



N° : Ing-GInfo-28-2025

Département Génie Informatique

RAPPORT DE PROJET DE FIN D'ETUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Génie Informatique

Spécialité : Nouvelles technologies et sécurité (NTS)

Par : CHERNI Rihab

**Développement d'une application web "AutoTest" :
Solution pour automatiser les tests fonctionnels, les
analyses de sécurité et l'audit de référencement naturel**

Réalisé au sein de ADDINN Tunis



Soutenu le 30 juin 2025, devant le jury composé de :

Président :

M. BOULARES Mehrez

Rapporteur :

Mme. GHRAIRI Salsabil

Encadrant universitaire :

Mme ELOUEDI Ines

Encadrant industriel :

M. NACHMI Omar

Année Universitaire 2024-2025

Dédicaces

Je dédie ce projet à mes parents, mes sœurs, mes amis, ainsi qu'à tous mes collègues et professeurs. Leur soutien m'a beaucoup aidé et m'a donné la force d'avancer tout au long de ce parcours.

À mes parents, pour leur amour, leur patience et leur courage, surtout dans les moments difficiles. Grâce à eux, j'ai pu recevoir une bonne éducation et avancer vers mes objectifs.

À mes deux sœurs, Arij et Rania, pour leur présence, leurs paroles gentilles et leurs encouragements qui m'ont toujours réconforté.

À mes amis, qui sont comme une seconde famille. Merci pour leur aide, leur bonne humeur et leur soutien tout au long de ce chemin.

Et à toutes les personnes qui m'ont aidé, de près ou de loin, un grand merci du fond du cœur.

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce projet de fin d'études et à l'enrichissement de cette expérience.

Tout d'abord, je remercie **Mme. Ines Elouedi**, mon encadrante universitaire, pour son suivi attentif, ses conseils avisés et son accompagnement constant tout au long de ce projet. Sa bienveillance et son expertise ont été d'un grand soutien dans la réalisation de ce travail.

Je tiens également à exprimer ma reconnaissance à **M. Omar Nachmi**, mon encadrant industriel au sein de ADDINN Tunis, pour sa disponibilité, son encadrement et ses orientations pertinentes, qui m'ont permis de mener ce projet à bien dans un cadre professionnel enrichissant.

Mes sincères remerciements vont aussi à toute l'équipe de ADDINN Tunis pour leur accueil chaleureux et leur esprit collaboratif, qui ont rendu cette expérience aussi formatrice qu'agréable.

Je remercie également les membres du jury **Mme. Salsabil Ghrairi** et **M. Mehrez Boulares** pour l'intérêt qu'ils portent à mon travail et pour leurs précieuses évaluations.

Je suis reconnaissant envers tous mes enseignants de l'**École Nationale Supérieure d'Ingénieurs de Tunis (ENSIT)** pour leur engagement et leur contribution à ma formation.

Enfin, je remercie ma famille et mes amis pour leur soutien inconditionnel, leurs encouragements constants et leur présence précieuse tout au long de cette aventure.

Un grand merci à tous.

Table des matières

Introduction générale	1
Chapitre 1: Cadre général du projet	3
 Introduction	3
1. Contexte général du projet	3
2. Présentation de l'organisme d'accueil	3
2.1. Aperçu général d'Addinn	3
2.2. Fondements de la Société	3
2.3. Fiche de présentation d'ADDINN	4
2.4. Histoire de croissance d'ADDINN et présence internationale	4
2.5. Secteurs d'expertise et partenaires stratégiques	5
3. Étude de l'existant	6
3.1. État actuel de l'application existante	6
3.2. Analyse et critique des applications similaires	9
4. Problématique	17
5. Solution proposée	17
6. Choix méthodologique de travail	18
 Conclusion	19
Chapitre 2: Analyse et planification du projet	20
 Introduction	20
1. Spécification et analyse des besoins	20
1.1. Identification des acteurs	20
1.2. Besoins fonctionnels	21
1.3. Besoins non fonctionnels	25
2. Diagramme de cas d'utilisation global	25
3. Pilotage du projet avec Scrum	26
3.1. Présentation de l'équipe de travail	26
3.2. Outils SCRUM utilisés	27
3.3. Backlog du produit	27
3.4. Planification des sprints	28
4. Planning prévisionnel du projet	28
5. Rétrospective Agile	29
6. Environnement de développement	30
6.1. Environnement matériel	30
6.2. Environnement logiciel	31
7. Architecture de l'application	35
 Conclusion	37

Chapitre 3: Release 1 : Automatisation des tests de sécurité et amélioration des fonctionnalités de base	38
Introduction	38
1. Planification de la release 1	38
2. Sprint 1.1 : Initialisation, authentification et gestion des permissions	38
2.1. Backlog du sprint 1.1	38
2.2. Analyse du sprint 1.1	42
2.3. Conception du sprint 1.1	44
2.4. Réalisation du sprint 1.1	46
3. Sprint 1.2 : Tests de sécurité et notifications	48
3.1. Backlog du sprint 1.2	48
3.2. Analyse du sprint 1.2	53
3.3. Conception du sprint 1.2	54
3.4. Réalisation du sprint 1.2	58
Conclusion	60
Chapitre 4: Release 2 : Automatisation des tests fonctionnels et SEO, génération des rapports et dockerisation	61
Introduction	61
1. Planification de la release 2	61
2. Sprint 2.1 : Tests automatisés fonctionnels et SEO	61
2.1. Backlog du sprint 2.1	61
2.2. Analyse du sprint 2.1	64
2.3. Raffinement des cas d'utilisation	65
2.4. Conception du sprint 2.1	67
2.5. Réalisation du sprint 2.1	70
3. Sprint 2.2 : Finalisation et dockerisation de l'application	73
3.1. Backlog du sprint 2.2	73
3.2. Analyse du sprint 2.2	76
3.3. Conception du sprint 2.2	76
3.4. Réalisation du sprint 2.2	77
3.5. Dockerisation de l'application	79
Conclusion	80
Conclusion générale	81
Annexe A : Limites de PENTRA	
Annexe B : Complément d'analyse et de conception	
Annexe C : Automatisation des outils de pentesting	
Annexe D : Automatisation d'analyse SEO et fonctionnels	
Annexe E : Interfaces de l'application AutoTest	
Bibliographie	

Table des figures

1.1	Logo d'ADDINN	3
1.2	Présence internationale d'ADDINN	5
1.3	Partenaires stratégiques d'ADDINN	5
1.4	Interface de la page d'accueil PENTRA	6
1.5	Interface du l'application Hostedscan	10
1.6	Interface du l'application Invicti	11
1.7	Interface de démonstration de Cypress exécutant une suite de tests	12
1.8	Interface du l'application Katalon Studio	13
1.9	Interface de l'outil Semrush	14
1.10	Interface de l'outil Ahrefs Site Audit	14
1.11	Cadre méthodologique et mode de fonctionnement SCRUM	19
2.1	Diagramme de cas d'utilisation global	26
2.2	Diagramme de Gantt	29
2.3	Scrum board Sprint 1	29
2.4	Burndown chart projet	30
2.5	Architecture de l'application	37
3.1	Diagramme de cas d'utilisation raffiné du sprint 1.1	42
3.2	Diagramme d'activité : Contrôle des accès et visualisation des permissions	44
3.3	Diagramme de classe du sprint 1.1	45
3.4	Interface d'inscription	46
3.5	Interface du profil utilisateur	47
3.6	Interface de gestion des utilisateurs et permissions (admin)	47
3.7	Diagramme de cas d'utilisation raffiné du sprint 1.2	53
3.8	Diagramme de classe du sprint 1.2	55
3.9	Diagramme de séquence de conception de cas «Lancer un scan de test sécurité»	57
3.10	Interface de sélection des outils de sécurité	58
3.11	Interface de lancement de scan avec ou sans authentification	59
3.12	Interface des notifications	60
4.1	Diagramme de cas d'utilisation raffiné «Gestion des analyses SEO»	64
4.2	Diagramme de cas d'utilisation raffiné «Gestion des analyses fonctionnelles»	65
4.3	Diagramme de classe du sprint 2.1	68

4.4	Diagramme de séquence de conception du cas « Créer un scénario de test »	70
4.5	Interface unifiée de gestion des analyses SEO	71
4.6	Interface d'historique des rapports fonctionnels	72
4.7	Interface de gestion des scénarios de tests fonctionnels	73
4.8	Diagramme de cas d'utilisation du sprint 2.2	76
4.9	Diagramme de déploiement	77
4.10	Interface de gestion des rapports d'analyse	78
4.11	Interface du tableau de bord d'administrateur	78
4.12	Interface du tableau de bord testeur	79
4.13	Contenu du fichier <code>docker-compose.yml</code>	80
A.1	Structure initiale de la base de données de l'application PENTRA	
A.2	Interface de connexion (PENTRA)	
A.3	Interface d'inscription (PENTRA)	
A.4	Interface de profil utilisateur (PENTRA)	
A.5	Interface de lancement du scan de test de pénétration (PENTRA)	
A.6	Interface de progression du scan avec Wapiti (PENTRA)	
A.7	Interface de progression du scan avec OWASP ZAP (PENTRA)	
A.8	Interface du tableau de résultats du scan avec Wapiti (PENTRA)	
A.9	Rapport PDF des vulnérabilités détectées par Wapiti (PENTRA)	
A.10	Rapport des vulnérabilités détectées envoyé par e-mail (PENTRA)	
A.11	Interface de configuration des paramètres d'envoi des rapports (PENTRA)	
A.12	Interface de gestion des utilisateurs par l'administrateur (PENTRA)	
A.13	Interface de gestion des rapports de scan par l'administrateur (PENTRA)	
B.14	Diagramme de séquence d'analyse du cas d'utilisation de «S'inscrire»	
B.15	Diagramme de séquence de conception de cas «Réinitialiser un mot de passe»	
E.16	Nouveau logo de l'application <i>AutoTest</i>	
E.17	Interface de la page d'accueil	
E.18	Autres sections de la page d'accueil	
E.19	Interface de connexion	
E.20	E-mail contenant le code OTP de vérification du compte	
E.21	Interface de saisie du code OTP pour la vérification du compte	
E.22	Interface de récupération du mot de passe oublié	
E.23	Confirmation d'envoi de l'e-mail de réinitialisation	
E.24	Interface de réinitialisation du mot de passe	
E.25	Menu latéral de Testeur	

E.26 Interfaces liées à la gestion des permissions
E.27 Interface d'administration : Messages reçus via le formulaire de contact
E.28 Interface de configuration des paramètres de scan
E.29 Interface de suivi en temps réel des scans (WebSocket)
E.30 Interface de visualisation des résultats de scan (Statistiques et détails techniques)
E.31 Interface de visualisation des résultats de scan (Logs)
E.32 Interface de visualisation des résultats de scan (agrégation multi-outils)
E.33 Interface de visualisation des résultats de scan (par outil)
E.34 Interface de l'historique des scans
E.35 Téléchargement d'un rapport de scan au format HTML
E.36 Menu de notifications
E.37 Menu de choix de telechargeemnt des rapports
E.38 Interface de paramétrage des canaux de diffusion
E.39 Interface de notification via Slack
E.40 Interface de notification par e-mail
E.41 Interface de planification automatique des scans
E.42 Formulaire de lancement d'une analyse fonctionnelle
E.43 Formulaire unifiée de lancement des analyses
E.44 Téléchargement d'un rapport d'analyse complet au format HTML (onglet SEO)

Liste des tableaux

1.1	Synthèse des objectifs stratégiques et des valeurs portées par ADDINN	4
1.2	Fiche de présentation de la société d'ADDINN	4
1.3	Étapes Clés d'Évolution d'ADDINN	5
1.4	Tableau comparatif des outils d'automatisation	15
2.1	Backlog produit	27
2.2	Planification des Livraisons et des Sprints	28
2.3	Environnement matériel	30
2.4	Langages utilisés	31
2.5	Logiciels utilisés	32
2.6	Bibliothèques utilisées	33
2.7	Outils de sécurité utilisés	34
3.1	Backlog du sprint 1.1	39
3.2	Description textuelle du cas d'utilisation : Attribuer des permissions	43
3.3	Backlog du sprint 1.2	49
3.4	Description textuelle du cas d'utilisation : Choisir les outils de sécurité	54
4.1	Backlog du sprint 2.1	62
4.2	Description textuelle du cas d'utilisation : Lancer un scan fonctionnel	65
4.3	Description textuelle du cas d'utilisation : Lancer une analyse SEO	66
4.4	Backlog du sprint 2.2	74
4.5	Description textuelle du cas d'utilisation : S'inscrire	

Liste des acronymes

ADDINN	ADD VALUE BY INNOVATION
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ASGI	Asynchronous Server Gateway Interface
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CVE	Common Vulnerabilities and Exposures
DAO	Data Access Object
ENSIT	École Nationale Supérieure d'Ingénieurs de Tunis
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IT	Information Technology
JSON	JavaScript Object Notation
JWT	JSON Web Token
MVVM	Model-View-ViewModel
NGINX	Engine X (serveur web)
NTS	Nouvelles Technologies et Sécurité
OAuth2	Open Authorization 2.0
ORM	Object Relational Mapping
OTP	One-Time Password
OWASP	Open Worldwide Application Security Project
QA	Assurance Qualité
REST	REpresentational State Transfer
SEO	Search Engine Optimization
Slack	Searchable Log of All Communication and Knowledge
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer

TLS	Transport Layer Security
UML	Unified Modeling Language
WAF	Web Application Firewall
XSS	Cross-site scripting
ZAP	Zed Attack Proxy

Introduction générale

La sécurité des applications web constitue un enjeu majeur face à l'augmentation des cyberattaques, dont les vulnérabilités peuvent entraîner la perte de données sensibles. Cependant, les méthodes traditionnelles de détection de ces failles restent souvent manuelles, lentes, coûteuses et sujettes aux erreurs humaines. Dans ce contexte, les tests de pénétration jouent un rôle crucial en permettant d'identifier les failles avant qu'elles ne soient exploitées par des attaquants.

En parallèle, les tests fonctionnels vérifient que les fonctionnalités d'une application respectent les spécifications et fonctionnent correctement dans divers scénarios. Ils permettent de s'assurer que les interactions, les flux de travail, les formulaires, les boutons et les diverses actions proposées à l'utilisateur se comportent comme prévu. Bien que souvent réalisés manuellement, leur automatisation permet un gain de temps, une meilleure couverture des tests et une réduction des erreurs humaines. Cela facilite la détection rapide des dysfonctionnements, améliore la qualité globale de l'application et assure une expérience utilisateur fluide. Ces tests sont essentiels pour garantir la robustesse du produit, prévenir les bugs et renforcer la confiance des utilisateurs.

De plus, l'optimisation du référencement naturel (audit SEO) est essentielle pour évaluer et améliorer la visibilité d'un site ou d'une application web sur les moteurs de recherche. Elle permet d'identifier les points forts, les faiblesses et les axes d'amélioration du positionnement. L'automatisation de ces tests permet de détecter rapidement ces problèmes et d'optimiser la qualité technique globale.

L'objectif de ce projet est de concevoir et développer une application web pour automatiser :

- **Les tests de pénétration :** Intégrer divers outils de sécurité afin d'améliorer la détection des vulnérabilités, qu'elles soient similaires ou complémentaires, dans le cadre des tests d'intrusion. Cela inclut la gestion de la progression des scans, la comparaison et la corrélation des vulnérabilités, la fusion des résultats en un rapport final unifié, la surveillance en temps réel des vulnérabilités détectées, l'implémentation d'une authentification dynamique pour simuler des scénarios d'attaque sur des applications protégées par des pages de connexion, ainsi que la validation automatique des faux positifs.
- **Les tests fonctionnels :** Automatiser les tests pour vérifier que les fonctionnalités principales des applications web fonctionnent correctement et tester les interfaces utilisateur pour s'assurer de leur conformité avec les attentes.

- **Les tests SEO :** Identifier automatiquement les erreurs techniques pouvant affecter le référencement naturel, telles que les liens cassés, les redirections défaillantes, l'absence ou la mauvaise configuration des balises essentielles, ainsi qu'un temps de chargement excessif, qui peuvent nuire à l'indexation et à la performance globale. Fournir ensuite des recommandations pour améliorer la visibilité de l'application sur les moteurs de recherche.

Le rapport présente les étapes du projet, structuré comme suit :

- **Chapitre 1 : "Cadre général du projet"** : Contexte général du projet, présentation de la société d'accueil, analyse des solutions existantes sur le marché et la méthodologie de travail.
- **Chapitre 2 : "Analyse et planification du projet"** : Analyse des besoins fonctionnels et non fonctionnels, découpage du projet et élaboration du backlog produit, ainsi que présentation détaillée des outils et technologies utilisés lors de l'implémentation et de l'architecture de l'application.
- **Chapitre 3 : "Sprint 1 - Automatisation des tests de sécurité et amélioration des fonctionnalités de base.**
- **Chapitre 4 : "Sprint 2 - Automatisation des tests fonctionnels et SEO, génération de rapports et dockerisation."**.

Enfin, une conclusion générale résumera les résultats réalisés et proposera des perspectives d'amélioration, tout en énumérant les compétences que nous avons acquises durant ce stage.

Cadre général du projet

Introduction

Ce premier chapitre présente le contexte général du projet et l'organisme d'accueil, en analysant et en critiquant les solutions existantes ainsi que l'état actuel de la première version l'application, tout en mettant en évidence leurs limites afin de justifier la nécessité d'une nouvelle approche. Il expose également la solution proposée ainsi que la méthodologie adoptée pour sa mise en œuvre.

1. Contexte général du projet

Dans le cadre de l'obtention du diplôme national d'ingénieur en génie informatique, spécialité Nouvelles Technologies et Sécurité (NTS), à l'École Nationale Supérieure d'Ingénieurs de Tunis (ENSIT), une opportunité a été offerte pour réaliser un projet de fin d'études au sein de la société ADD VALUE BY INNOVATION (ADDINN) Tunisie, au sein de l'équipe Assurance Qualité (QA), pour une durée de quatre mois, durant l'année universitaire 2024/2025.

2. Présentation de l'organisme d'accueil

Dans cette section, nous présentons l'organisme d'accueil en abordant son histoire, ses valeurs, ainsi que ses partenariats qui contribuent à son succès et à son développement.

2.1. Aperçu général d'Addinn

Addinn est un cabinet de conseil en transformation digitale spécialisé dans l'accompagnement des entreprises pour définir leur stratégie et mettre en œuvre leurs projets technologiques grâce à un écosystème innovant et complémentaire permet de relever les défis numériques avec succès[1].



FIGURE 1.1 – Logo d'ADDINN[1]

2.2. Fondements de la Société[1]

ADDINN a pour mission de créer de la valeur à chaque étape de la réflexion à la mise en œuvre et au déploiement des projets IT. Son engagement en faveur de l'innovation se reflète dans sa devise "**ADD VALUE BY INNOVATION**", illustrant sa volonté d'apporter une réelle valeur

ajoutée à ses clients. Grâce à son approche novatrice et son expertise, l'entreprise aide à relever les défis avec agilité et efficacité, garantissant ainsi une transformation réussie et durable.

Le tableau 1.1 présente les objectifs et les valeurs qui orientent l'approche d'ADDINN :

TABLE 1.1 – Synthèse des objectifs stratégiques et des valeurs portées par ADDINN

Objectifs d'ADDINN	Valeurs d'ADDINN
<ul style="list-style-type: none"> Optimiser les flux de travail afin de les rendre plus rapides et simples. Développer des systèmes et des solutions plus efficaces. Améliorer l'expérience de clients et accroître l'avantage concurrentiel de l'entité. 	<ul style="list-style-type: none"> Innovation : Pousser plus loin les idées grâce à la recherche et au développement. Agilité : Reposer sur la flexibilité et la collaboration comme principes fondamentaux. Excellence : Viser la qualité à travers une approche structurée et optimisée.

2.3. Fiche de présentation d'ADDINN

Le tableau 1.2 fournit un aperçu détaillé de l'organisme d'accueil.

TABLE 1.2 – Fiche de présentation de la société d'ADDINN [1][1]

Nom	ADDINN
Type	Société de conseil en transformation digitale
Adresses	<ul style="list-style-type: none"> ADDINN Groupe (France) : 121 Avenue Champs-Elysées Paris 75008. ADDINN Tunis : Immeuble Etraton, Rue Khadija Ben Arfa, Centre Urbain Nord 1082. ADDINN Sousse : Rue Hedi Nouira Akouda Sousse 4022. ADDINN Tozeur : Immeuble Akouri, 2ème étage, rue 2 Mars, Tozeur. ADDINN Africa (Congo Brazzaville) : N°3 Allée des Manguiers Beach Centre-ville – Brazzaville.
Chiffres Clés	<ul style="list-style-type: none"> Plus de 11 ans d'expérience en consulting IT. Plus de 100 consultants et ingénieurs qualifiés. 90% des projets gérés en Agile/Scrum. Plus de 40 experts certifiés.
Domaines d'expertise	Conseil en stratégie, conseil en transformation digitale, développement Information Technology (IT) et solutions numériques.
Email	contact@addinn.com et sales@addinn.com
Site web	https://addinn-group.com/

2.4. Histoire de croissance d'ADDINN et présence internationale[1]

L'histoire d'ADDINN débute en 2012, avec la rencontre de deux amis ambitieux souhaitant prendre part à la révolution digitale. Plus d'une décennie d'efforts, d'innovations technologiques et de stratégie d'expansion a permis à l'entreprise de franchir ses frontières d'origine.

Aujourd’hui, ADDINN Group s’impose comme un acteur clé du marché euro-méditerranéen et africain. Avec son siège social à Paris, le groupe a étendu ses activités en créant 3 filiales en Tunisie, en Mauritanie et au Congo-Brazzaville , comme le montre la figure 1.2.

Grâce à des partenariats solides, ADDINN maintient des relations privilégiées avec ses clients au Gabon, en République Démocratique du Congo et en Belgique, consolidant ainsi son expansion à l’international.

Le tableau 1.3 résume les étapes clés de cette croissance.



FIGURE 1.2 – Présence internationale d’ADDINN[1]

TABLE 1.3 – Étapes Clés d’Évolution d’ADDINN[1]

Année	Événement
2015	Fondation d’ADDINN Group en France, spécialisé dans le conseil en IT.
2018	Création de la Digital & Software Factory, dédiée au développement de solutions numériques sur mesure.
2022	Filialisation du groupe avec la création d’ADS Foundry, BeeWay Advisory et Qualifactory, ouvrant une nouvelle phase de développement et de croissance.

2.5. Secteurs d’expertise et partenaires stratégiques

L’expertise d’ADDINN couvre divers secteurs tels que l’assurance, la banque, le secteur public et le transport, offrant des solutions sur mesure. De plus, l’entreprise collabore avec plusieurs partenaires pour renforcer son offre et sa compétitivité, comme l’illustre la figure 1.3.



FIGURE 1.3 – Partenaires stratégiques d’ADDINN [1]

3. Étude de l'existant

L'étude de l'existant permet d'identifier les points forts et les limites de l'application actuelle, ainsi que d'analyser les solutions similaires, afin d'orienter le développement d'une nouvelle version mieux adaptée aux besoins des utilisateurs.

3.1. État actuel de l'application existante

L'application **PENTRA** (Figure 1.4) constitue une première version d'un outil automatisé de tests de pénétration, développée dans le cadre d'un projet de fin d'études réalisé l'année précédente au sein de la société **ADDINN**. Elle a permis de poser les bases sur lesquelles s'appuient les évolutions futures envisagées dans le cadre de notre projet actuel.

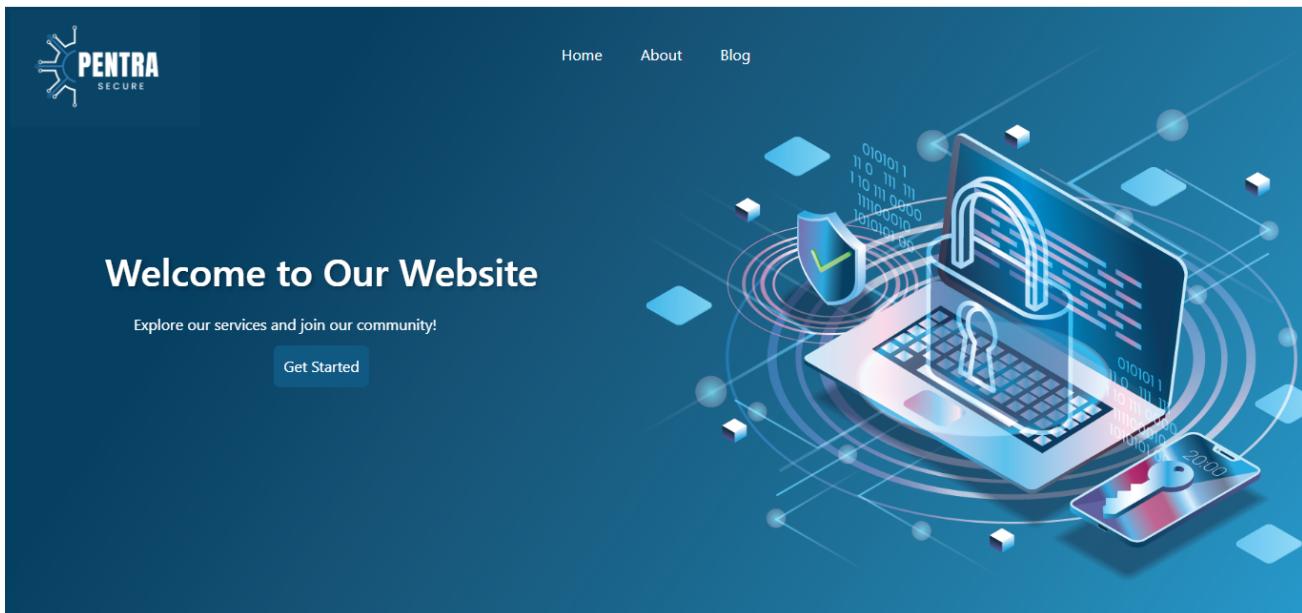


FIGURE 1.4 – Interface de la page d'accueil PENTRA

3.1.1. Analyse de la solution existante

Il s'agit d'une solution de détection des vulnérabilités web, reposant principalement sur l'intégration de deux outils de scan réputés, largement reconnus dans le domaine de la cybersécurité :

- **Open Worldwide Application Security Project (OWASP) ZAP (Zed Attack Proxy)** : Un scanner open source développé par l'OWASP pour identifier les failles de sécurité dans les applications web, telles que les injections SQL, les failles XSS, les problèmes d'authentification ou les erreurs de configuration. Il propose des analyses actives et passives, une interface graphique riche, ainsi qu'une Application Programming Interface (API) permettant son automatisation[2].
- **Wapiti** : un outil open source d'analyse de vulnérabilités web basé sur des tests d'injection, qui scanne les sites web à la recherche de failles comme les injections de commande, les XSS

ou les inclusions de fichiers. Il se distingue par sa légèreté et son approche modulaire[3].

Cette solution est développée avec Angular (version 15), FastAPI et PostgreSQL et intègre les fonctionnalités clés suivantes :

- **Gestion de l'authentification¹** : Interfaces d'inscription et de connexion permettant aux utilisateurs de créer un compte et d'accéder à la plateforme via leurs identifiants.
- **Lancement du scan²** : L'utilisateur saisit l'URL du site web à analyser dans le champ prévu. Les 2 outils de scan sont lancés et fonctionnent en mode multithread, ce qui permet de traiter plusieurs requêtes simultanément et d'accélérer le processus.
- **Tableau des résultats du scan par outil³** : Une interface affichant les résultats des scans sous forme de tableaux, avec la possibilité de télécharger les rapports détaillés au format PDF, offrant ainsi une vue d'ensemble claire des vulnérabilités détectées.
- **Paramétrage de l'envoi des rapports⁴** : permet de configurer l'envoi automatique des rapports vers Slack, Jira et par e-mail.
- **Tableaux de bord⁵** : Chaque outil dispose de sa propre interface, affichant les résultats des scans, leur progression, ainsi que des charts graphiques statiques.
- **Gestion des profils⁶** : Permet aux utilisateurs de modifier leurs informations personnelles.
- **Gestion des utilisateurs et des rapports de scan⁷** : L'administrateur peut gérer les comptes utilisateurs ainsi que consulter, organiser et superviser les rapports des scans.

3.1.2. Limites et critiques de la solution existante

L'application présente plusieurs lacunes affectant son ergonomie, ses performances et ses fonctionnalités. Ces limitations doivent être corrigées pour améliorer son efficacité globale.

✗ Problèmes liés aux processus de scan et à la génération des rapports :

Les processus actuels de scan et de génération de rapports présentent plusieurs lacunes qui nuisent à la clarté des résultats, à l'ergonomie des interfaces et à l'efficacité globale de l'analyse. Parmi les problèmes relevés, on peut citer :

- **Surcharge d'informations** : Les tableaux des résultats des scans et les rapports générés sont excessivement longs et contiennent souvent des informations inutiles.

1. Voir annexe A : Figures A.3 et A.2

2. Voir annexe A : Figure A.5

3. Voir Annexe A : figures A.8 et A.9

4. Voir annexe A : Figure A.11

5. Voir annexe A : Figures A.6 et A.7

6. Voir annexe A : Figures A.4

7. Voir annexe A : Figures A.12 et A.13

- **Affichage de résultats non optimisés et non pertinents :** La présence de vulnérabilités avec un compteur égal à zéro, ainsi que de champs excessivement détaillés ou peu utiles, nuit à la lisibilité et complique l'analyse des résultats. Ce manque d'optimisation rend l'interprétation des rapports plus difficile et chronophage.
 - **Incohérence dans la structure des tableaux :** Les colonnes varient entre les outils, compliquant la comparaison des résultats. De plus, l'absence de filtres ou d'options de recherche pour trier les données rend leur exploitation moins efficace.
 - **Problèmes de progression des scans :** Les outils affichent la progression différemment : ZAP utilise une barre de progression, tandis que Wapiti le fait sous forme textuelle indiquant les modèles scannés, ce qui crée une incohérence.
 - **Navigation peu intuitive entre les interfaces des résultats :** La navigation entre les interfaces est confuse, ce qui complique la lecture et l'interprétation des résultats, notamment sur les tableaux de bord de Wapiti et Zed Attack Proxy (ZAP) ainsi que sur les deux tableaux de résultats, rendant l'analyse des vulnérabilités plus difficile.
 - **Mauvaise gestion des scans simultanés :** Le système actuel ne gère pas correctement les lancements parallèles ou successifs de plusieurs scans par le même utilisateur ou par plusieurs utilisateurs. Cela provoque une surcharge du backend, des conflits d'accès aux ressources, voire un blocage partiel de l'application. De plus, certains outils sont déclenchés plusieurs fois en parallèle, entraînant des erreurs d'exécution et empêchant la génération des rapports.
 - **Diffusion des rapports limitée :** Les e-mails contenant les rapports⁸ sont trop basiques manquent de clarté et de modernité. Par ailleurs, l'envoi des rapports via Searchable Log of All Communication and Knowledge (Slack) et Jira ne fonctionne pas de manière fiable. L'absence d'options de téléchargement aux formats HyperText Markup Language (HTML), JavaScript Object Notation (JSON) et Comma-Separated Values (CSV) limite l'exportation et rend difficile l'analyse approfondie des résultats ou leur intégration avec d'autres outils.
- ✖ **Couverture incomplète des types d'audits :** Le système actuel se concentre principalement sur les audits de sécurité mais ne prend pas en charge l'ensemble des autres formes d'analyse essentielles. Il n'intègre ni les tests fonctionnels ni les audits Search Engine Optimization (SEO). Cette limitation réduit la portée globale des évaluations et empêche une vision complète de l'état et de la qualité du site web.

8. Voir annexe A : Figure A.10

- ✖ **Limites de la structure de la base de données :** La base repose sur deux tables (**utilisateur** et **rapport**)⁹, dans lesquelles les rapports sont stockés au format binaire sans structuration interne. Cette approche entraîne un manque de granularité empêchant l'interrogation directe d'éléments clés comme les vulnérabilités. Elle limite les capacités de recherche, de filtrage et d'agrégation des données via des requêtes SQL standards. À mesure que le volume de rapports augmente, cette structure nuit à la scalabilité et à la performance de la base en ralentissant l'accès aux données en augmentant la consommation de ressources et en rendant l'exploitation des données particulièrement difficile.
- ✖ **Problèmes des tableaux de bord statiques :** Les graphiques ne sont pas fonctionnels et restent statiques, empêchant une visualisation dynamique et claire des vulnérabilités.
- ✖ **Navigation confuse :** La coexistence d'une barre supérieure et d'une barre latérale de navigation mal structurées crée une expérience utilisateur désorientante, avec des liens incohérents, inutiles ou bloqués.
- ✖ **Ancienne version d'Angular :** L'application utilise actuellement Angular version 15, qui est dépassée et n'est plus utilisée par la société. Une migration vers une version plus récente est nécessaire.
- ✖ **Affichage non responsive :** Les interfaces ne sont pas totalement responsives, limitant une utilisation fluide sur différents types d'écrans (tablettes, mobiles...).
- ✖ **Accès non sécurisé aux interfaces sans authentification :** Bien que l'accès au backend soit protégé par authentification (erreur 403 en cas d'accès non autorisé), l'utilisateur peut accéder librement au frontend (dashboard, historique des scans, page de scan) sans authentification, ce qui crée une incohérence dans la gestion des droits d'accès.
- ✖ **Absence de gestion des mots de passe :** Il manque les pages dédiées à la réinitialisation ou à la récupération du mot de passe oublié, ce qui empêche les utilisateurs de récupérer l'accès à leur compte en cas d'oubli.

Des améliorations importantes sont indispensables afin d'optimiser l'ergonomie, les fonctionnalités et la sécurité de l'application, tout en offrant une meilleure expérience utilisateur.

3.2. Analyse et critique des applications similaires

L'analyse des applications concurrentes permet d'identifier les axes d'amélioration, afin d'optimiser notre solution et de concevoir une expérience utilisateur répondant aux attentes du marché.

9. Voir annexe A : Figure A.1

3.2.1. Outils d'automatisation des tests de pénétration[4]

Un scanner de vulnérabilités analyse un site web afin de détecter les risques liés au code source et aux liens en identifiant des menaces telles que les malwares, les injections SQL, les failles XSS. Le choix d'un outil dépend de plusieurs critères : la complexité du site, la couverture des vulnérabilités, la qualité des rapports générés et la capacité à effectuer des analyses régulières[4].

Nous avons sélectionné les outils suivants comme solutions similaires à notre application dans la partie dédiée aux tests de pénétration, afin d'en analyser les fonctionnalités et les limites.

a) **HostedScan[4]** : est un service en ligne basé sur des scanners 100% open-source, il automatise la détection des vulnérabilités sur les réseaux, serveurs et applications web. Il centralise la gestion des failles, facilite la priorisation des tâches, la génération de rapports et renforce la sécurité de leurs clients. Ce service inclut plusieurs types de scanners :

- **Des vulnérabilités réseau** : identifie les Common Vulnerabilities and Exposures (CVE) et les logiciels obsolètes.
- **Des applications web** : détecte les injections SQL, les bibliothèques JavaScript vulnérables, les scripts intersites (Cross-site scripting (XSS)) et autres menaces.
- **Des ports TCP/UDP** : repère les erreurs de configuration des pare-feu et du réseau.
- **TLS/SSL** : valide les certificats et recherche les vulnérabilités Heartbleed et Robot.

La Figure 1.5 présente l'interface du tableau de bord de l'application HostedScan, illustrant les résultats des analyses ainsi que les principales fonctionnalités proposées.

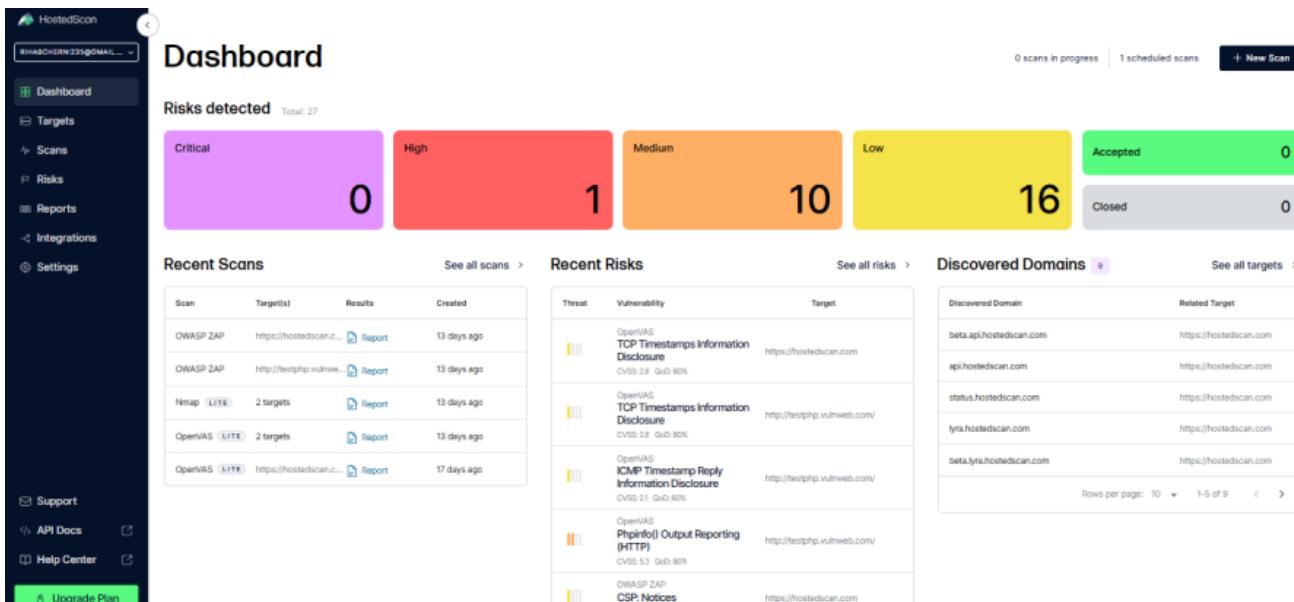


FIGURE 1.5 – Interface du l'application Hostedscan[5]

b) **Invicti[6]** : est un outil de sécurité web avancé permettant de protéger les applications, services web et API contre une large gamme de vulnérabilités, telles que les injections Structured Query Language (SQL), les failles XSS et les erreurs de configuration Secure Sockets Layer (SSL)/Transport Layer Security (TLS). Il réalise également des tests de configuration des serveurs web et se distingue par son moteur d'analyse basé sur la preuve, qui confirme automatiquement l'exploitabilité des failles détectées, réduisant ainsi les faux positifs. Son moteur d'exploration intelligent est compatible avec les technologies dynamiques modernes (JavaScript, Ajax, React, Angular), ce qui le rend efficace pour l'analyse des applications complexes.

Invicti s'intègre facilement aux pipelines DevOps et CI/CD, favorisant une automatisation continue de la sécurité. Il propose en outre des tableaux de bord interactifs, des rapports personnalisables et des fonctionnalités de gestion des vulnérabilités, facilitant la collaboration entre les développeurs et les équipes de sécurité.

La Figure 1.6 montre l'interface de l'outil Invicti, mettant en évidence les résultats d'analyse de sécurité web ainsi que les fonctionnalités de gestion des vulnérabilités.

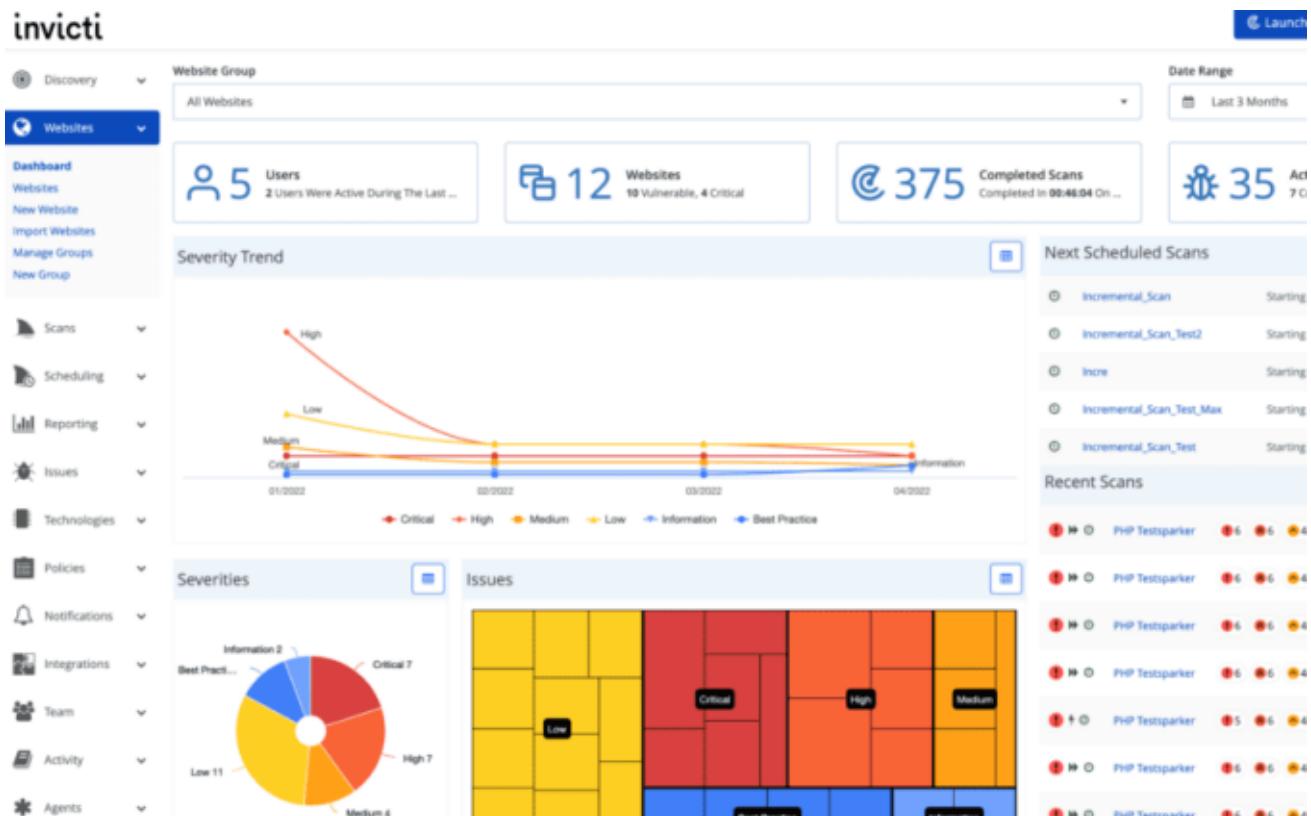


FIGURE 1.6 – Interface du l'application Invicti[6]

3.2.2. Outils d'automatisation des tests fonctionnels :

L'automatisation des tests fonctionnels est essentiel pour assurer la qualité et la fiabilité des applications. Plusieurs outils sont disponibles qui offrent des fonctionnalités variées comme :

- a) **Cypress**[7] est un outil d'automatisation de tests fonctionnels pour les applications web, apprécié pour sa rapidité, sa fiabilité et sa simplicité d'utilisation. Il permet d'automatiser des tests de composants, d'API et d'accessibilité, tout en assurant la compatibilité multi-navigateurs. Grâce à son interface de débogage en temps réel, son exécution locale rapide et sa facilité d'intégration, il constitue une solution complète pour les tests web.

