auxquelles ils se livrent en utilisant le système.

3.4.3 Diagramme conception de Cas d'utilisation global

Le diagramme de cas d'utilisation global nous offre une vue d'ensemble des différentes fonctionnalités. On distingue sur ce diagramme les cas d'utilisation de notre application, et les acteurs qu'on a identifiés préalablement.

La figure 3.2 représente le diagramme de cas d'utilisation global de notre application.

Les acteurs impliqués sont : utilisateur (ISMS MANAGER). Ce diagramme englobe les cas d'utilisations présentés dans la section « spécification des besoins ».

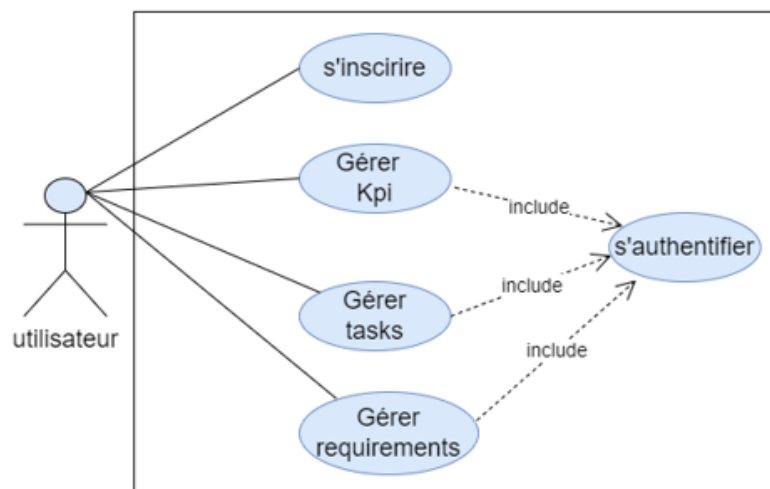


FIGURE 3.2 – le diagramme de cas d'utilisation global

Pour mieux comprendre les différents cas d'utilisation exprimés dans ce diagramme, nous allons en détailler les plus importants, dans ce qui suit. Dans cette partie, nous présentons quelques diagrammes des cas d'utilisation détaillés ainsi que leurs descriptions textuelles pour éviter la répétition.

3.4.4 Diagramme de Cas d'utilisation relatif au cas d'utilisation « gérer requirement».

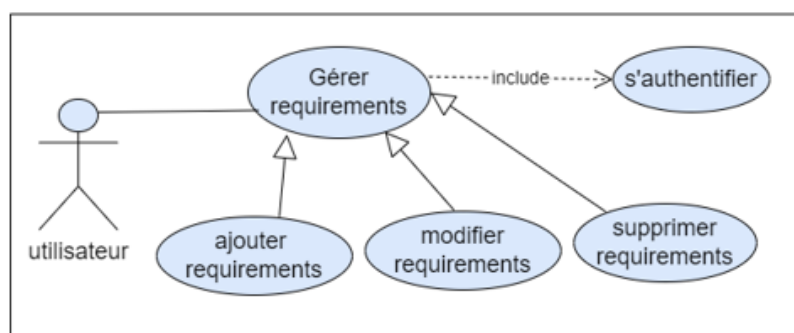


FIGURE 3.3 – le diagramme de cas d'utilisation « gérer requirements »

3.4.5 Description textuelle du cas d'utilisation « Ajouter un requirement»

La description textuelle de ces cas d'utilisation sera présentée sous forme de tableaux qu'identifient les scénarios de chaque cas d'utilisation en question.

Nous débutons donc par Le tableau suivant présente la description textuelle du cas d'utilisation « ajouter un requirement »

TABLEAU 3.2 – Description textuelle « ajouter un requirement »

Cas d'utilisation	Ajouter un requirement
Acteur	L'utilisateur
Objectif	À travers ce cas, l'utilisateur peut ajouter une nouvelle requirement.
Précondition	L'utilisateur soit authentifié.
Post-condition	Une nouvelle requirement est ajoutée.
Scénario nominal	1. L'utilisateur demande d'ajouter une requirement 2. Le système affiche le formulaire d'ajout, 3. L'utilisateur remplit le formulaire d'ajout et valide ou annule la requirement, 4. Le système effectue la sauvegarde.
Scénario alternatif	A1 : Un des champs est vide, l'ajout ne s'effectue pas. l'enchaînement démarre au point 3. A2 : Un des champs est invalide, l'ajout ne s'effectue pas. Cet enchaînement démarre au point 3.

3.4.6 Description textuelle du cas d'utilisation « modifier un requirement»

La description textuelle du cas d'utilisation "Modifier un requirement" détaille les différentes étapes et interactions impliquées lorsqu'un utilisateur souhaite modifier un requirement dans un système. Voici un exemple de description textuelle pour ce cas d'utilisation :

TABLEAU 3.3 – Description textuelle « Modifier un requirement »

Cas d'utilisation	modifier un requirement
Acteur	l'utilisateur
Objectif	requirement modifié
Précondition	L'utilisateur doit être authentifiée et avoir les droits nécessaires pour modifier un requirement
Post-condition	La requirement est mise à jour avec les nouvelles informations fournies par l'utilisateur.
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur accède à la liste des requirements existantes dans le système.2. L'utilisateur identifie la requirement qu'il souhaite modifier et sélectionne l'option de modification correspondante.3. Le système affiche les détails actuels de la requirement, y compris toutes les informations saisies précédemment.4. L'utilisateur apporte les modifications nécessaires aux informations de la requirement, telles que la description, titre, etc.5. L'utilisateur confirme qu'il a terminé les modifications et soumet la demande mise à jour.
Scénario alternatif	<p>Étape 4a - Informations incorrectes :</p> <ol style="list-style-type: none">1. Si l'utilisateur se rend compte que les informations actuelles de la requirement sont incorrectes, il peut annuler la modification ou revenir en arrière pour rétablir les données d'origine. <p>Étape 5a - Annulation :</p> <ol style="list-style-type: none">1. Si l'utilisateur décide de ne pas appliquer les modifications, il peut annuler le processus de modification sans enregistrer les changements. <p>Étape 5b - Validation des modifications :</p> <ol style="list-style-type: none">1. Le système vérifie les modifications apportées pour s'assurer qu'elles sont conformes aux règles et contraintes du système.2. Si les modifications ne sont pas valides, le système affiche un message d'erreur expliquant les problèmes spécifiques rencontrés et invite l'utilisateur à les corriger.

3.4.7 Description textuelle du cas d'utilisation « supprimer une requirement »

la description textuelle du cas d'utilisation "supprimer une requirement " détaille les différentes étapes et interactions impliquées lorsqu'un utilisateur souhaite modifier une requirement dans un système. Voici un exemple de description textuelle pour ce cas d'utilisation :

TABLEAU 3.4 – Description textuelle « supprimer une requirement »

Cas d'utilisation	supprimer un requirement
Acteur	L'utilisateur
Objectif	requirement supprimé
Précondition	L'utilisateur doit être authentifiée et avoir les droits nécessaires pour supprimer une requirement.
Post-condition	La requirement est supprimée du système.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la liste des requirements existantes dans le système. 2. L'utilisateur identifie la requirement qu'il souhaite supprimer et sélectionne l'option de suppression correspondante. 3. Le système affiche une confirmation de suppression pour demander la confirmation de l'utilisateur. 4. L'utilisateur confirme la suppression en cliquant sur le bouton de confirmation.
Scénario alternatif	<p>Étape 3a - Annulation de la suppression :</p> <ol style="list-style-type: none"> 1. Si l'utilisateur décide de ne pas supprimer la requirement, il peut annuler l'opération de suppression en cliquant sur le bouton d'annulation. <p>Étape 4c - Validation de la suppression :</p> <ol style="list-style-type: none"> 1. Avant de supprimer définitivement la requirement, le système peut effectuer des validations pour s'assurer que la demande peut être supprimée sans causer de problèmes, comme des dépendances avec d'autres données.

3.4.8 Diagramme de cas d'utilisation relatif au cas d'utilisation 2(gérer kpi)

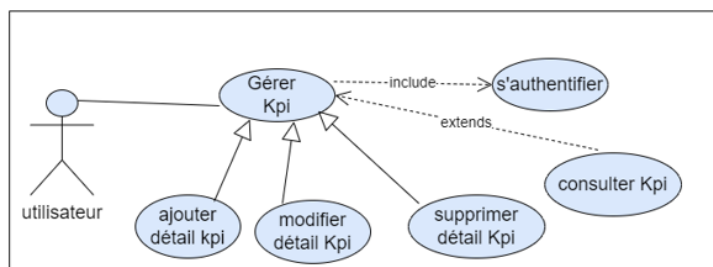


FIGURE 3.4 – le diagramme de cas d'utilisation « gérer Kpi»

3.4.9 Description textuelle (consulter kpi)

TABLEAU 3.5 – Description textuelle « consulter kpi »

Cas d'utilisation	Consulter Kpi
Acteur	L'utilisateur
Objectif	L'utilisateur souhaite consulter les indicateurs clés de performance (KPI)
Précondition	L'utilisateur soit authentifié.
Post-condition	L'utilisateur a pu consulter les Kpi
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de gestion des KPI. 2. Le système affiche une liste des catégories de KPI disponibles. 3. L'utilisateur sélectionne une catégorie de KPI parmi les options proposées. 4. Le système affiche la liste des KPI disponibles dans la catégorie sélectionnée. 5. L'utilisateur choisit un KPI spécifique qu'il souhaite consulter en cliquant dessus. 6. Le système affiche les détails du KPI sélectionné, y compris les graphiques, les valeurs numériques ou tout autre type de représentation visuelle. 7. L'utilisateur peut analyser les données affichées pour comprendre la performance de la catégorie correspondante. 8. L'utilisateur a la possibilité de revenir à la liste des KPI ou de quitter la consultation. 9. Si l'utilisateur souhaite continuer la consultation, il peut choisir un autre KPI à partir de la liste. 10. L'utilisateur peut également choisir de quitter la consultation des KPI à tout moment.

