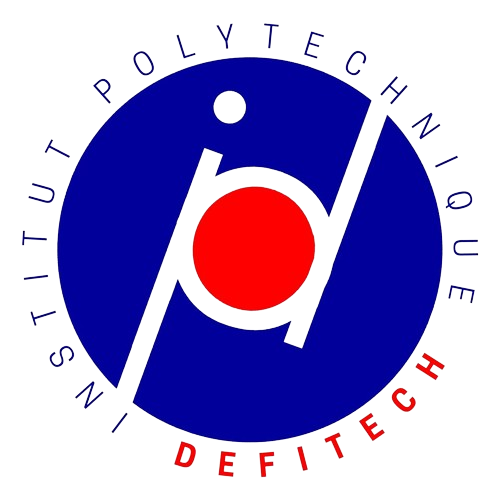
MINISTERE DE L’ENSEIGNEMENT

SUPÉRIEUR ET DE LA RECHERCHE



Institut Polytechnique DEFITECH

Tel : (+228) 22 26 25 25

Site web : [www.defitech.tg](http://www.defitech.tg)

PROJET DE CLASSE

DOMAINE : Science et technologies

MENTION : Informatique

SPÉCIALITÉ : Génie Logiciel

Thème:

PROGRAMMER UN SCANNER DE VULNÉRABILITÉS WEB EN PYTHON

Rédigé par :

Irène AMEDJI & Hubert ATSOU, Etudiants en 3e année Génie Logiciel.

Professeur:

Mr Idrissou GARA

SOMMAIRE

Glossaire

Liste des figures

Participants au projet

I-PRESENTATION DU PROJET

Introduction

Contexte

Problématique

Objectif

II-Résultat attendu

Etude et critique de l’existant

Proposition de solution

Choix de la solution

Périmètre du projet

Limites du produit

Description fonctionnelle des besoins

Fonctionnalités principales

Scénario d’utilisation

Contraintes et exigences

Évolutivité et maintenance

Planning prévisionnel de réalisation

Conclusion

III-ANALYSE ET CONCEPTION

Introduction

Présentation de la méthode d’analyse

Présentation de l’outil de modélisation

Conception détaillée du projet

Conclusion

IV-RÉALISATION ET MISE EN OEUVRE

Introduction

Matériels et outils de développement utilisés

Architecture matérielle et logicielle de l’application

Sécurité de l’application

Conclusion

V-GUIDE D’UTILISATION

Introduction

Description fonctionnelle de l’application

Présentation des différentes interfaces de l’application

Conclusion

VI-CONCLUSION GÉNÉRALE

GLOSSAIRE

SQL : Structured Query Language ;

HTTP :

LISTE DES FIGURES

*Tableau I : Liste des participants au projet*

*Tableau II : Planning prévisionnel du projet*

PARTICIPANTS AU PROJET:

*Tableau I Liste des participants au projet*

| Acteurs | Fonctions | Attribution |
| --- | --- | --- |
| Mlle Irène AMEDJI | Étudiante en 3e année licence option Génie Logiciel. | Chargée de l’étude et de la réalisation du projet. |
| Mr Hubert ATSOU | Étudiant en 3e année licence option Génie Logiciel. | Chargé de l’étude et de la réalisation du projet. |
| Mr Idrissou GARA | Enseignant à DEFITECH. | Superviseur |

I-PRESENTATION DU PROJET

1- Introduction

Dans un monde où les cybermenaces sont de plus en plus nombreuses, la sécurité des applications Web est primordiale. Cependant, même les meilleures pratiques de développement ne peuvent garantir l'absence totale de failles. C'est là que notre projet entre en jeu.

Nous allons explorer comment utiliser Python, l'un des langages de programmation les plus populaires et polyvalents, pour créer un outil capable de détecter les vulnérabilités potentielles dans les applications Web. Ce document vous guidera à travers les différentes étapes de développement, en mettant l'accent sur les technologies et les techniques utilisées. En fin de compte, notre objectif est de fournir une solution pratique et efficace pour renforcer la sécurité des applications Web.

2- Contexte

Nous avons constaté un besoin croissant d'outils automatisés capables d'identifier et de signaler rapidement les failles de sécurité dans les applications Web. Notre objectif est de fournir aux développeurs un moyen efficace de détecter et de corriger ces vulnérabilités avant qu'elles ne soient exploitées par des attaquants malveillants. C'est dans ce contexte que nous avons entrepris de créer un scanner de vulnérabilités Web en Python.