La Figure 1.7 illustre l'environnement de test automatisé proposé par Cypress, affichant les scénarios exécutés, les résultats des tests et les options de débogage.

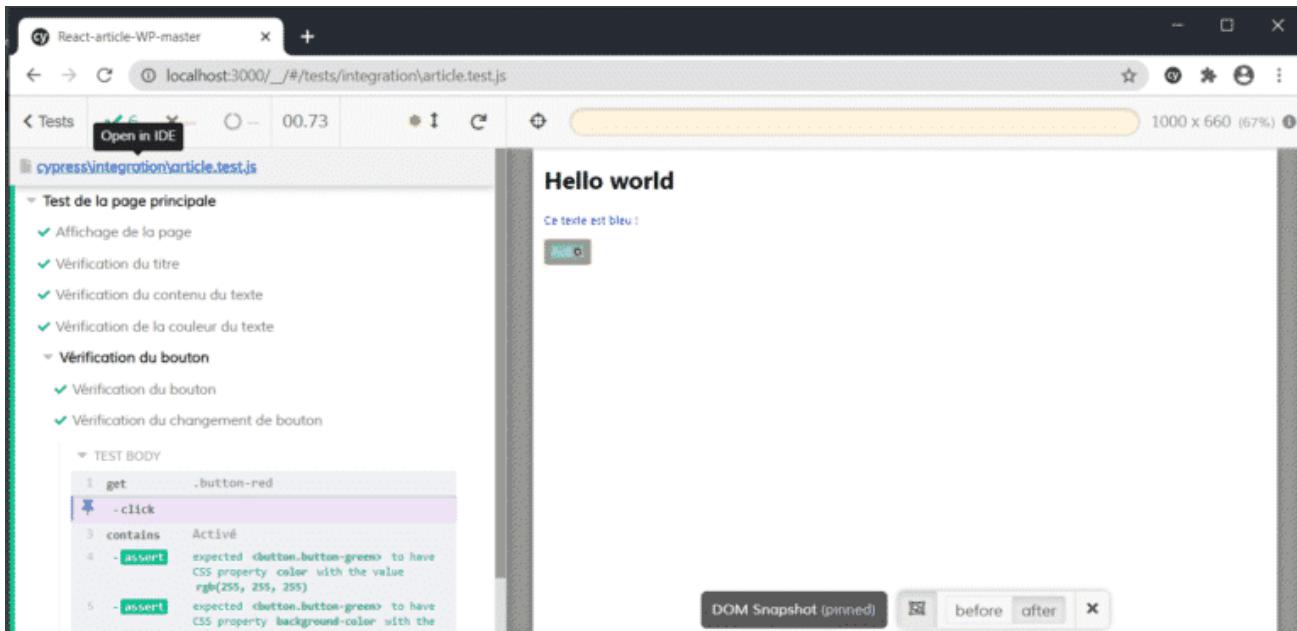


FIGURE 1.7 – Interface de démonstration de Cypress exécutant une suite de tests [8]

- b) **Katalon Studio**[9] : est un outil d'automatisation des tests pour les applications web et mobiles, basé sur Selenium. Il offre une interface conviviale permettant de créer et d'exécuter des tests sans compétences techniques avancées. Son intégration avec des outils comme Jira et Slack, ainsi que son support des tests continus et du suivi instantané des résultats, en fait une solution efficace pour assurer la qualité tout au long du développement.

La Figure 1.8 présente l'interface de Katalon Studio, illustrant l'organisation des cas de test ainsi que le suivi de leur exécution.

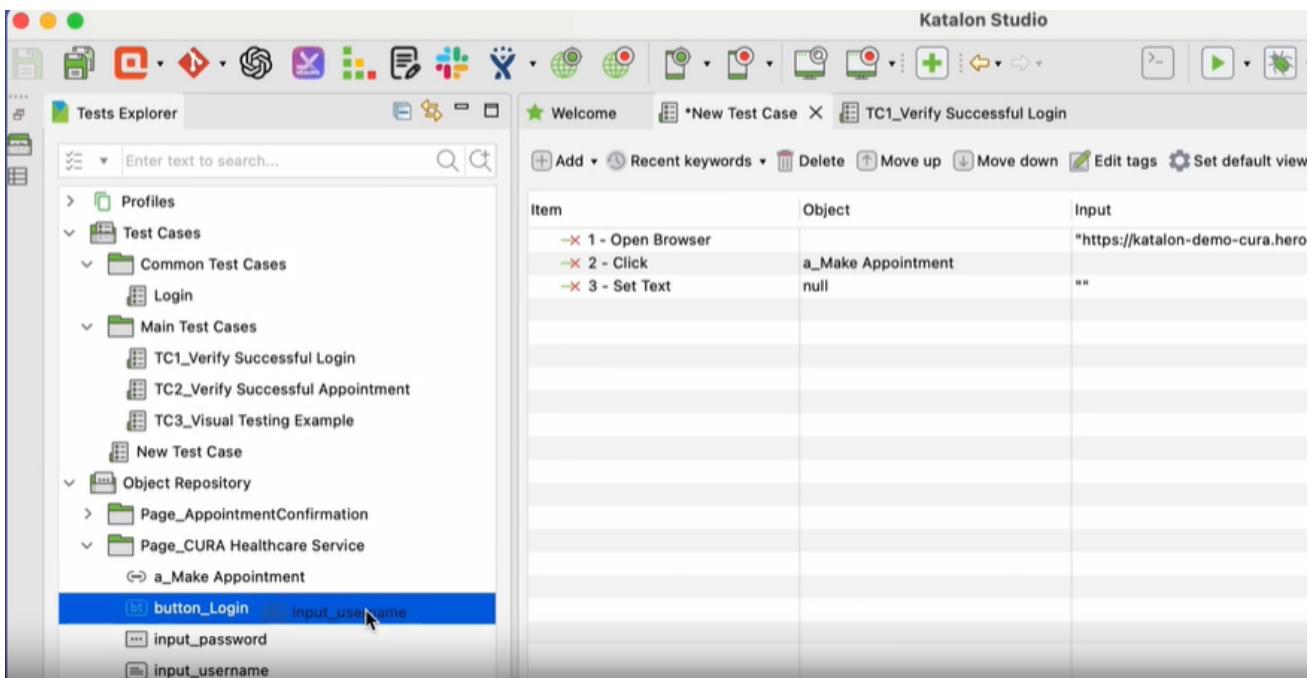


FIGURE 1.8 – Interface du l’application Katalon Studio[9]

3.2.3. Outils d’automatisation des tests SEO

Le **SEO (Search Engine Optimization)** est indispensable pour renforcer la visibilité d’un site sur les moteurs de recherche. Il permet d’identifier les erreurs, d’optimiser les performances et de vérifier le respect des bonnes pratiques. Plusieurs outils automatisent ces audits en analysant les mots-clés, le contenu, les aspects techniques, le positionnement et en générant des recommandations et des rapports. Voici une analyse des plus pertinents.

- a) **Semrush[10]** : est un outil SEO les plus populaires et complets du marché. Il propose une large gamme de fonctionnalités : l’analyse de mots-clés, l’audit technique, la veille concurrentielle, le suivi de positionnement, la gestion des backlinks, l’analyse du temps moyen passé sur le site ainsi que l’identification des pages les plus performantes. Son outil de suivi de positionnement fournit des analyses précises et évolutives, permettant aux professionnels du marketing digital de surveiller efficacement la performance de leurs mots-clés sur les moteurs de recherche. Il permet également de réaliser des audits SEO détaillés et de générer des rapports visuels pour le suivi des performances dans le temps.

La Figure 1.9 illustre le tableau de bord de SEMrush, mettant en évidence les résultats de l’audit SEO ainsi que les performances globales du site web.

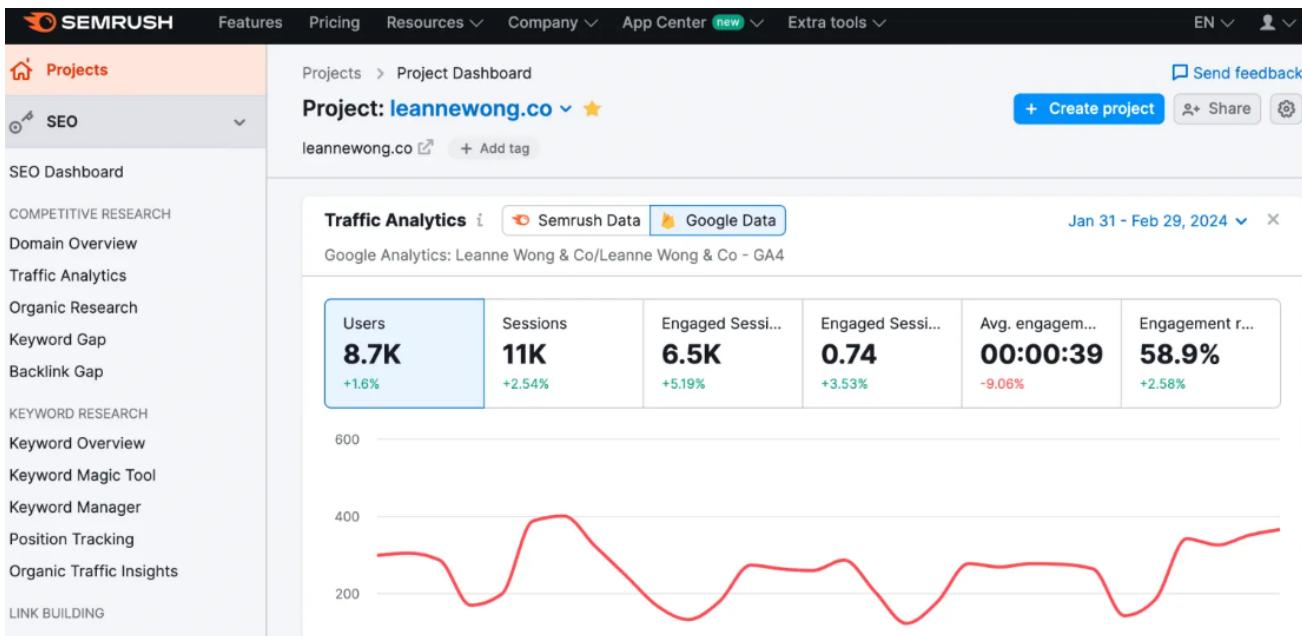


FIGURE 1.9 – Interface de l’outil Semrush[10]

- b) **Ahrefs**[10] est une plateforme SEO reconnue pour la puissance de sa base de données de backlinks et ses capacités avancées d’analyse concurrentielle. Elle permet d’ajuster efficacement sa stratégie de contenu en observant les performances des concurrents sur les moteurs de recherche. L’outil offre des fonctionnalités telles que le suivi du positionnement des mots-clés, l’identification des pages les plus génératrices de trafic, l’analyse des liens entrants et l’accès à l’historique du positionnement via des visualisations graphiques.

La Figure 1.10 présente l’interface d’Ahrefs, mettant en évidence le suivi du positionnement ainsi que les principaux indicateurs de performance SEO.

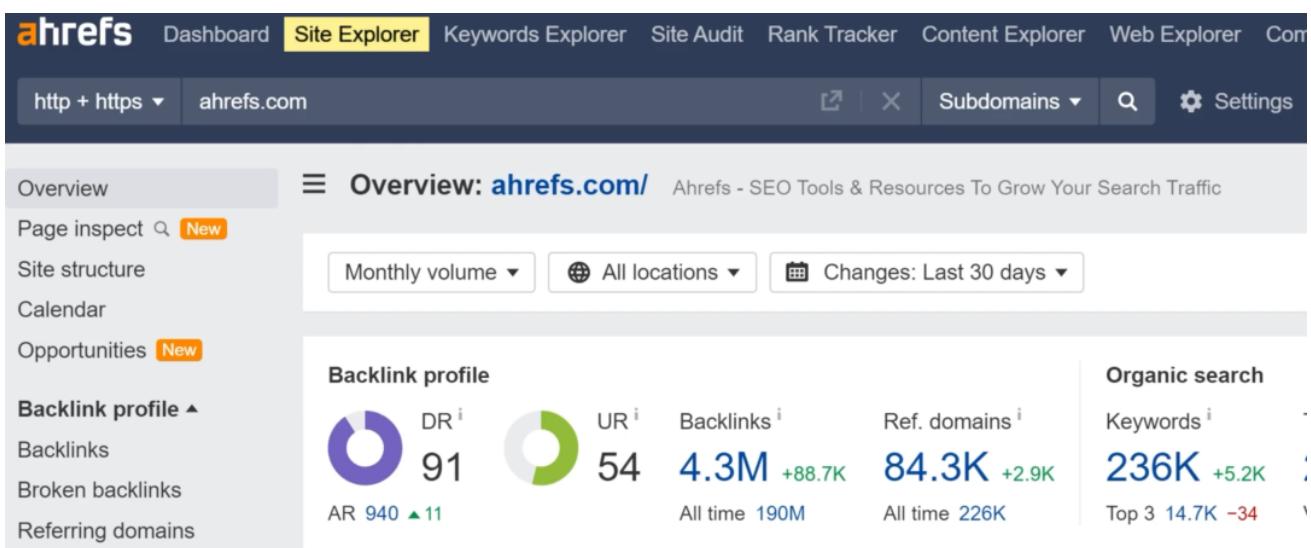


FIGURE 1.10 – Interface de l’outil Ahrefs Site Audit[10]

3.2.4. Critique de l'existant :

Le tableau 1.4 présente une comparaison détaillée des principaux outils analysés précédemment.

TABLE 1.4 – Tableau comparatif des outils d'automatisation

Outils d'automatisation des tests de pénétration [4]		
Application	HostedScan	Invicti
Score ¹⁰	4.2/5	4.5/5
Profondeur d'analyse	Réseaux, serveurs, applications web	API, sites, applications, services et serveurs web.
Fonctions principales	Scanners open-source et gestion centralisée des vulnérabilités.	Analyse basée sur la preuve et exploitation automatique des vulnérabilités.
Avantages	<ul style="list-style-type: none"> ✓ Détection et notification en temps réel des menaces ✓ Analyse approfondie. ✓ Rapports personnalisables. 	<ul style="list-style-type: none"> ✓ Excellent support client. ✓ Fournit des rapports et des analyses détaillés. ✓ Faible nombre de faux positifs.
Inconvénients	<ul style="list-style-type: none"> ✗ Interface utilisateur peu ergonomique ✗ Limité aux scanners open-source 	<ul style="list-style-type: none"> ✗ Absence de tarification initiale. ✗ Les analyses prennent parfois beaucoup de temps
Tarification	<ul style="list-style-type: none"> • Gratuit : 3 analyses par mois • Basique : 39\$ mois • Premium : 109\$ mois 	Offre des prix personnalisés en fonction de vos besoins spécifiques.
Outils d'automatisation des tests fonctionnels[11]		
Application	Cypress	Katalon Studio
Score ¹¹	4.7/5	4.4/5
Profondeur d'analyse	Tests : de bout en bout, de composants, d'API, d'accessibilité.	Tests : UI, d'API, de charge, de performance des applications web.
Avantages	<ul style="list-style-type: none"> ✓ Exécution rapide, compatible multi-navigateurs. ✓ Facilité de configuration. ✓ Débogage puissant avec l'interface graphique. ✓ Tests fiables et résultats cohérents. 	<ul style="list-style-type: none"> ✓ Facilité d'utilisation. ✓ Documentation complète et support client efficace. ✓ Intégration facile avec les pipelines CI/CD. ✓ Compatibilité multi-navigateurs et tests multiplateformes.

10. Le score Geekflare est attribué par une équipe éditoriale selon plusieurs critères techniques et fonctionnels (source : Geekflare).

11. Le score Capterra reflète la moyenne des avis des utilisateurs sur différents critères (ergonomie, fonctionnalités, support, rapport qualité/prix)(source : ClickUp).

Inconvénients	<ul style="list-style-type: none"> ✗ Certains utilisateurs ont constaté des bugs. ✗ Manque d'assistance intégrée pour les tests d'applications mobiles. 	<ul style="list-style-type: none"> ✗ Décalage lors de l'utilisation sur certains ordinateurs portables. ✗ Des fonctionnalités manquantes comme l'exécution parallèle.
Tarification	<ul style="list-style-type: none"> • Gratuit : Plan communautaire limité • Équipes : 75\$/mois. • Business : 300\$/mois • Entreprise : Prix personnalisé. 	<ul style="list-style-type: none"> • Gratuit : 0\$ (Plan de base avec certaines limitations). • Premium : 218\$/utilisateur/mois • Entreprise : Prix personnalisé.

Outils d'audit SEO automatisé[10]

Application	Semrush	Ahrefs Site Audit
Score ¹²	4.6/5	4.7/5
Interface utilisateur	Simple, intuitive et rapide à utiliser	Tableaux de bord graphiques et interactifs
Fonctions principales	Audit HTML, SEO on-page, détection de liens cassés, analyse de la vitesse, compatibilité mobile, avec export des résultats en PDF	Visualisation graphique, suivi des erreurs, historique des audits, recommandations, avec export des résultats sous forme de graphiques et de rapports web
Avantages	<ul style="list-style-type: none"> ✓ Recherche et suivi des mots-clés. ✓ Analyse concurrentielle. ✓ Suivi du positionnement en temps réel. ✓ Audit technique SEO complet. ✓ Optimisation des backlinks. ✓ Interface personnalisable. 	<ul style="list-style-type: none"> ✓ Base de données de backlinks ultra-complète. ✓ Analyse approfondie des sites concurrents pour identifier leurs stratégies SEO. ✓ Suivi précis des performances et suggestions de contenu pertinentes.
Inconvénients	<ul style="list-style-type: none"> ✗ Tarification élevée pour les petites entreprises. ✗ Une interface complexe pour les débutants. 	<ul style="list-style-type: none"> ✗ Moins complet que Semrush pour l'optimisation on-site. ✗ Peu d'options pour générer des rapports personnalisés.

Chaque outil présente des atouts spécifiques selon le type de tests à automatiser, qu'il s'agisse de sécurité, de fonctionnalité ou d'audit SEO. Le choix de l'outil dépendra des besoins techniques, du budget disponible et du niveau d'intégration souhaité dans le processus de développement.

12. Le score Capterra correspond à la moyenne des avis utilisateurs, évaluant des critères tels que l'ergonomie, les fonctionnalités, le support et le rapport qualité/prix (sources : Ahrefs et SEMrush).

4. Problématique

Les tests fonctionnels, les tests de sécurité ainsi que les tests SEO représentent trois aspects essentiels pour garantir la fiabilité, la performance et la visibilité des applications. Toutefois, leur mise en œuvre reste principalement manuelle, ce qui la rend chronophage et sujette à des erreurs humaines. La question qui se pose est de concevoir une solution capable d'automatiser ces types de tests tout en assurant une couverture complète et une analyse fiable ?

5. Solution proposée

Afin de surmonter les nombreuses limites identifiées dans l'application existante, la solution proposée vise à développer une plateforme complète "**AutoTest**" pour l'automatisation des tests de sécurité, des audits SEO et des tests fonctionnels avec une analyse fiable et complète des résultats. Elle vise à renforcer la visibilité des applications tout en garantissant leur bon fonctionnement. Cette nouvelle solution repose sur une approche évolutive avec les objectifs suivants :

- **Intégration optimisée des outils de pénétration** : Utilisation de ZAP[2], Wapiti[3], Nuclei[12], Nikto[13], SQLMap[14], XSSStrike[15] et autres, chacun spécialisé dans une famille de vulnérabilités, pour détecter les failles de sécurité et couvrir un périmètre d'analyse plus large et plus fiable.
- **Mécanisme de comparaison, de consolidation des résultats et de validation automatique des faux positifs** : regroupement des vulnérabilités similaires détectées par différents outils, avec une pondération selon leur niveau de risque, afin d'éliminer les doublons, de clarifier les rapports, d'améliorer leur précision, de filtrer les résultats pertinents, de réduire les erreurs de détection et d'optimiser le temps d'analyse.
- **Scan d'authentification automatisé** : Détection des formulaires de connexion après identification des champs login/password, cookies et tokens d'authentification pour permettre les scans dans les zones protégées en reproduisant des scénarios d'attaque.
- **Automatisation des tests fonctionnels** : en exécutant des scénarios de tests simulant le comportement utilisateur (remplissage de formulaires, clics, redirections), afin de vérifier que les fonctionnalités clés restent opérationnelles et conformes aux exigences des utilisateurs.
- **Automatisation des tests SEO** : pour analyser les techniques du référencement (temps de chargement, structure HTML, balises, liens cassés).
- **Système de reporting** : Génération de rapports synthétiques et clairs avec des filtres (par outil, par niveau de gravité, par catégorie), visualisation graphique interactive, export en PDF, HTML, JSON ou CSV, envoi par e-mail, Slack ou Jira.

- **Refonte de la base de données** : adoption d'une structure relationnelle optimisée avec des tables normalisées permettant un stockage détaillé des rapports, une gestion granulaire des vulnérabilités ainsi qu'une meilleure exploitation des données via des requêtes avancées.
- **Gestion intelligente des scans concurrents** : Mise en place d'un système de file d'attente et de verrouillage par utilisateur afin d'empêcher le lancement simultané ou successif non contrôlé de plusieurs scans. Une logique de planification permet d'exécuter les scans de manière ordonnée, tout en assurant un contrôle d'accès aux ressources partagées. Des vérifications en temps réel préviennent les doublons, les surcharges ou les blocages. Des statuts d'exécution offrent à l'utilisateur la possibilité de suivre l'avancement d'un scan ou de l'annuler si nécessaire.
- **Interface responsive et modernisée** : Refonte complète du frontend en Angular dernière version 18, accès contrôlé par authentification JWT[16] et rôle utilisateur et responsive design pour mobile et tablette.
- **Sécurisation complète des accès** : Mise en place d'un contrôle d'accès uniforme sur l'ensemble des routes, avec redirections automatiques, gestion des sessions et mécanisme de réinitialisation de mot de passe.

Cette solution permettra de corriger en profondeur les failles de l'application actuelle en rendant les processus de tests efficaces et compréhensibles. Elle contribuera aussi à fournir une plateforme unifiée adaptée aux besoins réels des développeurs, des testeurs et des responsables sécurité.

6. Choix méthodologique de travail

Les méthodes agiles proposent une approche flexible et itérative du développement, centrée sur la réduction des risques, l'adaptabilité et la satisfaction du client. Scrum est la méthode agile la plus couramment adoptée[17].

Le choix méthodologique retenu pour notre application, défini dès le début du stage, résulte d'une réflexion collective. Il a été guidé par les besoins du projet et la stratégie de l'entreprise, notamment en matière d'approche, de langage de modélisation et de processus de développement.

Fonctionnement général de Scrum[18]

Scrum repose sur des itérations successives "sprints", comme l'illustre la figure 1.11, permettant une livraison progressive du produit et visant à apporter de la valeur au client en favorisant la transparence, la collaboration et l'amélioration continue.

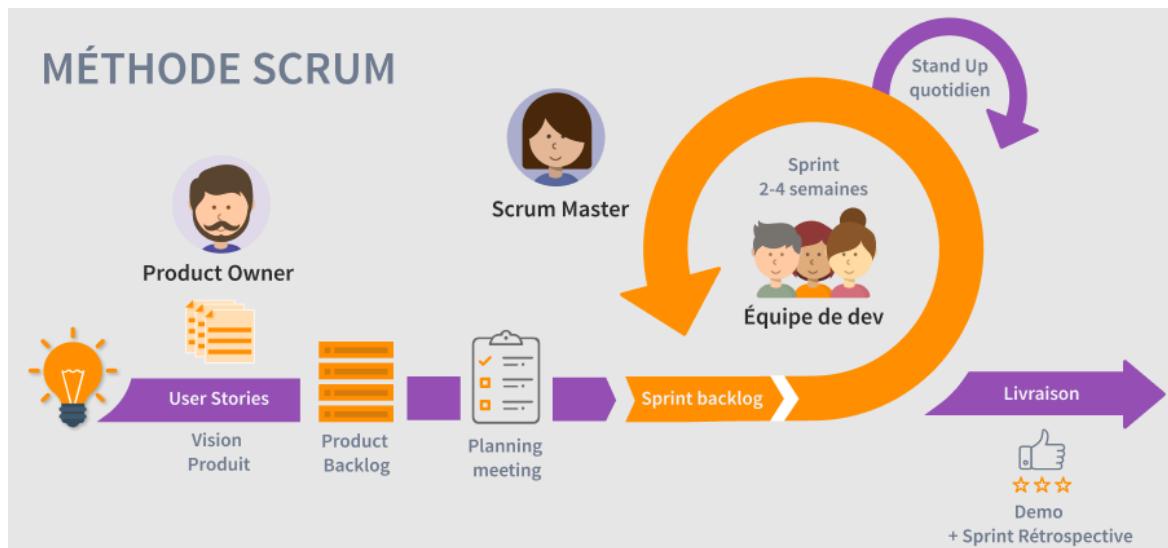


FIGURE 1.11 – Cadre méthodologique et mode de fonctionnement SCRUM [19]

- L'équipe Scrum se compose de trois rôles clés :

- **Product Owner** : Gère le backlog et définit les priorités selon les besoins client.
- **Scrum Master** : Garantit le respect de la méthodologie et facilite le travail de l'équipe.
- **Équipe de développement** : Chargée de la réalisation des incrémentums du produit.

Les événements Scrum rythment le projet et favorisent son amélioration continue. Parmi eux, le **Daily Scrum** permet une synchronisation quotidienne de l'équipe, la **Sprint Review** offre un moment de présentation de l'incrémentum aux parties prenantes pour ajuster les priorités et la **Sprint Retrospective** vise à identifier les axes d'amélioration pour optimiser les sprints suivants.

Les artefacts Scrum, quant à eux, structurent le travail et assurent une transparence constante sur l'avancement du projet. Ils incluent le **backlog produit** (liste évolutive des besoins), le **backlog de sprint** (tâches planifiées pour un sprint), l'**incrémentum** (version fonctionnelle livrée) et le **burndown chart** (graphique de suivi de la progression).

En résumé, l'adoption de Scrum permet d'améliorer la qualité logicielle et la satisfaction client grâce à des livraisons rapides et des ajustements fréquents. Elle contribue également à une meilleure dynamique d'équipe, en favorisant la motivation et l'autonomie des développeurs.

Conclusion

Ce chapitre a introduit le projet en présentant son contexte, les caractéristiques de l'organisme d'accueil, ainsi que les lacunes de l'outil existant et des solutions similaires, soulignant ainsi la nécessité d'adopter une nouvelle approche. Il a exposé la solution proposée et la méthodologie de travail. Le chapitre suivant portera sur l'analyse préliminaire de cette nouvelle solution.

Analyse et planification du projet

Introduction

Dans ce chapitre, nous allons présenter la première phase de la méthodologie Scrum qui est la phase de planification. Il comprend l'identification des acteurs, des besoins fonctionnels et non fonctionnels, la création du diagramme de cas d'utilisation, la présentation de l'équipe de travail, l'élaboration du backlog produit, ainsi que la planification des sprints. Nous y décrivons également l'environnement logiciel et matériel adopté, ainsi que l'architecture générale retenue.

1. Spécification et analyse des besoins

La capture des besoins permet au client d'exprimer ses attentes. Dans cette section, nous présentons les acteurs, les besoins fonctionnels et non fonctionnels que notre solution doit respecter.

1.1. Identification des acteurs

Un acteur est une entité externe, qu'il s'agisse d'une personne, d'un processus ou d'un objet, qui interagit avec le système. Il exécute des tâches spécifiques et joue un rôle essentiel dans la définition des besoins du système[20].

Dans ce projet, nous identifions 3 acteurs :

- **Visiteur** : Il s'agit d'un utilisateur ordinaire qui souhaite explorer et consulter les informations disponibles sur le site. Il peut également devenir un client potentiel en s'inscrivant sur la plateforme afin d'accéder à ses fonctionnalités.
- **Testeur** : Son travail consiste à effectuer des tests afin d'identifier l'éventuelles vulnérabilités. Il analyse ensuite les résultats obtenus, consulte les rapports générés et suit l'évolution des correctifs mis en place.
- **Administrateur** : est l'acteur responsable de la gestion globale du système. Ses missions incluent la gestion de l'application et des utilisateurs, la configuration des paramètres des scans ainsi que l'assurance du bon fonctionnement et de la sécurité du système.

1.2. Besoins fonctionnels

Les besoins fonctionnels définissent les actions attendues du système et les services à implémenter pour répondre aux attentes des utilisateurs[21].

Dans cette section, nous présentons les fonctionnalités que notre application doit assurer.

1. Consulter la page d'accueil : Accessible sans création de compte.

- Parcourir la page d'accueil et consulter les sections publiques telles que la FAQ, le guide utilisateur, les services proposés, la présentation de l'équipe, la tarification...
- Contacter l'administrateur via un formulaire de contact.

2. Authentification et gestion du profil utilisateur : Permet de gérer l'accès sécurisé à la plateforme.

- S'inscrire : permettre à un utilisateur de créer un compte.
- Vérifier l'adresse e-mail après l'inscription : valider l'identité de l'utilisateur.
- S'authentifier : accéder à la plateforme via des identifiants valides.
- Se déconnecter : mettre fin à une session utilisateur.
- Réinitialiser le mot de passe en cas d'oubli : retrouver l'accès à son compte.
- Gérer le profil utilisateur : modifier les informations personnelles.

3. Gestion des scans de tests de sécurité : Assurer la détection automatique des vulnérabilités dans l'application web cible.

- Permettre la configuration des paramètres de scan selon les besoins spécifiques.
- Sélectionner les outils de sécurité à utiliser pour l'analyse.
- Lancer un scan de sécurité.
- Lancer un scan de sécurité avec authentification dynamique (cookies, jetons ou identifiants) afin de tester les parties protégées de l'application.
- Annuler un scan en cours : stopper un test lancé par erreur ou jugé inutile.
- Relancer un scan : exécuter à nouveau une analyse avec les mêmes paramètres.
- Planifier des scans automatiques pour une surveillance continue et garantir une sécurité régulière.
- Suivre la progression du scan en temps réel via WebSocket pour une visibilité immédiate de l'exécution.
- Visualiser les résultats du scan de sécurité : consulter les vulnérabilités détectées.

- Automatiser la communication des résultats critiques en les intégrant à Jira, en les notifiant sur Slack et en les diffusant par e-mail aux parties prenantes.
- Gérer l'historique des rapports des scans de sécurité : conserver une trace des analyses précédentes.
- Téléchargement des rapports aux formats HTML, JSON, PDF, CSV et ZIP.

4. Gestion des scans fonctionnels : Vérifier le bon fonctionnement des différentes fonctionnalités métier dans l'application web cible.

- Gérer les flux de travail (scénarios de test) :
 - Créer un scénario de test : définir un parcours utilisateur à valider.
 - Modifier un scénario de test existant : ajuster les tests en fonction des évolutions.
 - Exécuter un scénario de test : déclencher le test fonctionnel.
 - Récupérer et afficher les scénarios de test, y compris leurs configurations et états d'exécution : assurer le suivi des tests.
- Gérer les cas de test associés à chaque scénario de test :
 - Créer un cas de test : définir une étape précise à valider.
 - Modifier un cas de test existant.
 - Exécuter un cas de test.
 - Lister les cas de test : afficher tous les cas disponibles.
 - Récupérer et afficher les cas de test : accéder aux détails de chaque test.
- Gérer les différentes étapes associées à chaque cas de test :
 - Créer une étape : définir une action précise à exécuter dans un cas de test.
 - Modifier une étape : adapter une étape existante selon les nouvelles exigences.
 - Supprimer une étape : retirer une action devenue obsolète ou incorrecte.
 - Lister les étapes d'un cas de test : afficher l'enchaînement complet des actions définies.
 - Récupérer et afficher les détails d'une étape.
- Lancer un scan de test fonctionnel.
- Planifier des scans fonctionnels automatiques réguliers.
- Relancer un scan fonctionnel : réexécuter un test fonctionnel existant.
- Suivre en temps réel l'exécution du scan fonctionnel via WebSocket pour une meilleure visibilité du processus.
- Visualiser les résultats : consulter les dysfonctionnements ou anomalies détectés.

- Centraliser le suivi des anomalies fonctionnelles en automatisant leur création dans Jira, en notifiant les équipes via Slack et en envoyant les résultats par e-mail aux parties prenantes.
- Gérer l'historique des rapports des scans fonctionnels : conserver une trace des scénarios et cas de test exécutés.
- Téléchargement des rapports des scans fonctionnels aux différents formats.

5. **Gestion des scans SEO** : Évaluer la qualité de référencement et les technologies utilisées dans l'application web cible.

- Lancer une analyse SEO complète (balises HTML, performance, accessibilité...).
- Relancer une analyse SEO : effectuer à nouveau un audit avec les mêmes critères.
- Identifier les technologies et frameworks utilisés.
- Extraire les mots-clés pertinents et analyser le contenu textuel.
- Générer une capture d'écran de la page cible.
- Calculer un score SEO global avec rapport détaillé.
- Suivre en temps réel l'exécution du scan SEO via WebSocket pour une meilleure visibilité du processus.
- Visualiser les résultats : consulter les points forts et axes d'amélioration du référencement.
- Intégrer avec Jira : création automatique de tickets pour les problèmes SEO majeurs et suivi des actions correctives.
- Notifier via Slack : transmission rapide des résultats aux équipes concernées.
- Envoyer automatiquement les rapports SEO par e-mail aux parties prenantes.
- Gérer l'historique des rapports des analyses SEO : conserver une trace des audits réalisés.
- Téléchargement des rapports des analyses SEO aux différents formats.

6. **Gestion complète des analyses (fonctionnelles, de sécurité et SEO)** : Permet l'exécution simultanée des trois types d'analyses afin d'optimiser le temps de test, de centraliser les résultats dans un seul rapport consolidé et de faciliter la corrélation entre vulnérabilités, dysfonctionnements fonctionnels et recommandations SEO.

7. **Consultation des statistiques des scans via les tableaux de bord interactifs** : Offrir une visualisation graphique personnalisée des résultats pour chaque type d'utilisateur.

- **Tableau de bord administrateur** : Présenter sous forme de graphiques dynamiques l'ensemble des activités de scans (fonctionnels, sécurité, SEO) classées par période, type d'analyse ou niveau de gravité, ainsi que l'état global du système.

- **Tableau de bord testeur :** Afficher sous forme des graphes les résultats des scans personnels, la répartition des vulnérabilités détectées, la progression des tests et la couverture fonctionnelle.

8. Notifications en temps réel : Améliorer la réactivité face aux alertes critiques.

- Être notifié en temps réel des résultats pendant l'exécution des scans pour suivre en direct l'analyse.
- Envoyer des alertes à l'utilisateur pour les vulnérabilités critiques pour agir rapidement en cas de danger.

9. Configuration des paramètres d'envoi des rapports(Email, Jira et Slack) : Permet de définir les canaux de diffusion automatisée des résultats d'analyse.

- Définir les identifiants et jetons d'accès pour Slack : assurer l'envoi automatisé des rapports.
- Saisir les adresses e-mail des destinataires pour la diffusion des rapports.
- Configurer l'URL, l'identifiant du projet et les clés API pour l'intégration avec Jira.
- Sélectionner les types de rapports à envoyer (sécurité, fonctionnel, SEO).
- Choisir les formats de rapports à transmettre (HTML, PDF, JSON ...).
- Activer ou désactiver chaque canal de notification selon les préférences.

10. Gestion des utilisateurs : Assurer une administration des comptes.

- Rechercher un utilisateur selon plusieurs filtres : retrouver rapidement un profil.
- Exporter la liste des utilisateurs : conserver une copie administrative.
- Consulter la liste des utilisateurs : avoir une vue globale des membres de la plateforme.
- Gérer les permissions des utilisateurs.
- Supprimer un utilisateur.

11. Gestion des rapports pour l'administrateur : Permettre à l'administrateur de superviser tous les rapports générés par les différents types de scans.

- Accéder à l'ensemble des rapports de sécurité, fonctionnels et SEO générés par tous les utilisateurs.
- Filtrer les rapports par type d'analyse, période ou utilisateur.
- Rechercher un rapport spécifique.
- Télécharger les rapports consolidés ou individuels dans différents formats.
- Supprimer des rapports obsolètes pour maintenir une base de données claire et à jour.

1.3. Besoins non fonctionnels

Les besoins non fonctionnels définissent comment le système doit fonctionner incluant les contraintes d'implémentation, d'environnement, de performance, de dépendances, de maintenance, d'extensibilité, de sécurité, de fiabilité et d'ergonomie. Une analyse approfondie de ces besoins est essentielle pour garantir la qualité globale et l'efficacité d'un produit logiciel[22].

Pour optimiser l'expérience utilisateur dans ce projet, il est important de respecter les exigences de qualité suivantes :

- **Sécurité** : Le système doit garantir que les données des utilisateurs ainsi que les résultats des scans soient stockés de manière sécurisée. Les rapports doivent être protégés et accessibles uniquement aux utilisateurs autorisés.
- **Scalabilité** : Le système doit être capable de gérer une charge importante de tests effectués simultanément, sans dégradation de la performance. Il doit permettre un traitement parallèle efficace, où chaque test s'exécute de manière indépendante, garantissant que plusieurs utilisateurs puissent fonctionner simultanément sans interférer les uns avec les autres.
- **Fiabilité** : Le système doit être conçu pour être tolérant aux pannes et permettre une reprise automatique des scans en cas de défaillance. De plus, il doit garantir l'envoi immédiat d'alertes en temps réel lorsqu'un problème est détecté, comme des vulnérabilités critiques durant un scan, afin de permettre une réaction rapide et appropriée.
- **Disponibilité** : Le système doit être disponible en permanence, garantissant un fonctionnement sans interruption, 24h/24 et 7j/7.
- **UX/UI intuitive** : L'interface utilisateur doit être claire et ergonomique, permettant une navigation fluide et intuitive pour tous les utilisateurs. L'objectif est d'assurer une interaction simple et rapide, sans courbe d'apprentissage complexe, avec des retours immédiats lors des actions et une navigation optimisée.
- **Compatibilité et adaptabilité** : L'interface doit être responsive, s'adaptant de manière optimale à toutes les tailles et résolutions d'écrans, tout en offrant une expérience cohérente sur tous les navigateurs et plateformes, quel que soit l'appareil utilisé.

2. Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation offre une vue d'ensemble du système en représentant les interactions entre les acteurs et les cas d'utilisation correspondant aux principales fonctionnalités attendues. Il permet de définir les exigences fonctionnelles, de clarifier les objectifs du projet, de délimiter le périmètre du système et d'identifier les besoins des utilisateurs [23].

Dans ce diagramme, présenté à la figure 2.1, sont regroupés tous les cas d'utilisation de base afin d'offrir une perspective globale du fonctionnement de notre système.

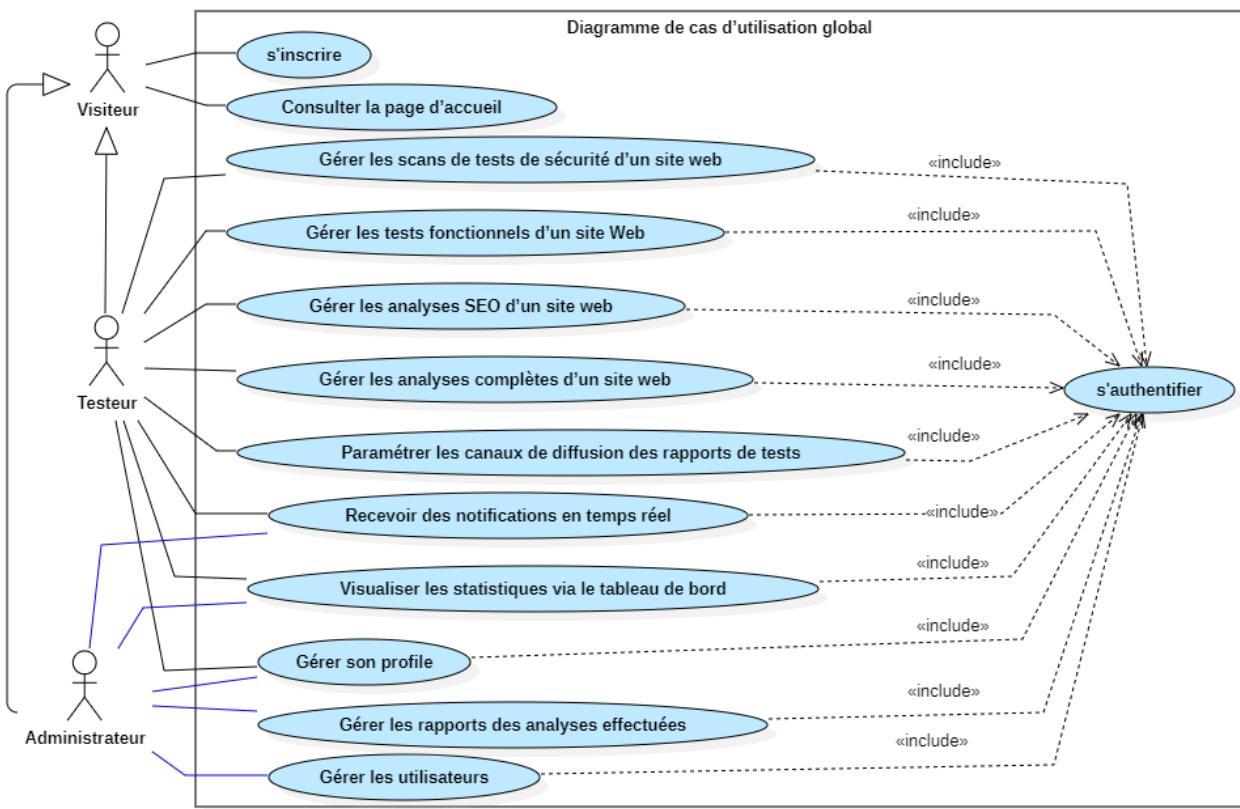


FIGURE 2.1 – Diagramme de cas d'utilisation global

3. Pilotage du projet avec Scrum

Cette section décrit le pilotage du projet selon la méthodologie Scrum, en présentant les outils utilisés, la composition de l'équipe et ses rôles, le backlog du produit et le découpage du projet.

