République Tunisienne

*****

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

*****

Université Virtuelle de Tunis

RAPPORT DU PROJET DE FIN D'ÉTUDES

Pour l'obtention du diplôme de

Mastère Professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)

Etude et développement d'une plateforme d'analyse des fichiers logs

Réalisé par :

Basma Mezni

Encadrants

Mr Imed Jabri

Mr Nabil Hosni

Année Universitaire 2014-2015

Dédicaces

A mes parents,

Pour tout leur soutien, pour tous leurs sacrifices, pour leur amour et pour tout l'enseignement qu'ils m'ont transmis.

A mes frères et mes sœurs,

Avec tous les souhaits d'un grand succès dans leur vie.

A mon fiancé

Pour leurs encouragements incessants.

A mes amis et mes collègues,

Qui ont contribué à mon épanouissement .Merci d'être toujours près de moi, merci de m'avoir aidé chaque jour à avancer.

MEZNI Basma

Remerciements

Je tiens à exprimer toute ma grande reconnaissance à l'endroit de mes encadreurs **M. Imed JABRI**, maître de conférence à Ecole Nationale Supérieure d'Ingénieurs de Tunis (ENSIT) et **M. Nabil HOSNI**, chef service à l'Agence Nationale de la Sécurité Informatique (ANSI) qui n'ont cessé de suivre chacun de mes pas tout au long de ce projet, pour ses encouragements, ses conseils ses rigueurs dans le travail et surtout ses qualités humaines qui nous ont permis de travailler avec confiance dans un climat détendu.

Tous mes remerciements à tout le **corps enseignant d'UVT** pour leurs multiples efforts et sacrifices déployés pour nous garantir une bonne formation.

Enfin, je témoigne ici à tous **les membres du jury**, toute ma reconnaissance et le respect que j'ai pour eux pour avoir accepté d'évaluer mon travail.

Sommaire

Introduction générale 1
Chapitre 1 : Etat de l'art 3
I. Présentation de l'organisme d'accueil 3
II. Etude théorique 4
1. Les fichiers journaux 4
2. Les types de fichiers journaux 4
3. Format de fichier journal 4
4. Code de réponse de serveur 5
5. Dictionnaire d'attaques 7
III. Présentation de projet 7
1. Problématique 7
2. Etude de l'existant 8
3. Critique de l'existant 9
4. Solution proposée 9
IV. Méthodologie à adopter 11
V. Choix de méthodologie à adopter 12
VI. Choix de langage de modélisation 13
Analyse et spécification des besoins 14
I. Spécification des besoins 14
1. Les besoins fonctionnels 14
2. Les besoins non fonctionnels 15
3. Les besoins architecturaux 15
a. Architecture centralisée 15
b. Architecture client/serveur 16
II. Modélisation des besoins 18
1. Les acteurs 18
2. Diagramme de cas d'utilisation 19
a. Cas d'utilisation «gérer investigateur » pour administrateur 20
b. Cas d'utilisation «gérer incident » pour administrateur 21
c. Cas d'utilisation «gérer profil » pour utilisateur 22

d. Cas d'utilisation «analyser incident » pour investigateur 23
e. Cas d'utilisation «gérer rapport » pour investigateur 23
1. Description des cas d'utilisation 24
a. Description du cas d'utilisation : « S'authentifier » 24
b. Description du cas d'utilisation : « Afficher les statistiques des attaques » 25
c. Description du cas d'utilisation : « Inscrire à la plateforme » 25
d. Description du cas d'utilisation : «Affecter investigateur à un incident » 26
e. Description du cas d'utilisation : « Analyser incident » 27
f. Description du cas d'utilisation : « Déclarer incident » 28
Conception 30
I. Architecture de système 30
II. Déploiement 31
III. Conception 31
1. Diagramme de classe 31
a. Représentation des classes 31
b. Conception de la base de données 32
2. Diagramme de séquence 33
a. Diagramme de séquence de cas d'utilisation « s'authentifier » 33
b. Diagramme de séquence de cas d'utilisation «inscrire à la plateforme » 34
c. Diagramme de séquence de cas d'utilisation «Déclarer incident » 34
d. Diagramme de séquence de cas d'utilisation «affecter investigateur à un incident » 35
e. Diagramme de séquence de cas d'utilisation «analyser incident » 36
f. Diagramme de séquence de cas d'utilisation « afficher statistiques des attaques » .. 36
Réalisation 38
I. Environnement matériel 38
II. Environnement logiciel 38
III. Présentation des interfaces 38
1. Interface d'inscription 39
2. Interface d'authentification 39
3. Interface de déclaration 40
4. Interface de travail 42
5. Interface de gestion des incidents 42
6. Interface d'affectation un investigateur à un incident 43

7. Interface de consultation un dictionnaire d’attaque 43
8. Interface de statistique 44
Conclusion générale 46
Webographie 47

Liste des figures

Figure 1:Extrait d'un fichier log de format ELF 5
Figure 2:Extrait d'un fichier log de format CLF 5
Figure 3: Les phases du traitement des fichiers logs 10
Figure 4: Le processus 2TUP 12
Figure 5: Architecture 2-tiers 16
Figure 6: Architecture 3-tiers 17
Figure 7: Diagramme de cas d'utilisation général 20
Figure 8: Diagramme de cas d'utilisation "gérer investigateur" 21
Figure 9: Diagramme de cas d'utilisation "gérer incident" 22
Figure 10: Diagramme de cas d'utilisation « gérer profil » 22
Figure 11: Diagramme de cas d'utilisation « analyser incident » 23
Figure 12: Diagramme de cas d'utilisation « gérer rapport » 23
Figure 13: Architecture de système 30
Figure 14 : Diagramme de déploiement 31
Figure 15: Diagramme de classe 32
Figure 16 : Diagramme de séquence de cas d'utilisation « s'authentifier » 33
Figure 17: Diagramme de séquence de cas d'utilisation «inscrire à la plateforme » 34
Figure 18: Diagramme de séquence de cas d'utilisation «Déclarer incident » 35
Figure 19: Diagramme de séquence de cas d'utilisation «affecter investigateur » 35
Figure 20: Diagramme de séquence de cas d'utilisation «analyser incident » 36
Figure 21:Diagramme de séquence de cas d'utilisation « afficher statistiques» 37
Figure 22 : Interface d'inscription 39
Figure 23 : Interface d'authentification 40
Figure 24 : Interface de déclaration un incident (information générale) 40
Figure 25: Interface de déclaration un incident (type de l'incident) 41
Figure 26: Interface espace de travail de l'administrateur 42
Figure 27: Interface de gestion des incidents 43
Figure 28: Interface d'affectation 43
Figure 29: Interface de consultation dictionnaire d'attaque 44
Figure 30: Interface de statistique 44

Liste des tableaux

Tableau 1:Liste des codes de réponse de serveur [2] 7
Tableau 2: Comparaison entre les principales méthodologies de développement [8] 12
Tableau 3 : Comparaison entre les différentes architectures 18
Tableau 4 : Description du cas d'utilisation « authentifier utilisateur » 24
Tableau 5: description du cas d'utilisation « consulter les statistiques». 25
Tableau 6: Description du cas d'utilisation « inscrire à la plateforme » 26
Tableau 7: Description du cas d'utilisation : «Affecter investigateur à un incident » 27
Tableau 8: Description du cas d'utilisation : « Analyser incident » 28
Tableau 9: Description du cas d'utilisation : « Déclarer incident » 29

Introduction générale

La pérennité de l'entreprise passe par une disponibilité permanente de son système d'information. Cette réalité influe de nos jours le comportement des entreprises qui devient de plus en plus mature dans les investissements de la sécurité du système d'information qui est un élément absolument vital.

Les efforts de sécurisation ne peuvent pas être efficaces que si ces investissements sont correctement étudiés et ciblés, en mettant en place les moyens de protection qui apportent un niveau de sécurité favorable adapté aux enjeux de l'entreprise.

Généralement, et pour des besoins d'optimisation, certaines entreprises utilisent un service appelé « SYSLOG » afin de centraliser les journaux d'évènements du réseau qui sont envoyés par des imprimantes, serveurs, routeurs, firewalls, IDS et IPS dans un serveur SYSLOG. Ce dernier facilite la consultation de l'ensemble des fichiers de journalisation dans une mission d'audit.

Le contexte de notre projet est de réaliser une plateforme qui permet d'investiguer sur un incident afin de déterminer les responsables et les scénarios d'attaques en partant les traces (preuves ou logs) nécessaires.

Ce rapport présente l'ensemble des étapes suivies pour développer la solution. Il contient quatre chapitres organisés.

Le premier chapitre sera dédié à la présentation de l'état de l'art. Nous allons tout d'abord présenter l'organisme d'accueil et le projet. Ensuite nous passerons à l'étude et à la critique de l'existant pour enfin proposer une solution adéquate. La méthodologie utilisée y sera également définie pour nous permettre de réaliser convenablement notre travail.

Le second chapitre sera consacré sur l'analyse des besoins fonctionnels et non fonctionnels. Nous modéliserons les besoins des utilisateurs via les diagrammes de cas d'utilisation.

Le troisième chapitre sera intitulé conception et fera dans un premier temps une étude préliminaire en présentant l'architecture de la solution proposée précédemment. Dans un

second temps, en se référant à la méthodologie de travail choisie, elle illustrera la plupart des diagrammes de conception.
Le quatrième chapitre quant à lui sera réservé à la réalisation. Celui-ci, passera en revue l'environnement de travail et les résultats obtenus.

Pour finir, une conclusion générale de tout le rapport sera nécessaire où nous proposerons les éventuelles améliorations susceptibles d'être ajoutées ultérieurement.

Chapitre 1 : Etat de l'art

Introduction

Dans ce premier chapitre, nous allons introduire le cadre du projet, à savoir l'organisme d'accueil. Ensuite, nous passerons à la présentation du sujet, et pour terminer, nous spécifierons la méthodologie du développement et le langage de modélisation à suivre durant notre projet.

I. Présentation de l'organisme d'accueil

Notre projet a été réalisé au sein de L'ANSI (L'Agence Nationale de la sécurité Informatique) qui est un organisme public à caractère non administrative crée le 04 février 2004 sous la loi n°5-2004.

L'agence effectue un contrôle général des systèmes informatiques et des réseaux relevant des divers organismes publics et privés, elle est chargée des missions suivantes:

- Veiller à l'exécution des orientations nationales et de la stratégie générale en systèmes de sécurité des systèmes informatiques et des réseaux,
- Suivre l'exécution des plans et des programmes relatifs à la sécurité informatique dans le secteur public à l'exception des applications particulières à la défense et à la sécurité nationale et assurer la coordination entre les intervenants dans ce domaine,
- Assurer la veille technologique dans le domaine de la sécurité informatique,
- Etablir des normes spécifiques à la sécurité informatique et élaborer des guides techniques en l'objet et procéder à leur publication,
- Œuvrer pour encourager le développement de solutions nationales dans le domaine de la sécurité informatique et à les promouvoir conformément aux priorités et aux programmes qui seront fixés par l'agence,
- Participer à la consolidation de la formation et du recyclage dans le domaine de la sécurité informatique,
- Veiller à l'exécution des réglementations relatives à l'obligation de l'audit périodique de la sécurité des systèmes informatiques et des réseaux.

II. Etude théorique

1. Les fichiers journaux

Chaque action d'un système informatique (ouverture d'une session, installation d'un programme, navigation sur Internet...) produit un fichier log. Ces fichiers textes listent chronologiquement les évènements exécutés par un serveur ou une application informatique. Ils s'avèrent utiles pour comprendre la provenance d'une erreur en cas de bug.

S'agissant d'un serveur Web, un fichier log va enregistrer la date et l'heure de la tentative d'accès, l'adresse IP du client, le fichier cible, le système d'exploitation utilisé, le navigateur, la réponse du serveur à cette requête, éventuellement le type d'erreur rencontré, etc. [1]

2. Les types de fichiers journaux

Les systèmes informatiques génèrent de nombreux types de fichiers journaux afin de fournir les informations essentielles sur le système. Parmi ses types, on peut citer les exemples suivants :

- Les fichiers log issus des serveurs,
- Les fichiers log issus des sites web,
- Les fichiers log issus des systèmes de détection d'intrusion,
- Les fichiers log issus des systèmes de surveillance de réseau,
- Les fichiers log issus des pare-feu,
- Les fichiers log d'accès,
- Les fichiers log d'erreurs.

3. Format de fichier journal

La structure et le contenu de fichier log permettent d'obtenir de plus amples informations après certains traitements. Il existe de type de format CLF (Common LogFile) et ELF (Extended LogFormat), ce dernier est le plus répandu de fichier log.

Dans cette partie, nous avons expliqué ces deux formats à travers deux exemples.

```
161.31.132.116 - - [21/Dec/2001:08:42:55 -0500] "GET /home.htm HTTP/1.0" 200 4392
"http://fr.search.yahoo.com/fr?p=peinture" "Mozilla/4.7 [en] (Win98)"
```

Figure 1:Extrait d'un fichier log de format ELF

```
161.31.132.116- - [21/Dec/2001:08:42:55 -0500] "GET /home.htm HTTP/1.0" 200 4392
```

Figure 2:Extrait d'un fichier log de format CLF

La figure 2 montre un exemple de format CLF de fichier log, ce format est composé, pour chaque requête d'un utilisateur par les champs suivants :

- 161.31.132.116 : L'adresse IP de l'utilisateur qui a envoyé la requête.
- [21/Dec/2001:08:42:55 -0500] : La date est l'heure de la requête.
- GET /home.htm HTTP/1.0 : La méthode de requête, la page demandée et le protocole utilisé.
- 200 : Le numéro de code de réponse de serveur.
- 4392 : La taille de la page demandée en octets.

Le format ELF a les mêmes champs que le format CLF, en rajoutant d'autre paramètres pour plus de détails, tel que :

- http://fr.search.yahoo.com/fr?p=peinture : La page de référence qui à partir de laquelle la requête est lancée.
- Mozilla/4.7 [en] (Win98) : Le navigateur et le système d'exploitation utilisés par l'utilisateur.

Dans l'exemple de la figure 1, la machine ayant l'adresse IP **161.31.132.116** a émis la requête **GET /home.htm HTTP/1.0** le **21 décembre 2001 à 8 heures, 42 minutes et 55 secondes**. Sa requête a été accepté par le serveur (code **200** signifie l'acceptation) et la machine a reçu un fichier de taille **4392 octets**. La page de référence qui à partir de laquelle la machine lance la requête est **http://fr.search.yahoo.com/fr?p=peinture** en utilisant le navigateur **Mozilla 4.7** version **anglais** sous l'environnement **Windows 98**.