3-Problématique

Face à la prolifération des cyberattaques ciblant les applications Web, comment pouvons-nous concevoir un scanner de vulnérabilités efficace en Python pour aider les développeurs à détecter et à corriger rapidement les failles de sécurité, garantissant ainsi la protection des données sensibles et la confiance des utilisateurs ?

Telles sont les questions auxquelles nous tenterons de répondre tout au long de la réalisation de ce projet.

4-Objectifs du projet

En général, ce projet vise à programmer un scanner de vulnérabilité web en Python. En terme d’objectifs spécifiques, il nous faut :

* Concevoir un scanner de vulnérabilités Web en Python capable d'identifier un large éventail de failles de sécurité courantes telles que les injections SQL, les cross-site scripting (XSS), et les failles d'inclusion de fichiers.
* Intégrer des mécanismes d'analyse automatique des réponses HTTP pour différencier les réponses normales des réponses indiquant une vulnérabilité potentielle.
* Développer une interface utilisateur conviviale permettant aux utilisateurs de configurer et de lancer des analyses de manière intuitive, ainsi que de visualiser les résultats de manière claire et détaillée.
* Implémenter des fonctionnalités avancées telles que la possibilité de vérifier la conformité aux bonnes pratiques de sécurité et de générer des rapports complets pour faciliter la correction des vulnérabilités détectées.
* Fournir une documentation complète et des exemples d'utilisation pour faciliter l'adoption et la compréhension du scanner par les développeurs et les professionnels de la sécurité.

II-Résultat attendu

1- Etude et critique de l’existant

Afin de mener à bien ce projet, nous avons fait des recherches sur internet et testé quelques logiciels pour mieux comprendre le fonctionnement du système existant et mieux cerner les réels enjeux auxquels nous ferons face durant la réalisation du projet. Ces applications ou logiciels disposent de 4 fonctionnalités dont on peut citer : Scanne de sites web, grande base de données, mises à jour régulière, injection payloads, etc…

Cependant, en raison de la complexité de ces logiciels et les interfaces qui ne sont pas intuitives, on assiste à une non utilisation. De plus, c’est complexe à configurer pour les débutants. Cela peut entraîner une réception de fausse réponses. En somme, il est évident que les systèmes actuels présentent des problèmes d'accessibilité, de gestion de l'espace, de fausses manipulations si utilisés de manière incorrecte ou, de consommation de ressources élevées, ... Ces défis soulignent alors l'importance de considérer une approche de proposition d’autres conceptions d’application pour améliorer l'efficacité et la sécurité de ce travail.

2- Proposition de solution

a- OWASP ZAP (Zed Attack Proxy)

Avantages

- Interface utilisateur conviviale.

- Grande communauté et support actif.

- Fonctionnalités avancées telles que l'exploration automatisée, l'injection de payloads, etc.

Inconvénients

- Peut être complexe à configurer pour les débutants.

- Consomme beaucoup de ressources système lors de l'exécution de scans intensifs.

- Certaines fonctionnalités avancées peuvent nécessiter une expertise approfondie pour une utilisation efficace.

b- Nikto

Avantages

- Simple à utiliser, avec une ligne de commande claire.

- Scanne rapidement les sites web pour détecter les vulnérabilités courantes.

- Grande base de données de vulnérabilités et mises à jour régulières.

Inconvénients

- Moins de fonctionnalités avancées par rapport à d'autres outils.

- Peut générer des faux positifs.

- Pas aussi efficace pour les scans complexes ou personnalisés.

c- SQLMap

Avantages

- Spécialisé dans la détection des vulnérabilités SQL Injection.

- Automatise le processus de détection et d'exploitation des failles SQL.

- Prise en charge de nombreux types de bases de données et de payloads.

Inconvénients

- Principalement axé sur les injections SQL, donc moins polyvalent pour d'autres types de vulnérabilités.

- Nécessite une connaissance approfondie des injections SQL pour une utilisation efficace.

- Peut entraîner des fausses manipulations si utilisé de manière incorrecte ou non contrôlée.

