**Système d'Envoi de Messages avec Signature Numerique**

**1. Introduction**

**Contexte du Travail**

Le besoin croissant de sécuriser les communications électroniques a motivé la conception et la mise en œuvre d'un système d'envoi de messages avec signature numérique. Ce système vise à garantir l'authenticité et l'intégrité des messages échangés entre les utilisateurs.

**Objectif**

L'objectif principal de ce projet est de développer une plateforme robuste et sécurisée permettant aux utilisateurs d'envoyer des messages signés numériquement, assurant ainsi la confidentialité et la non-répudiation des communications.

**Méthodologie**

Pour atteindre cet objectif, nous avons suivi une approche itérative de développement logiciel basée sur les principes de l'ingénierie logicielle. Nous avons utilisé des méthodes agiles pour assurer une évolution continue du système en réponse aux besoins des utilisateurs.

**Technologies Utilisées**

* Environnement de Développement : Visual Studio Code
* Langage de Programmation : Python
* Framework Web : Flask
* Interfaces Utilisateur : HTML, CSS

**2. Théorie sur l'Algorithme de Chiffrement RSA**

1. **Historique de l’algorithme**

L'algorithme de chiffrement RSA tire son nom des initiales de ses inventeurs : Ron Rivest, Adi Shamir et Leonard Adleman. Développé en 1977, cet algorithme révolutionnaire a marqué une avancée majeure dans le domaine de la cryptographie asymétrique. En effet, il a introduit la notion de clés publique et privée, permettant ainsi d'établir des communications sécurisées sur des réseaux non sécurisés.

L'idée fondamentale derrière l'algorithme RSA repose sur la difficulté mathématique de factoriser de grands nombres premiers, un problème considéré comme complexe à résoudre, même pour les ordinateurs les plus puissants. Cette caractéristique fait de l'algorithme RSA l'un des systèmes de chiffrement les plus fiables et largement utilisés dans le domaine de la sécurité informatique.

1. **Utilité de l’algorithme RSA**

L'algorithme RSA est largement utilisé dans de nombreux domaines pour assurer la confidentialité, l'authenticité et l'intégrité des données. Voici quelques-unes de ses principales utilisations :

1. **Chiffrement des Communications** : L'algorithme RSA est utilisé pour chiffrer les communications électroniques, telles que les e-mails, les transactions en ligne et les messages instantanés. En utilisant la clé publique du destinataire, l'expéditeur peut chiffrer le message de manière sécurisée.
2. **Signature Numérique** : L'algorithme RSA est également utilisé pour créer des signatures numériques, permettant de garantir l'authenticité et l'intégrité des données. En signant un message avec sa clé privée, l'expéditeur peut prouver son identité et empêcher la falsification des données. Tel est le cas utilisé ce présent travail.
3. **Authentification** : Dans les protocoles d'authentification, l'algorithme RSA est utilisé pour établir des preuves d'identité sécurisées. Les certificats numériques, basés sur RSA, sont largement utilisés pour vérifier l'identité des sites web et des utilisateurs.
4. **Échanges de Clés** : L'algorithme RSA est également utilisé dans les protocoles d'échange de clés, tels que le protocole SSL/TLS, pour sécuriser les connexions réseau. En utilisant RSA pour échanger des clés de session, les communications peuvent être protégées contre les attaques de type "homme du milieu".
5. **Fonctionnement de l’algorithme**

L'algorithme de chiffrement RSA repose sur des concepts mathématiques complexes mais fondamentaux. Voici comment fonctionne la logique de l'algorithme de chiffrement RSA :

1. **Génération des Clés** :
   * Deux nombres premiers distincts, p et q, sont choisis aléatoirement.
   * Le produit n = p \* q est calculé. Ceci est utilisé comme le module de chiffrement.
   * La fonction d'Euler *ϕ*(*n*)=(*p*−1)(*q*−1) est calculée.
2. **Choix de l'Exposant de Chiffrement** :
   * Un exposant de chiffrement e est choisi tel que 1 < e < *ϕ*(*n*) et que e soit premier avec *ϕ*(*n*).
   * Habituellement, e est choisi comme un petit nombre premier, souvent 65537 (0x10001 en hexadécimal).
3. **Calcul de l'Exposant de Déchiffrement** :
   * L'exposant de déchiffrement d est calculé en trouvant l'inverse modulaire de e modulo *ϕ*(*n*).
   * Cela signifie que d est l'entier tel que  *e*∗*d*≡1mod*ϕ*(*n*).
4. **Chiffrement d'un Message** :
   * Pour chiffrer un message M, on calcule *C*=mod *n*.
   * Le message chiffré C est transmis au destinataire
5. **Déchiffrement du Message Chiffré** :
   * Pour déchiffrer le message chiffré C, le destinataire calcule *M*=mod *n*.
   * Le message original M est ainsi retrouvé.

