République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira-Bejaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de cycle
En vue d'obtenir le diplôme de master professionnel en Informatique
Administration et Sécurité des Réseaux

Thème

Conception et Réalisation d'une Application Web pour la Délivrance et l'Authentification des Documents Académiques et Administratifs

«SIgDoc »

Réalisé par :

BOUKHIMA Lyes

MANSOURI Mohand

Devant le jury composé de :

Président : M^me^ TAHAKOURTHE Zineb (MAA)

Examinateur: Mr SEBAA Abdrezak (MAA)

Encadreur: M^me^ BOUTRID Samia (MAA)

Promotion : 2015-2016

Remerciements

Nous remercions en premier lieu Dieu tout puissant qui nous a dotés d'une grande volonté et d'un savoir adéquat pour mener à terme notre projet.

La rédaction de ce mémoire fut un exercice quotidien stimulant qui n'aurait pu se concrétiser sans l'apport déterminant et apprécié de plusieurs personnes.

Notre profonde gratitude et sincères remerciements à notre encadreur, en l'occurrence **Mme BOUTRID Samia** *pour avoir accepté de nous encadrer et pour l'intérêt qu'elle a porté à notre travail, son suivi, sa disponibilité et ses conseils et orientations.*

Aux membres de la commission qui jugeront notre travail à savoir :

Mme TAHAKOURT Zineb *pour nous avoir fait l'honneur de présider l'honorable jury.*

Mr SEBAA Abdrezak *pour avoir été parmi les membres du jury et d'avoir examiné notre travail.*

Nos remerciements sont adressés également à nos chers parents pour tous les sacrifices consentis à notre égard et leur énorme soutien.

A tous nos proches amis (e)

A tous nos enseignants et membres du département informatique de l'université **ABDERRAHMANE MIRA.**

Dédicaces

Ce modeste travail est dédié :
A nos chers parents qui nous ont soutenus et encouragés durant toute notre scolarité.
A nos frères et sœurs
A nos enseignants
A nos amis(e)
A toutes les personnes qui nous ont apportés de l'aide.

Table de matières.

Chapitre I : Concepts de base sur la sécurité et la cryptographie.

Introduction

Chapitre II : Analyse & conception.

Introduction

Chapitre III : Réalisation.

Liste des figures

Liste des tableaux

Liste des abréviations

SIG DOC: Signature Documents.

DOS: Denial-of-service.

IP: Internet Protocol.

DES: Data Encryption Standard.

AES: Advanced Encryption Standard.

RSA: R.Rivest A.Shamir et L.Adelman.

AC: Autorité de Certification.

UML: Unifier Modeling Language.

UP: Unifier Process.

QR: Quick Reponse.

BDD: Base de données.

HTML: Hypertext Markup Language.

PHP: Hypertext Preprocessor.

MYSQL: My Structured Query Language.

CSS: Cascading Style Sheets.

HTTP: Hypertext Tranfer Protocol.

Introduction Générale

L'un des principaux avantages de la cryptographie à clé publique est qu'elle offre une méthode d'utilisation des signatures numériques.
La signature numérique est un mécanisme qui permet d'authentifier un message, autrement dit, de prouver qu'un message provient bien d'un expéditeur donné, à l'instar d'une signature sur un document papier.
Une signature numérique a la même utilité qu'une signature manuscrite. Cependant, une signature manuscrite peut être facilement imitée, alors qu'une signature numérique est pratiquement infalsifiable. De plus, elle atteste du contenu des informations, ainsi que de l'identification du signataire.

Vu que la falsification des documents administratifs se montre à la portée de tous ceux qui s'y intéressent, alors, la nécessité de mettre en pratique un moyen de satisfaire une authentification et chiffrement des documents numériquement devient primordial. L'objectif de notre travail, consiste alors à concevoir et à réaliser une application web dynamique pour la délivrance et l'authentification des documents académiques et administratifs. En effet, cette application à mettre en œuvre facilitera, et offrira une meilleure gestion et garantira une confidentialité et la non répudiation de manière très significative pour les secteurs administratifs.

Ce travail est organisé en trois chapitres :

Le premier est dédié à la description des généralités sur la sécurité informatique et la cryptographie, il a pour objectifs de définir les concepts essentiels sur lesquels est basé notre travail.

Le deuxième s'intitule Analyse & conception, où nous ferons une étude préliminaire et une analyse des besoins du système à développer qui est le noyau de notre travail. Nous allons d'abord recenser les acteurs qui interagissent avec notre application, puis nous décrirons les besoins de chaque acteur sous forme de cas d'utilisation. Enfin, pour chaque cas d'utilisation, nous allons établir le diagramme de séquence associé. Afin de concevoir le projet, nous présentons le diagramme de classe associé à notre système et le modèle logique de données obtenu par l'application des règles de passage.

La réalisation et l'implémentation fera l'objet du troisième chapitre dans lequel nous définirons les outils de développement à utiliser. Nous illustrerons également quelques interfaces de l'application à développer.

Enfin, nous conclurons ce travail en résumant les connaissances acquises durant la réalisation du projet.

Chapitre I

Concepts de base sur la Sécurité et la cryptographie

La sécurité informatique

La cryptographie

Introduction

Dans la première section de ce chapitre, nous présentons tout ce qui concerne la sécurité informatique, sa définition, ses services, ses menaces et ses solutions.

La sécurité des systèmes informatiques combine des solutions techniques, organisationnelles, juridiques et humaines dans le but de conserver, garantir et de rétablir la sécurité du système informatique.

Tandis que dans la deuxième section, nous présentons tout ce qui concerne la cryptographie, l'écriture secrète qui utilise les mathématiques pour chiffrer et déchiffrer les données.

1. La sécurité informatique

La sécurité informatique est de nos jours devenue un problème majeur dans la gestion des réseaux d'entreprise, ainsi que pour les particuliers toujours plus nombreux à se connecter à Internet. La transmission d'informations sensibles et le désir d'assurer la confidentialité de celles-ci est devenue un point primordial dans la mise en place des réseaux informatiques.

1.1 Définition de la sécurité

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. Elle s'occupe de la prévention d'actions non autorisées par les utilisateurs d'un système informatique.

1.2 Les services de la sécurité

Pour remédier aux failles et pour contrer les attaques, la sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace. Et parmi ces services [1], [2], [3], [5] :

1.2.1 La confidentialité

La confidentialité est le premier problème posé à la sécurité. C'est la propriété montrant qu'une information n'est ni disponible ni divulguée aux personnes, entités ou processus non autorisés.

1.2.2 L'authentification

L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple l'utilisation d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

1.2.3 L'intégrité

Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).

1.2.4 La non-répudiation

La non-répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction. En d'autre terme, c'est empêcher le démenti d'engagement sur une action. C'est l'une des composantes les plus importantes dans le commerce sur Internet.

1.2.5 La disponibilité

L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources.

1.3 Les menaces

Actuellement, le problème consiste à définir correctement les risques engendrés par la criminalité informatique. Il faut pour cela avoir une vision globale du problème et connaitre globalement les techniques utilisées par les nouveaux flibustiers. Il s'agira ensuite d'analyser correctement les vulnérabilités propres à chaque site, de définir le niveau de sécurité requis et enfin de mettre en place une politique de sécurité acceptable. [4]

Les menaces sont de plusieurs types, elles peuvent être subdivisées en deux types :

1.3.1 Les menaces non intentionnelles

Ce type de menace peut être d'origine humaine ou naturelle. Les personnes mal ou peu formées aux outils qu'elles utilisent, les catastrophes naturelle.

Elles peuvent être aussi causées par des pannes du système informatique (disfonctionnement matériel ou logiciel), Des erreurs de manipulation de système, erreur de manipulation d'information, ou erreur de conception d'application.

On cite quelques exemples ci-dessus :

- **Défaillance matérielle :** Tout équipement physique est sujet à défaillance (usure, vieillissement, défaut…) L'achat d'équipements de qualité et standard accompagnés d'une bonne garantie avec support technique est essentiel pour minimiser les délais de remise en fonction. Seule une forme de sauvegarde peut cependant protéger les données.

- **Défaillance logicielle :** Tout programme informatique contient des bugs. La seule façon de se protéger efficacement contre ceux-ci est de faire des copies de l'information à risque. Une mise à jour régulière des logiciels et la visite des sites consacrés à ce type de problèmes peuvent contribuer à en diminuer la fréquence.

- **Accidents (pannes, incendies, inondations…) :** Une sauvegarde est indispensable pour protéger efficacement les données contre ces problèmes.

1.3.2 Les menaces intentionnelles

C'est l'ensemble des actions malveillantes qui constituent la plus grande partie de risques. Elles font l'objet principal des mesures de protection. A titre d'exemple :

- **Virus provenant de support de stockage :** Ce risque peut-être réduit en limitant le nombre de lecteur de service. L'installation de programmes antivirus peut s'avérer une protection efficace mais elle est coûteuse, diminue la productivité, et nécessite de fréquentes mises à jour.

- **Piratage et virus réseau :** Cette problématique est plus complexe et l'omniprésence des réseaux, notamment l'Internet, lui confère une importance particulière.

1.4 Les attaques

On peut classifier les attaques en deux groupes principaux : les attaques passives et les attaques actives, qui sont bien évidemment plus dangereuses [5].

1.4.1 Les attaques passives

Consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables, mais une prévention est possible.

1.4.2 Les attaques actives

Consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

1.4.3 Quelques techniques d'attaque

- **Les Denial-of-Service (Dos)** : Les attaques de type Denial-of-Service ont pour but de saturer un routeur ou un serveur afin de le "crasher" ou en préambule d'une attaque massive. Ces types d'attaques sont très faciles à mettre en place et très difficile à empêcher [6].

- **L'IP Spoofing :** La technique de l'IP Spoofing est une technique dont le principe est relativement ancien (aux alentours de 1985) mais la première attaque connue l'utilisant ne remonte qu'à 1995. Kevin Mitnick, un célèbre "Hacker", l'a utilisé afin de s'infilter dans le réseau d'un expert en sécurité informatique, Tsutomo Shimomura. Le Spoofing n'est pas l'attaque en tant que tel, il s'agit d'une technique permettant de s'infiltrer dans un ordinateur en se faisant passer pour un autre en qui il a confiance (Trusted Host) [6].

- **Les Backdoors :** Depuis que les intrusions informatiques existent, leurs adeptes ont mis au point un certain nombre de techniques leur facilitant l'accès aux systèmes pénétrés. La technique la plus connue, et sans doute la plus utilisée, est celle des Backdoors (portes dérobées ou portes de service). Elles permettent, à celui qui en connait l'existence et le fonctionnement, de revenir sur un système de façon détournée, c'est-à-dire sans passer par les méthodes d'authentifications habituelles. Il existe déférents types de Backdoors, certaines n'ont une utilité qu'une fois l'accès à la station accordé, d'autres permettent par exemple de contourner les différents types de Firewalls [6].

1.5 Les solutions

Les solutions proposées s'articulent entre deux axes majeurs : la prévention et la détection [1].

1.5.1 La prévention

La prévention consiste à écarter le risque d'attaque en implémentant un ensemble de mécanismes de protection contre la manipulation illicite des informations, afin de garder le réseau fonctionnel le plus longtemps possible. Néanmoins, ces informations possèdent des rôles et des natures différentes, nécessitant chacune un mécanisme de protection adéquat.

1.5.2 La détection

La détection représente le mécanisme central de chaque protocole de sécurité. Il définit la procédure curative lors de la violation de l'une des règles établit par le mécanisme de protection.

1.5.3 Les mécanismes de sécurité

Pour répondre aux besoins de la sécurité, le système informatique doit implanter des mécanismes, parmi eux :

- Les accréditations de l'authentification.
- Contrôle d'accès.
- Certificat numérique.
- Cryptographie.

2. La cryptographie

La cryptographie est maintenant devenue une discipline scientifique à part entière dont les applications sont si vastes aujourd'hui qu'il est difficile de définir a priori ce qui en relève. Dans cette partie nous aborderons les aspects techniques et applicatifs de la cryptographie.

