

Table des matières :

Table des matières :	1
Table de figure :	4
Table des tableaux :	5
Liste des abréviations :	6
Présentation générale :	7
Introduction générale :	8
1. Introduction.....	9
2. Service de messagerie :	9
3. Fonctionnement de la messagerie :	9
4. Les risques existants :	9
4.1 Perte de confidentialité :	10
4.2 Usurpation d'identité :	10
4.3 Modification du contenu d'un message :	10
5 Le sens d'une application de messagerie et d'échanges sécurisé :	11
6 Langage UML et le processus UP :	11
6.1 Le langage UML	11
6.1.1 Définition :	11
6.1.2 UML est une norme	12
6.2 UML est un langage de modélisation objet	12
6.2.1 Points forts d'UML :	12
6.2.2 Points faibles d'UML :	12
6.2.3 Différents types de diagramme UML	12
6.3 UP (unified process)	13
6.3.1 Définition:	13
6.3.2 Les Caractéristiques d'UP :	13
6.4 Les Activités.....	14
6.4.1 Expression des besoins :	14
6.4.2 Analyse :	14
6.4.3 Conception	14
6.4.4 Implémentation.....	14
6.4.5 Test	15
7 Conclusion	15

Spécifications des besoins :	16
1 Introduction.....	17
2 Présentation du projet :	17
3 Expression des besoins :.....	17
3.1 Les besoins fonctionnels :.....	17
3.2 Les besoins non fonctionnels :	17
4 Identification des acteurs :.....	18
5 Diagramme de cas d'utilisation globale:	18
5.1 Le cas d'utilisation "Authentification"	19
5.2 Le cas d'utilisation "Inscription"	20
5.3 Le cas d'utilisation « Ajout d'un amis »	22
5.4 Le cas d'utilisation « Envoyer message »	23
5.5 Le cas d'utilisation « Passer un appel vidéo »	24
5.6 Le cas d'utilisation « Déconnexion»	25
6 Conclusion :	25
Analyse	26
1 Introduction :.....	27
2 Définition de diagramme de séquence :	27
2.1 Cas d'authentification :	27
2.2 Cas Inscription :	28
2.3 Cas envoi de message :.....	28
2.4 Cas ajouter des amis :.....	29
2.5 Cas appel vidéo :.....	30
2.6 Cas déconnexion :.....	30
3 Conclusion :	31
Conception :	32
1 Introduction.....	33
2 Réalisation du diagramme de classe :	33
3 Le modèle logique de données :	34
3.1 Règles de passages	34
3.2 Règles de normalisation	34
4 Diagramme de déploiement :.....	34
5 Conclusion	35
Réalisation	36

1	Introduction.....	37
2	Outils et environnement de développement :.....	37
3	Moyens de sécurité:	40
4	Présentation des interfaces de l'application :	43
4.1	Interface « Accueil » :	43
4.1	Interface « Authentification » :	43
4.2	Interface « Inscription » :	44
4.4	Interface « Envoi de message» :.....	45
4.5	Interface « liste des amis » :	46
4.3	Interface « invitation » :	46
4.3	Interface « stockage de message dans la base de donnée » :	46
5	Conclusion	47
	Conclusion générale :	48
	Bibliographie :	49

Table de figure :

Figure 1 Risque perte de confidentialité du message.	9
Figure 2 : Risque Usurpation d'identité.	9
Figure 3 : Risque Modification du contenu d'un message.	9
Figure 4 : Diagramme UML	12
Figure 5 : Démarche type UP	14
Figure 6 : Diagramme de contexte	16
Figure 7 : Diagramme global des cas d'utilisation	16
Figure 8 : Diagramme des cas d'utilisations « Authentification ».	18
Figure 9: Diagramme des cas d'utilisations « Inscription »	19
Figure 10 : Diagramme des cas d'utilisations « Ajout d'un ami »	20
Figure 11 : Diagramme des cas d'utilisations « Envoyer message»	21
Figure 12 : Diagramme des cas d'utilisations « Appel vidéo»	22
Figure 13 : Diagramme des cas d'utilisations « Déconnexion»	23
Figure 14 : Diagramme de séquence de cas d'utilisations « Authentification»	24
Figure 15 : Diagramme de séquence de cas d'utilisations « Inscription»	25
Figure 16 : Diagramme de séquence de cas d'utilisations «Envoyer un message»	26
Figure 17 : Diagramme de séquence de cas d'utilisations «Ajouter des amis»	27
Figure 18 : Diagramme de séquence de cas d'utilisations «Appel vidéo»	28
Figure 19 : Diagramme de séquence de cas d'utilisations « Déconnexion»	29
Figure 20 : Diagramme de classe.	30
Figure 21 : Diagramme de déploiement	31
Figure 22 : fonctionnement de l'algorithme AES.....	35

Table des tableaux :

Tableau 1 : descriptif du cas d'utilisation «Authentification »	17
Tableau 2 : descriptif du cas d'utilisation « Inscription »	19
Tableau 3 : descriptif du cas d'utilisation « Ajout d'un ami »	20
Tableau 4 : descriptif du cas d'utilisation « Envoyer message »	21
Tableau 5 : descriptif du cas d'utilisation "Passer un appel vidéo"	22
Tableau 6 : descriptif du cas d'utilisation « Déconnexion »	23

Liste des abréviations :

Https : *Hypertext Transfer Protocol Secure*

Ssl : Secure Socket Layers

AES : Advanced Encryption Standard

PHP : **H**ypertexte **P**réprocesseur

WampServer : « Windows Apache MySQL PHP » server

PMA : PhpMyAdmin

MySQL : My Structured Query Language

HTML5 : **H**yper**T**ext **M**arkup **L**anguage

CSS : Cascading Style Sheet

Js :JavaScript

UP : Processus Unifié

UML : Unified Modeling Language

SQL : Structured Query Language

Présentation générale :

Il ne fait désormais plus aucun doute que les technologies de l'information et de la communication représentent la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité moderne. Ces utilisations apportent des multiples confort et propose des solutions à tous les problèmes, aussi bien dans le domaine professionnel que pour des applications personnelles.

Parmi ces Technologies de l'information et de la communication (TIC), le service messagerie est rapidement développé dans les organisations aux cours de ces dix dernières années, par sa facilité d'utilisation et son utilité perçue. Désormais, elle représente l'outil de travail le plus utilisé.

C'est dans ce cadre que s'intègre ce projet qui consiste dans l'étude, la conception et le réalisation d'une application de messagerie et d'échange sécurisé destinée au public qui vise à garantir la sécurité des échanges et la communication entre utilisateurs

Le présent rapport s'articule autour de cinq chapitres qui sont les suivant :

Dans le premier chapitre, nous allons présenter brièvement notre thème ensuite la démarche up et le langage de modélisation UML

Le deuxième chapitre, nous allons établir l'expression des besoins dans lequel nous exposons les besoins fonctionnels qui conduisent à l'élaboration des modèles de cas d'utilisation et les besoins non fonctionnels (techniques)

Le troisième chapitre, est dédié à l'analyse des besoins qui va accéder à une compréhension détaillée des besoins et des exigences du client

Le quatrième chapitre, la conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation

Le cinquième chapitre, réalisation (implémentation et test) précise l'environnement du travail et les codes associés et présente les principales interfaces de l'application.

Finalement, nous clôturons le rapport par une conclusion générale qui présente le bilan de ce projet.

CHAPITRE I :

Introduction générale

1. Introduction

La messagerie occupe une place de plus en plus primordiale par rapport aux moyens de communication traditionnels pour cela on va s'approfondir sur ces détails dans ce qui suit, et pour modéliser celle ci d'une manière claire et précise la structure et le comportement de notre système indépendamment de tout langage de programmation nous allons adopter la démarche up et le langage de modélisation UML.

2. Service de messagerie :

Un service de messagerie, dans sa forme la plus basique, est un service permettant essentiellement l'échange de messages textuels entre les différents utilisateurs enregistrés (ayant une adresse électronique valide) et connectés à un réseau informatique via un intermédiaire d'ordinateur. Cet échange de messages peut s'effectuer en différentes manières, c'est-à-dire il n'est pas nécessaire que le destinataire soit connecté au moment de l'envoi, son message sera enregistré sur un serveur et il pourra le consulter ultérieurement. On parle à ce moment de la messagerie simple. Par ailleurs, l'échange peut aussi se faire en temps réel et on parle à ce moment de la messagerie instantanée.

3. Fonctionnement de la messagerie :

La messagerie est donc un moyen de communiquer en privé avec d'autre personnes. Le client se connecte à l'application qui contient les informations sur tous les utilisateurs inscrits, connectés ou non. Chaque personne possède un pseudonyme qui n'est pas forcément unique, et un identifiant unique dans la base de données du serveur. Un mot de passe est aussi indispensable pour se connecter au service. Deux personnes peuvent communiquer en direct si elles sont simultanément connectées. Sinon, elles ont la possibilité de consulter leurs messages dans leur boîte aux lettres au moment où elles se connectent.

4. Les risques existants :

Comme tout système informatique, la messagerie se trouve face à des risques et des menaces qui touchent à l'intégrité et la confidentialité des données. On peut identifier trois catégories de risque qui sont les suivantes :

1. **Perte de confidentialité** : les messages, quand ils sont envoyés en clair ils peuvent être facilement lus par des tierces parties (voir figure 1.2).



Figure 1.1– Risque : perte de confidentialité du message.

2. **Usurpation d'identité** : Un message peut être " forgé ", c'est à dire fabriqué de toutes pièces, en indiquant une fausse identité pour l'expéditeur. Un attaquant peut alors prendre l'identité d'une personne connue ou inconnue (voir figure 3)



Figure 1.2 – Risque : Usurpation d'identité.

3. **Modification du contenu d'un message** : Un message peut être intercepté, modifié, puis relayé à son destinataire. Par exemple, il est techniquement possible d'intercepter un message, de modifier le corps du texte puis de le relayer normalement. Rien ne garantit que le message que l'on reçoit correspond bien à celui qui a été envoyé par son expéditeur. Voir figure 1.4.