3.4.10 description textuelle ajouter détails kpi

TABLEAU 3.6 – Description textuelle « ajouter détails kpi »

Cas d'utilisation	ajouter détails kpi
Acteur	L'utilisateur
Objectif	L'utilisateur qui souhaite créer les détails d'un nouveau KPI.
Précondition	L'utilisateur est connecté au système.
Post-condition	Les détails spécifiques du KPI sont enregistrés et associés à l'objet KPI en cours de création.
Scénario nominal	<p>L'utilisateur accède au formulaire de création des détails du KPI.</p> <p>Le système affiche le formulaire avec des champs à remplir pour les détails spécifiques du KPI.</p> <p>L'utilisateur saisit les détails spécifiques du KPI :</p> <p>Objectifs cibles : Les niveaux de performance souhaités, y compris les valeurs à atteindre et les seuils à respecter.</p> <p>Sources de données : Les sources à partir desquelles les données nécessaires pour calculer le KPI seront extraites.</p> <p>Méthode de calcul : La formule ou l'algorithme utilisé pour calculer le KPI à partir des données sources.</p> <p>L'utilisateur confirme les détails spécifiques du KPI.</p> <p>Le système intègre les détails spécifiques du KPI à l'objet KPI en cours de création dans le modèle.</p>

3.5 Diagrammes de séquences

Le diagramme de séquence[7] permet de montrer les interactions d'objets dans le cadre d'un scénario, d'un diagramme, des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme et les acteurs secondaires éventuels à droite du système. Le but est de d'écrire comment se déroulent les interactions entre les acteurs ou objets

3.5.1 Diagramme de séquence relatif au cas d'utilisation 1(s'authentifier)

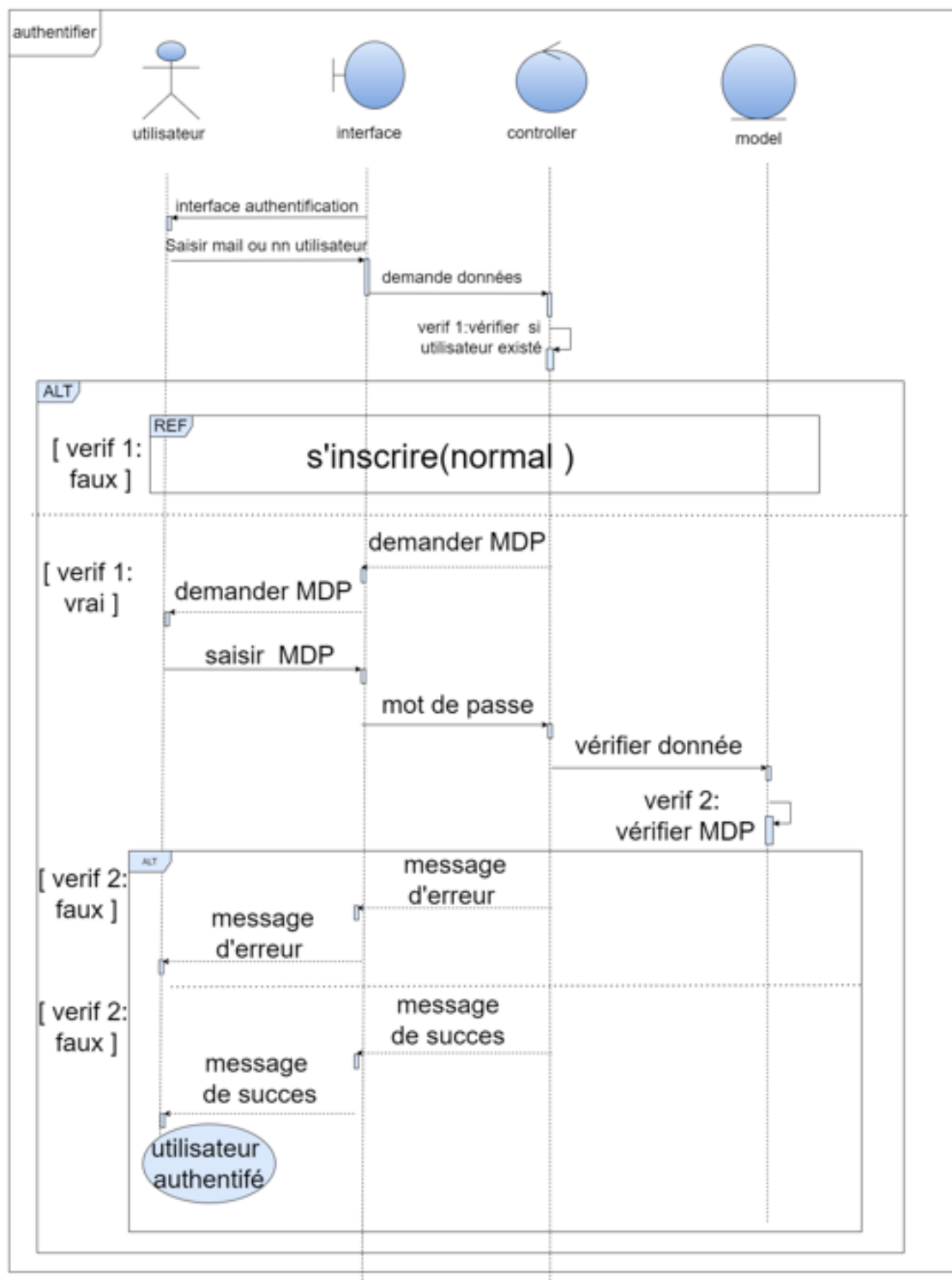


FIGURE 3.5 – Diagramme de séquence relatif au cas d'utilisation s'authentifier

La figure 3.5 représente le diagramme de séquence pour le cas d'utilisation Authentification Lorsque l'utilisateur fournit ses informations d'identification (par exemple, nom d'utilisateur et mot de passe). Si les paramètres sont valides, l'utilisateur aura ensuite l'accès aux différentes fonctionnalités de la plateforme.