2.1 Définition

La cryptographie est la science qui utilise les mathématiques pour chiffrer et déchiffrer des données. La cryptographie vous permet de stocker des informations sensibles ou de les transmettre à travers des réseaux non sûrs (comme Internet) de telle sorte qu'elles ne puissent être lues par personne à l'exception du destinataire convenu. Alors que la cryptographie est la science de la sécurisation des données, la cryptanalyse est la science de l'analyse et du cassage des communications sécurisées. La cryptanalyse classique mêle une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, de patience, de détermination, et de chance. Les cryptanalystes sont aussi appelés attaquants. La cryptologie embrasse à la fois la cryptographie et la cryptanalyse .

2.2 Le fonctionnement de la cryptographie

Un algorithme cryptographique, ou chiffre, est une fonction mathématique utilisée dans le processus de chiffrement et de déchiffrement. Un algorithme cryptographique fonctionne en combinaison avec une clé – un mot, un nombre, ou une phrase – pour chiffrer le texte clair. Le même texte clair se chiffre en un texte chiffré différent si l'on utilise des clés différentes. La sécurité des données chiffrées est entièrement dépendante de deux choses: la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constitue un cryptosystème.

2.3 Chiffrement et déchiffrement

Les données qui peuvent être lues et comprises sans mesures spéciales sont appelées texte clair. Le procédé qui consiste à dissimuler du texte clair de façon à cacher sa substance est appelée chiffrement (dans le langage courant on parle plutôt de cryptage et de ses dérivés: crypter, décrypter).

On utilise le chiffrement pour garantir que l'information est cachée à quiconque, elle n'est pas destinée, même ceux qui peuvent lire les données chiffrées. Le processus de retour du texte chiffré à son texte clair original est appelé déchiffrement.

Figure 1 : chiffrement et déchiffrement

2.4 Les types de la cryptographie

Il existe deux grandes familles d'algorithmes cryptographiques : la cryptographie symétrique et asymétrique [7].

2.4.1 La cryptographie symétrique

Le chiffrement symétrique consiste à utiliser la même clé pour le chiffrement que pour le déchiffrement .Il est donc nécessaire que les deux interlocuteurs se soient mis d'accord Sur une clé privée, où ils doivent utiliser un canal sécurisé pour l'échanger.

Il existe deux catégories de systèmes symétriques : les chiffrements par blocs et les chiffrements de flux.

Chiffrement par bloc

Dans ce mode de chiffrement, le texte clair est tout d'abord séparé en blocs de tailles fixes puis l'algorithme se charge de chiffrer un bloc à la fois. Ensuite les blocs résultants seront fusionnés pour former le texte chiffré final. La taille des blocs a un impact sur la sécurité et sur la complexité de l'algorithme. Les blocs de grandes dimensions sont plus sécuritaires mais sont plus lourds à implémenter.

Chiffrement par flux

Les algorithmes de chiffrements de flux peuvent être définis comme étant des algorithmes de chiffrement par blocs, ou le bloc à une taille unitaire (1 bit, 1 octet, etc.). Leurs avantages viennent du fait que le chiffrement peut varier à chaque symbole du texte et du fait qu'ils soient extrêmement rapides.

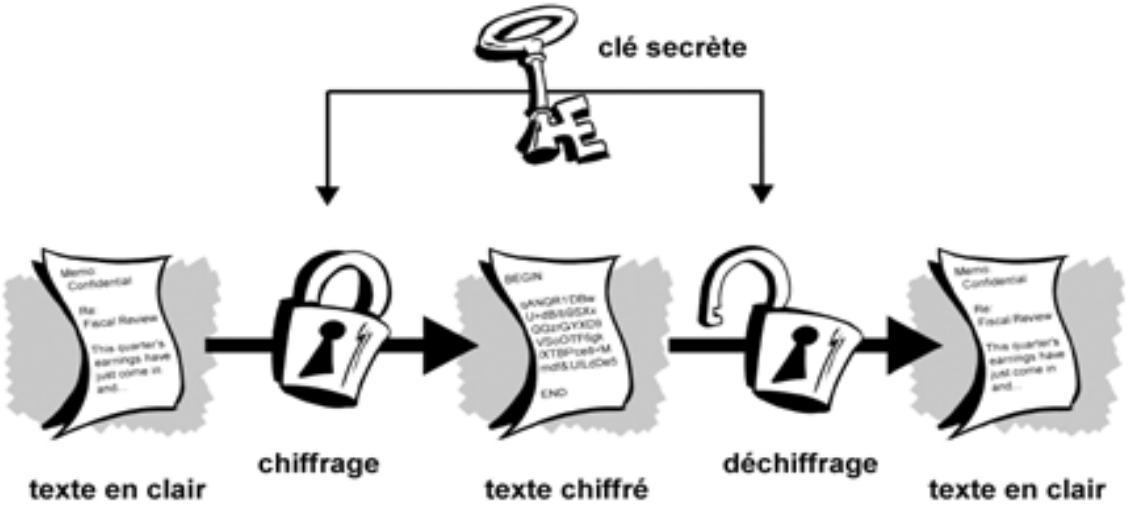


Figure 2 : Le schéma général de la cryptographie symétrique

2.4.2 Les méthodes symétriques les plus utilisées

Le DES (Data Encryption Standard)

DES est un algorithme agissant par blocs. Le texte en clair est découpé en blocs de 64 bits, où chaque bloc est chiffré séparément et donne un bloc chiffré de 64 bits. Ensuite les blocs résultants seront fusionnés pour former le texte chiffré final. Il consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée code produit [8].

L'AES (Advanced Encryption Standard)

Grâce à son niveau élevé de sécurité, sa performance et son efficacité, l'algorithme RHINDAEL a été choisi parmi plusieurs autres algorithmes. L'AES est un algorithme symétrique avec une clé de 128 bits, 192 bits ou 256 bits (généralement 128 bits). L'AES opère sur des blocs de 128 bits qu'il transforme en blocs cryptés de 128 bits par une séquence d'opérations ou « rounds », à partir d'une clé de 128, 192 ou 256 bits. Suivant la taille de celle-ci, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds.

2.4.3 Les problèmes de la cryptographie symétrique

Le principal problème posé par l'utilisation de la cryptographie symétrique est l'échange de La clé privée, comment faire parvenir la clé à son destinataire sans qu'une personne ne l'intercepte ? La méthode la plus sûre c'est que l'émetteur et le récepteur doivent se mettre d'accord à l'avance sur la clé à utiliser, autre méthode moins sûre est d'utiliser un canal sécurisé pour la transmission de la clé et un réseau public pour la transmission du texte chiffré.

Autre inconvénient est que chaque entité doit disposer d'autant de clés secrètes qu'elle a d'interlocuteurs (problème de gestion de clés). Ce type de chiffrement n'est pas pratique dans le cas d'Internet où les entités communicantes ne se connaissent pas. Les problèmes de l'échange et de la distribution de la clé sont restés sans solution jusqu'à l'invention de la cryptographie asymétrique [9].

2.4.4 La cryptographie asymétrique

Pour résoudre le problème de l'échange de la clé secrète, un nouveau type de cryptographie a été inventé, c'est la cryptographie asymétrique. La cryptographie asymétrique désigne une méthode cryptographique faisant intervenir une paire de clés asymétriques : une clé publique et une clé privée. Elle utilise cette paire de clés pour le chiffrement et le déchiffrement. La clé publique est rendue publique et distribuée librement, la Clé privée n'est jamais distribuée, et doit être gardée secrète.

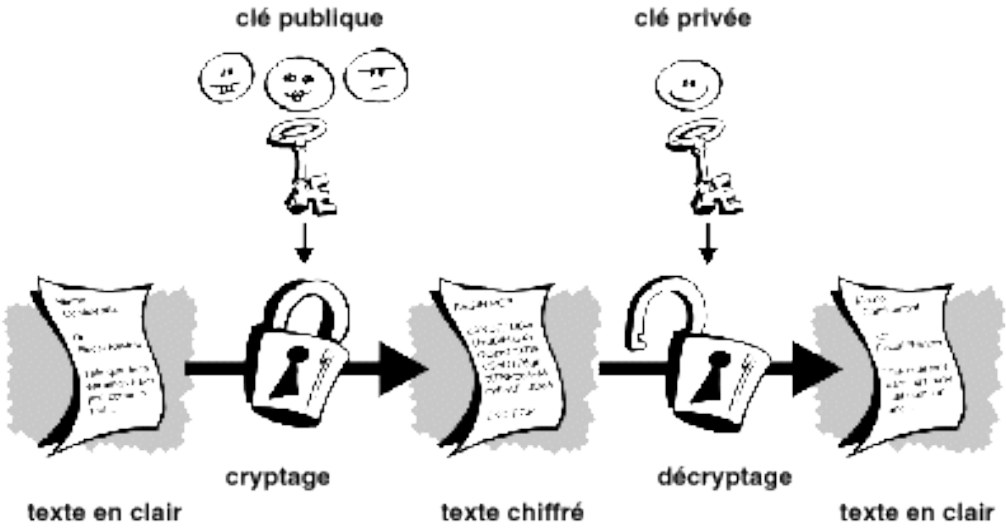


Figure 3 : Le schéma général de la cryptographie asymétrique

L'expéditeur chiffre des messages à l'aide de la clé publique de destinataire, ce dernier les déchiffrent à l'aide de sa clé privée. Le principe de la cryptographie asymétrique est fondé sur l'existence de fonctions de hachage, et à brèche secrète, une telle fonction est difficile à inverser, à moins de posséder une information particulière.

2.4.5 Les méthodes asymétriques modernes

L'algorithme DSA (Degital Signature Algorithm)

L'algorithme à clé publique DSA permet de signer un message, c'est-à-dire de convaincre un destinataire que son expéditeur est bien la personne qu'elle prétend être. Le principe du DSA est de signer un document en réduisant ce dernier à une chaîne de longueur constante au moyen d'une fonction de hachage. Le processus se fait en trois étapes :

1. Génération des clés.
2. Signature du document.
3. Vérification du document signé.

L'algorithme RSA

L'algorithme RSA a été inventé en 1977 par R.Rivest, A.Shamir et L.Adelman. Le RSA est le Système cryptographique à clé publique le plus utilisé de nos jours. Il est composé de deux clés :

-une clé publique qui est diffusée sur quelques services Spécifiques.

-une clé privée qui est gardée secrète par son possesseur.

2.5 Fonction de hachage

2.5.1 Définition

Une fonction de hachage est une fonction à sens unique qui transforme un message de taille quelconque en un résumé court de taille fixe, appelé le condensé de message, l'empreinte du message. Le résumé du message ou encore le message haché. Cependant elle doit en pratique Vérifier les conditions suivantes:

-Résistance à la détermination d'une pré-image, ce qui signifie qu'il doit être impossible en pratique, à partir d'un résumé m de retrouver un message M ayant ce résumé m=h(M).

- Résistance aux collisions, ce qui signifie qu'il est impossible en pratique de construire deux messages M1 et M2 ayant le même résumé : h(M1)=h(M2) [10].

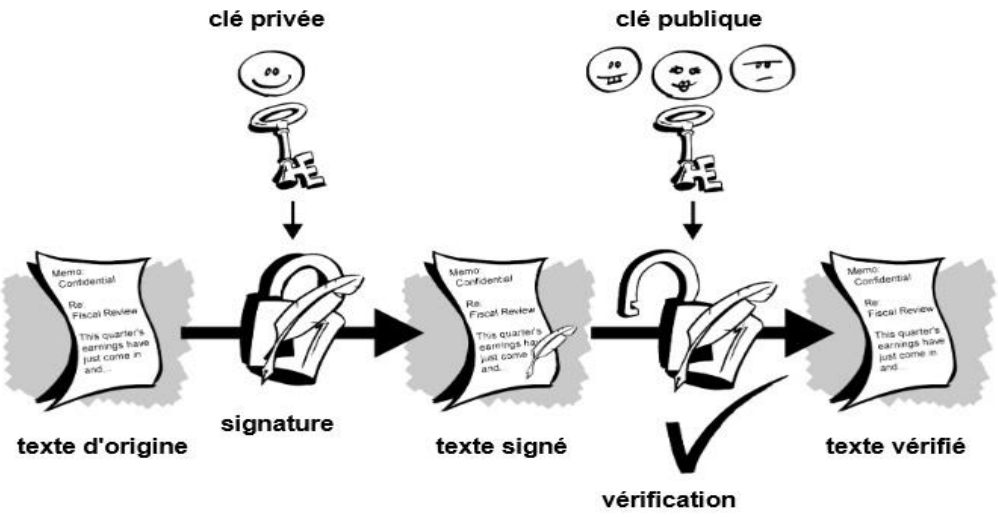
2.6 Signatures numériques

2.6.1 Définition

L'un des principaux avantages de la cryptographie à clé publique est qu'elle offre une méthode d'utilisation des signatures numériques. Celles-ci permettent au destinataire de vérifier leur authenticité, leur origine, mais également de s'assurer qu'elles sont intactes. Ainsi, les signatures numériques de clé publique garantissent l'authentification et l'intégrité des données. Elles fournissent également une fonctionnalité de non répudiation, afin d'éviter que l'expéditeur ne prétende qu'il n'a pas envoyé les informations. Ces fonctions jouent un rôle tout aussi important pour la cryptographie que la confidentialité.