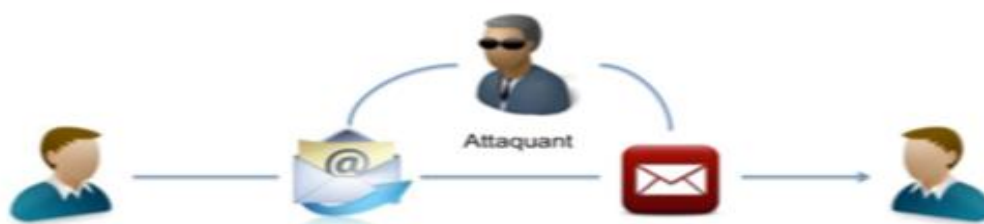


Figure 1.3 – Risque : Modification du contenu d'un message.

5. Le sens d'une application de messagerie et d'échanges sécurisé :

le but de notre application qui est destinée au large public, et qui nous offre la liberté d'accéder à nos messages depuis n'importe quel poste, n'importe quel FAI (Fournisseur accès Internet) et n'importe quel pays dans le monde (car c'est une messagerie web) , dans le but de garantir une communication munis d'une amélioration de la sécurité pour protéger la vie privée des utilisateurs pour cela on a intégrer un mécanisme fondamental, il s'agit du chiffrement qui consiste à garantir la confidentialité des données numériques stockées sur des systèmes informatiques ou transmises via Internet ou d'autres réseaux. Les algorithmes de chiffrement modernes jouent un rôle crucial dans la sécurité des systèmes informatiques et des communications, car ils assurent non seulement la confidentialité, mais également les éléments de sécurité essentiels suivants :

- **Authentification** : permet de vérifier l'origine d'un message.
- **Intégrité** : apporte la preuve que le contenu d'un message n'a pas été modifié depuis son envoi.
- **Non-répudiation** : empêche l'expéditeur d'un message de nier avoir envoyé ce message.

6. Langage UML et le processus UP

6.1. Le langage UML :

6.1.1. Définition :

C'est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter, spécifier les artefacts de systèmes logiciels, de plus il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquissé des architectures logicielles, concevoir des solutions et communiquer des points de vue, comme il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

6.1.2. UML est une norme :

Il est nécessaire qu'une méthode objet soit définie de manière rigoureuse et unique afin de lever les ambiguïtés. De nombreuses méthodes objet ont été définies, mais aucune n'a su s'imposer en raison du manque de standardisation. C'est pourquoi l'ensemble des acteurs du monde informatique a fondé en 1989 l'**OMG** (**O**bject **M**anagement **G**roup), une organisation à but non lucratif, dont le but est de

mettre au point des standards garantissant la compatibilité entre des applications programmées à l'aide de langages objet et fonctionnant sur des réseaux hétérogènes (de différents types). A partir de 1997, UML est devenue une norme de l'OMG, ce qui lui a permis de s'imposer en tant que méthode de développement objet et être reconnue et utilisée par de nombreuses entreprises.

6.2. UML est un langage de modélisation objet

UML comble une lacune importante des technologies objet, il permet d'exprimer, d'élaborer et de modéliser au sens de la théorie des langages, de ce fait il contient les éléments constitutifs de ce derniers : concepts, une syntaxe et une sémantique.

6.2.1. Points forts d'UML :

- Un gain de précision.
- Un gage de stabilité.
- Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel.

6.2.2. Points faibles d'UML :

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- L'intégration d'UML dans un processus n'est pas triviale, et améliorer un processus est une tâche complexe et longue.
-

6.3. Différents types de diagramme UML

UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagramme sont répartis en deux grands groupes représentés ci-dessous:

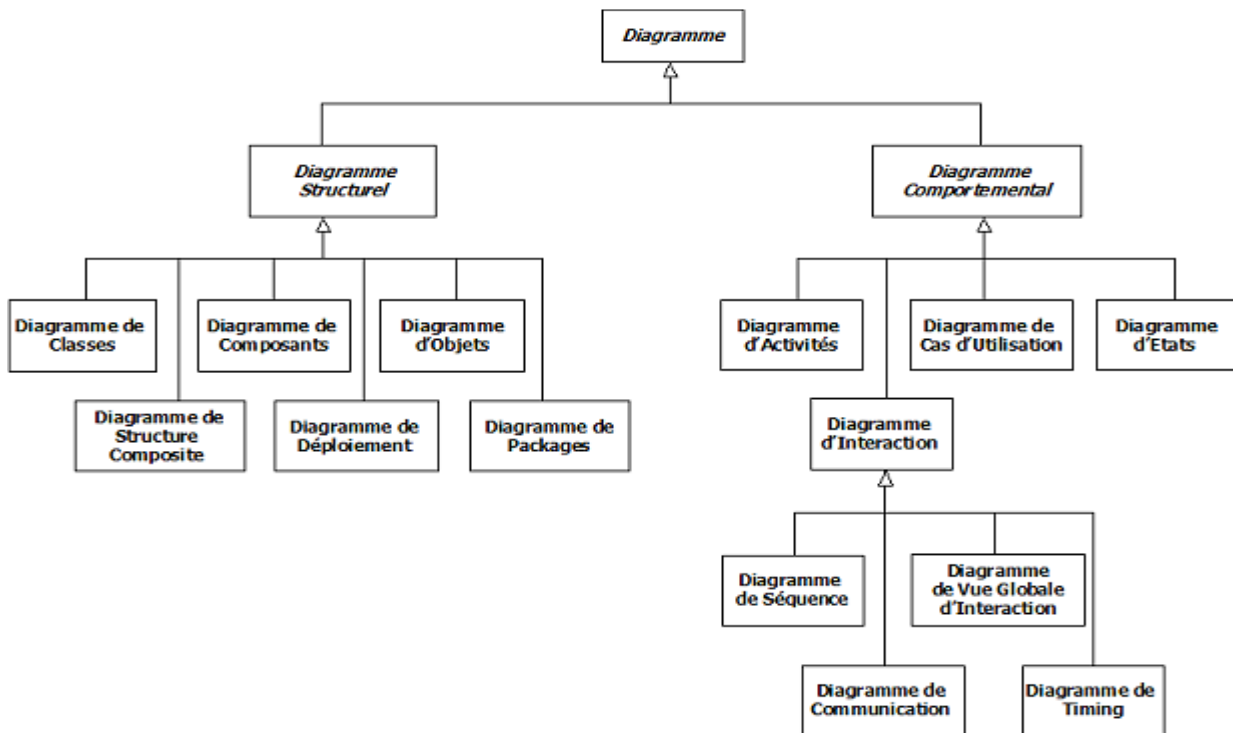


Figure 1.4 – Risque :Diagramme UML.

6.4. UP (unified process)

6.4.1 Définition:

Le processus unifié est un processus de développement logiciel construit sur UML. Il est itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. Il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

C'est un patron de processus pouvant être adaptée à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

6.4.2 Les Caractéristiques d'UP :

- **Itératif et incrémental** : le projet est découpé en itérations de courte durée qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- **Centré sur l'architecture** : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML.

- **Piloté par les risques** : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- **Conduit par les cas d'utilisation** : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

6.4.3. Les Activités

Expression des besoins :

L'expression des besoins permet de:

- inventorer les besoins principaux et fournir une liste de leurs fonctions
- recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation
- appréhender les besoins non fonctionnels (technique) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

Analyse :

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de réaliser des spécifications permettant de concevoir la solution. Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les Structures sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système. Il peut être considéré comme une première ébauche du modèle de conception.

Conception :

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle constitue un point de départ à l'implémentation :

- elle décompose le travail d'implémentation en sous-système.
- elle crée une abstraction transparente de l'implémentation.

Implémentation

L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type. Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

Test

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

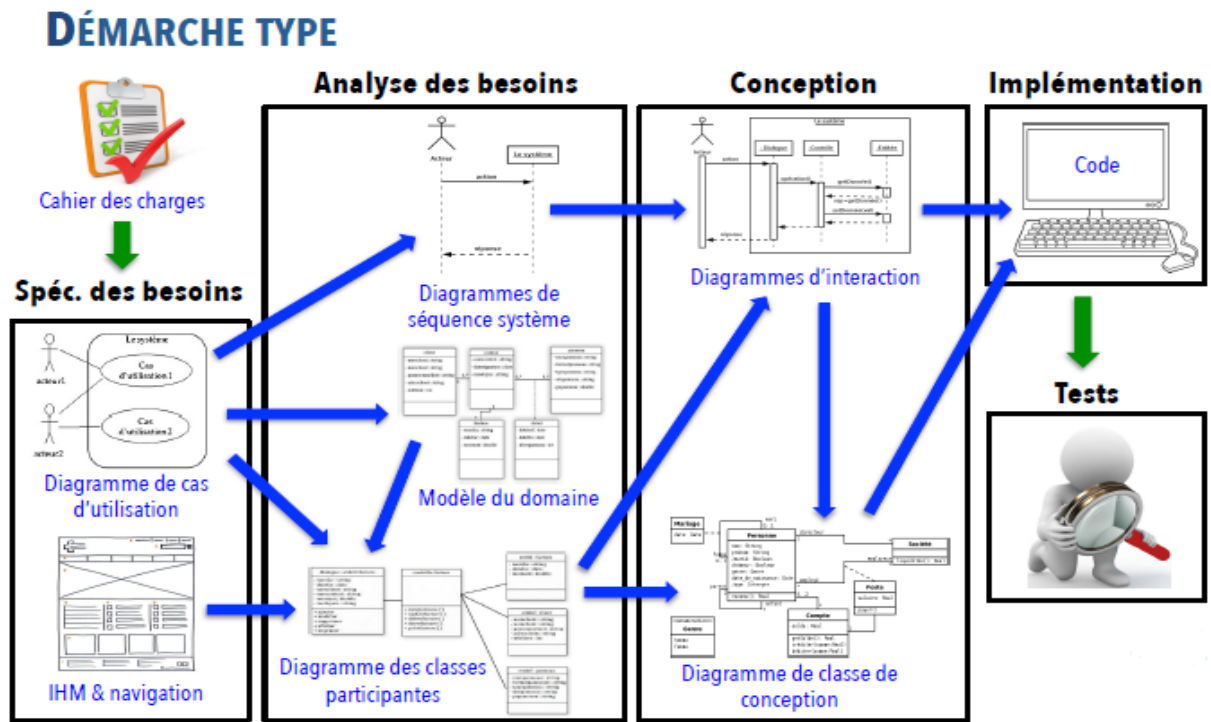


Figure 1.5 – Démarche type UP.