4. Code de réponse de serveur

Les codes de réponse de serveur web sont l'état du protocole http, qui permet à un serveur web de transmettre des informations et des pages à un client ou un navigateur web.

Ces codes ont été divisés en 5 familles, le tableau 1 présente les différents codes de chaque famille ainsi que ces significations.

Code	Signification
1XX	**Information**
100	Continue
101	Changement de protocole
2XX	**Succès**
200	OK
201	Créé
202	Accepté
203	Information ne faisant pas autorit
204	Pas de contenu
205	Rétablir le contenu
206	Contenu partiel
3XX	**Redirection**
301	Déplacement définitif
302	Déplacement temporaire
303	Voir autre
304	Non modifié
304	Utiliser un proxy
307	Redirection temporaire
4XX	**Erreur de client**
400	Mauvaise demande
401	Non autorisé
403	Interdit

404	Non trouvé
405	Méthode non admise
406	Pas acceptable
408	Délai écoulé pour la requête
410	Page indisponible définitivement
5XX	**Erreur du serveur**
500	Erreur interne du serveur
501	Non mise en œuvre
502	Mauvaise passerelle
503	Service indisponible
504	Expiration de la passerelle
505	Code d'extension de version http non prise en charge.

Tableau 1:Liste des codes de réponse de serveur [2]

5. Dictionnaire d'attaques

Le grand nombre d'attaque et le changement perpétuel de leurs formes nous a obligé à créer une BD sous forme d'un dictionnaire englobant toutes les attaques ainsi que leurs structures les plus connues. La comparaison entre un log et une expression rationnelle se fait en se référant à ce dictionnaire.

Les composants de dictionnaire d'attaques sont : l'identifiant de l'attaque, la catégorie d'attaque, l'impact et la description de chaque ligne d'attaque.

III. Présentation de projet

1. Problématique

L'évolution de l'utilisation du réseau Internet sur le territoire tunisien a été suivie en parallèle par l'augmentation des attaques cybernétiques telles que defacement web, phishing, attaque virale (malware), etc. Pour cela l'ANSI doit utiliser tous les moyens et les techniques afin de renforcer la sécurité du cyber-espace tunisien et protéger les systèmes d'informations du gouvernement et des entreprises tunisiennes.

En cas d'incidents, l'ANSI prend en charge leur traitement afin de déterminer les responsables et les scénarios d'attaque possible, en utilisant des traces (fichiers log) nécessaires.

L'investigation sur un incident consiste à analyser des fichiers de journalisation log de façon manuelle en utilisant une ligne de commande Unix. Cette méthode a plusieurs inconvénients tels que :

- Prendre beaucoup de temps,
- Inefficace
- Ne détermine pas tous les responsables et les scénarios d'attaques.

2. Etude de l'existant

Il existe sur le marché plusieurs logiciels pour l'analyse des fichiers log payants ou open source. Parmi ces logiciels nous retrouvons :

- **AWStats :**

AWStats est un analyseur de log web mais, FTP, Streaming et mail offrant des vues graphiques statiques mais aussi dynamiques des statistiques d'accès aux serveurs web. Il permet d'afficher le nombre de visites, de visiteurs uniques, de pages, de hits, de transfert, par domaine, hôte, heure, navigateur, OS, etc. AWStats est un logiciel libre. [3]

- **Webalizer**

Webalizer, mot-valise formé à partir de Web et d'analyser, est un logiciel permettant d'analyser l'utilisation des serveurs web en générant, à partir de leurs journaux d'accès (log), des comptes rendus sous forme de pages web. Diffusé sous licence GPL, c'est aujourd'hui un des outils d'administration de serveur web les plus utilisés, en particulier sur les architectures LAMP.

Les statistiques communément reportées par Webalizer incluent : le nombre de hits et de visites, le pays d'origine des visiteurs, les champs référents, la quantité de données téléchargées. Ces mesures peuvent être représentées graphiquement, et selon différentes échelles de temps telles que par mois, par jour, ou par heure. [4]

- **Advanced Log Analyzer**

Advanced Log Analyzer est un puissant logiciel d'analyse de trafic de site Web. Il génère un grand nombre de compte-rendu traditionnels comme les sites référant au visiteur, le nombre de téléchargements par jour, le nombre de clics et d'hôtes par jour, les moteurs de recherche les plus populaires, les mots clés de recherche, etc. Il peut être utilisé en tant que compteur de visiteurs normal et outil de suivi pour surveiller l'activité d'un site Web. L'avantage principal de cet outil d'analyse réside dans ses comptes rendus. Il peut recréer le chemin du visiteur à partir des fichiers log, créer un modèle Web du site et produire des comptes rendus sur le flux d'utilisateurs sur le site. [5]

- **Sawmill**

Sawmill est un analyseur de log universel, capable d'analyser et rendre compte de tout type de données du journal textuel. Sawmill supporte 850 formats de journaux communes différentes, à partir de la gamme complète de dispositifs populaires: serveurs web, serveurs de médias, serveurs de messagerie, les pare-feu, passerelles, etc. [6]

- **XpoLog**

XpoLog fait une plate-forme d'analyse du journal pour les applications, les serveurs et les applications de cloud computing. Centre XpoLog fournit la gestion du journal, journal de l'observation, l'analyse des journaux, rapports, analyse des problèmes, la corrélation et de nombreuses autres fonctionnalités qui aident les groupes d'applications, les opérations et les administrateurs pour accélérer l'investigation et le monitoring d'applications. [7]

3. Critique de l'existant

Une analyse des solutions existantes montre que la plupart de ces logiciels fournissent des informations générales et des fonctionnalités de base d'analyse de fichier journaux et ne déterminent pas les responsables et essentiellement le scénario possible de l'incident.

4. Solution proposée

Après une étude comparative des différentes solutions existantes, il est donc primordial au regard des inconvénients recensés de proposer une solution qui pourra répondre à nos besoins.

Pour cela nous avons choisi de développer une application d'analyse des fichiers logs dont l'objectif est : Automatiser l'analyse de l'investigation,

- Identifier des cybers criminels,
- Déterminer le scénario possible de l'incident,
- Modéliser des données des attaques selon plusieurs axes.

La figure 3 présente les trois phases du traitement des fichiers logs par la solution proposée:

- Dépôt et traitement : les étapes à suivre pour cette phase sont :
  - ✓ Définition de traitement : collecte les informations pour traiter l'incident.
  - ✓ Collection de preuves qui sont les fichiers logs.
  - ✓ Analyse de fichier log en utilisant l'algorithme suivant :
    - Lancer un script d'optimiser les fichiers logs,
    - Comparer une partie de logs avec le dictionnaire d'attaque,
    - Classer par modules les scénarios des attaques.
  - ✓ Synthèse et rédaction du rapport.
- Validation d'une liste du rapport logs traitées.
- Publication : cette phase comprend un rapport valide avec un scénario probable.

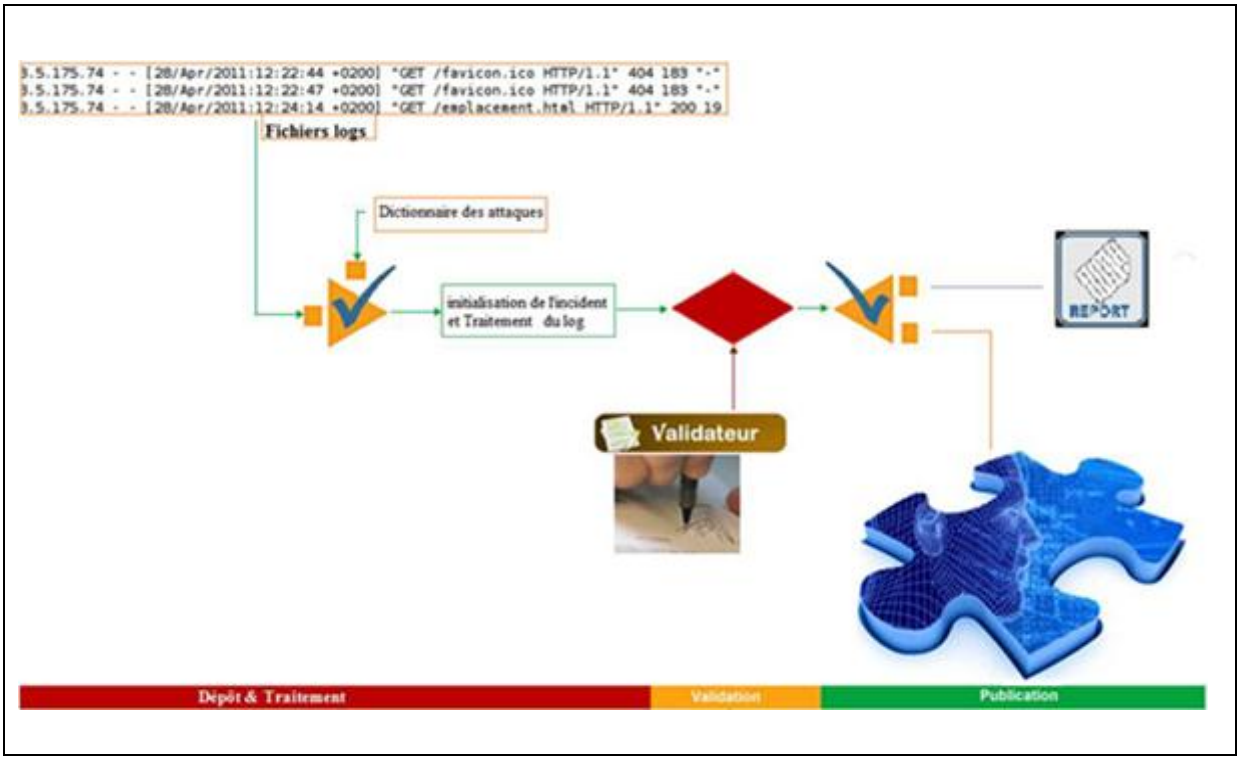

Figure 3: Les phases du traitement des fichiers logs

IV. Méthodologie à adopter

Pour assurer un bon rendement de développement en termes de qualité et de productivité le choix de la méthodologie en informatique est primordial. Vue la complexité des systèmes de nos jours, le génie logiciel doit tenter de remédier à cette complexité en offrant une démarche à suivre avec des étapes bien précises. C'est le principe des méthodologies de travail.

Le tableau 2 contient un comparatif entre les principales méthodologies de développement que nous avons choisi vu la diversité de ces méthodes.

Méthodologies	**Description**	**Points forts**	**Points faibles**
Cascade	- Les phases sont déroulées d'une manière séquentielle.	- Distingue clairement les phases du projet.	- Non itératif, - Pas de modèles pour les documents.
RUP (Rational Unified Process)	- Promu par rational, - Le RUP est à la fois une méthodologie et un outil prêt à l'emploi. - Cible des projets de plus de 10 personnes.	- Itératif, - Spécifie le dialogue entre les différents intervenants du projet, - Propose des modèles de documents pour des projets types.	- Assez flou dans sa mise en œuvre, - Ne couvre pas les phases en amont et en aval au développement.
2TUP (Two Truck Unified Process)	- S'articule autour de l'architecture, - Propose un cycle de développement en Y, - Cible des projets de toutes tailles.	- Itératif, - Laisse une large place à la technologie et à la gestion des risques, - Définit les profils des intervenants, les plannings, les prototypes.	- Plutôt superficiel sur les phases situées en amont et en aval du développement, - Ne propose pas de documents types.
XP (eXtreme Programming)	- Ensemble des bonnes pratiques	- Itératif, - Donne une	- Assez flou dans sa mise en œuvre,

	de développement, - Cible : moins de 10 personnes.	importance aux aspects techniques, - Innovant : programmation en duo.	- Ne couvre pas les phases en amont et en aval au développement.

Tableau 2: Comparaison entre les principales méthodologies de développement [8]

V. Choix de méthodologie à adopter

Suite à la comparaison entre les principales méthodologies de développement et afin de de mener à bon terme notre projet, nous avons opté pour le processus 2TUP pour plusieurs raisons :

- 2TUP donne une grande importance à la technologie ce qui est important pour notre projet,
- 2TUP est un processus en Y qui contient une branche technique, une branche fonctionnelle et une branche réalisation.

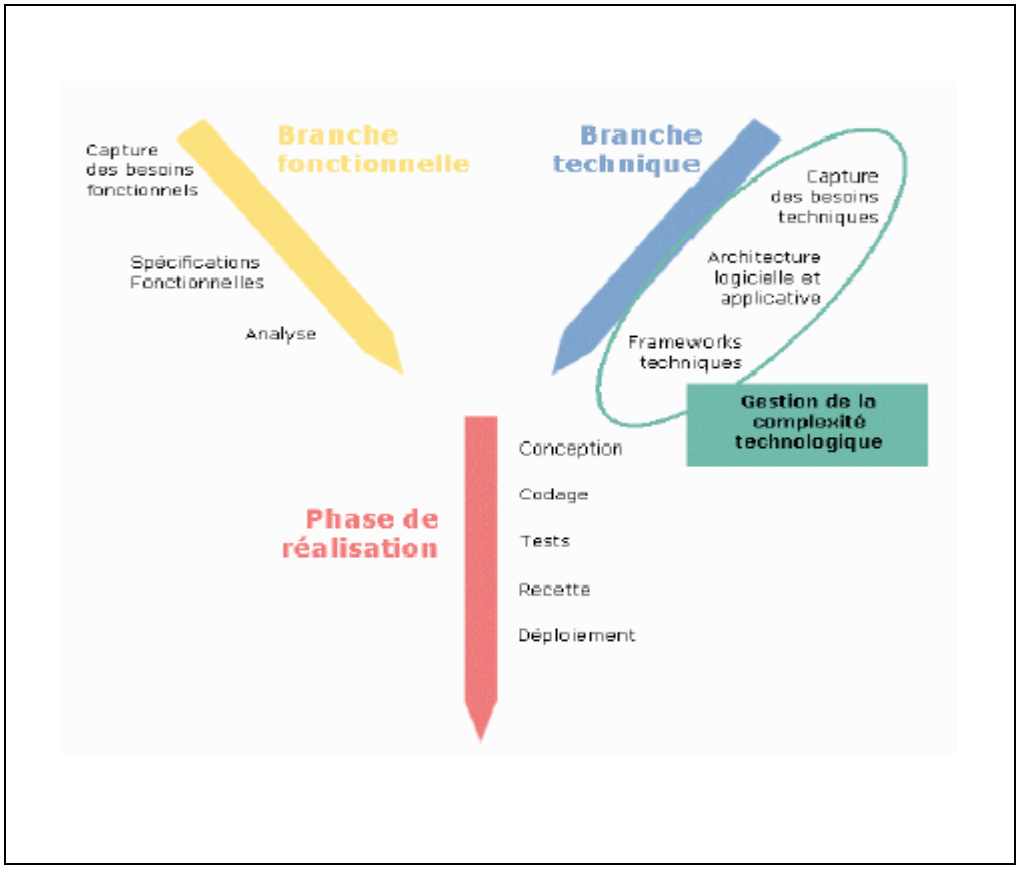

Figure 4: Le processus 2TUP

La figure 4 détaille les étapes développement des trois branches du processus 2TUP.