Une signature numérique a la même utilité qu'une signature manuscrite. Cependant, une signature manuscrite peut être facilement imitée, alors qu'une signature numérique est pratiquement infalsifiable. De plus, elle atteste du contenu des informations, ainsi que de l'identification du signataire.

La méthode de base utilisée pour créer des signatures numériques est illustrée sur la Figure (4). Au lieu de chiffrer l'information en utilisant la clé publique d'autrui, vous la chiffrez avec votre propre clé privée. Si l'information peut être déchiffrée avec votre clé publique, c'est qu'elle provient bien de vous.



- FIGURE 4 : Signatures numériques simples

2.6.2 Propriétés de la signature numérique

- On ne peut pas imiter la signature numérique d'une personne si on ne lui a pas dérobé sa clé privée ou tant que l'algorithme de chiffrement asymétrique est reconnu sûr.

- On ne peut pas recopier la signature numérique d'un document pour l'apposer sur un autre document à cause de l'utilisation d'empreinte numérique dans la signature.

- De plus, et contrairement à la signature manuscrite, l'intégrité du document numérique est garantie, c'est à dire toute modification du document après sa signature rend la signature invalide.

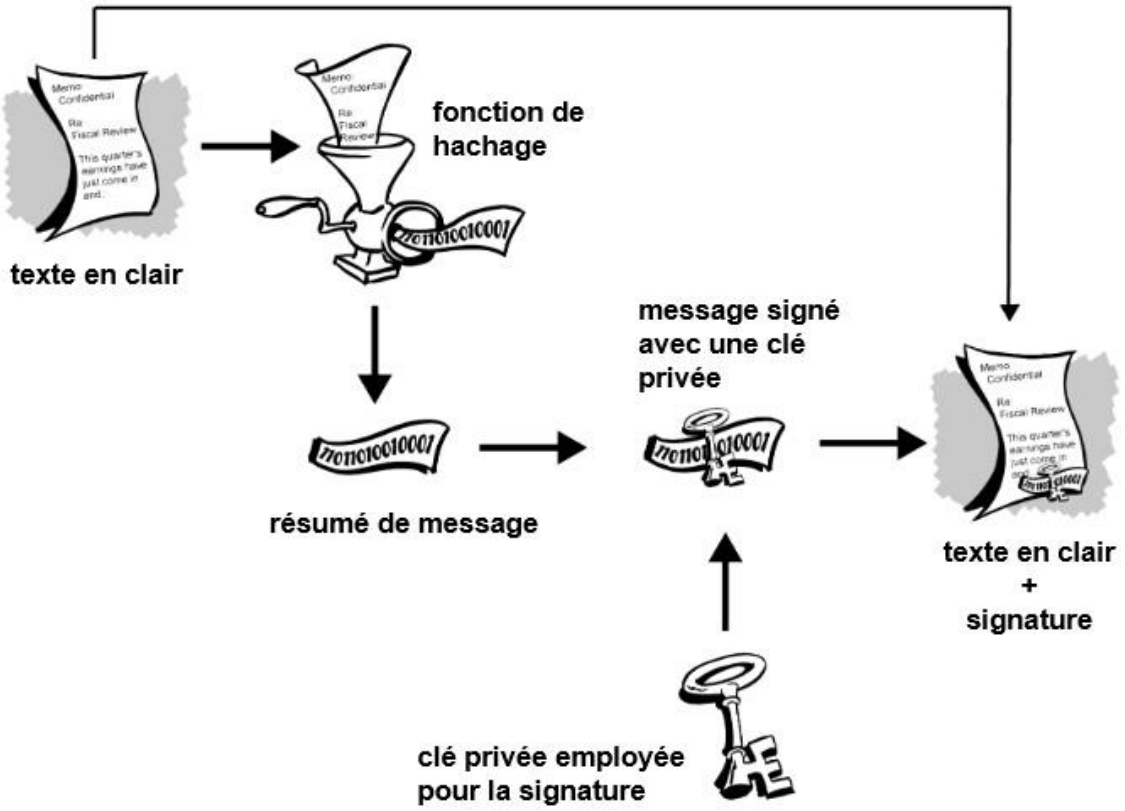


FIGURE 5 : Signatures numériques sécurisées

2.7. Certificat numériques

Un certificat est un élément d'information qui prouve l'identité du propriétaire d'une clé publique. Un certificat est une preuve reconnue de l'identité d'une personne. Les certificats sont signés et transmis de façon sécuritaire par un tiers de confiance appelé autorité de certification(AC).

Un certificat contient notamment :

- l'identité de l'AC ;
- L'identité du propriétaire ;
- la clé publique du propriétaire ;
- la date d'expiration du certificat ;
- la signature de l'AC qui a délivré le certificat ;
- d'autres informations qui n'entrent pas dans la portée de ce mémoire.

En disposant d'un certificat au lieu d'une clé publique, le destinataire peut maintenant vérifier un certain nombre d'aspects au sujet de l'émetteur pour s'assurer que le certificat est valide et qu'il appartient bien à la personne à qui il est censé appartenir.

Il peut notamment :

- comparer l'identité du propriétaire.
- vérifié que le certificat est toujours valide.
- vérifié que le certificat a été signé par un AC de confiance.
- vérifier la signature du certificat de l'émetteur pour s'assurer que ce dernier n'a pas été altéré.

La signature de l'AC peut, à son tour, être vérifiée à l'aide du certificat de cette AC.

Conclusion

Nous avons défini dans ce premier chapitre la sécurité informatique et ses services de base, et évoqué ensuite, les attaques la menaçant. Par la suite, on s'est intéressé aux aspects fondamentaux de la cryptographie, où nous avons présenté ses deux branches : la cryptographie symétrique et asymétrique.

Dans le chapitre suivant, nous allons entamer, la description l'analyse et la conception de notre application web, en nous basant sur UML.

Chapitre II

Analyse & Conception

Description du projet

Modélisation avec UML

Analyse et conception

Les diagrammes de séquence

Le diagramme de classe

Introduction

Dans ce deuxième chapitre qui est l'analyse et conception, nous allons donner la description et l'objectif du projet et toutes les exigences fonctionnelles à respecter.

Dans la deuxième section, nous nous sommes beaucoup plus focalisés sur la modélisation avec UML, et l'analyse et conception.

La conception est l'étape de description de la structure, le comportement et l'architecture d'un système, elle sert à présenter les différentes perspectives de ce dernier.

Dans la dernière section, nous avons élaboré quelques diagrammes de séquence et le diagramme de classe.

1. Description du projet

Notre projet a pour nom:« conception et réalisation d'une application web pour la délivrance et l'authentification des documents académiques et administratifs », comme son nom l'indique, celui-ci a pour objectif de réaliser une application web « SigDoc».

Notre application devra donc regrouper toutes les fonctionnalités nécessaires au partage des documents telles que : consulter vérifier et imprimer ces documents, gérer son journal, demander ces documents, et toutes les fonctionnalités techniques comme : gérer son compte, s'inscrire, s'authentifier, se déconnecter…etc. Mais aussi, elle devra répondre à des exigences non fonctionnelles par sa qualité et ses performances.

« SigDoc » est une banque de documents sécurises à imprimer.

1.1. Objectifs

Nous nous mettons comme but de réaliser une application web qui pourra être en grande partie administrable par une personne n'ayant aucune connaissance technique dans les langages du Web (HTML/PHP/MySQL). Ce projet se basera donc sur les objectifs suivants:

- Manipulation et mise en place d'une signature numérique sécurisée.

(Une signature numérique est un procédé permettant de remplir la même fonction qu'une signature manuscrite c'est-à-dire d'engager la responsabilité du signataire sur le contenu du message signé. Cette signature ne doit pas pouvoir être reniée et doit être vérifiable par tout le monde).

La signature numérique offre principalement :

- la possibilité de signer un document sans se rencontrer (réduction des déplacements).
- la possibilité de signer un document sans l'imprimer (économie de papier).
- la possibilité d'envoyer le document par e-mail (économie de timbre).
- la possibilité de conserver le document au format numérique (simplification et suppression de l'archivage papier).

- La construction d'une signature qui dépendra du signataire mais aussi du contenu du document de sorte qu'une signature valide ne puisse pas être utilisée avec un autre document que le document signé initialement (création de certificats, développement d'une technique de hachage sur le document).

La signature numérique permet, pour un document numérique, de garantir :

- l'identité du signataire ;
- la non-répudiation par le signataire du document signé ;
- l'intégrité du document signé, c'est-à-dire son absence de modification
- La vérification de la validité d'un document.
- Distribution simplifiée des clés.

Cette application doit être donc facilement administrable, grâce à une interface simple et intuitive. Celui-ci devra bien entendu être accessible de l'extérieur.

De manière plus technique, il nous a aussi fallu nous pencher sur le problème du langage de développement, Après une courte réflexion nous avons opté pour PHP/MySQL.

2. Démarche de développement :

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie.

Afin de réaliser notre travail, nous avons utilisé UML comme un langage de modélisation et UP comme une démarche d'Analyse des besoins et de conception de notre application.

2.1. UML

UML (Unifier Modeling Language) n'est pas une méthode, mais un langage graphique qui permet de représenter, et de communiquer les divers aspects d'un système d'information, des textes qui expliquent leurs contenus sont associés aux graphiques. UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet. Aujourd'hui, étant le langage de modélisation d'applications informatiques le plus important du marché. Il est supporté par la quasi-totalité des outils de développement, lesquels permettent l'édition de modèles UML et offrent des capacités telles que la génération de code et le test.

Vu que UML ne nous procure pas une méthodologie à suivre, on a utilisé processus unifié UP.

2.2 Le Processus Unifié

Le Processus Unifié (*Unified Process*) est un processus itératif et incrémental de développement logiciel, il est centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques. L'objectif d'un tel processus, est de maîtriser la complexité

des projets informatiques en diminuant les risques. Nous définissons dans ce qui suit les différents principes de ce processus [11].

Itératif et incrémental : Le projet est découpé en itérations de courtes durées qui aident à mieux suivre l'avancement global de celui-ci. A 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 facilitée. Cette architecture qu'elle soit fonctionnelle, logique, matérielle ou autre doit être modélisée en UML et pas seulement documentée en texte.
Il est important de définir le plus tôt possible, même à grandes mailles, l'architecture type qui sera retenue pour le développement, l'implémentation et ensuite le déploiement du système.
Le vecteur des cas d'utilisation peut aussi être utilisé pour la description de l'architecture.

Conduit par les cas d'utilisation : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation permettent d'exprimer les interactions du système avec les utilisateurs, donc de capturer les besoins. Il sert aussi à montrer comment ces derniers constituent un vecteur structurant pour le développement et les tests du système.
Ainsi le développement peut se décomposer par cas d'utilisation et la réception du logiciel sera elle aussi articulée par ces cas d'utilisation.

Piloté par les risques : l'analyse des risques doit être présente à tous les niveaux du développement d'un système. Il est important de bien évaluer ces risques afin d'aider à la bonne prise de décision. Du fait de l'application du processus itératif, le processus unifié contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives.

2.2.1. Activités du processus unifié

Pour mener efficacement un tel cycle, les développeurs ont besoin de toutes les représentations du produit logiciel qui se définissent à partir des étapes suivantes :

Expression des besoins : le processus unifié (noté UP) propose d'appréhender l'expression des besoins en se fondant sur une bonne compréhension du domaine concernant le système à développer et une modélisation des procédures du système existant. Ainsi, le processus unifié distingue deux types de besoins :

- Les besoins fonctionnels qui conduisent à l'élaboration des cas d'utilisation,
- Les besoins non fonctionnels (techniques) qui aboutissent à la rédaction d'une matrice des exigences.