Conclusion

Dans ce chapitre, nous avons présenté la démarche et le langage de modélisation que nous utiliserons pour la réalisation de notre application et quelques notions qui se rapportent à notre thème. Prochainement, nous allons aborder l'étude des besoins et l'élaboration de cahier des charges.

CHAPITRE II

Expression des besoins

Introduction

Afin de garantir la réussite et l'efficacité de notre projet, il faut à ce stade du travail définir avec précision la bordure de la solution à développer. Pour cela Dans ce chapitre, nous commençons par définir les besoins fonctionnels et non fonctionnels de notre solution. Ainsi, nous exposons la description du contexte en identifiant les différents acteurs de notre application afin de mieux clarifier le domaine d'étude.

1. Présentation du projet :

L'application qui sera mise en œuvre par nos soins aura pour nom «Tchat », Cette dernière a pour but de communication et d'échange sécurisé.

2. Expression des besoins :

2.1. Les besoins fonctionnels :

Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Cette application doit couvrir principalement les besoins fonctionnels suivants :

- Enregistrement des utilisateurs lors de l'inscription.
- Authentification des utilisateurs.
- Envoyer et recevoir des messages.
- Gestion d'amis (Ajout, suppression et blocage des amis par nom d'utilisateur)
- Affichage de la liste des amis avec leurs statut(en ligne/hors ligne), accepter ou refuser des invitations.
- Se déconnecter (quitter l'application).
- Sécurité d'échange des messages.
- Permettre des conversations audio visuelle.

2.2 Les besoins non fonctionnels :

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système. Les principaux besoins non fonctionnels de notre application se résument comme suit :

- Le code doit être clair pour permettre des futures évolutions ou améliorations.
- L'ergonomie : l'application offre une interface conviviale et facile à utiliser.
- La sécurité : l'application doit respecter la confidentialité des données.
- Garantir l'intégrité et la cohérence des données à chaque mise à jour et à chaque insertion.

3. Identification des acteurs :

Un acteur représente un rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Dans notre cas, nous avons un acteur qui est l'Utilisateur ayant un accès au système via un contrôle d'accès (login et mot de passe). Les opérations qu'il peut effectuer sont mentionnées sur le diagramme de cas d'utilisation globale ci-dessous.



Figure 2.1-Diagramme de contexte

4. Diagramme de cas d'utilisation globale:

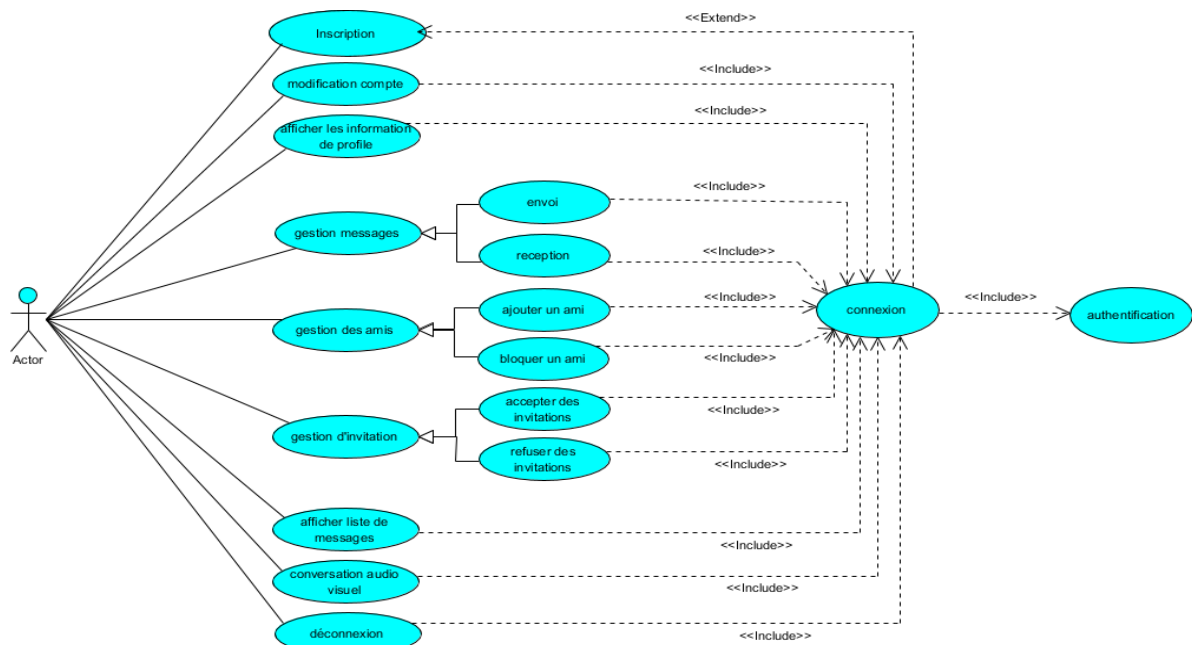


Figure 2.2-Diagramme global des cas d'utilisations

4.1 Le cas d'utilisation "Authentification" :

Sommaire d'identification	
<i>Titre</i>	Authentification
<i>But</i>	L'authentification de l'utilisateur
<i>Résumé</i>	Vérification de l'identité de l'utilisateur
<i>Acteur</i>	L'utilisateur
<i>Description des enchainements</i>	
<i>Pré condition</i>	
L'utilisateur doit avoir un compte	
<i>Scénario nominal</i>	
<ol style="list-style-type: none"> 1. L'utilisateur accède à l'application 2. L'utilisateur demande le formulaire de connexion 3. Le système affiche le formulaire 4. L'utilisateur saisi les informations 5. Le système vérifie les champs et les informations 6. L'utilisateur valide le formulaire 7. Le système affiche l'interface correspondante 	
<i>Enchainement alternatif</i>	
<p>* Champs obligatoires vides ou informations incorrectes</p> <ul style="list-style-type: none"> - Le système affiche un message d'erreur. - Le scénario reprend l'étape 4. 	

Table 2.1- descriptif du cas d'utilisation "Authentification" .

Le diagramme de cas d'utilisation Authentication:

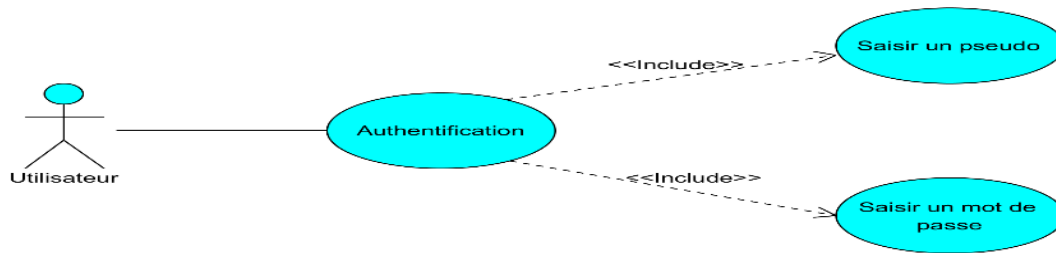


Figure 2.3-Diagramme des cas d'utilisations « Authentication ».

4.2 Le cas d'utilisation "Inscription"

Sommaire d'identification	
Titre	Inscription d'un nouvel utilisateur
But	Inscription
Résumé	L'utilisateur doit remplir le formulaire d'inscription et valider
Acteur	L'utilisateur
Description des enchainements	
<i>Pré condition</i>	<i>Post condition</i>
L'utilisateur remplit le formulaire d'inscription	L'utilisateur valide le formulaire d'inscription
Scénario nominal	
<ol style="list-style-type: none">1. L'utilisateur accède à l'application2. Le système affiche le formulaire d'inscription3. L'utilisateur saisi les informations4. Le système vérifie les champs5. L'utilisateur valide le formulaire6. Stocker les informations du formulaire dans la BDD	

Enchaînement alternatif

*** Champs obligatoires vides**

- Le système affiche un message d'erreur.
- Le scénario reprend l'étape 3.

Table 2.2-descriptif du cas d'utilisation "Inscription"

Le diagramme de cas d'utilisation Inscription:

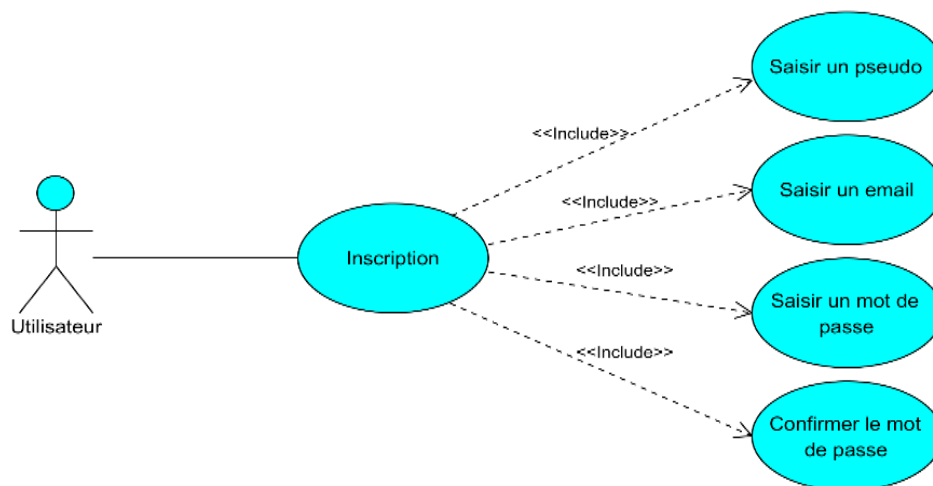


Figure 2.3-Diagramme des cas d'utilisations « Inscription ».