3.5.2 Diagramme de séquence relatif au cas d'utilisation 2 (supprimer requirement)

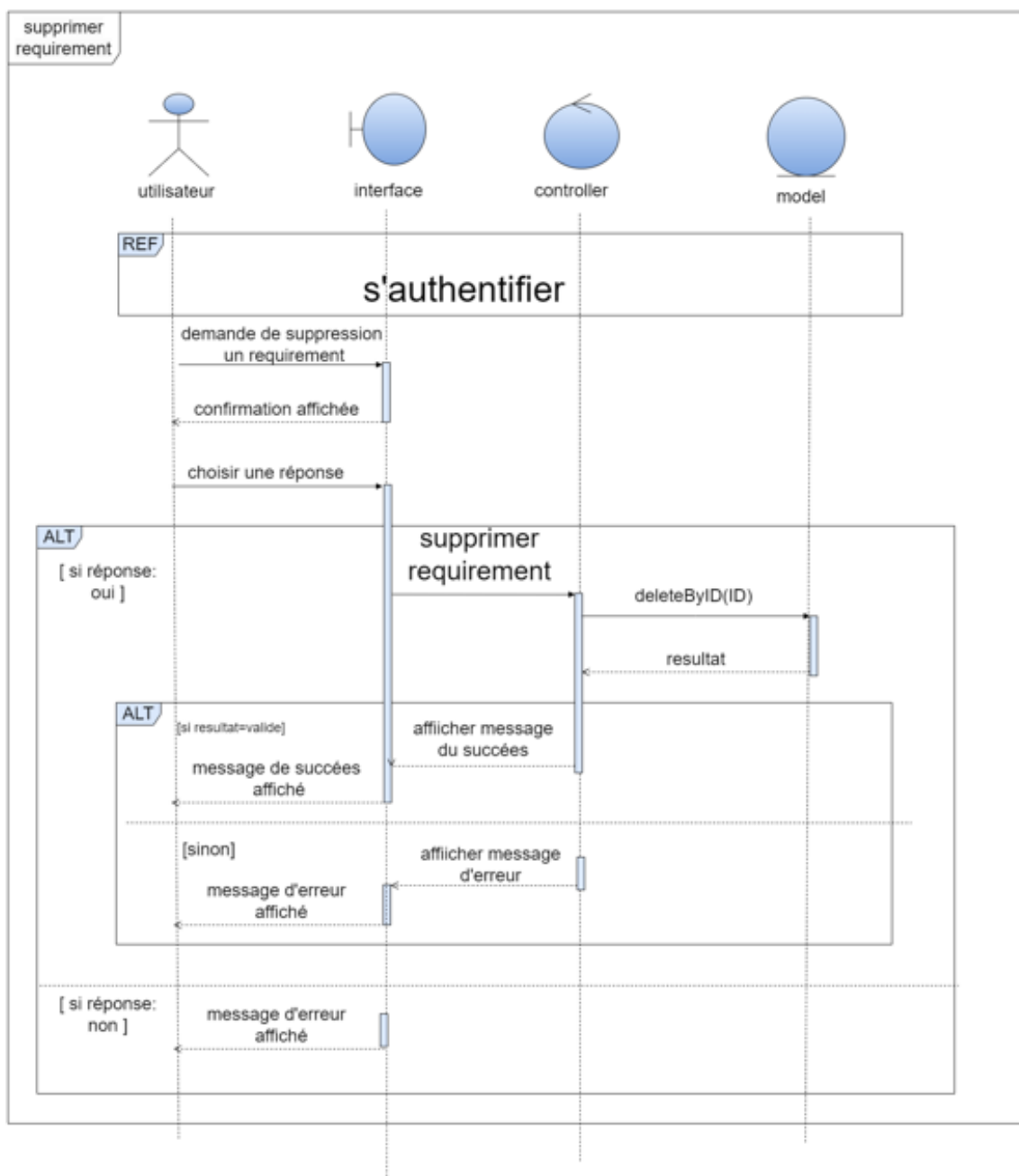


FIGURE 3.6 – Diagramme de séquence relatif au cas d'utilisation supprimer requirement

La figure 3.6 illustre le diagramme de séquence pour le cas d'utilisation Supprimer requirement. L'utilisateur peut supprimer des requirements (demandes) après la consultation grâce à la fonctionnalité Supprimer. Une demande de confirmation sera affichée s'il valide la suppression, dans ce cas la demande sera supprimée de la base de données et le système affiche un message de succès, sinon en cas d'erreur l'application affiche un message d'échec.

3.6 Diagramme d'activité

Le diagramme d'activité permet de mettre l'accent sur les traitements. Il est donc particulièrement adapté à la modélisation du cheminement des flots de contrôle et des flots de données. Il permet ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

55 Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles. Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre.

Dans ce qui suit, nous présentons notre diagramme d'activité pour l'authentification et Gérer KPI.

La figure 3.7 illustre le diagramme d'activité du process authentification.

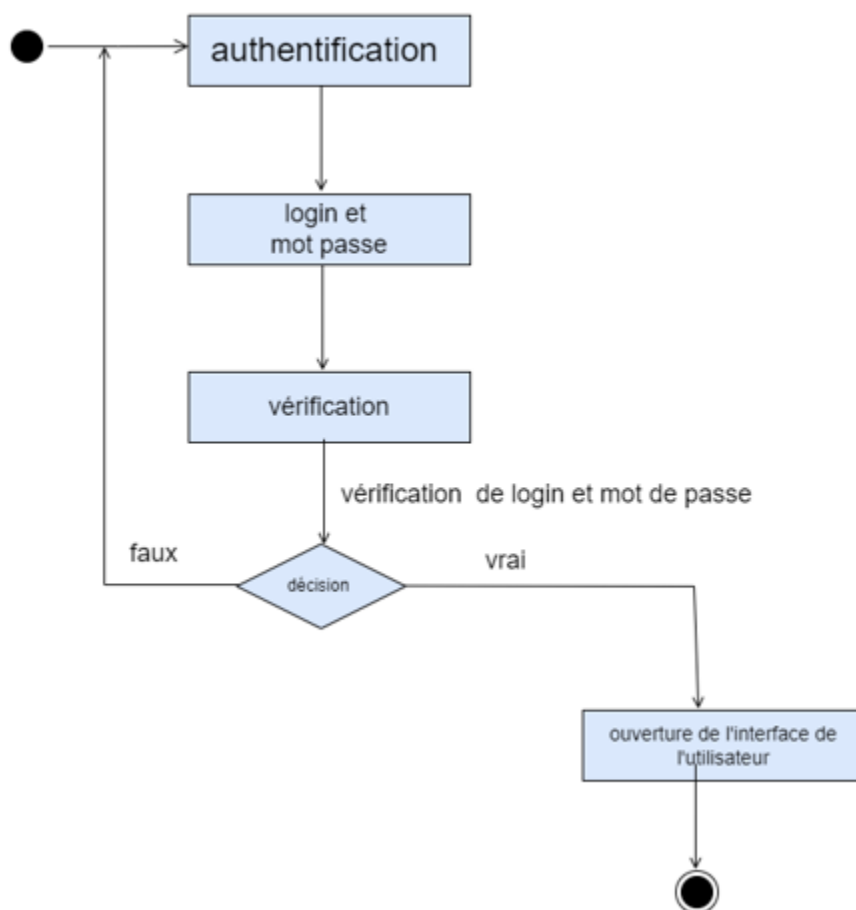


FIGURE 3.7 – diagramme d'activité pour l'authentification

La figure 3.8 illustre le diagramme d'activité du process Gérer KPI.

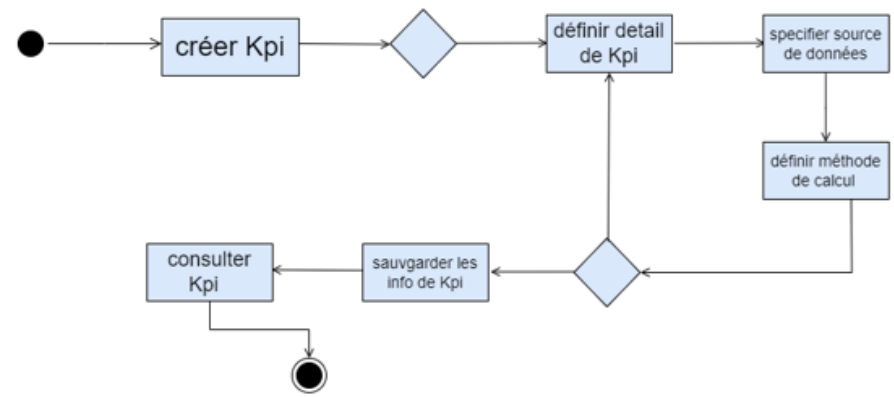


FIGURE 3.8 – diagramme d'activité pour Gérer KPI

Conclusion

La phase de conception sert à identifier les différents objets contribuant pour assurer les fonctionnalités souhaitées et les types des relations qui existent entre eux.

Cette phase est une préparation à la phase d'implémentation garantissant une organisation claire et précise et une facilité d'implémentation des classes invoquées et les relations qui existent entre ces dernières. Le chapitre suivant décrit l'implémentation de ces différentes classes et décrit les fonctionnalités réalisées suite à cette phase conceptuelle.

CHAPITRE 4

APERÇU CONCEPTUEL

Introduction

La phase de conception est une étape critique dans le cycle de développement d'un projet. Elle nécessite tout d'abord une clarification de la vue globale, suivie d'une description détaillée de la conception. Cette description comprend les vues statiques via le diagramme de classes d'entités, ainsi que les vues dynamiques détaillant le fonctionnement de l'application à travers les diagrammes de séquence, de participant et de classe de conception. Enfin, nous concluons avec les diagrammes de navigation de conception pour finaliser l'architecture générale.

4.1 Architecture physique

Architecture 3-tiers

Pour la réalisation de notre solution, nous avons opté pour l'architecture 3-tiers[?]qui est un modèle logique visant à séparer nettement trois couches logicielles au sein d'une même application. Les trois niveaux de cette architecture sont décrits brièvement comme suit :

- **Couche présentation** il s'agit de la partie visible de l'application qui permet à l'utilisateur d'interagir avec le système.
- **Couche applicative** : c'est la partie qui implémente la logique, les différentes règles de gestion et qui décrit les opérations que l'application effectue sur les données en fonction des requêtes des utilisateurs réalisées au travers de la couche présentation.
- **Couche accès aux données** : c'est la couche qui gère l'accès aux données. Ces données sont récupérées et retournées à la couche applicative pour être traitées et renvoyées à l'utilisateur.

4.2 Patrons de conception

4.2.1 Patron de conception MVC :

Le patron de conception MVC (Modèle-Vue-Contrôleur) est un modèle d'architecture logicielle largement utilisé dans le développement d'applications. Il vise à séparer les préoccupations en divisant l'application en trois composants distincts, chacun ayant un rôle spécifique :

1. Modèle (Model) : Le modèle représente les données et la logique métier de l'application. Il gère la récupération, la manipulation et la gestion des données. Le modèle est généralement indépendant de l'interface utilisateur ou de la présentation.

2. Vue (View) : La vue est responsable de l'interface utilisateur et de l'affichage des données au format approprié. Elle se concentre sur la présentation et l'interaction avec l'utilisateur. La vue n'a pas de logique métier significatif et reflète simplement l'état du modèle.