3.1. Présentation de l'équipe de travail

La réussite du projet repose sur une bonne organisation de l'équipe et une répartition claire des rôles, assurées par la méthodologie Scrum. Dans le cadre de ce projet, les trois rôles clés de Scrum sont attribués aux membres de l'équipe de la manière suivante :

- **Scrum Master :** Représenté par notre encadrant professionnel, Monsieur **Omar NACHMI**, qui a guidé l'équipe dans l'adoption de Scrum, assurant une organisation efficace et une amélioration continue des processus de travail.
- **Product Owner :** Représenté également par Monsieur **Omar NACHMI**, qui a proposé et dirigé ce travail, en fournissant des explications approfondies sur les diverses exigences et fonctionnalités requises par le système.

- **Équipe de développement :** Le projet a été entièrement réalisé par moi-même, **Rihab Cherni**, en assurant toutes les étapes du développement.

3.2. Outils SCRUM utilisés

Pour le suivi de notre projet, nous avons adopté des outils collaboratifs : **OpenProject** pour la gestion des tâches et le suivi quotidien de notre progression, **Microsoft Teams** pour la communication entre les membres de l'équipe, l'organisation des réunions quotidiennes (daily meetings), des réunions RH et la coordination des sprints et **GitLab** pour le travail collaboratif sur le code et la gestion de version.

3.3. Backlog du produit

Le backlog produit représente une liste des fonctionnalités nécessaires au développement du produit. Géré par le Product Owner, il permet d'assurer l'alignement entre les besoins des utilisateurs et les objectifs du projet. Il constitue un outil de référence pour l'équipe de développement, en collaboration avec le Scrum Master et les parties prenantes [24].

Les epics structurent le backlog à un niveau stratégique en regroupant des fonctionnalités clés du produit. Plus génériques et moins détaillés que les user stories, ils doivent être découpés progressivement pour être développés. Ils facilitent la planification, la priorisation, la communication avec les parties prenantes et la gestion de la complexité tout au long du projet[25].

Le tableau 2.1 présente le backlog produit de notre projet, en précisant pour chaque épique sa priorité, le niveau de risque, ainsi que son effort estimé en jours.

TABLE 2.1 – Backlog produit

ID	Épics	Priorité	Risques	Estimation(j)
1	Initialisation du projet	Élevée	Basse	15
2	Consultation de la page d'accueil	Basse	Basse	5
3	Authentification et gestion du profil	Moyenne	Basse	4
4	Gestion des scans de tests de sécurité d'un site web	Élevée	Élevée	20
5	Gestion des tests fonctionnels d'un site web	Élevée	Élevée	15
6	Gestion des analyses SEO d'un site web	Élevée	Moyenne	10
7	Gestion des analyses complètes	Élevée	Élevée	6
8	Visualisation des statistiques via le tableau de bord	Élevée	Élevée	8
9	Notifications en temps réel	Élevée	Basse	2

10	Paramétrage des canaux de diffusion des rapports de tests	Moyenne	Moyenne	2
11	Gestion des utilisateurs	Moyenne	Basse	2
12	Gestion des rapports des analyses effectuées	Moyenne	Basse	3
13	Déploiement de l'application	Moyenne	Moyenne	5
TOTAL			97(jours)	

3.4. Planification des sprints

Pour la planification, nous avons choisi de répartir le travail en deux livraisons (releases), chacune étant divisée en deux sprints, comme indiqué dans le tableau 2.2.

TABLE 2.2 – Planification des Livraisons et des Sprints

Livraison 1 (Total : 50 jours)	Livraison 2 (Total : 47 jours)
<p>Sprint 1.1 (26 jours) :</p> <ul style="list-style-type: none"> – Initialisation du projet – Consultation de la page d'accueil – Authentification – Gestion des utilisateurs <p>Sprint 1.2 (24 jours) :</p> <ul style="list-style-type: none"> – Gestion des scans de tests de sécurité – Paramétrage des canaux de diffusion des rapports de tests – Notifications en temps réel 	<p>Sprint 2.1 (25 jours) :</p> <ul style="list-style-type: none"> – Gestion des scans fonctionnels – Gestion des scans SEO <p>Sprint 2.2 (22 jours) :</p> <ul style="list-style-type: none"> – Gestion des scans complètes (fonctionnels, sécurité et SEO) – Visualisation du tableau de bord – Gestion des rapports des analyses effectuées – Déploiement de l'application

Cette répartition a permis d'ajuster le projet au fil des sprints tout en intégrant les fonctionnalités essentielles dès les premières étapes du développement.

4. Planning prévisionnel du projet

Pour une gestion optimale du projet, nous avons utilisé un diagramme de Gantt, permettant de répartir les tâches de manière structurée et de visualiser leur durée ainsi que leur enchaînement. Comme illustré dans la figure 2.2, chaque tâche est associée à un intervalle de temps précis, facilitant le suivi de l'avancement et la gestion des priorités. Cet outil aide à identifier les étapes clés et à ajuster les délais si nécessaire, garantissant ainsi une coordination efficace et un déroulement fluide du notre projet.



FIGURE 2.2 – Diagramme de Gantt

5. Rétrospective Agile

Dans cette section, nous allons présenter les différents artefacts Scrum que nous avons utilisés pour accomplir notre projet et faciliter la gestion des tâches.

- Scrum board

Le Scrum board est un outil visuel qui organise les tâches d'un sprint en colonnes (à faire, en cours, terminées) pour faciliter le suivi de l'avancement[26]. Nous avons utilisé **OpenProject** pour gérer nos tâches réparties en quatre colonnes (à faire, en cours, à tester, terminées), comme illustré à la figure 2.3 pour le Sprint 1.

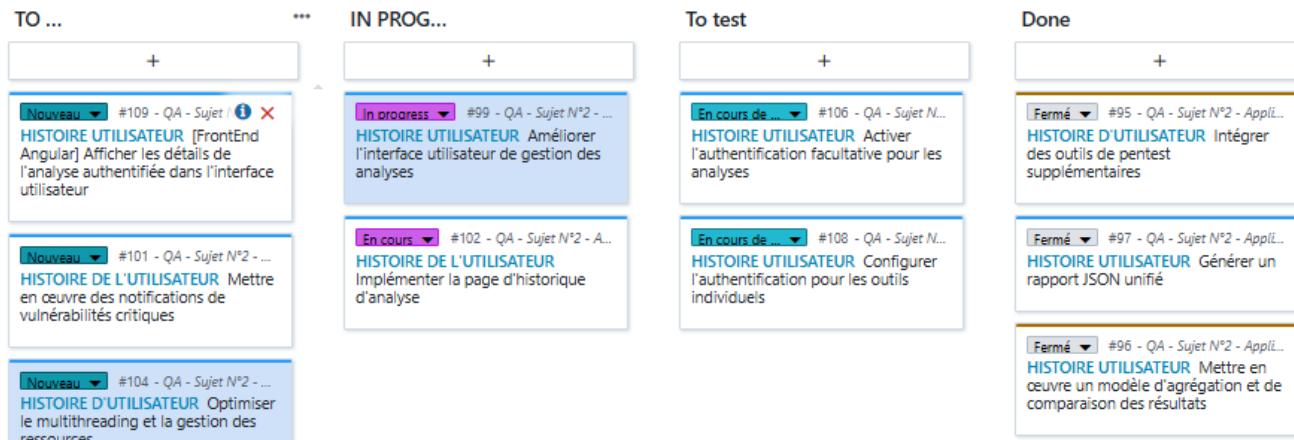


FIGURE 2.3 – Scrum board Sprint 1

- Burndown Chart

Le Burndown Chart est un graphique visuel qui montre la quantité de travail terminée et celle restant à faire dans un sprint ou un projet, en fonction du temps écoulé. Il aide à estimer la capacité de l'équipe à atteindre ses objectifs dans les délais impartis et à détecter rapidement d'éventuelles dérives. La mise à jour régulière de ce graphique est essentielle pour anticiper les retards et ajuster le rythme de travail[27].

Nous avons généré les Burndown Charts à partir de l'avancement des tâches de chaque sprint. Ces graphiques nous ont permis de suivre efficacement notre progression et d'adapter notre charge de travail, comme illustré dans la figure 2.4, représentant le Burndown chart du projet.

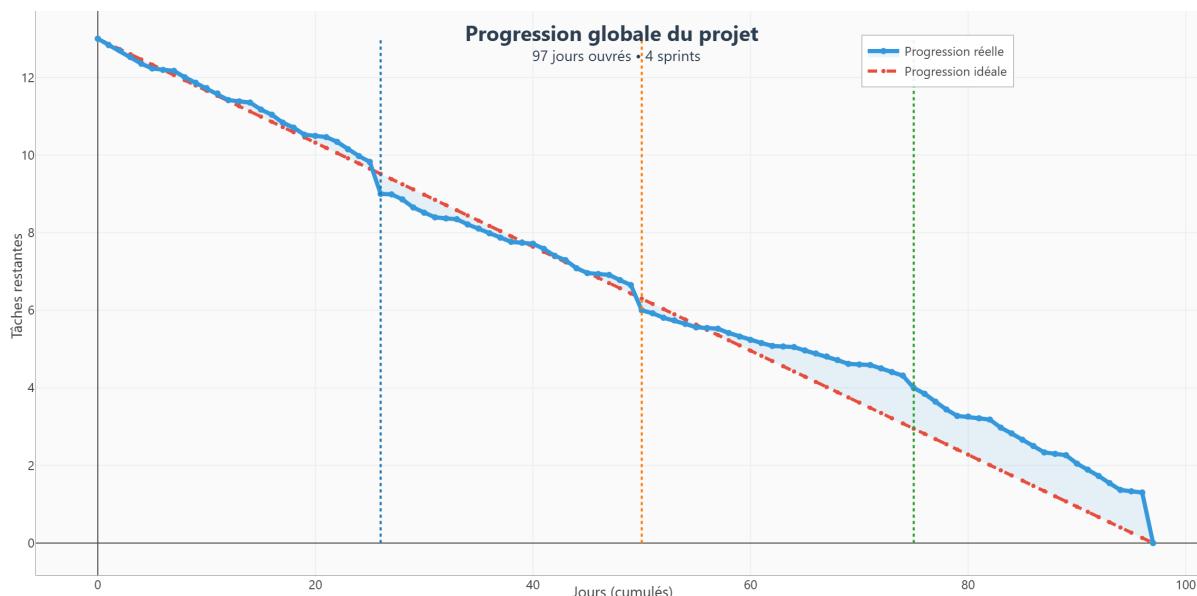


FIGURE 2.4 – Burndown chart projet

6. Environnement de développement

Cette section est dédiée pour la description de l'environnement matériel ainsi que l'environnement logiciel qui ont été employés pour la mise en œuvre de notre application

6.1. Environnement matériel

Le tableau 2.3 présente les caractéristiques de la machine utilisée pour la réalisation de notre application.

TABLE 2.3 – Environnement matériel

Type	Ordinateur Portable
Marque	ASUS
Processeur	Intel Core i5
RAM	12 Go
Disque dur	256 Go SSD
Système d'exploitation	Windows 10 (64 bits)

6.2. Environnement logiciel

Pour la réalisation de notre application, plusieurs outils logiciels ont été utilisés, incluant les environnements de développement, les bibliothèques, les frameworks, ainsi que des outils et des services externes. Cette section présente une description de ces éléments.

6.2.1. Langages utilisés

Les langages présentés dans le tableau 2.4 ont été utilisés pour développer les différentes composantes de l'application.

TABLE 2.4 – Langages utilisés

Langage	Description
	Python est utilisé pour le développement du backend avec le framework FastAPI . Ce langage est reconnu pour sa lisibilité et sa simplicité, ce qui facilite la création d'API REST performantes et évolutives[28].
	TypeScript est un sur-ensemble typé de JavaScript utilisé dans le développement de l'interface utilisateur via le framework Angular . Il permet de sécuriser le code et d'améliorer la maintenabilité des composants[29].
	HTML est le langage standard utilisé pour structurer les pages web. Il définit les éléments de base tels que les titres, paragraphes, liens, tableaux, formulaires[30].
	CSS est utilisé pour styliser les éléments HTML. Il permet de définir l'apparence des pages (couleurs, polices, marges, positionnement...) afin d'améliorer l'expérience utilisateur[31].
	SQL est utilisé pour la gestion de la base de données relationnelle via le système PostgreSQL . Il permet de créer, manipuler et interroger les données de manière structurée[32].
	YAML est utilisé pour la configuration des conteneurs dans Docker Compose . Il permet de définir les services, les volumes, les ports et les dépendances nécessaires au déploiement de l'application[33].
	JSON est utilisé comme format d'échange de données entre le frontend Angular et le backend FastAPI, ainsi que pour la documentation des API via Swagger/OpenAPI[34].
	Bash est utilisé pour automatiser l'exécution de scripts liés aux tests de sécurité, au déploiement des conteneurs Docker, à la gestion des dépendances et à l'enchaînement des outils d'analyse[35].

6.2.2. Logiciels utilisés

Le tableau 2.5 présente un récapitulatif des outils employés pour la mise en œuvre de la solution.

TABLE 2.5 – Logiciels utilisés

Logiciel	Description
	Visual Studio Code (VS Code) est un éditeur de code source open-source de Microsoft, réputé pour sa polyvalence, sa légèreté et ses fonctionnalités avancées. Il offre un environnement extensible grâce à des extensions, ce qui simplifie le travail des programmeurs[36].
	GitLab est une plateforme open source qui centralise le cycle de vie des projets logiciels, incluant gestion de code, CI/CD, gestion de projet et sécurité. Elle facilite la collaboration, l'automatisation et la traçabilité, optimisant ainsi le développement, les tests et le déploiement. Sa richesse fonctionnelle en fait un outil clé du développement moderne[37].
	Git permet la gestion efficace des branches pour travailler simultanément sur plusieurs projets sans conflits, en offrant une interface conviviale, des algorithmes de fusion intelligents pour gérer les modifications simultanées de fichiers[38].
	Docker Desktop est une application pour Windows, macOS et Linux qui fournit une interface utilisateur graphique (GUI) pour gérer les conteneurs, images et réseaux Docker localement. Elle inclut Docker Engine, Docker CLI, Docker Compose, Kubernetes et d'autres outils utiles pour le développement et le test de conteneurs[39].
	WebSocket est un protocole de communication bidirectionnelle full-duplex permettant des échanges en temps réel entre client et serveur. Il est utilisé pour transmettre en direct les résultats des scans et notifier l'utilisateur via l'interface Angular[40].
	Overleaf est une plateforme en ligne de rédaction collaborative en temps réel de documents LaTeX, conçue pour la création de documents scientifiques, académiques et techniques[41].
	StarUML est un logiciel de modélisation Unified Modeling Language (UML) utilisé pour concevoir des diagrammes pour la conception de logiciels. Il offre des fonctionnalités de modélisation avancées pour les concepteurs de logiciels[42].
 OpenProject	OpenProject est un logiciel de gestion de projet open source conçu pour être utilisé avec des méthodologies de gestion de projet traditionnelles, mais aussi dans un environnement agile ou une méthodologie hybride. Collaboratif, il offre un suivi de projet et une suite complète de fonctionnalités permettant de gérer des projets complexes[43].

	Postman est un outil de développement d'API qui permet aux développeurs de tester, de documenter et de surveiller les API de manière efficace. Il offre une interface conviviale pour créer des requêtes HTTP, automatiser des tests et collaborer sur des projets d'API [44].
	Swagger est un langage de description d'interface permettant de définir des API REST à l'aide du format JSON. Il s'appuie sur la spécification OpenAPI, qui standardise la description de la structure d'une API. Il permet de concevoir, documenter, tester et consommer des API REST facilitant ainsi la compréhension et l'intégration des services web[45].
	Selenium est un outil open-source largement utilisé pour l'automatisation des tests des applications web. Il permet d'écrire des scripts de test dans divers langages de programmation comme Java, Python, C#. Il permet de simuler des interactions utilisateur telles que les clics, la saisie de texte et la navigation entre les pages, ce qui est essentiel pour les tests fonctionnels des applications web[46].

6.2.3. Frameworks utilisées

Le tableau 2.6 présente les principales frameworks utilisées dans le développement de l'application, réparties entre le backend et le frontend. Ces frameworks ont permis de faciliter le développement, d'améliorer les performances et de garantir la sécurité et la maintenabilité du projet.

TABLE 2.6 – Bibliothèques utilisées

Bibliothèque	Description
	FastAPI est une bibliothèque Python moderne permettant de créer des API web de manière rapide, performante et avec une documentation automatique générée via OpenAPI/Swagger[47].
	Angular est un framework de développement frontend basé sur TypeScript, utilisé pour concevoir des applications web dynamiques, modulaires et performantes. Il propose une architecture robuste fondée sur des composants, un système de routage, des formulaires réactifs ainsi qu'une gestion d'état, facilitant le développement d'interfaces utilisateur complexes [48].
	Bootstrap est une bibliothèque CSS open-source qui facilite le développement de sites web réactifs. Elle fournit des composants préconçus (buttons, cartes, grilles) et garantit une compatibilité multi-appareils[49].

6.2.4. Outils de sécurité utilisés

Le tableau 2.7 présente les outils de sécurité employés pour les tests de vulnérabilités et l'analyse de sécurité de l'application.

TABLE 2.7 – Outils de sécurité utilisés

Outil	Description
	ZAP (OWASP Zed Attack Proxy) est un outil open-source de test de sécurité des applications web. Il permet de détecter automatiquement les vulnérabilités courantes telles que les failles XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) et les injections SQL dans les applications web[2].
	SQLMap est un outil automatisé open-source spécialisé dans la détection et l'exploitation des vulnérabilités d'injection SQL. Il prend en charge de nombreux systèmes de gestion de bases de données et offre des techniques avancées pour l'extraction de données et la prise de contrôle des bases de données vulnérables[14].
	Wapiti est un scanner de vulnérabilités web open-source qui effectue des audits de sécurité en analysant les pages web pour détecter les scripts et les formulaires où il pourrait injecter des données. Il permet d'identifier diverses vulnérabilités comme les injections SQL, XSS et les inclusions de fichiers[3].
	Nikto est un scanner de vulnérabilités web open-source qui effectue des tests complets contre les serveurs web pour multiple vulnérabilités, incluant plus de 6700 fichiers/programmes potentiellement dangereux, les versions obsolètes de serveurs et les problèmes de configuration spécifiques aux serveurs[13].
	Nuclei est un scanner de vulnérabilités rapide et personnalisable basé sur des templates YAML. Il permet d'effectuer des scans de sécurité à grande échelle avec une approche modulaire, supportant la détection de diverses vulnérabilités web, réseau et cloud[12].
	Nmap (Network Mapper) est un outil open-source de découverte réseau et d'audit de sécurité. Il permet de scanner les ports ouverts, identifier les services en cours d'exécution, détecter les systèmes d'exploitation et effectuer des scripts de reconnaissance avancés sur les réseaux[50].
	XSSStrike est un outil avancé de détection des vulnérabilités Cross-Site Scripting (XSS). Il utilise des techniques de fuzzing intelligentes et une analyse contextuelle pour identifier les failles XSS complexes que les scanners traditionnels pourraient manquer[15].
	PwnXSS est un outil spécialisé dans la détection avancée des vulnérabilités Cross-Site Scripting. Il offre des capacités de scanning automatisé avec des payloads personnalisés et une analyse approfondie des réponses pour identifier les failles XSS dans diverses configurations d'applications web[51].

	<p>WafW00f est un outil Python conçu pour identifier et fingerprinter les Web Application Firewalls (WAF) protégeant une application web. Il permet aux testeurs de sécurité de déterminer la présence et le type de Web Application Firewall (WAF) afin d'adapter leurs stratégies de test en conséquence[52].</p>
	<p>WhatWeb est un outil de reconnaissance web qui identifie les technologies utilisées par un site web, incluant les systèmes de gestion de contenu, les frameworks, les serveurs web, les bibliothèques JavaScript et d'autres composants technologiques. Il est essentiel pour le fingerprinting et la reconnaissance passive[53].</p>

6.2.5. Système de gestion de base de données : PostgreSQL



PostgreSQL est un système de gestion de base de données relationnelle (SGBDR) open-source reconnu pour sa robustesse, sa conformité aux standards SQL et son extensibilité. Il est utilisé pour stocker les données de manière fiable, en assurant l'intégrité des transactions, la cohérence et la sécurité des informations.

Dans notre application, PostgreSQL permet de gérer les entités principales telles que les utilisateurs, les messages, les enregistrements de données... Il est accédé via SQLAlchemy, un ORM Python, facilitant les interactions entre les objets du backend (FastAPI) et la base de données.

6.2.6. Système de gestion des files de messages : RabbitMQ



RabbitMQ est un broker de messages open-source basé sur le protocole AMQP, permettant une communication asynchrone entre microservices et composants distribués. Il facilite le traitement parallèle ainsi que la gestion des files d'attente dans l'architecture backend [54].

Dans notre application, RabbitMQ gère la file d'attente des tâches de scans de sécurité pour limiter les exécutions simultanées et éviter la surcharge. Les scans sont mis en queue puis traités par des workers dédiés, assurant ainsi une exécution fiable, scalable et tolérante aux pannes. L'intégration via la bibliothèque Python `pika` connecte RabbitMQ au backend FastAPI, orchestrant la production et la consommation des messages pour optimiser la charge et les performances.

7. Architecture de l'application

Cette section décrit l'architecture de notre solution. Elle repose sur une architecture classique en trois tiers, avec des services conteneurisés pour optimiser le déploiement et l'exécution.

L'architecture comprend trois couches principales :

- **Couche de présentation (Frontend)** : développée avec **Angular** (v.18), elle offre l’interface utilisateur accessible via navigateur sur le port **4200**. Conteneurisée avec **Docker** et servie par **NGINX** sur le port **80**, elle suit le modèle **MVVM**[55] :
 - **Model** : définitions des données via interfaces **TypeScript**.
 - **View** : composants visuels en **HTML/CSS**.
 - **ViewModel** : **Components** Angular et **Services** gérant la logique client et la communication backend.

Le frontend intègre des mécanismes assurant sécurité et interactivité :

- **Guards et routing Angular** : gestion des routes selon authentification et rôle.
- **Services Angular** : centralisent appels API REST et traitements locaux.
- **Modèles TypeScript** : formatent les échanges avec le backend.
- **WebSocket** : communication bidirectionnelle temps réel pour les mises à jour.
- **Couche applicative (Backend)** : basée sur **FastAPI**, accessible sur les ports **8000** (HTTP) et **8001** (WebSocket), elle gère la logique métier, l’authentification, les services REST, la base de données et la coordination des couches. Conteneurisée avec **Docker**, elle s’appuie sur **Uvicorn** (serveur Asynchronous Server Gateway Interface (ASGI)) pour gérer requêtes synchrones et asynchrones.

1. Composants clés FastAPI[56] :

- **Routes** : points d’entrée API REST.
- **CRUD** : logique métier et traitement des données.
- **Models** : classes ORM SQLAlchemy représentant la base.
- **Schemas Pydantic** : validation et sérialisation des données échangées.
- **Data Access Object (DAO)** : accès aux données via SQLAlchemy.
- **JWT / OAuth2** : gestion sécurisée des sessions et accès utilisateurs.
- **Canal WebSocket** : communication temps réel avec les clients.

2. Extensions spécifiques au projet :

- **RabbitMQ + Workers** : gestion asynchrone des tâches (scans) via file d’attente sur le port **5672** et interface d’administration sur **15672**, permettant une répartition optimale des ressources et notifications en temps réel.
- **Service SMTP** : envoi d’e-mails via port **587** (TLS).
- **Intégrations Slack et Jira** : notifications en temps réel et création automatique de tickets lors de détection de vulnérabilités critiques.

- **Outils de sécurité** : plusieurs scanners (OWASP ZAP, SQLMap, Wapiti, Nikto, Nuclei, Nmap, XSSStrike, PwnXSS, WafW00f, WhatWeb) exécutés dans des conteneurs Docker isolés, avec centralisation et unification des résultats.
- **Tests fonctionnels et audits SEO** :
 - **Selenium** : simulation d’interactions complexes (port **4444**).
 - **BeautifulSoup** : analyse DOM pour extraction des balises importantes.
- **Couche de données** : base relationnelle **PostgreSQL** sur le port **5432**, stockant comptes, résultats, historiques et notifications. Accès via ORM **SQLAlchemy** garantissant cohérence entre code et base.

Tous les services (Angular, FastAPI, PostgreSQL, RabbitMQ, WebSocket...) sont conteneurisés avec **Docker** et orchestrés par **Docker Compose**, communiquant via un réseau virtuel interne.

La figure 2.5 illustre cette architecture.

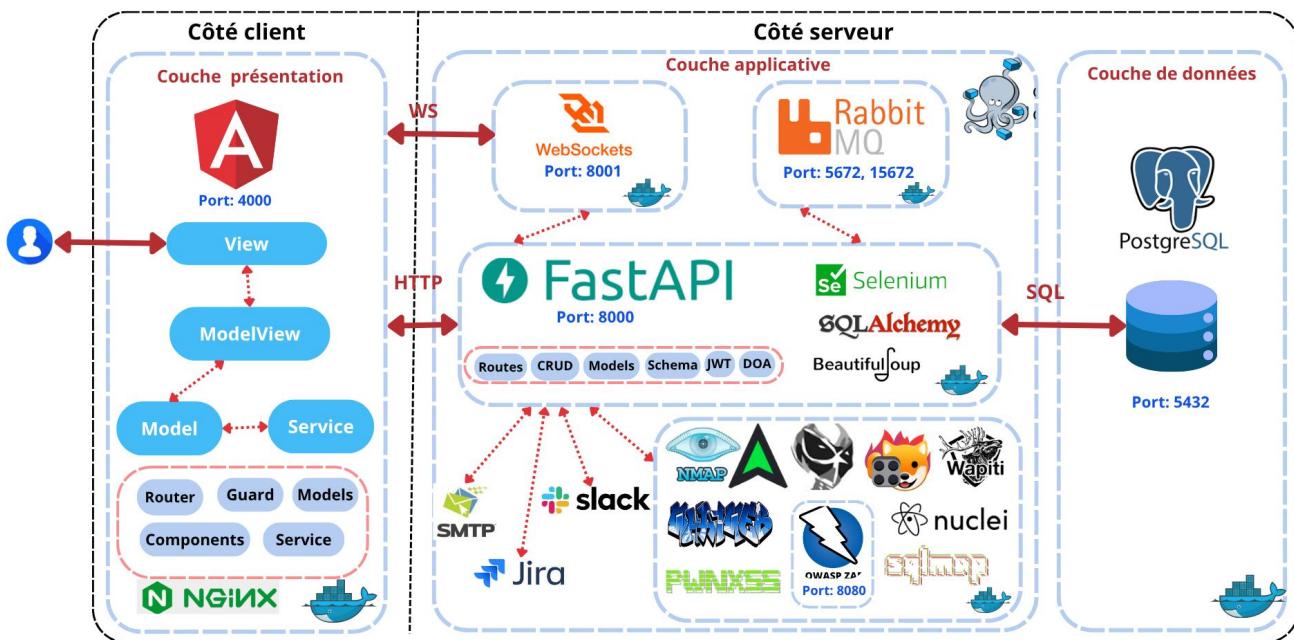


FIGURE 2.5 – Architecture de l’application

Conclusion

Dans ce chapitre, nous avons spécifié les besoins fonctionnels et non fonctionnels du futur système. Nous avons également planifié les sprints à réaliser et établi notre backlog produit, en présentant la rétrospective Agile ainsi que l’environnement de développement et l’architecture utilisée pour cette application. À ce stade, nous pouvons passer au chapitre suivant, qui détaillera le premier sprint et ses artefacts.

Release 1 : Automatisation des tests de sécurité et amélioration des fonctionnalités de base

Introduction

Dans le chapitre précédent, nous avons analysé les besoins et défini le découpage du projet. Cette première release, organisée en deux sprints sur 50 jours, marque le lancement de notre travail selon la méthodologie Scrum. Pour chaque sprint, nous décrivons la phase d'analyse, de conception et de réalisation.

1. Planification de la release 1

La release 1 a permis de livrer une première version fonctionnelle de l'application et de préparer l'intégration des modules prévus dans la prochaine version, à travers deux sprints principaux :

- **Sprint 1.1 : Initialisation, authentification et gestion des permissions (26 jours ouvrés)** : du 3 février 2025 au 11 mars 2025.
- **Sprint 1.2 : Tests de sécurité et notifications (24 jours ouvrés)** : du 12 mars 2025 au 15 avril 2025.

2. Sprint 1.1 : Initialisation, authentification et gestion des permissions

Ce premier sprint a couvert quatre axes majeurs :

- **Initialisation du projet** : création du dépôt Git, configuration des outils de travail, structuration du projet, ainsi que compréhension et analyse du code de l'ancienne application.
- **Développement de la page d'accueil** destinée aux visiteurs.
- **Authentification et gestion des profils** :
- **Gestion des utilisateurs** : développement des fonctionnalités permettant d'afficher la liste des utilisateurs, de consulter leurs informations et de gérer leurs rôles ou autorisations.

2.1. Backlog du sprint 1.1

Cette section présente le backlog du sprint 1.1, comme illustré dans le tableau 3.1.

TABLE 3.1 – Backlog du sprint 1.1

ID US	User Story	Description	ID tâche	Tâches	Priorité	Risques	Estimation (Jours)
EPIC 1 : Initialisation du projet							
1.1	Installation de l'environnement de travail	En tant que développeur, je dois installer les outils nécessaires pour travailler.	1.1.A 1.1.B	- Installer Python, VSCode, Docker, Git, npm, Angular... - Configurer l'environnement.	Élevée	Basse	1
1.2	Formation Scrum.	En tant que développeur, je dois être formé à la méthodologie Scrum afin d'organiser le travail.	1.2.A	- Participer à une session de formation SFC de SCRUMstudy. et passer un test pour obtenir la certification.	Basse	Basse	2
1.3	Formation aux tests de pénétration.	En tant que développeur, je dois être formé aux principes du pentesting pour comprendre les besoins de sécurité.	1.3.A 1.3.B	- Étudier les techniques de pentesting. - Trouver les principaux outils de base de pentesting Web et les vulnérabilités les plus courantes.	Élevée	Basse	3
1.4	Formation Selenium.	En tant que développeur, je dois être formé à l'automatisation avec Selenium pour automatiser les tests fonctionnels.	1.4.A 1.4.B 1.4.C	- Suivre une formation Selenium. - Installer et configurer Selenium. - Comprendre les bases de la manipulation des éléments Web.	Élevée	Basse	3
1.5	Analyse de la solution existante.	En tant que développeur, je dois analyser la solution existante pour identifier les fonctionnalités et les limites.	1.5.A 1.5.B	- Analyser la solution existante et identifier ses limites. - Formaliser les besoins utilisateurs en spécifications fonctionnelles.	Élevée	Moyenne	3
1.6	Formation RabbitMQ.	En tant que développeur, je dois suivre une formation sur RabbitMQ afin de comprendre le fonctionnement de la communication asynchrone entre services.	1.6.A 1.6.B	- Étudier les concepts de base de RabbitMQ (file d'attente, échange, routage). - Installer RabbitMQ en local et tester la communication entre deux services via des files de messages.	Élevée	Moyenne	3
EPIC 2 : Consultation de la page d'accueil							

2.1	Accéder à la page d'accueil	En tant que visiteur, je souhaite accéder à la page d'accueil sans avoir besoin de créer un compte.	2.1.A	- Configurer l'accès libre à la page d'accueil depuis l'URL principale.	Moyenne	Basse	1/2
2.2	Parcourir les sections de la page d'accueil	En tant que visiteur, je souhaite consulter les sections publiques (FAQ, guide utilisateur, services, équipe, tarification...) pour m'informer.	2.2.A 2.2.B	- Développer les sections publiques avec leur contenu respectif. - Intégrer les textes descriptifs, images et icônes explicatives.	Moyenne	Moyenne	4
2.3	Soumettre une demande de contact	En tant que visiteur, je souhaite envoyer un message via un formulaire pour poser une question ou obtenir de l'aide.	2.3.A	- Implémenter un formulaire de contact avec champs (nom, e-mail, message) et envoi vers l'administrateur.	Moyenne	Moyenne	1/2

EPIC 3 : Authentification et gestion du profil

40

3.1	S'inscrire	En tant que visiteur, je dois créer un compte afin d'accéder à l'application.	3.1.A	- Implémenter un formulaire d'inscription avec validation et gestion des tokens pour sécuriser les accès.	Moyenne	Moyenne	1
3.2	Vérifier l'adresse e-mail après l'inscription.	En tant que visiteur, je dois recevoir un lien de vérification afin de valider mon compte et sécuriser l'accès.	3.2.A 3.2.B	- Générer un lien de vérification unique après l'inscription. - Implémenter un service pour envoyer l'email de vérification.	Moyenne	Moyenne	1/2
3.3	S'authentifier	En tant qu'un utilisateur, je veux pouvoir me connecter pour gérer mon accès.	3.3.A	- Mettre en place une page de connexion avec un formulaire d'authentification et validation des identifiants.	Moyenne	Basse	1/2
3.4	Réinitialiser le mot de passe en cas d'oubli.	En tant qu'un utilisateur, je veux réinitialiser mon mot de passe si je l'oublie, pour retrouver l'accès.	3.4.A 3.4.B	- Créer une page de réinitialisation de mot de passe. - Implémenter la validation par email pour réinitialiser le mot de passe.	Moyenne	Moyenne	1
3.5	Gérer le profil utilisateur.	En tant qu'un utilisateur, je veux modifier mes informations personnelles pour garder mes données à jour.	3.5.A 3.5.B	- Créer une page pour modifier les informations personnelles. - Implémenter la mise à jour des utilisateur dans la base de données.	Moyenne	Moyenne	1/2

3.6	Se déconnecter	En tant qu'un utilisateur, je veux me déconnecter de l'application.	3.6.A	- Implémenter un bouton de déconnexion sur l'interface frontend avec suppression du jeton de session.	Moyenne	Basse	1/2
EPIC 11 : Gestion des utilisateurs							
11.1	Rechercher un utilisateur	En tant qu'administrateur, je souhaite rechercher un utilisateur avec différents filtres pour retrouver rapidement un profil.	11.1.A	- Implémenter un champ de recherche avec filtres (nom, email, rôle...).	Élevée	Moyenne	1/2
11.2	Exporter la liste des utilisateurs	En tant qu'administrateur, je souhaite exporter la liste des utilisateurs afin de la conserver.	11.2.A	- Générer un export (PDF/CSV/HTML /JSON) de la liste des utilisateurs.	Moyenne	Basse	1/2
11.3	Consulter la liste des utilisateurs	En tant qu'administrateur, je souhaite voir la liste des utilisateurs pour une vue globale.	11.3.A	- Afficher un tableau paginé listant les comptes avec leurs informations principales.	Moyenne	Basse	1/4
11.4	Supprimer un utilisateur	En tant qu'administrateur, je souhaite pouvoir supprimer un utilisateur du système.	11.4.A	- Ajouter un bouton de suppression avec confirmation de sécurité.	Élevée	Moyenne	1/4
11.5	Gérer les permissions des utilisateurs	En tant qu'administrateur, je souhaite attribuer, modifier ou révoquer les permissions d'un utilisateur afin de contrôler ses droits d'accès.	11.5.A	- Créer une interface permettant d'attribuer, modifier, révoquer et visualiser les permissions d'accès aux différents types de scans (fonctionnel, SEO, sécurité) pour chaque utilisateur.	Élevée	Moyenne	1/2
			11.5.B	- Implémenter des vérifications côté backend sur les routes sensibles.			
			11.5.C	- Restreindre l'accès selon les permissions via des guards côté frontend.			
TOTAL							26UE

2.2. Analyse du sprint 1.1

Nous passons à la phase d'analyse de ce sprint afin de présenter le diagramme de cas d'utilisation de ce sprint ainsi que les descriptions textuelles de quelques cas d'utilisation.

2.2.1. Diagramme de cas d'utilisation raffiné du sprint 1.1

La figure 3.1 présente le diagramme de cas d'utilisation raffiné du sprint 1.1.

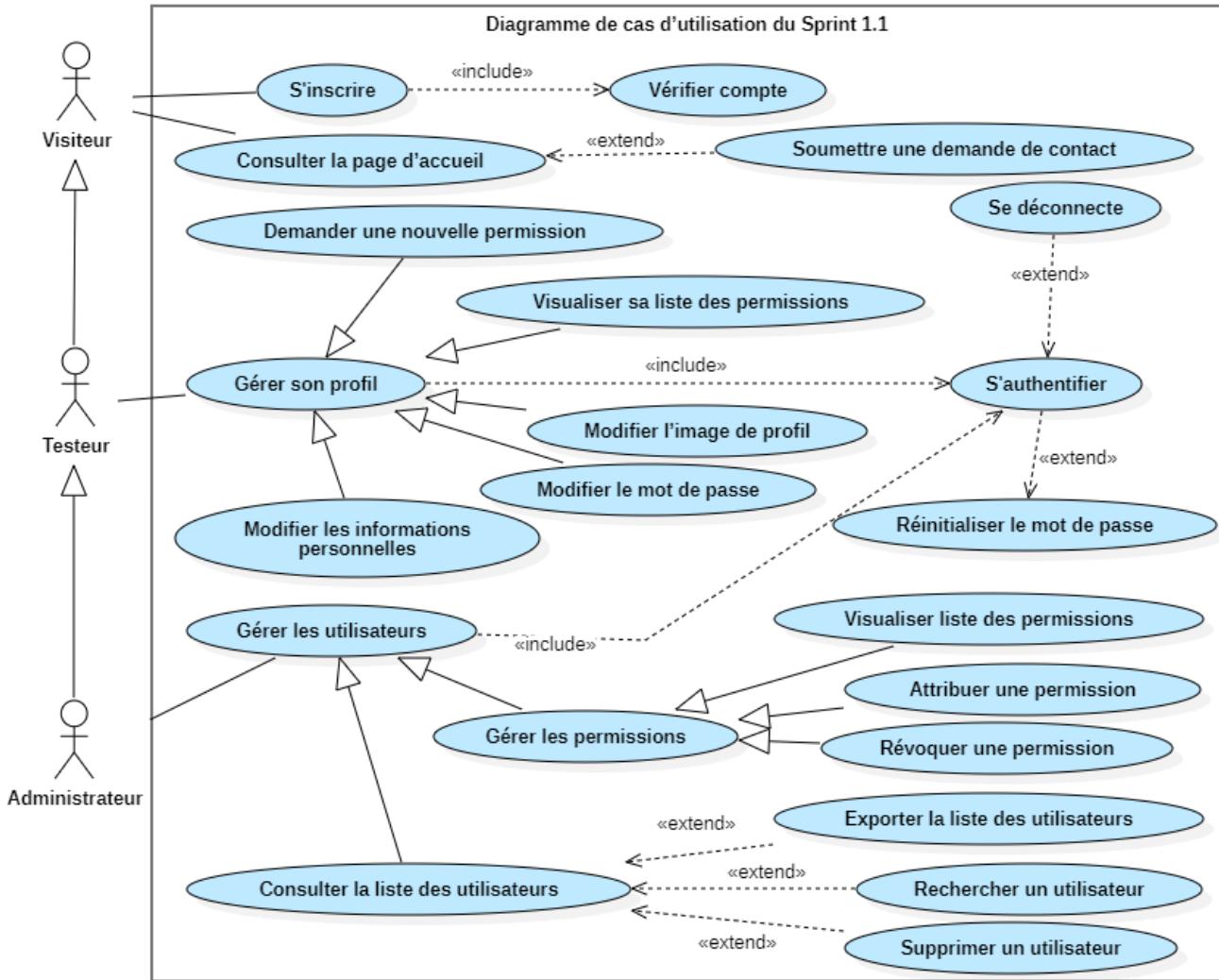


FIGURE 3.1 – Diagramme de cas d'utilisation raffiné du sprint 1.1

2.2.2. Raffinement des cas d'utilisation

Cette étape a permis de mieux comprendre les interactions, de repérer les dépendances et de découper les fonctionnalités.

a) Raffinement du cas d'utilisation «S'inscrire» :

L'inscription inclut la saisie et la validation des informations, la confirmation par e-mail avec un code One-Time Password (OTP), ainsi que la gestion des erreurs pour garantir une expérience utilisateur optimale (voir Annexe B)¹.

1. Voir Annexe B : tableau de description et diagramme de séquence d'analyse

b) **Raffinement du cas d'utilisation «Gérer les permissions des utilisateurs» :**

Cette section détaille le raffinement du cas d'utilisation «Gérer les permissions des utilisateurs». Ce cas d'utilisation permet à un administrateur de gérer les droits d'accès des utilisateurs aux différentes permissions (fonctionnel, SEO, sécurité...).

□ **Description textuelle du cas d'utilisation "Attribuer des permissions"**

Le tableau 3.2 décrit textuellement le cas d'utilisation «Attribuer des permissions».

TABLE 3.2 – Description textuelle du cas d'utilisation : Attribuer des permissions

Titre	Attribuer des permissions
Acteur	Administrateur
Résumé	Ce cas d'utilisation permet à l'administrateur d'attribuer à un utilisateur des permissions d'accès aux tests (fonctionnel, SEO, sécurité).
Pré-conditions	L'utilisateur concerné doit exister dans le système. L'administrateur doit être connecté.
Post-conditions	Les permissions sélectionnées sont enregistrées et appliquées à l'utilisateur dans la base de données.
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur accède à l'interface de gestion des permissions.2. Il sélectionne un utilisateur.3. Il choisit les permissions à autoriser.4. Il valide la configuration.5. Le système enregistre les nouvelles permissions.
Scénario d'erreur	Erreur d'enregistrement : En cas d'échec d'écriture en base de données, le système signale une erreur et annule l'opération.