Analyse : l'analyse permet une formalisation du système à développer en réponse à l'expression des besoins formulés par les utilisateurs. Elle se concrétise par l'élaboration de tous les diagrammes donnant une représentation du système tant statique (diagramme de classes principalement), que dynamique (diagramme des cas d'utilisation, de séquence, d'état transition, d'activités ...).

Conception : la conception prend en compte les choix d'architecture technique retenus pour le développement et l'exploitation du système. Elle permet d'étendre la représentation des diagrammes effectuée lors de l'analyse en y intégrant les aspects techniques les plus proches des préoccupations physiques.

Implémentation : cette phase correspond à la production du logiciel sous forme de composants, c'est-à-dire de codes sources, de scripts, d'exécutables et d'autres éléments du même type de bibliothèques ou de fichiers.

3. Expression des besoins

3.1. Exigences fonctionnelles

Inscription : Un internaute peut faire une inscription, où il saisit ses informations personnelles (nom, prénom, numéro de sa carte identité, numéro de téléphone et adresse), crée son profil afin de devenir membre dans la société.

Authentification : L'authentification consiste à assurer la confidentialité des données, afin d'accéder au site et pouvoir avoir part aux différentes fonctionnalités (consulter demander d'un document, …). L'authentification se montre alors indispensable pour tout utilisateur.

L'ajout d'un document : Après l'authentification de l'administrateur, il peut approuver les documents administratifs automatiquement selon les demandes reçues.

Après l'authentification, l'institut peut ajouter des documents académiques automatiquement selon les demandes reçues.

Demande d'un document : Après l'authentification, un membre peut effectuer une demande selon les différents documents.

- Documents administratifs (acte de naissance, la résidence…) pour administrateur
- Documents académiques (certificat de scolarité, diplôme..) pour l'institut.

Consultation des documents : A tout moment, après l'authentification un membre peut consulter des documents personnels ou administratifs, après les avoir reçus de la part d'un administrateur ou d'institut, afin de les imprimer plus rapidement.

Impression des documents : Après la consultation, un membre peut effectuer une impression selon les différents documents où il sélectionnera le document à imprimer puis effectue une demande d'impression.

Vérification d'un document : Un internaute a la possibilité de vérifier la validité d'un document sur papiers en utilisant l'application.

Compte : Un institut a des droits sur ses informations personnelles. Il peut donc modifier à tout moment ses données telles que son nom, adresse, numéro de téléphone…et même son mot de passe.

3.2. Exigences non fonctionnelles

Exigences de qualité

Afin que l'application donne envie aux membres de le faire connaître à leurs proches et de lui être fidèle, il est important de répondre aux exigences de qualité suivantes :

Ergonomie efficace : l'obtention des documents (administratifs, académiques) doit être un plaisir. La mise en page du site doit faciliter au maximum la démarche à l'aide d'une présentation claire et intuitive.

Interface graphique : Les différentes couleurs et choix typographiques doivent permettre à un utilisateur de repérer les différentes fonctionnalités qui s'offrent à lui.

Disponibilité : l'utilisateur doit pouvoir demander et consulter ses documents académiques et administratifs à tout le moment.

Exigences de performance

Il faut aussi prendre en compte les exigences quantitatives :

La Complexité de l'algorithme doit être optimale.

La rapidité du service.

3.3. Cahier des charges

Après avoir vu les différents objectifs du site, nous avons établi le cahier des charges suivant :

– Une fois le projet terminé, le responsable doit être capable d'administrer le site seul.

– Ce projet ne doit poser aucun problème quelconque au responsable.

– L'interface d'administration doit être intuitive et simple d'utilisation.

– Un document signé est dissimulé par une signature numérique basée sur le chiffrement RSA et une technique de hachage SHA dont le document permet de retrouver l'utilisateur indélicat.

– La sécurité doit être suffisamment présente afin que des personnes malintentionnées ne puissent pas modifier le site.

– Celui-ci doit être dans les tons gris et blanc, nous avons décidé de rendre ces couleurs plus pâles pour ne pas agresser l'œil et de manière à améliorer la lisibilité.

3.4. Identification des acteurs (les cas d'utilisations)

Nous allons répondre aux questions suivantes : Quels sont les utilisateurs du système ? Quelles sont leurs interactions avec celui-ci ? Il faut donc identifier les différents acteurs ainsi que les cas d'utilisation c'est-à-dire les différentes fonctionnalités du système.

Les acteurs pour le site « SigDoc » sont les suivants :

· **L'internaute :** personne qui n'est pas membre du site et qui souhaite le devenir, qui peut vérifier la validité d'un document.

• **L'administrateur :** L'administrateur est une personne qui ouvre droit à tout accès, et contrôle de manière intégrale du site web et satisfaire les besoins des clients.

• **L'institut :** société qui possède un compte, qui peut consulter et satisfaire les demandes des clients, et qui modifie son compte.

• **L'utilisateur :** personne qui possède un compte, qui peut demander, consulter ses documents, ainsi que modifier son compte, il peut être membre de l'institut.

Voici donc le diagramme des cas d'utilisation.

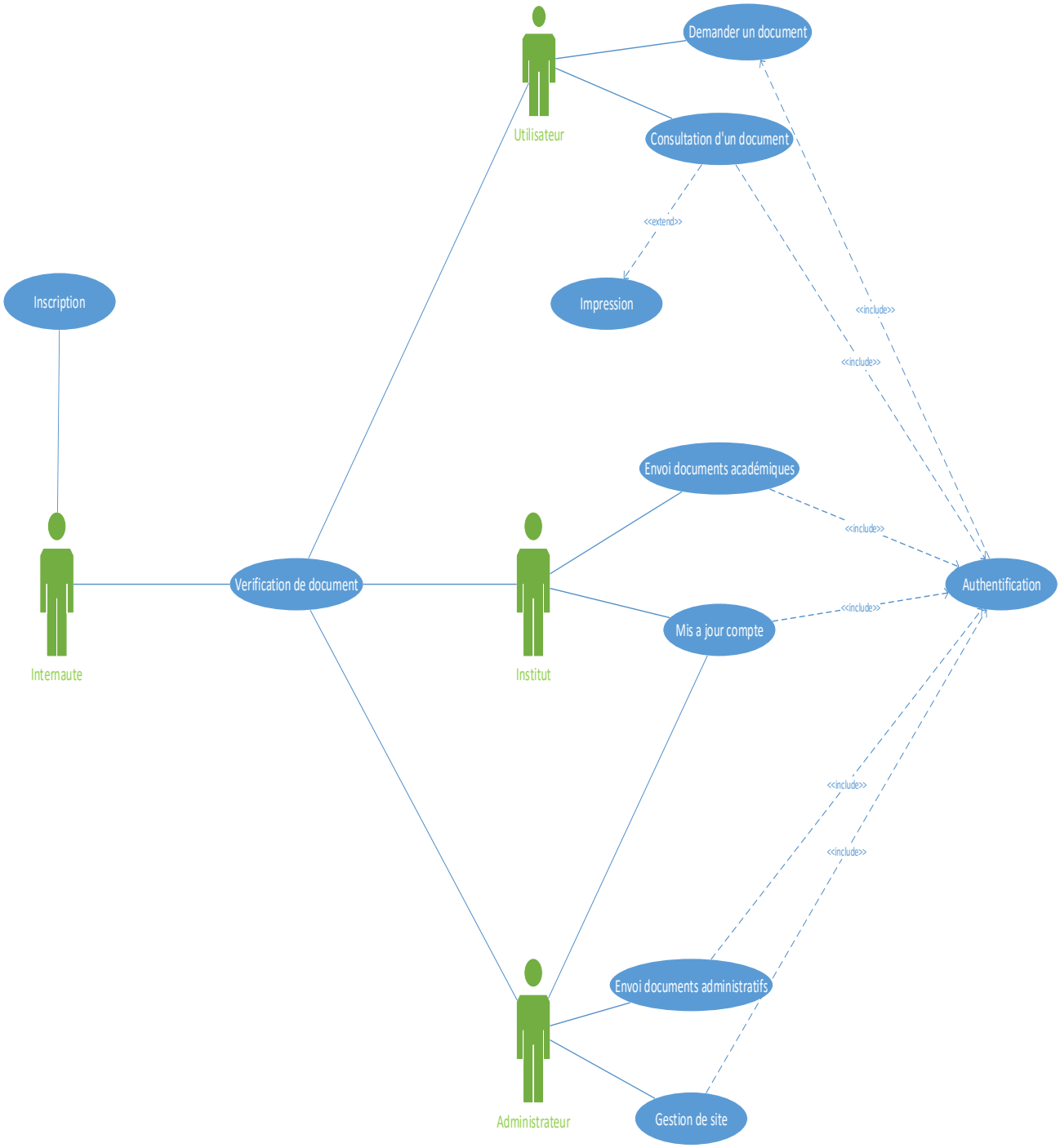


FIGURE 6- Le diagramme des cas d'utilisation.

4. Analyse et conception

Nous allons maintenant nous pencher sur l'analyse et conception du projet en lui-même, et pour cela on va commencer par la modélisation dynamique qui comporte différents diagrammes de séquence, et on passera à la modélisation statique qui contient le dictionnaire de données, diagramme de classe et le model logique de données.

4.1. Modélisation dynamique

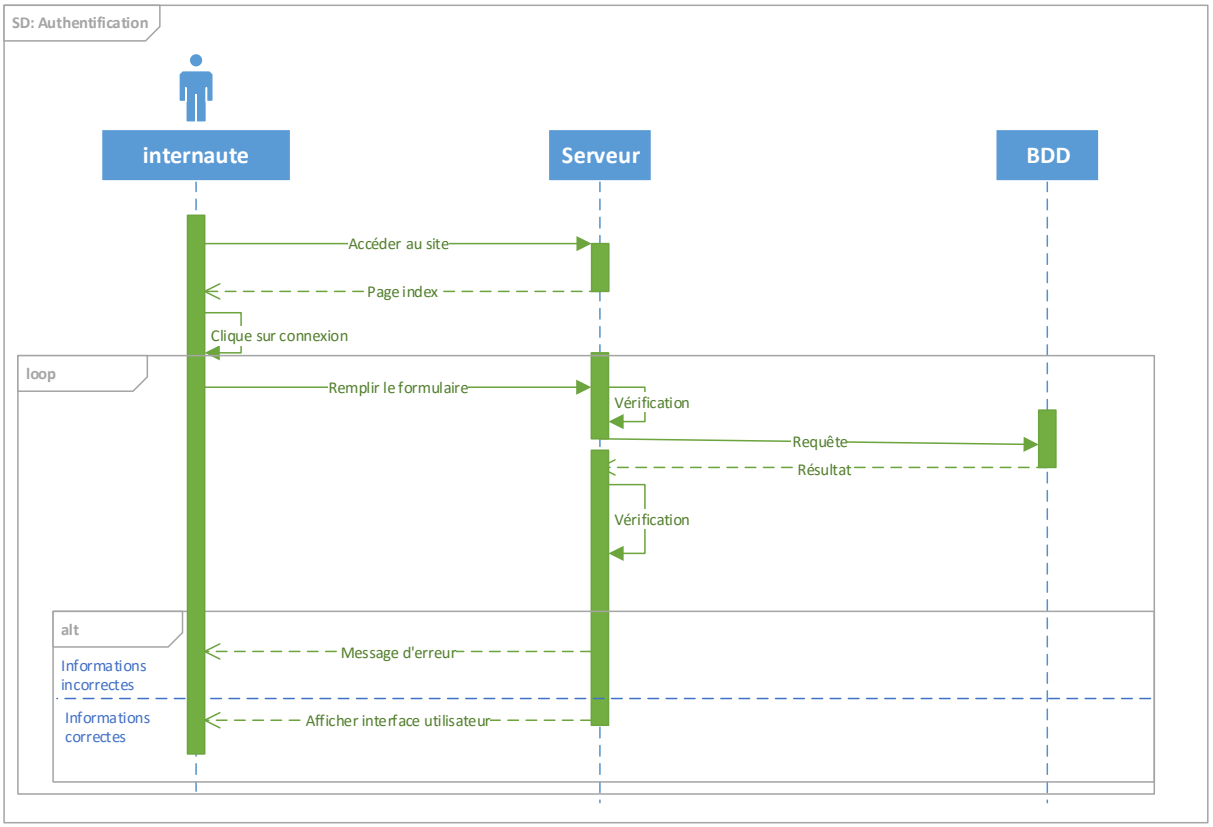
Pour modéliser la partie dynamique de notre système, nous devrions analyser l'interaction entre les différents acteurs, serveurs et base de données, qui est généralement illustrée dans les diagrammes de séquence.