VI. Choix de langage de modélisation

La modélisation permet de mieux comprendre le fonctionnement du système, c'est également un bon moyen pour maitriser sa complexité et d'assurer sa cohérence.
Pour cela nous avons choisi le langage de modélisation UML (Unified Modeling Language) qui permet de :

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, spécifications, etc...
- Documenter un projet.
- Générer automatiquement la partie logicielle d'un système. [9]

Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil ANSI. Puis, nous avons procédé à une étude des solutions existantes qui nous a conduite par la suite à proposer une solution dont le but est de combler les limites de ces dernières.

Finalement, nous avons analysé la méthodologie de développement ainsi que le langage de modélisation à mettre en œuvre.

Le chapitre suivant est consacré à l'analyse et la spécification des besoins du projet.

Analyse et spécification des besoins

Introduction

Dans ce chapitre, nous détaillons les besoins fonctionnels, non fonctionnels et architecturaux afin de bien démarrer notre projet. Enfin, nous ajouterons la modélisation des besoins de l'application ainsi que les diagrammes de cas d'utilisation.

I. Spécification des besoins

1. Les besoins fonctionnels

Les besoins fonctionnels doivent répondre aux exigences de notre future application en termes de fonctionnalités, et comme notre projet consiste à développer une application d'investigation, cette application doit couvrir les besoins fonctionnels suivants:

- La gestion d'accès à la plateforme d'investigation via un login et mot de passe,
- L'inscription dans la plateforme,
- La déclaration d'un incident,
- La gestion d'investigateur et de client,
- La gestion de profil utilisateur,
- La gestion d'organisme, d'incident, de dictionnaire d'attaque et de rapport
- Analyser un incident.
- Validation de rapport analysé,
- Consulter statistique,
- Télécharger rapport,
- Upload fichier log.

2. Les besoins non fonctionnels

A part les besoins fondamentaux, notre future application doit répondre aux critères suivants :

- La sécurité au niveau de session de l'utilisateur : Les informations concernant l'identité de la personne connectée sont cryptées.
- La compatibilité de la plateforme avec n'importe quel navigateur web ou système d'exploitation.
- La disponibilité de la plateforme 24h/24 et 7 jours / 7.
- L'intégrité des données : Il y a certains traitements pour les mauvaises entrées des données, tel que les alertes immédiates pour l'utilisateur en cas d'erreur.

- La convivialité : la future application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

3. Les besoins architecturaux

Nous devons savoir qu'il existe plusieurs types d'architectures. Parmi ces architectures, nous pouvons citer :

a. Architecture centralisée

Cette architecture se compose d'un unique serveur, dont le but est de recenser les fichiers proposés par les différents clients. Il dispose principalement de deux types d'informations : celles sur le fichier (nom, taille, ...), et celles sur l'utilisateur (nom utilisé, IP, nombre de fichiers, type de connexion, ...) Du côté du client, rien de plus simple : une fois connecté grâce au logiciel spécifique, il peut faire des recherches comme avec un moteur classique. On obtient alors une liste d'utilisateurs disposant de la ressource désirée. Il suffit alors de cliquer sur le bon lien pour commencer le téléchargement.
Les avantages de cette architecture sont :

- Simplicité d'utilisation.
- Facilité de recherche.

Les inconvénients sont les suivantes :

- Faible sécurité : si le serveur est supprimé, le réseau est hors service
- Pas d'anonymat : chaque utilisateur est identifié sur le serveur [10]

b. Architecture client/serveur

L'architecture client/serveur est subdivisée entre deux entités client et serveur qui coopèrent ensemble via des requêtes. Nous distinguons plusieurs types de cette architecture :

- Architecture 2-tiers :

Une architecture 2-tiers est composée de deux éléments, un client et un serveur qui se contente de répondre aux requêtes du client.
Ce type d'application permet d'utiliser toute la puissance des ordinateurs présents sur le réseau et permet de fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données qui restent gérées de façon centralisée.

La gestion des données est prise en charge par un SGBD centralisé sur un serveur dédié. On interroge ce serveur à travers un langage de requête, le plus courant étant SQL.

La figure 5 illustre le dialogue entre le client et le serveur se résume donc à l'envoi de requêtes et aux données en réponse. [11]

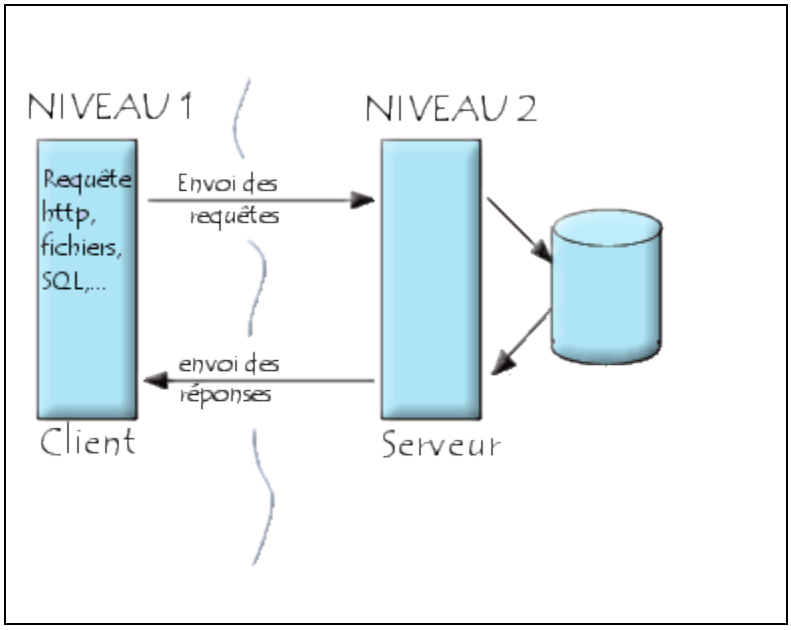

Figure 5: Architecture 2-tiers

- Architecture 3-tiers :

Cette architecture 3-tiers sépare l'application en 3 couches des services distincts :

- ✓ Couche présentation : l'affichage et les traitements locaux qui sont pris en charge par le poste client,
- ✓ Couche application : les traitements applicatifs globaux sont pris en charge par le service applicatif,
- ✓ Couche métier : les services de base de données sont pris en charge par un SGBD.

La figure 6 montre les 3 couches de cette architecture.

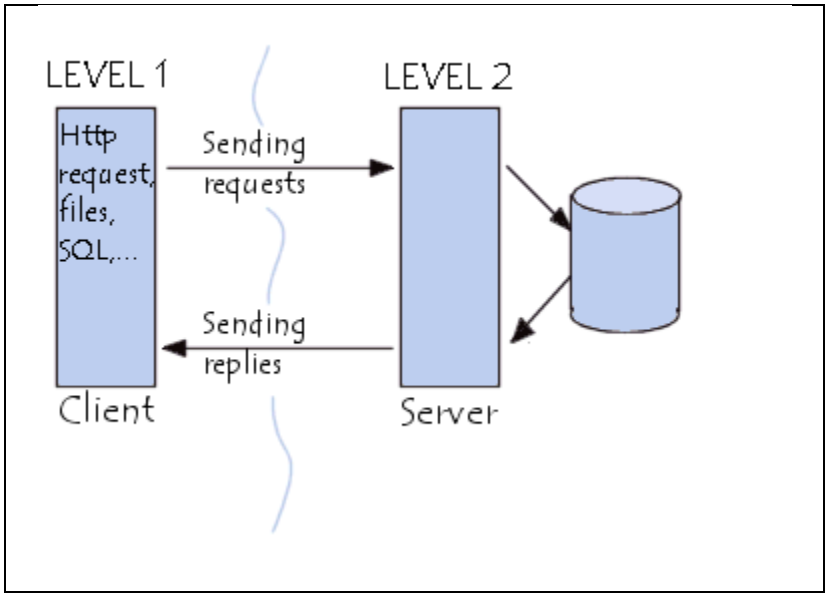

Figure 6: Architecture 3-tiers

- Architecture n-tiers :

L'architecture n-tiers est aussi appelée architecture distribuée ou architecture multi-tiers. L'architecture n-tiers qualifie la distribution d'applications entre de multiples services et non la multiplication des niveaux de service : les trois niveaux d'abstraction d'une application sont toujours pris en compte.

- Comparaison des architectures 2 tiers, 3 tiers et n tiers

Le tableau 3 présente les différences qu'apportent les architectures 3 et n tiers aux anciennes architecture 2-tiers. [12]

	2 tiers	3 et n tiers
Administration du système	Complexe (la couche application est physiquement répartie sur plusieurs postes clients)	Moins complexe (les applications peuvent être gérées centralement sur le serveur)
Sécurité	Faible (sécurité au niveau des données)	Elevée (raffinée au niveau des services ou des méthodes)
Encapsulation des données	Faible (les tables de données sont directement accessibles)	Elevée (le client fait appel à des services ou méthodes)
Performance	Faible (plusieurs requêtes SQL sont transmises sur les réseaux, les données sélectionnées doivent être acheminées vers le client pour analyse)	Bonne (seulement les appels de services et les réponses sont mis sur le réseau)
Lien Serveur-serveur	Non	Oui (via le middleware Serveur/Serveur)
Flexibilité d'architecture matérielle	Limitée	Excellente (possibilité de faire résider les couches 2 et 3 sur une ou plusieurs machines)
Relève en cas de pannes	Faible	Excellente (possibilité d'avoir la couche du centre "middle-tier" sur plusieurs serveurs)

Tableau 3 : Comparaison entre les différentes architectures

II. Modélisation des besoins

1. Les acteurs

L'administrateur, l'investigateur et le client sont les acteurs interagissent avec notre système.

- L'administrateur peut :

✓ S'authentifier,

✓ Gérer profil,

✓ Gérer client, investigateur,

- ✓ Gérer incident, dictionnaire d'attaque et d'organisme,
- ✓ Valider rapport,
- ✓ Consulter statistique,
- ✓ Télécharger rapport.

- L'investigateur peut :

- ✓ S'authentifier,
- ✓ Gérer profil,
- ✓ Analyser incident,
- ✓ Gérer rapport,
- ✓ Télécharger rapport.

- Le client peut :

- ✓ S'authentifier,
- ✓ Gérer profil,
- ✓ Déclaration d'un incident,
- ✓ Upload fichier log,
- ✓ Inscription à la plateforme,
- ✓ Télécharger rapport.

2. Diagramme de cas d'utilisation

La figure 7 représente le diagramme de cas d'utilisation général de notre système d'investigation de log. Nous y retrouvons comme convenu les acteurs principaux et leurs rôles.

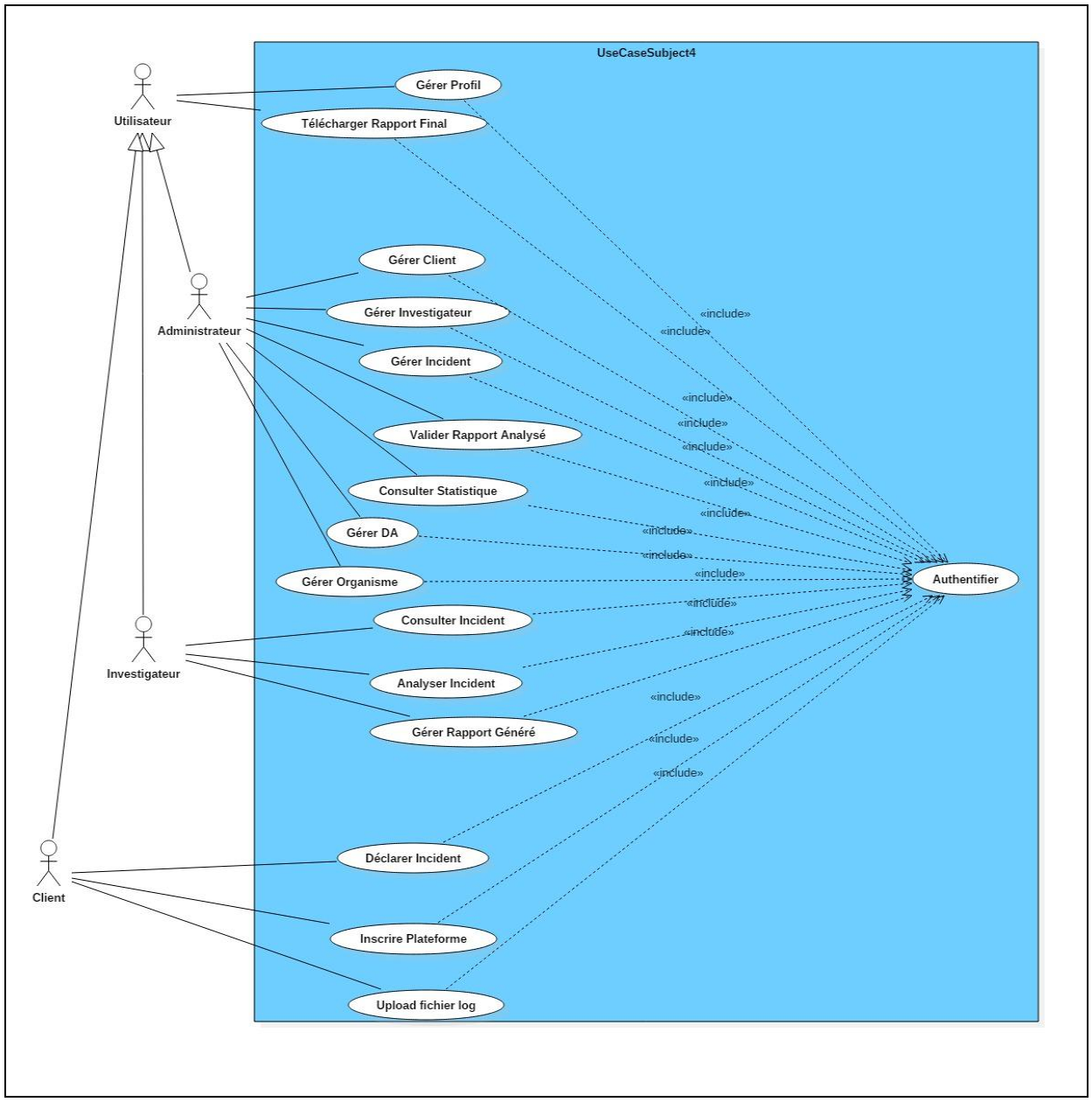

Figure 7: Diagramme de cas d'utilisation général

a. Cas d'utilisation «gérer investigateur » pour administrateur

La figure 8 présente de façon plus détaillé les différentes fonctions pour gérer un investigateur, alors il est possible de créer, modifier, supprimer ou consulter un investigateur.