La sécurité de l'algorithme RSA repose sur la difficulté de factoriser le grand nombre n en ses facteurs premiers p et q. La connaissance de p et q permettrait de calculer *ϕ*(*n*) et ainsi de déterminer l'exposant de déchiffrement d. Sans cette connaissance, il est extrêmement difficile de casser le chiffrement RSA.

En résumé, l'algorithme de chiffrement RSA joue un rôle crucial dans la sécurisation des communications et des transactions numériques. Sa robustesse mathématique et sa polyvalence en font un outil indispensable dans le domaine de la cryptographie moderne.

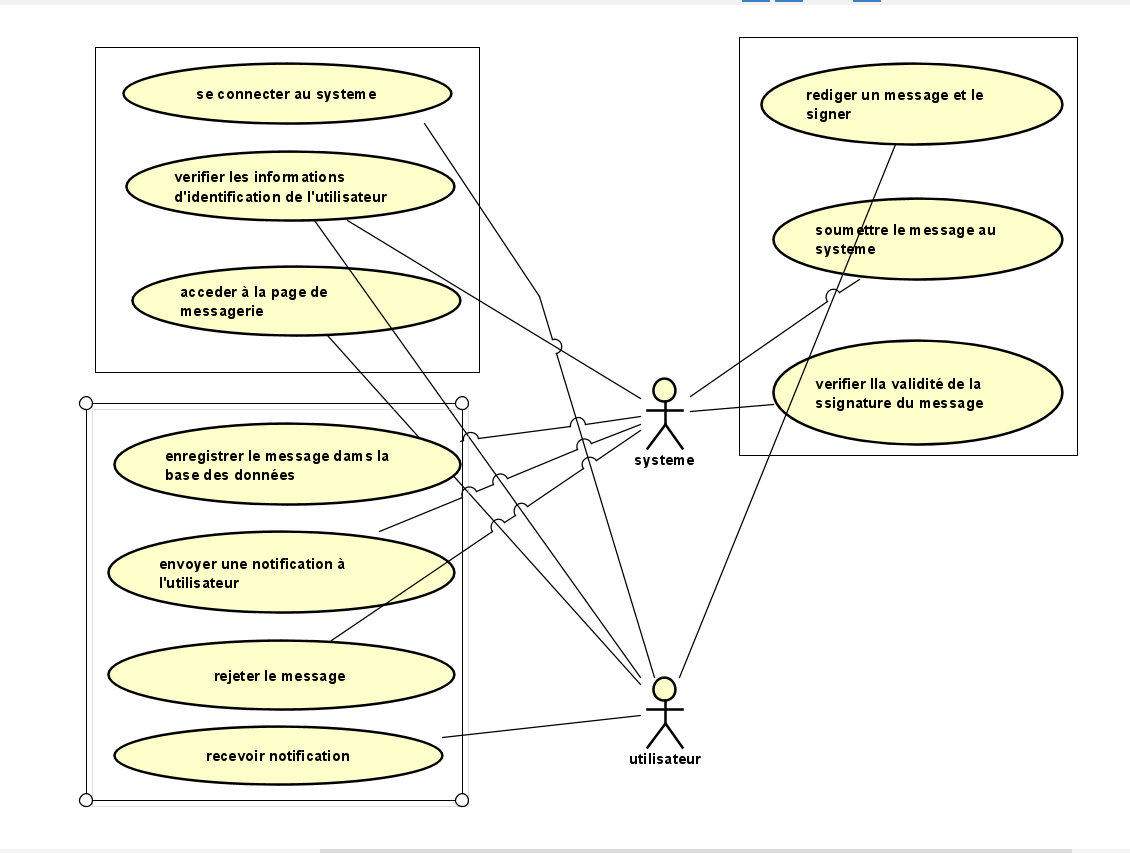
Cette théorie élargie sur l'algorithme de chiffrement RSA met en lumière son importance historique, son utilité pratique, son fonctionnement et ses applications variées dans le domaine de la sécurité informatique.