□ **Diagramme d'activité : Contrôle des accès et visualisation des permissions**

Cette Figure 3.2 illustre le processus de contrôle d'accès dans l'application. Après authentification, le système vérifie le rôle de l'utilisateur ainsi que ses permissions spécifiques. Un administrateur bénéficie d'un accès complet, tandis que les autres utilisateurs ont leurs droits finement contrôlés pour chaque fonctionnalité (tests, envoi ou planification de rapports, exécution de tests). Le diagramme montre également comment la liste des permissions est affichée de manière dynamique, permettant de visualiser précisément les pages et sections accessibles selon le profil de l'utilisateur.

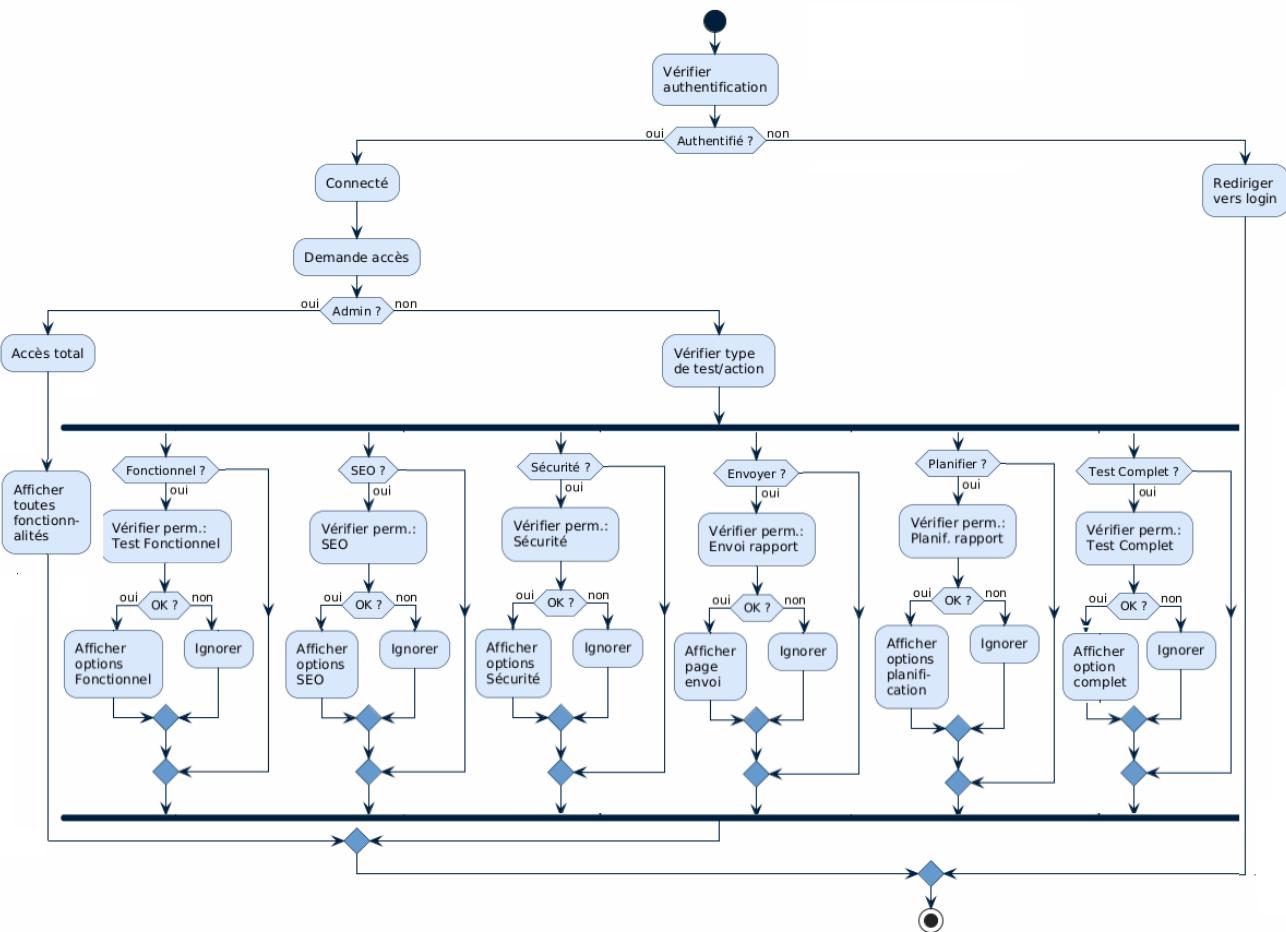


FIGURE 3.2 – Diagramme d'activité : Contrôle des accès et visualisation des permissions

2.3. Conception du sprint 1.1

La conception de ce sprint commence par la présentation du diagramme de classes.

2.3.1. Diagramme de classe du sprint 1.1

La figure 3.3 illustre le diagramme de classes du sprint 1.1, présentant les principales classes modélisées durant cette phase.

- * **Utilisateur** : Représente les comptes utilisateurs de l'application. Elle contient des informations personnelles (nom, prénom, adresse, email, mot de passe et rôle (`RoleEnum`) ainsi que des méthodes relatives à la gestion du compte et d'administration des utilisateurs.
- * **Contact_message** : Permet aux utilisateurs de soumettre des messages via un formulaire de contact. Elle comprend des attributs et des méthodes pour stocker les détails du message.
- * **RoleEnum (énumération)** : Définit les différents rôles possibles dans l'application, notamment *Administrateur* et *Testeur*, utilisés pour la gestion des droits d'accès.
- * **Permission** : Représente les permissions associées aux utilisateurs, avec les attributs suivants : type de permission, date de demande, date d'attribution, statut, ainsi que les méthodes : révoquer, attribuer et refuser une demande de permission.

- * **Token** : Modélise un jeton d'authentification, avec ses valeurs et sa date d'expiration, utilisé pour la gestion des sessions utilisateurs.
- * **Administrateur** : Hérite de la classe Utilisateur et dispose de méthodes spécifiques pour la gestion des utilisateurs, comme consulter la liste des utilisateurs, rechercher, modifier leurs permissions, supprimer des utilisateurs et exporter la liste.

Les associations entre classes sont également représentées dans le diagramme, notamment :

- Un **Utilisateur** possède un rôle défini par **RoleEnum** et peut avoir plusieurs **Permission**.
- **Administrateur** : utilisateur avec des priviléges étendus pour la gestion des autres comptes.
- Un **Utilisateur** peut générer un ou plusieurs **Token** pour gérer ses sessions.
- Un **Utilisateur** peut envoyer un ou plusieurs **Contact_message**, tandis qu'un **Contact_message** est toujours associé à un seul **Utilisateur**.

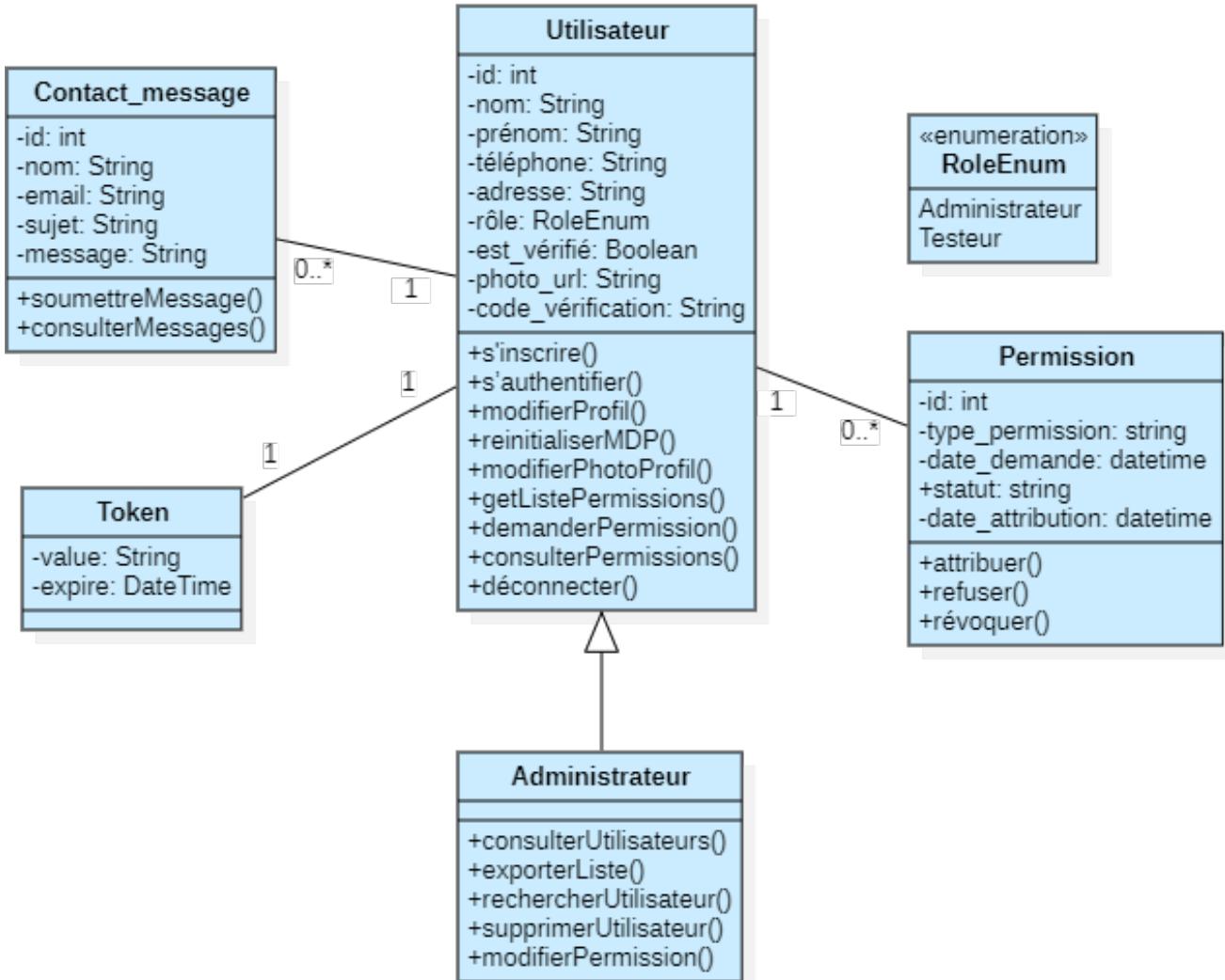


FIGURE 3.3 – Diagramme de classe du sprint 1.1

Ce diagramme constitue une base essentielle pour la suite du développement, en assurant une structure cohérente et maintenable du code tout au long des itérations agiles.

2.3.2. Diagramme de séquence de conception de cas «Réinitialiser un mot de passe»

Le diagramme de séquence présenté dans la figure B.15² décrit le processus de réinitialisation du mot de passe, depuis la demande initiale de l'utilisateur jusqu'à la mise à jour effective. Il met en évidence les interactions entre l'utilisateur, l'interface, le contrôleur d'authentification, les modèles de données et le service d'envoi d'email, garantissant ainsi la sécurité du mécanisme.

2.4. Réalisation du sprint 1.1

Dans cette section, nous présentons les principales interfaces développées durant ce sprint.

- **Interface de la page d'accueil** : Les figures (E.17 et E.18)³ présentent l'interface moderne et intuitive de la page d'accueil, offrant un aperçu général des principales fonctionnalités de l'application.
- **Interface d'inscription** : La figure 3.4 illustre l'interface d'inscription. Elle permet à un nouvel utilisateur de créer un compte en renseignant ses informations personnelles.

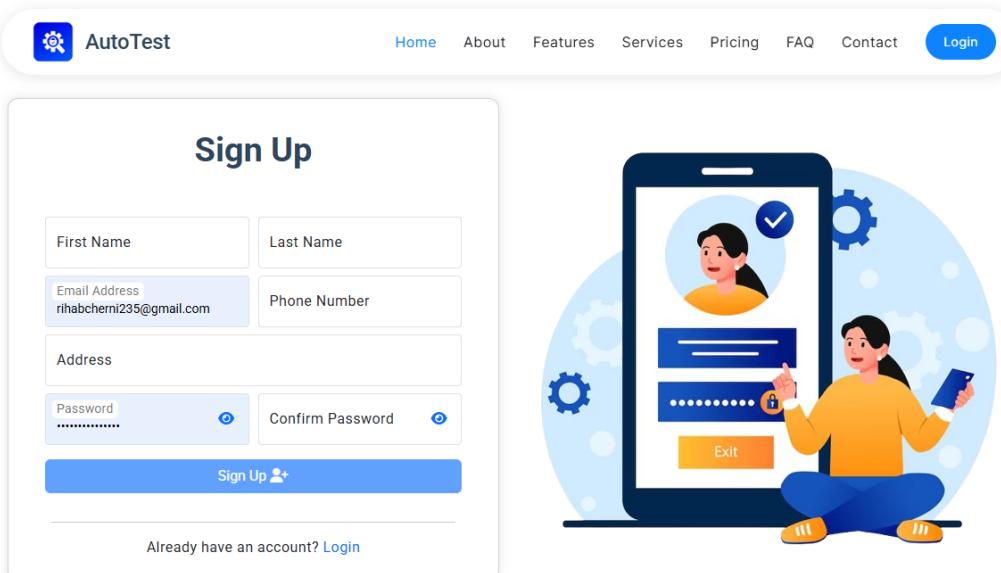


FIGURE 3.4 – Interface d'inscription

- **Interface de connexion** : La figure E.19⁴ montre l'interface de connexion, incluant des vérifications pour gérer les erreurs telles que identifiants invalides ou champs manquants.
- **Vérification par e-mail (OTP)** : Après inscription, l'utilisateur reçoit un e-mail avec un code de vérification. Les figures E.20⁵ et E.21⁶ illustrent l'interface de réception et de saisie de ce code.

2. Voir annexe B : Figure B.15

3. Voir annexe E : Figures E.17 et E.18

4. Voir annexe E : Figure E.19

5. Voir annexe E : Figure E.20

6. Voir annexe E : Figure E.21

- Mot de passe oublié et réinitialisation** : Le système propose une fonctionnalité de récupération de mot de passe, illustrée par les figures E.22, E.23 et E.24⁷.
- Profil utilisateur** : La figure 3.5 présente l'interface du profil utilisateur, permettant la consultation et la modification des informations personnelles, ainsi que la visualisation des permissions attribuées. Un bouton permet aussi de soumettre une demande d'accès à des permissions supplémentaires via une boîte de dialogue E.26c⁸ affichant la liste des permissions disponibles.

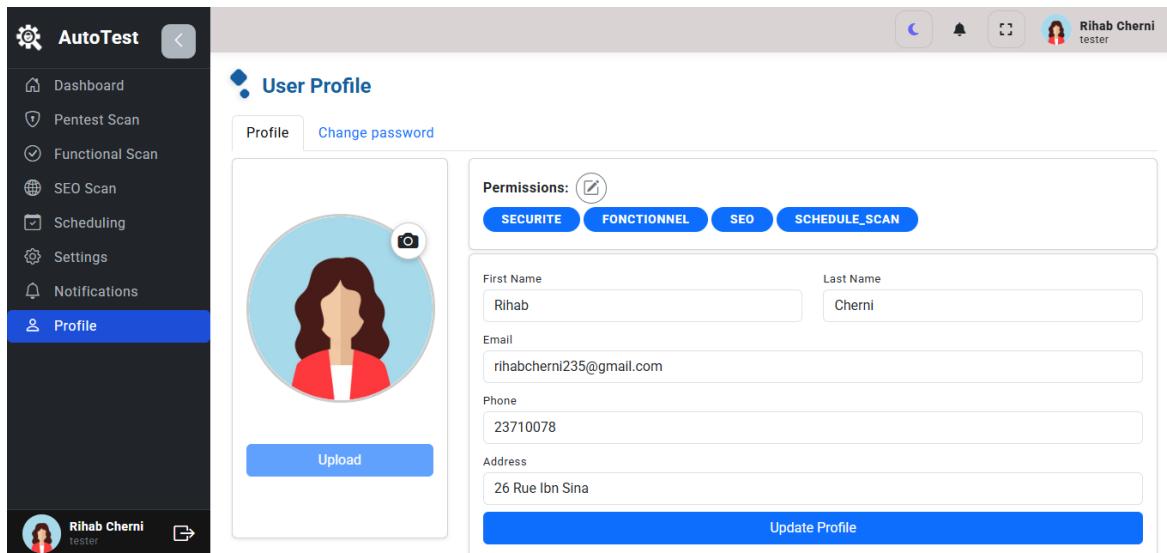


FIGURE 3.5 – Interface du profil utilisateur

- Gestion des utilisateurs et permissions** : La figure 3.6 illustre l'interface d'administration des utilisateurs, permettant leur gestion et l'attribution des permissions.

Name	Email	Phone	Address	Role	Status	Permission	Actions
admin admin	admin@gmail.com	55142365	Tunis	Admin	Verified	All	
MG	maissa@gmail.com	55123456	50 Rue 5800 Tunis	Tester	Verified	Securite Seo	
AC	arijcherni001@gmail.com	21547896	Tunis	Tester	Pending	Securite Seo	
Rihab Cherni	rihabcherni235@gmail.com	23710078	26 Rue Ibn Sina	Tester	Verified	Securite Fonctionnel Seo Schedule_scan	

FIGURE 3.6 – Interface de gestion des utilisateurs et permissions (admin)

7. Voir annexe E : Figures E.22, E.23 et E.24

8. Voir annexe E : Figure (c) E.26c

- **Interfaces liées à la gestion des permissions et guards** : Cette partie décrit les interfaces conditionnées par le système de gestion des permissions et les guards de sécurité implémentés dans l'application.
 - ◊ **Menu latéral selon les permissions** : La figure E.25⁹ illustre l'adaptation dynamique du menu latéral selon les droits d'accès de l'utilisateur, avec 3 cas : (a) testeur sans permissions, affichant seulement les éléments de base, (b) testeur avec permissions spécifiques, accédant uniquement aux modules autorisés et (c) testeur avec tous les droits, ayant un accès complet aux modules.
 - ◊ **Interfaces de gestion des permissions** : La figure E.26¹⁰ présente les interfaces illustrant les mécanismes de gestion et de demande de permissions.
 - * (a) Boîte de dialogue des permissions pour l'administrateur, avec toutes les permissions désactivées grâce à la permission spéciale **tous**, accordant un accès complet.
 - * (b) Boîte de dialogue d'édition des permissions, accessible uniquement à l'administrateur, permettant d'ajouter ou de révoquer les droits d'un **testeur**.
 - * (c) Interface de demande de permissions, permettant au testeur de visualiser les accès manquants et de soumettre une requête à l'administrateur. Une notification est automatiquement envoyée. Ce composant a été conçu avec une logique évolutive en vue de la commercialisation de l'application : certaines permissions pourront, à l'avenir, être conditionnées par un abonnement payant. Une option de paiement ou un lien vers la facturation pourra alors s'activer dynamiquement lors de la sélection.
- **Soumettre un message de contact** : L'interface illustrée dans la figure E.18¹¹ permet à tout utilisateur (même non connecté) de soumettre un message à l'administrateur.
- **Interface de liste des messages (côté administrateur)** : La figure E.27¹² présente l'interface réservée à l'administrateur pour consulter les messages du formulaire de contact.

3. Sprint 1.2 : Tests de sécurité et notifications

Ce sprint a introduit les fonctionnalités suivantes : la gestion des scans de tests de sécurité, le paramétrage des canaux de diffusion des rapports de tests et les notifications en temps réel.

3.1. Backlog du sprint 1.2

Dans cette section, nous présentons le backlog du sprint 1.2, tel qu'illustré dans le tableau 4.4.

9. Voir annexe E : Figure E.25

10. Voir annexe E : Figure E.26

11. Voir annexe E : Figure E.18

12. Voir annexe E : Figure E.27

TABLE 3.3 – Backlog du sprint 1.2

ID US	User Story	Description	ID tâche	Tâches	Priorité	Risques	Estimation (Jours)
EPIC 4 : Gestion des scans de tests de sécurité d'un site web							
4.1	Configurer les paramètres de scan selon les besoins.	En tant que testeur, je veux personnaliser les paramètres de scan pour adapter les analyses aux besoins spécifiques.	4.1.A 4.1.B	- Développer une interface pour configurer les paramètres de scan. - Permettre à l'utilisateur de choisir la profondeur des scans.	Moyenne	Moyenne	1/2
4.2	Sélectionner les outils de sécurité à utiliser pour l'analyse.	En tant que testeur, je veux choisir les outils à utiliser afin d'adapter l'analyse aux besoins de la cible, enregistrer mes préférences et les réutiliser lors des scans suivants.	4.2.A 4.2.B	- Développer une interface avec des cases à cocher permettant de sélectionner les outils souhaités avant le lancement d'un scan avec une sélection multiple et des boutons « Tout sélectionner » / « Tout désélectionner ». - Enregistrer les outils préférés de chaque utilisateur dans la base de données et les recharger automatiquement pour les scans suivants.	Élevée	Élevée	1
4.3	Lancer un scan de test de sécurité.	En tant que testeur, je veux initier un scan de test de sécurité sur une cible pour identifier ses vulnérabilités.	4.3.A 4.3.B 4.3.C 4.3.D 4.3.E	- Développer une interface de lancement de scan avec champ URL de la cible à tester. - Corriger l'automatisation des outils existants ZAP et Wapiti en vérifiant leur configuration et optimisant les paramètres de détection. - Intégrer et automatiser l'exécution d'outils spécialisés tels que SQLMap, Nuclei, Nmap... - Utiliser le multithreading pour exécuter les outils en parallèle. - Unifier les résultats et générer un rapport JSON via un modèle de comparaison, en s'appuyant sur une base de données des vulnérabilités détectées que nous avons créée.	Élevée	Moyenne	1
4.4	Lancer un scan de sécurité avec authentification dynamique.	En tant que testeur, je souhaite initier un scan authentifié afin d'identifier les vulnérabilités présentes dans les zones protégées.	4.4.A 4.4.B	- Intégrer l'authentification dynamique pour chaque outil. - Tester les mécanismes d'authentification pour chaque outil (cookies, jetons, mots de passe).	Élevée	Moyenne	3

4.5	Planifier des scans de test sécurité automatiques.	En tant que testeur, je veux définir une planification automatique des scans pour assurer une surveillance régulière.	4.5.A 4.5.B	- Créer une interface de planification pour automatiser les scans. - Tester le bon déroulement des scans planifiés.	Élevée	Moyenne	1	
4.6	Suivre la progression du scan en temps réel via WebSocket.	En tant que testeur, je veux visualiser en temps réel l'évolution des scans pour suivre leur progression.	4.6.A 4.6.B	- Intégrer WebSocket pour suivre la progression. - Mettre à jour l'interface utilisateur avec des informations en temps réel.	Élevée	Moyenne	1	
4.7	Visualiser les résultats des scans.	En tant que testeur, je peux consulter les résultats afin d'analyser la sécurité de l'application.	4.7.A 4.7.B	- Implémenter une interface pour visualiser les résultats des scans. - Fournir des options de filtrage pour faciliter l'analyse des résultats.	Élevée	Basse	2	
4.8	Accéder à l'historique des scans précédents.	En tant que testeur, je dois accéder aux rapports des anciens scans pour suivre l'évolution des vulnérabilités et conserver une trace.	4.8.A 4.8.B	- Afficher les historiques de scans avec pagination, filtres, recherche et suppression pour chaque rapport. - Permettre l'accès aux détails complets d'un scan (résumé global, vulnérabilités par outil et logs associés).	Moyenne	Basse	1	
4.9	Télécharger les rapports aux formats JSON, PDF et CSV.	En tant que testeur, je dois télécharger les rapports sous différents formats pour faciliter leur traitement et archivage.	4.9.A 4.9.B	- Développer des options d'exportation pour les rapports. - Ajouter des boutons pour télécharger les rapports en formats JSON, PDF et CSV.	Moyenne	Moyenne	1	
4.10	Intégrer et visualiser les rapports via Jira.	En tant que testeur, je dois intégrer les résultats dans Jira pour créer des tickets et assurer un suivi structuré des incidents détectés.	4.10.A 4.10.B	- Créer une intégration avec Jira pour la création automatique de tickets. - Visualiser les résultats dans des dashboards Jira.	Élevée	Élevée	3/2	
4.11	Accéder aux rapports via Slack.	En tant que testeur, je dois recevoir les rapports via Slack pour une communication rapide au sein de l'équipe.	4.11.A 4.11.B	- Mettre en place une intégration avec Slack pour envoyer les rapports. - Ajouter des notifications Slack pour chaque scan terminé.	Moyenne	Basse	3/2	

51	4.12	Recevoir les rapports directement par e-mail.	En tant que testeur, je dois recevoir automatiquement les rapports par e-mail pour assurer leur disponibilité hors plate-forme.	4.12.A 4.12.B	- Configurer l'envoi automatique de rapports par e-mail. - Ajouter un modèle d'e-mail pour l'envoi des rapports.	Moyenne	Basse	3/2
	4.13	Déetecter automatiquement les pages de login, en excluant les pages d'inscription.	En tant que testeur, je souhaite que le système identifie automatiquement les pages d'authentification d'un site pour configurer correctement les scans authentifiés.	4.13.A 4.13.B	- Analyser le code HTML des pages pour dé-déTECTER les formulaires de connexion. - Mettre en place des règles pour exclure les pages d'inscription et les pages non pertinentes.	Élevée	Moyenne	2
	4.14	Annuler un scan de sécurité en cours.	En tant que testeur, je souhaite pouvoir annuler un scan de sécurité en cours d'exécution afin d'arrêter une analyse inutile ou incorrectement configurée.	4.14.A 4.14.B	- Ajouter un bouton "Annuler" dans l'interface de suivi en temps réel du scan. - Implémenter la logique backend pour interrompre proprement l'exécution des outils lancés (thread/process/containers).	Élevée	Moyenne	2
	4.15	Relancer un scan à partir de la configuration précédente.	En tant que testeur, je souhaite relancer un scan de sécurité en utilisant les paramètres d'un ancien scan pour gagner du temps et assurer la reproductibilité des tests.	4.15.A 4.15.B	- Permettre la duplication d'un scan depuis l'historique avec récupération automatique de la configuration (outils, paramètres, cible, type d'authentification, ...). - Développer une interface "Relancer ce scan" accessible depuis les détails d'un scan précédent.	Moyenne	Moyenne	1
	EPIC 9 : Notifications en temps réel							
9.1	Être notifié des résultats pendant l'exécution des scans.	En tant que testeur, je dois recevoir des notifications immédiates pour suivre l'état des scans et détecter rapidement les incidents.	9.1.A 9.1.B	- Configurer le système de notifications en temps réel. - Tester l'envoi de notifications pendant l'exécution des scans.	Élevée	Basse	1	

9.2	Envoyer des alertes pour les vulnérabilités critiques.	En tant que testeur, je dois recevoir des alertes spécifiques pour les vulnérabilités critiques afin de pouvoir réagir rapidement.	9.2.A 9.2.B	- Définir les critères de vulnérabilités critiques pour l'envoi d'alertes. - Automatiser l'envoi des alertes en fonction de la gravité des vulnérabilités détectées.	Élevée	Élevée	1
EPIC 10 : Paramétrage des canaux de diffusion des rapports de tests							
10.1	Définir les identifiants et jetons d'accès pour Slack.	En tant que testeur, je veux configurer les identifiants d'accès Slack pour activer l'envoi automatique des rapports dans les canaux de l'équipe.	10.1.A 10.1.B	- Ajouter un formulaire de saisie des tokens Slack. - Tester l'envoi automatisé d'un rapport via Slack.	Élevée	Moyenne	1/2
10.2	Saisir les adresses e-mail des destinataires.	En tant que testeur, je veux définir les adresses email des destinataires pour permettre la diffusion automatique des rapports.	10.2.A 10.2.B	- Créer une interface de configuration des adresses e-mail. - Tester l'envoi de rapports PDF par e-mail.	Moyenne	Faible	1/2
10.3	Configurer les paramètres d'intégration Jira.	En tant que testeur, je veux configurer Jira pour automatiser la création de tickets à partir des résultats de scan.	10.3.A 10.3.B	- Développer une interface de saisie des paramètres Jira (URL, l'ID projet et les clés API). - Tester l'intégration avec la création d'un ticket depuis un rapport.	Élevée	Moyenne	1/2
10.4	Sélectionner les types et formats de rapports à envoyer.	En tant que testeur, je souhaite choisir quels types (sécurité, fonctionnel, SEO) et quels formats (HTML, PDF, JSON) de rapports seront transmis.	10.4.A 10.4.B	- Implémenter une interface pour sélectionner les types et formats de rapport. - Tester l'envoi avec les différentes combinaisons sélectionnées.	Moyenne	Moyenne	1/4
10.5	Activer ou désactiver les canaux de notification.	En tant que testeur, je veux activer ou désactiver les notifications Slack, Email ou Jira.	10.5.A	- Ajouter des boutons d'activation/désactivation pour chaque canal.	Faible	Faible	1/4
TOTAL						24 (Jours)	

3.2. Analyse du sprint 1.2

Nous entamons à présent la phase d'analyse du sprint 1.2. Cette section présente le diagramme de cas d'utilisation correspondant aux fonctionnalités ciblées durant ce sprint, accompagné de descriptions textuelles détaillées pour certains cas d'usage.

3.2.1. Diagramme de cas d'utilisation raffiné du sprint 1.2

La figure 3.7 illustre le diagramme de cas d'utilisation raffiné du sprint 1.2. Il met en avant les différents cas d'usage planifiés pour ce sprint, en mettant l'accent sur les interactions entre les utilisateurs et les composants du système.

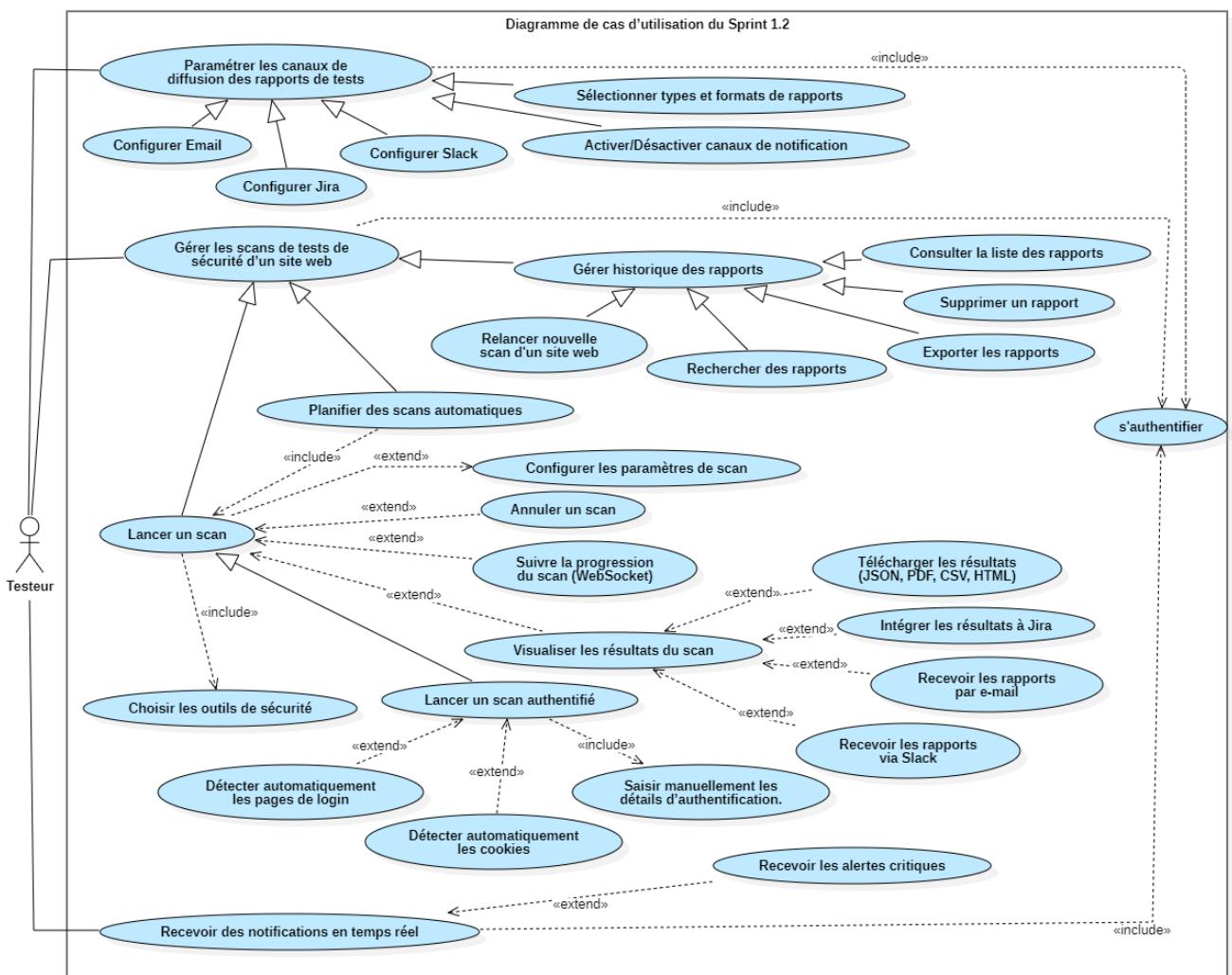


FIGURE 3.7 – Diagramme de cas d'utilisation raffiné du sprint 1.2

3.2.2. Raffinement des cas d'utilisation

Cette phase d'affinement a permis de clarifier les interactions entre les acteurs et le système, d'identifier les éventuelles dépendances fonctionnelles et de décomposer les fonctionnalités complexes en sous-cas d'utilisation plus précis.

- Description textuelle du cas d'utilisation "Choisir les outils de sécurité"

Le tableau 3.4 présente la description textuelle du cas d'utilisation "Choisir les outils de sécurité".

TABLE 3.4 – Description textuelle du cas d'utilisation : Choisir les outils de sécurité

Titre	Choisir les outils de sécurité
Acteur	Testeur
Résumé	Ce cas d'utilisation permet au testeur de sélectionner les outils qu'il souhaite utiliser pour les futurs scans et enregistre cette configuration pour la réutiliser.
Pré-conditions	Le testeur est connecté. Les outils disponibles sont listés dynamiquement depuis le backend.
Post-conditions	Les préférences du testeur sont enregistrées en base de données et réutilisées automatiquement dans les scans suivants.
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur accède à l'interface de sélection.2. Il choisit les outils à utiliser via des cases à cocher.3. Il valide sa sélection.4. Le backend enregistre la configuration.5. Un message de confirmation apparaît.
Scénario d'erreur	<ul style="list-style-type: none">• Étape 3 (Informations incomplètes) : ✗ Si aucun outil n'est sélectionné alors le système affiche un message d'erreur.• Étape 4 (Erreur d'enregistrement) : ✗ En cas d'échec lors de l'enregistrement des données, le système affiche un message d'erreur "préférences non enregistrées" et invite à réessayer ultérieurement.

3.3. Conception du sprint 1.2

La phase de conception du sprint 1.2 débute par l'élaboration du diagramme de classes, suivi de diagrammes de séquence représentant divers cas d'utilisation.

3.3.1. Diagramme de classe du sprint 1.2

Ce diagramme vise à modéliser les principales entités métier du système, leurs attributs, leurs méthodes ainsi que les relations existantes entre elles.

La figure 3.8 présente le diagramme de classes correspondant à ce sprint.

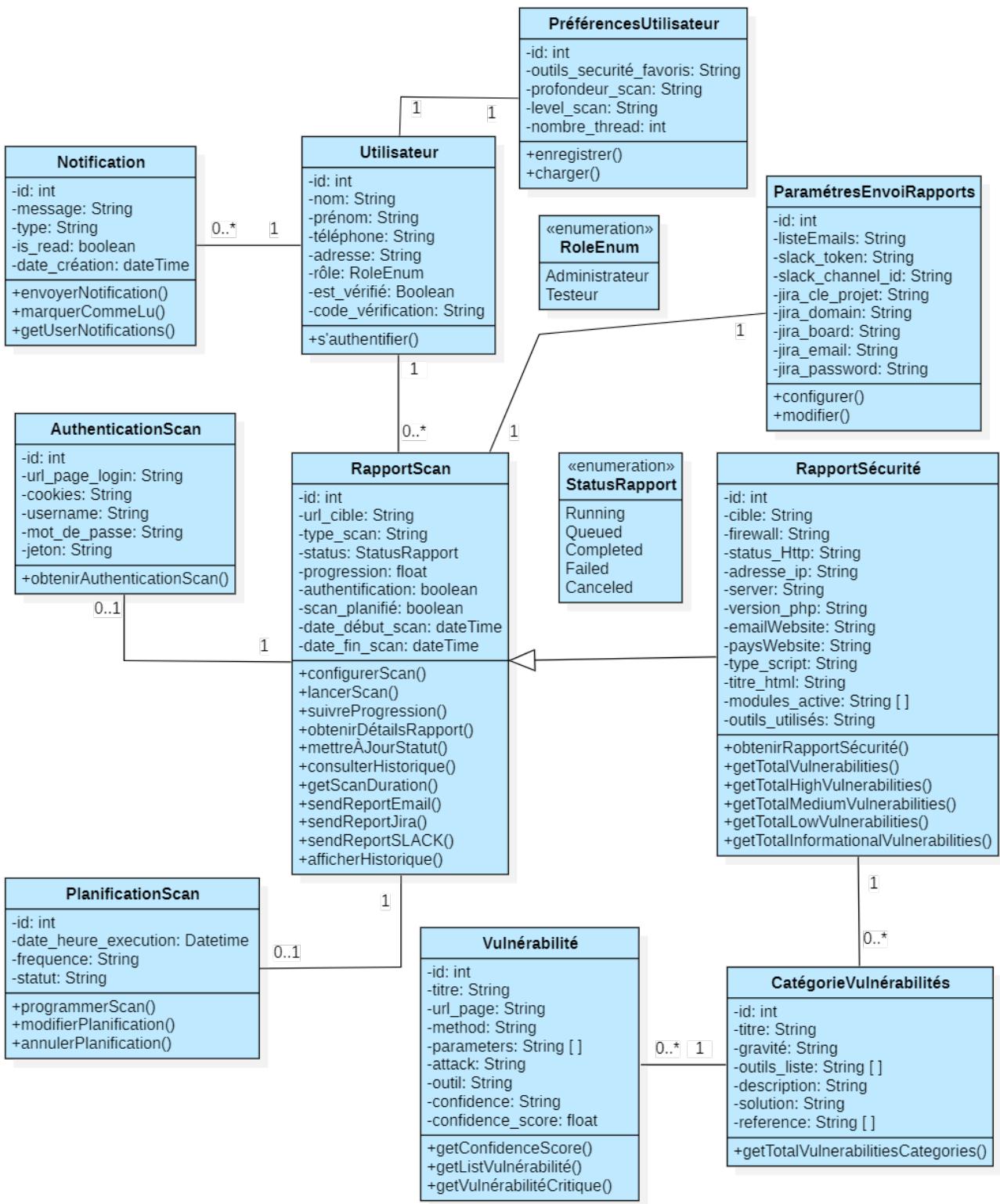


FIGURE 3.8 – Diagramme de classe du sprint 1.2

Les principales classes modélisées sont les suivantes :

- * **Utilisateur** : Représente un utilisateur de l'application.
- * **PréférencesUtilisateur** : contient les préférences d'un utilisateur pour les scans (profondeur, nombre de threads, outils favoris), associée en relation 1-à-1 avec la classe Utilisateur.
- * **Notification** : Gère les notifications envoyées aux utilisateurs, avec des attributs comme le

message, le type et la date de création. Un utilisateur peut recevoir plusieurs notifications

- * **ParametresEnvoiRapports** : Stocke les paramètres d'envoi des rapports (jetons, identifiants...) associés à chaque canal de communication (email, Slack, Jira).
- * **RapportScan** : Regroupe les données liées à un scan de sécurité, comme le type de scan, son état (via **StatusRapport**), la cible, les dates de début/fin, les outils utilisés... Cette classe contient également des méthodes pour configurer, lancer et suivre un scan.
- * **AuthenticationScan** : Contient les informations nécessaires pour effectuer un scan authentifié (page de login, identifiants, cookies, jetons).
- * **RapportSecurite** : Fournit des détails techniques sur l'environnement cible d'un scan : en-têtes HTTP, serveur, version PHP, adresse IP, pays d'hébergement, titre HTML...
- * **PlanificationScan** : Permet de planifier l'exécution automatique d'un scan à une date donnée avec une fréquence spécifique. Elle offre des méthodes de gestion comme la mise à jour ou l'annulation d'un scan planifié. Un scan peut être associé à une planification.
- * **Vulnérabilité** : Décrit une vulnérabilité détectée, avec ses détails techniques (type d'attaque, méthode HTTP, paramètres concernés, gravité, score de confiance...) avec des méthodes pour calculer le nombre de vulnérabilités par criticité..
- * **CatégorieVulnérabilités** : Catégorise les vulnérabilités détectées selon leur nature, leur niveau de risque, les outils les ayant détectées, la description, la solution recommandée... Une catégorie peut regrouper plusieurs vulnérabilités.
- * **StatusRapport (Énumération)** : Représente l'état d'avancement d'un rapport de scan. Les valeurs possibles sont : `Running`, `Queued`, `Completed`, `Failed` et `Canceled`.

Les associations entre classes sont également représentées dans le diagramme, notamment :

- Un **utilisateur** peut recevoir plusieurs **notifications**.
- Un **scan** peut être planifié via la classe "**PlanificationScan**".
- Un **rapport de sécurité** est associé à un rapport de scan.

3.3.2. Diagramme de séquence de conception du cas «Lancer un scan de test de sécurité»

La figure 3.9 illustre l'enchaînement des interactions entre les composants du système lors du lancement d'un scan de test de sécurité. Le processus commence lorsque le testeur initie un scan via l'interface utilisateur, ce qui entraîne la création d'une tâche dans la file RabbitMQ. Un worker prend alors en charge cette tâche : si la file est pleine, le scan est mis en attente ; sinon, il est exécuté de manière multithreadée. Les informations de progression sont transmises en temps

réel à l'utilisateur via WebSocket et toute vulnérabilité critique détectée déclenche une alerte immédiate. Une fois le scan terminé, les résultats sont enregistrés et envoyés à l'utilisateur.