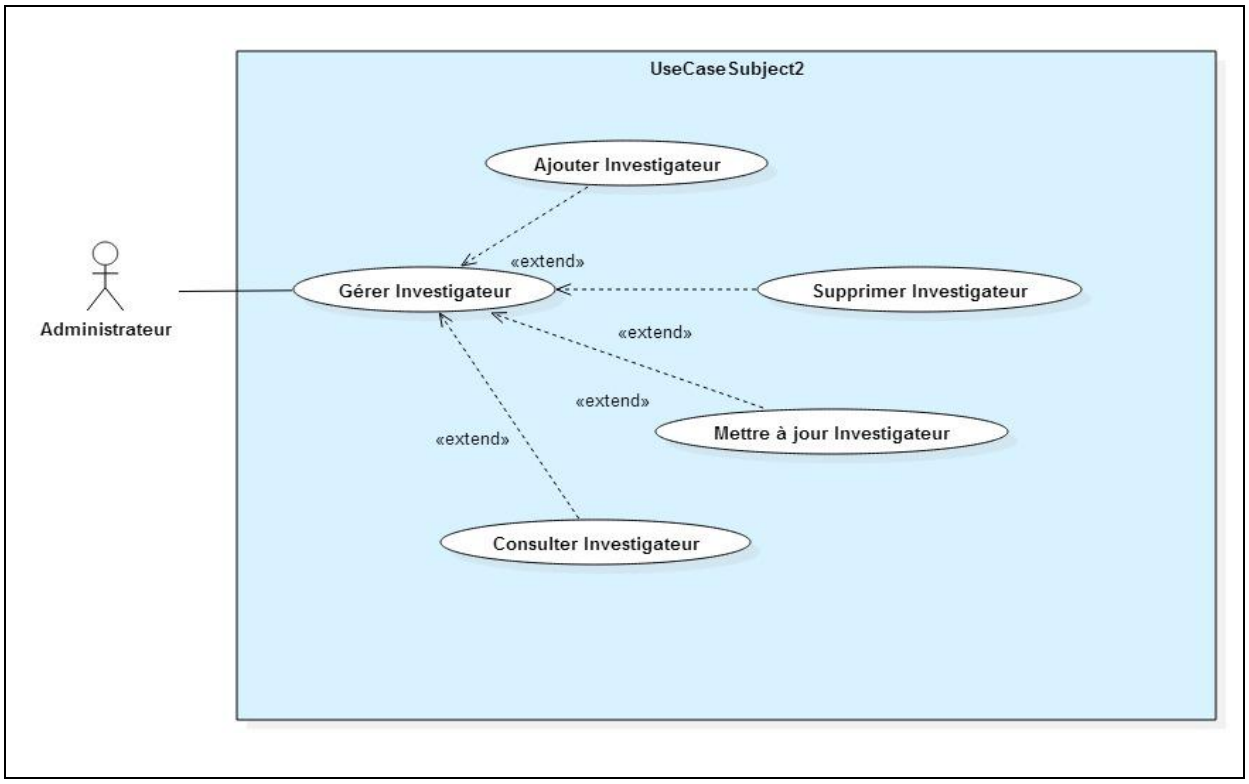

Figure 8: Diagramme de cas d'utilisation "gérer investigateur"

b. Cas d'utilisation «gérer incident » pour administrateur

Comme montre la figure 9, pour gérer un incident, l'administrateur peut le consulter, le supprimer ou affecter un investigateur pour l'analyser.

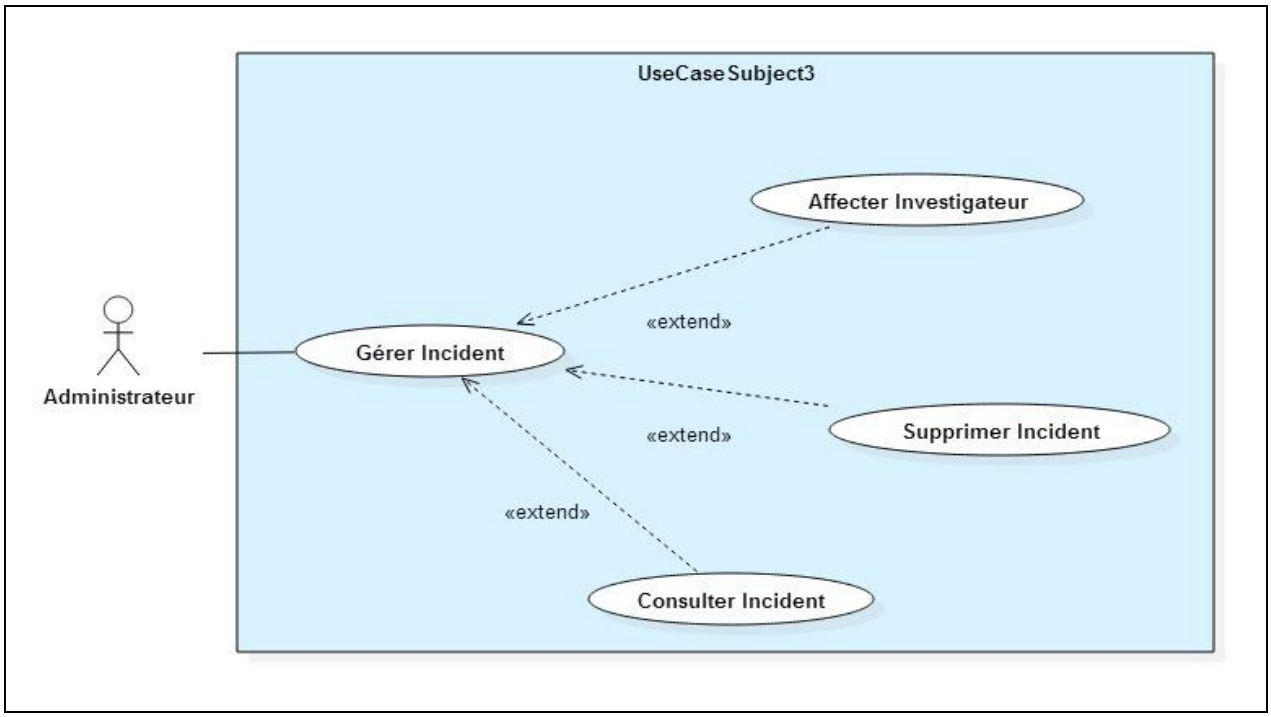

Figure 9: Diagramme de cas d'utilisation "gérer incident"

c. Cas d'utilisation «gérer profil » pour utilisateur

Chaque utilisateur quel que soit administrateur, investigateur ou client peut gérer son profil, alors il est possible d'après la figure 10 de modifier ou de consulter le profil.

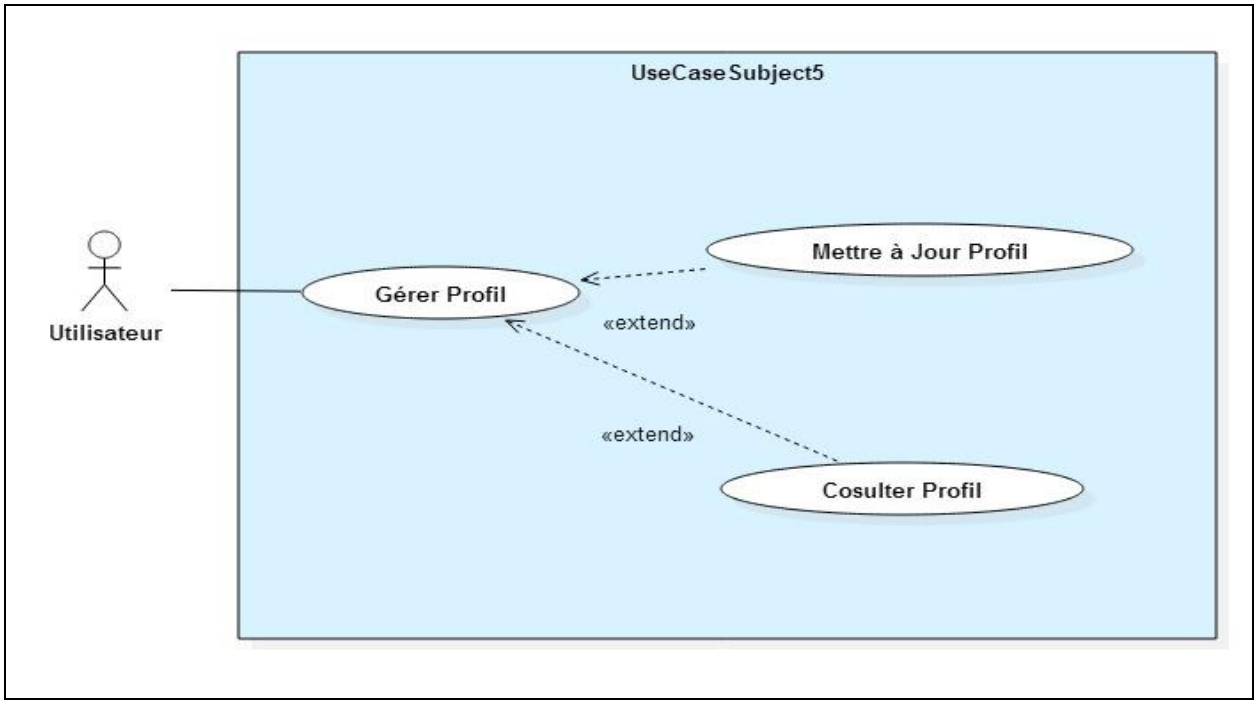

Figure 10: Diagramme de cas d'utilisation « gérer profil »

d. Cas d'utilisation «analyser incident » pour investigateur

La figure 11 montre que l'analyse d'un incident exige de générer un rapport et de gérer le dictionnaire d'attaque.

Pour gérer un dictionnaire d'attaque, il faut ajouter une attaque, un impact et une description

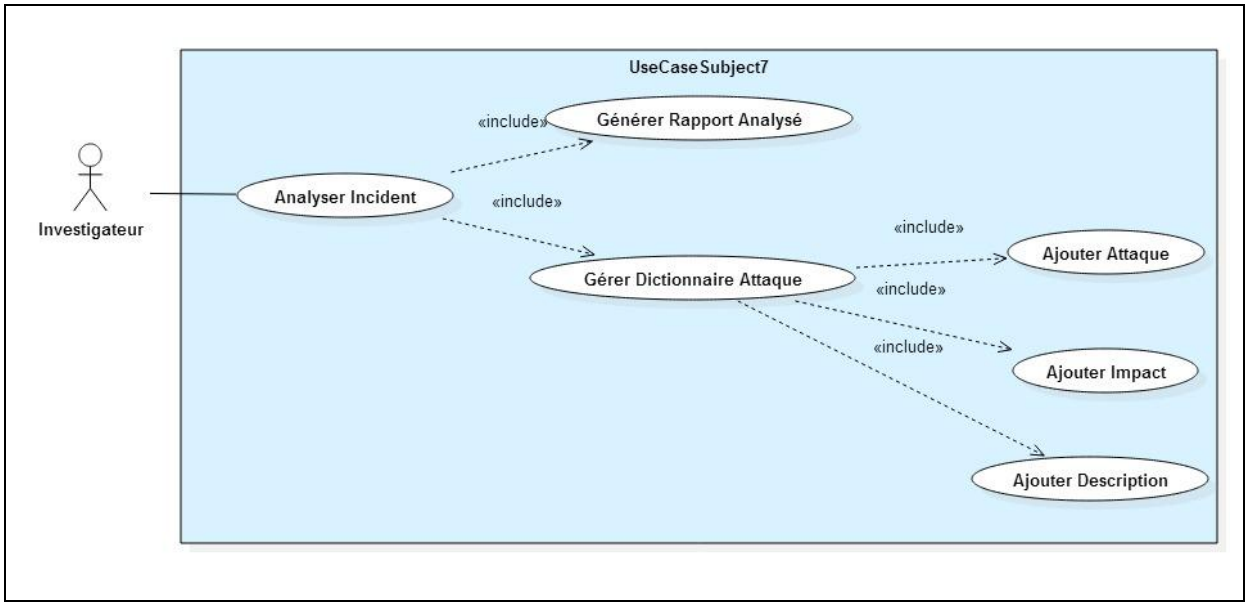

Figure 11: Diagramme de cas d'utilisation « analyser incident »

e. Cas d'utilisation «gérer rapport » pour investigateur

La figure 12 présente les fonctions nécessaires pour gérer un rapport, l'investigateur télécharge le rapport pour le vérifier et modifier son statut. Si le rapport n'est pas validé, l'investigateur le supprime.