3. Contrôleur (Controller) : Le contrôleur agit comme un médiateur entre le modèle et la vue. Il reçoit les entrées de l'utilisateur via la vue, traite ces entrées et met à jour le modèle en conséquence. Le contrôleur gère également les interactions entre les autres composants et assure la cohérence globale de l'application.

L'objectif principal du patron MVC est de favoriser la séparation des préoccupations et la maintenabilité du code. Cela permet aux développeurs de travailler indépendamment sur chaque composant et de réduire les interdépendances.

Voici comment chaque composant interagit dans le patron MVC :

1. L'utilisateur interagit avec l'interface utilisateur (vue) en fournissant des entrées.
2. Le contrôleur reçoit ces entrées et les traite.
3. Le contrôleur met à jour le modèle avec les modifications appropriées.
4. La vue observe les changements dans le modèle et les reflète à l'utilisateur.

L'architecture MVC est utilisée dans de nombreuses technologies et frameworks, y compris les applications web, les applications mobiles et les applications de bureau. Cela aide à rendre le code plus organisé, maintenable et évolutif.

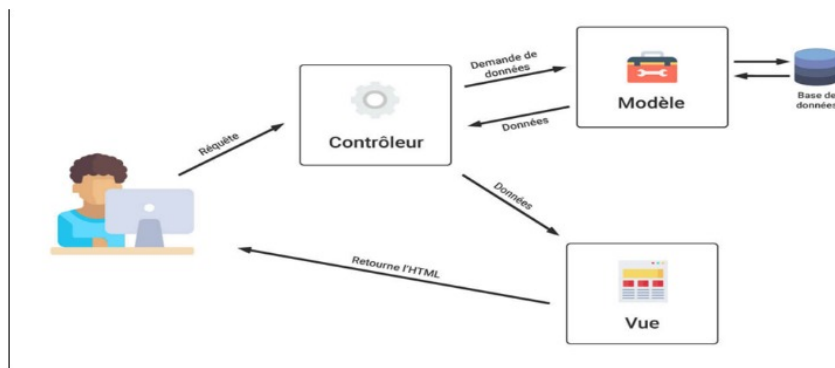


FIGURE 4.1 – architecture MVC

4.2.2 Patron de conception DAO :

Le patron de conception DAO est un patron de conception qui permet de séparer la logique métier de la logique d'accès aux données.

Dans un projet utilisant le patron DAO, les objets de données sont représentés par des classes d'entités, qui sont généralement simples et contiennent uniquement les données nécessaires. Les classes DAO fournissent des méthodes pour accéder aux données stockées dans la source de données, comme les méthodes CRUD (CREATE, READ, UPDATE, DELETE).

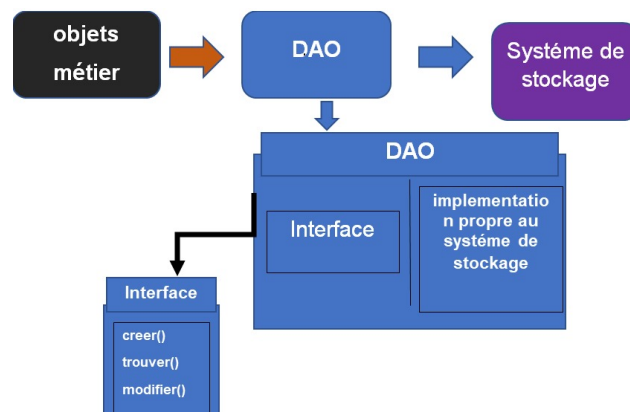


FIGURE 4.2 – principe du patron de conception DAO

4.2.3 Patron de conception DTO :

Le patron de conception DTO, illustré par la figure 3.4 est un modèle qui permet de séparer la logique métier de l'application de la logique de transferts de données. Le principe de ce patron est de créer une classe qui contient les données que l'on souhaite transférer, sans comportement métier associé. Cette classe est souvent appelée DTO et est utilisée pour encapsuler les données et les transférer entre différentes couches de l'application ou entre différents systèmes.

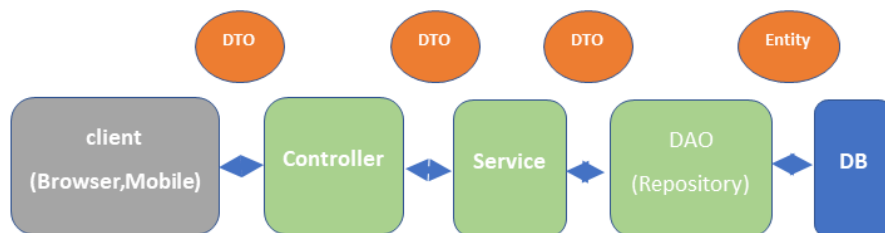


FIGURE 4.3 – principe du patron de conception DTO

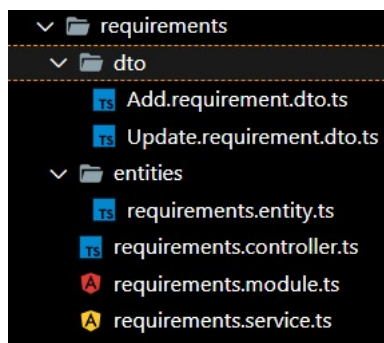


FIGURE 4.4 – exemple dto de l'application

4.2.4 Patron de conception IOC

Le patron de conception Inversion de Contrôle (IoC) est un concept fondamental en développement logiciel qui vise à inverser le contrôle de certaines parties d'une application. L'IoC encourage la délégation du contrôle de la création et de la gestion des objets à un conteneur IoC (également appelé conteneur de dépendances) plutôt que de les créer directement dans le code.

L'IoC a plusieurs avantages, notamment la réduction du couplage entre les composants d'une application, la facilitation du test unitaire et la gestion des dépendances.

Un des patrons de conception couramment utilisés pour mettre en œuvre l'IoC est le patron de conception "Conteneur de dépendances" (Dependency Injection Container).

Voici comment fonctionne l'IoC et la Dependency Injection (DI) :

1. Inversion de Contrôle (IoC) : Traditionnellement, dans un programme, le contrôle de la création et de la gestion des objets est entre les mains du développeur. Avec IoC, le contrôle est inversé – le conteneur IoC prend en charge la création et la gestion des objets.

2. Dependency Injection (DI) : L'une des pratiques clés de l'IoC est l'injection de dépendances. Plutôt que de créer des dépendances à l'intérieur d'une classe, elles sont injectées depuis l'extérieur. Cela rend

les classes plus modulaires et faciles à tester, car vous pouvez injecter des dépendances simulées lors des tests.

L'utilisation de l'IoC et de la DI peut varier selon le langage de programmation et le framework que vous utilisez, mais le concept fondamental reste le même : déléguer la gestion des dépendances et le contrôle de la création d'objets à un conteneur IoC.

4.3 Conception détaillée

4.3.1 Diagramme de classe

Le diagramme de classe[8] permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Généralement, il est adapté pour détailler, décomposer ou illustrer la réalisation particulière.