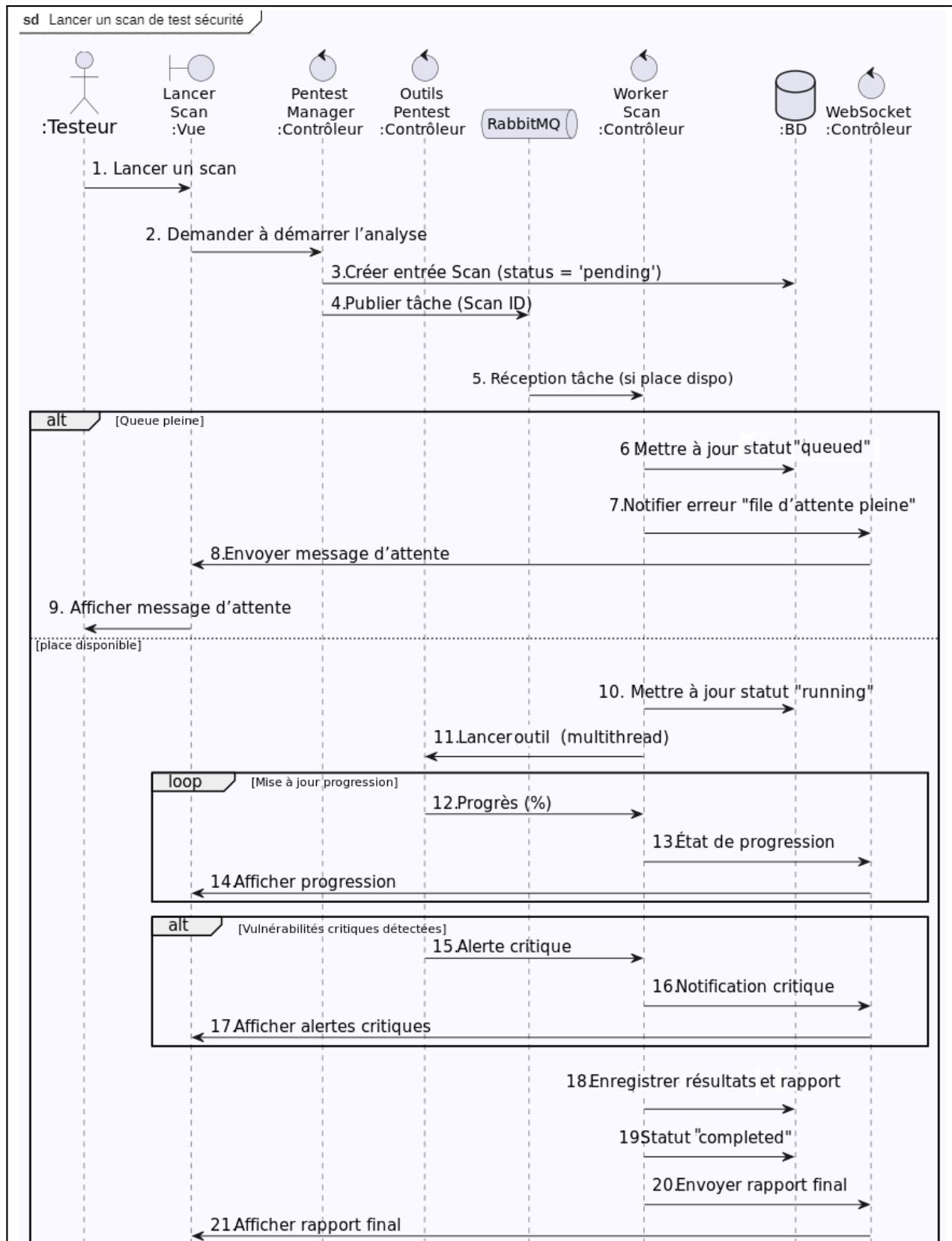


FIGURE 3.9 – Diagramme de séquence de conception de cas «Lancer un scan de test sécurité»

3.4. Réalisation du sprint 1.2

Dans cette section, nous présentons les principales interfaces réalisées au cours du sprint 1.2.

- **Interface de configuration des paramètres de scan** : La figure E.28¹³ illustre l'interface de configuration des outils de scan. Elle permet à l'utilisateur de personnaliser les paramètres d'analyse, tels que la profondeur de crawl, l'activation de modules spécifiques, ainsi que le réglage détaillé des options propres à chaque outil.
- **Interface de sélection des outils de sécurité** : La figure 3.10 présente la sélection des outils de sécurité. Cette interface propose une liste de cases à cocher pour activer ou désactiver les outils disponibles, accompagnée de boutons «Tout sélectionner» et «Tout désélectionner». Les préférences de l'utilisateur sont sauvegardées pour les réutiliser dans les futurs scans.

The screenshot shows the 'Security Scanner' interface. On the left is a vertical toolbar with icons for settings, home, shield, checkmark, globe, file, gear, and a wrench. The main area has a header 'Scan URLs in real time to ensure your websites stay secure.' and sub-headings 'Scan Configuration', 'Select Tools' (which is underlined in blue), and 'Advanced Options'. Below this is a section titled 'Select Scanning Tools' with ten cards arranged in two rows of five. Each card contains an icon, the tool name, and a brief description. Buttons 'Select All' and 'Deselect All' are at the top right of the tool list. The tools listed are: ZAP, Wapiti, Nikto, Nmap, Nuclei, PwnXSS, SQLMap, XSSStrike, Whatweb, and Wafw00f. The Nmap card is highlighted with a blue border.

FIGURE 3.10 – Interface de sélection des outils de sécurité

- **Interface utilisateur pour le lancement de scans, avec ou sans authentification** : La figure 3.11 présente l'interface de lancement d'un test de sécurité, où l'utilisateur saisit l'URL cible, sélectionne éventuellement les outils et paramètres, ou utilise des préférences précédentes. Un bouton lance l'analyse. Une section d'authentification dynamique, activée par boutons radio, affiche les champs adaptés (identifiants, jetons ou cookies) pour tester les zones protégées.

13. Voir Annexe E, figure E.28

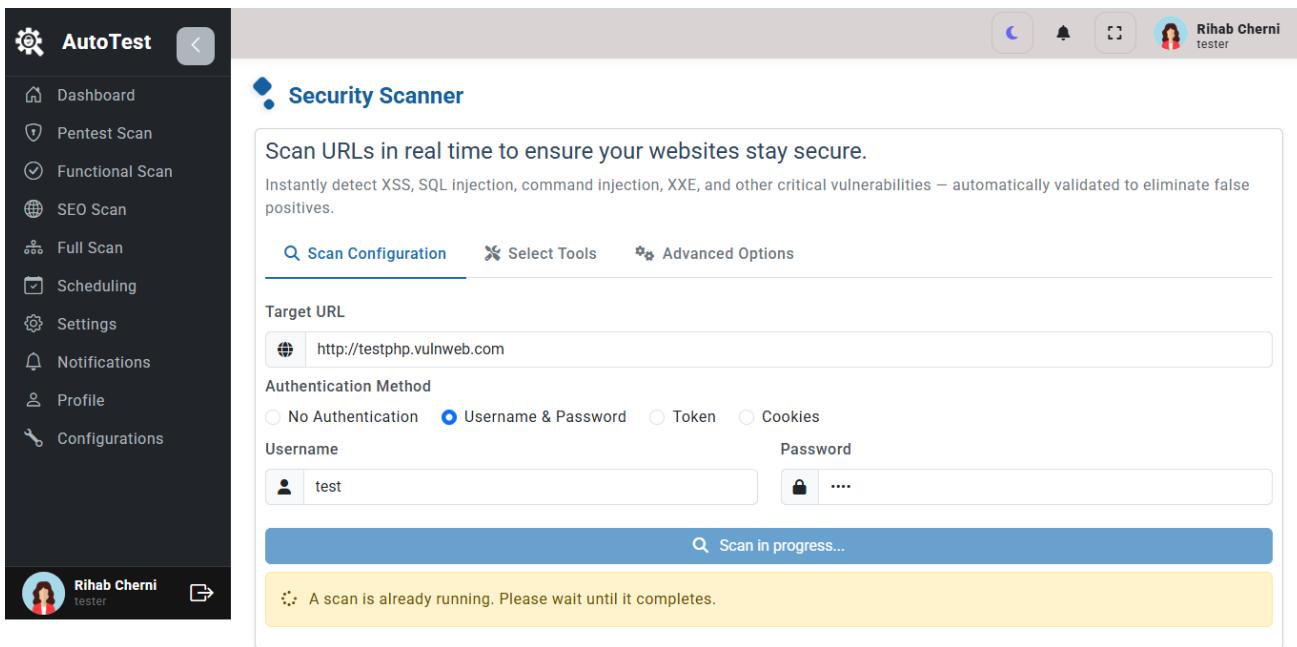


FIGURE 3.11 – Interface de lancement de scan avec ou sans authentification

- **Interface de suivi en temps réel des scans** : Comme illustré dans la figure E.29¹⁴, cette interface permet de suivre l'évolution du scan en temps réel grâce à l'intégration des WebSockets, avec un indicateur de progression affiché de 0 à 100 %.
- **Interfaces de visualisation des résultats** : Les figures (E.30, E.33, E.32 et E.31)¹⁵ montre l'écran de consultation des vulnérabilités détectées. Des filtres sont proposés pour affiner l'analyse et chaque vulnérabilité est accompagnée d'un résumé, de son risque, de sa source et d'une suggestion de correction. Les figures présentent successivement : les statistiques, la liste des vulnérabilités classées par outil, l'agrégation des résultats multi-outils, ainsi que les journaux d'exécution détaillés des outils utilisés.
- **Interface de l'historique des scans** : L'interface illustrée par la figure E.34¹⁶ permet d'accéder aux rapports précédemment générés. Une table paginée présente les scans enregistrés, accompagnée de filtres, d'un champ de recherche et d'une option de suppression.
- **Interface de téléchargement des rapports** : Comme présenté dans la figure E.35¹⁷, cette interface permet de télécharger les rapports aux formats JSON, PDF ou CSV, selon les besoins de l'utilisateur.
- **Interface pour le paramétrage des canaux de diffusion** : La figure E.38¹⁸ illustre l'interface dédiée à la configuration détaillée des canaux de diffusion des rapports. Elle permet à l'utilisateur de spécifier, pour chaque type de test (sécurité, SEO, fonctionnel),

14. Voir Annexe E, figure E.29

15. Voir Annexe E, les figures E.30, E.33, E.32 et E.31

16. Voir Annexe E, figure E.34

17. Voir Annexe E, figure E.35

18. Voir Annexe E, figure E.38

les canaux de notification souhaités (Slack, Jira, Email), le format du rapport associé (PDF, HTML, JSON). Des champs dynamiques permettent d'ajouter plusieurs destinataires e-mail et d'entrer des clés API sécurisées pour Slack et Jira. Cette interface vise à centraliser et automatiser la diffusion ciblée des rapports vers les bons interlocuteurs.

- **Interface d'envoi des résultats de scan** : Les figures (E.39 et E.40)¹⁹ illustrent l'envoi automatique des rapports via Slack et e-mail, déclenché à la fin du scan. Les notifications contiennent un résumé des vulnérabilités au format JSON, PDF...
- **Interface de planification automatique des scans** : La figure E.41²⁰ illustre l'écran de planification permettant à l'utilisateur de définir des scans récurrents. Il peut spécifier la fréquence (quotidienne, hebdomadaire, mensuelle), l'heure d'exécution, la cible concernée...
- **Interface des notifications** : La figure 3.12 ainsi que le menu de notifications présenté en figure E.36²¹ illustrent une interface d'affichage des notifications en temps réel via WebSocket. Celle-ci permet de consulter les notifications reçues avec des filtres par type (alerte, erreur, avertissement, succès) et par statut (lues ou non lues).

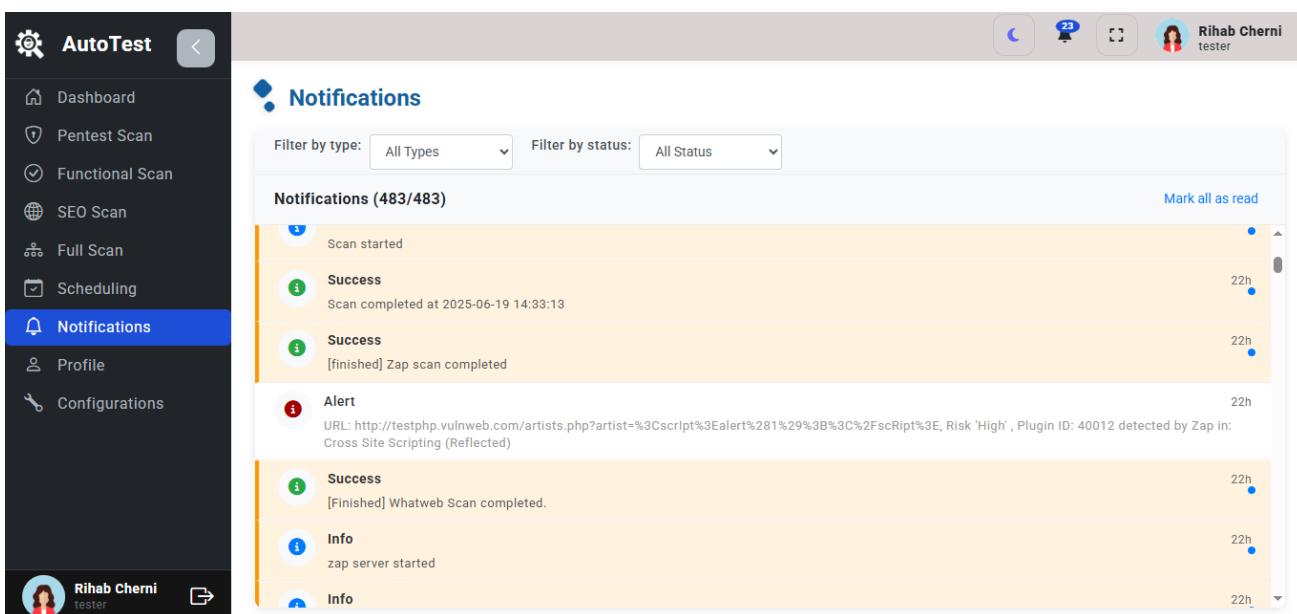


FIGURE 3.12 – Interface des notifications

Conclusion

Ce premier sprint a constitué une étape fondamentale dans le développement de notre application, avec l'implémentation de l'authentification, de la gestion des utilisateurs, des scans de sécurité et des notifications en temps réel. Le sprint suivant visera à enrichir ces fonctionnalités, à améliorer l'ergonomie et à poursuivre l'automatisation des tests fonctionnels et SEO.

19. Voir Annexe E, figures E.39 et E.40

20. Voir Annexe E, figure E.41

21. Voir Annexe E, figure E.36

Release 2 : Automatisation des tests fonctionnels et SEO, génération des rapports et dockerisation

Introduction

Dans le chapitre précédent, nous avons décrit le premier sprint et abouti à une première version de l'application. Dans ce chapitre, nous commencerons à travailler sur le deuxième incrément, qui sera principalement axé sur l'automatisation des tests fonctionnels, la gestion des résultats, la génération de rapports, la création de tableaux de bord et la gestion des utilisateurs.

1. Planification de la release 2

Cette seconde itération a permis d'intégrer les fonctionnalités avancées et d'assurer la finalisation du projet. La release 2 s'est déroulée selon l'organisation suivante :

- **Sprint 2.1 — Intégration des fonctionnalités clés de gestion des scans (25 jours ouvrés)** : du 16 avril 2025 au 21 mai 2025.
- **Sprint 2.2 — Finalisation et dockerisation de l'application (22 jours ouvrés)** : du 22 mai 2025 au 23 juin 2025.

2. Sprint 2.1 : Tests automatisés fonctionnels et SEO

Ce sprint a porté sur l'implémentation des autres types de scans.

- **Gestion des scans fonctionnels** : développement des fonctionnalités permettant de lancer et de suivre les tests fonctionnels sur les sites web.
- **Gestion des analyses SEO** : intégration des outils d'analyse SEO pour évaluer la performance et la qualité du référencement des sites web.

Ce sprint a permis d'enrichir l'application avec des fonctionnalités essentielles pour l'analyse complète et automatisée des sites web.

2.1. Backlog du sprint 2.1

Dans cette section, nous présenterons le Backlog du sprint 2, illustré dans le tableau 4.1, en tenant compte des modifications et ajustements apportés depuis le sprint précédent.

TABLE 4.1 – Backlog du sprint 2.1

ID US	User Story	Description	ID tâche	Tâches	Priorité	Risques	Estimation(j)
EPIC 5 : Gestion des tests fonctionnels d'un site web							
5.1	Gérer les scénarios de test "Workflow".	En tant que testeur, je dois gérer les scénarios de test pour assurer la couverture fonctionnelle.	5.1.A 5.1.B	- Ajouter la possibilité de créer, modifier, exécuter et supprimer des scénarios de test. - Afficher les résultats des scénarios.	Élevée	Moyenne	2
5.2	Gérer les cas de test associés à chaque scénario de test.	En tant que testeur, je dois manipuler les cas de test pour valider les différentes fonctionnalités.	5.2.A 5.2.B	- Implémenter la création, la modification, la suppression et l'exécution de cas de test pour chaque scénario. - Afficher les résultats et les erreurs liées aux cas de test.	Élevée	Moyenne	2
5.3	Gérer les étapes associées à chaque cas de test.	En tant que testeur, je dois gérer les étapes de test pour garantir le bon déroulement des tests.	5.3.A 5.3.B	- Implémenter la gestion des étapes de test pour chaque cas. - Ajouter un suivi de l'état des étapes de test exécutées.	Moyenne	Basse	2
5.4	Lancer un scan de test fonctionnel.	En tant que testeur, je souhaite exécuter automatiquement des tests fonctionnels afin de vérifier la fiabilité de l'application.	5.4.A 5.4.B	- Ajouter la fonctionnalité pour démarrer un scan de test fonctionnel. - Générer et afficher les rapports de tests après exécution.	Moyenne	Basse	2
5.5	Planifier l'exécution des scans fonctionnels	En tant que testeur, je souhaite planifier l'automatisation des tests fonctionnels pour garantir une exécution régulière.	5.5.A 5.5.B	- Ajouter un système de planification (scheduler). - Permettre l'exécution automatique selon un planning défini.	Élevée	Moyenne	2
5.6	Suivre l'exécution fonctionnels en temps réel	En tant que testeur, je veux suivre les tests via WebSocket pour visualiser la progression.	5.6.A	- Intégration WebSocket pour monitoring en temps réel.	Moyenne	Basse	1
5.7	Visualiser les résultats des tests fonctionnels	En tant que testeur, je souhaite consulter les résultats des tests exécutés pour corriger les bugs.	5.7.A 5.7.B	- Créer une interface pour visualiser les anomalies et logs de tests. - Ajouter des filtres et détails pour chaque scénario ou cas de test.	Élevée	Moyenne	2

5.8	Intégrer les résultats de tests avec Jira, Slack et email	En tant que testeur, je souhaite envoyer les anomalies détectées vers Jira, Slack et email automatiquement.	5.8.A 5.8.B 5.8.C	- Ajouter une intégration API avec Jira pour la création de tickets. - Envoi automatisé des résultats via Slack. - Envoi des rapports par email.	Moyenne	Moyenne	2
5.9	Gérer l'historique et télécharger les rapports.	En tant que testeur, je souhaite consulter l'historique des rapports des scans fonctionnels et les exporter.	5.9.A 5.9.B	- Stocker l'historique des rapports (base de données ou fichiers). - Exporter les rapports HTML, JSON, CSV, PDF, ZIP.	Élevée	Moyenne	2
EPIC 6 : Gestion des analyses SEO d'un site web							
6.1	Lancer une analyse SEO complète	En tant que testeur, je veux analyser le site cible pour évaluer sa qualité SEO.	6.1.A 6.1.B	- Implémenter l'analyse SEO : balises, performances, mots clés... - Calculer un score global SEO.	Élevée	Moyenne	2
6.2	Identifier les technologies et mots-clés	En tant qu'auditeur, je veux connaître les frameworks utilisés et les mots-clés extraits.	6.2.A	- Extraire les CMS, JS, langages backend et mots-clés textuels.	Moyenne	Moyenne	1
6.3	Générer une capture d'écran de la page cible	En tant que testeur, je veux voir un aperçu visuel de la page analysée.	6.3.A	- Générer automatiquement une capture d'écran avec Puppeteer ou outil équivalent.	Moyenne	Faible	1
6.4	Suivre l'analyse SEO en temps réel	En tant que testeur, je souhaite voir la progression du scan SEO en live.	6.5.A	- Intégration WebSocket pour progression de l'analyse.	Basse	Faible	1
6.5	Visualiser et exploiter les résultats SEO	En tant que testeur, je veux lire les points forts et axes d'amélioration SEO.	6.5.A	- Créer une interface de visualisation du rapport SEO et classer par catégorie : technique, contenu, performance...	Élevée	Moyenne	2
6.6	Intégration Jira / Slack / Email pour SEO	En tant qu'auditeur, je veux notifier et suivre les anomalies SEO détectées.	6.6.A 6.6.B 6.6.C	- Créer des tickets Jira automatiquement pour problèmes critiques. - Notification via Slack. - Envoi du rapport SEO par email.	Moyenne	Faible	2
6.7	Historique et export des rapports SEO	En tant que testeur, je veux pouvoir retrouver et télécharger les rapports SEO précédents.	6.7.A	- Gérer l'historique et exporter les rapports (HTML, JSON, PDF, ZIP, CSV).	Moyenne	Moyenne	1

2.2. Analyse du sprint 2.1

Dans cette phase, nous présentons les cas d'utilisation détaillés du sprint 2.1 en les divisant en deux diagrammes raffinés. Cette séparation permet de mieux structurer les fonctionnalités en fonction du type d'analyse réalisé.

2.2.1. Diagramme de cas d'utilisation raffiné : Gérer les analyses SEO

La figure 4.1 illustre les interactions entre le testeur et les fonctionnalités liées aux analyses SEO. Elle inclut la configuration des paramètres d'analyse, le lancement des tests SEO, ainsi que la visualisation et l'export des résultats.

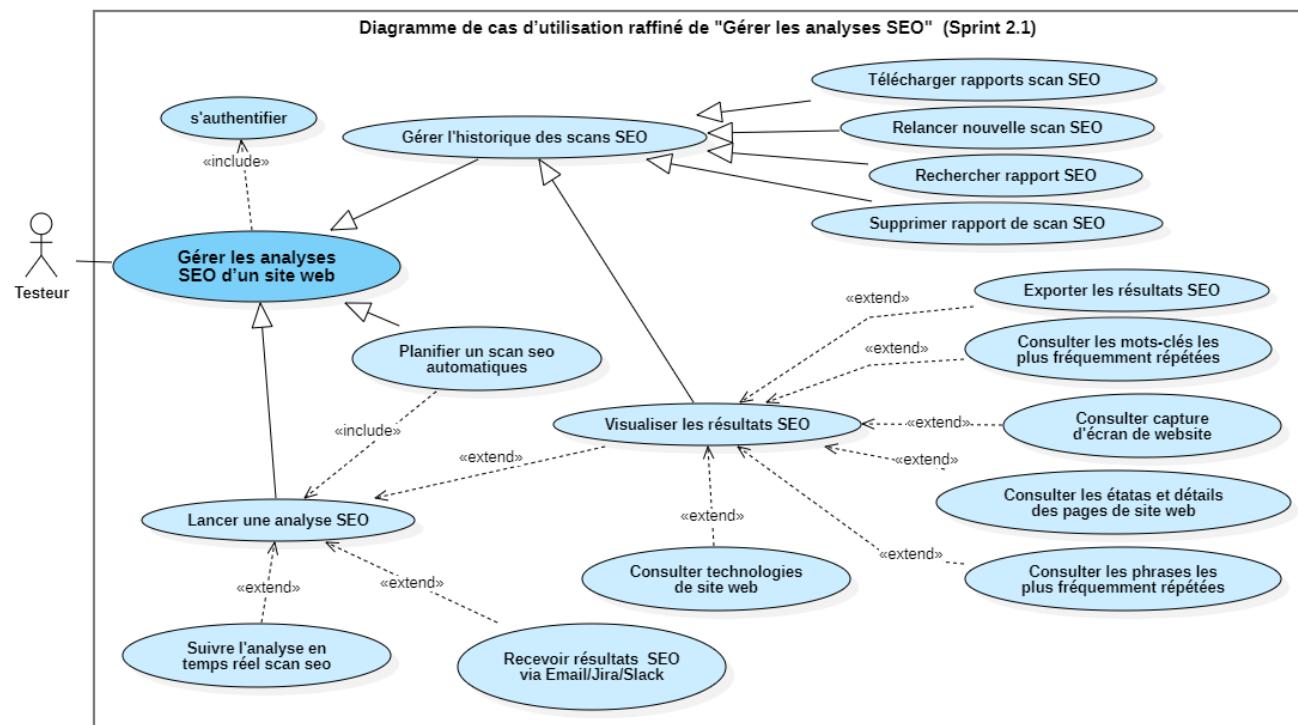


FIGURE 4.1 – Diagramme de cas d'utilisation raffiné «Gestion des analyses SEO»

2.2.2. Diagramme de cas d'utilisation raffiné : Gérer les analyses fonctionnelles

La figure 4.2 présente les cas d'utilisation spécifiques à la gestion des tests fonctionnels, tels que le paramétrage des scénarios, le déclenchement des tests, ainsi que la gestion des résultats associés.

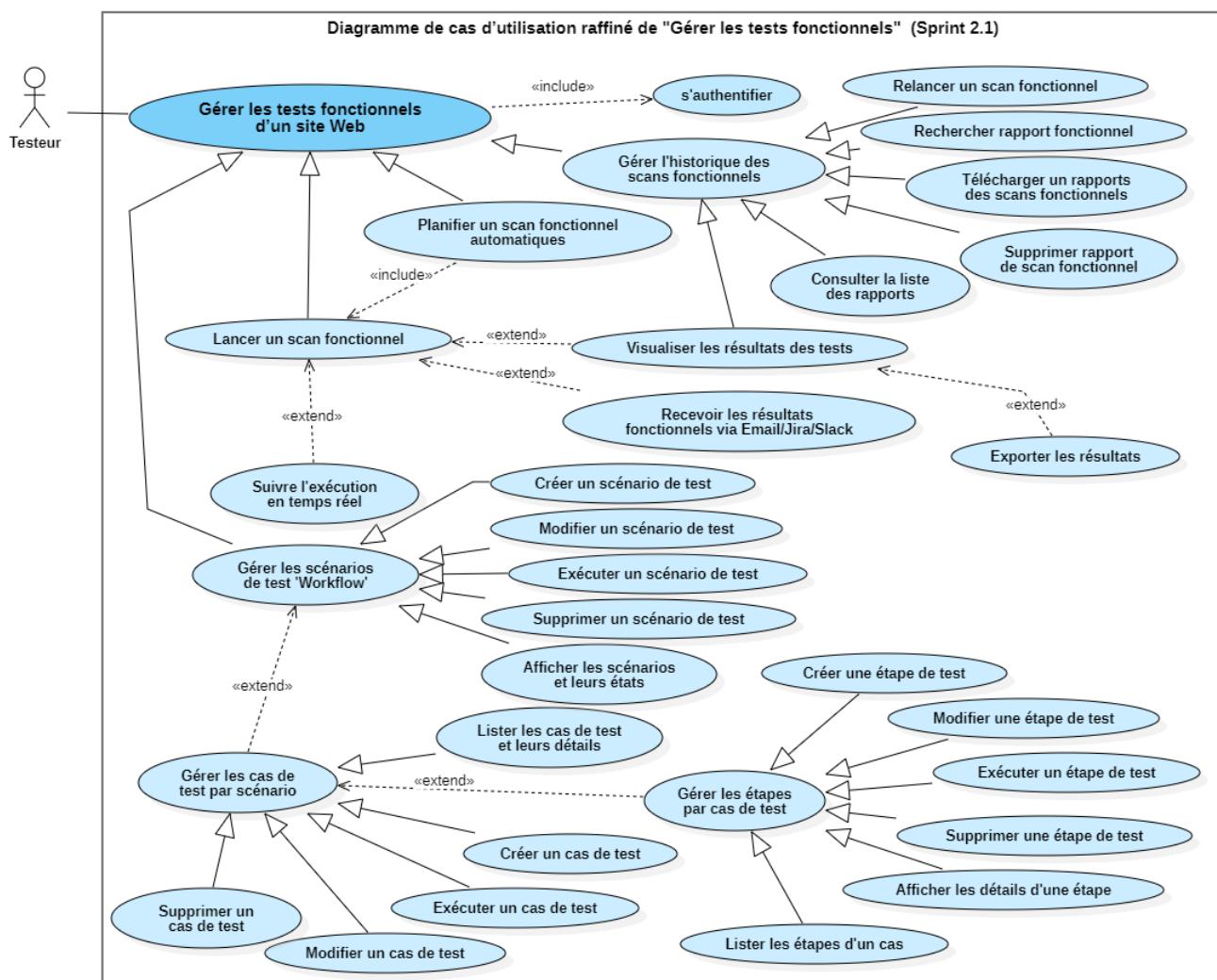


FIGURE 4.2 – Diagramme de cas d'utilisation raffiné «Gestion des analyses fonctionnelles»

2.3. Raffinement des cas d'utilisation

Ce raffinement clarifie les étapes et interactions clés pour la mise en œuvre du cas d'utilisation.

2.3.1. Description textuelle du cas d'utilisation « Lancer un scan fonctionnel »

Le tableau 4.2 présente la description textuelle du cas d'utilisation "Lancer un scan fonctionnel".

TABLE 4.2 – Description textuelle du cas d'utilisation : Lancer un scan fonctionnel

Titre	Lancer un scan fonctionnel
Acteur	Testeur
Résumé	Ce cas d'utilisation permet au testeur de déclencher l'exécution automatisée d'un ensemble de scénarios de test fonctionnels sur une application web afin de valider le comportement de l'application avec un suivi en temps réel.
Pré-conditions	<ul style="list-style-type: none"> — L'utilisateur est authentifié en tant que Testeur — Au moins un scénario de test existe dans le système

Post-conditions	<p>En cas de succès :</p> <ul style="list-style-type: none"> • Le scan fonctionnel est lancé avec succès avec un identifiant unique de scan est généré • Les notifications sont envoyées selon la configuration • L'historique est mis à jour <p>En cas d'échec :</p> <ul style="list-style-type: none"> • Le scan n'est pas lancé et un message d'erreur explicite est affiché • Aucune notification n'est envoyée
Scénario nominal	<ol style="list-style-type: none"> 1. Le testeur accède à l'interface de lancement de scan 2. Le système affiche les scénarios disponibles 3. Le testeur sélectionne les scénarios à exécuter. 4. Le testeur confirme le lancement 5. Le système génère un ID unique, crée une session d'exécution et met à jour le statut à "En cours" 6. Le système envoie les notifications de début 7. Le système lance l'exécution des tests en arrière-plan 8. Le système active le suivi en temps réel 9. Si l'envoi des rapports est configuré, le système crée automatiquement des tickets Jira ou envoie un message via Slack ou par email, selon le type de configuration.
Scénario d'erreur	<p>— Étape 3 (Aucun scénario sélectionné) :</p> <ul style="list-style-type: none"> ✗ Le système affiche un message d'erreur ✗ Le système invite le testeur à sélectionner au moins un scénario ✗ Retour à l'étape 3

2.3.2. Description textuelle du cas d'utilisation «Lancer une analyse SEO»

Le tableau 4.3 présente la description textuelle du cas d'utilisation "Lancer une analyse SEO".

TABLE 4.3 – Description textuelle du cas d'utilisation : Lancer une analyse SEO

Titre	Lancer une analyse SEO
Acteur	Testeur
Résumé	Ce cas d'utilisation permet au testeur de lancer une analyse d'un site web afin d'évaluer sa qualité SEO à travers différents indicateurs techniques et de contenu.
Pré-conditions	Le testeur doit avoir accès à l'interface de scan et l'URL du site cible doit être valide et accessible publiquement.

Post-conditions	Un rapport SEO est généré, incluant les résultats de l'analyse (balises, mots clés, performances...) et un score global est calculé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le testeur accède à l'interface d'analyse SEO. 2. Il saisit ou colle l'URL du site à analyser. 3. Il lance l'analyse en cliquant sur le bouton dédié. 4. Le système récupère les données du site (HTML, métadonnées, performances...). 5. Le système effectue les vérifications SEO : balises manquantes, densité des mots clés, temps de chargement, structure... 6. Le système calcule un score global basé sur les critères définis. 7. Un rapport d'analyse détaillé est généré et affiché à l'écran.
Scénario d'erreur	<ul style="list-style-type: none"> • Étape 2 (URL invalide) : <ul style="list-style-type: none"> ✗ Le système affiche un message d'erreur indiquant que l'URL saisie est invalide. • Étape 4 (Échec de récupération de données) : <ul style="list-style-type: none"> ✗ Le système affiche une alerte mentionnant l'échec de connexion au site ou un temps d'attente dépassé. • Étape 5 (Erreur d'analyse) : <ul style="list-style-type: none"> ✗ Si une erreur technique survient lors du parsing SEO, une notification avec détail de l'erreur est proposée au testeur.

2.4. Conception du sprint 2.1

Dans cette section, nous présentons la conception détaillée des fonctionnalités développées au cours du sprint, à travers des diagrammes de classes et de séquences.

2.4.1. Diagramme de classe du Sprint 2.1

Ce diagramme vient enrichir l'architecture définie au sprint précédent, en intégrant la gestion des tests fonctionnels automatisés et les analyses SEO avancées.

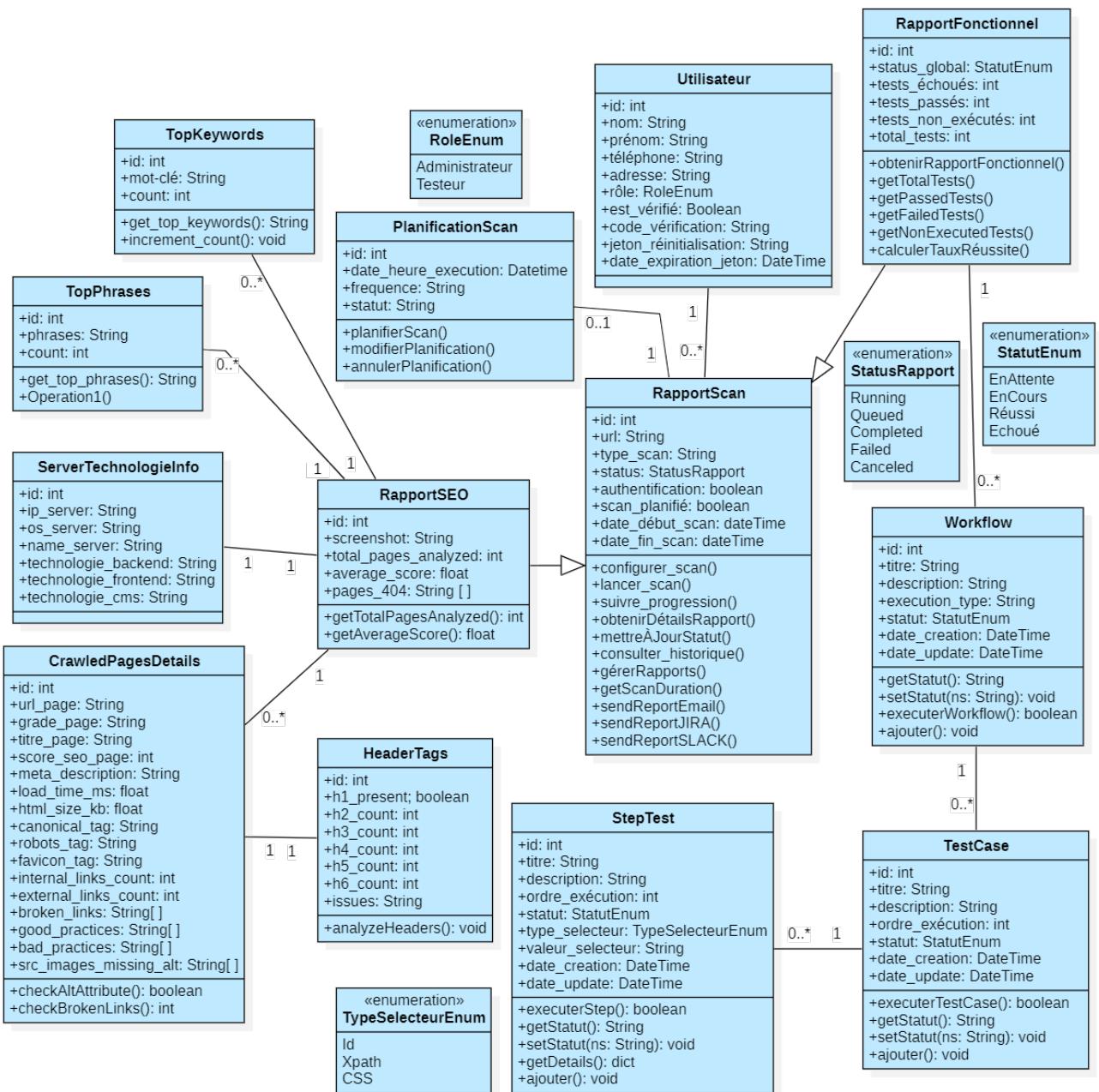


FIGURE 4.3 – Diagramme de classe du sprint 2.1

Les principales classes modélisées du sprint 2.1 sont les suivantes :

- * **Utilisateur** : Classe représentant les utilisateurs du système.
- * **PlanificationScan** : Gère la planification des scans avec des fonctionnalités de programmation, modification et annulation des analyses automatisées.
- * **RapportScan** : Classe principale pour les rapports de scan incluant la configuration, le lancement, le suivi de progression et la gestion des résultats d'analyse.
- * **RapportSEO** : Spécialise les rapports pour l'analyse SEO avec des métriques spécifiques comme le nombre total de pages analysées et le score moyen.
- * **RapportFonctionnel** : Représente les résultats des tests fonctionnels réalisés avec des

informations sur le statut global, les différents tests effectués et leur réussite ou échec.

- * **TopKeywords** : Contient les mots-clés les plus importants extraits lors des analyses SEO, utilisés pour le suivi des performances et des tendances.
- * **TopPhrases** : Contient les phrases clés les plus pertinentes issues de l'analyse SEO, servant à enrichir les rapports et améliorer le référencement.
- * **ServerTechnologieInfo** : Fournit des détails sur les technologies serveur détectées, telles que les systèmes d'exploitation, CMS, frameworks front-end et back-end utilisés.
- * **CrawledPagesDetails** : Représente les informations détaillées des pages web crawlées durant les scans(URL, score SEO associé, métriques de performance...).
- * **HeaderTags** : Regroupe les métadonnées extraites des pages web, comme les balises Hn, les descriptions meta et d'autres éléments influençant le référencement naturel.
- * **Workflow** : Modélise la chaîne ou séquence d'exécution des tests automatisés, incluant la gestion du statut et des étapes du processus.
- * **TestCase** : Définit un cas de test fonctionnel détaillé, incluant la description, l'ordre d'exécution et les critères de réussite ou d'échec.
- * **StepTest** : Représente une étape précise dans un scénario de test fonctionnel, avec un suivi du statut et des résultats partiels.
- * **StatutEnum (énumération)** : Définit les différents états possibles d'un test ou rapport (EnAttente, EnCours, Réussi, Échoué).
- * **StatusRapport (énumération)** : Spécifie les états des rapports de scan (Running, Queued, Completed, Failed, Canceled).
- * **RoleEnum (énumération)** : Définit les rôles des utilisateurs (Administrateur, Testeur).
- * **TypeSelecteurEnum (énumération)** : Définit les types de sélecteurs utilisés dans les tests (Id, Xpath, CSS).

Les associations entre classes dans Sprint 2.1 incluent :

- Un **Utilisateur** peut avoir plusieurs **PlanificationScan** et **RapportScan** associés.
- Un **RapportFonctionnel** centralise plusieurs **Workflow**, chacun structurant l'exécution de plusieurs **TestCase**, eux-mêmes composés de plusieurs **StepTest**.
- Les **TopKeywords**, **TopPhrases**, **ServerTechnologieInfo** et **CrawledPagesDetails** sont liés à un **RapportSEO**.
- Les **HeaderTags** sont extraits pour chaque **CrawledPagesDetails**.

Cette modélisation structure le traitement et le suivi des tests fonctionnels et SEO, permettant une meilleure organisation et automatisation des analyses au sein du projet.

2.4.2. Diagramme de séquence de conception du cas « Crée un scénario de test »

La figure 4.4 présente le déroulement de la création d'un scénario de test fonctionnel (**Workflow**), initiée par l'utilisateur via l'interface. Après saisie des détails (nom, description), le **Workflow** est créé et sauvegardé en base pour exécution ou modification ultérieure.

L'utilisateur est ensuite redirigé vers une interface dédiée à la configuration du scénario, permettant l'ajout de **TestCase** et **StepTest**, ainsi que le lancement direct du workflow ou de ses tests. Cette interface garantit une gestion interactive et complète du cycle de vie des scénarios.