4.1.1. Les diagrammes de séquence

Dans cette partie, nous nous intéressons aux cas d'utilisation illustrant quelques cas disponibles sur le site. Les diagrammes de cas d'utilisation prélevés sont représentés ci-dessous

4.1.1.1. Cas d'utilisation « Authentification »

L'authentification consiste à assurer la confidentialité. Elle se base sur la vérification de l'information associée à un utilisateur (généralement un pseudonyme et un mot de passe). Ces informations sont préétablies dans une base de données, ou les serveurs récupèrent le haché du mot de passe, qui sera comparé au haché du mot de passe introduit. Lors de l'authentification d'un utilisateur, deux cas peuvent se présenter : informations correctes ou information incorrecte, ce qui explique l'utilisation de l'opérateur « alt ». Si les informations fourniers sont correctes, alors le serveur accorde l'accès à l'interface appropriée. En revanche, si l'utilisateur saisit des informations incorrectes, le serveur lui génère un message d'erreur. Ce procédé est exécuté à chaque fois que l'utilisateur tente de s'authentifier, c'est pourquoi nous avons utilisé l'opérateur « loop ».



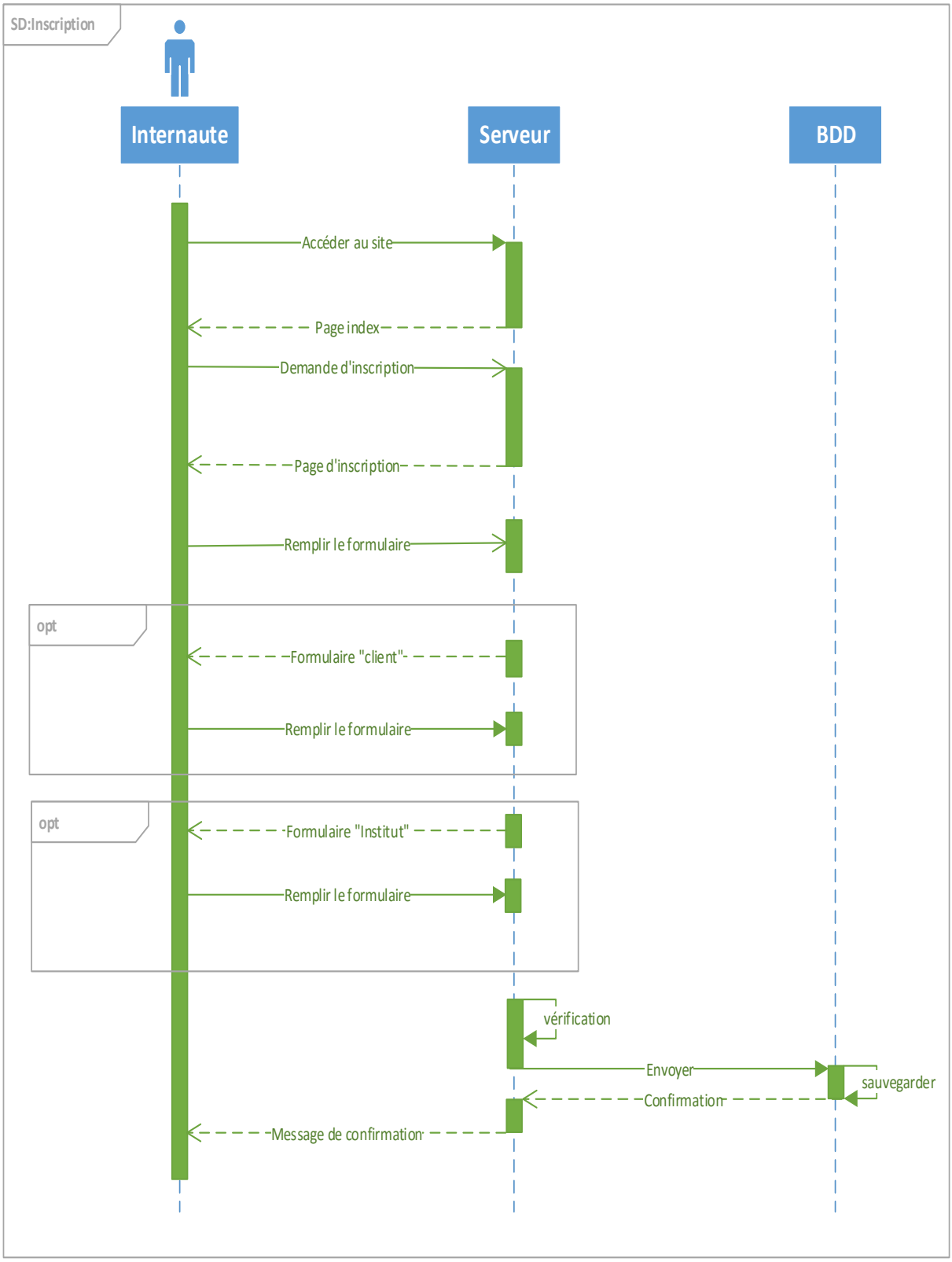
FIGUERE 7- diagramme de séquence du cas d'utilisation «Authentification»

4.1.1.2. Cas d'utilisation « Inscription »

Apres avoir accéder au site, l'internaute peut s'inscrire. Et pour cela, il devra émettre une demande d'inscription en cliquant sur le bouton «Inscription» et une page d'inscription lui sera renvoyer avec deux options, inscription en tant que client et une autre en tant qu'institut.

En suite après le choix de l'option, l'Internaute va remplir le formulaire en saisissant ces informations personnelles « nom, prénom, pseudo, numéro de sa carte d'identité, numéro de téléphone et l'adresse » et lorsqu'il termine en cliquant sur le bouton «S'inscrire », ses information seront envoyer a la base de donné, et cette dernière sauvegarde et répond avec une confirmation.

Enfin le system répond à l'Internaute avec un message de confirmation.



FIGUERE 8- diagramme de séquence du cas d'utilisation « Inscription »

4.1.1.3. Cas d'utilisation « Demande de document »

Après authentification, si l'utilisateur veut demander un document il devra émettre une demande de document en cliquant sur le bouton « Demande » un formulaire lui sera remis avec quatre options :

- Formulaire « acte de naissance »
- Formulaire « résidence »
- Formulaire « certificat de scolarité »
- Formulaire « Diplôme »

Après avoir choisi une option, l'utilisateur remplie le formulaire de l'option choisie, et Il confirme sa demande en cliquant sur le bouton « Envoyer ».

Le serveur envoi une requête à la base de données contenant l'information demander par l'utilisateur, et la base de données enregistre ces informations et confirme au serveur que la demande a bien été enregistré, et en dernier lieu le serveur confirme à son tour à l'utilisateur, par un message, que ça demande a été prise en charge.

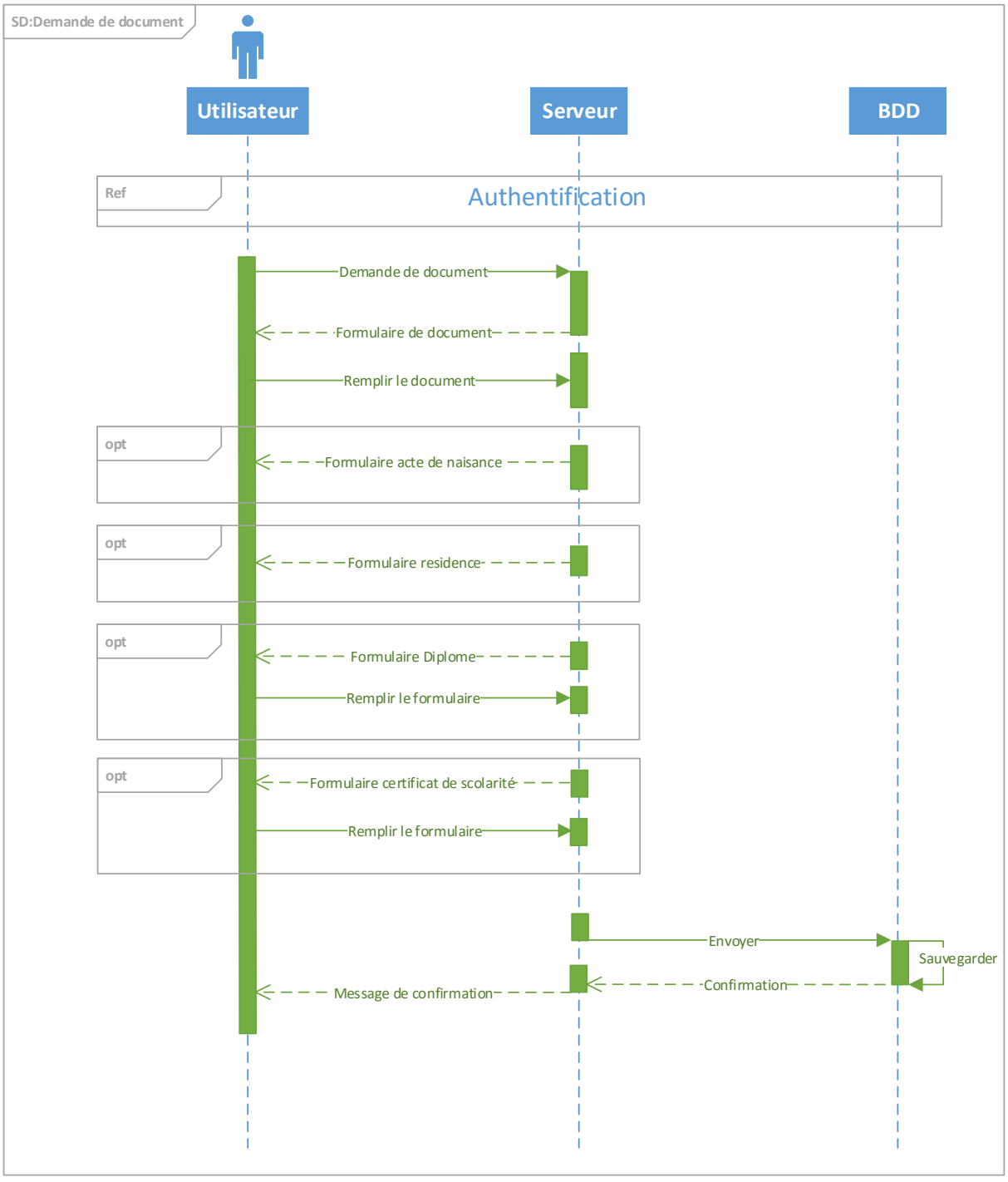


FIGURE 9- Diagramme de séquence du cas d'utilisation « Demande de document ».

4.1.1.4. Cas d'utilisation « Imprimer »

Après authentification, dans le cas où le client désire imprimer un document, il passera d'abord par la liste de ses documents, et une page avec deux options s'affichera, une personnel qui contient des documents tel que : acte de naissance et résidence. Une autre administratif qui peut contenir : diplôme et certificat de scolarité.

Le client aura qu'à ouvrir une des options, celle dont il est intéressé, afin de choisir le document voulu et emmètre une demande de visualisation, le serveur envoie à son tour une requête à la base de données, une réponse sera renvoyée à partir de la base de donnés vers le serveur, la visualisation de documents s'affichera dans le navigateur après la réponse du serveur.

Enfin pour l'impression de ce document, le client va cliquer sur le bouton « Imprimer », une demande d'impression va être envoyée depuis le navigateur vers le serveur.

Le serveur aura besoin de la clé privée de l'institut d'où le document provient. Une requête va être demandée pour obtenir cette clé à partir de la base de donnés.

En retour cette dernière récupère la clé privée pour la transmettre au serveur.

Le serveur va hacher les informations qui se trouvent dans le document demandé à l'aide d'une fonction de hachage, le résultat sera signé en utilisant la clé privée récupéré. Le résultat final s'affichera sous forme d'une image QR CODE dans le navigateur et le document peut être imprimé.