**3. SYSTEME**

1. Diagramme de classe

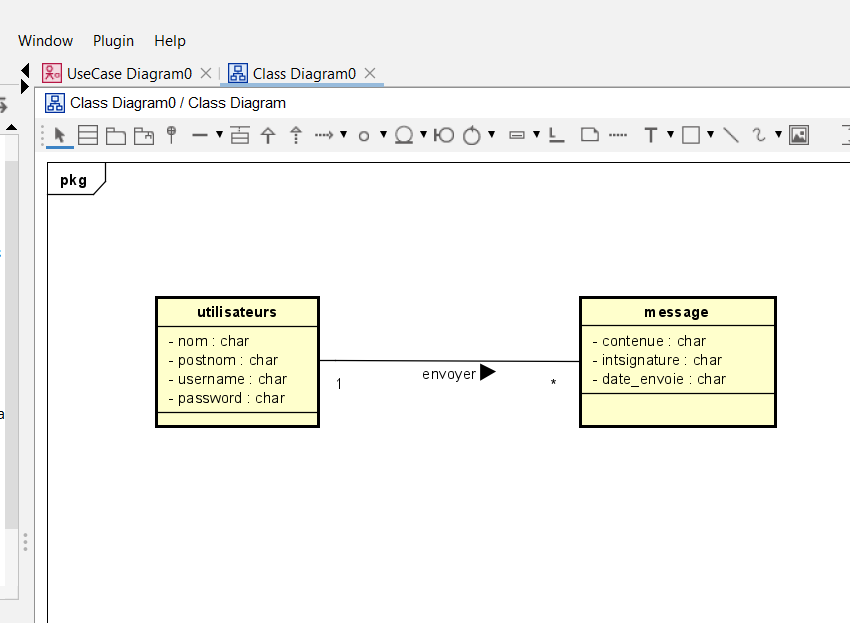
Le diagramme de classe de notre système est composé de deux classes qui sont les deux tables présentes dans le système. Il est illustré sur l’image suivant :

Fugure1



1. Le diagramme de classe

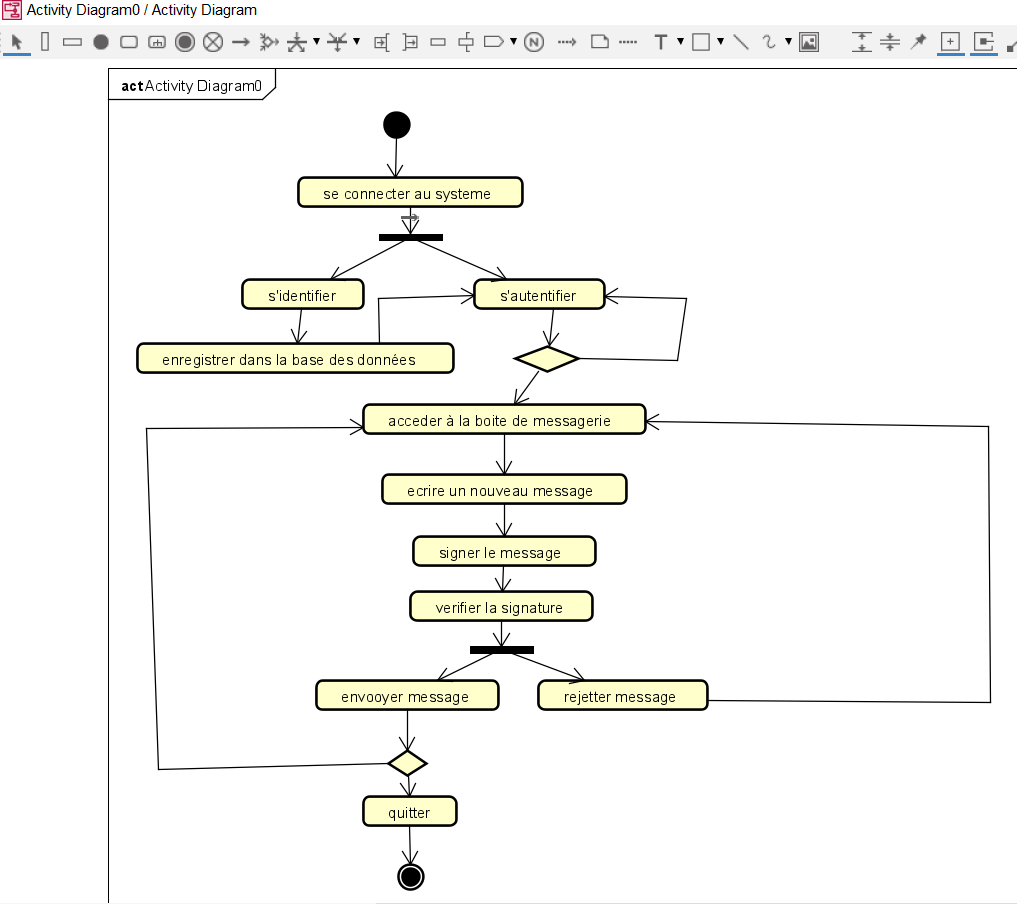
Le diagramme de classe de notre système se présente de la manière suivante :