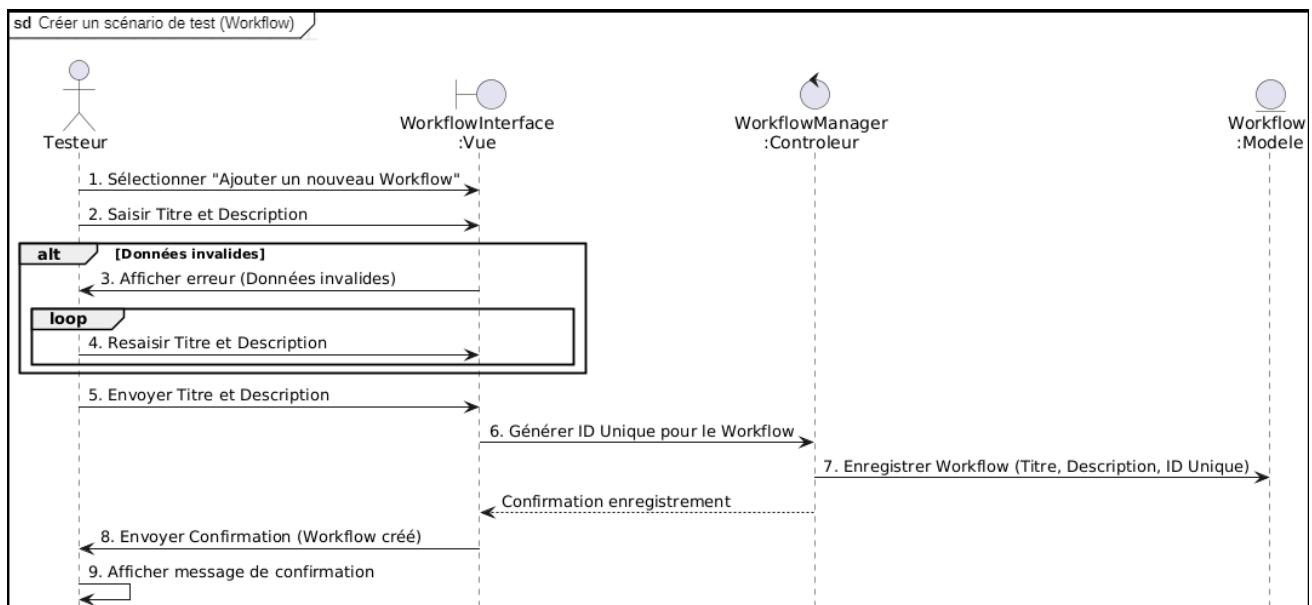


FIGURE 4.4 – Diagramme de séquence de conception du cas « Crée un scénario de test »

2.5. Réalisation du sprint 2.1

Dans cette section, nous présentons les principales interfaces développées durant ce sprint 2.1, en commençant par la gestion des tests fonctionnels, puis les interfaces liées à l'analyse SEO.

- **Interface unifiée de gestion des analyses SEO :** La figure 4.5¹ présente une interface complète regroupant la visualisation des résultats SEO par catégories (contenu, technique, performance), l'historique des rapports avec options d'export, l'identification des technologies et mots-clés SEO, ainsi que l'aperçu des captures d'écran des pages analysées.

1. Voir annexe E : Figure 4.5

FIGURE 4.5 – Interface unifiée de gestion des analyses SEO

- **Formulaire de lancement d'une analyse fonctionnelle :** La figure E.42² présente un formulaire pour saisir l'URL du site à tester, avec la possibilité d'ajouter des informations d'authentification (utilisateur, mot de passe, jeton, cookies) si nécessaire.
- **Interface d'historique des rapports fonctionnels :** La figure 4.6³ illustre une interface récapitulative des campagnes de tests fonctionnels, offrant un accès centralisé à l'historique

2. Voir annexe E : Figure E.42
3. Voir annexe E : Figure 4.6

des rapports, aux statistiques globales des résultats (tests réussis, échoués ou en attente), ainsi qu'à une navigation détaillée au sein des workflows exécutés, de leurs cas de test, des étapes associées et des erreurs rencontrées. L'interface permet également la création de nouveaux workflows afin de planifier et structurer de nouveaux scénarios de test.

The screenshot shows a web-based application for managing functional test scans. At the top, there's a header with a user profile for 'Rihab Cherni tester'. Below the header, a search bar and a 'New Scan' button are visible. The main area displays three completed functional scans:

- Scan 1:** URL: <http://testphp.vulnweb.com>, Start: 6/19/25, 5:33 PM. Status: Passed.
- Scan 2:** URL: <https://get.adobe.com/>, Start: 6/17/25, 10:20 PM. Status: Pending.
- Scan 3:** URL: <https://dribbble.com/search/ecommerce>, Start: 6/17/25, 8:27 AM. Status: Failed.

Below the scans, a section titled 'Functional Scan Details' provides an overview of the total tests (4), passed tests (3), failed tests (1), and pending tests (0). A progress bar indicates 4/4 completed. The 'Test Progress' section shows a horizontal bar with a blue gradient.

The 'Workflows' section contains two entries:

- Workflow 1 - User Authentication and Profile Management:** Status: Passed. It includes a sub-section for '#2 User Login' with three steps: '2. Enter Password' (passed), '3. Click Login' (passed), and '1. Enter Username' (passed).
- Workflow 2 - Classic User Journey:** Status: Failed. It includes a sub-section for '#1 Homepage Accessibility' (passed) and '#2 Product Search' (failed).

At the bottom, there are pagination controls for items per page (5) and a page indicator (1 - 3 of 3).

FIGURE 4.6 – Interface d'historique des rapports fonctionnels

- **Interface de création et de gestion des workflows de tests fonctionnels :** La figure 4.7⁴ illustre une interface graphique interactive dédiée à la construction et à la gestion des scénarios de test fonctionnel. Elle permet de créer, modifier et exécuter visuellement des tests automatisés, avec une gestion détaillée des cas de test et de leurs étapes. Chaque

4. Voir annexe E : Figure 4.7

étape est définie par une action, une cible, une valeur attendue et un statut. L’interface intègre un éditeur de type Drawflow pour une visualisation intuitive des workflows.

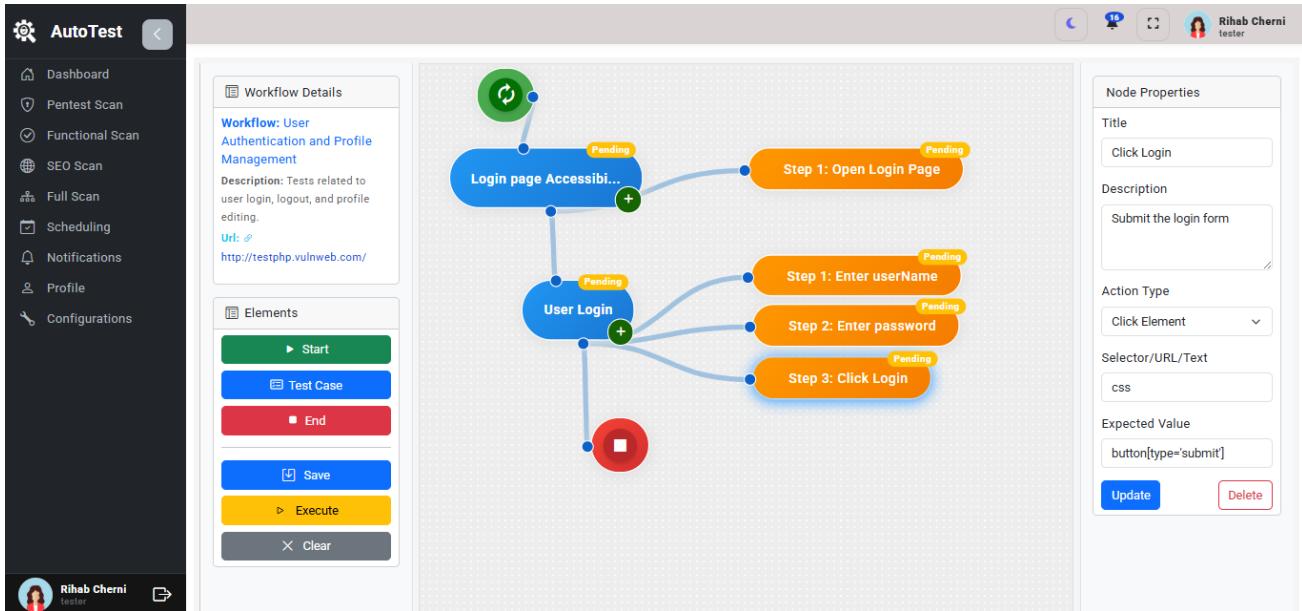


FIGURE 4.7 – Interface de gestion des scénarios de tests fonctionnels

- **Interface de planification des scans fonctionnels** : La figure E.41⁵ présente une interface unifiée de planification, permettant de programmer automatiquement les scans fonctionnels, de sécurité et SEO via un calendrier centralisé et personnalisable.

3. Sprint 2.2 : Finalisation et dockerisation de l’application

Ce sprint a permis de finaliser les fonctionnalités avancées et d’assurer la mise en production :

- **Gestion des scans multiples** : intégration des scans fonctionnels, sécurité et SEO dans une interface unifiée.
- **Visualisation des statistiques via le tableau de bord** : développement du tableau de bord pour un suivi clair des analyses.
- **Gestion des rapports des analyses effectuées** : mise en place des fonctionnalités de consultation et export des rapports.
- **Dockerisation de l’application** : conteneurisation des composants backend et frontend à l’aide de Docker pour faciliter l’exécution, la portabilité et l’orchestration de l’application.

Ce sprint a permis d’achever le projet en offrant une application complète et opérationnelle.

3.1. Backlog du sprint 2.2

Dans cette section, nous présenterons le Backlog du sprint 2.2, illustré dans le tableau 4.4, en tenant compte des modifications et ajustements apportés depuis le sprint précédent.

5. Voir annexe E : Figure E.41

TABLE 4.4 – Backlog du sprint 2.2

ID US	User Story	Description	ID tâche	Tâches	Priorité	Risques	Estimation(j)
EPIC 7 : Gestion des analyses complètes (fonctionnels, sécurité, SEO)							
7.1	Lancer en parallèle les scans fonctionnels, de sécurité et SEO selon le choix de l'utilisateur.	En tant que testeur, je dois pouvoir exécuter simultanément des tests fonctionnels, de sécurité et SEO pour gagner du temps et obtenir une analyse complète.	7.1.A 7.1.B	- Créer un interface et implémenter une logique permettant de choisir un ou plusieurs types de tests à exécuter (fonctionnel, sécurité, SEO) simultanément. - Centraliser les résultats et générer un rapport unifié.	Élevée	Élevée	2
7.2	Consulter les résultats des différents types d'analyses de manière consolidée.	En tant que testeur, je souhaite consulter de manière centralisée les résultats fonctionnels, de sécurité et SEO afin de mieux comprendre les impacts croisés.	7.2.A 7.2.B	- Concevoir une interface unifiée pour la visualisation des résultats. - Regrouper les vulnérabilités, erreurs fonctionnelles et recommandations SEO dans une vue consolidée.	Élevée	Moyenne	2
7.3	Gérer les rapports d'analyse (suppression, export, recherche, relancement).	En tant qu'administrateur, je veux pouvoir rechercher, supprimer, exporter ou relancer un scan en utilisant la configuration d'un rapport existant pour faciliter la gestion des résultats.	7.3.A 7.3.B 7.3.C 7.3.D	- Ajouter la possibilité de rechercher des rapports par mots-clés, type de test ou date. - Permettre la suppression manuelle ou automatique des rapports. - Intégrer une fonction d'export (PDF/JSON...). - Offrir la possibilité de relancer un scan avec les paramètres d'un ancien rapport.	Moyenne	Moyenne	2
EPIC 8 : Visualisation des statistiques via le tableau de bord							
8.1	Visualiser des statistiques personnalisées des scans via le tableau de bord.	En tant que testeur, je souhaite suivre l'évolution de mes tests à travers un tableau de bord pour faciliter l'analyse.	8.1.A 8.1.B	- Concevoir un tableau de bord interactif permettant d'afficher les résultats des scans. - Représenter les statistiques personnelles sous forme graphique avec des filtres.	Élevée	Moyenne	4

8.2	Visualiser les statistiques globales via le tableau de bord administrateur.	En tant qu'administrateur, je souhaite disposer d'un tableau de bord centralisé pour superviser l'activité des utilisateurs, des scans...	8.2.A 8.2.B	- Créer une interface d'administration affichant les statistiques globales. - Ajouter des filtres dynamiques par période, gravité, type d'analyse, avec des représentations graphiques.	Moyenne	Moyenne	4
EPIC 12 : Gestion des rapports des analyses effectuées							
12.1	Consulter tous les rapports générés.	En tant qu'administrateur, je souhaite accéder à tous les rapports générés.	12.1.A 12.1.B	- Créer une interface d'affichage de rapports filtrables par type, date, utilisateur. - Ajouter une fonction de recherche.	Moyenne	Basse	1
12.2	Télécharger et supprimer les rapports.	En tant qu'administrateur, je souhaite pouvoir télécharger ou supprimer les rapports.	12.2.A 12.2.B	- Ajouter les options de téléchargement dans différents formats (HTML, JSON, PDF...). - Implémenter la suppression sécurisée des rapports obsolètes.	Moyenne	Moyenne	2
EPIC 13 : Dockerisation de l'application							
13.1	Conteneuriser les composants de l'application.	En tant que développeur, je souhaite conteneuriser les différents modules de l'application afin d'assurer la portabilité, l'homogénéité des environnements et la facilité d'exécution sur toute machine.	13.1.A 13.1.B 13.1.C	- Conteneuriser les services principaux : PostgreSQL, backend FastAPI, frontend Angular, RabbitMQ. - Intégrer les outils de sécurité dans des conteneurs distincts pour garantir leur isolement et réutilisabilité. - Rédiger un fichier 'Dockerfile' adapté pour chaque composant avec les dépendances nécessaires.	Élevée	Moyenne	2
13.2	Orchestration des conteneurs avec Docker Compose.	En tant que développeur, je souhaite orchestrer tous les conteneurs à l'aide de Docker Compose pour automatiser leur configuration, leurs connexions réseau et leur démarrage coordonné.	13.2.A 13.2.B 13.2.C	- Rédiger un fichier 'docker-compose.yml' regroupant l'ensemble des services. - Définir les volumes, réseaux, variables d'environnement et dépendances inter-conteneurs. - Tester l'environnement complet avec 'docker-compose up' pour s'assurer de la bonne communication entre les services.	Moyenne	Moyenne	3

۲۵

3.2. Analyse du sprint 2.2

Dans cette section, nous analysons les besoins fonctionnels couverts durant le sprint 2.2.

3.2.1. Diagramme de cas d'utilisation du sprint 2.2

Cette section présente le diagramme de cas d'utilisation élaboré pour le sprint 2.2, illustré dans la figure 4.8. Il met en évidence les différentes interactions entre les utilisateurs et le système, notamment l'exécution simultanée des analyses (fonctionnelles, de sécurité et SEO), la gestion des rapports par l'administrateur, ainsi que la visualisation des statistiques via le tableau de bord.

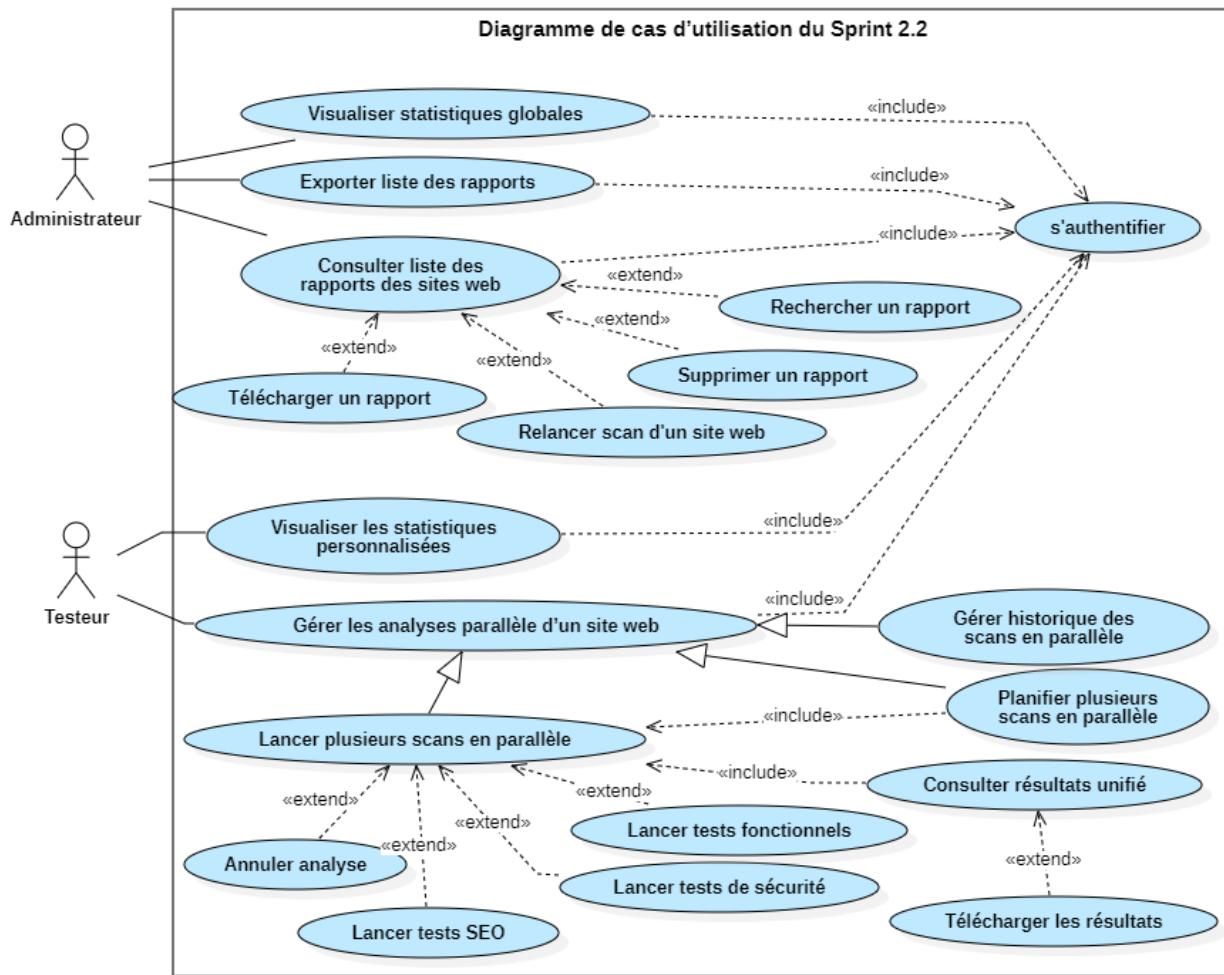


FIGURE 4.8 – Diagramme de cas d'utilisation du sprint 2.2

3.3. Conception du sprint 2.2

Dans cette section, nous détaillons la conception des fonctionnalités développées au cours du sprint 2.2.

3.3.1. Diagramme de classe du Sprint 2.2

Dans ce sprint, aucun diagramme de classes n'est présenté, car ils ont tous été traités dans les sprints précédents. La plupart des classes sont en effet réutilisées pour implémenter les fonctionnalités liées à la création des dashboards, au lancement de scans multiples et à la gestion des rapports. Pour plus de détails sur ces classes, se référer aux figures 3.3, 3.8 et 4.3.

3.3.2. Diagramme de déploiement Docker

La figure 4.9 illustre le diagramme de déploiement de notre application, mettant en évidence l'interaction entre l'utilisateur, le navigateur web, le frontend (Angular), le backend (FastAPI et WebSocket), le message broker RabbitMQ, la base de données PostgreSQL, ainsi que divers outils de scan pour l'analyse de sécurité et Selenium. Ce diagramme présente les différents conteneurs, leurs ports exposés, ainsi que les interactions et communications entre eux afin de garantir un fonctionnement cohérent.

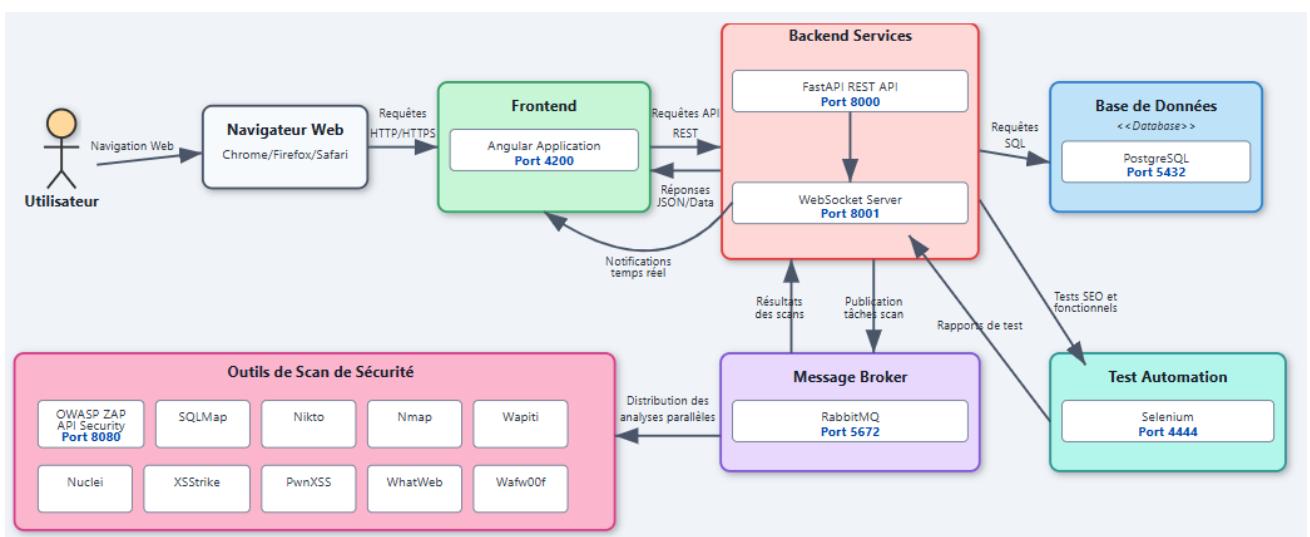


FIGURE 4.9 – Diagramme de déploiement

3.4. Réalisation du sprint 2.2

Dans cette section, nous présentons les principales interfaces développées durant cette sprint 2.2, en commençant par celles liées à l'exécution parallèle des tests, la gestion des rapports, puis la visualisation des statistiques.

- **Interface de gestion des rapports d'analyse** : Comme illustré dans la figure 4.10, cette interface permet à l'administrateur de rechercher, supprimer, exporter les rapports. Les filtres intégrés (type, date, utilisateur) rendent la gestion des rapports plus fluide et performante.

User	URL	Type	Status	Schedule	Started-At	Finished-At	Actions
AC Arij Cherni	http://testphp.vulnweb.com	security	Running	Instant	6/15/25, 6:38 PM		Delete
AC Arij Cherni	http://testphp.vulnweb.com	security	Running	Instant	6/14/25, 9:55 PM		Delete
Rihab Cherni	test.com	functional	Running	Scheduled	6/17/25, 8:27 AM		Delete
Rihab Cherni	https://www.cisco.com/	security	Completed	Instant	6/15/25, 7:04 PM	6/15/25, 8:05 PM	Delete
Rihab Cherni	https://get.adobe.com/	functional	Running	Instant	6/17/25, 10:20 PM		Delete
Rihab Cherni	http://testphp.vulnweb.com/	seo	Completed	Instant	6/16/25, 1:19 PM	6/16/25, 1:19 PM	Delete

FIGURE 4.10 – Interface de gestion des rapports d’analyse

- **Tableau de bord administrateur** : Pour le profil administrateur, la figure 4.11 met en évidence les indicateurs globaux relatifs à l’activité de la plateforme, tels que le nombre de scans effectués, le nombre d’utilisateurs et la répartition par type d’analyse. Des filtres par période ou par type d’analyse sont également disponibles pour affiner l’affichage.

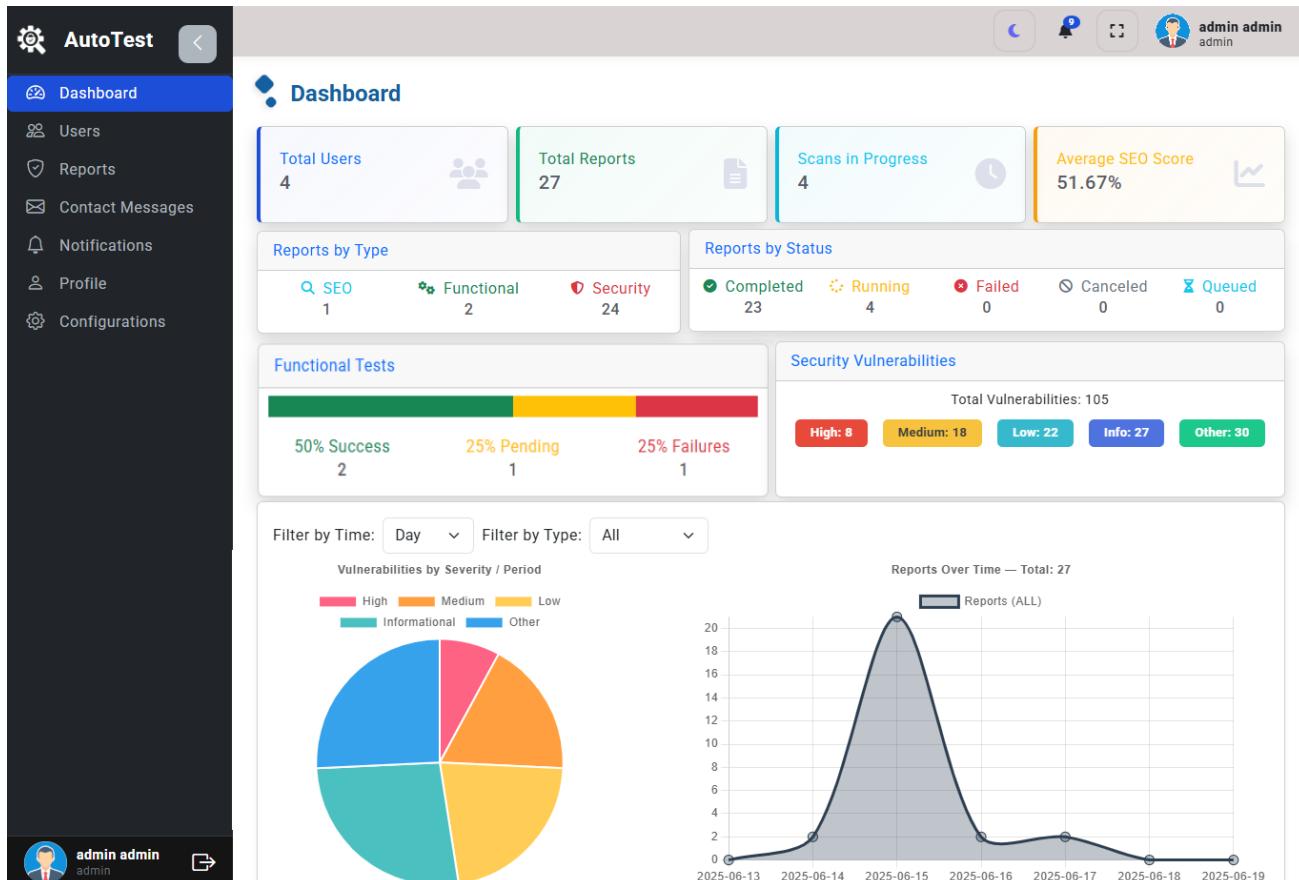


FIGURE 4.11 – Interface du tableau de bord d’administrateur

- **Tableau de bord utilisateur** : La figure 4.12 présente le tableau de bord dédié aux testeurs. Il affiche des statistiques filtrables et interactives sur les campagnes de test, à travers des graphiques dynamiques.

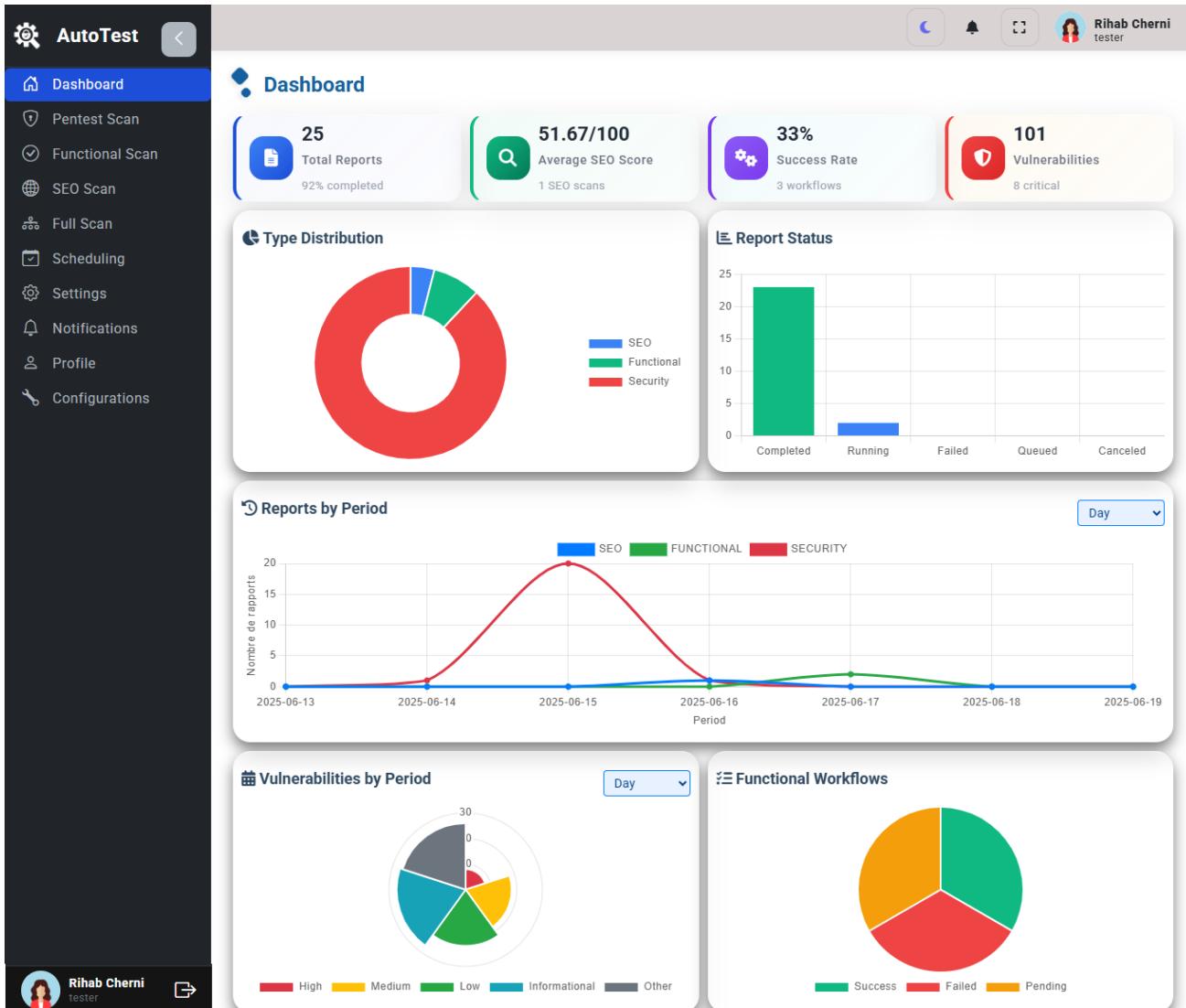


FIGURE 4.12 – Interface du tableau de bord testeur

- **Interface unifiée de lancement des analyses** : La figure E.44⁶ présente un formulaire centralisé permettant de lancer un ou plusieurs types de tests (fonctionnels, sécurité, SEO) sur une URL cible. L’utilisateur peut également configurer une authentification personnalisée (via identifiants, jeton ou cookies) pour les sites protégés, garantissant une exécution adaptée selon le contexte de la cible.

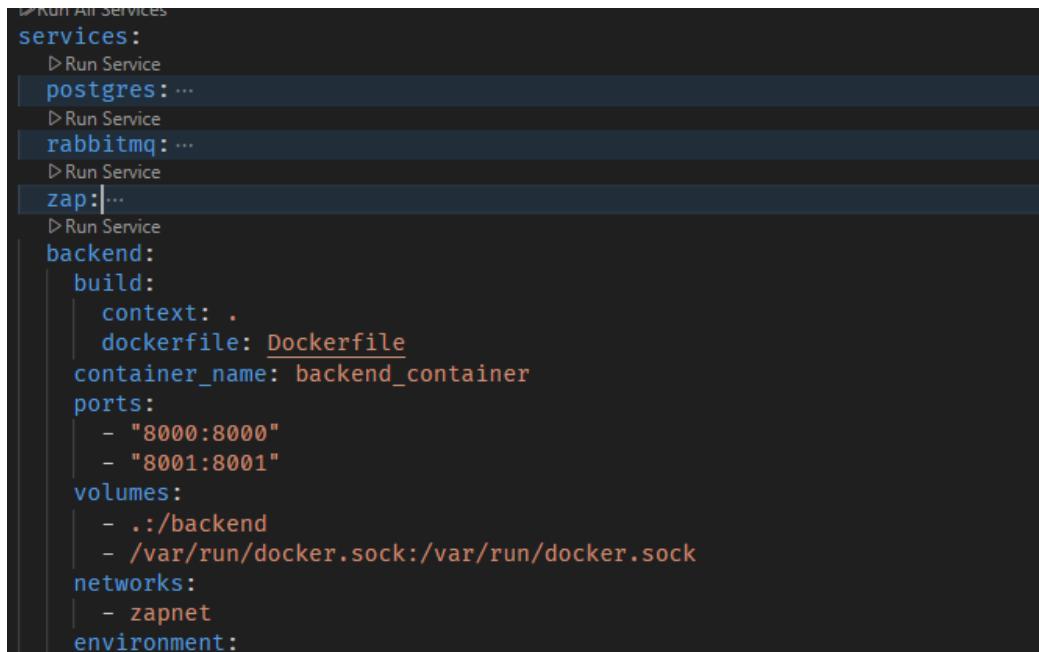
3.5. Dockerisation de l’application

Dans le cadre du sprint 2.2, la dockerisation complète de l’application a constitué une étape clé pour assurer sa portabilité, sa simplicité d’exécution et sa maîtrise en environnement de

6. Voir annexe E : Figure E.44

production. L'ensemble des composants comme backend FastAPI, frontend Angular, base de données PostgreSQL, ainsi que les outils tiers de scan ont été conteneurisés et orchestrés à l'aide d'un fichier `docker-compose.yml`. Cette orchestration garantit une exécution cohérente et reproductible, facilitant les phases de test, de déploiement et de maintenance.

La figure 4.13 présente une capture d'écran du fichier `docker-compose.yml`.



```
▶ Run All Services
services:
  ▶ Run Service
  postgres: ...
  ▶ Run Service
  rabbitmq: ...
  ▶ Run Service
  zap: |...
    ▶ Run Service
  backend:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: backend_container
    ports:
      - "8000:8000"
      - "8001:8001"
    volumes:
      - .:/backend
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - zapnet
    environment:
```

FIGURE 4.13 – Contenu du fichier `docker-compose.yml`

Cette dockerisation rend l'application facilement exécutable sur toute machine supportant Docker, simplifiant ainsi la phase de mise en production et les évolutions futures.

Conclusion

Au terme de cette deuxième release, nous avons achevé la dernière phase de développement de notre application. Ce résultat a été atteint grâce à une démarche structurée, allant de l'analyse à la conception, puis à l'implémentation et à la réalisation finale.

Conclusion générale

Ce projet de fin d'études nous a permis de répondre à un besoin croissant dans le domaine de la cybersécurité et de la qualité web : l'automatisation des tests de sécurité, des tests fonctionnels et des audits SEO. En adoptant une démarche agile (Scrum) et en utilisant des technologies modernes comme Angular, FastAPI et PostgreSQL, nous avons conçu une application web performante, évolutive et maintenable.

Ce travail nous a permis de mieux appréhender les enjeux liés à la sécurité et à la qualité des applications web, notamment à travers la détection automatisée de vulnérabilités, la validation des faux positifs et l'amélioration de la visibilité sur les moteurs de recherche. Les résultats obtenus ont confirmé la pertinence de l'approche, avec des tests fiables, reproductibles et des rapports clairs.

Au-delà des aspects techniques, ce projet a renforcé nos compétences en développement full-stack, en sécurité informatique, en gestion agile et en automatisation des tests. Il constitue une étape clé dans notre parcours d'ingénieur, ouvrant la voie à des évolutions futures.

Parmi les pistes d'amélioration envisagées :

- **Intégration de l'intelligence artificielle** : pour optimiser l'analyse des résultats, détecter les vulnérabilités, anticiper les failles et générer automatiquement des scénarios de tests.
- **Extension aux tests mobiles** : prise en charge des applications mobiles (Android/iOS) afin d'évaluer leur sécurité, leur performance et leur conformité aux bonnes pratiques.
- **Extension des audits SEO** : avec des critères avancés comme l'expérience utilisateur ou la performance mobile.
- **Support des environnements cloud** : exécution des tests sur des infrastructures distribuées, intégration avec les pipelines CI/CD.
- **Support multilingue** : pour rendre l'application accessible à un public international.

En somme, ce projet constitue une base solide pour le développement d'une solution complète, adaptable et intégrée dans des processus de développement sécurisés à grande échelle.

Annexe A : Limites de PENTRA

Cette annexe présente les interfaces et la structure de base de données de la première version de l'application **PENTRA**. Cette version initiale comporte plusieurs limitations, tant en termes d'expérience utilisateur que d'architecture technique.

1. Structure initiale de la base de données :

Le schéma de la figure A.1 montre une base de données.

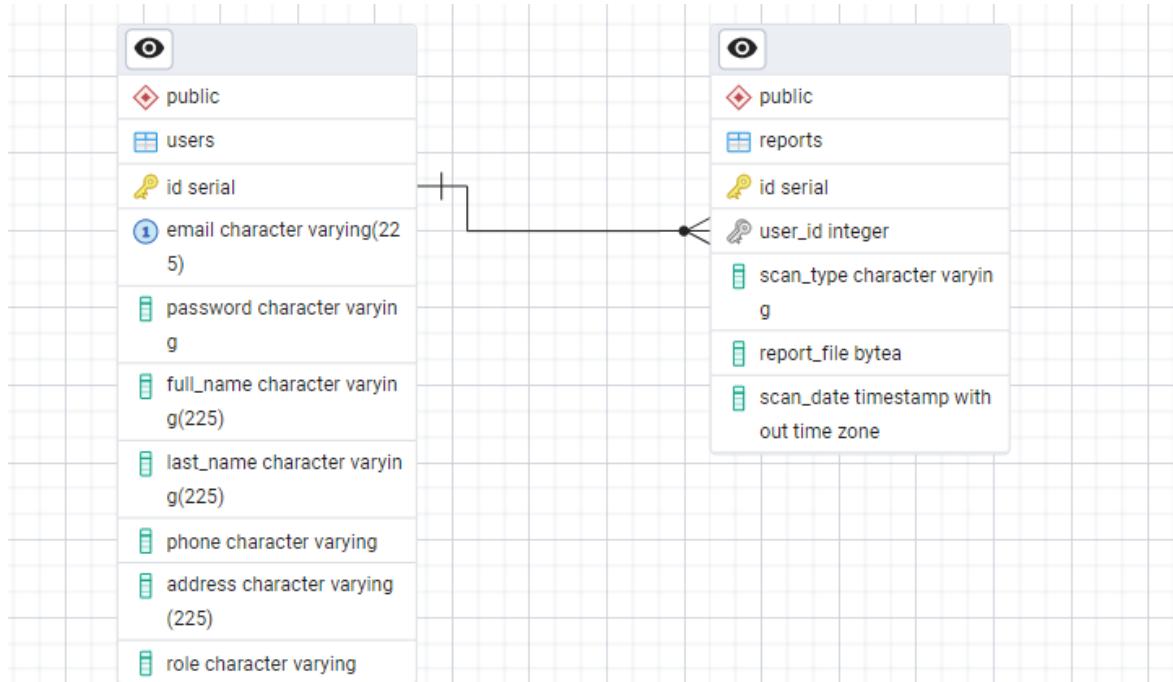


FIGURE A.1 – Structure initiale de la base de données de l'application PENTRA

2. Interfaces de la version initiale :

Les interfaces de cette version présentent certaines limites tant sur le plan ergonomique que fonctionnel. Les figures suivantes illustrent les principales fonctionnalités de l'application.

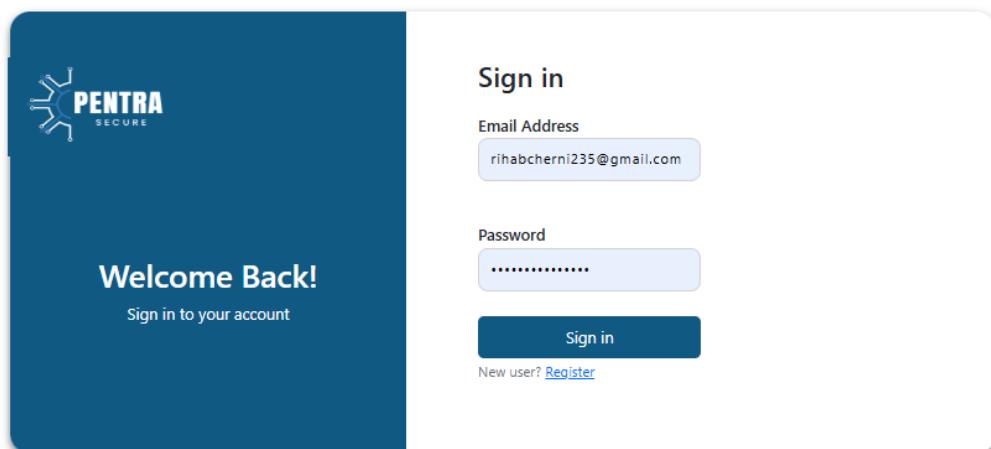


FIGURE A.2 – Interface de connexion (PENTRA)

[I already have an account](#)

Sign up

Full Name

Last Name

Email Address Please enter a valid email address

Password Password is required

Address

Phone

By signing up, you agree to our [Terms of services](#) and [Privacy Policy](#)

[Create Account](#)

FIGURE A.3 – Interface d’inscription (PENTRA)

Information	
Email	Phone
rhabcherni235@gmail.co m	23710078
last_name	address
Cherni	26 Rue Ibn Sina

FIGURE A.4 – Interface de profil utilisateur (PENTRA)

Use our free website to scan URLs in real-time and ensure your websites are safe.

Quickly detect XSS, SQL injection, Command injection, XXE and other critical issues - automatically validated to eliminate false positives.