4.3 Le cas d'utilisation « Ajout d'un ami » :

Sommaire d'identification	
Titre	Ajout d'un ami
But	Ajouter des amis
Résumé	L'utilisateur peut ajouter plusieurs amis
Acteur	L'utilisateur
Description des enchainements	
Pré condition	
L'utilisateur doit d'abord s'authentifier	
Scénario nominal	
<ol style="list-style-type: none"> 1. L'utilisateur accède à l'application 2. L'utilisateur s'authentifie 3. L'utilisateur demande l'interface d'ajout 4. Le système affiche l'interface 5. L'utilisateur saisis le pseudo de l'ami à ajouter 6. Confirmer l'action 	

Table 2.3-descriptif du cas d'utilisation "Ajout d'un ami"

Le diagramme de cas d'utilisation Ajout d'un ami:

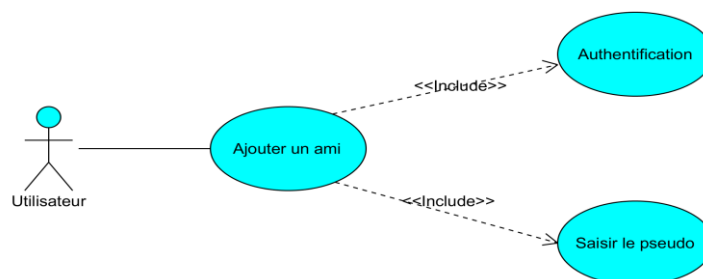


Figure 2.4-Diagramme des cas d'utilisations « Ajout d'un ami ».

4.4 Le cas d'utilisation « Envoyer message »

Sommaire d'identification	
Titre	Envoyer message
But	Envoyer des messages
Résumé	L'utilisateur peut envoyer des messages à ces amis
Acteur	L'utilisateur
Description des enchainements	
Pré condition	
L'utilisateur doit d'abord s'authentifier, le récepteur doit être dans la liste des amis	
Scénario nominal	
<ol style="list-style-type: none">1. L'utilisateur choisit l'ami dans la liste2. Le système affiche le formulaire des messages3. L'utilisateur saisit le message puis clique sur le bouton « Envoyer »	

Table 2.5-descriptif du cas d'utilisation "Envoyer message"

Le diagramme de cas d'utilisation « Envoyer messages »

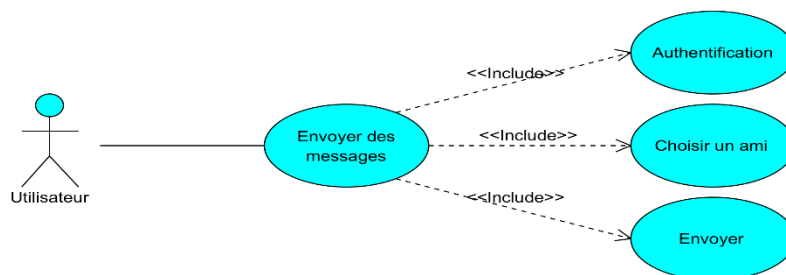


Figure 2.3-Diagramme des cas d'utilisations « Envoyer message ».

4.5. Le cas d'utilisation « Passer un appel vidéo »

Sommaire d'identification	
Titre	Passer un appel vidéo
But	Passer des appels vidéo
Résumé	L'utilisateur peut passer des appels vidéo à ces amis
Acteur	L'utilisateur
Description des enchainements	
Pré condition	
L'utilisateur doit d'abord s'authentifier, le récepteur doit être dans la liste des amis	
Scénario nominal	
1. L'utilisateur choisit l'ami dans la liste	
2. Le système affiche le formulaire des messages	
3. L'utilisateur clique sur le bouton « appel vidéo »	

Table 2.6-descriptif du cas d'utilisation "Passer un appel vidéo"

Le diagramme de cas d'utilisation « Appel vidéo »

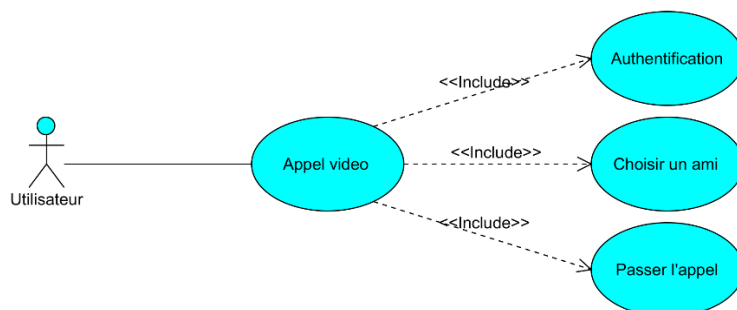


Figure 2.4-Diagramme des cas d'utilisations « Appel vidéo ».

4.6. Le cas d'utilisation « Déconnexion »

Sommaire d'identification	
<i>Titre</i>	Déconnexion
<i>But</i>	Déconnexion
<i>Résumé</i>	L'utilisateur peut quitter son compte
<i>Acteur</i>	L'utilisateur
<i>Description des enchainements</i>	
<i>Pré condition</i>	
L'utilisateur doit d'abord s'authentifier	
<i>Scénario nominal</i>	
<ol style="list-style-type: none">1. L'utilisateur accède à l'application2. L'utilisateur accède au menu de l'application3. Cliquer sur le bouton « Déconnexion »4. Le système affiche la page d'accueil	

Table 2.7-descriptif du cas d'utilisation "Déconnexion"

Le diagramme de cas d'utilisation « Déconnexion »

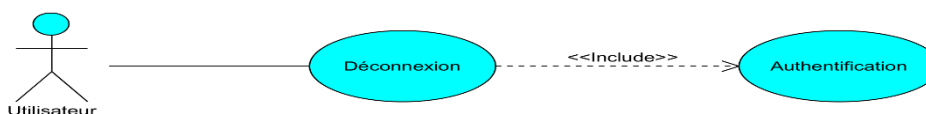


Figure 2.5-Diagramme des cas d'utilisations « Déconnexion ».

Conclusion :

A ce stade on a présenté les besoin fonctionnel et non fonctionnel de notre application qui seront encore détaillés dans les phases suivantes.

CHAPITRE III

Analyse

1. Introduction :

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension, la Préparation, la modification et la maintenance du futur système, L'objectif est d'accéder à une compréhension des besoins et des exigences du client. Dans notre cas on a utilisé le diagramme de séquence.

2. Définition de diagramme de séquence :

Le diagramme de séquence représente les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

2.1. Cas d'authentification :

Le diagramme de séquence suivant illustre les interactions nécessaires pour l'authentification.

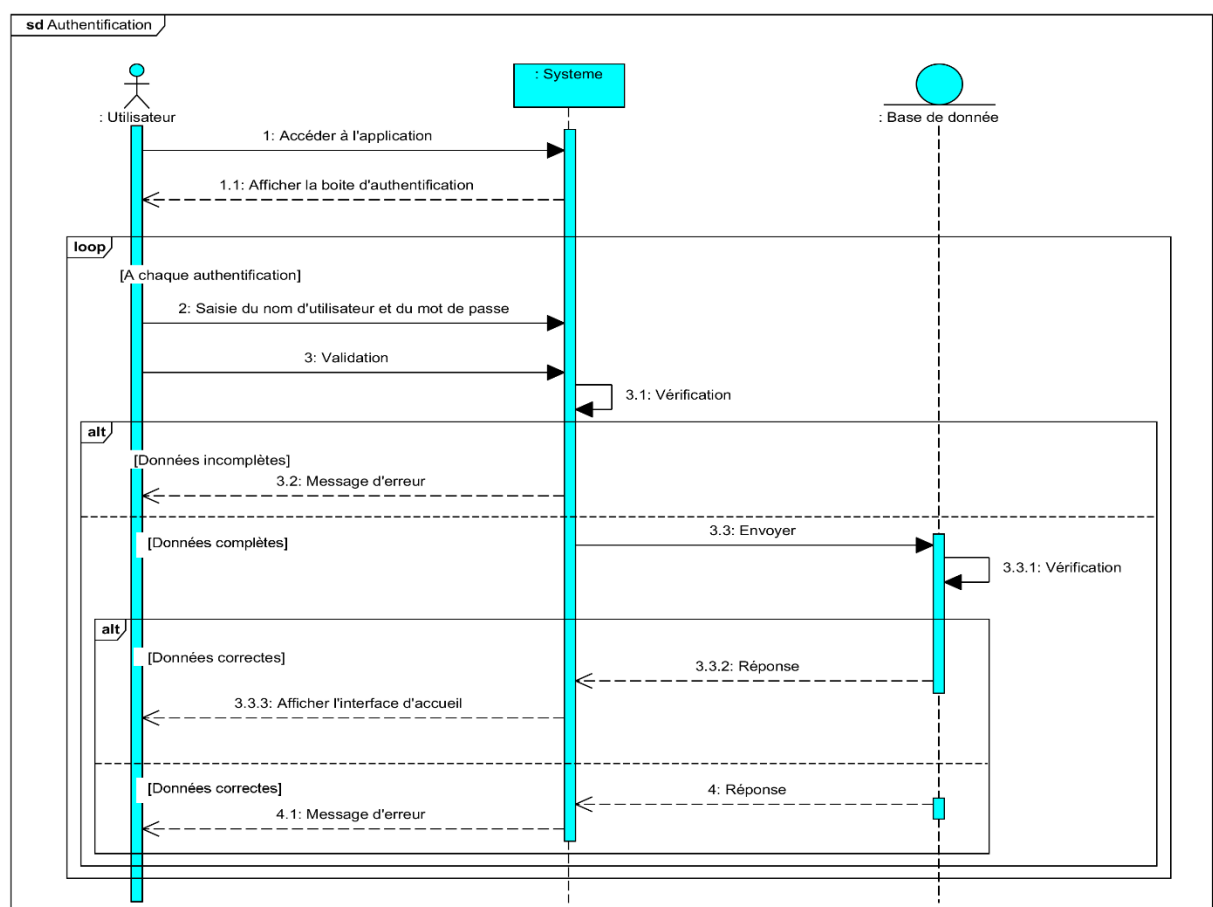


Figure 3.1-Diagramme de séquence de cas d'utilisations « Authentification ».