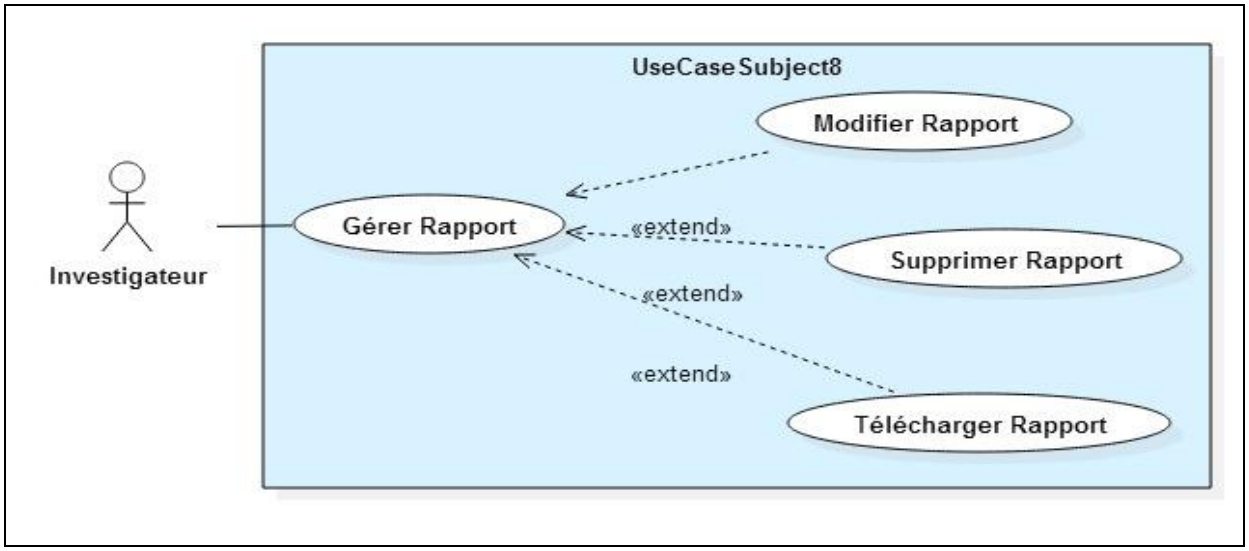

Figure 12: Diagramme de cas d'utilisation « gérer rapport »

1. Description des cas d'utilisation

Dans cette partie, nous donnons plus de détails sur les différents cas d'utilisation présentés ci-dessus. Les cas d'utilisation présentés qui suivent : « s'authentifier », « inscrire à la plateforme», « déclarer incident », « affecter investigateur à un incident », « analyser incident», « afficher les statistiques des attaques».

a. Description du cas d'utilisation : « S'authentifier »

Pour assurer la sécurité de système d'informations, il est important que chaque acteur passe par une phase d'authentification. C'est le cas de notre système où l'administrateur, l'investigateur et le client se sont authentifiés avant de pouvoir bénéficier des services offerts par notre application. Le tableau 4 présente une description de cas d'utilisation « s'authentifier ».

Etapes	Description
Résumé	• **Acteurs :** utilisateur (administrateur, investigateur et client) • **Titre :** authentifier administrateur, investigateur et client. • **Description :** le système identifie à ce niveau l'administrateur, l'investigateur et le client qui veut utiliser la plateforme.
Pré-conditions	• L'administrateur, l'investigateur et le client s'est connecté au système. • Le système est opérationnel.
Scénario nominal	• L'administrateur, l'investigateur et le client entre ses paramètres de connexion login et mot de passe. • Le système vérifie les informations saisies. • Le système affiche l'espace de travail de chaque utilisateur.
Scénario alternatif	• Les données sont erronées. • Le système affiche un message d'erreur que l'utilisateur n'existe pas.
Post-conditions	• L'utilisateur est sur son espace de travail. • Le système attend désormais qu'il exécute une nouvelle action.

Tableau 4 : Description du cas d'utilisation « authentifier utilisateur »

b. Description du cas d'utilisation : « Afficher les statistiques des attaques »

Dans le tableau 5, nous décrivons le scénario nominal et le scénario alternatif du cas d'utilisation « consulter les statistiques des attaques ».

Etapes	Description
Résumé	• **Acteurs :** administrateur. • **Titre :** consulter les statistiques des attaques. • **Description :** le système affiche à l'administrateur le pourcentage de chaque attaque.
Pré-conditions	• L'administrateur s'est authentifié. • Le système est opérationnel. • Les rapports ont généré. • Les attaques ont été ajoutées. • L'administrateur accède à son espace de travail.
Scénario nominal	• L'administrateur clique sur le lien « statistique » • Le système renvoi sur sa page le graphe indiquant le pourcentage et le nom de chaque catégorie d'attaque.
Scénario alternatif	• Le système n'a pas pu afficher les statistiques car aucun rapport n'a été généré.
Post-conditions	• Les statistiques des catégories des attaques ont affiché. • Le système attend désormais qu'il exécute une nouvelle action.

Tableau 5: description du cas d'utilisation « consulter les statistiques».

c. Description du cas d'utilisation : « Inscrire à la plateforme »

L'inscription en ligne à notre plateforme permet au client de bénéficier de ses services. Le tableau 6 montre la description de cas d'utilisation « inscrire à la plateforme ».

Etapes	Description
Résumé	• **Acteurs :** client. • **Titre :** inscrire à la plateforme d'investigation des logs. • **Description :** le système affiche au client le formulaire d'inscription.
Pré-conditions	• Le système est opérationnel. • Le client veut bénéficier des services offerts par l'application.
Scénario nominal	• Le client clique sur le lien « inscription » • Le système affiche au client le formulaire d'inscription. • Le client saisit ses informations personnelles, son login et son mot de passe. • Le système affiche un message de confirmation concernant l'inscription.
Scénario alternatif	• Le login de client existe déjà dans le système. • Le système affiche un message d'erreur concernant le type des informations saisies.
Post-conditions	• Le système renvoi sur la page d'authentification pour accéder à l'espace de travail. • Le système attend désormais qu'il exécute une nouvelle action.

Tableau 6: Description du cas d'utilisation « inscrire à la plateforme »

d. Description du cas d'utilisation : «Affecter investigateur à un incident »

Le tableau 7 décrit à son tour les détails du cas d'utilisation «Affecter investigateur à un incident » qui s'accompagne du scénario nominal et du scénario alternatif.

Etapes	Description
Résumé	• **Acteurs :** administrateur. • **Titre :** affecter un investigateur à un incident. • **Description :** le système affiche à l'administrateur la liste des

	incidents afin de les affecter un investigateur.
Pré-conditions	• L'administrateur s'est authentifié. • Le système est opérationnel. • Le système a au préalable des incidents en cours d'affectation. • Le système a au préalable des investigateurs. • L'administrateur accède à son espace de travail.
Scénario nominal	• L'administrateur clique sur le lien «gestion incident » • L'administrateur choisit l'incident en cours d'affectation. • L'administrateur télécharge le fichier log pour le vérifier et le consulter. • L'administrateur clique sur le lien « affecter investigateur ». • L'administrateur affecte un investigateur parmi les investigateurs. • Le système renvoi sur la page « gestion incident » que l'incident a été affecté à un investigateur.
Scénario alternatif	• Le système n'a pas des incidents en cours d'affectation.
Post-conditions	• Le système renvoi sur l'espace de travail de l'investigateur qu'il a un incident à analyser. • Le système attend désormais qu'il exécute une nouvelle action.

Tableau 7: Description du cas d'utilisation : «Affecter investigateur à un incident »

e. Description du cas d'utilisation : « Analyser incident »

Le but principal de notre application est l'analyse d'un incident, le tableau 8 décrit le scénario nominal de cas d'utilisation « analyser incident » ainsi que le scénario alternatif.

Etapes	Description
Résumé	• **Acteurs :** investigateur. • **Titre :** analyser un incident. • **Description :** le système identifie les cybers criminels et le scénario possible de l'incident.

Pré-conditions	• L'administrateur s'est authentifié. • Le système est opérationnel. • Les incidents ont été affectés pour l'investigateur. • L'administrateur accède à son espace de travail. • L'administrateur télécharge et vérifie le fichier log.
Scénario nominal	• L'investigateur clique sur le lien « analyser incident ». • Le système affiche la liste des incidents à analyser. • L'investigateur choisit l'incident et clique sur le lien « analyser ». • Le système lance un script pour optimiser le fichier log. • Le système compare une partie de log avec le dictionnaire d'attaque. • Le système rédige le rapport de type xml et text. • Le système renvoi sur l'espace de travail de l'investigateur que l'incident a été analysé.
Scénario alternatif	• Le système n'a pas d'incident en cours d'analyser.
Post-conditions	• Le système modifie le statut de l'incident de en cours à analyser. • Le système affiche les rapports générés pour la vérification. • Le système attend désormais qu'il exécute une nouvelle action.

Tableau 8: Description du cas d'utilisation : « Analyser incident »

f. Description du cas d'utilisation : « Déclarer incident »

Après la phase d'inscription, un client peut déclarer des incidents en ligne, les étapes à suivre pour ce cas d'utilisation « déclarer incident » ont décrit dans le tableau 9.

Etapes	Description
Résumé	• **Acteurs :** client. • **Titre :** déclarer un incident en ligne. • **Description :** le système affiche au client le formulaire de déclaration.
Pré-conditions	• L'administrateur s'est authentifié.

	• Le système est opérationnel. • Le client accède à son espace de travail.
Scénario nominal	• Le client clique sur le lien « déclaration incident » • Le système affiche au client le formulaire de déclaration. • Le client saisit les informations générales sur l'incident et les informations sur le type d'incident. • Le client upload le fichier log. • Le système affiche un message de confirmation concernant la déclaration.
Scénario alternatif	• Le système affiche un message d'erreur concernant les informations saisies. • Le système affiche un message d'erreur concernant le type de fichier log.
Post-conditions	• Le système renvoi sur l'espace de travail de client. • Le système attend désormais qu'il exécute une nouvelle action.

Tableau 9: Description du cas d'utilisation : « Déclarer incident »

Conclusion

Ce chapitre nous a été utile pour monter notre but, nous avons spécifié les différents besoins auxquels doit répondre notre système, ainsi que, nous avons fait une étude des différents cas d'utilisation de notre solution.

Enfin, il nous reste à élaborer une bonne conception afin d'assurer le bon fonctionnement du système. C'est ainsi que nous pourrons entamer le prochain chapitre sur la conception.

Conception

Introduction

L'activité de conception détaillée consiste à façonner le système et à lui donner une forme et une architecture répondant à tous les besoins y compris les besoins non fonctionnels et architecturels. Elle consiste une base et un point de départ aux activités d'implémentation et de test.

I. Architecture de système

Au niveau de cette section, nous avons proposé l'architecture physique à adopter pour notre application. Notre choix s'est porté sur l'architecture client-serveur et plus précisément sur l'architecture 3-tiers pour les raisons suivantes :

- Elle correspond le mieux à la structure de notre système qui composera d'un serveur d'application, d'un serveur de base des données et des clients.
- Deuxièmement, vu que nous avons besoin d'un client léger (navigateur) qui n'aura qu'à se connecter au serveur. Donc, il nous a paru d'opter pour une architecture plus évoluée que l'architecture 2-tiers.
- Finalement, l'architecture 3-tiers est plus sécurisée et plus performante que l'architecture 2-tiers.

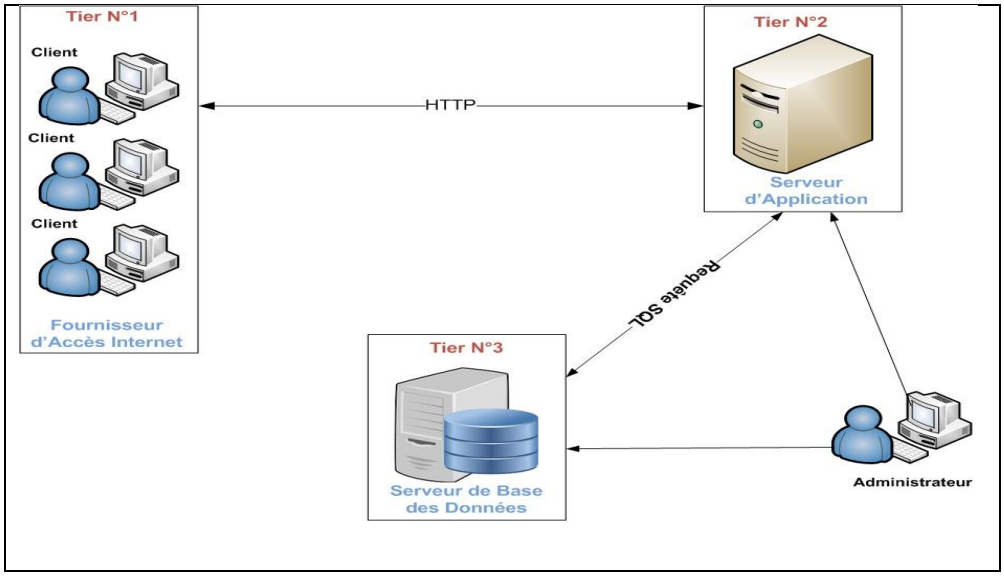

Figure 13: Architecture de système

La figure 13 présente les trois niveaux de notre architecture de système qui sont le client, le serveur d'application et le serveur de base de données.

II. Déploiement

La figure 14 présente le diagramme de déploiement qui modélise l'architecture physique de notre système ainsi qu'il affiche les relations entre les composants logiciels et matériels de notre système.

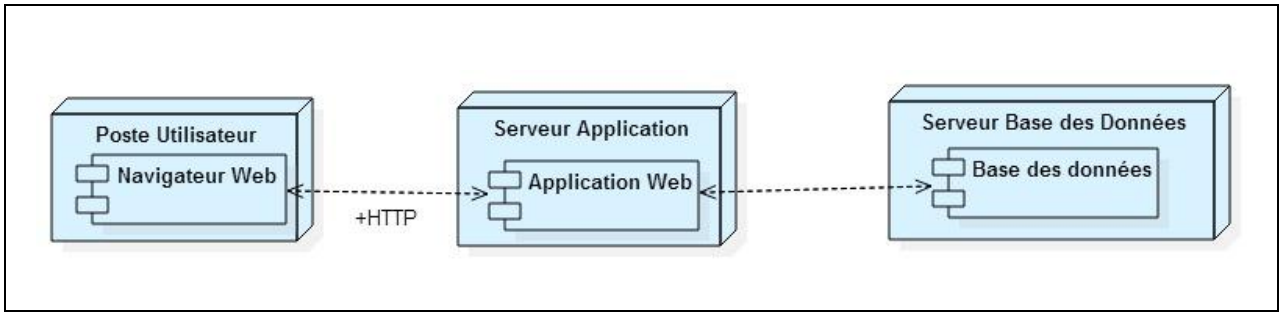

Figure 14 : Diagramme de déploiement

III. Conception

La conception est l'étape la plus importante dans le cycle de développement de l'application. Au niveau de cette partie, nous avons détaillé l'aspect statique présenté par le diagramme de classe ainsi que l'aspect dynamique décrit par le diagramme de séquence de notre système.