Enter your URL website [Scan](#)

Services

- Clean Code**
Lorem ipsum dolor sit amet consectetur adipisicing elit
- Problem Solving**
Lorem ipsum dolor sit amet consectetur adipisicing elit
- Best Domain**
Lorem ipsum dolor sit amet consectetur adipisicing elit
- Secure Website**
Lorem ipsum dolor sit amet consectetur adipisicing elit

FIGURE A.5 – Interface de lancement du scan de test de pénétration (PENTRA)

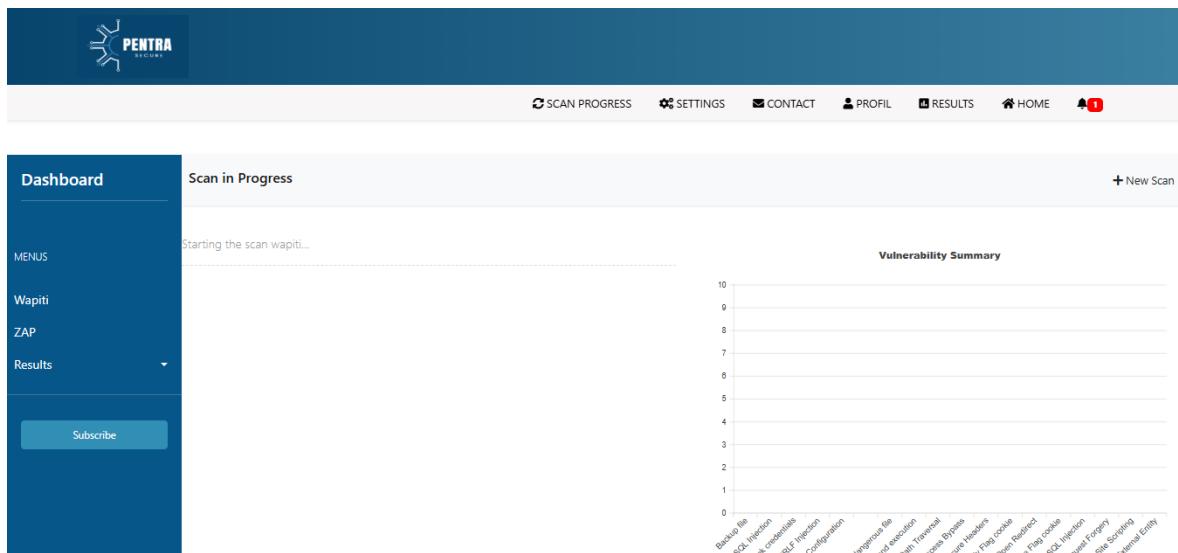


FIGURE A.6 – Interface de progression du scan avec Wapiti (PENTRA)

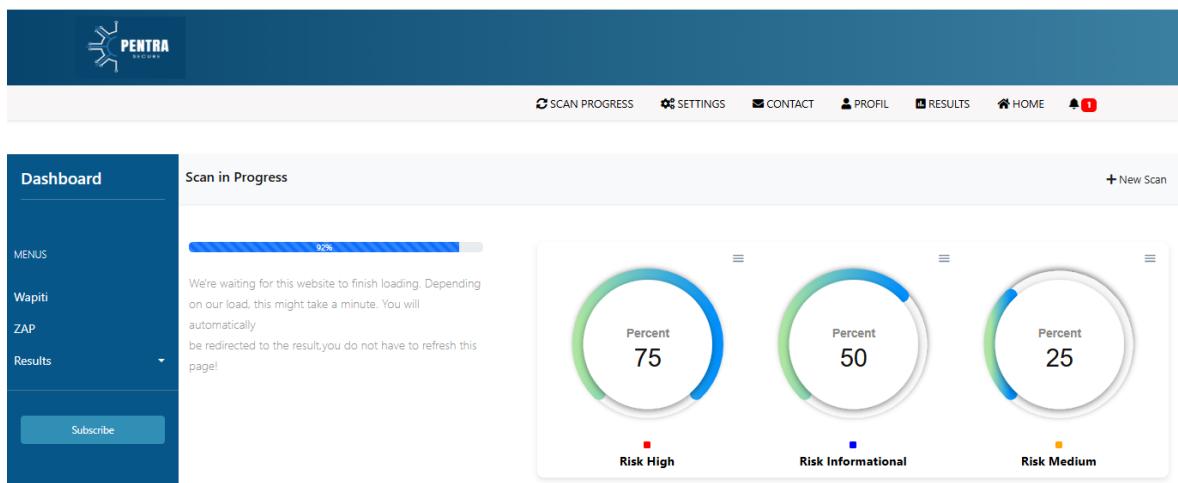


FIGURE A.7 – Interface de progression du scan avec OWASP ZAP (PENTRA)

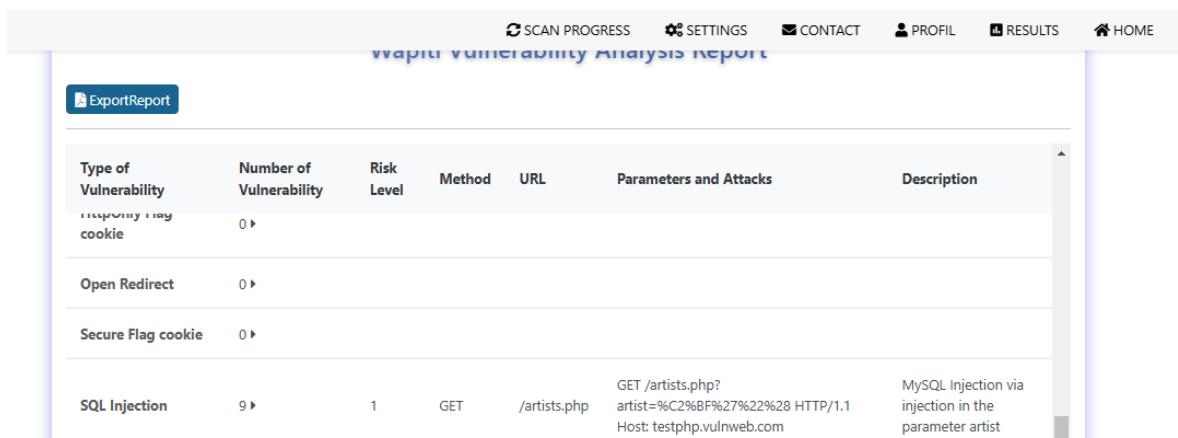


FIGURE A.8 – Interface du tableau de résultats du scan avec Wapiti (PENTRA)

Wapiti Vulnerability Analysis Report

[Export Report](#)

Type of Vulnerability	Number of Vulnerability	Risk Level	Method	URL	Parameters and Attacks	Description
Backup file	0					
Blind SQL Injection	0					
Weak credentials	0					
CRLF Injection	0					
Content Security Policy Configuration	0					
Cross Site Request Forgery	0					
Potentially dangerous file	0					
Command execution	0					
Path Traversal	0					
Httpaccess Bypass	0					
HTTP Secure Headers	0					
HttpOnly Flag cookie	0					
Open Redirect	0					
Secure Flag cookie	0					
SQL Injection	9	1	GET	/artists.php	GET /artists.php?artist=%C2%BF%27%22%28 HTTP/1.1 Host: testphp.vulnweb.com	MySQL Injection via injection in the parameter artist
Server Side Request Forgery	0					
Cross Site	0					

FIGURE A.9 – Rapport PDF des vulnérabilités détectées par Wapiti (PENTRA)

Scan Results Report for <http://testphp.vulnweb.com/listproducts.php?cat=2>

rihabcherni000@gmail.com

À moi

Traduire en français

Hello PenTesting Team,

Please find attached the report of the scan results.

Best regards,
The PenTesting Team

1 pièce jointe • Analyse effectuée par Gmail

test_wapiti_repor...

FIGURE A.10 – Rapport des vulnérabilités détectées envoyé par e-mail (PENTRA)

SETTING NOTIFICATION FORM

Configure Slack
 Configure Gmail
 Configure Jira

Slack configuration selected. Please provide the necessary details.

Slack

your_token your_id_channel

Gmail configuration selected. Please provide the necessary details.

Send

FIGURE A.11 – Interface de configuration des paramètres d'envoi des rapports (PENTRA)

Dashboard

Users Table

ID	Username	Email	Actions
1	Marwa	admin@gmail.com	
2	Rihab Chemi	rihabchemi235@gmail.com	

Items per page: 5 0 of 0 < > >>

FIGURE A.12 – Interface de gestion des utilisateurs par l'administrateur (PENTRA)

Dashboard

Reports Table

ID	Scan Type	Scan Date	Actions
533	wapiti	2025-02-24T23:13:52	
537	PwnXSS	2025-02-26T18:21:02	
588	nmap	2025-03-03T12:29:42	
534	wapiti	2025-02-24T23:16:26	
538	PwnXSS	2025-02-26T18:56:33	
589	wapiti	2025-03-03T14:41:53	
535	solmap	2025-02-24T23:36:38	

Items per page: 5 0 of 0 < > >>

FIGURE A.13 – Interface de gestion des rapports de scan par l'administrateur (PENTRA)

L'application **PENTRA** souffre de limites au niveau de la structure des données et de l'ergonomie. Une refonte complète est nécessaire pour améliorer la navigation, l'esthétique, l'intuitivité et l'exploitation des données, tout en assurant la maintenabilité et l'évolutivité du projet.

Annexe B : Complément d'analyse et de conception

Cette annexe présente plusieurs diagrammes et descriptions complémentaires relatifs aux phases d'analyse et de conception du sprint 1.1.

□ Description textuelle du cas d'utilisation "S'inscrire" :

Le tableau 4.5 présente la description textuelle du cas d'utilisation "S'inscrire".

TABLE 4.5 – Description textuelle du cas d'utilisation : S'inscrire

Titre	S'inscrire
Acteur	Visiteur
Résumé	Ce cas d'utilisation décrit le processus d'inscription permettant au visiteur de créer un compte personnel.
Pré-conditions	Le visiteur doit disposer d'un accès à Internet via un dispositif connecté (ordinateur, tablette, smartphone).
Post-conditions	Un nouveau compte utilisateur est créé et un email de confirmation contenant un code OTP est envoyé afin de vérifier l'adresse email saisie.
Scénario nominal	<ol style="list-style-type: none">1. Le visiteur accède à la page d'inscription et clique sur le bouton "Inscrire".2. Le système affiche un formulaire de saisie des informations d'inscription.3. Le visiteur remplit le formulaire avec ses informations, puis le soumet.4. Le système valide les informations fournies.5. Le système enregistre les données du visiteur dans la base de données.6. Il génère automatiquement des identifiants de connexion.7. Le système envoie un e-mail de confirmation contenant un code OTP.8. L'utilisateur saisit ce code dans un champ dédié.9. Une fois le code validé, l'adresse est vérifiée et l'inscription est finalisée.
Scénario d'erreur	<ul style="list-style-type: none">• Étape 4 (Informations incomplètes) :<ul style="list-style-type: none">✗ Le système affiche un message d'erreur si des champs sont vides.✗ L'utilisateur doit fournir les informations manquantes.• Étape 4 (Annulation) :<ul style="list-style-type: none">✗ L'utilisateur peut annuler l'inscription avant soumission.• Étape 7 (Erreur d'enregistrement) :<ul style="list-style-type: none">✗ En cas d'échec d'enregistrement, un message invite à réessayer plus tard.• Étape 9 (Expiration du code OTP) :<ul style="list-style-type: none">✗ Le code OTP est valide pour une durée limitée.✗ À l'expiration, l'utilisateur est informé et peut générer un nouveau code via le lien dans l'e-mail.• Étape 10 (Code OTP incorrect) :<ul style="list-style-type: none">✗ En cas de code OTP erroné, un message invite à ressaisir le code.✗ Après plusieurs échecs, la validation OTP est temporairement bloquée et un nouveau code peut être envoyé.

□ Diagramme de séquence d'analyse du cas d'utilisation "S'inscrire" :

Les étapes de déroulement du cas d'utilisation "S'inscrire" sont représentées par un diagramme de séquence (voir Annexe B, Figure B.14). Ce diagramme illustre l'enchaînement des interactions entre l'utilisateur et le système lors de la procédure d'inscription.

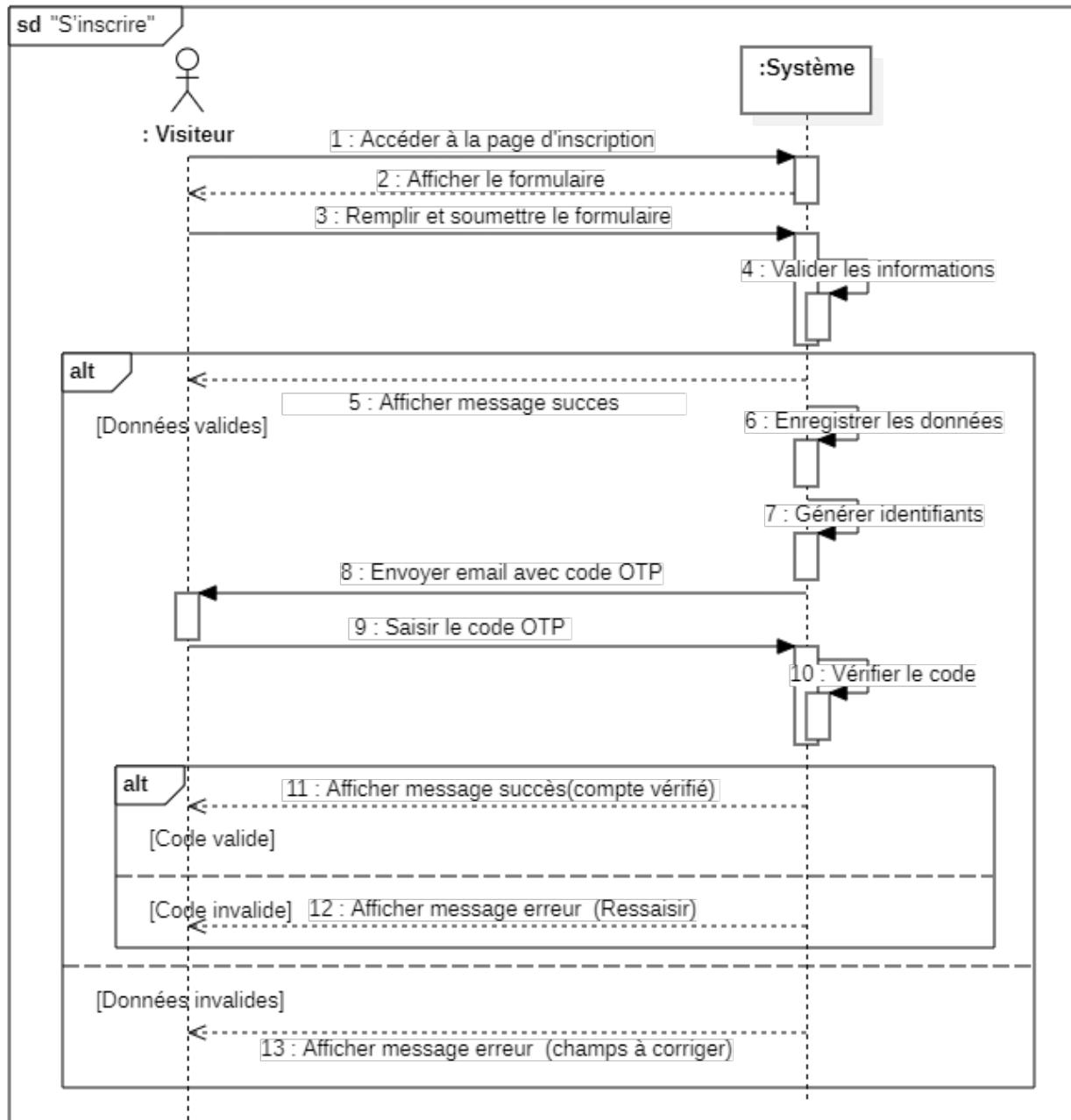


FIGURE B.14 – Diagramme de séquence d'analyse du cas d'utilisation de «S'inscrire»

□ Diagramme de séquence de conception de cas «Réinitialiser un mot de passe» :

Le diagramme de séquence de conception du cas « Réinitialiser un mot de passe » est illustré à la figure B.15.

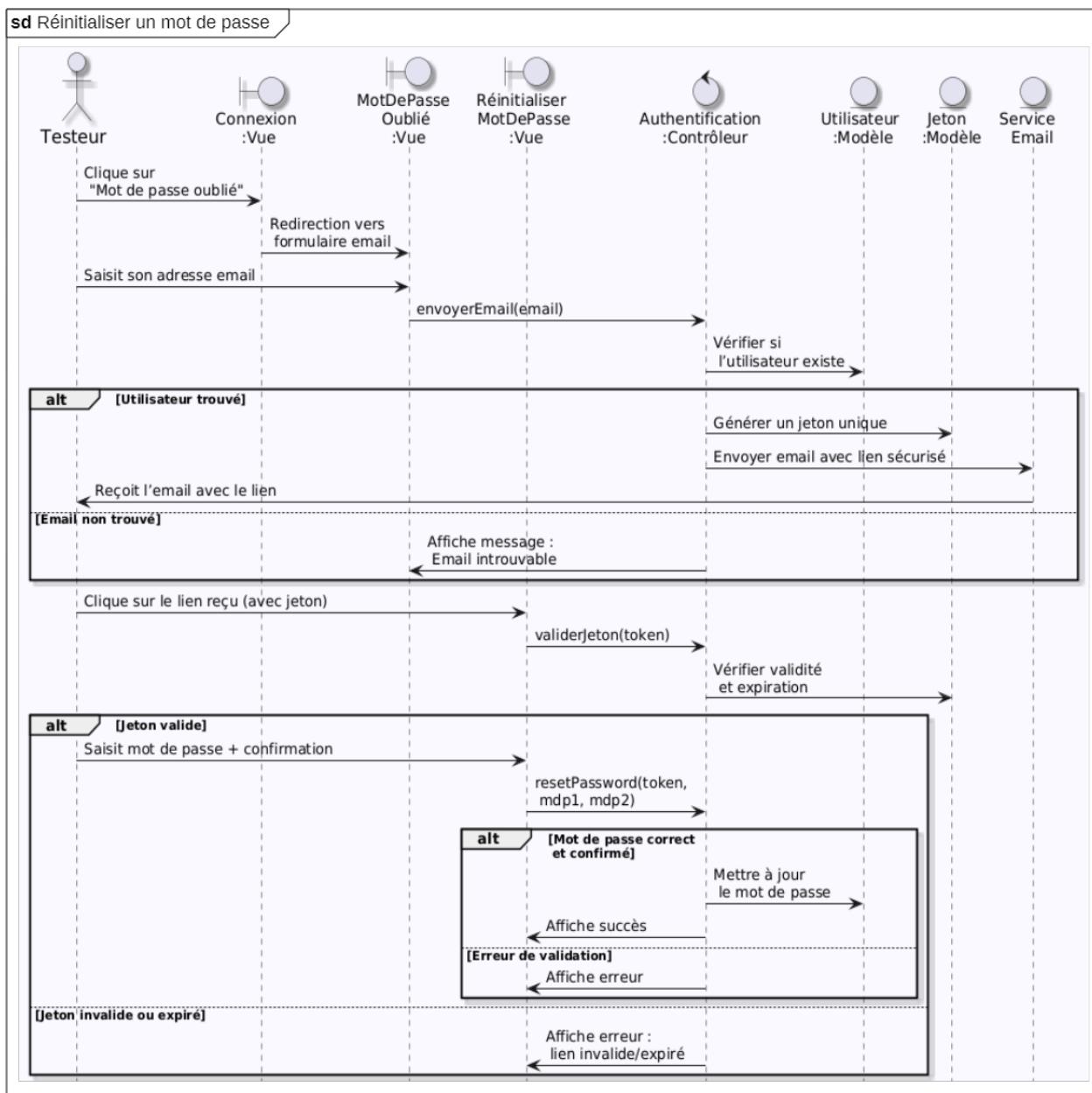


FIGURE B.15 – Diagramme de séquence de conception de cas «Réinitialiser un mot de passe»

Annexe C : Automatisation des outils de pentesting

Dans le cadre de l'automatisation des tests de sécurité web, divers outils ont été intégrés pour détecter un large éventail de vulnérabilités. Cette annexe synthétise les outils utilisés, leurs commandes principales et les résultats attendus.

1. OWASP ZAP[2]

- **Objectif :** Réaliser des scans automatisés de sécurité web via l'API REST.
- **Fonctionnalités :** Analyse passive et active, support de l'authentification par formulaire ou HTTP Basic, Spider classique et AJAX, contrôle total via l'API Python.
- **Analyse sans authentification :** L'outil explore l'application en tant que visiteur anonyme à l'aide du Spider, identifie les ressources accessibles publiquement, puis effectue un scan actif sur ces pages pour détecter les vulnérabilités (XSS, injections, configuration faible...).
- **Analyse avec authentification :** Après définition d'un contexte ZAP, l'authentification (formulaire ou HTTP Basic) est configurée via l'API. Un utilisateur est ensuite ajouté dans ce contexte avec ses identifiants. Le scan est exécuté en tant qu'utilisateur authentifié à l'aide du Spider (classique ou AJAX), permettant l'accès aux pages protégées et leur analyse.
- **Processus :** Définition du contexte → Configuration de l'authentification → Création de l'utilisateur → Exécution du Spider et du scan actif → Génération du rapport.
- **Résultat :** Analyse complète de l'application (publique et protégée), détection des vulnérabilités selon les différents niveaux d'accès, export du rapport en HTML, JSON ou XML.

2. Nmap[50]

- **Objectif :** Analyse réseau, identification ports ouverts et vulnérabilités.
- **Commande :** nmap -sS -p- -A -T4 --script=vuln {target} -oX log.xml
- **Paramètres clés :** -sS (scan SYN furtif), -p- (tous ports), -A (détection OS/versions), -script=vuln (scripts vulnérabilités).
- **Résultat :** Ports ouverts et vulnérabilités identifiées.

3. SQLMap[14]

- **Objectif :** Détection/exploitation injections SQL via 5 techniques (UNION, Boolean-blind, Error-based, Stacked queries, Time-based).
- **Commande :** python {sqlmap_path} -u {url} --batch --level={level} --risk={risk} --crawl={crawl} --threads={n} --output-dir={output}
- **Résultat :** Rapport complet vulnérabilités SQL et exploitations potentielles.

4. Nikto[13]

- **Objectif** : Scanner serveurs web pour vulnérabilités connues.
- **Commande** : perl nikto.pl -h {url} -o report_log.json -Format json
- **Résultat** : Rapport JSON vulnérabilités (headers non sécurisés, fichiers sensibles).

5. Wapiti[3]

- **Objectif** : Audit sécurité via simulation d'attaques (XSS, SQLi, CRLF, File Disclosure, Command Injection).
- **Commande** : wapiti -u {url} -m all -f json -o {output_dir}
- **Résultat** : Fichier JSON vulnérabilités classées par type et URL.

6. Nuclei[12]

- **Objectif** : Scanner avec templates communautaires (CVE, XSS, SQLi, Misconfig).
- **Commande** : nuclei -u {url} -json-export {output_path}/nuclei.json
- **Résultat** : Rapport JSON détaillant vulnérabilités par template appliqué.

7. PwnXSS[51]

- **Objectif** : Détection automatique XSS via crawling et analyse dynamique.
- **Commande** : python pwnxss.py -u {url} --depth 2 > result.json
- **Résultat** : Vecteurs XSS identifiés avec URL, paramètre vulnérable et payload.

8. XSSStrike[15]

- **Objectif** : Détection/exploitation XSS avancée avec contournement protections anti-XSS.
- Fonctionnalités** : Analyse paramètres HTTP, génération payloads personnalisés, crawling multi-threadé.
- **Commande** : python {XSSStrike_path} -u {url} --crawl -l {depth} --threads {n}
- **Résultat** : Exploration pages internes, analyse paramètres vulnérables, détection éléments exploitables.

9. WhatWeb[53]

- **Objectif** : Identification technologies utilisées via détection passive.
- **Commande** : whatweb -a {level} --log-json={output_file} {url}
- **Résultat** : Rapport JSON technologies identifiées (WordPress, Apache, PHP).

10. WafW00f[52]

- **Objectif** : Détection pare-feux applicatifs web (WAF) via signatures comportementales.
- **Commande** : wafw00f {url} --output {output_path}/wafw00f.json
- **Résultat** : Type et fournisseur WAF avec techniques de contournement.

Cette combinaison d'outils assure une couverture complète des vecteurs d'attaque. Les résultats sont centralisés dans un rapport JSON final, facilitant leur exploitation et leur corrélation.

Annexe D : Automatisation d'analyse SEO et fonctionnels

1- Automatisation d'analyse SEO

Dans le cadre du projet, un module a été conçu pour automatiser l'analyse technique d'un site web à des fins d'optimisation SEO et d'identification des technologies utilisées.

- Objectifs

- Évaluer la qualité SEO d'une page à travers des critères techniques.
- Identifier les technologies côté serveur (backend) et client (frontend), y compris les CMS.
- Extraire des mots-clés dominants pour l'analyse sémantique.
- Générer une capture d'écran et un rapport synthétique avec un score global.

- Technologies utilisées

- **FastAPI, Requests, Jinja2** : pour la communication API et la génération HTML.
- **Selenium + Chrome Headless** : pour la capture visuelle.
- **BeautifulSoup, NLTK** : pour le parsing HTML et l'analyse linguistique.

- Fonctionnalités principales

- a) **Analyse SEO de base** : évalue la structure HTML (balises, temps de chargement, taille) et calcule un score sur 100.
 - b) **Détection technologique** : identifie le serveur, CMS, frameworks JS/CSS et système d'exploitation via les en-têtes HTTP et l'analyse HTML.
 - c) **Capture d'écran** : fournie par Selenium pour représenter visuellement la page analysée.
 - d) **Extraction de mots-clés** : grâce à NLTK, les mots et expressions les plus représentatifs sont identifiés.
 - e) **Analyse des liens** : distingue les liens internes (maillage) et externes (ressources tierces).
- **Rapport généré** : Le rapport fournit une synthèse technique du site analysé, incluant les titres HTML, les performances de chargement, les statistiques sur les liens et les images, les technologies détectées, une capture d'écran encodée, ainsi qu'un score final accompagné de recommandations.

Ce module d'analyse SEO automatisée fournit un audit rapide et complet, utile pour les développeurs, référenciers et pentesters souhaitant évaluer la performance technique et le positionnement d'un site web.

2- Automatisation des tests fonctionnels avec Selenium

Une autre composante du projet consiste à automatiser les tests fonctionnels d'un site web à l'aide de Selenium. Cette automatisation repose sur un éditeur interactif permettant à l'utilisateur de définir ses scénarios de test sous forme de workflows dynamiques.

- Objectifs

- Offrir une interface intuitive permettant de décrire des scénarios de test personnalisés.
- Automatiser l'exécution des cas de test définis sans intervention manuelle.
- Identifier les erreurs fonctionnelles en simulant des comportements utilisateurs réels.

- Principe de fonctionnement

L'utilisateur crée un **workflow** de test via une interface visuelle, composé de plusieurs éléments :

- **Test Cases** : chaque cas représente un objectif fonctionnel à valider (ex. envoi d'un message via un formulaire).
- **Step Cases** : chaque étape décrit une action précise (ex. cliquer sur un bouton).
- **Paramètres** : pour chaque étape, l'utilisateur peut spécifier :
 - * Le **sélecteur CSS** ou XPath ciblant l'élément HTML.
 - * Le **type d'action** : click, input, select...
 - * La **valeur associée** (texte à saisir, bouton à cliquer...).

Une fois le scénario complété, l'utilisateur peut lancer le workflow. Le système exécute alors toutes les actions dans l'ordre défini, à l'aide de **Selenium WebDriver** et enregistre les résultats de chaque étape.

- Technologies utilisées

- **Selenium WebDriver** : moteur d'exécution des tests sur navigateur.
- **Chrome Headless** : pour les tests sans interface graphique.
- **FastAPI** : pour gérer les scénarios, orchestrer les exécutions et stocker les résultats.

- Fonctionnalités principales

- a) **Création de scénarios personnalisés** : via une interface intuitive.
- b) **Exécution automatisée des étapes définies**.
- c) **Contrôle de présence d'éléments**.
- d) **Rapport détaillé** : statut des étapes, messages d'erreur, captures d'écran.

Ce module offre à l'utilisateur la possibilité de décrire, exécuter et analyser ses tests fonctionnels sans coder manuellement. Il constitue un outil puissant pour valider automatiquement les fonctionnalités critiques d'un site web tout en simplifiant le processus de test.

Annexe E : Interfaces de l'application AutoTest

Cette annexe présente les interfaces de la nouvelle application **AutoTest**. Elles ont été repensées et améliorées à partir des observations faites sur la version initiale (voir Annexe A), dans une démarche d'optimisation de l'ergonomie et de l'expérience utilisateur.

- **Nouveau logo :** Après le changement de nom de l'application, passant de *Pentra* (spécialisée dans les tests de sécurité) à *AutoTest* (qui automatise les tests fonctionnels, les tests de sécurité et l'audit SEO), il a été nécessaire de modifier également le logo de l'application. Le nouveau logo a été conçu en cohérence avec la charte graphique choisie, en utilisant une dominante bleue, inspirée des couleurs du logo de l'entreprise d'accueil Addinn.



FIGURE E.16 – Nouveau logo de l'application *AutoTest*

The screenshot shows the homepage of the AutoTest website. At the top, there is a navigation bar with links for Home, About, Features, Services, Pricing, FAQ, Contact, and a user profile for Rihab Cherni. Below the navigation bar, there is a large banner with the heading "Automated testing platform for website applications". The banner features an illustration of two people working on a computer, surrounded by icons related to web development and security. A call-to-action button at the bottom right of the banner says "Start discovering your app's vulnerabilities. Take action today". Below the banner, there are four sections highlighting the platform's features: "3x Awarded Excellence" (with a trophy icon), "6.5k Secure Deployments" (with a shield icon), "80k Tests Run" (with a bar chart icon), and "6x Trusted by Industry Leaders" (with a shield icon). Each section includes a brief description of the feature.

FIGURE E.17 – Interface de la page d'accueil

MORE ABOUT US

Automated Security & Functional Testing

AutoTest is a powerful platform built to streamline and automate both security and functional testing for modern web applications. It combines industry-leading tools and custom workflows into a single, user-friendly interface.

From penetration testing to SEO audit scans, and from login validation to detailed reporting, our solution empowers teams to find vulnerabilities, validate behavior, and secure their applications faster and more effectively.

Whether you are a cybersecurity expert, QA engineer, or developer, AutoTest simplifies your testing process and ensures robust application security and functionality.

- ✓ Automated security scans
- ✓ Centralized dashboard
- ✓ Functional test automation
- ✓ SEO audit and validation
- ✓ Customizable testing flows
- ✓ Detailed vulnerability reports

Trusted Expertise
Proven excellence in cybersecurity and automated testing solutions

75 Targets Scanned **124** Vulnerabilities Detected **380** Security Tests Executed **210** Functional Tests Executed

Enhance Your Security Today!

Automate your testing process and identify vulnerabilities faster. Our tool provides seamless integration, comprehensive scans, and real-time reporting to ensure your systems stay secure.

[Start Your Free Trial](#)

Key Features

Discover the essential features designed to help you automate security testing and streamline vulnerability management.

- [Security Scans](#)
- [Functional Scans](#)
- [SEO Scans](#)

Automated Security Scans

Automate comprehensive security scans to detect vulnerabilities across your web applications, from OWASP Top 10 to custom security tests.

- ✓ Scan for vulnerabilities like XSS, SQL Injection, and more.
- ✓ Automate vulnerability detection with continuous testing.
- ✓ Customizable scan profiles for different threat models.

Services

Our services streamline security testing, offering automated vulnerability scans, functional testing, and real-time alerts to enhance your workflow and ensure system safety.

Security Scan Automation

Automate security tests using integrated tools like ZAP, Wapiti, SQLMap, Nikto, and more to detect a wide range of vulnerabilities.

Functional Testing

Create and manage test scenarios, test cases, and steps to ensure comprehensive functional coverage of your applications.

Authenticated Scanning

Perform security scans on protected areas by leveraging dynamic authentication with cookies, tokens, or passwords.

Real-time Monitoring

Track scan progress in real-time via WebSockets technology and receive instant notifications for critical vulnerabilities.

Comprehensive Reports

Generate detailed reports in multiple formats (JSON, PDF, CSV) and integrate with tools like Jira, Slack, and email.

Scheduled Scans

Set up automated scanning schedules to regularly monitor your applications for new vulnerabilities.

Pricing Plans

We have adjusted our pricing plans to better meet your needs. Explore the new options below.

Free Plan	Professional Plan	Premium Plan
0.0 / month	9.9 / month	19.9 / month
Ideal for students or small development teams and startups.		
Featured included:		
<ul style="list-style-type: none"> ✓ Security scanning with basic tools ✓ Manual scan initiation ✓ Basic reporting ✓ Email support 		
Buy Now →	Buy Now →	Buy Now →

Contact Information

We are available to support your needs in IT security and automated testing.

Address
Immeuble Eratior, Rue Khadija Ben Arfa, Centre Urbain Nord 1082, Tunis, Tunisia

Phone
+216 50 099 824

Email
contact@autodinn.com

[Send Message](#)

Get In Touch

AutoTest is a powerful web application for automated security testing. It helps identify vulnerabilities, streamline scans, and generate actionable reports.

Quick Links

- [Home](#)
- [FAQ](#)
- [About](#)
- [Contact](#)
- [Features](#)
- [Policy](#)
- [Services](#)
- [Login](#)
- [Pricing](#)