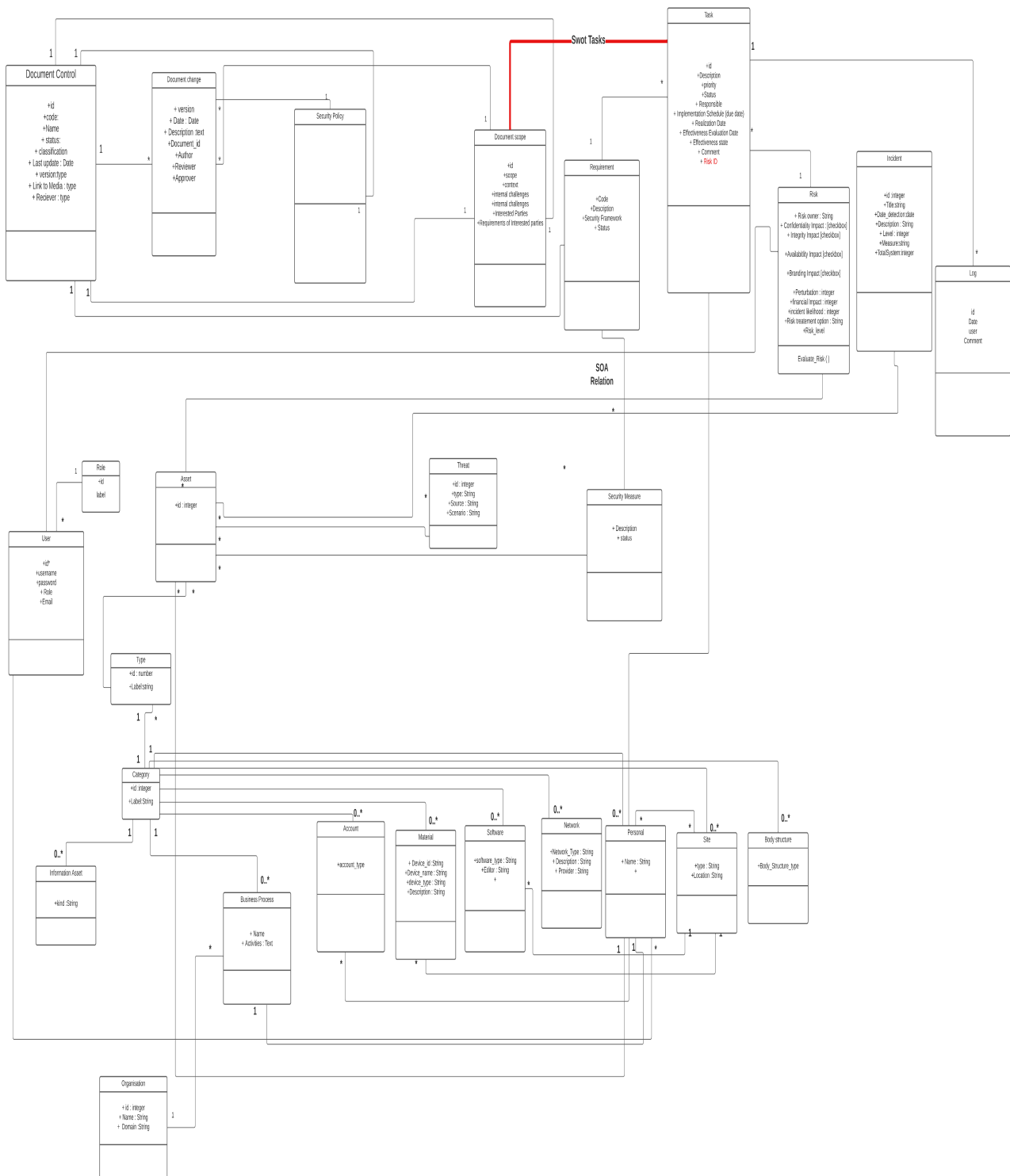


FIGURE 4.5 – Diagramme de classe

Conclusion

La conception facilite la mise en œuvre du logiciel en détaillant les différentes perspectives du système à l'aide de divers diagrammes. Le prochain chapitre mettra en œuvre les différents outils utilisés pour la réalisation de notre application, ainsi que les résultats obtenus.

CHAPITRE 5

RÉALISATION

Introduction

Dans ce chapitre, nous présentons les étapes de réalisation de notre solution. Nous décrivons dans une première partie l'environnement de développement logiciel ainsi que les outils utilisés et aussi les frameworks. Dans une seconde partie, nous présentons l'architecture adoptée ainsi qu'une description des interfaces de l'application avec quelques captures d'écrans. Et enfin, on trouvera la partie

5.1 Diagramme de déploiement

Un diagramme de déploiement est un type de diagramme UML qui montre l'architecture d'exécution d'un système, y compris les nœuds tels que les environnements d'exécution matériels ou logiciels, et l'intergiciel qui les relie.

Les diagrammes de déploiement sont généralement utilisés pour visualiser le matériel physique et les logiciels d'un système. En l'utilisant, vous pouvez comprendre comment le système sera physiquement déployé sur le matériel.

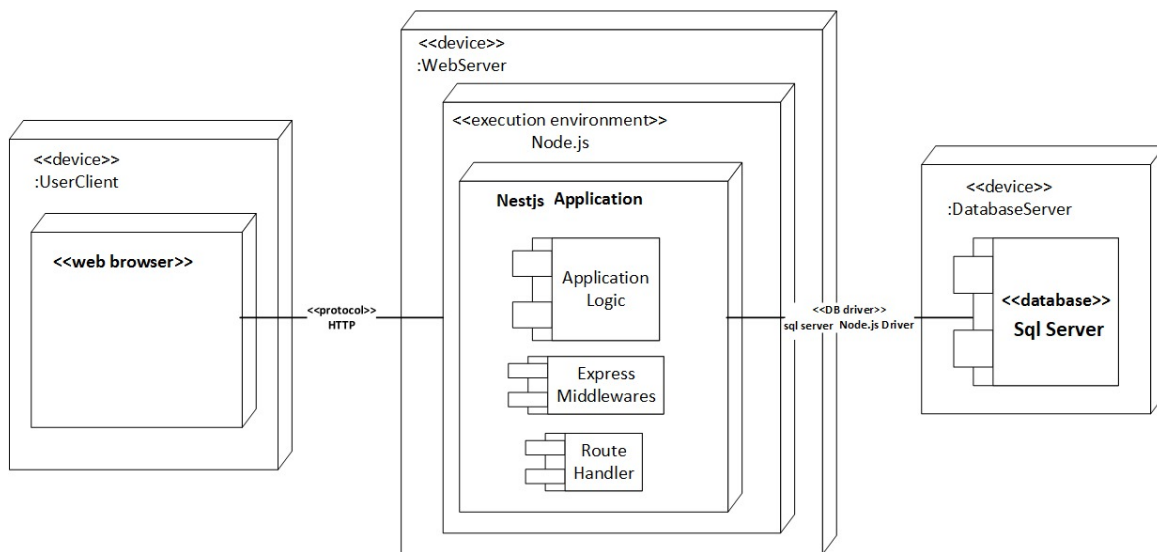


FIGURE 5.1 – Diagramme de déploiement

5.2 Le style architectural

5.2.1 Partie front-end

Architecture MVVM

En architecture logicielle, MVVM (Model View ViewModel) est un Design pattern (Modèle de conception) visant à séparer la logique de présentation d'une application en 3 couches

- Model** : Le modèle contient les données liées à la logique métier. Il peut s'agir par exemple d'entités issues de bases de données ou encore d'API externes. Les modèles métiers sont conçus et optimisés pour le bon fonctionnement de la persistance et/ou de la transmission des données (backend). Les modèles métiers ne sont en aucun cas dépendants de la manière dont elles seront présentées sur une interface graphique (frontend).
- View** : La vue est la description de l'interface graphique, elle fait le lien entre les actions de l'utilisateur et le modèle de vue. Elle définit où et comment sont placés les composants graphiques sur l'interface et décrit les liaisons de données (Data Bindings) entre les valeurs affichées et le modèle de vue.
- ViewMode** : Le modèle de vue est chargé de transformer et d'organiser les modèles métiers afin d'exposer les données à afficher par la vue. Si un changement critique intervient au niveau du modèle métier, le view modèle peut être adapté sans en impacter la vue. En revanche, la vue peut évoluer sans impacter les modèles de vue ou les modèles métiers.

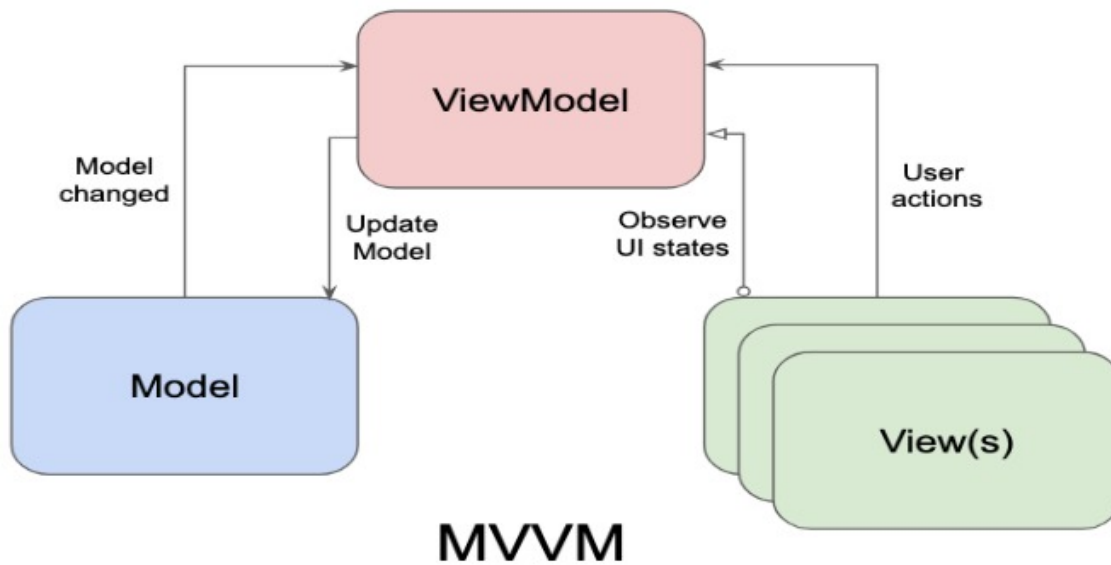


FIGURE 5.2 – Architecture MVVM

5.2.2 Partie back-end

Architecture Microservices

Dans ce qui suit, nous allons présenter l'architecture microservices et les avantages de son utilisation.

A-Présentation ;

De nos jours, l'adoption de l'architecture Microservices est devenue une obligation pour les entreprises qui travaillent sur des systèmes informatiques très larges. C'est une approche architecturale parfaitement compatible avec les méthodes de management Agile et la culture DevOps.

Microservices et DevOps un couple en parfaite harmonie : la mise en place d'une architecture en microservices est couplée à la mise en place d'une organisation DevOps. L'architecture Microservices repose sur la réduction et le fractionnement des composants d'application. Ceci est conforme à la culture DevOps qui favorise l'utilisation des conteneurs permettant aux composants de fonctionner indépendamment les uns des autres.

La figure suivante représente l'architecture en microservices.