2.2 Cas Inscription :

Le diagramme de séquence suivant illustre les interactions nécessaires pour l'inscription.

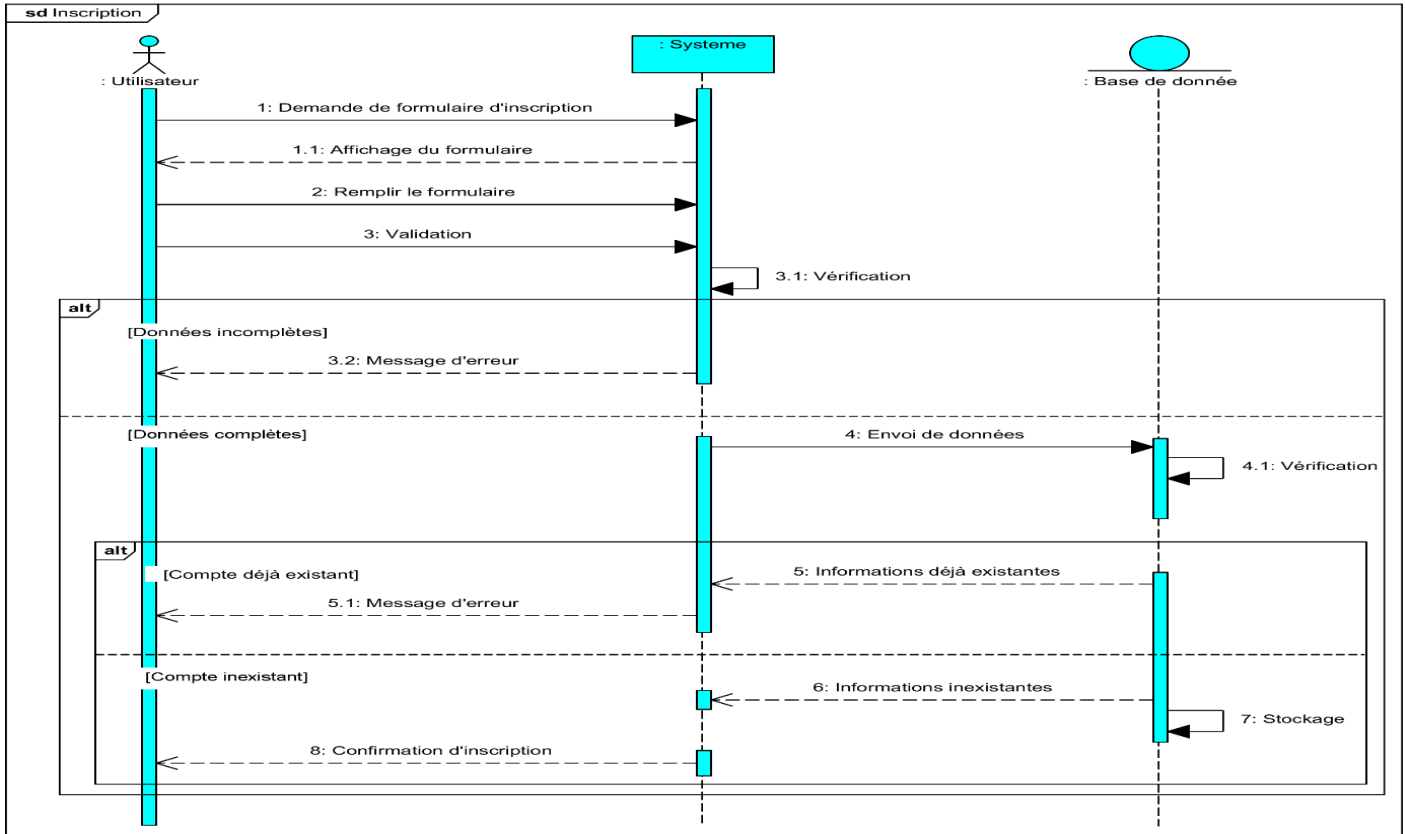


Figure 3.2-Diagramme de séquence de cas d'utilisations « Inscription ».

2.3. Cas envoi de message :

Le diagramme de séquence suivant illustre les interactions nécessaires pour envoyer des messages.

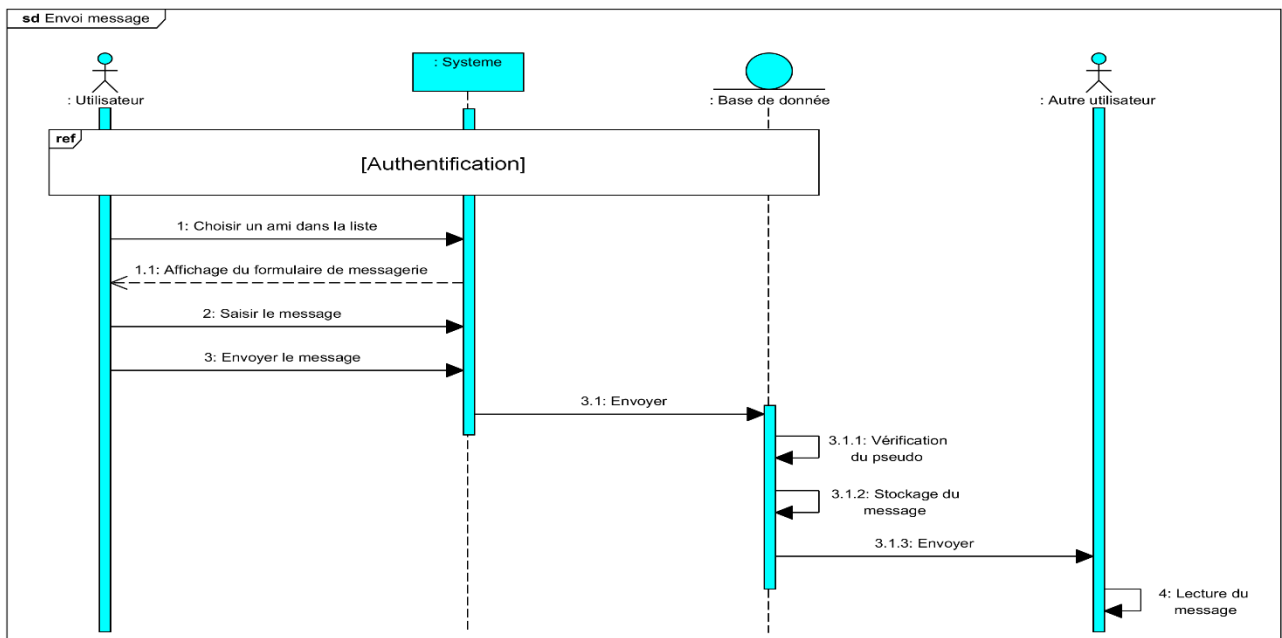


Figure 3.3-Diagramme de séquence de cas d'utilisations «Envoyer un message».

2.4. Cas ajouter des amis :

Le diagramme de séquence suivant illustre les interactions nécessaire pour ajouter des amis.

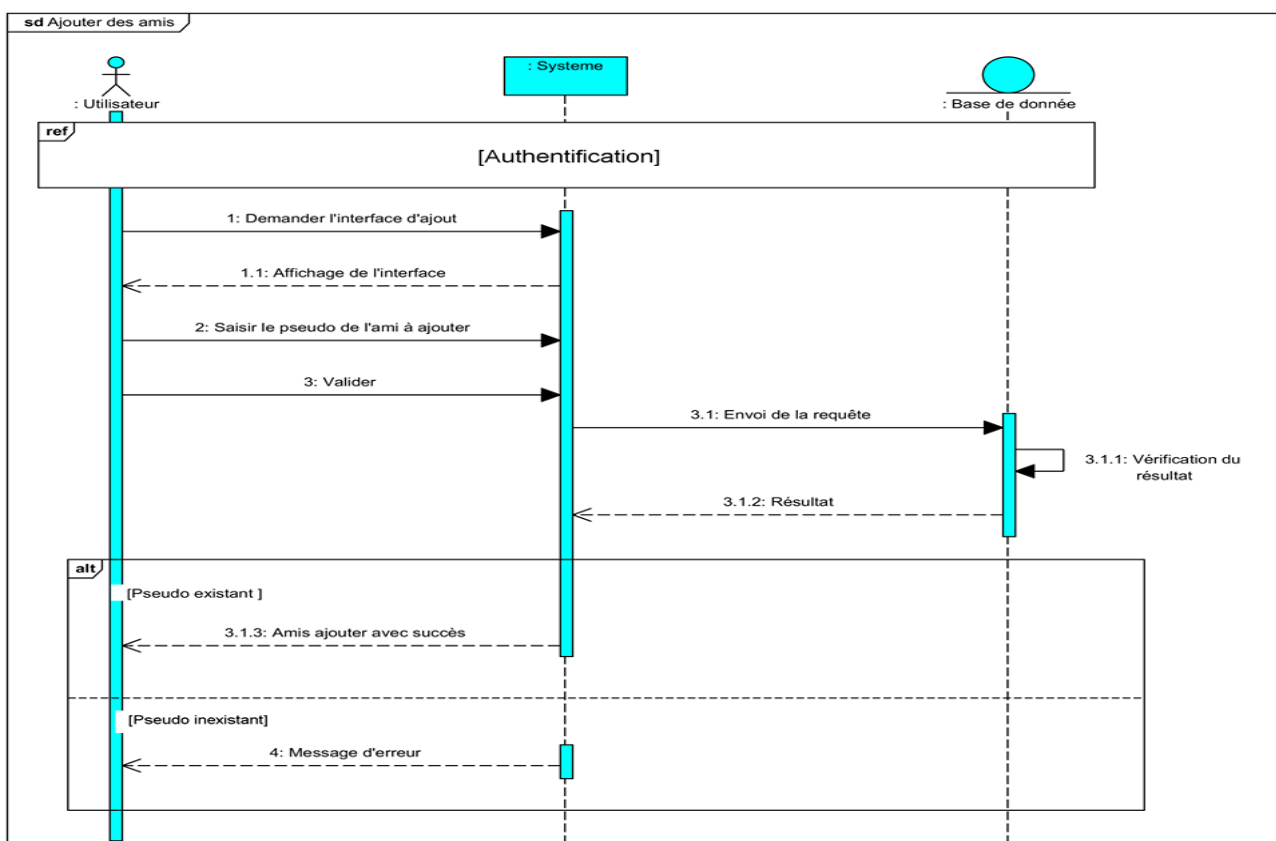


Figure 3.4-Diagramme de séquence de cas d'utilisations «Ajouter des amis».