Localisation

© Copyright 2023 AutoTest. All Rights Reserved

FIGURE E.18 – Autres sections de la page d'accueil

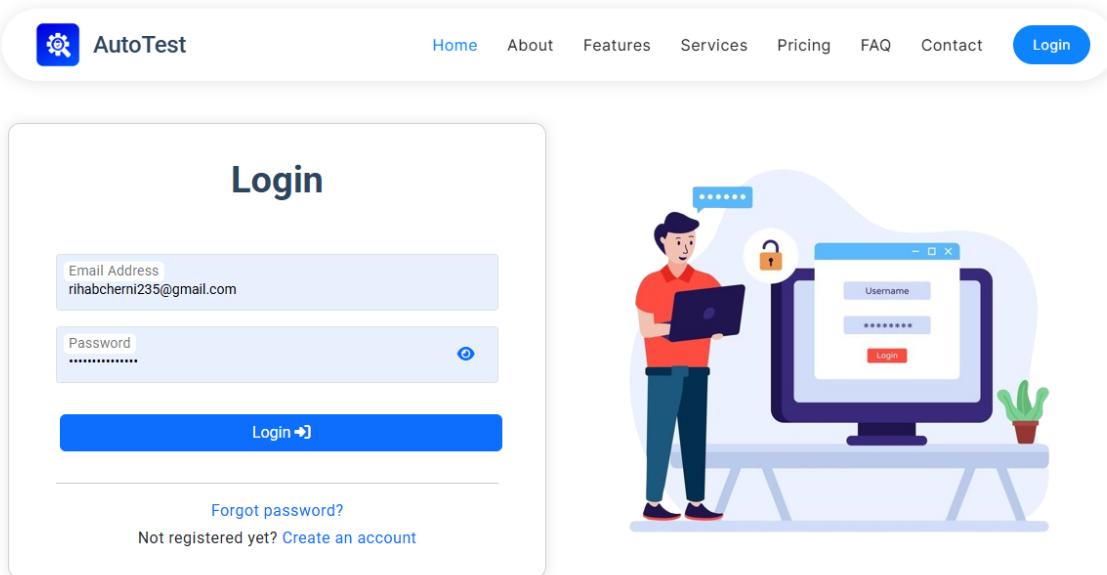


FIGURE E.19 – Interface de connexion

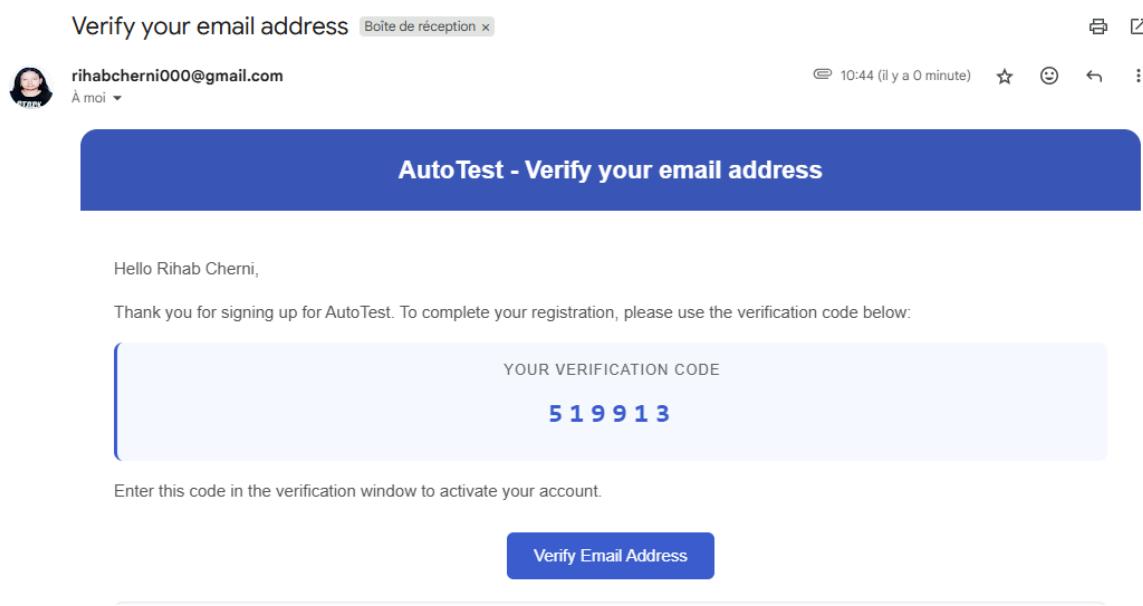


FIGURE E.20 – E-mail contenant le code OTP de vérification du compte

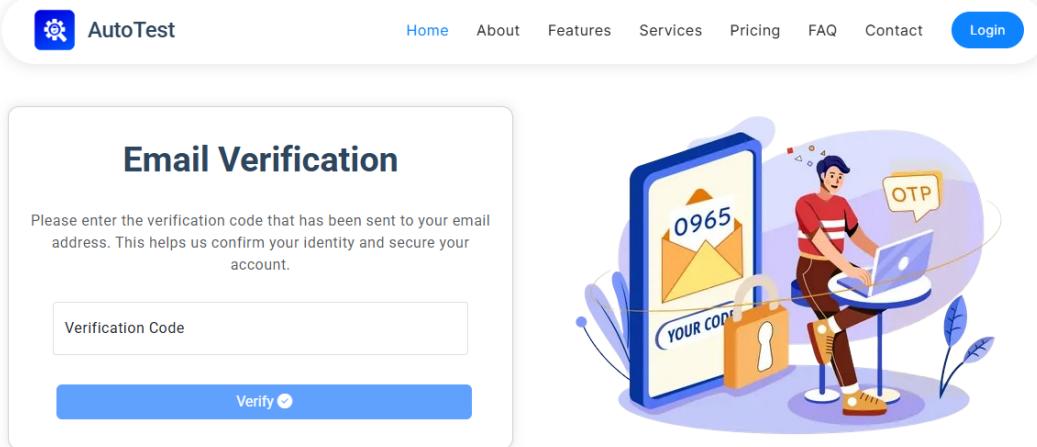


FIGURE E.21 – Interface de saisie du code OTP pour la vérification du compte

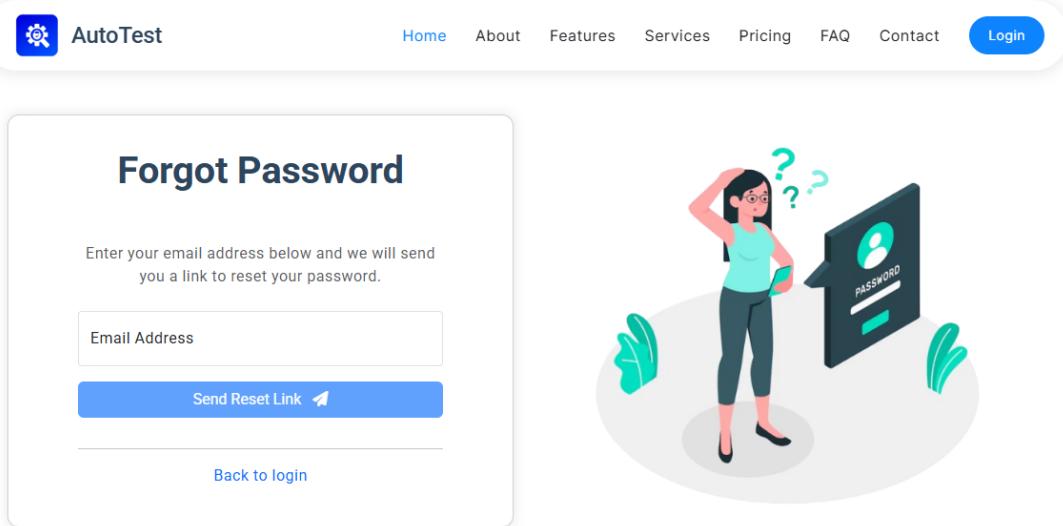


FIGURE E.22 – Interface de récupération du mot de passe oublié

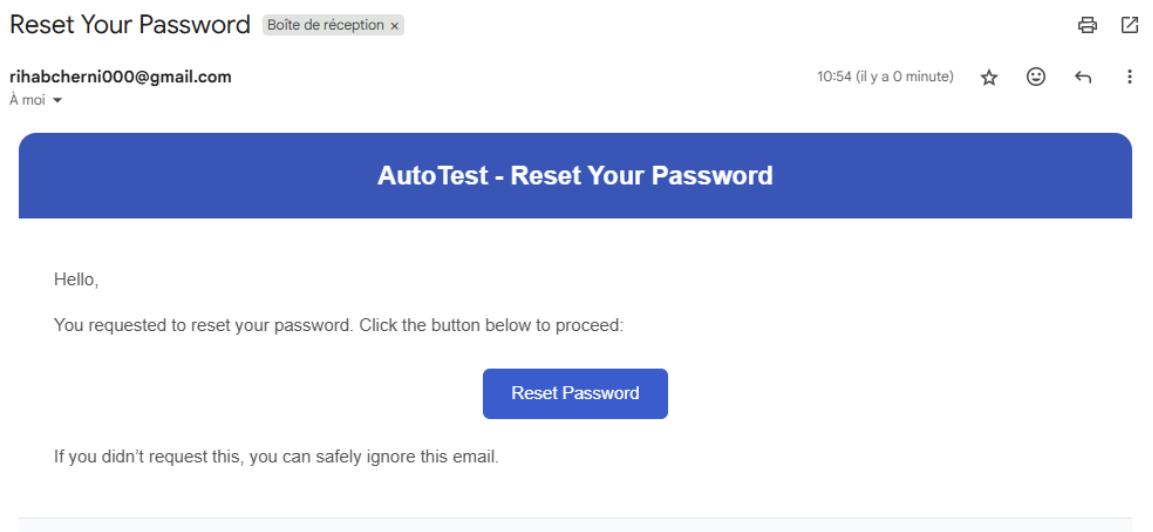


FIGURE E.23 – Confirmation d'envoi de l'e-mail de réinitialisation

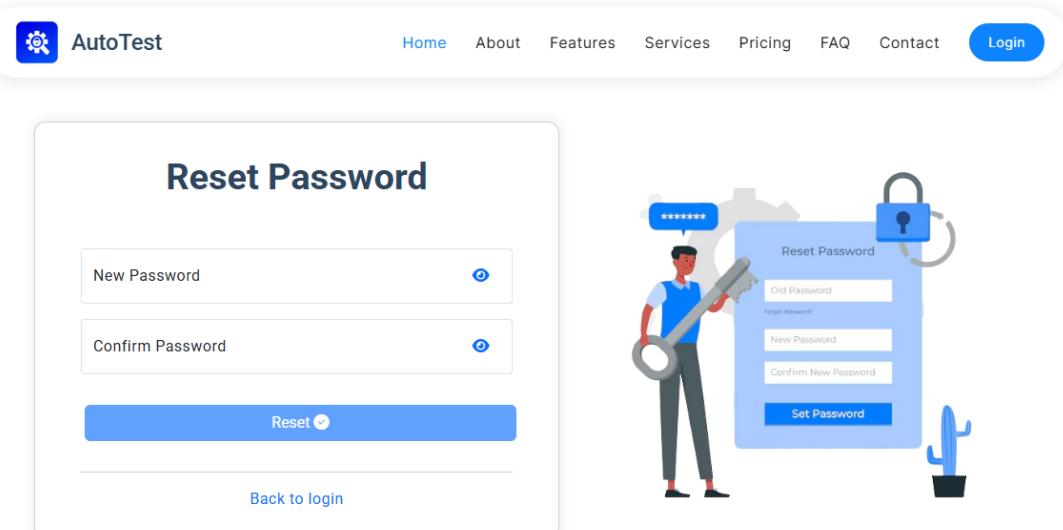


FIGURE E.24 – Interface de réinitialisation du mot de passe

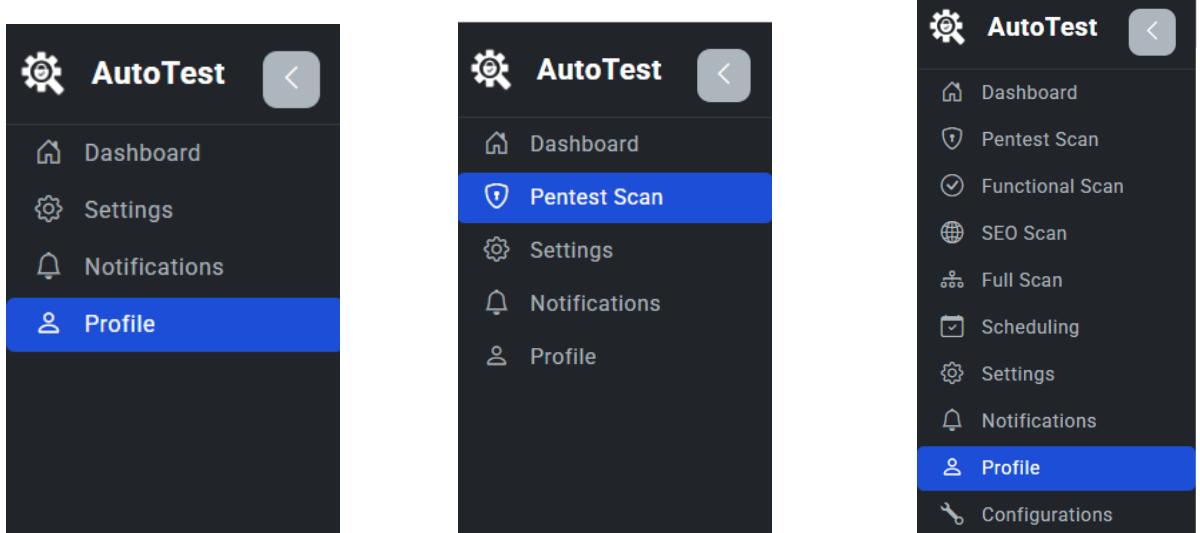


FIGURE E.25 – Menu latéral de Testeur

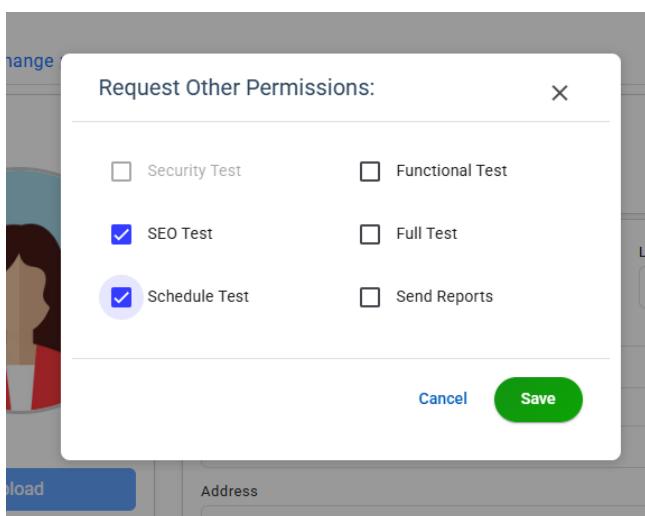
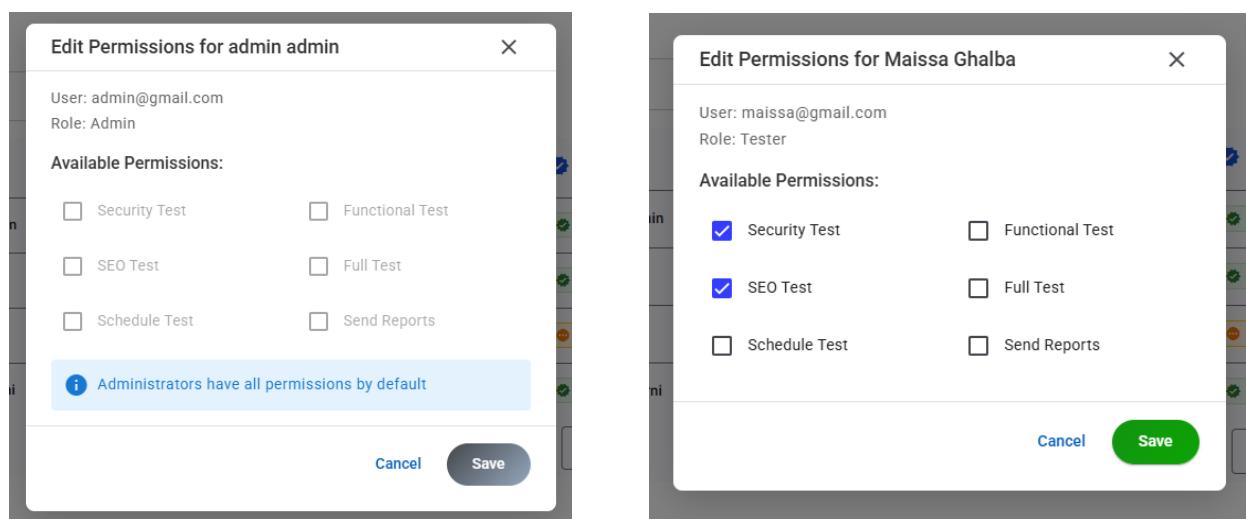


FIGURE E.26 – Interfaces liées à la gestion des permissions

The screenshot shows the 'Contact message' section of the AutoTest application. On the left, there's a sidebar with navigation links: Dashboard, Users, Reports, Contact Messages (which is the current page), Notifications, Profile, and Configurations. At the top right, there are user profile icons for 'admin admin' and standard system icons for moon, bell, and refresh. A search bar with a magnifying glass icon and a 'Download' button are at the top. The main content area has a table with columns: Name, Email, Subject, and Message. It lists five messages from users Karim Haddad, Nour Hammami, Sara Bouzid, Youssef El Amrani, and Amina Cherif, each with their respective details and a snippet of the message content. Below the table are pagination controls: 'Items per page: 5', '1 - 5 of 6', and navigation arrows.

FIGURE E.27 – Interface d’administration : Messages reçus via le formulaire de contact

The screenshot shows the 'Security Scanner' configuration interface. On the left, there's a sidebar with various icons for navigation. At the top right, there are user profile icons for 'Rihab Cherni tester' and standard system icons. The main content area has a heading 'Scan URLs in real time to ensure your websites stay secure.' Below it, a sub-heading says 'Instantly detect XSS, SQL injection, command injection, XXE, and other critical vulnerabilities – automatically validated to eliminate false positives.' There are several tabs: 'Scan Configuration', 'Select Tools' (disabled), and 'Advanced Options' (selected). Under 'Advanced Options', there are several sections: 'General Options' (Number of Threads: 4, Crawl Depth: 3, Enable Crawling: Yes), 'ZAP Scanner Options' (ZAP Directory Check: 5, ZAP Depth: 5), 'Wapiti Scanner Options' (Scan Time (minutes): 5, Wapiti Level: Level 2 - Advanced), 'SQLMap Options' (SQLMap Level: Level 3, Risk: Risk 3 - High, Threads: 5, Technique: BEUSTQ), 'Nikto Scanner Options' (Timeout: 300), 'Nuclei Scanner Options' (Rate Limit: 100), 'Nmap Scanner Options' (Timing: T4 - Aggressive), 'WhatWeb Scanner Options' (Aggression: Level 3 - Aggressive), and 'PwnXSS Scanner Options' (Threads: 10).

FIGURE E.28 – Interface de configuration des paramètres de scan

FIGURE E.29 – Interface de suivi en temps réel des scans (WebSocket)

FIGURE E.30 – Interface de visualisation des résultats de scan (Statistiques et détails techniques)

FIGURE E.31 – Interface de visualisation des résultats de scan (Logs)

Scan Progress

Scan completed at Jun 21, 2025, 5:34:22 PM

Vulnerabilities

Search vulnerabilities... All Severities All Tools

Cross Site Scripting (High) 4 vulnerabilities

Category Information

Detection Tool: XSSStrike, PwnXSS, wapiti, nmap, zap

Description
Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser obj... [Show more](#)

General Solution
Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. Examples of libraries and fr... [Show more](#)

References
<https://owasp.org/www-community/attacks/xss/> <https://cwe.mitre.org/data/definitions/79.html>

Cross Site Scripting (Reflected) (Confidence: Medium, Score: 20.0, zap)

Vulnerability Information

Basic Information

Method: POST
Detected By: zap
Confidence: Medium
Confidence Score: 20.0%

Target Information

URL: [http://testphp.vulnweb.com/search.php?test='<scrIpt>alert\(1\);</scRipt>](http://testphp.vulnweb.com/search.php?test='<scrIpt>alert(1);</scRipt>)

Attack Evidence

Raw Attack Data

```
1. ''<scrIpt>alert(1);</scRipt>
```

Cross Site Scripting (Reflected) (Confidence: Medium, Score: 20.0, zap)

Cross Site Scripting (Reflected) (Confidence: Medium, Score: 20.0, zap)

Cross Site Scripting (Reflected) (Confidence: Medium, Score: 20.0, zap)

SQL Injection (High) 4 vulnerabilities

User Agent Fuzzer (Informational) 166 vulnerabilities

FIGURE E.32 – Interface de visualisation des résultats de scan (agrégation multi-outils)

The screenshot shows the AutoTest application interface. On the left is a dark sidebar with a navigation menu:

- Dashboard
- Pentest Scan
- Functional Scan
- SEO Scan
- Full Scan
- Scheduling
- Notifications
- Profile
- Configurations

At the bottom of the sidebar is a user profile for "Rihab Cherni" (tester).

The main content area has a header "Security Scanner (Results)". Below it is a "Scan Progress" section with a green progress bar at 100% and the text "Scan completed at Jun 21, 2025, 5:34:22 PM".

Below the progress bar are tabs: "Summary", "Vulnerabilities", "Details" (which is selected), and "Logs". There are also "New Scan" and "Download" buttons.

The main content area displays two sections: "Whatweb" (0 findings) and "zap" (30 findings). The "zap" section shows a list of vulnerabilities.

FIGURE E.33 – Interface de visualisation des résultats de scan (par outil)

The screenshot shows the AutoTest application interface. On the left is a dark sidebar with a navigation menu, where "Pentest Scan" is currently selected.

The main content area has a header "Security Scan (Reports)". It includes a search bar and buttons for "+ New Scan" and "Download".

The main content area displays a "Security Scan Details" section for a scan of "http://testphp.vulnweb.com" that started at 6/21/25, 5:27 PM and finished at 6/21/25, 5:34 PM. The report shows the following findings:

TOTAL	HIGH	MEDIUM	LOW	INFO	OTHER
175	8	0	0	167	0

Below this, there is a status summary:

Status	Completed	Authentication	Scheduled
Started	6/21/25, 5:27 PM	Finished	6/21/25, 5:34 PM
TOOLS	Whatweb zap		

Below the details, there is a list of previous scans:

- http://testphp.vulnweb.com (Start: 6/21/25, 4:38 PM)
- http://testphp.vulnweb.com (Start: 6/21/25, 4:35 PM)
- http://testphp.vulnweb.com (Start: 6/21/25, 4:11 PM)
- http://testphp.vulnweb.com (Start: 6/21/25, 3:59 PM)

At the bottom are pagination controls: "Items per page: 5", "1 - 5 of 51", and navigation arrows.

FIGURE E.34 – Interface de l'historique des scans

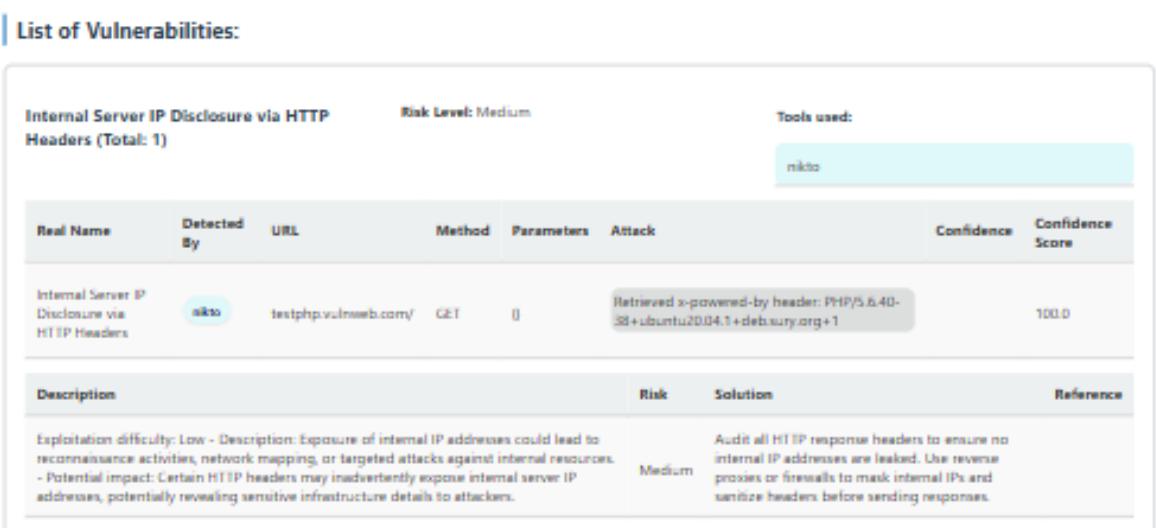


FIGURE E.35 – Téléchargement d'un rapport de scan au format HTML

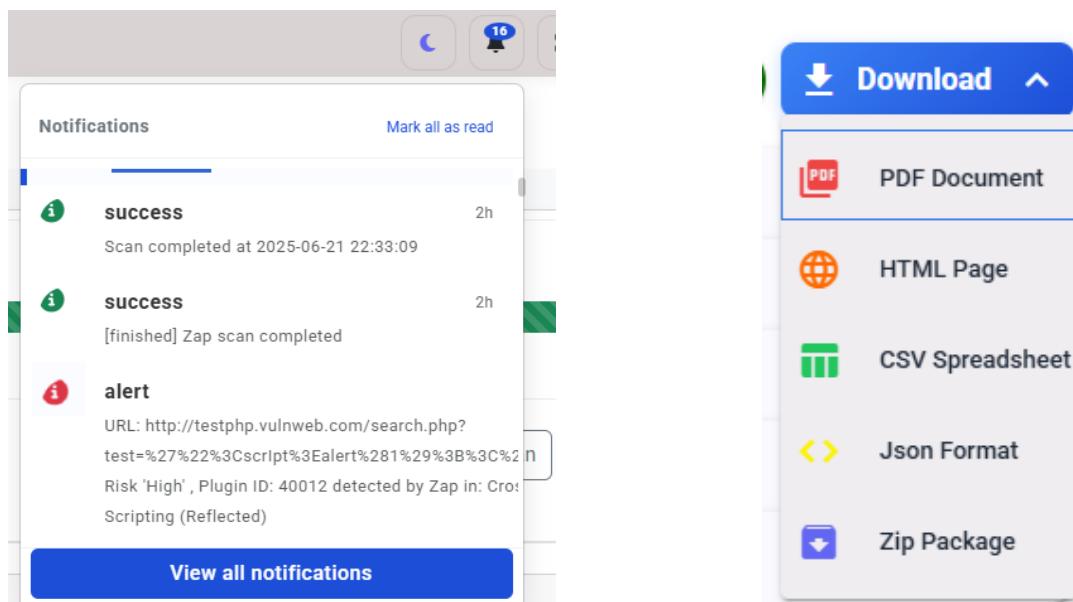
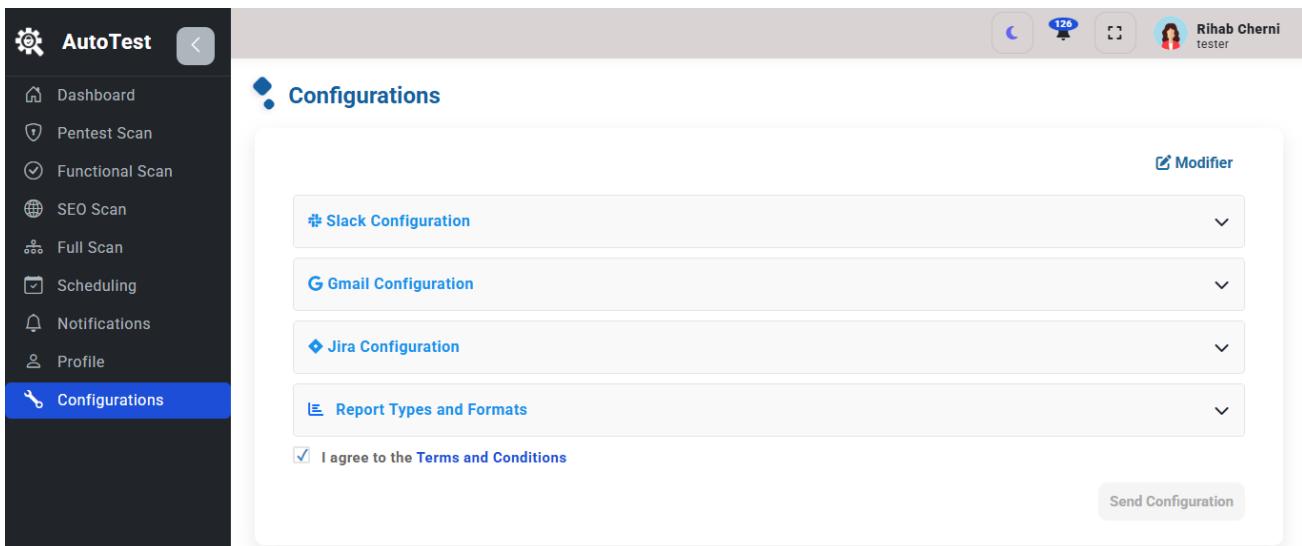


FIGURE E.36 – Menu de notifications

FIGURE E.37 – Menu de choix de téléchargement des rapports



(a) Interface de configuration

Slack Configuration

Slack configuration selected. Please provide the necessary details.

Enter your token	Enter your channel ID
------------------	-----------------------

Token is required. Channel ID is required.

(b) Configuration Slack

G Gmail Configuration

Gmail configuration selected. Please provide the necessary details.

rihabcherni235@gmail.com	<input type="button" value="X"/>
rihabcherni23@gmail.com	<input type="button" value="X"/>

(c) Configuration des Emails

♦ Jira Configuration

Jira configuration selected. Please provide the necessary details.

Enter your Jira Email	Enter your token
-----------------------	------------------

Jira Email is required. Jira Token is required.

Enter your Jira Domain	Enter your Jira Project Key
------------------------	-----------------------------

Jira Domain is required. Jira Project Key is required.

Enter your Jira Board

Jira Board is required.

(d) Configuration Jira

≡ Report Types and Formats

Select the desired report types and formats.

Report Types	Output Formats
<input type="checkbox"/> Seo Report	<input checked="" type="checkbox"/> HTML
<input checked="" type="checkbox"/> Security Report	<input checked="" type="checkbox"/> PDF
<input type="checkbox"/> Functional Report	<input type="checkbox"/> CSV
<input type="checkbox"/> Full Report	<input type="checkbox"/> ZIP
	<input type="checkbox"/> JSON

(e) Configuration formats et types d'envoi des rapports

FIGURE E.38 – Interface de paramétrage des canaux de diffusion

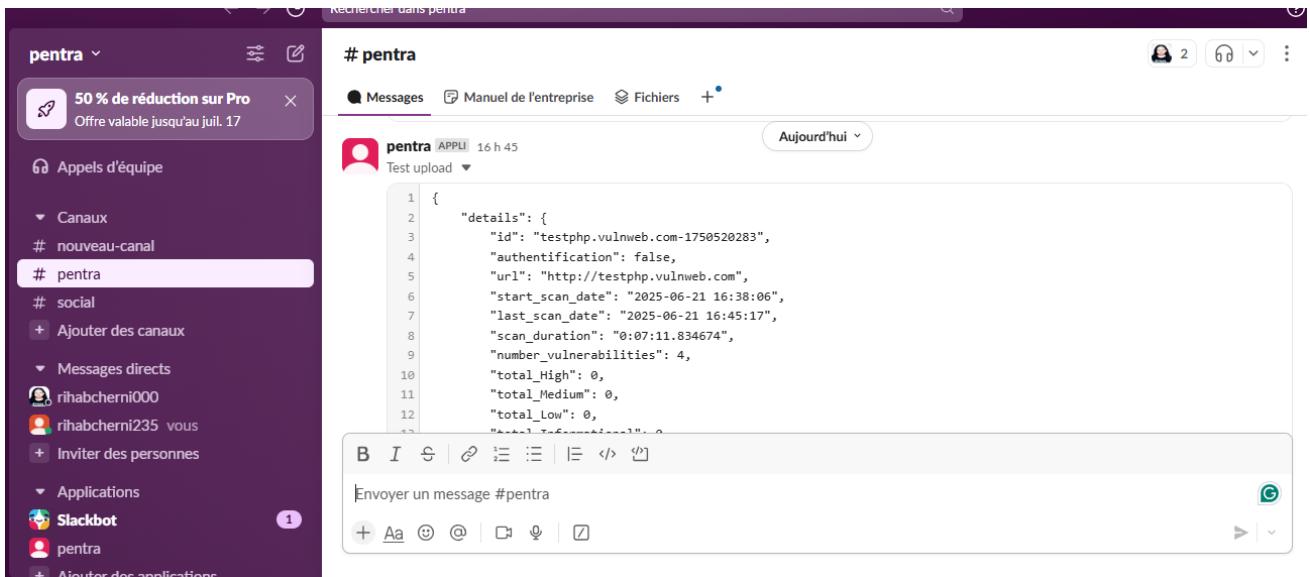


FIGURE E.39 – Interface de notification via Slack

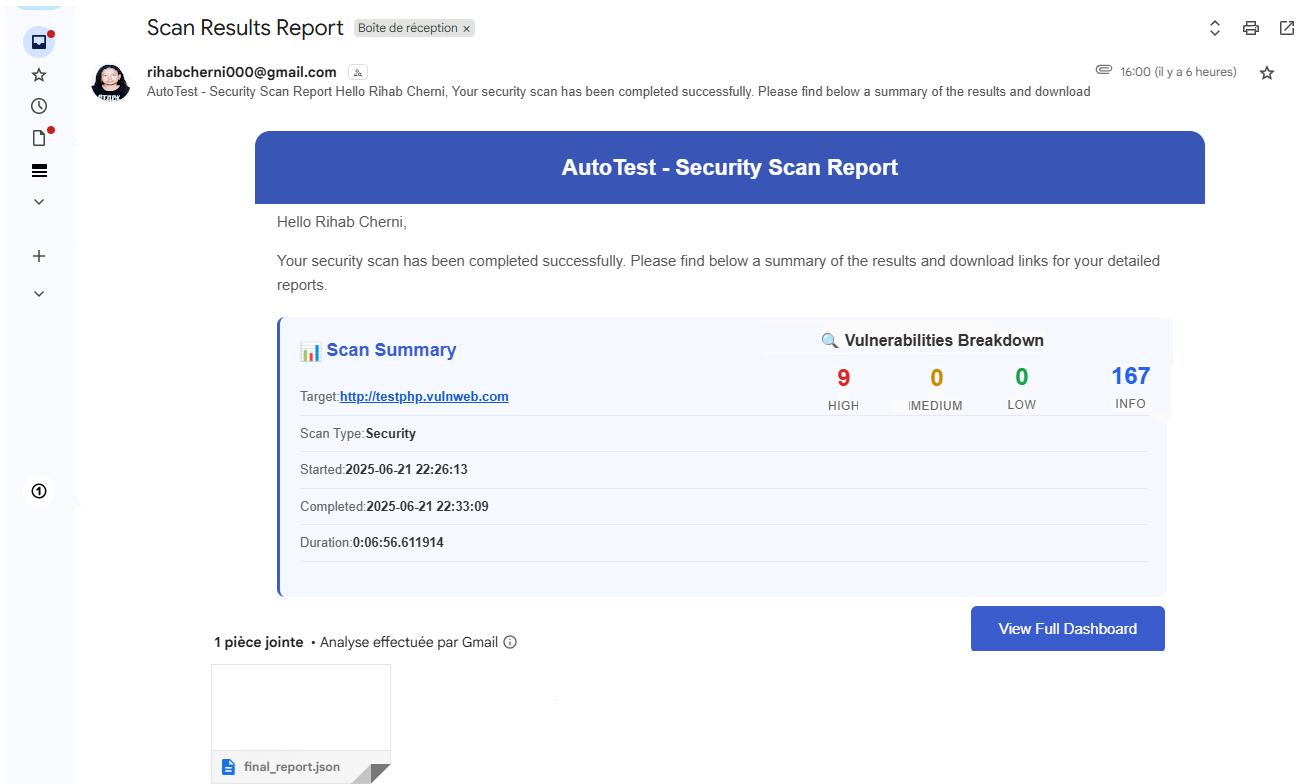


FIGURE E.40 – Interface de notification par e-mail

Security Scanner (Schedule)

Juin 2025

URL	Type	Date/Time	Status	Progress
https://testphp.vulnweb.com	Fonctionnel	16/06/2025 at 19:00	cancelled	0%
https://demo.testfire.net	SEO	15/06/2025 at 08:00	queued	0%
https://zero.webappsecurity.com	Sécurité	11/06/2025 at 19:45	completed	100%
https://example.com	Sécurité	16/06/2025 at 08:45	cancelled	0%
https://vulnerable-app.com	SEO	02/07/2025 at 14:45	running	62%
https://juice-shop.herokuapp.com	SEO	17/06/2025 at 08:30	completed	100%
https://scanme.nmap.org	SEO	02/07/2025 at 09:15	failed	0%
https://target1.internal	SEO	30/06/2025 at 14:00	queued	0%
https://staging.website.net	Fonctionnel	23/06/2025 at 16:00	running	89%
https://legacy.insecure.dev	SEO	30/06/2025 at 08:15	running	35%

FIGURE E.41 – Interface de planification automatique des scans

Create Functional Report

Target URL **
Target URL is required

Enable Authentication

Authentication Type — **Token**

Login Page URL

Authentication Token

X Cancel **> Create Report**

Create Full Scan Report

Target URL ** — <http://testphp.vulnweb.com>

Select Test Type(s):

Functional Security SEO

Enable Authentication

Authentication Type — **Cookies**

Login Page URL

Authentication Cookies

X Cancel **Launch Scan**

FIGURE E.42 – Formulaire de lancement d'une analyse fonctionnelle

FIGURE E.43 – Formulaire unifiée de lancement des analyses

Complet Analysis Report - <https://www.ensit.tn/>

General Information

Scan Type:	Full
Status:	COMPLETED
Authentication:	DISABLED
Scheduled Scan:	NO
Started at:	2025-06-22 at 02:42
Finished at:	2025-06-22 at 02:42

[SEO Analysis](#) [Security](#) [Functional Tests](#) [Authentication](#)

SEO Summary

Average Score:	72.85
Pages Analyzed:	21

Server Information

Server IP:	79.137.112.24
Server OS:	
Server Software:	OVHcloud
Backend Technologies:	PHP
Frontend Technologies:	jQuery, Bootstrap, Next.js

Keywords

Keyword	Frequency
meta	21
ensit	23
master	2
missing	21
habilitations	2

Recurring Phrases

Phrase	Frequency
ensit nos	2
missing meta	15
ensit missing	4
ensit diplôme	10
description ensit	20

Crawled Pages (Details)

- [ENSIT](#)
- [ENSIT | ENSIT](#)

Canonical URL: <https://www.ensit.tn/ensit/>
Meta Description: ❌ Missing meta description
Load Time: 436 ms
HTML Size: 39.57 KB
Favicon: -Favicon
Internal Links: 57 | **External Links:** 7
SEO Score: 75 (Grade: C)

Good Practices ✓:

- ✓ Title tag is present
- ✓ 1 <h1> tag(s) found
- ✓ 1 <h2> tag(s) found
- ✓ 6 <h3> tag(s) found
- ✓ 5 <h4> tag(s) found
- ✓ Canonical tag is defined
- ✓ Robots meta tag is present
- ✓ Favicon is present

Bad Practices ✗:

- ✗ Missing meta description
- ✗ No <h5> tag found
- ✗ No <h6> tag found
- ✗ 2 images missing alt attributes

Header Tags:

H1 tags:

H2 tags:

H3 tags:

 (empty)
 Actualités
 Catégories
 ENSIT
 Menu rapide
 Contactez-nous

H4 tags:

 Formations certifiantes SolidWorks
 Journée culturelle Paléorientaine
 OUVERTURE DES CANDIDATURES AUX MASTERS DE L'ENSI
 Résultat de la troisième session TOEIC
 Résultat de la troisième session TOEFL
 Résultat de la troisième session IELTS

H5 tags:

 ✗ No <h5> tag found

FIGURE E.44 – Téléchargement d'un rapport d'analyse complet au format HTML (onglet SEO)

Bibliographie

- [1] *Addinn Group : Qui sommes-nous ?* <https://www.addinn-group.com/about-us/>, Consulté le 04/03/2025.
- [2] *OWASP Zed Attack Proxy (ZAP)*, <https://www.zaproxy.org/>, Consulté le 10/03/2025.
- [3] *Wapiti - Web Application Vulnerability Scanner*, <https://wapiti-scanner.github.io/>, Consulté le 10/03/2025.
- [4] *15 Meilleur scanner de sites Web pour trouver les vulnérabilités de sécurité et les logiciels malveillants en 2025*, <https://geekflare.com/fr/cybersecurity/best-website-security-scanner/>, Consulté le 06/03/2025.
- [5] *Tableau de bord pour HostedScan*, <https://hostedscan.com/dashboard>, Consulté le 18/03/2025.
- [6] *Invicti – Site officiel*, <https://www.invicti.com/>, Consulté le 18/03/2025.
- [7] *Pourquoi Cypress ?* <https://docs.cypress.io/app/get-started/why-cypress>, Consulté le 17/03/2025.
- [8] *Testez les interfaces de vos applications React avec Cypress*, <https://www.ie-concept.fr/2020/11/06/testez-les-interfaces-de-vos-applications-react-avec-cypress/>, Consulté le 17/03/2025.
- [9] *Simplifiez l'automatisation des tests avec Katalon Studio*, <https://katalon.com>, Consulté le 17/03/2025.
- [10] *Les 5 meilleurs outils SEO pour optimiser votre référencement*, <https://blog.mrsuricate.com/5-meilleurs-outils-seo-optimiser-referencement>, Consulté le 17/03/2025.
- [11] *20 meilleurs outils de test d'automatisation 2025 | ClickUp*, <https://clickup.com/fr-FR/blog/228197/les-outils-de-test-d-automatisation>, Consulté le 18/03/2025.
- [12] *Nuclei - Fast and Customizable Vulnerability Scanner*, <https://nuclei.projectdiscovery.io/>, Consulté le 17/04/2025.
- [13] *Nikto - Web Server Scanner*, <https://cirt.net/Nikto2>, Consulté le 17/04/2025.
- [14] *SQLMap - Automatic SQL injection and database takeover tool*, <https://sqlmap.org/>, Consulté le 17/04/2025.
- [15] *XSSStrike*, <https://github.com/s0md3v/XSSStrike>, Consulté le 12/04/2025.

- [16] *JWT (JSON Web Token) : vulnérabilités, attaques courantes et bonnes pratiques sécurité*, <https://www.vaadata.com/blog/fr/jwt-json-web-token-vulnerabilites-attaques-courantes-et-bonnes-pratiques-securite/>, Consulté le 18/06/2025.
- [17] *La méthode agile*, <https://www.appvizer.fr/magazine/operations/gestion-de-projet/methode-agile>, Consulté le 09/03/2025.
- [18] *Méthode Scrum : présentation et avantages*, <https://asana.com/fr/resources/what-is-scrum>, Consulté le 07/03/2025.
- [19] *La méthode Scrum et ses bénéfices dans les développements web*, <https://www.bocasay.com/fr/methode-scrum-benefices-developpements-web/>, Consulté le 07/03/2025.
- [20] *UML 2*, <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>, Consulté le 01/04/2025.
- [21] *Comment définir les besoins fonctionnels en gestion de projet ?* <https://www.advaloris.ch/gestion-de-projet/definir-besoins-gestion-projet>, Consulté le 06/03/2025.
- [22] *Besoins Non-Fonctionnels : Guide d'Analyse et de Validation*, <https://www.fongecif.com/gestion-de-projet/besoins-non-fonctionnels-guide-danalyse-et-de-validation/>, Consulté le 07/03/2025.
- [23] *Diagramme de cas d'utilisation*, https://atefsd.weebly.com/uploads/5/0/3/6/503639/csi_02_chap02.pdf, Consulté le 01/03/2025.
- [24] *Backlog Produit : Rôle du Product owner*, <https://www.solidpepper.com/blog/backlog-produit-role-du-product-owner-exemples-2025>, Consulté le 02/04/2025.
- [25] *Définition d'un Epic en Agile Scrum*, <https://www.pm-coaching.org/blog/blog162>, Consulté le 20/03/2025.
- [26] *Scrumboard, qu'est-ce que c'est ?* <https://www.qrpinternational.fr/blog/methode-agile/scrumboard-ou-tableau-scrum-quest-ce-que-cest/>, Consulté le 26/04/2025.
- [27] *Burndown charts*, <https://www.atlassian.com/agile/tutorials/burndown-charts>, Consulté le 27/04/2025.
- [28] *Documentation officielle de Python*, <https://www.python.org/doc/>, Consulté le 10/04/2025.
- [29] *TypeScript*, <https://www.typescriptlang.org/>, Consulté le 10/04/2025.
- [30] *Spécification HTML - WHATWG*, <https://html.spec.whatwg.org/>, Consulté le 10/04/2025.

- [31] *CSS : Feuilles de style en cascade*, <https://developer.mozilla.org/fr/docs/Web/CSS>, Consulté le 10/04/2025.
- [32] *Documentation officielle de PostgreSQL*, <https://www.postgresql.org/docs/>, Consulté le 10/04/2025.
- [33] *Spécification YAML 1.2*, <https://yaml.org/spec/1.2/spec.html>, Consulté le 10/04/2025.
- [34] *Présentation du format JSON*, <https://www.json.org/json-fr.html>, Consulté le 10/04/2025.
- [35] *Manuel de référence de Bash*, <https://www.gnu.org/software/bash/manual/bash.html>, Consulté le 10/04/2025.
- [36] *Visual Studio Code*, <https://www.blogdumoderateur.com/tools/visual-studio-code/>, Consulté le 01/04/2025.
- [37] *Qu'est-ce que Gitlab, la plateforme de DevOps open source ?* <https://www.intelligence-artificielle-school.com/ecolet/technologies/gitlab-tout-connaître-sur-le-logiciel-open-source/>, Consulté le 01/04/2025.
- [38] *Qu'est-ce que Git ?* <https://www.next-decision.fr/wiki/qu-est-ce-que-git>, Consulté le 01/04/2025.
- [39] *Docker Desktop*, <https://docs.docker.com/desktop/>, Consulté le 11/04/2025.
- [40] *WebSockets : fonctionnement, vulnérabilités et bonnes pratiques sécurité*, <https://www.vaadata.com/blog/fr/websockets-fonctionnement-vulnerabilites-et-bonnes-pratiques-securite/>, Consulté le 15/04/2025.
- [41] *Overleaf*, <https://technopedagogie.uliege.be/produit/overleaf>, Consulté le 02/04/2025.
- [42] *StarUML*, <https://staruml.fr.softonic.com/>, Consulté le 02/04/2025.
- [43] *Introduction à OpenProject*, <https://www.openproject.org/docs/getting-started/openproject-introduction/>, Consulté le 05/04/2025.
- [44] *Postman*, <https://welovedevs.com/fr/articles/postman/>, Consulté le 02/04/2025.
- [45] *Swagger : développer des API plus confortablement*, <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/quest-ce-que-swagger/>, Consulté le 02/04/2025.

- [46] *Découvrir Selenium*, <https://www.blogdumoderateur.com/tools/selenium/>, Consulté le 20/04/2025.
- [47] *FastAPI – Site officiel*, <https://fastapi.tiangolo.com/>, Consulté le 12/04/2025.
- [48] *Angular – Site officiel*, <https://angular.fr/>, Consulté le 12/04/2025.
- [49] *Bootstrap – Site officiel*, <https://www.kernix.com/bootstrap/>, Consulté le 12/04/2025.
- [50] *Nmap : Network Discovery and Security Auditing*, <https://nmap.org/>, Consulté le 17/04/2025.
- [51] *PwnXSS - Stored XSS Vulnerability Scanner*, <https://github.com/pwn0sec/PwnXSS>, Consulté le 17/04/2025.
- [52] *wafw00f - Web Application Firewall Detection Tool*, <https://github.com/EnableSecurity/wafw00f>, Consulté le 17/04/2025.
- [53] *WhatWeb - Next Generation Web Scanner*, <https://github.com/urbanadventurer/WhatWeb>, Consulté le 17/04/2025.
- [54] *rabbitmq – Site officiel*, <https://www.rabbitmq.com/>, Consulté le 15/04/2025.
- [55] *MVC and MVVM Patterns in Angular*, <https://medium.com/front-end-world/mvc-and-mvvm-patterns-in-angular-7397e0bc7b07>, Consulté le 12/04/2025.
- [56] *Architecture FastAPI*, <https://www.geeksforgeeks.org/fastapi-architecture/>, Consulté le 13/04/2025.

Résumé

Ce projet de fin d'études, réalisé dans le cadre de l'obtention du diplôme national d'ingénieur en informatique à l'ENSIT, porte sur le développement d'une application web automatisant les tests de pénétration, fonctionnels et les audits SEO, dans le but de détecter plus efficacement les erreurs et les vulnérabilités et ainsi améliorer la qualité, la sécurité et la visibilité des sites web. L'automatisation permet d'optimiser la rapidité, la précision et la couverture des tests, tout en réduisant les erreurs humaines. Le développement a été mené selon la méthodologie Scrum, avec Angular pour l'interface utilisateur et FastAPI associé à PostgreSQL pour la partie serveur.

Mots clés : Tests de pénétration, tests fonctionnels, référencement naturel, SEO, automation, application web, Scrum, Angular, FastAPI, PostgreSQL.

Abstract

This final-year project, carried out as part of the requirements for obtaining the National Engineering Degree in Computer Science at ENSIT, focuses on the development of a web application that automates penetration testing, functional testing, and SEO audits. The goal is to more effectively detect errors and vulnerabilities in order to improve the quality, security, and visibility of websites. Automation helps optimize the speed, accuracy, and coverage of tests, while reducing human errors. The project was developed using the Scrum methodology, with Angular for the user interface and FastAPI with PostgreSQL for the backend.

Keywords : Penetration testing, functional testing, SEO, automation, web application, Scrum, Angular, FastAPI, PostgreSQL.

المالخص

يتمثل هذا المشروع في إطار مشروع تخرج للحصول على الشهادة الوطنية لمهندس في الإعلامية من المدرسة الوطنية العليا للمهندسين بتونس (ENSIT)، ويهدف إلى تطوير تطبيق ويب يقوم بأتمتة اختبارات الاختراق، والاختبارات الوظيفية، وتقديقات تحسين محركات البحث (SEO)، وذلك للكشف بشكل أكثر فعالية عن الأخطاء والثغرات بهدف تحسين جودة وأمان وظهور موقع الويب. تساهم الأتمتة في تحسين سرعة ودقة وشمولية الاختبارات، مع تقليل الأخطاء البشرية. تم تنفيذ المشروع باعتماد منهجية Scrum، باستخدام Angular للواجهة الأمامية، وFastAPI مع PostgreSQL في الجانب الخلفي.

الكلمات المفتاحية: اختبارات الاختراق، الاختبارات الوظيفية، تحسين محركات البحث، التشغيل الآلي، تطبيق ويب، PostgreSQL، FastAPI، Angular، Scrum.