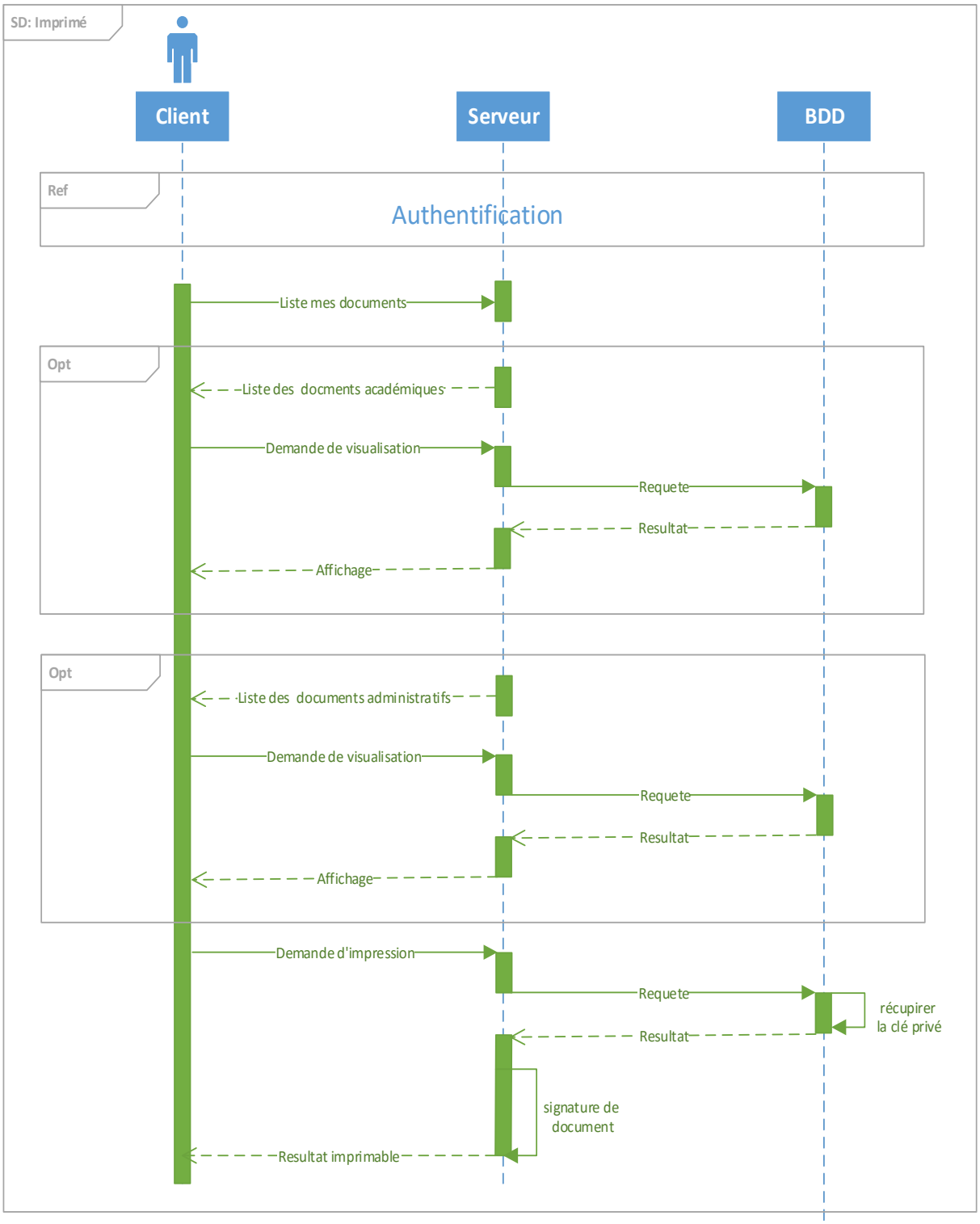


FIGURE 10- Diagramme de séquence du cas d'utilisation « Imprimer ».

4.1.1.5. Cas d'utilisation « Répondre sur un document »

Après authentification, Lorsqu'un institut reçoit une demande de document, il va répondre à cette demande en cliquant sur demande. Un formulaire lui sera renvoyé selon le type du document demandé et l'institut concerné, après avoir remplie le formulaire, l'institut clique sur le bouton envoie, une requête va être envoyée depuis le navigateur au serveur qui transmettra les informations saisis à la base de données. La base à son tour sauvegardera et confirmera la sauvegarde.

Enfin, le serveur va répondre avec un message de confirmation au navigateur de l'institut.

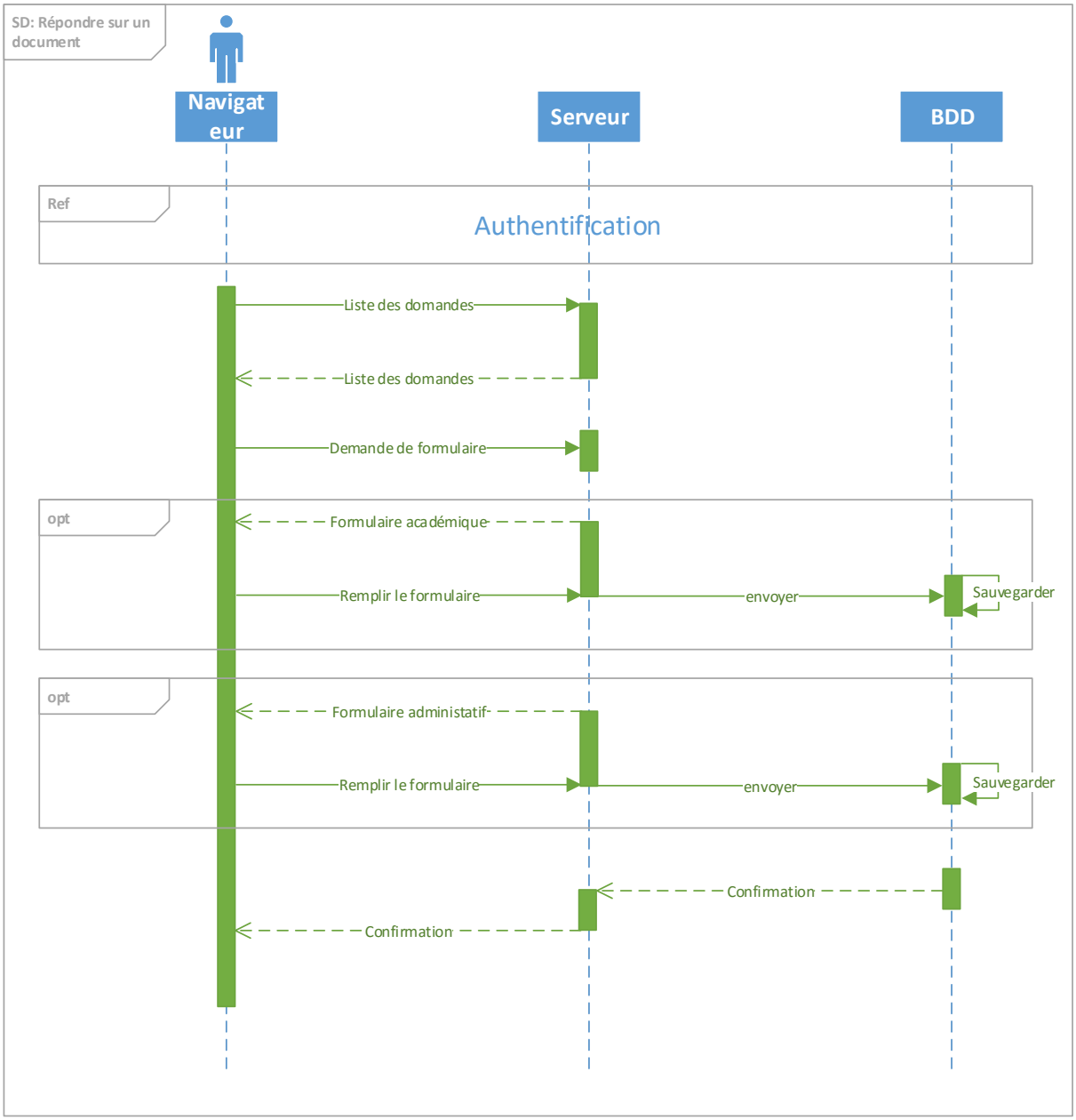


FIGURE 11- Diagramme de séquence du cas d'utilisation « Répondre sur un document ».

5. modélisation statique

Pour modéliser la partie statique du système, la première étape est la réalisation du dictionnaire de données représenté ci-dessous, et le modèle logique de données déduit à partir de du diagramme de classe conçu.

5.1. Dictionnaire de données

Avant de pouvoir réaliser le diagramme de classe, on doit d'abord définir le dictionnaire de données illustré ci-dessous :

Classe	Définition de l'attribut	Attribut	Type
compte	Identifiant d'authentification	id	Numérique
	Pseudonyme d'authentification	pseudo	Texte
	Mot de passe d'authentification	MotDePasse	Numérique
	Email d'authentification	email	texte
	Numéro de téléphone	tel	Numérique
Document	Identifiant de document	idDoc	Numérique
	Date création de document	dateCreation	Date
	Date péremption de document	datePeremption	Date
Adresse	Identifiant d'adresse	idAdress	Numérique
	Numéro de rue	numRue	Numérique
	Nom de rue	nomRue	Texte
	Commune d'adresse	Commune	Texte
	Daïra d'adresse	diara	Texte
	Wilaya d'adresse	wilaya	Texte
	Commentaire	Commentaire	Texte
Institut	Identifiant de l'institut	idIns	Numérique
	Nom de l'institut	nomIns	Texte
	Clé publique de l'institut	publicKey	Numérique
	Clé privée de l'institut	privateKey	Numérique
Personne	Identifiant de la personne	idPer	Numérique
	Nom de la personne	nom	Texte
	Prénom de la personne	Pronom	Texte
	Numéro de carte national	numéroDeLaCarte	Numérique
	Sexe de la personne	sexe	Texte (M, F)
Document Académique	Identifiant de document administratif	idDocAdm	Numérique
	Numéro de référence de document	numeroref	Numérique
	Par qui le document a été délivré	delevrepar	Texte
	Spécialité de demandeur de document	Specialite	texte
	La date de délivrance de document	delevrele	date
	Matricule de document	Matricule	texte
	Année d'inscription	anneeIns	date

	Diplôme préparé	diplomePrepare	Texte
	Formation suivi dans l'institut	formation	texte
	Vu de document	vu	Texte
Document Administratif	Formation de diplôme	idDocPer	Numérique
	Date délibération de diplôme	delivrea	Texte
	Identifiant de l'acte naissance	delivrele	Date
	Date de naissance	dateNai	Date
	Lieu de naissance	lieuNai	Texte
	Prénom de père de l'acte naissance	prenomPere	Texte
	La profession de père	jobPere	Texte
	Nom de mère	nomMere	Texte
	Prénom de mère	prenomMere	Texte
	Profession de mère	jobMere	Texte
	Temoin de document	temoin	Texte
	Date de mariage	dateMariage	Date
	Partenaire	partenaire	Texte
	Degré de la parenté	degreDeParente	Texte
	Preuve fournie	PreuvesFournies	Texte
Demande	Identifiant de la demande	idDem	Numérique
	Type de la demande	typeDoc	Texte
	Date de demande	dateDemande	Date
	Date de service de demande	dateService	Date

TABLEAU 1 : Dictionnaire de données

5.2. Diagramme de classe

Dans cette partie, nous étudierons les entités statiques du système. Ceci est illustré par le diagramme de classes suivant :