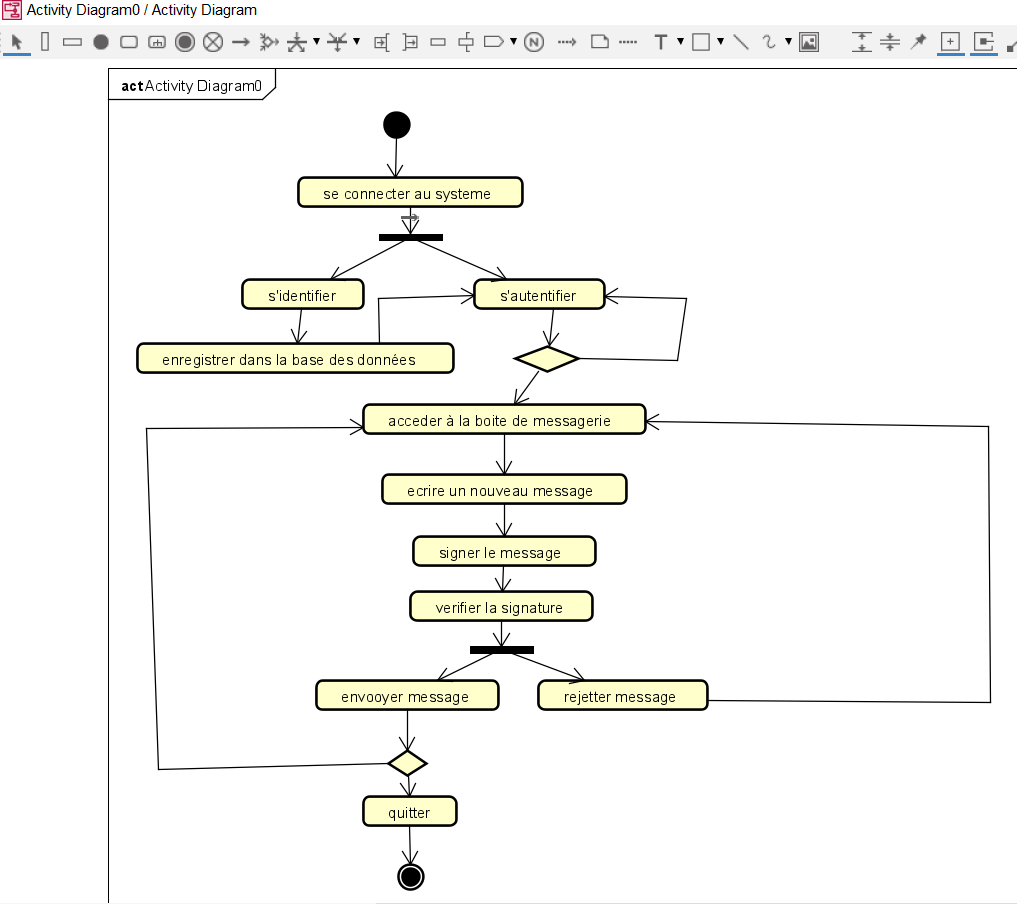
Figure2

1. Diagramme d’activité

Le diagramme d’activité résume les différentes activités du faites dans le système. Notre diagramme d’activité se présente comme suit :

Figure3





**4. Conclusion.**

Ce projet d'ingénierie développé ci-haut vise à mettre en place un système d'envoi de messages sécurisés avec signature numérique, en utilisant les technologies Python, Flask et HTML. L’utilisation de l'algorithme de chiffrement RSA nous a servi pour signer les messages lors de l’envoie. L’usage de cet algorithme garantie la confidentialité et l'authenticité des messages échangés. Dans les lignes précédentes, nous avons abordé certains points dont premièrement une traduction contenant le contexte du travail, l’objectif, la méthodologie ainsi que les technologies utilisées. Deuxièmement nous avons énoncé la théorie du chiffrement RSA qui comporte son historique, son utilité, et le fonctionnement de l’algorithme. Troisièmement nous avons présenté la modélisation du système en trois diagrammes qui : le diagramme de cas d’utilisation, le diagramme de classe et le diagramme d’activité. Ces points soulignés sont ceux qui nous ont permis d’établir et présenter ce présent travail.