1. Diagramme de classe

Le diagramme de classe est une modélisation statique du notre système en termes de classes et de relations entre ces classes.

a. Représentation des classes

La figure 15 illustre les différentes classes intervenant dans notre application, ainsi que les attributs, les opérations et associations entre les classes.

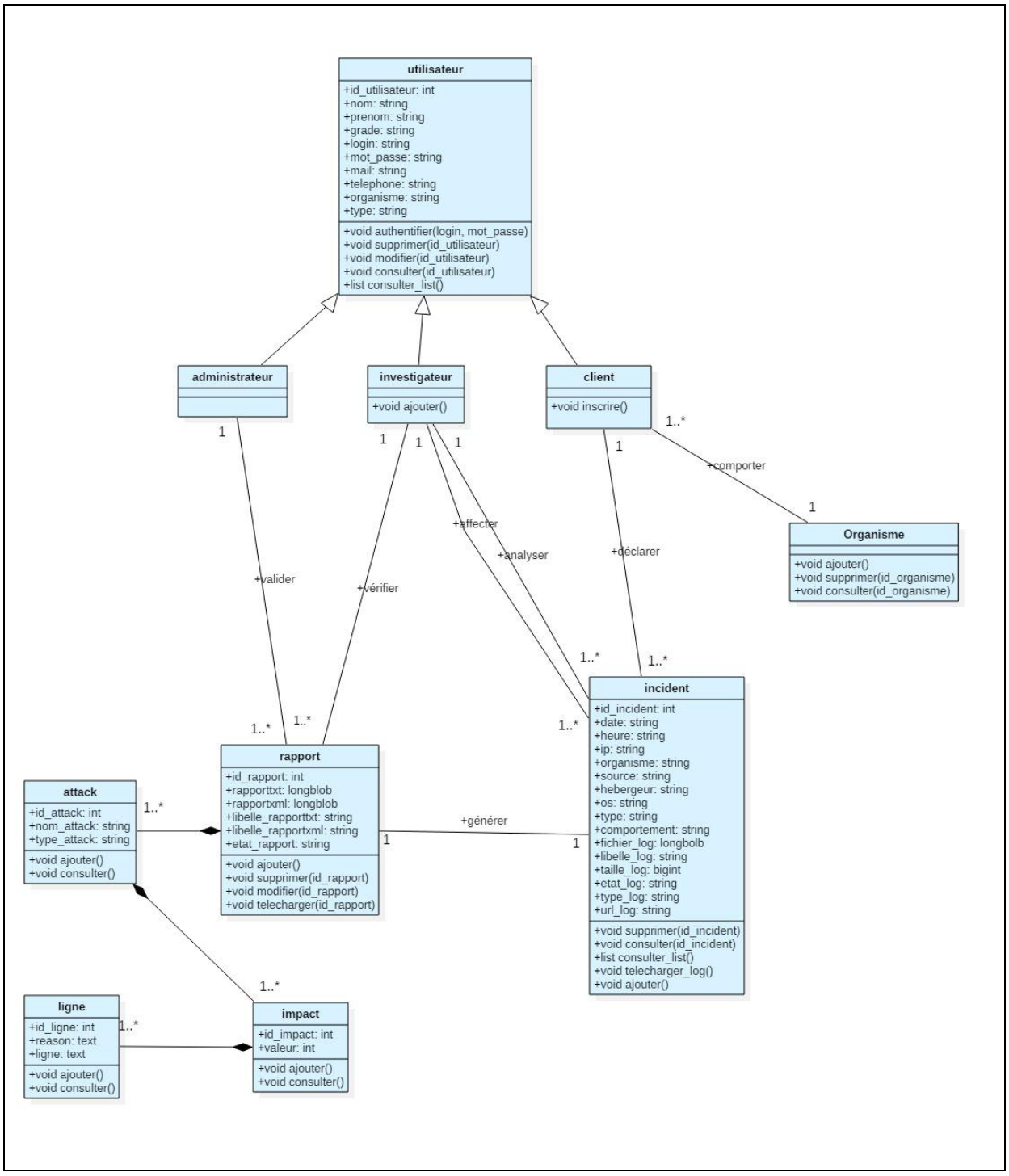

Figure 15: Diagramme de classe

b. Conception de la base de données

Notre base de données est composée de ces tables suivantes :

organisme (id_organisme, libelle_organisme)

utilisateur (id_utilisateur, #id_organisme, nom, prenom, grade, login, mot_passe, mail, organisme, type, telephone)

incident (id_incident, #id_investigateur, #id_client, date, heure, ip, organisme, source, hebergeur, os, type, comportement, fichier_log, libelle_log, taille_log, etat_log, type_log, url_log)
rapport (id_rapport, #id_incident, #id_investigateur, #id_client, rapporttxt, rapportxml, libelle_rapporttxt, libelle_rapportxml, etat_rapport)
attack (id_attack, #id_rapport, nom_attack, type_attack)
impact (id_impact, # id_attack, valeur)
ligne (id_ligne, #id_impact, reason, ligne)

2. Diagramme de séquence

Le diagramme de séquence est une modélisation dynamique de notre système. Il représente graphiquement des interactions entre les acteurs et le système selon un ordre chronologique.
Dans ce qui suit, nous présentons le diagramme de séquence pour chaque cas d'utilisation de notre application.

a. Diagramme de séquence de cas d'utilisation « s'authentifier »

Pour accéder à son espace de travail, chaque utilisateur quel que soit l'administrateur, l'investigateur ou le client, doit se connecter en utilisant son login et mot de passe.
Le diagramme de séquence présenté dans la figure 16 présente l'enchainement de la phase d'authentification.

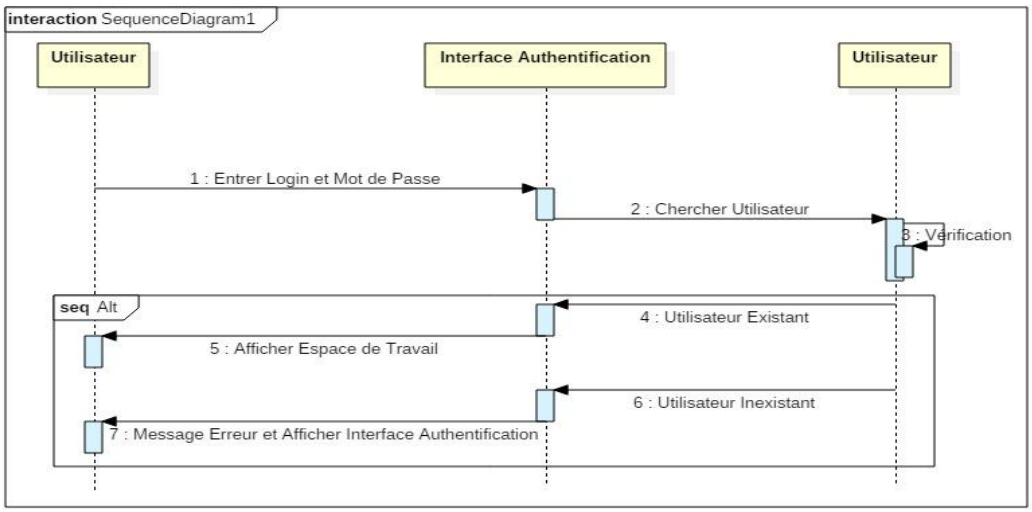

Figure 16 : Diagramme de séquence de cas d'utilisation « s'authentifier »

b. Diagramme de séquence de cas d'utilisation «inscrire à la plateforme »

L'inscription dans la plateforme d'investigation nécessite l'enregistrement des informations concernant chaque client. La figure 17 illustre le diagramme de séquence des différentes étapes à suivre pour s'inscrire.

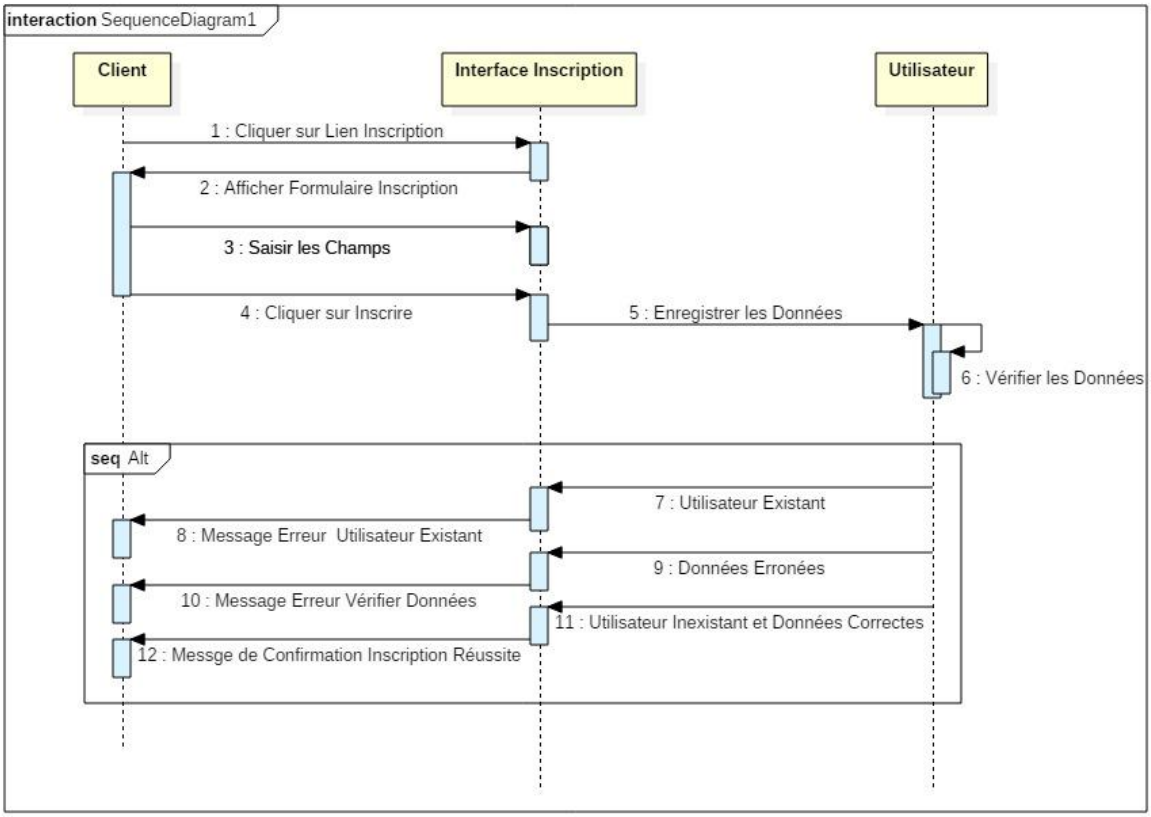

Figure 17: Diagramme de séquence de cas d'utilisation «inscrire à la plateforme »

c. Diagramme de séquence de cas d'utilisation «Déclarer incident »

Après l'authentification, le client peut déclarer un incident en mettant les informations sur l'incident ainsi que le fichier log pour faire le traitement nécessaire, ces étapes sont présentées dans le diagramme de séquence montré dans la figure 18.

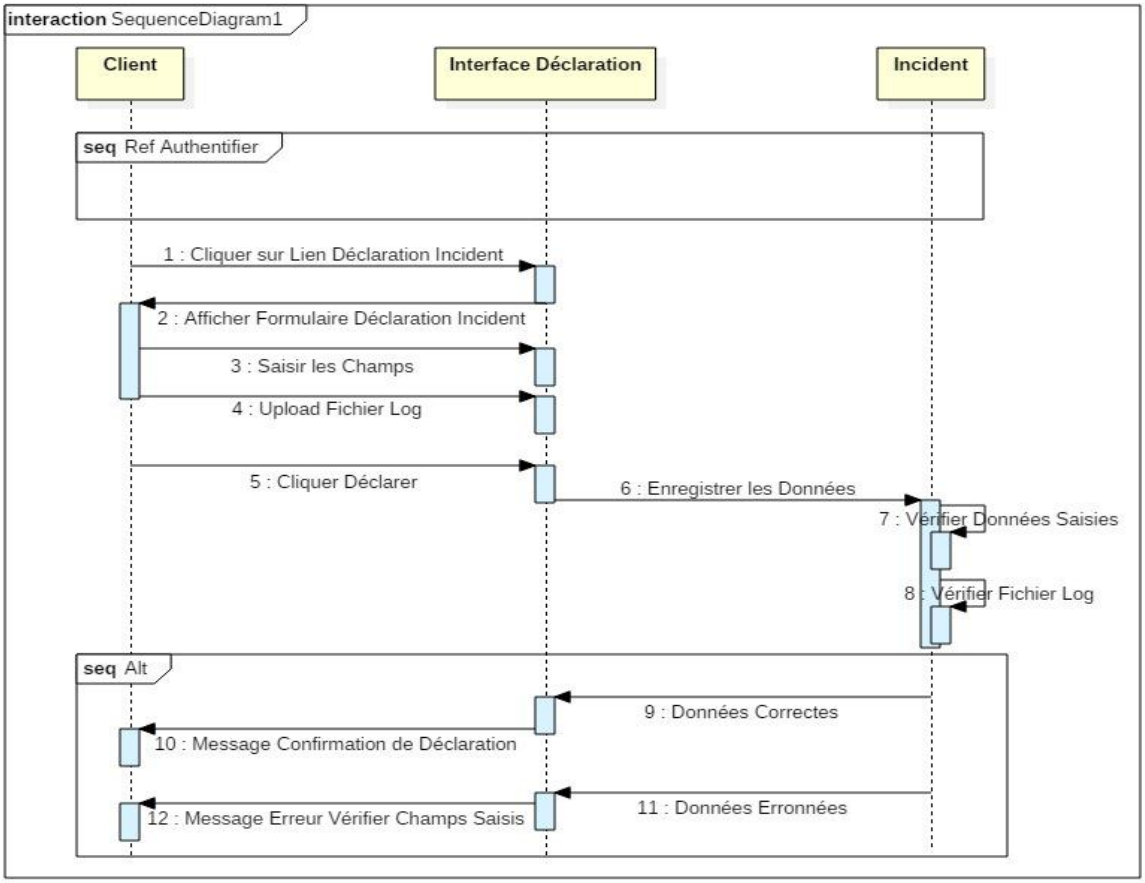

Figure 18: Diagramme de séquence de cas d'utilisation «Déclarer incident »

d. Diagramme de séquence de cas d'utilisation «affecter investigateur à un incident »

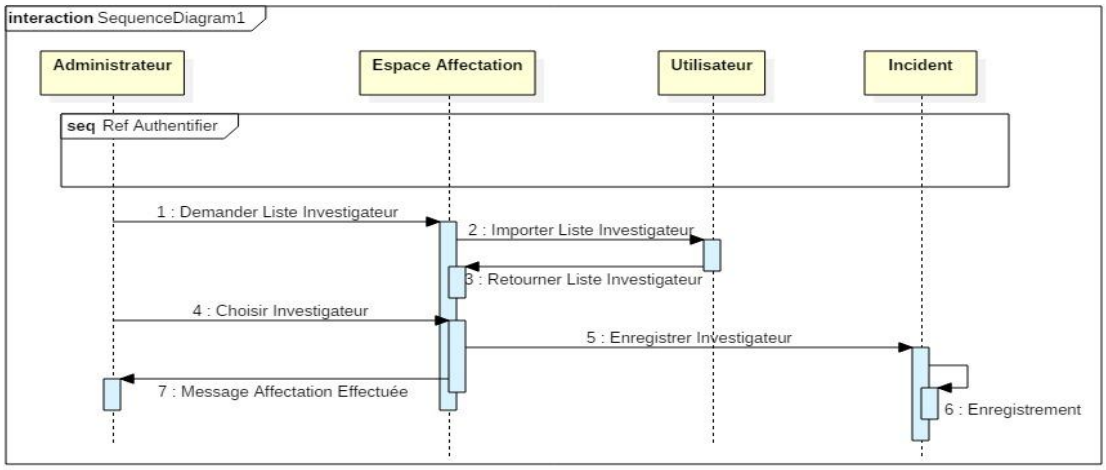

Figure 19: Diagramme de séquence de cas d'utilisation «affecter investigateur »

La figure 19 montre le diagramme de séquence de scénario « Affectation d'un investigateur pour l'analyse d'un incident »

e. Diagramme de séquence de cas d'utilisation «analyser incident »

Le diagramme de séquence de la figure 20, illustre les étapes effectuées par le système pour analyser un incident et générer un rapport décrivant le scénario d'attaque.