3- Choix de la solution

En prenant en considération les avantages et les inconvénients de chaque solution, la décision d'opter pour la deuxième proposition, qui est la réalisation d'une application web personnalisée est optimale. Cette option sera plus adaptée aux besoins particuliers de la division comptabilité et prendra en compte ses contraintes. Elle permettra également une plus grande flexibilité et une meilleure évolutivité par rapport à la première solution. Pour ce faire, nous avons utilisé le Framework Django associé à Python comme langage de programmation côté back-end, python pur et certaines de ses librairies.

Nous appellerons cette futur application : VulScan.

4- Périmètre du projet

Le projet de programmation de scanner de vulnérabilité web en python vise à une mise à jour régulière et forte connaissance en la matière. Il comprend un stockage de données, une interface intuitive, un algorithme puissant et consommation de moins de ressources. Dans cette section, le périmètre de notre projet sera clairement défini et les limites bien établies, ce qui permettra de garantir une mise en œuvre efficace et réussie.

5- Limite du produit

Le projet de programmation de scanner de vulnérabilité web en python est un projet essentiel pour détecter les failles dans les sites web. La mise en place d'une application web permettra de faciliter l'accès et l’opération du scansur ces sites. Cependant, il est important de noter que ce projet présente quelques limites :

❖ complexité des systèmes : la mise en place d'un système de scanner peut être complexe, en particulier si ces sites web en question existantes sont volumineuses et gigantesques ;

❖ dépendance aux technologies : la programmation de cet outil implique une dépendance accrue aux technologies. Des problèmes techniques tels que les pannes de système, les incompatibilités logicielles ou les problèmes de stockage peuvent entraver l'accès aux informations et compromettre la continuité des opérations.

7- Description fonctionnelle des besoins

Ce projet a pour objectif de programmer un scanner de vulnérabilité web en Python.

Cette application doit être fiable, sûre, évolutive et accessible au public. Dans cette description, nous allons identifier les fonctionnalités principales, les scénarios d'utilisation, les contraintes et les exigences de ce projet. Elle prend également en compte les conditions d'évolutivité et de maintenance de l’application.

8- Fonctionnalités principales

Les fonctionnalités principales dont disposera notre application sont les suivantes :

❖ Entrer l’url complète du site web à scanner ;

❖ Voir les résultats du scan ;

❖ numériser les résultats du scan ;

❖ Enregistrer les résultats dans l’historique;

❖ Recolter les commentaires des utilisateurs.

9- Scénario d’utilisation

Notre application doit être en mesure de répondre à différents scénarios d'utilisation, tels que :

❖ l’accès facile et rapide à l’application;

❖ la protection des données ;

❖ la gestion optimisée de l’espace ;

❖ Réponses rapides.

10- Contraintes et exigences

Notre projet de digitalisation comporte un certain nombre de contraintes et d’exigences à prendre en compte. Il s’agit de :

❖ la qualité du programme d’exécution : le programme pour scanner les sites doivent être efficaces et réduites de moins d’erreurs possibles;

❖ la sécurité des données : les données des utilisateurs ne doivent pas être divulguées chez les autres ;

❖ Une réponse claire et rapide : Quand l’utilisateur va entrer son URL pour scanner son site, sa réponse attendue doit être claire et rapide.

11- Évolutivité et maintenance

Pour assurer l'évolutivité et la maintenance de cette application de scanner de vulnérabilité web en Python il est important de :

❖ planifier des mises à jour régulières pour maintenir la compatibilité avec les nouveaux systèmes et technologies ;

❖ disposer d'une équipe dédiée pour gérer et résoudre les problèmes techniques ;

❖ veiller à ce que les données soient sauvegardées régulièrement pour éviter toute perte de données;

❖ recueillir régulièrement les retours des utilisateurs pour identifier les problèmes et les améliorations à apporter à l'application.

12- Planning previsionnel de realisation

La planification de projet correspond à l’organisation des tâches à réaliser tout le long du projet. Il comporte les dates de début et les deadlines de chaque tâche, il joue ainsi un rôle important dans le pilotage et le suivi de projet. Le tableau ci-dessous résume le planning prévisionnel du projet.