2.5. Cas appel vidéo :

Le diagramme de séquence suivant illustre les interactions nécessaires pour appel vidéo.

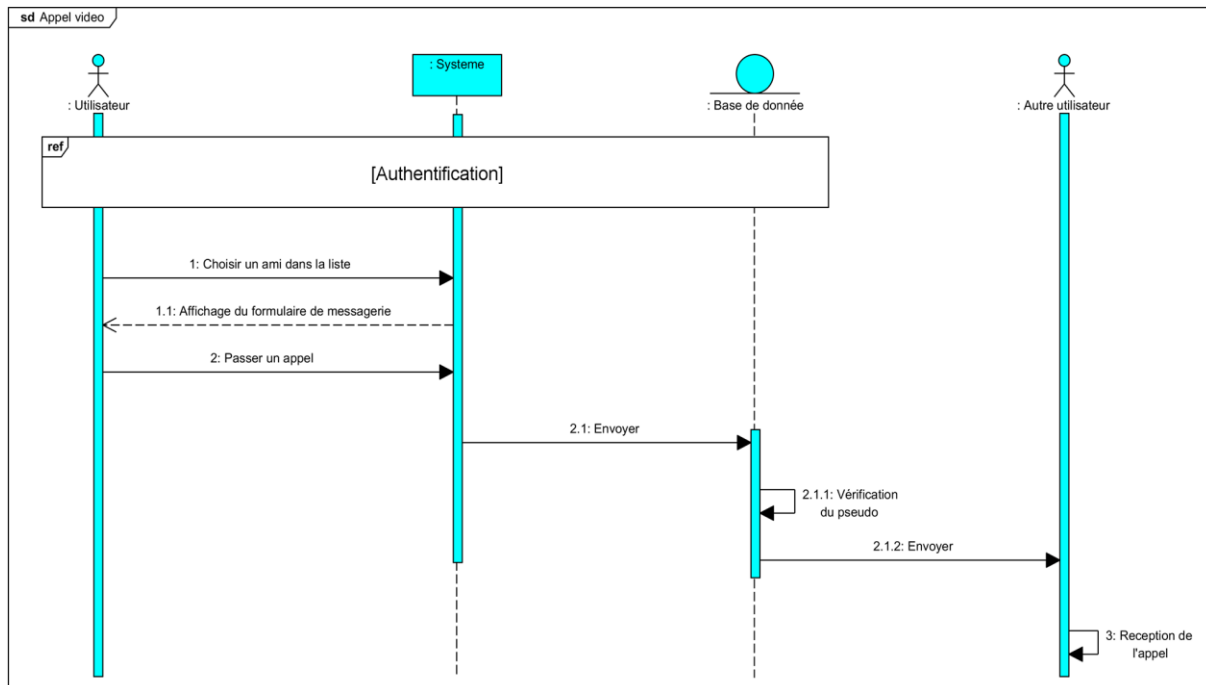


Figure 3.5-Diagramme de séquence de cas d'utilisations «Appel vidéo».

2.6. Cas déconnexion :

Le diagramme de séquence suivant illustre les interactions nécessaires pour déconnecter.

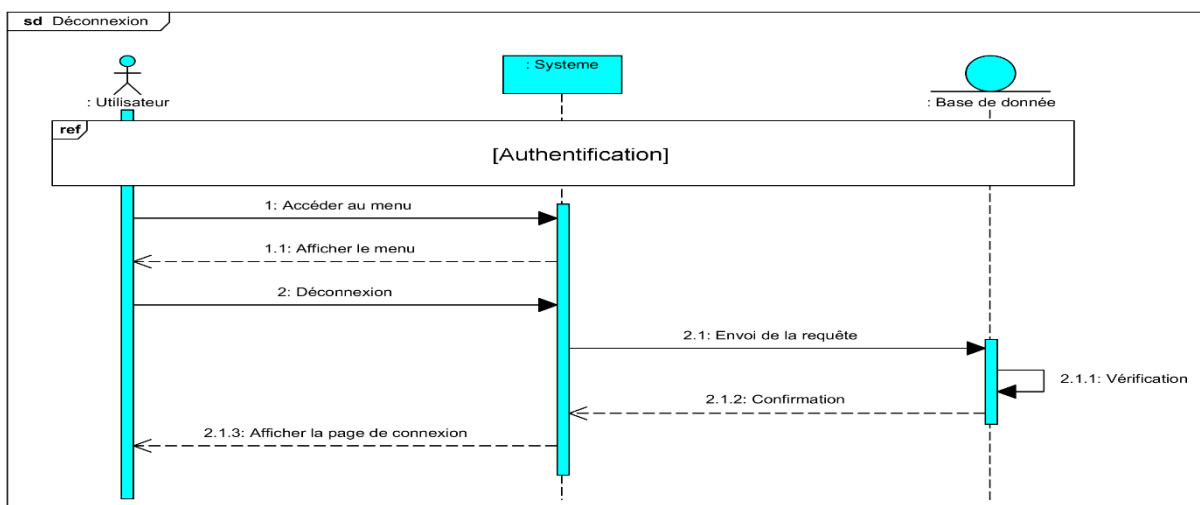


Figure 3.6-Diagramme de séquence de cas d'utilisations «Appel vidéo».

3. Conclusion :

En fin, nous avons cerné les objectifs de notre application. Ces objectifs doivent tenir compte des problèmes de la solution existante. Cette phase va nous être utile pour bien élaborer le modèle de conception de l'application. Dans la prochaine partie nous aborderons la partie conception décrivant la modélisation des besoins exprimés dans cette section.

CHAPITRE IV

Conception

1. Introduction :

Dans la démarche de Processus Unifié, la phase de conception suit immédiatement la phase d'Analyse, par ailleurs la conception de logiciel est un art qui nécessite de l'expérience, et elle consiste à traduire les besoins en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation. En effet, dans ce chapitre nous essayons d'étendre la représentation des diagrammes effectués au niveau de l'analyse en y intégrant les aspects techniques plus proches des préoccupations physiques.

2. Réalisation du diagramme de classe :

Le diagramme de classes est le point central dans un développement orienté objet. En conception, le diagramme de classes représente la structure d'un code orienté.

- **Une classe** : Représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type.
- **Un objet**: Est une entité aux frontières bien définies, possédant une identité et encapsulant un état et un comportement. Un objet est une instance (ou occurrence) d'une classe.
- **Un attribut** : Représente un type d'information contenu dans une classe
- **Une opération**: Représente un élément de comportement (un service) contenu dans une classe.
- **Une association**: Représente une relation sémantique durable entre deux classes.
- **Une superclasse** : Est une classe plus générale reliée à une ou plusieurs autres classes plus spécialisées (**sous-classes**) par une relation de
- **généralisation**. Les sous-classes « Héritent » des propriétés de leur .
- **superclasse** et peuvent comporter des propriétés spécifiques supplémentaires.

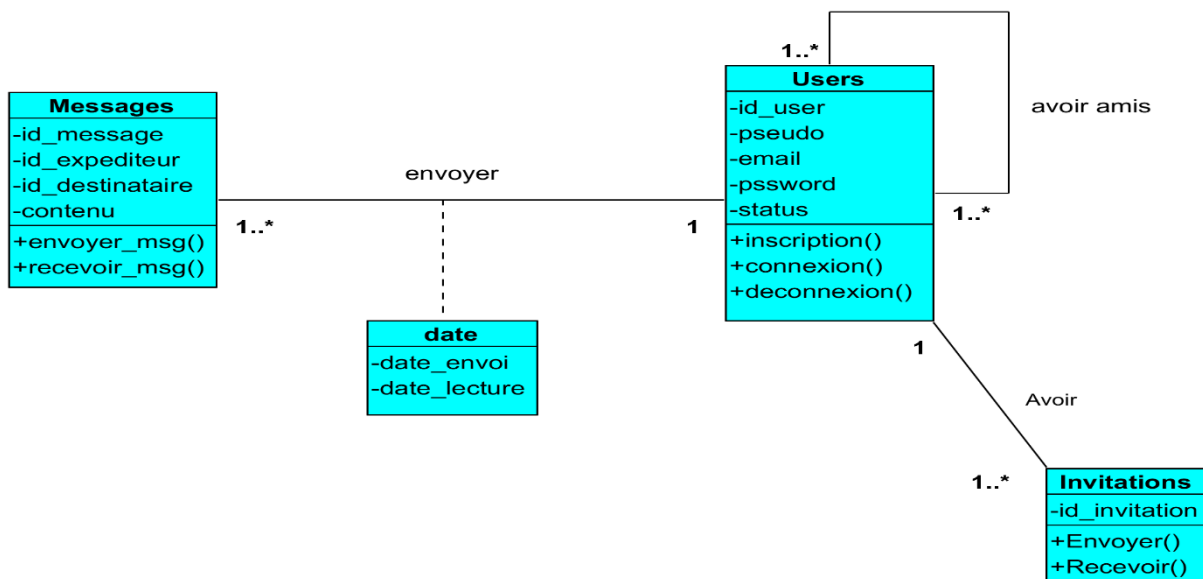


Figure 4.1-Diagramme de classe

3. Le modèle logique de données :

3.1 Règles de passages

Dans ce qui suit, nous allons présenter les différentes règles de passages, qui nous ont servis lors de l'élaboration du modèle logique des données.