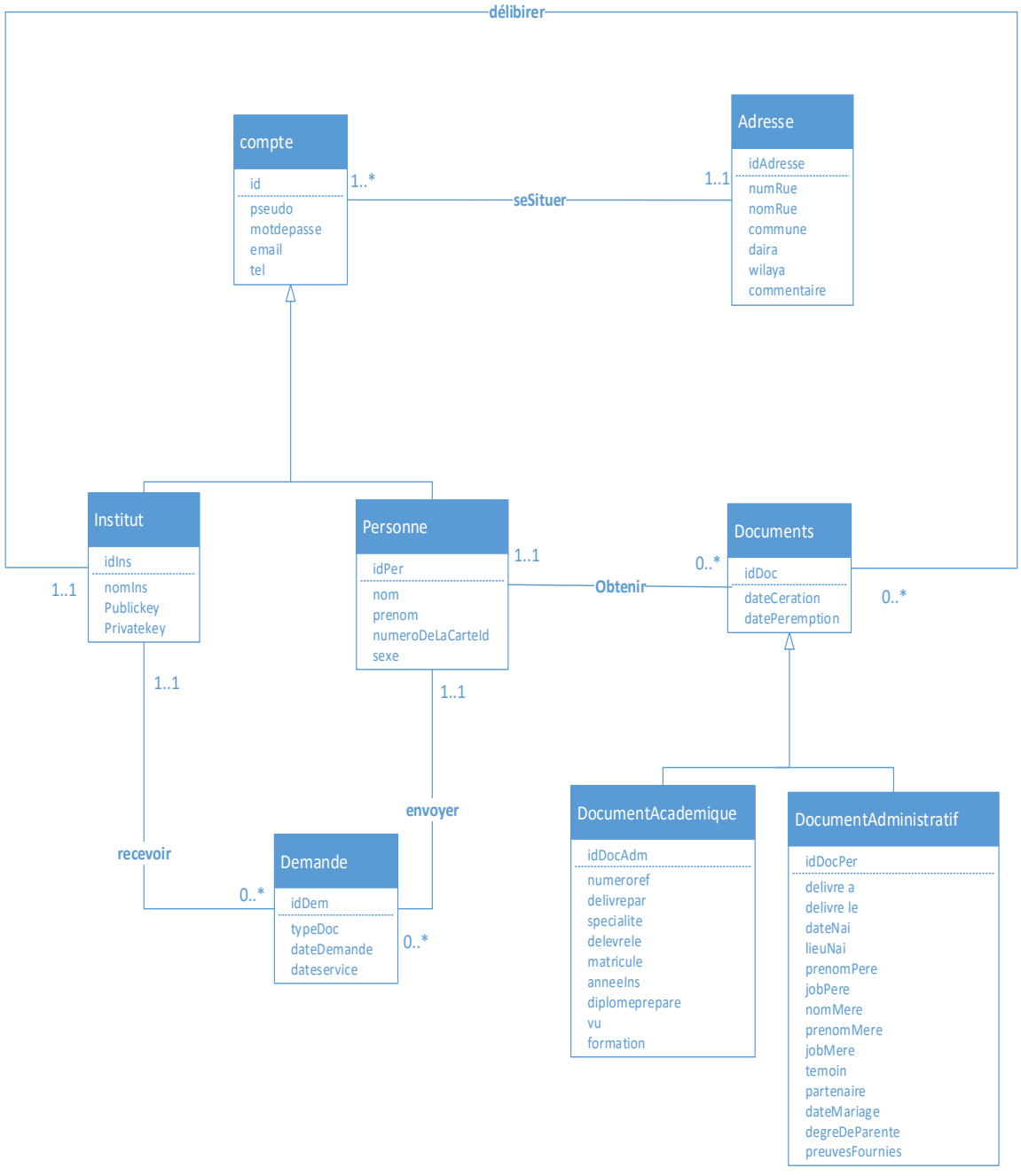


FIGURE 12- Le Diagramme de classe.

5.3. Modèle logique des données

Le modèle logique de données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements. Ainsi, le modèle logique est dépendant du type de base de données utilisé.

Le modèle logique de données associé au diagramme de classes relatif au site web à réaliser.

Compte (Id, pseudo, motDePasse, tel, IdAdresse#) ;

Personne (IdPer, nom, prenom, numeroDeLaCarteId, sexe, Id#) ;

Institut (IdIns, nomIns, Publickey, Privatekey, Id#);

Documents (IdDoc, dateCreation, datePeremation, IdPer#, IdIns#);

DocumentAcademique (IdDocAdm, numeroref, delivrepar, specialite, delevrele, matricule, anneeIns, diplomeprepare, vu, formation, IdDoc#);

DocumentAdministratif (IdDocPer, delivrea, delevrele, dateNai, lieuNai, prenomPere, jobPere, nomMere, prenomMere, jobMere, temoin, partenaire, datedeMariage, degreDeParente, preuveFournies, IdDoc#);

Adresse (IdAdresse, numRue, nomRue, commune, daira, wilaya, comentaire) ;

Demande (IdDem, typeDoc, datedemande, dateservice, IdPer#, IdIns#) ;

Conclusion :

Ce chapitre a été consacré à l'analyse et à la conception de notre site web, où nous avons détaillés les différents acteurs et composants de notre système et nous l'avons modélisé avec UML.

La phase conception est terminée, dans le chapitre qui suit, nous allons nous intéresser à la réalisation de notre application.

Chapitre III

Réalisation

Langages utilisés

Outils et logiciels

IHM et sécurité d'application

Introduction

Dans ce chapitre intitulé « Réalisation », les différents langages, outils et logiciels mis en œuvre pour l'implémentation de notre application web seront présentés de manière organisée en commençant par une brève description du diagramme de déploiement, suivi de la figure qui le caractérise.

Les langages utilisés pour la réalisation de l'application seront brièvement définis dans le point suivant; et puisque un langage de programmation ne peut se détacher des outils et des logiciels de programmation, chacun de ces derniers se verra attribué une brève description en dessous d'une introduction. Ensuite on va s'intéresser à l'IHM et aux interfaces de l'application.

Enfin, nous allons nous intéresser à la sécurité des sites web.

3.1 Le Diagramme de déploiement

Le diagramme de déploiement du système que nous avons réalisé, est illustré par la figure suivante :

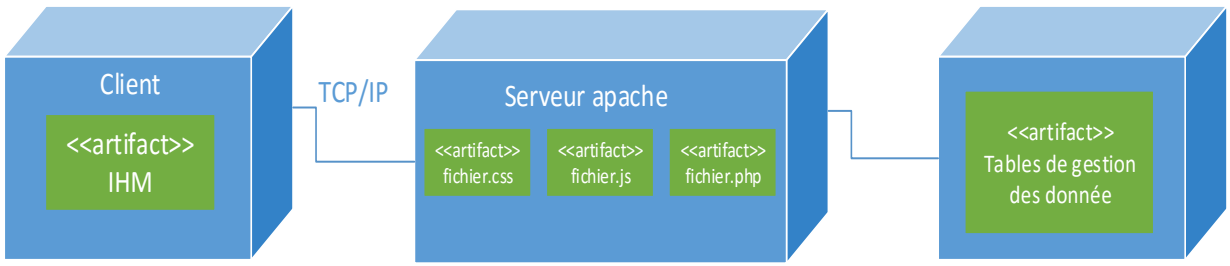


FIGURE 13 : Diagramme de déploiement du site.

Chaque utilisateur a son propre poste client qui est un « PC » connecté au serveur web Apache, qui est lui-même un serveur d'applications sur lequel est déployée en particulier l'application d'authentification. Chacun de ces utilisateurs partage la même interface homme-machine qui utilise le service d'authentification contenu par le serveur Apache. Toutes les tables sont stockées dans une base de données spécifique qui est hébergée par le serveur de base de données « MySQL ».

3.2 Langages et Outils utilisés

Au cours de la réalisation de l'application web, une multitude de langages de programmation ont été mis en œuvre afin que cette dernière puisse être implémentée comme elle a été décrite lors de la phase de conception.

- Voici une brève description de chacun des langages utilisés.

3.2.1 HTML (HyperText Markup Language)

Toute page web comprend une base de langage HTML. Il s'agit d'un langage de balisage qui définit essentiellement la structure de la page web (titres, tableaux, paragraphes, etc.).

C'est un langage qui permet de créer des hyperliens, à savoir des liens d'un document à un autre ou d'un endroit d'un document à un autre endroit du même document (identificateur de fragment).

3.2.2 JavaScript

JavaScript est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le site), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au-dessus d'un élément, du redimensionnement de la page...). La version standardisée de JavaScript est l'ECMAScript. [12]

3.2.3 AJAX

AJAX n'est ni une technologie ni un langage de programmation mais une manière de développer des pages web en se basant sur certaines technologies comme HTML et CSS pour la présentation, DOM (Document Object Model) pour la représentation en objets de la page web, JavaScript et en particulier l'objet XMLHTTPRequest pour manipuler des requêtes et des réponses.

AJAX (*Asynchrones JavaScript And XML*, traduisez *JavaScript asynchrone et XML*) est une méthode de développement web basée sur l'utilisation d'un script JavaScript pour effectuer des requêtes web à l'intérieur d'une page web sans recharger la page. AJAX rend plus interactifs les sites web et offre une meilleure ergonomie ainsi qu'une réactivité améliorée en permettant de modifier interactivement une partie de l'interface web seulement. [13]

3.2.4 PHP

Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage orienté-objet. [14]

3.2.5 MySQL

MySQL est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données. [15]

3.2.6 CSS

Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML. [16]

3.3 Outils et logiciels utilisés

Un logiciel ou une application est un ensemble de programmes, qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier (exemple : logiciel de gestion de la relation client, logiciel de production, logiciel de comptabilité, logiciel de gestion des prêts).

On distingue en général, dans un système informatique, la partie matérielle (l'ordinateur et ses périphériques) et la partie logicielle, immatérielle (les programmes " écrits " sur le disque dur).

Le logiciel un bien immatériel, mais surtout c'est un bien non-rival, c'est-à-dire qu'il ne s'use pas, c'est un bien dont la consommation par un individu donné n'empêche pas d'autres consommateurs d'en jouir simultanément.

Le terme logiciel est souvent employé pour désigner un programme informatique, et inversement, bien qu'un logiciel puisse être composé d'un seul ou d'une suite de programmes.

Ce dernier cas est d'autant plus fréquent que la capacité réduite de calcul de l'ordinateur oblige à une segmentation des tâches en plusieurs modules séparés ; cependant, les énormes capacités des micro-ordinateurs actuels en regard des applications typiques de la bureautique ont permis la réalisation d'applications monolithiques.

Généralement, les programmes sont accompagnés d'un ensemble de données permettant de les faire fonctionner (par exemple, un jeu viendra avec de nombreuses images, animations, sons, etc.).

Pour fonctionner, un logiciel nécessite l'utilisation d'un ordinateur (micro-ordinateur, station de calcul, mainframe, supercalculateur, etc.) sur lequel existe à l'origine un " logiciel-moteur " (système d'exploitation) qui accepte le " logiciel-application ".[17]

3.3.1 Aptana Studio

Aptana IDE est un environnement complet pour le développement web multiplateforme (Linux, MacOS, Windows) Open-Source et gratuit. Aptana est basé sur Eclipse, éditeur très apprécié des développeurs, et propose de nombreuses fonctionnalités : auto-complétion, fonction de preview, debugger JavaScript, gestion ftp...etc.

3.3.2 Environnement Apache/MySQL/PHP (WampServer)

Le serveur web « Apache » est le serveur Web le plus utilisé sur Internet. Il se met en attente des requêtes transmises à son attention sur le réseau par un programme client. Apache est gratuit et open source offrant une grande stabilité et flexibilité grâce notamment à sa structure. [18]

MySQL et PHP sont respectivement un Système de Gestion de Bases de Données (SGBD) et un langage impératif orienté objet.

WampServer (anciennement WAMP5) est une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que PHPMyAdmin pour l'administration web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

3.3.3 Paint.NET

Paint.NET est un outil de retouche photo particulièrement simple à prendre en main. De nombreuses fonctionnalités sont proposées faisant du produit un logiciel incontournable pour les photographes amateurs. [19]

3.3.4 Visio

Microsoft Visio (officiellement Microsoft Office Visio) est un logiciel de diagrammes pour Windows qui fait partie de la suite bureautique Microsoft Office.

- Les principales fonctionnalités de Visio sont :

  - Créer des diagrammes professionnels rapidement avec des formes mises à jour et de nouveaux outils et options de mise en forme.

  - Associer des diagrammes à des données dynamiques et interpréter les données complexes.

  - Collaborer facilement en équipe sur les diagrammes. [20]

3.3.5 Les Navigateurs Web

Un navigateur Web est un logiciel conçu pour consulter le World Wide Web. Techniquement, c'est au minimum un client HTTP.

Il existe de nombreux navigateurs Web, pour toutes sortes de matériels (ordinateur personnel, tablette tactile, téléphones mobiles, etc.) et pour différents systèmes d'exploitation (GNU–Linux, Windows, Mac OS, iOS et Android). Les plus utilisés à l'heure actuelle sont, Google Chrome, Mozilla Firefox, Internet Explorer, Safari et Opera.