*Tableau I : Planning prévisionnel du projet*

| N° | TACHES | Date de début | Date de fin | Durée (jours) |
| --- | --- | --- | --- | --- |
| 1 | Prise de contact et de connaissance du projet | 02 avril 2024 | 03 avril 2024 | 01 jour |
| 2 | Recueil d’information et Élaboration du cahier des charges | 04 avril 2024 | 09 avril 2024 | 05 jours |
| 3 | Validation et correction du cahier des charges | 09 avril 2024 | 11 avril 2024 | 02 jours |
| 4 | Apprentissage des outils logiciels et techniques à utiliser | 12 avril 2024 | 15 avril 2024 | 03 jour |
| 5 | Développement front-end | 16 avril 2024 | 17 avril 2024 | 01 jour |
| 6 | Développemnt du back-end & test au fur et à mésure | 18 avril 2024 | 27 avril 2024 | 09 jours |
| 7 | Hébergement/ mise en ligne | 28 avril 2024 | 28 avril 2024 | 01 jour |
|  | Total |  |  | 26 jours |

13- Conclusion

Ce chapitre a été consacré essentiellement à la présentation du cahier des charges. L’objectif général du projet est de programmer un scanner de vulnérabilité web en Python. Le prochain chapitre va être consacré à l’analyse et conception de la solution de programmation.

II.ANALYSE ET CONCEPTION

1- Introduction

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système. La modélisation est la conception et l’utilisation d’un modèle. Modéliser un système avant sa réalisation, permet de mieux comprendre le fonctionnement du système. C’est également le meilleur moyen de maîtriser sa complexité et d’assurer sa cohérence. Ce chapitre nous aidera à présenter d’abord la méthode d’analyse à utiliser ensuite l’outil de modélisation et enfin finir avec une conception détaillée de notre solution.

2-Présentation de la méthode d’analyse

2-1 Présentation des méthodes

Les méthodes d'analyse sont essentielles dans le développement d'applications informatiques, offrant une approche systématique pour comprendre les besoins et contraintes du projet. Elles guident la conception et l'implémentation de solutions logicielles adaptées, en collectant, analysant et spécifiant les exigences du système, ainsi qu'en planifiant et organisant les étapes de développement. Parmi ces méthodes, certaines suivent une approche séquentielle, comme la méthode en cascade, où chaque étape est linéaire et prédéfinie. D'autres adoptent une approche itérative et incrémentale, comme l'Unified Process, permettant des ajustements et des améliorations continus par cycles répétés. En outre, certaines méthodes encouragent la participation des utilisateurs finaux, favorisant la collaboration étroite entre les analystes et les utilisateurs pour comprendre et valider les fonctionnalités du système. De toutes ces méthodes d’analyse, nous pouvons citer :

❖ Méthodes traditionnelles séquentielles : modèle en cascade ou en V;

❖ Méthodes itératives et incrémentales : Unified Process.

2-1-Méthode séquentielles

Les méthodologies séquentielles ne sont que des approches traditionnelles bien établies dans le domaine du développement d'applications informatiques. Elles suivent une séquence linéaire d'étapes distinctes, où chaque phase est réalisée dans un ordre fixe et avec des résultats définis avant de passer à la suivante. Ces méthodologies accordent une importance particulière à la planification minutieuse, à la documentation approfondie et au suivi précis de l'avancement du projet.

Voici une présentation des points positifs et négatifs de ces méthodologies [7] :

❖ AVANTAGES :

▪ structuration claire : les méthodologies séquentielles offrent une structure bien définie avec des étapes séquentielles, ce qui facilite la planification, la coordination et le suivi du projet ;

▪ documentation approfondie : avec un cahier des charges détaillé et une planification précise dès le départ, Waterfall permet d’avoir une idée précise du budget nécessaire à la réalisation du projet et de la date de livraison.

❖ INCONVENIENTS :

▪ la méthodologie Waterfall ne convient pas aux projets plus larges et/ou complexes ;

▪ flexibilité limitée : les méthodologies séquentielles peuvent être moins flexibles lorsqu'il est nécessaire d'apporter des changements ou des ajustements importants en cours de projet, car revenir en arrière dans les étapes précédentes peut être complexe.