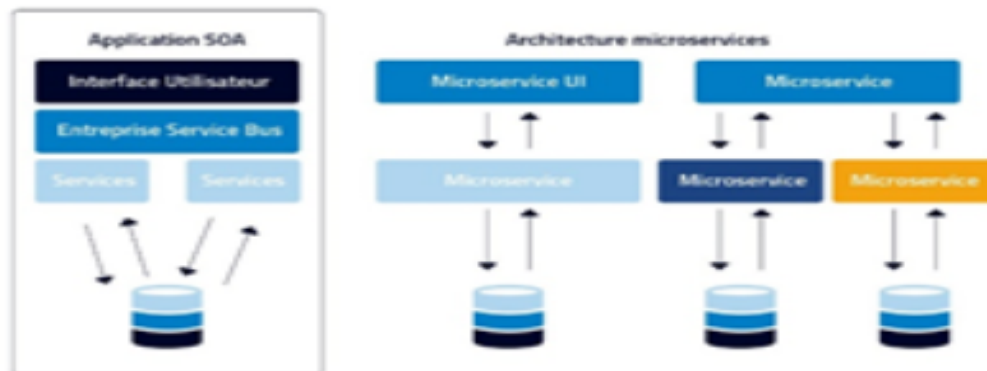


FIGURE 5.3 – L'architecture Microservices

B-Avantages ;

Dans une architecture Microservice, le logiciel est décomposé en un ensemble de services hautement indépendants.

Chaque Microservice peut être :

- Conçu et développé indépendamment
- Testé indépendamment
- Déployé indépendamment

C-Intégration des microservices

L'architecture Microservices[9] est implémentée pour des applications logicielles complexes et volumineuses.

Une application développée suivant l'architecture microservices est présentée sous forme d'un ensemble de services faiblement couplés indépendants l'un de l'autre. Ces services sont aussi autonomes et isolés, mais ils peuvent communiquer entre eux, fournissant les données nécessaires.

Notre application est composée de plusieurs microservices pouvant communiquer entre eux.

La communication entre les clients et les microservices, ou entre les microservices eux-mêmes, est implémentée à travers un protocole d'échanges http REST.

La figure suivante montre nos différents microservices « Requirements » « tasks » « Process vérification KPI » qui sont déployés indépendamment avec des adresses différentes [Wi].

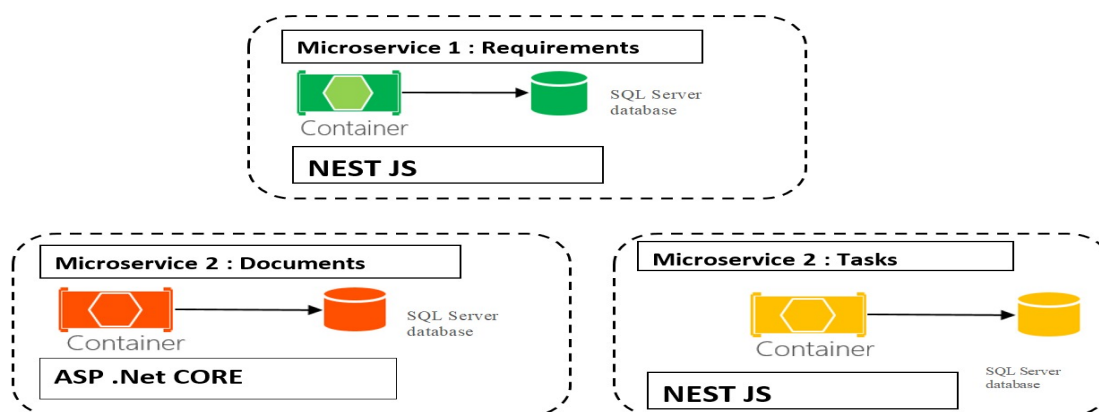


FIGURE 5.4 – L'architecture Microservices

5.3 Environnement de développement

5.3.1 Technologie de programmation

Node.js

Node.js est une plateforme logicielle open-source, basée sur le moteur JavaScript V8 de Google, qui permet d'exécuter du code JavaScript côté serveur



FIGURE 5.5 – Logo node.js

React

React[10] est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état

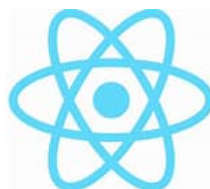


FIGURE 5.6 – Logo React

5.3.2 Langage de programmation

TypeScript

TypeScript est un langage de programmation open-source développé par Microsoft. Il s'agit d'une surcouche (superset) de JavaScript qui ajoute des fonctionnalités supplémentaires au langage, notamment le typage statique.



FIGURE 5.7 – Logo Typescript

Nestjs

Nest.js est un framework open-source pour la création d'applications côté serveur en utilisant TypeScript, basé sur Node.js. Il est conçu pour être extensible, modulaire et hautement performant. Nest.js utilise une approche basée sur les classes et les décorateurs pour faciliter le développement d'applications backend robustes et évolutives.



FIGURE 5.8 – Logo Nestjs

5.3.3 Base de données

Sql server

Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.



FIGURE 5.9 – Logo Sql server

5.3.4 Outils

Azur devops

Azure est la plateforme de cloud public de Microsoft. Azure offre une grande collection de services, qui inclut les fonctionnalités PaaS (platform as a service), IaaS (infrastructure as a service) et du service de base de données managée



FIGURE 5.10 – Logo Azur devops

Visual studio code

visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft. C'est un logiciel gratuit et open-source qui est largement utilisé par les développeurs pour écrire, modifier et déboguer du code dans différents langages de programmation.



FIGURE 5.11 – Logo Visual studio code

Postman

Postman[11] est une application permettant de tester des API, créée en 2012 par Abhinav Asthana



FIGURE 5.12 – Logo Azur devops

Draw.io

C'est un outil qui permet aux utilisateurs de créer différents types de diagrammes tels que des organigrammes, des diagrammes de flux, des diagrammes de réseau, des schémas de base de données, etc. de manière collaborative et intuitive.



FIGURE 5.13 – Logo Draw.io

Overleaf

est un éditeur LaTeX collaboratif en ligne et en temps réel



FIGURE 5.14 – Logo Overleaf

GitHub

GitHub[12] est une plateforme de développement collaboratif basée sur Git. Elle permet aux développeurs de travailler ensemble sur des projets de logiciels en utilisant un système de contrôle de version distribué



FIGURE 5.15 – Logo Github

5.4 Développement des interfaces

5.4.1 L'interface authentification

Dans la figure 5.16, nous avons inclus une représentation visuelle de l'interface de connexion. Cette interface est conçue pour permettre aux utilisateurs d'accéder au système en se connectant avec leurs identifiants et leurs informations d'authentification. L'objectif principal de cette interface est de fournir aux utilisateurs un moyen sécurisé et convivial pour accéder à leurs comptes et utiliser les fonctionnalités du système en tant qu'utilisateur autorisé.