- Affecter une table à chaque classe.
- Une association « un à plusieurs » engendre la migration de la clé primaire de la table mère à la table fille.
- Une association « plusieurs à plusieurs » est représentée par une table ayant pour clé primaire la concaténation des clés primaires des deux tables associées.

3.2 Règles de normalisation

Une table est sous la troisième forme normale si, à tout moment, chaque ligne est constituée d'un identificateur d'objet unique associé à un certain nombre d'attributs indépendants.

Après l'application des règles, nous avons dégagé les différentes tables relatives au diagramme de classe, elles sont comme suit:

```
* Users (id_user, pseudo, email, password, statut) ;  
* Amis (id_user1#, id_user2#, statut) ;  
* Messages (id_message, id_user1#, id_user2#, contenu) ;  
* Invitations (id_inv, id_user1#, id_user2#);  
Id_user1= expéditeur, id_user2 = destinataire
```

4. Diagramme de déploiement :

Pour notre cas d'étude, nous utiliserons l'architecture Peer2Peer décentralisée, à vrai dire chaque utilisateur est un peu client, et un peu serveur, autrement dit de serveur central. D'où, lorsqu'un poste envoie un message, joue le rôle d'un client, cependant quand il reçoit un message joue le rôle d'un serveur.

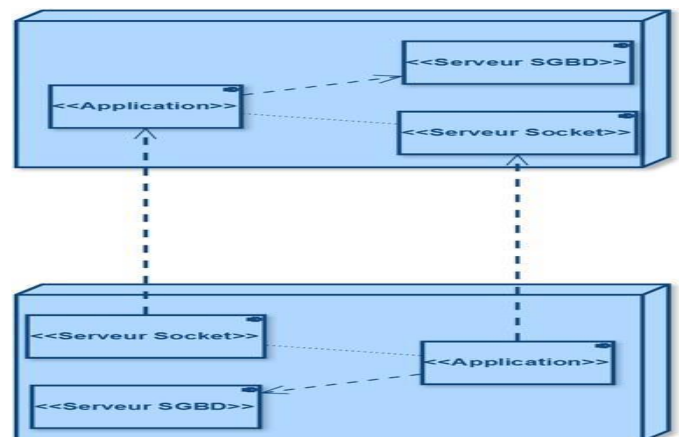


Figure 4.2-Diagramme de déploiement.

5. Conclusion

Dans ce chapitre, nous avons réalisé la conception appropriée à notre application selon les concepts de base de diagramme de classe ainsi que les règles de modélisation. Ensuite, nous avons recensé les règles de passage de diagramme de classe vers le modèle relationnel qui nous permet d'avoir le schéma de la base de données de l'application à réaliser. Cela fait la base pour la phase de réalisation telle qu'on va garantir la fiabilité et l'efficacité.

CHAPITRE V

Réalisation

Introduction

L'étape de réalisation est la dernière de notre projet, elle se présente comme étant la phase la plus cruciale puisque elle traite l'onglet pratique de notre projet

Nous commençons d'abord par une brève illustration de l'environnement de travail ainsi que l'ensemble des logiciels qu'on a utilisé dans la réalisation de l'application dynamique et l'implémentation de base de données, puis nous passons à un aperçu des interfaces les plus importantes de notre application.

Outils et environnement de développement :

➤ Visual Paradigm :

Pour Réaliser les diagrammes UML qui ont servis à modéliser notre application web, nous avons utilisé un logiciel de création de diagrammes Visual Paradigm, qui possède plusieurs options permettant une large possibilité de modélisation en UML, et parmi ses principales fonctionnalités :

- Créations des différents types de schémas comme diagrammes de classe, de cas d'utilisation etc...
- Analyse et manipulation de code sources.



➤ Sublime Text :

C'est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.



➤ **HTML5 (HyperText Markup Language) :**

C'est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.



➤ **CSS3 (Cascading Style Sheet) :**

Feuilles de style en cascade (en anglais, Cascading Style Sheets, "CSS"), est un langage de mise en forme d'un document HTML. Il définit les règles de style et de disposition appliqués aux éléments d'un document html. On utilise le CSS pour modifier le style de n'importe quel élément html pour corriger ses dimensions, couleurs, bordures... etc.



➤ **JavaScript :**

C'est un langage de programmation de scripts principalement employé dans les pages web interactives, il est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.



➤ **PHP (Hypertexte Préprocesseur) :**

PHP (Hypertexte Préprocesseur ou Personal Home Page), est un langage de scripte conçu spécifiquement pour agir sur les serveurs web. Il s'agit d'un langage de programmation, très proche syntaxiquement du langage C, destiné à être intégré dans des pages HTML. Contrairement à d'autres langages, PHP est principalement dédié à la production de pages HTML générées dynamiquement. Il est sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.



➤ **WampServer :**

est une plateforme de développement Web de type WAMP («Windows», «Apache », « MySQL », « PHP »), permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.



➤ **Apache :**

Apache ou bien Apache HTTP Server est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.



Moyens de sécurité :

➤ Le protocole HTTPS :

HTTPS (l'acronyme pour *Hypertext Transfer Protocol Secure*) est la **version sécurisée du HTTP**, le protocole par lequel les données sont échangées entre votre navigateur et le site web auquel vous êtes connectés. Le «S» à la fin de HTTPS signifie «sécurisé». Cela signifie que toutes les communications entre votre navigateur et le site web sont **cryptées**, c'est à dire que les données informatiques ne sont pas lisibles par un humain ou une machine. Le protocole HTTPS est utilisé pour protéger les données de vos utilisateurs et des données sensibles et pour **éviter les intrusions de hackers**.



➤ Le protocole de sécurité SSL

Est le protocole de sécurité le plus répandu qui crée un canal sécurisé entre deux machines communiquant sur Internet ou un réseau interne.

TLS (ou SSL) fonctionne suivant un mode client-serveur. Il permet de satisfaire aux objectifs de sécurité suivants :

- L'authentification du serveur .
- La confidentialité des données échangées (ou session chiffrée) .
- L'intégrité des données échangées .



➤ L'algorithme AES :

AES est un algorithme symétrique de chiffrement par blocs utilisé dans le monde entier sur des supports matériels et logiciels pour protéger les données sensibles.

Le système de chiffrement à clé secrète AES est un système basé sur le système Rijndael construit par Joan Daemen et Vincent Rijmen.

Pour AES les blocs de données en entrée et en sortie sont des blocs de 128 bits, c'est à dire de 16 octets. Les clés secrètes ont au choix suivant la version du système : 128 bits (16 octets), 192 bits (24 octets) ou 256 bits (32 octets).

• Principe de fonctionnement de l'AES :

L'AES procède par blocs de 128 bits, avec une clé de 128 bits également. Chaque bloc subit une séquence de 5 transformations répétées 10 fois, La figure 2.5 présente Schéma fonctionnement de l'algorithme AES.

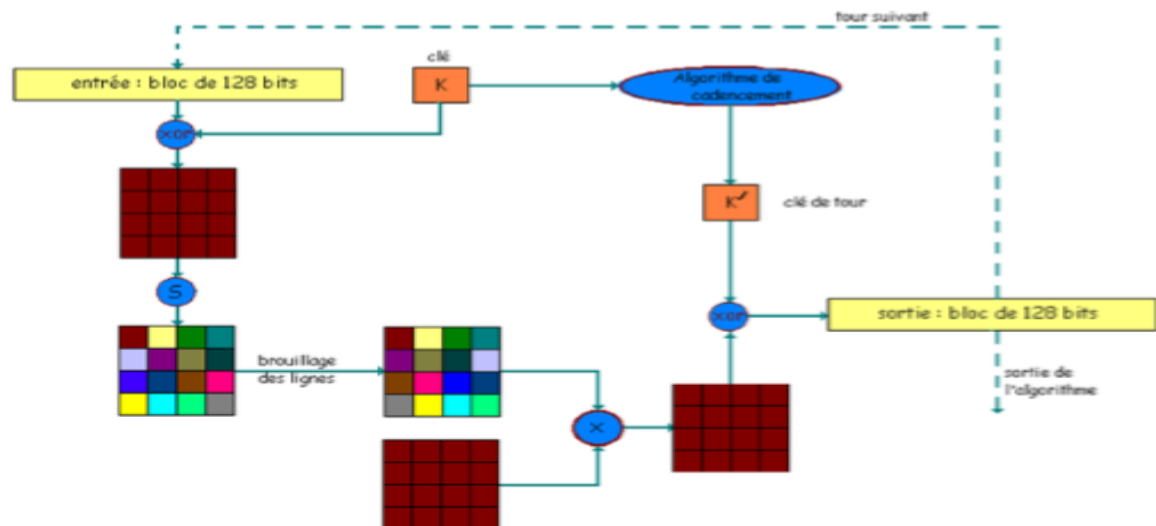


Figure 5.1-fonctionnement de l'algorithme AES

• Les étapes à suivre pour fonctionner l'algorithme :

- Addition de la clé secrète (par un ou exclusif).
- Transformation non linéaire d'octets : les 128 bits sont répartis en 16 blocs de 8 bits (8 bits=un octet), eux-mêmes dispatchés dans un tableau 4x4. Chaque octet est transformé par une fonction non linéaire S. S peut être simplement vue comme une substitution sur les entiers compris entre 1 et 256. En particulier, elle peut être implantée sur ordinateur par un simple tableau.