2-1-2- Méthodes itératives

Les méthodes itératives sont des approches de développement d'applications informatiques qui se caractérisent par une progression itérative et incrémentale du projet. Elles mettent l'accent sur la flexibilité, l'adaptabilité et la collaboration étroite avec les parties prenantes. L’une de ces méthodologies se trouve être l’Unified Process (UP) ou Processus Unifié en français. Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques [8]. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents niveaux de compétences et à différentes tailles de l'entreprise. Le cycle de développement du processus unifié organise les tâches et les itérations en quatre phases :

❖ l’Inception1 : courte pour estimer, planifier, partager une même vision du problème, et engager les hostilités ;

❖ l’Elaboration : développement de façon incrémentale l’architecture du noyau, les risques et la plupart des besoins identifiés ;

❖ la Construction : élaboration des sous-ensembles exécutables et stables du produit final ; ❖ la Transition : mise à la disposition des utilisateurs de la version bêta. Les méthodes Unified Process offrent plusieurs avantages et inconvénients qu'il convient de prendre en compte :

❖ AVANTAGES :

▪ adaptabilité : l'approche itérative et incrémentale de l'UP permet de s'adapter aux changements et aux évolutions des exigences tout au long du projet.

▪ structuration : l'UP fournit une structure claire et un cadre de travail pour le développement d'applications informatiques ;

▪ collaboration : l'UP encourage la collaboration étroite entre les parties prenantes, favorisant ainsi une meilleure compréhension des besoins et une prise de décision collective ; ▪ qualité du produit final : grâce aux itérations et aux tests réguliers, l'UP permet de développer un produit final de meilleure qualité, répondant aux attentes des utilisateurs [8]. ❖ INCONVENIENTS :

▪ temps et coûts : les itérations supplémentaires dans l'UP peuvent entraîner une augmentation des délais et des coûts du projet, notamment si la gestion des itérations n'est pas bien gérée ;

▪ nécessité de compétences : l'UP nécessite des compétences solides en gestion de projet et en développement logiciel pour être mis en œuvre efficacement.

1-2- Choix de la méthode

1-2-1- Le processus 2TUP

Après description et études des différentes méthodes, nous avons retenu le processus 2TUP pour les raisons suivantes :

❖ peu exigeant par rapport à la disponibilité du client ;

❖ la séparation des besoins fonctionnels/Architecturaux ;

❖ permet la modélisation graphique (UML) ;

❖ le besoin de documentation de la conception pour faciliter la maintenance future.

2TUP (Two Track Unified Process), est un processus de développement logiciel qui met en œuvre la méthode du processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de trois phases essentielles :

❖ la branche gauche (fonctionnelle) : Capitalise la connaissance du métier de l’entreprise. Elle constitue généralement un investissement pour le moyen et le long terme. Les fonctions du système d’information sont en effet indépendantes des technologies utilisées. Cette branche comporte les étapes suivantes :

✓ la capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs ;

✓ l’analyse.

❖ la branche droite (architecture technique) : Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l’être en effet indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

✓ la capture des besoins techniques ;

✓ la conception générique

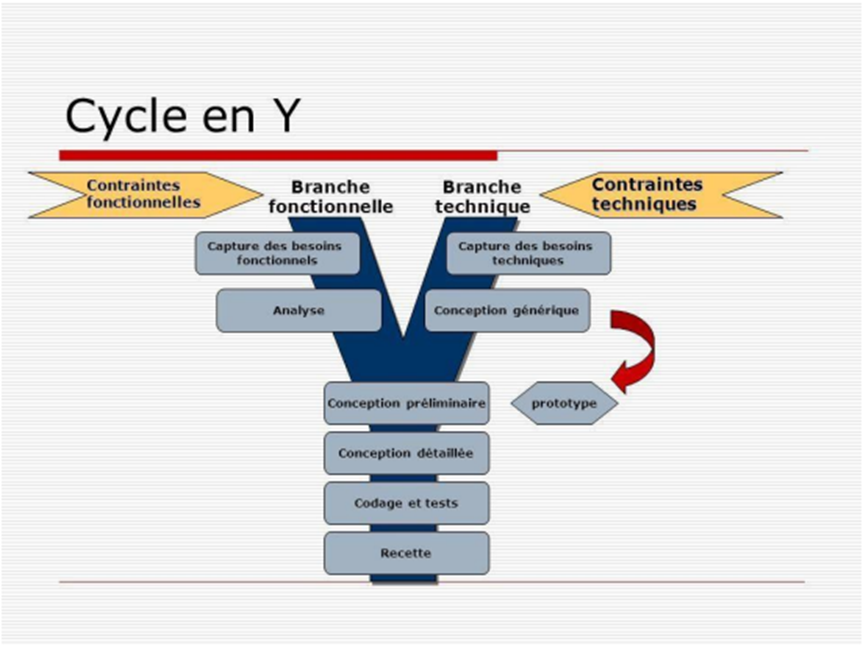
❖ la branche du milieu : À l’issue des évolutions du modèle fonctionnel et de l’architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l’obtention d’un processus en forme de Y. Cette branche comporte les étapes suivantes :