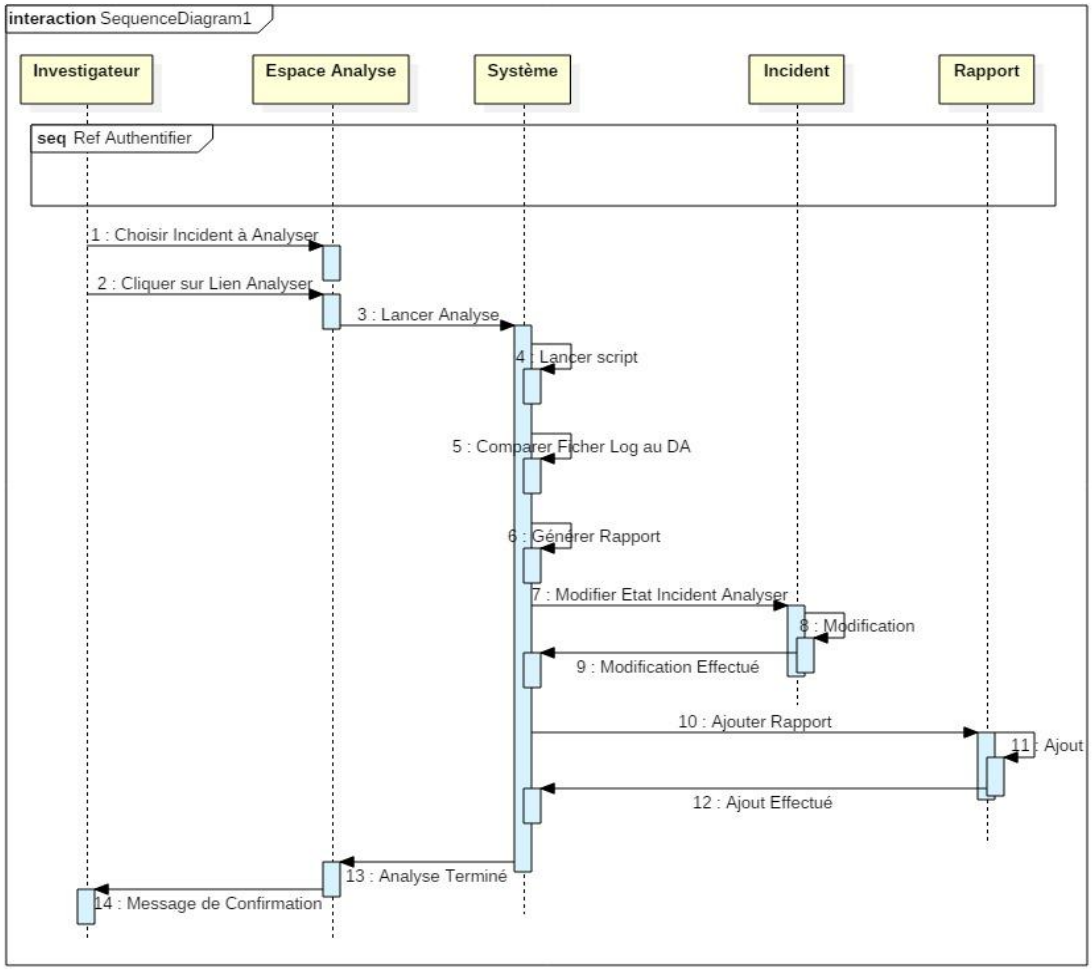

Figure 20: Diagramme de séquence de cas d'utilisation «analyser incident »

f. Diagramme de séquence de cas d'utilisation « afficher statistiques des attaques »

Selon la figure 21, le système invite l'administrateur à consulter les graphiques, correspondant aux statistiques des différentes catégories d'attaque, identifiées lors de l'analyse des fichiers logs.

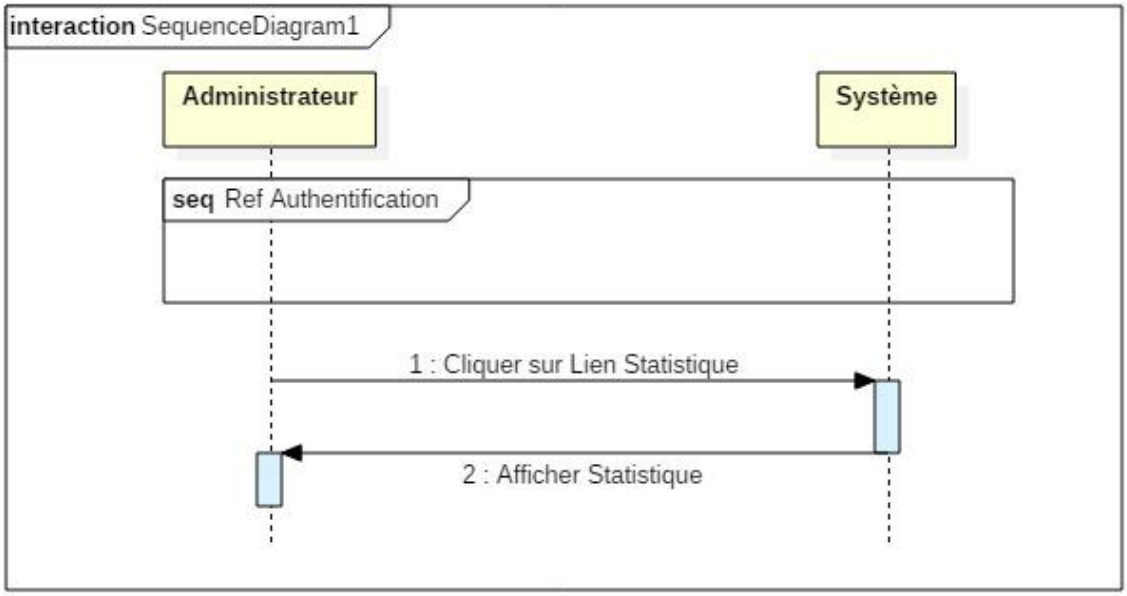

Figure 21:Diagramme de séquence de cas d'utilisation « afficher statistiques»

Conclusion

Ce chapitre nous a permis de mettre en avant les phases nécessaires à la réalisation de notre application à savoir l'architecture du système et les diagrammes de conception.

A cette étape, il devient possible de passer à la phase de réalisation qui constitue l'objet du chapitre suivant.

Réalisation

Introduction

Dans ce chapitre, nous allons commencer par présenter l'environnement matériel et logiciel du projet. Ensuite, nous nous intéressons à la description de quelques interfaces de notre application.

I. Environnement matériel

Nous avons utilisé pour les besoins de ce projet un ordinateur portable DELL Inspiron ayant caractérisé par :

- Un Processeur : Inlel(R) Core(TM) i3-2370M CPU @ 2.40GHz 2.40 GHz
- RAM : 4 Go
- Système d'exploitation Windows 7 Professionnel 32 bits.

II. Environnement logiciel

L'environnement logiciel utilisé pour développer notre application est :

- Langage de modélisation : UML.
- Langage de programmation : PHP, Python.
- Langage de développement : HTML, XML, JavaScript, CSS.
- Serveur Web : Apache.
- Système d'exploitation : Ubuntu 14.10
- Logiciel : StartUML, Visio 2007.
- Machine virtuelle : VMware Workstation version 8.

III. Présentation des interfaces

Nous exposerons quelques interfaces de notre application, en essayant à chaque fois de décrire les différents objets interactifs mis à la disposition de l'utilisateur.

1. Interface d'inscription

Lorsque le client n'a pas de compte sur la plateforme, il pourrait accéder, via le lien inscription, tout en saisissant quelques informations personnelles et les paramètres d'accès. La figure 22 illustre ces informations.

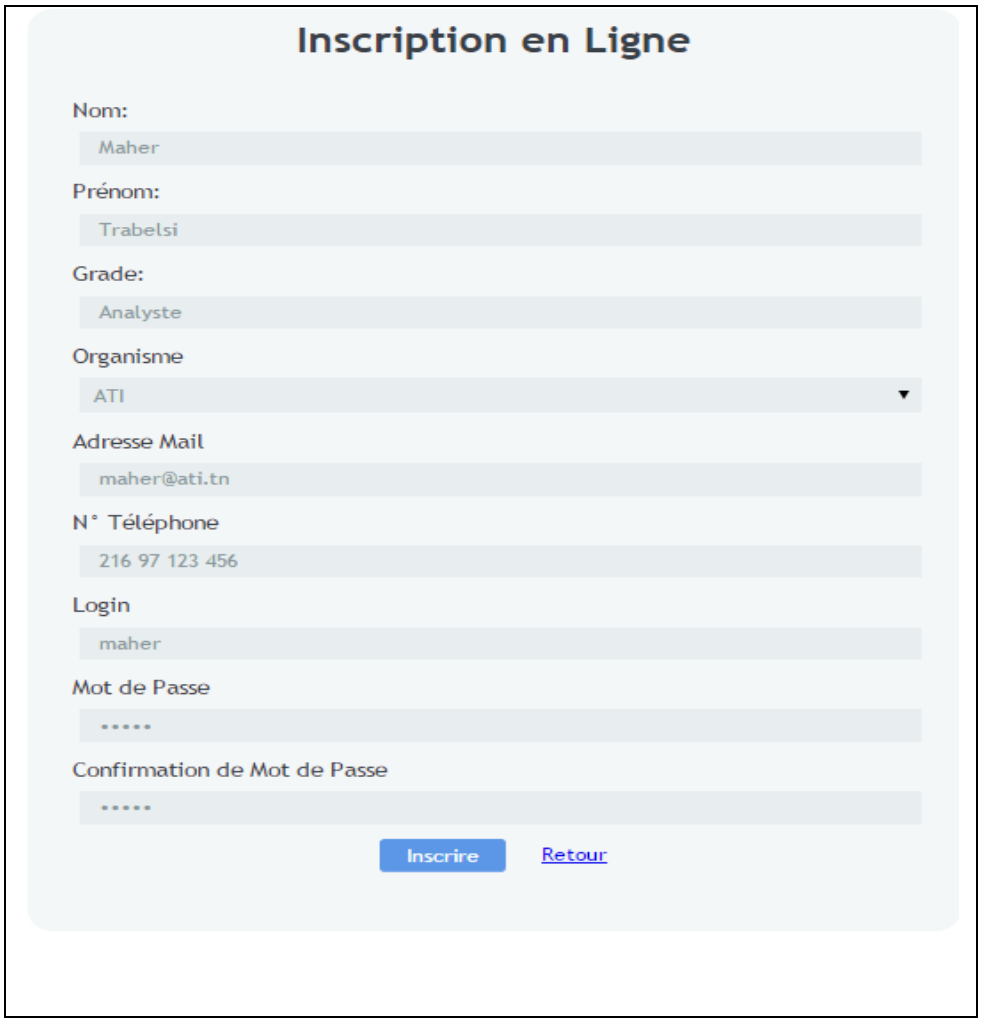

Figure 22 : Interface d'inscription

2. Interface d'authentification

La figure 23 représente l'interface d'authentification de notre application. Seul les utilisateurs internes (administrateur, investigateur) et les utilisateurs externes (client) qui sont inscris dans la plateforme et peuvent y accéder.

Figure 23 : Interface d'authentification

3. Interface de déclaration

Figure 24 : Interface de déclaration un incident (information générale)

Pour déclarer un incident, un client doit enregistrer toutes les informations nécessaires concernant cet incident pour qu'il soit convenablement analyser. Le client accède au formulaire de déclaration et saisi les informations générales, parmi ces informations montrées dans la figure 24 sont la date, l'heure, l'organisme impacté, la source d'identification, l'hébergeur, l'adresse IP et le système d'exploitation.

Autre que les informations générales, le client identifie le type de l'incident, ainsi que télécharger le fichier log à analyser afin de déterminer le scénario d'attaque montré dans la figure 25.