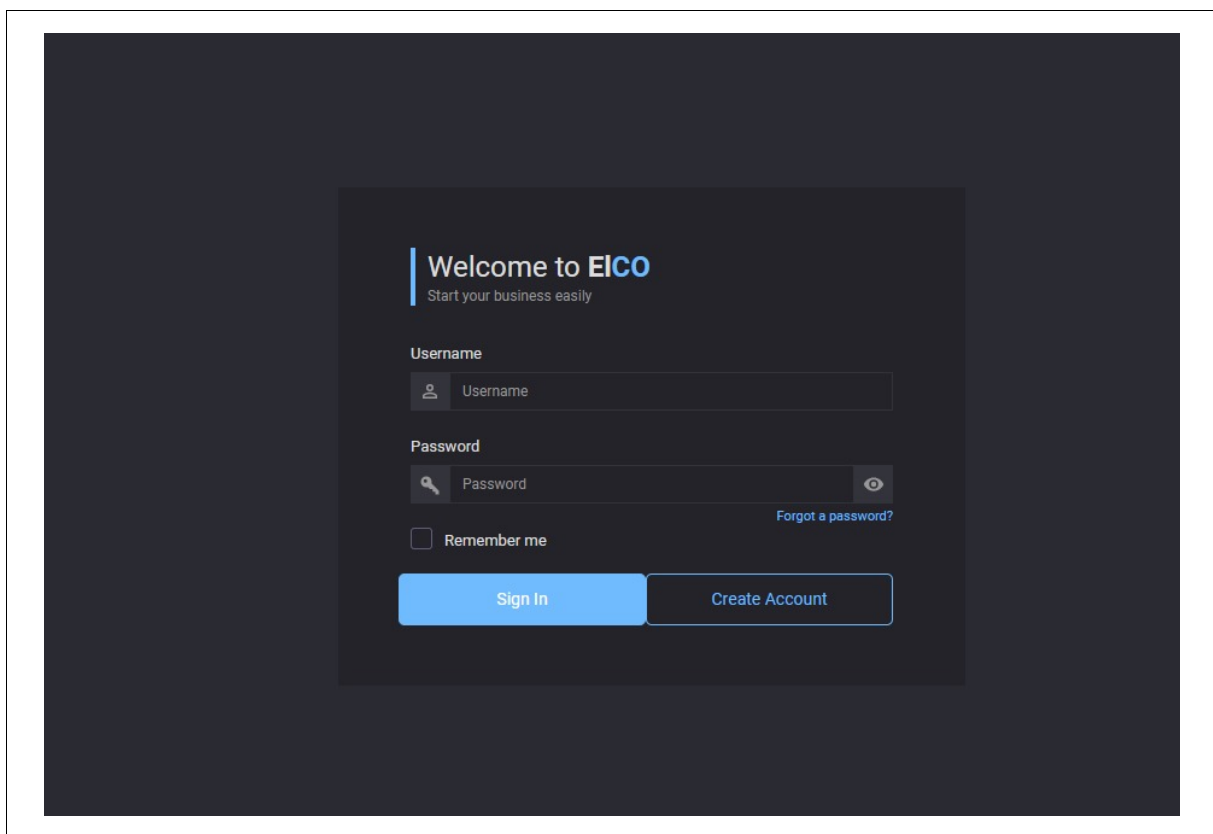


FIGURE 5.16 – L'interface authentification

5.4.2 L'interface Dashboard

Dans la figure numérotée 5.17 de notre rapport, nous présentons une illustration visuelle de l'interface du tableau de bord. Le tableau de bord est une composante cruciale de notre système, conçu pour fournir une vue consolidée et visuellement informative des performances clés du projet. Cette interface offre une perspective synthétique de la manière dont le projet progresse en termes d'indicateurs clés de performance (KPI) que nous avons définis.

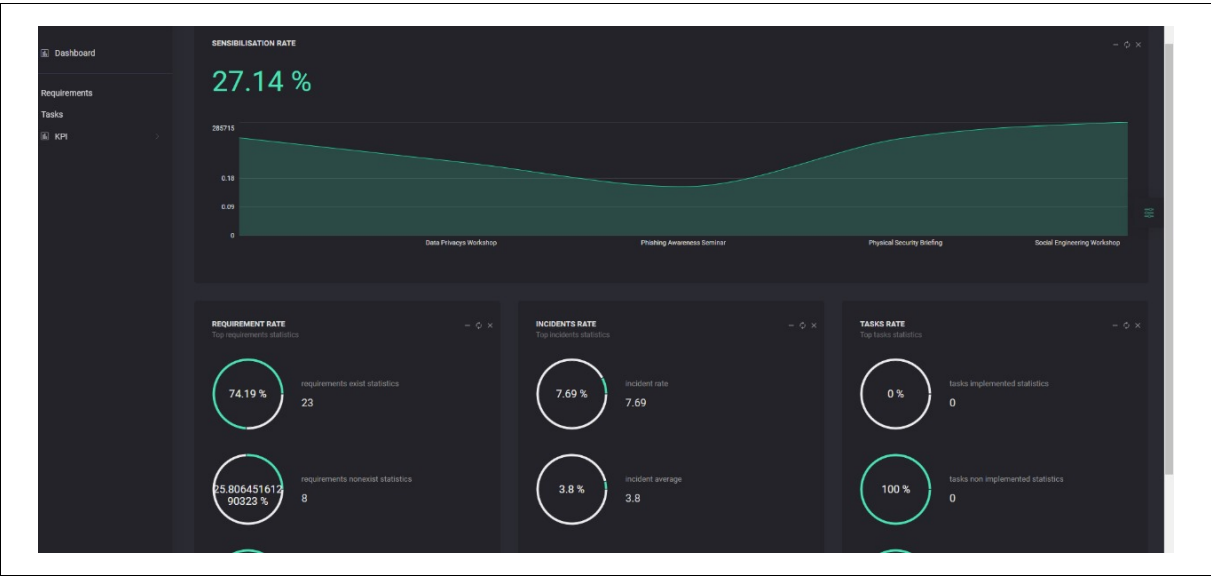


FIGURE 5.17 – L'interface Dashboard

L'interface affichée dans la figure 5.18 offre un aperçu ciblé des performances du projet en ce qui concerne la conformité aux exigences définies (Requirements). Les KPI calculés pour évaluer la conformité et l'atteinte des objectifs liés aux exigences sont rassemblés et visualisés de manière claire et concise dans cette section.

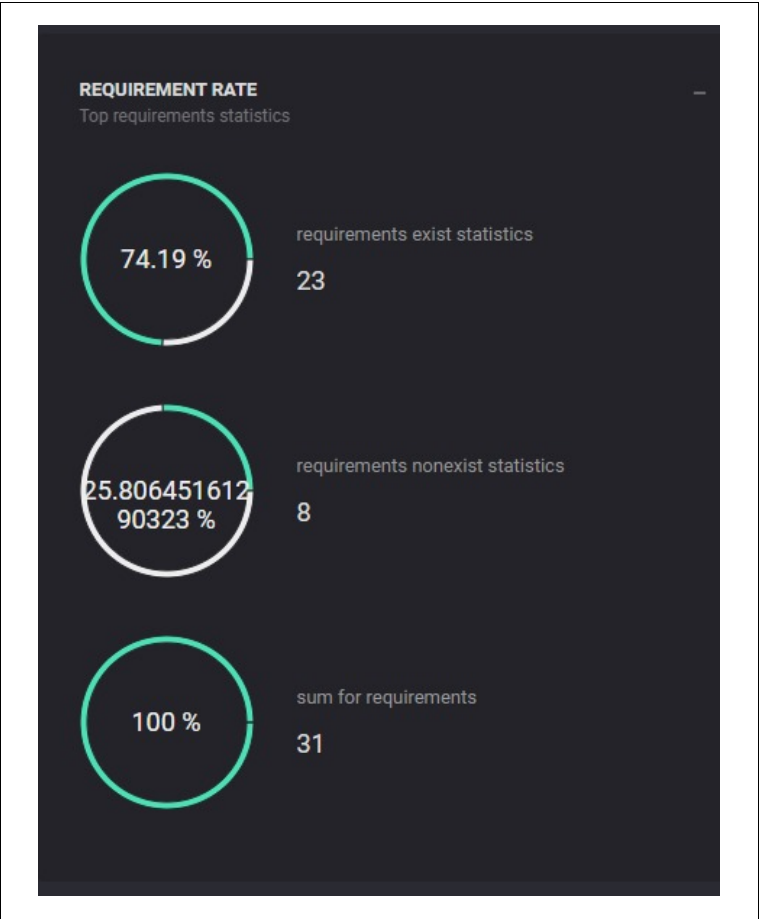


FIGURE 5.18 – L'interface results requiremets

L'interface illustrée dans la figure 5.19 offre un espace dédié pour évaluer l'efficacité des initiatives de sensibilisation en matière de sécurité au sein du projet. Les KPI calculés pour mesurer le niveau de sensibilisation des utilisateurs aux pratiques de sécurité sont regroupés et affichés de manière claire et concise dans cette section.

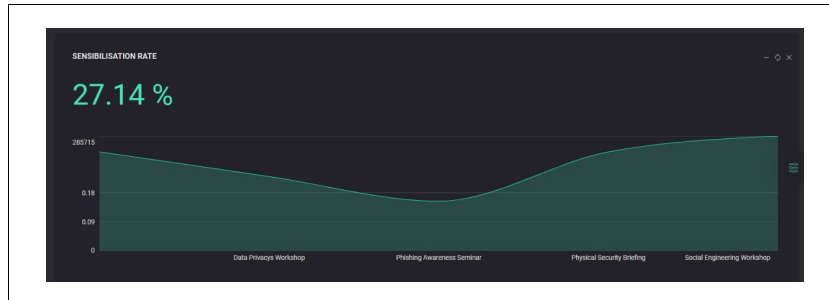


FIGURE 5.19 – L'interface results Kpi sensibilisation

5.4.3 L'interface requirements

Les figures 5.20 et 5.21 se combinent pour présenter une vue approfondie de l'interface de gestion des requirements. Dans la première partie (Figure 5.19), nous décrivons un formulaire dédié à l'ajout et à la modification des requirements, accompagné d'une barre de progression qui affiche un résultat de calcul spécifique.

L'interface du formulaire de gestion des requirements, illustrée dans la Figure 5.19, offre aux utilisateurs un moyen structuré pour ajouter de nouvelles requirements ou modifier celles existantes. Le formulaire est conçu pour faciliter la saisie et la mise à jour des informations relatives aux requirements du projet. Cette section met en évidence l'importance d'un processus organisé et convivial pour capturer les exigences de manière précise et efficace.

La barre de progression intégrée à l'interface revêt une signification particulière. Elle sert à afficher un résultat de calcul spécifiquement lié aux exigences que l'utilisateur saisit ou modifie. Cette approche visuelle permet aux utilisateurs de visualiser instantanément comment les modifications qu'ils apportent aux exigences influencent un résultat déterminé. Par exemple, la barre de progression pourrait représenter le niveau de conformité atteint par rapport aux objectifs définis pour les exigences spécifiques.