✓ la conception préliminaire ;

✓ la conception détaillée ;

✓ le codage ;

✓ l’intégration.



1-2-2- Langage de modélisation avec UML

Pour modéliser notre système, nous avons besoin d’utiliser une méthode. Il a donc fallu associer un processus au langage UML.

LANGAGE + PROCESSUS = METHODE D’ANALYSE.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent. L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientée objet. Fin 1997, UML est devenu une norme OMG.



UML est actuellement à sa version 2.5.1 proposant 14 types de diagrammes séparés en trois catégories à savoir :

❖ les diagrammes de structure ou diagramme statistiques rassemblant :

✓ le diagramme de classe ;

✓ le diagramme d’objet ;

✓ le diagramme de composant ;

✓ le diagramme de déploiement ;

✓ le diagramme de paquet ;

✓ le diagramme de structure composite ;

✓ le diagramme de profils.

❖ les diagrammes de comportement rassemblant :

✓ le diagramme des cas d’utilisation ;

✓ le diagramme états-transitions ;

✓ le diagramme d’activité.

❖ les diagrammes d’interaction ou diagrammes dynamiques :

✓ le diagramme de séquence ;

✓ le diagramme de communication ;

✓ le diagramme global d’interaction ;

✓ le diagramme de temps.

RELATION ENTRE UML et 2TUP

Le processus 2TUP s’appuie sur UML tout au long du cycle de développement, ainsi les différents diagrammes de ce dernier permettent, grâce à leur facilité et clarté, de bien modéliser le système à chaque étape.

| Eléments du processus 2TUP | Diagrammes UML |
| --- | --- |
| Cas des besoins fonctionnels | - Diagramme de cas d’utilisation  - Diagramme de séquence  - Diagramme de contexte statique |
| Analyse fonctionnelle | - Diagramme de classe  - Diagramme d’état transition |
| Capture de besoins techniques | - Diagramme de cas d’utilisation |
| Conception générique | - Diagramme de déploiement |
| Conception préliminaire | - Diagramme de déploiement  - Diagramme de composant |
| Conception détaillées | - Diagramme de classe  - Diagramme de séquence  - Diagramme de collaboration  - Diagramme d’état  - Diagramme de composante  - Diagramme d’activités |

3- Présentation de l’outil de modélisation

Les logiciels de modélisation de données sont des applications essentielles qui facilitent la création de structures de base de données à l'aide de modèles visuellement attrayants et simples. Ils agissent comme des plans détaillés, guidant la façon dont un projet sera organisé et géré pour atteindre ses objectifs prédéfinis. La modélisation des données est souvent comparée à l'activité d'un architecte créant un plan avant la construction d'une maison. Comme dans le cas d'une construction, un projet informatique nécessite une planification minutieuse qui prend en compte la structure du projet et les besoins définis. Ainsi, plusieurs outils, tels que PowerAMC, StarUML et Diagrams.net, sont disponibles pour améliorer l'efficacité de ce processus. Notre choix s'est porté sur PowerAMC pour ses fonctionnalités et son adaptabilité aux besoins spécifiques de notre projet.



I-4- Conception détaillée du projet

Cette partie va nous permettre de modéliser les différents diagrammes qui montrent les besoins et les interactions entre les utilisateurs et l’application. Les diagrammes qui vont nous permettre de réaliser notre solution sont les suivants :

❖ Diagramme de cas d’utilisation ;

❖ Diagramme de classes ;

❖ Diagramme d’activité ;

❖ Diagramme de séquence ;

❖ Diagramme de déploiement.