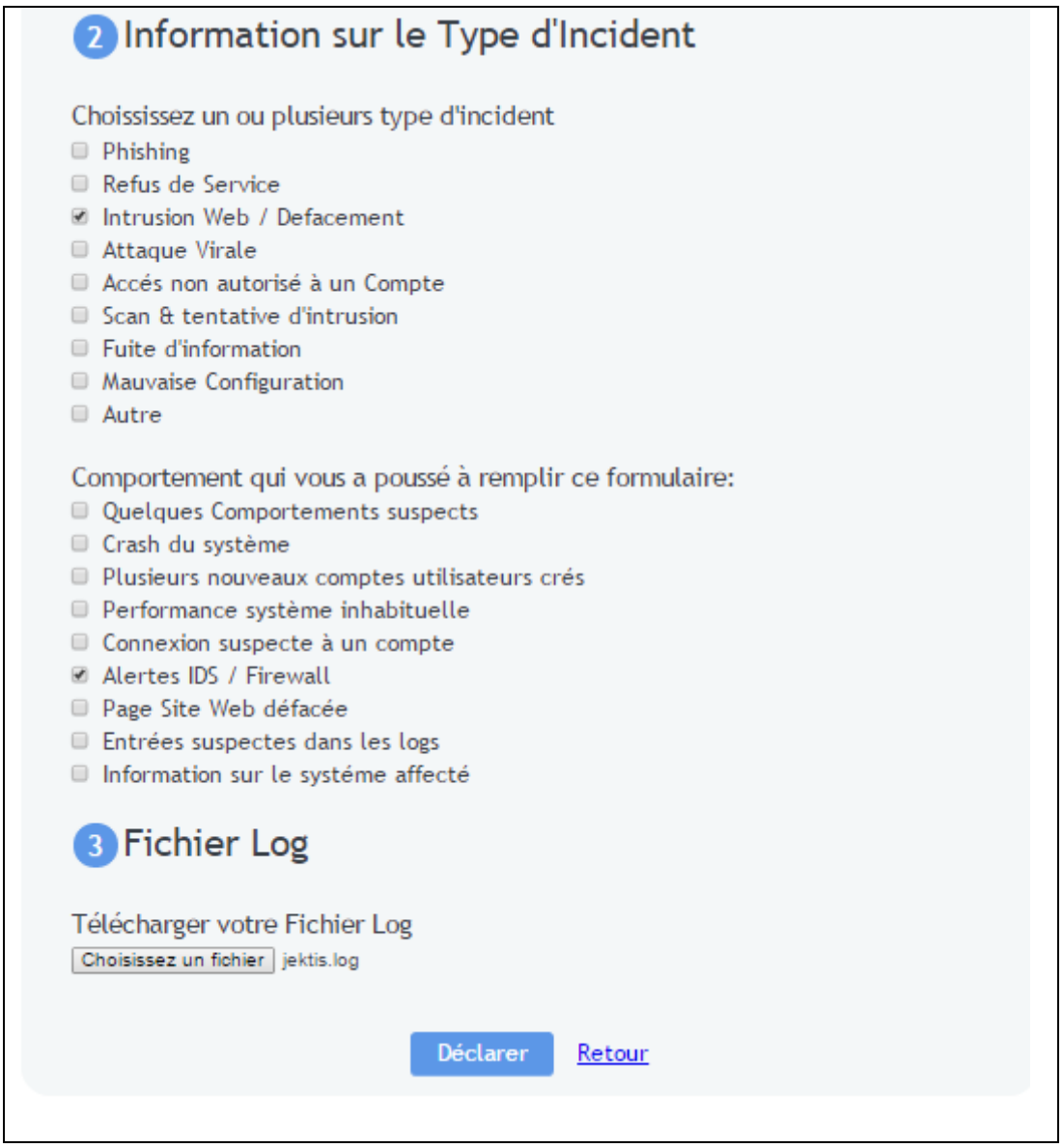

Figure 25: Interface de déclaration un incident (type de l'incident)

4. Interface de travail

Après authentification, chaque utilisateur accède à son espace de travail. La figure 26 montre l'espace de travail de l'administrateur qui contient les différentes tâches que l'administrateur doit effectuer tels que la gestion de son profil, la gestion de client, d'investigateur, d'incident et la consultation de statistique.

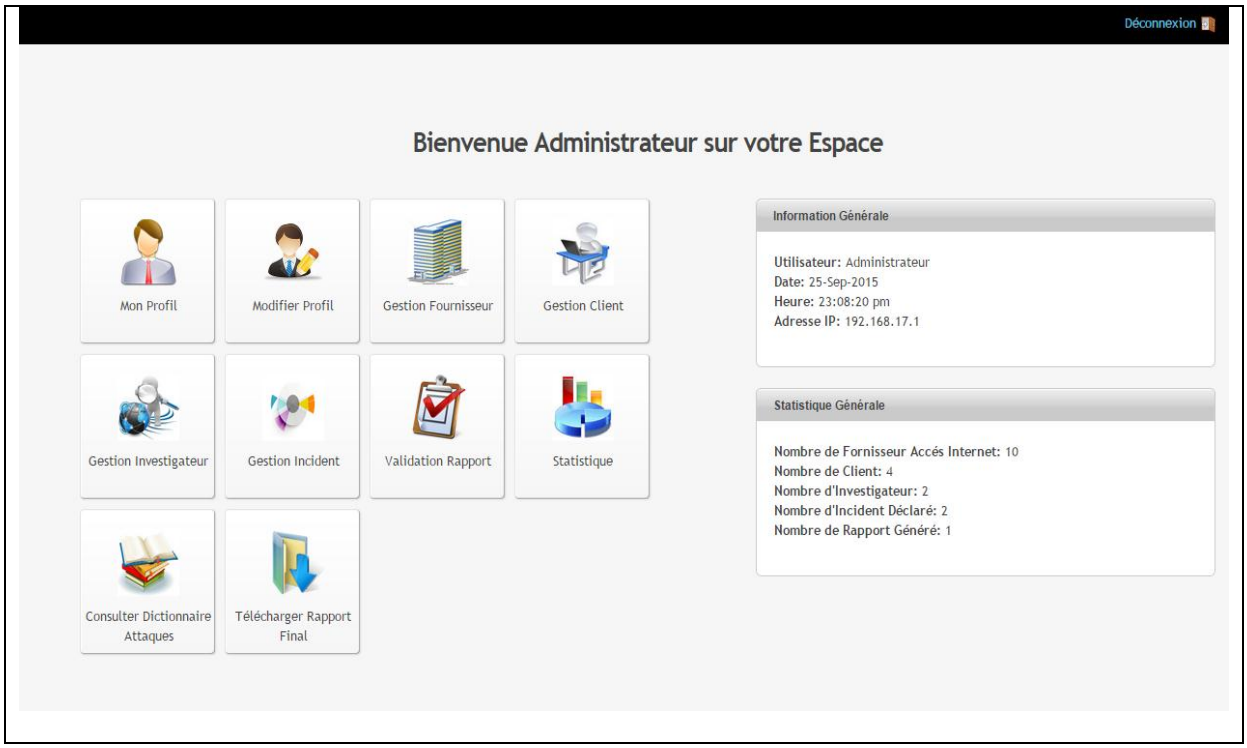

Figure 26: Interface espace de travail de l'administrateur

5. Interface de gestion des incidents

La figure 27 présente la liste des incidents déclarés, cette liste contient les informations concernant chaque incident ainsi que l'action à faire sur cet incident. Alors l'administrateur peut consulter ou supprimer un incident, affecter un investigateur ou bien télécharger le fichier log.

Figure 27: Interface de gestion des incidents

6. Interface d'affectation un investigateur à un incident

Après la sélection d'un incident en cours d'affectation, l'administrateur choisit l'action « affecter investigateur ». Alors, il sélectionne un investigateur parmi la liste illustrée dans la figure 28 pour l'affecter à l'incident afin de l'analyser.

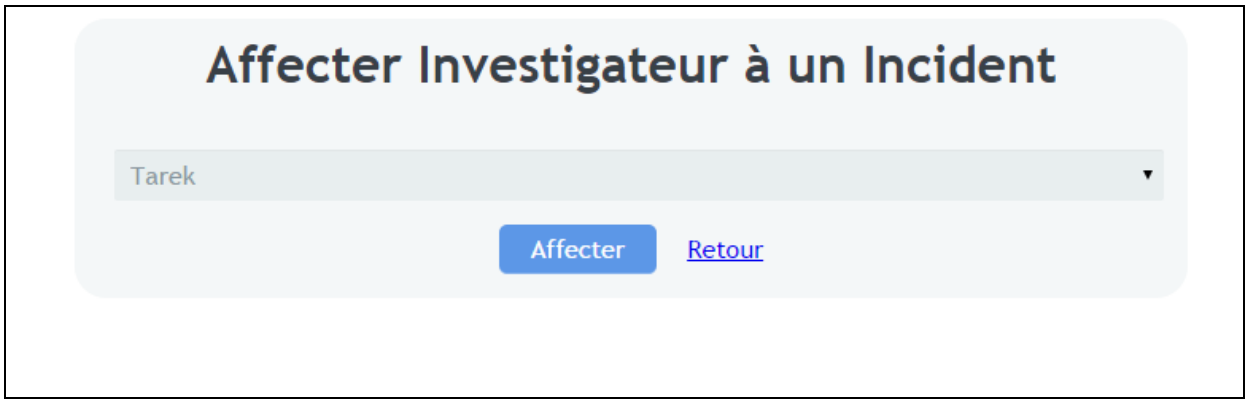

Figure 28: Interface d'affectation

7. Interface de consultation un dictionnaire d'attaque

Après l'analyse de l'incident et la génération du rapport, l'administrateur peut consulter le dictionnaire d'attaque de chaque rapport. Le tableau affiché dans la figure 29 contient les informations contenant dans chaque rapport généré tels que le nom d'attaque, l'impact de de chaque attaque, le raison de chaque impact et la ligne correspond à cette attaque.

Figure 29: Interface de consultation dictionnaire d'attaque

8. Interface de statistique

Parmi les tâches de l'administrateur, il peut consulter les statistiques concernant les différentes catégories d'attaque, identifiées lors de l'analyse des fichiers logs. La figure 30 montre le graphique qui affiche le pourcentage de chaque attaque.

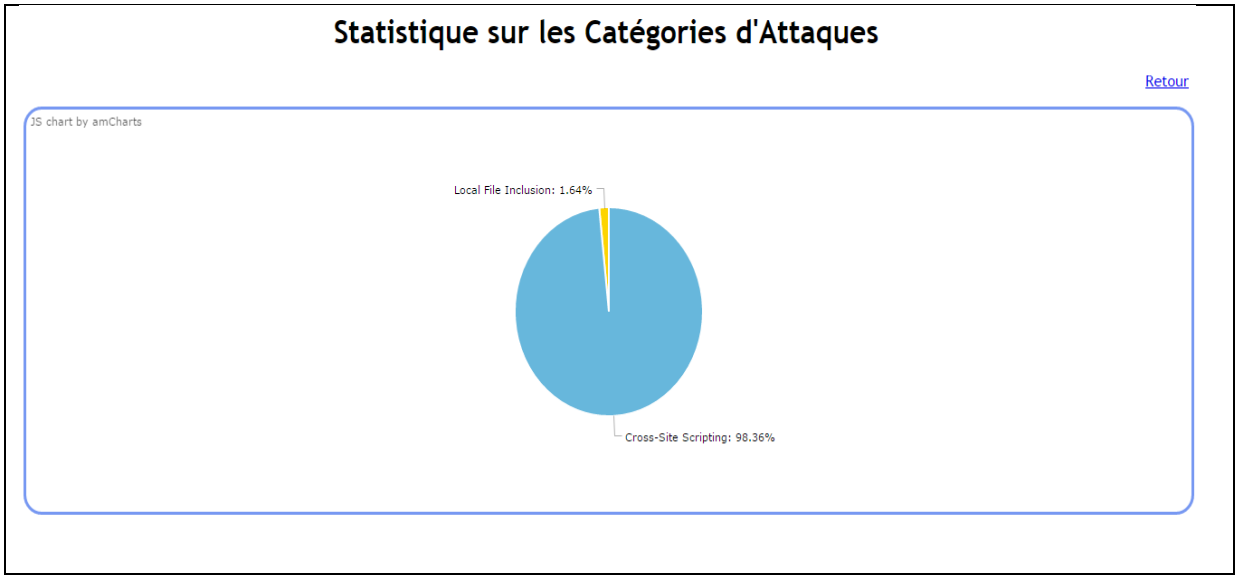

Figure 30: Interface de statistique

Conclusion

Nous sommes parvenus au terme de ce chapitre dont l'objectif principal était de présenter le produit final obtenu. A cet effet, nous avons tour à tour présentés les outils matériels et logiciels utilisés d'une part, puis nous avons présenté des interfaces du notre système.

Conclusion générale

L’objectif de ce projet de fin d’étude était de développer une application qui permet d’automatiser l’analyse des fichiers logs afin d’identifier les cybers criminel, de déterminer le scénario possible de l’incident et de modéliser les données des attaques selon plusieurs axes.

Nous avons confrontés des contraintes dans la phase du développement. En effet, nous avons passé plus de temps pour comprendre le script « scalp » programmé en langage « phyton » et les expressions régulières de dictionnaire d’attaque « défault filter ».

Cependant, nous pouvons toujours y apporter quelques améliorations qui feront de cette application. En ce qui concerne la côté ergonomique de l’application. En effet nous pourrons aussi développer une partie de l’application dédiées aux mobiles.

De plus, nous pouvons même développer une application qui permet la génération des modèles de rapports spécifiques à l'ANSI.

Webographie

[1]http://www.journaldunet.com/developpeur/algo-methodes/tutoriel-pratique/les-fichiers-log-des-indicateurs-utiles.shtml

[2]http://www.journaldunet.com/solutions/dsi/erreur-et-codes-http.shtml

[3]https://fr.wikipedia.org/wiki/AWStats

[4] https://fr.wikipedia.org/wiki/Webalizer

[5]http://www.01net.com/telecharger/windows/Internet/vrml/fiches/23742.html

[6]http://www.haage partner.de/sawmill/workshops/using_sawmill_to_report_on_custom_log_data/using_sawmill_to_report_on_custom_log_data.html

[7]http://www.predictiveanalyticstoday.com/list-security-event-management-log-analysis-software/

[8]http://www.freewebs.com/fresma/Formation/UML/Processus%20Unifie.pdf

[9]http://www.uml-sysml.org/modelisation-objet/pourquoi-uml

[10]http://wapiti.telecom-lille1.eu/commun/ens/peda/options/st/rio/pub/exposes/exposesser2010-ttnfa2011/lemarchand-hardy/architectures_p2p.htm

[11]http://deptinfo.cnam.fr/new/spip.php?pdoc5259

[12]http://www.grosmax.uqam.ca/Martin/INF5153/tab23tier.htm