- Décalage de lignes : les 3 dernières lignes sont décalées cycliquement vers la gauche : la 2ème ligne est décalée d'une colonne, la 3ème ligne de 2 colonnes, et la 4ème ligne de 3 colonnes.
- Brouillage des colonnes : Chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne (ce qui revient à multiplier la matrice 4x4 par une autre matrice 4x4). Les calculs sur les octets de 8 bits sont réalisés dans le corps à 28 éléments.
- Addition de la clé de tour : A chaque tour, une clé de tour est générée à partir de la clé secrète par un sous-algorithme (dit de cadencement). Cette clé de tour est ajoutée par un ou exclusif au dernier bloc obtenu.

- **Caractéristiques et points forts de l'AES :**

Le choix de cet algorithme répond à de nombreux critères plus généraux dont nous pouvons citer les suivants :

- Sécurité ou l'effort nécessaire pour une éventuelle cryptanalyse
- Puissance de calcul qui entraine une grande rapidité de traitement.
- Besoins en ressources et mémoire très faibles
- Flexibilité d'implémentation, cela inclut une grande variété de plateformes et d'applications ainsi que des tailles de clés et de blocs supplémentaires
- Compatibilité hardware et software, il est possible d'implémenter l'AES aussi bien sous forme logicielle que matérielle
- Simplicité, le design de l'AES est relativement simple

Présentation des interfaces de l'application :

➤ Interface d'accueil :

Cette interface affiche à l'utilisateur désirant jouir les fonctionnalités de notre application, s'il dispose d'un compte ,il n'a qu'à cliquer sur se connecter pour connecter ,dans le cas contraire il est invité à s'inscrire .



➤ Le logo :

Le choix de ce logo revient au but de notre application, qui s'agit d'une application web de messagerie sécurisée d'où la sécurité joue le rôle plus important pour ce la le cadenas sur l'enveloppe signifie que les message sont bien sécurisés.



➤ Interface « Authentification » :

Si l'utilisateur est inscrit , cette interface lui permet de s'authentifier afin d'accéder à l'application.

The screenshot shows the login page of the 'Tchat' application. It has a dark blue header with the 'Tchat' logo on the left and 'S'inscrire' and 'Se connecter' buttons on the right. Below the header, the text 'Se connecter' is underlined. The main content area has two input fields: 'Votre adresse email' and 'Votre mot de passe'. At the bottom is a dark blue button labeled 'Se connecter'.

➤ Interface « Inscription » :

Dans le cas où l'utilisateur veut s'inscrire il doit remplir tous les champs suivants correctement et effectuer le test de CAPTCHA.

Un CAPTCHA est une suite de signes un peu déformés qu'on vous demande de recopier quand vous voulez vous enregistrer sur un site ou laisser un commentaire. Le but de l'opération : prouver que vous êtes bien un être humain et pas un programme informatique

Tchat

S'inscrireSe connecter

S'inscrire


Votre nom

Votre adresse email

Votre mot de passe

Confirmez votre mot de passe

☐ Je ne suis pas un robot


reCAPTCHA
Confidentialité - Conditions

S'inscrire

➤ Inscription avec erreur :

Tchat

S'inscrireSe connecter

S'inscrire

Votre nom


Votre adresse email

Votre mot de passe

Confirmez votre mot de passe

Confirmez que vous n'êtes pas un robot

☐ Je ne suis pas un robot


reCAPTCHA
Confidentialité - Conditions

S'inscrire

Tchat

S'inscrireSe connecter


S'inscrire

Votre nom

Votre adresse email

Votre mot de passe

Confirmez votre mot de passe

☐ Je ne suis pas un robot
 

Confidentialité - Conditions

S'inscrire

➤ Inscription bien remplie :

Tchat

S'inscrireSe connecter


S'inscrire

Votre nom
exemple

Votre adresse email
exemple@exemple.com

Votre mot de passe
••••••

Confirmez votre mot de passe
••••••

☒ Je ne suis pas un robot
 

Confidentialité - Conditions

S'inscrire

➤ Interface « Envoi de message » :

Sur cette interface l'utilisateur peut se communiquer avec ces amis en envoyant des messages, les messages sont partagés en temps réel.

Tchat

MembresMes amisMes invitationsDéconnexion

user

exemple

Salut user


Salut a vous !!

Comment est ce que vous allez ?

Je vais très bien merci

et vous ?

Votre message



➤ **Interface « liste des amis » :**

L'utilisateur connecté peut avoir plusieurs amis qui seront affichés comme suit,

Tchat	Membres	Mes amis	Mes invitations	Déconnexion
<div>Mes amis</div> <div> <div>Hakim KEDJAR hakim.kedjar@gmail.com</div> <div>Kamilia MAIZIA kami@kami.com</div> </div>				

➤ **Interface invitation :**

l'utilisateur connecté peut recevoir des invitations, puis la décision lui revient soit il accepte ou il refuse cette invitation en cliquant sur le bouton.

Tchat	Membres	Mes amis	Mes invitations	Déconnexion
<div>Mes invitations</div> <div> <div>Hakim KEDJAR hakim.kedjar@gmail.com</div> <div>Kamilia MAIZIA kami@kami.com</div> <div>Hanane KESSOURI hanane@hanane.com</div> <div>Nazim KEDJAR nazim@nazim.com</div> </div>				

➤ **Interface de stockage de message dans la base de données :**

Dans la base de données on aura tous les utilisateurs inscrits et les messages envoyés ces derniers sont chiffrés avec un mode de chiffrement, d'où l'impossibilité de deviner les messages communiqués.

Serveur: Local Databases

Base de données: tchat

Table: messages

Afficher

Structure

SQL

Rechercher

Insérer

Export

Import

Privileges

Operations

Declencheurs

+ Options

id

sender

receiver

message

date

Modifier

Copier

Effacer

10

hakim.kedjar@gmail.com

kami@kami.com

bWFFdCt0UjJzc3I3aEZzcFJmcIVLZz09OjQDQ52R9EaeihZ+u...

2019-06-04 23:11:26

Modifier

Copier

Effacer

11

exemple@exemple.com

user@user.com

cWV4bnlsUEpkIXVsbVITK3MwQW1UT09Oj1oT5ctxUQJz6JBh...

2019-06-07 20:41:30

Modifier

Copier

Effacer

12

exemple@exemple.com

exemple@exemple.com

UTA4aGVwcTh5dmxoQ2tzSzNEUW5odsz09OjUjZ5wpTsVGZ4MUz...

2019-06-07 20:42:05

Modifier

Copier

Effacer

13

exemple@exemple.com

user@user.com

Qk1TQ0hhNGdLb1BEEdFNReGhJb3BsemIMbU80ZHBrVWNYME9pWD...

2019-06-07 20:42:29

Modifier

Copier

Effacer

14

user@user.com

exemple@exemple.com

TEtxQVQ2YjJ2VElVZDkxZiR5YUJzQWt2a0V2dVpRZmU4YXRkZ...

2019-06-07 20:42:53

Modifier

Copier

Effacer

15

user@user.com

exemple@exemple.com

bjZQemdCMzNubIBPum5RZE9kblNqQT09OjgHFYO18ki0S+kp...

2019-06-07 20:43:05

Modifier

Copier

Effacer

16

kami@kami.com

boualem@boualem.com

S0VGnzJ1Q09vdHNxNEhBYVhBTEZlZz09OjQD9WC40FRhO0Ti...

2019-06-08 00:14:07

Modifier

Copier

Effacer

17

boualem@boualem.com

kami@kami.com

VjJmWRE9CRzVuK1Z5d01m0ZyY0tUT09OjRkys0TZE909YnOF...

2019-06-08 00:14:27

Modifier

Copier

Effacer

18

kami@kami.com

boualem@boualem.com

V1ltbGhBUzBPcm9WYDY1N3FhQ29FQT09OjYoaXN98e73l2YG/...

2019-06-08 00:14:38

Modifier

Copier

Effacer

19

boualem@boualem.com

kami@kami.com

WFFvOFRsbVldURjRnBqdm5SQWlkZYZEY0K3d3UGxQdXgNGxCOG...

2019-06-08 00:15:00

Modifier

Copier

Effacer

20

boualem@boualem.com

kami@kami.com

OENFUE5PZnJQcW10E3qL2NOUTILemVqN0dpdmJIRVcrb0lwK0...

2019-06-08 00:15:16

Modifier

Copier

Effacer

21

kami@kami.com

boualem@boualem.com

QUlaZG9wRk0yVDQyNlVUOG44akFFZnBUNFBRScjYXZ4VkpJeE...

2019-06-08 00:15:40

Tout cocher

Pour la selection :

Modifier

Copier

Effacer

Export

Tout afficher

Nombre de lignes : 25

Filtrer les lignes: Chercher dans cette tabl

Trier sur l'index: Aucune

Console de requêtes SQL

Utilité de la requête

Conclusion

Dans ce chapitre, nous avons présenté l'environnement de développement matériel et logiciel avec lesquels ce projet a été réalisé. Nous l'avons conclu par la suite avec quelques interfaces de notre application.

Conclusion générale :

L'objectif de notre projet , présenté dans ce rapport, est la conception et la réalisation d'une application web de messagerie simple afin de gérer la sécurité et garanti la confidentialité entre les utilisateurs. Pour réaliser cette application, le choix s'est posé sur le processus de développement UP et le formalisme UML en utilisant un ensemble de diagrammes qui nous a permis une modélisation d'une manière claire du comportement du système. Par la suite, nous avons mis en œuvre une base de donnée relationnelle. Pour interagir avec cette dernière. Le développement de notre application demande beaucoup de compétences dans divers domaines dans lesquels nous étions limitées, c'est pourquoi nous avons dû approfondir nos connaissances dans les différents langages et outils auxquels nous avons eu recours: à savoir PHP, UML, WAMPserver, MySQL , ArgoUML .Dans un premier temps, nous avons défini et analysé les besoins de système .Ensuite ,nous avons entamé la phase de conception qui a permis de structurer et définir les besoins attendus du système . Enfin, nous avons abordé la réalisation en utilisant les outils d'implémentation appropriés, Nous espérons que ce travail facilitera et permettra une meilleure communication entre les utilisateurs.