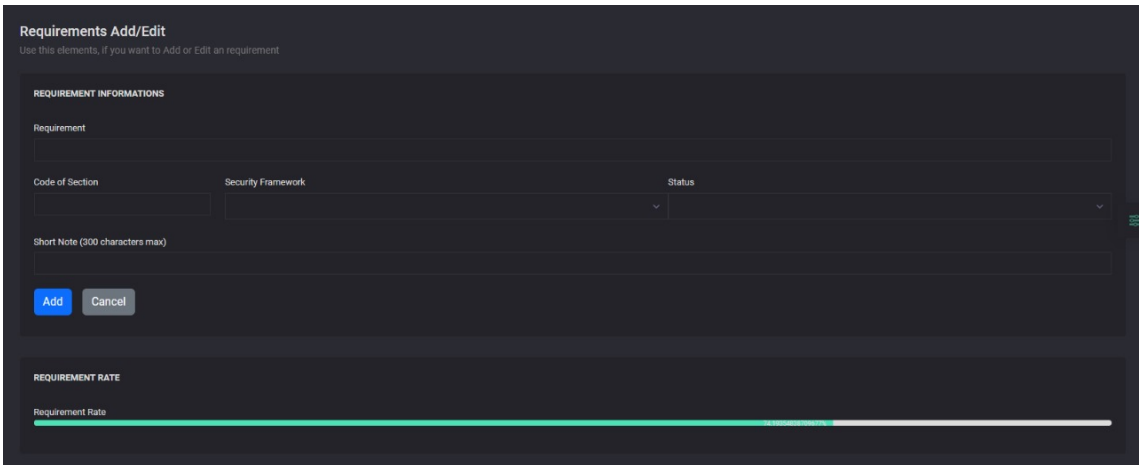
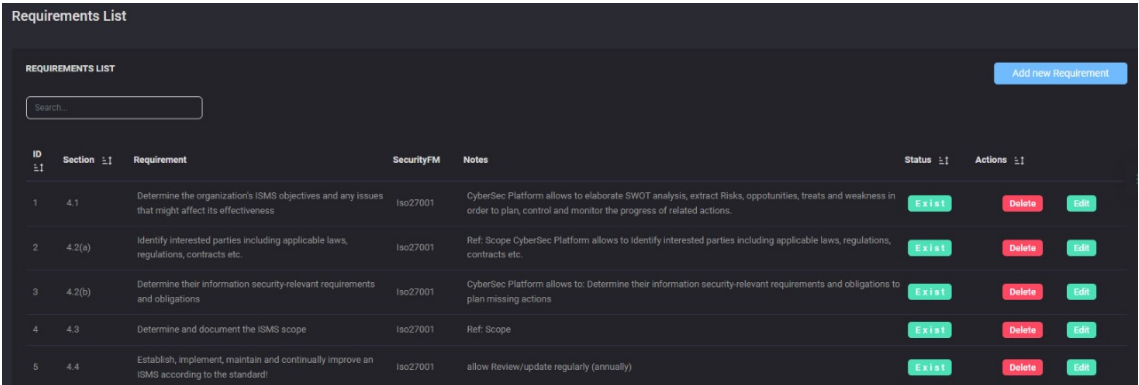


FIGURE 5.20 – L'interface d'ajouter et de modifier de requirements

L'interface présentée dans la Figure 5.21 permet aux utilisateurs d'accéder à une vue panoramique de toutes les exigences enregistrées dans le système. Les requirements sont affichées dans une disposition tabulaire, avec chaque ligne représentant une requirement distincte. Cette présentation structurée facilite l'inspection rapide de l'ensemble des requirements et permet aux utilisateurs de prendre des décisions éclairées concernant la gestion de celles-ci.

Dans cette interface, chaque ligne d'exigence est associée à des boutons de suppression et de modification. Ces boutons offrent aux utilisateurs la capacité d'effectuer des actions spécifiques pour chaque requirement. Le bouton "Supprimer" permet de retirer une exigence du système, tandis que le bouton "Modifier" autorise les utilisateurs à apporter des ajustements aux détails de l'exigence. Cette fonctionnalité garantit une gestion dynamique et flexible des requirements, en permettant des mises à jour en temps réel et des ajustements selon les besoins du projet.



The screenshot shows a web interface titled "Requirements List". It features a search bar at the top left and an "Add new Requirement" button at the top right. Below these is a table with the following columns: ID, Section, Requirement, SecurityFM, Notes, Status, and Actions. The table contains 5 rows of data, each with a unique ID and a corresponding requirement description. The Status column for all rows shows "Exist". The Actions column for each row contains two buttons: "Delete" (red) and "Edit" (green).

ID	Section	Requirement	SecurityFM	Notes	Status	Actions
1	4.1	Determine the organization's ISMS objectives and any issues that might affect its effectiveness	Iso27001	CyberSec Platform allows to elaborate SWOT analysis, extract Risks, opportunities, treats and weakness in order to plan, control and monitor the progress of related actions.	Exist	Delete Edit
2	4.2(a)	Identify interested parties including applicable laws, regulations, contracts etc.	Iso27001	Ref: Scope CyberSec Platform allows to identify interested parties including applicable laws, regulations, contracts etc.	Exist	Delete Edit
3	4.2(b)	Determine their information security-relevant requirements and obligations	Iso27001	CyberSec Platform allows to: Determine their information security-relevant requirements and obligations to plan missing actions	Exist	Delete Edit
4	4.3	Determine and document the ISMS scope	Iso27001	Ref: Scope	Exist	Delete Edit
5	4.4	Establish, implement, maintain and continually improve an ISMS according to the standard	Iso27001	allow Review/update regularly (annually)	Exist	Delete Edit

FIGURE 5.21 – L'interface liste de requirements

Conclusion

À travers ce chapitre, nous avons présenté la phase réalisation où nous avons utilisé des environnements techniques et des logiciels spécifiques pour développer notre projet. Nous avons présenté notre application en illustrant ses principales fonctionnalités. En conclusion, nous avons réussi à établir une version de l'application qui sera améliorée dans les prochaines versions.

CONCLUSION GÉNÉRALE

En conclusion, ce rapport de projet de fin d'études a mis en évidence l'importance cruciale de la mise en place d'un système de gestion de la sécurité de l'information (ISMS) conforme aux normes ISO 27001 et TISAX. À travers une analyse approfondie, des recherches exhaustives et une mise en œuvre pratique, ce projet a démontré les avantages tangibles et les défis inhérents à l'adoption de ces normes de sécurité reconnues à l'échelle internationale.

L'implémentation de l'ISMS a permis d'établir des politiques, des procédures et des contrôles rigoureux pour protéger les informations sensibles et garantir la confidentialité, l'intégrité et la disponibilité des données. La démarche a non seulement renforcé la posture de sécurité de l'organisation, mais a également renforcé la confiance des parties prenantes, des clients et des partenaires dans la gestion des risques liés à la sécurité de l'information.

Les efforts entrepris dans le cadre de ce projet ouvrent la voie à plusieurs perspectives prometteuses. Tout d'abord, la continuité du processus d'ISMS doit être assurée, avec des révisions régulières pour garantir que les contrôles demeurent pertinents et efficaces face à l'évolution des menaces et des technologies. L'intégration d'outils de surveillance et d'analyse des vulnérabilités peut également renforcer davantage la résilience de l'organisation.

En outre, l'adoption d'une approche plus proactive de la sensibilisation à la sécurité de l'information peut être envisagée. Des formations régulières et des simulations d'attaques peuvent sensibiliser davantage les employés aux risques et renforcer leur rôle en tant que maillons forts de la chaîne de sécurité.

Enfin, l'exploration de synergies entre l'ISMS et d'autres cadres de sécurité, tels que les réglementations de protection des données (RGPD) ou les normes spécifiques à l'industrie, peut aider à créer une architecture de sécurité holistique.

En somme, ce projet offre une base solide pour la pérennisation et l'amélioration continue du système ISMS selon les normes ISO 27001 et TISAX. Les perspectives tracées permettent d'apporter des améliorations continues et de faire face aux défis émergents dans un paysage en constante évolution de la sécurité de l'information tel que l'ajout d'un système de reporting .

BIBLIOGRAPHIE

- [1] ELCO SOLUTIONS URL : : <https://www.elco-solutions.de> [consulté le 15/02/2023]
- [2] Les méthodes Agiles , URL : <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-agile> [consulté le 15/02/2023]
- [3] SMSI, URL : <https://www.isms.online/information-security-management-system-isms/> [consulté le 15/02/2023]
- [4] iso27001 <https://www.iso.org/fr/standard/27001> : :text=La [consulté le 15/02/2023]
- [5] TISAX , URL : <https://www.dqsglobal.com/fr-tn/certifier/certification-tisax> consulté le 08/03/2023 [consulté le 15/02/2023]
- [6] UML, URL : <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi>
- [7] Diagramme de séquence, URL : <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours05.html> [consulté le 15/06/2023]
- [8] Diagramme de Classes, URL : <https://laurent-audibert.developpez.com/Cours-UML/page=diagramme-classes> [consulté le 02/04/2023]
- [9] Microservices, URL : <https://www.redhat.com/fr/topics/microservices/what-are-microservices> [consulté le 22/04/2023]
- [10] React JS , URL : <https://fr.reactjs.org/> [consulté le 18/03/2023]
- [11] Postman , URL : <https://www.logiciels.pro/logiciel-saas/postman/> [consulté le 15/02/2023]
- [12] GitHub , URL : <https://git-scm.com/> [consulté le 15/02/2023]