4-1- Diagramme de cas d’utilisation

4-1-1- Présentation

Le diagramme de cas d'utilisation est l'un des nombreux diagrammes utilisés dans le domaine de l'ingénierie logicielle pour modéliser les interactions entre les acteurs (utilisateurs) et un système. Il fait partie des premiers diagrammes de notre modélisation car il permet de représenter graphiquement les fonctionnalités principales d'un système. Le diagramme de cas d'utilisation met l'accent sur les interactions entre les acteurs et le système. Il fournit une vision globale des fonctionnalités du système du point de vue des utilisateurs et des différentes actions qu'ils peuvent entreprendre.

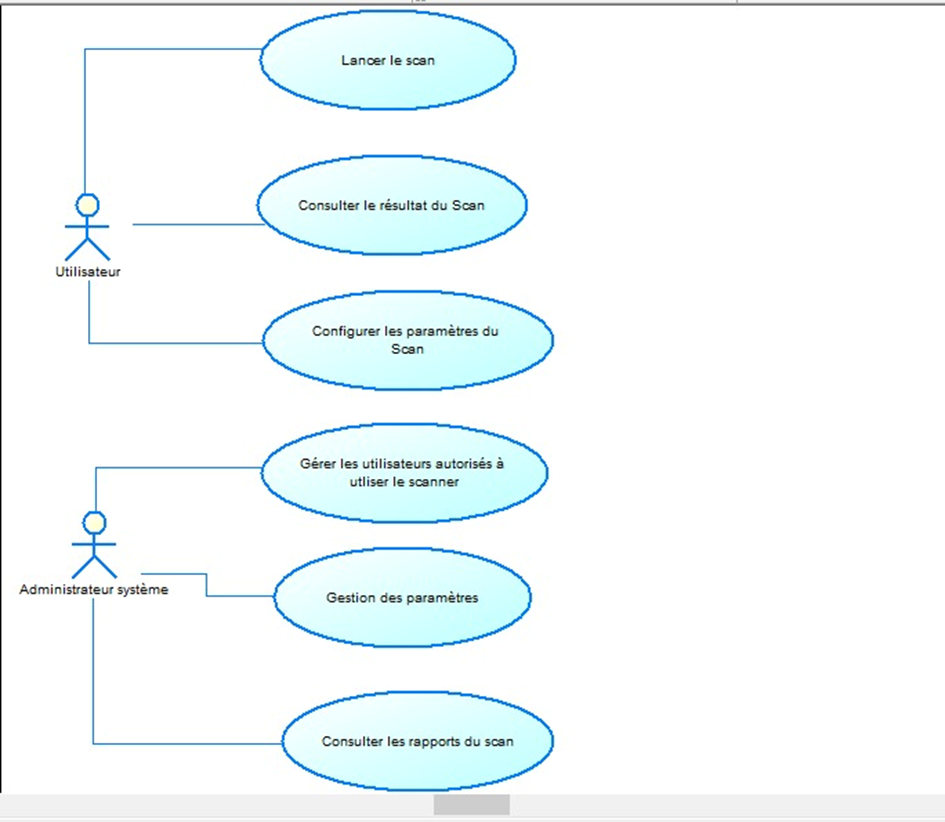
Les acteurs de notre application sont :

❖L’Utilisateur

Administrateur

4-1-2- Modélisation des diagrammes

❖ Diagramme des cas d’utilisation « global » Ce diagramme global fournit une vue d'ensemble des interactions entre les acteurs et le système, mettant en évidence les fonctionnalités principales du système d'un point de vue de chaque acteur.



4-2- Diagramme de classes

4-2-1- Présentation

Un diagramme de classes est un outil de modélisation essentiel en UML pour représenter les classes, objets, attributs, méthodes et relations dans un système logiciel. Il permet de visualiser la structure statique d'un système. Selon la complexité, un seul ou plusieurs diagrammes peuvent être utilisés pour modéliser les composants du système. Ces diagrammes sont utiles à différentes étapes de conception. En analyse, ils aident à comprendre les exigences du domaine et à identifier les composants. Dans le langage UML, une classe représente un objet ou un ensemble d'objets qui partagent une structure et un comportement commun. Les classes ou instances de classes sont des éléments de modèle communs dans les diagrammes UML.

4-2-2- Modélisation du diagramme

Le diagramme de classes illustré ci-dessous représente toutes entités utilisées dans le système ainsi que les relations entre elles.