Le terme navigateur Web est inspiré de Netscape Navigator. D'autres métaphores sont ou ont été utilisées. Le premier terme utilisé était browser, comme en anglais. Par la suite, on a vu fureteur (surtout utilisé au Québec), butineur, brouteur, arpenteur, fouineur ou encore explorateur (inspiré d'Internet Explorer). Le terme navigateur internet, bien qu'incorrect, est également souvent rencontré.

- **Google Chrome**

Chrome est un navigateur web développé par Google basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS.

Il est annoncé le 1er septembre 2008 dans une bande dessinée de Scott McCloud, la veille de la sortie de la première version beta. La première version stable est quant à elle dévoilée le 11 décembre 2008. Selon l'institut StatCounter, Google Chrome devient en juin 2012 le navigateur le plus utilisé dans le monde, avec environ un tiers des utilisateurs. [11]

- **Internet Explorer**

Internet Explorer (officiellement Windows Internet Explorer depuis la version 7, anciennement Microsoft Internet Explorer), parfois abrégé IE, MIE ou MSIE, est le navigateur Web développé par Microsoft, installé par défaut avec Windows. Depuis qu'il a détrôné Netscape Navigator à la fin des années 1990 et jusque vers 2012, c'est le navigateur Web qui a le plus été utilisé au monde. Ses principaux concurrents sont Mozilla Firefox (depuis 2004) et Google Chrome (depuis 2008). [11]

- **Safari**

Safari est un navigateur web pour Mac, Windows et iOS développé par Apple.

Il est téléchargeable gratuitement depuis le 7 janvier 2003, soit depuis Mac OS X v10.2. Le 8 juin 2009 la version 4.0 est sortie pour les plateformes Mac OS X v10.4 (ou ultérieur), Windows XP et Vista (ou ultérieur). Ce navigateur est celui installé par défaut sur tous les ordinateurs Mac depuis Mac OS X v10.3. [11]

- **Opera**

Opera est un navigateur Web gratuit et multiplateforme développé par la société norvégienne Opéra, qui propose plusieurs logiciels relatifs à Internet.

Opera est un navigateur relativement peu utilisé par rapport aux autres navigateurs web, totalisant 1,30 % de parts de marché en janvier 2013 . Il peut cependant se féliciter d'être le troisième navigateur mobile mondial avec 13,65 % de parts de marché en février 2014. [11]

- **Mozilla Firefox**

C'est un navigateur Web libre et gratuit, développé et distribué par la Mozilla Foundation avec l'aide de milliers de bénévoles, grâce aux méthodes de développement du logiciel libre (open source) et à la liberté du code source.

Firefox est à l'origine un programme dérivé du logiciel Mozilla (actuellement connu sous le nom de SeaMonkey), mais reprenant uniquement les fonctions de navigation de celui-ci. Ce logiciel multiplateforme est compatible avec diverses versions de Windows, Mac OS et GNU/Linux (incluant Android). [11]

3.4 IHM (interface homme – machine)

Les ingénieurs dans le domaine d'IHM étudient la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes qui soient ergonomiques, efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation.

3.4.1 Définition

L'acronyme IHM peut signifier :

- **Interface homme – machine :** ensemble des dispositifs matériels et logiciels permettant à un utilisateur d'interagir avec un système interactif.
- **interaction homme – machine :** ensemble des aspects de la conception, de l'implémentation et de l'évaluation des systèmes informatiques interactifs.

- IHM est de domaine pluridisciplinaire de par son utilisation dans plusieurs disciplines :

- Informatique.
  - Programmation.
  - synthèse et reconnaissance de parole, langue naturelle.
  - Image.
  - système…
- Psychologie cognitive.
- Ergonomie cognitive, ergonomie des logiciels.
- Sciences de l'éducation, didactique.
- Anthropologie, sociologie, philosophie, linguistique …
- Communication, graphisme, audiovisuel, design.

3.4.2 IHM et programmation

- La plupart des applications informatiques sont interactives.
- L'IHM est souvent un élément clé du logiciel (en + ou -).
- La conception de l'interaction représente plus de 50% du coût de développement.
- L'IHM peut représenter 80% du code d'une application.
  - elle peut être modifiée/reconstruite de multiples fois.
  - importance de l'indépendance interface / cœur du système.

3.4.3 Présentation des interfaces du site

Notre application web contient quatre catégories d'utilisateurs, le simple internaute, L'utilisateur, l'institut, et l'administrateur. En ce qui suit nous présenterons quelques interfaces de l'application.

➢ **Page d'accueil**

C'est l'interface qui s'affiche à tout utilisateur voulant visiter le site de SigDoc, chaque utilisateur doit passer par un système d'authentification sécurisé en saisissant son login et son mot de passe afin d'accéder à son espace utilisateur ou bien Administrateur.si ses informations existent dans la base de données et qu'elles correspondent l'une à l'autre, le système affiche la page convenue , sinon un message d'erreur s'affiche, et aussi elle permet la vérification de la validité d'un document.

FIGURE 14 : Page d'accueil.

FIGURE 15 : Page d'accueil 1.

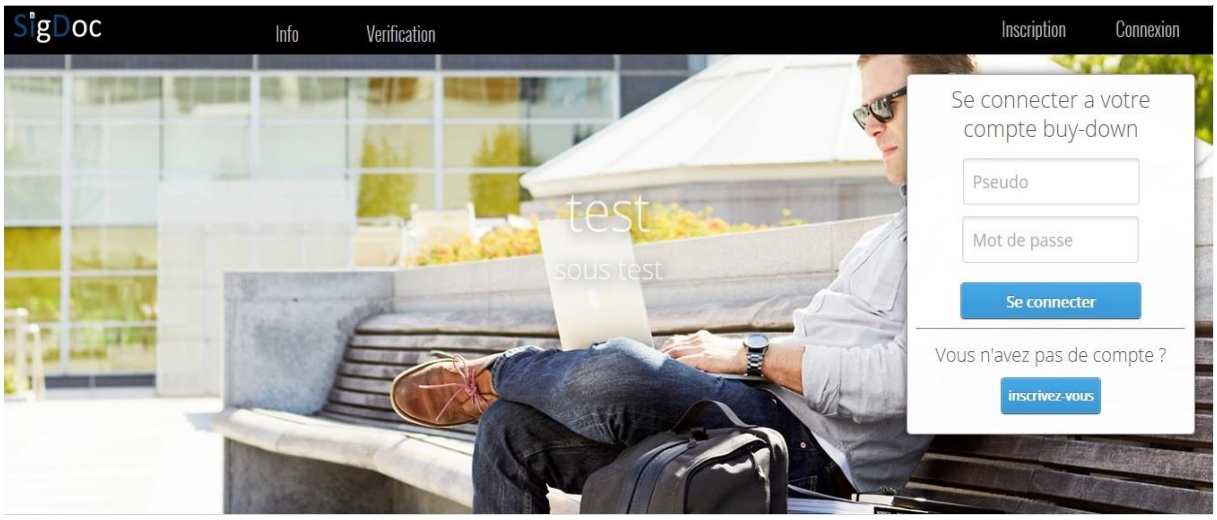


FIGURE 16 : Fenêtre d'authentification.

- **Page inscription :**

Cette interface s'affiche lorsque le visiteur du site choisit de devenir un utilisateur ou bien un institut, alors il n'a qu'à remplir le formulaire qui correspond au type de l'inscription qu'il voudra et confirmer en cliquant sur le bouton s'inscrire.

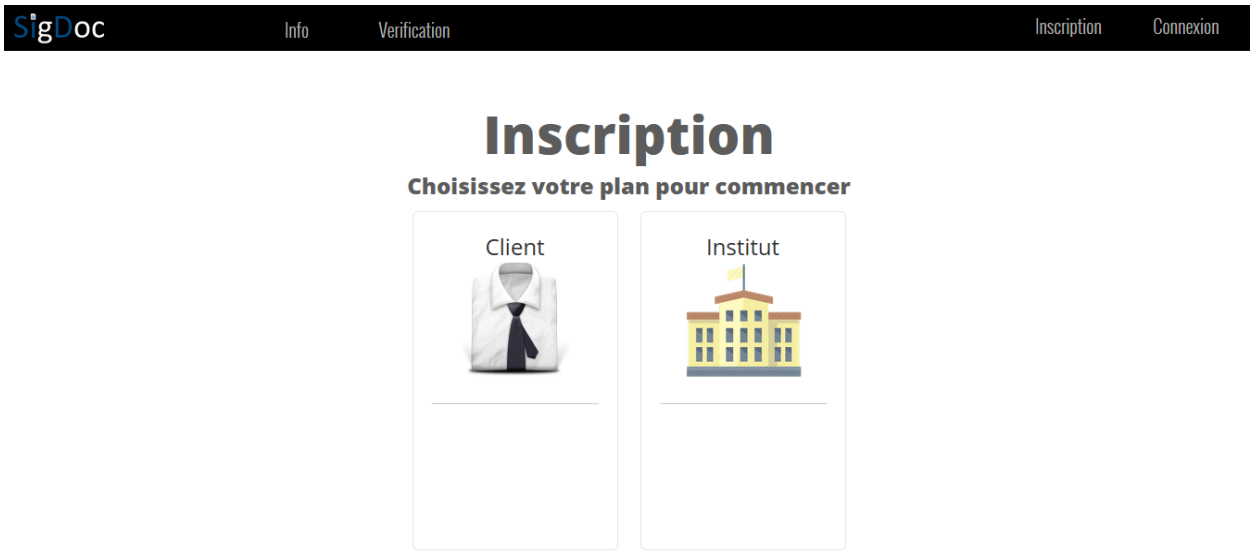


FIGURE 17 : Fenêtre de d'inscription.

➢ **Page d'inscription pour un Institut :**

Cette interface permet de s'inscrire en tant qu'institut en remplissant les champs du formulaire correctement car les champs seront vérifié un par un.

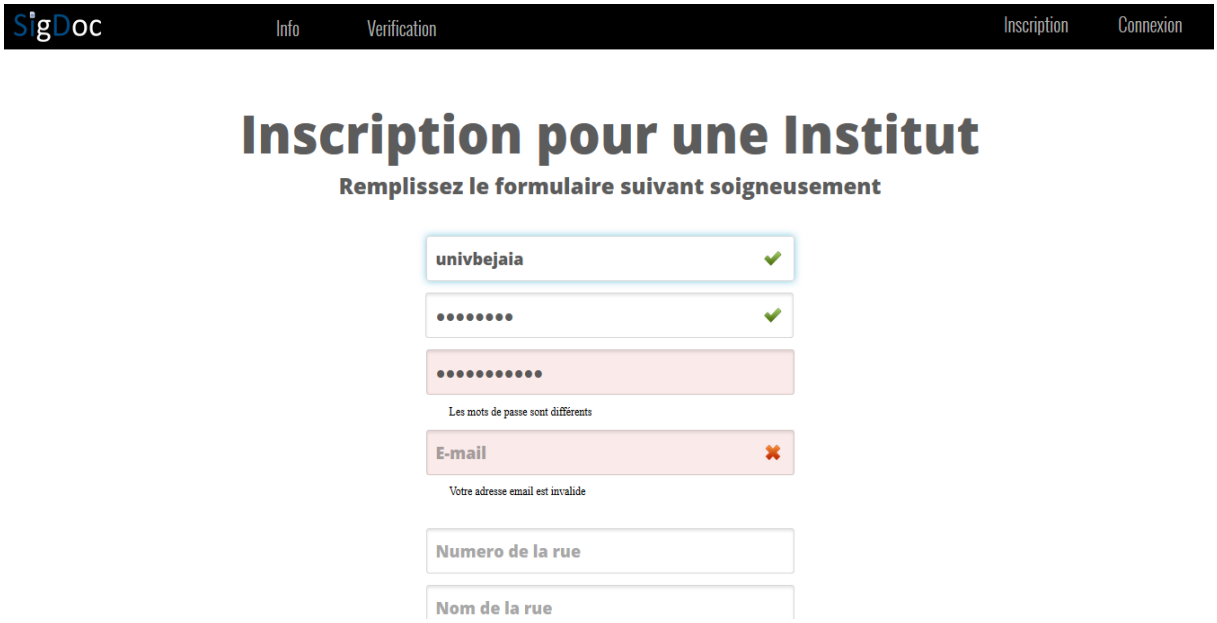


FIGURE 18 : Fenêtre d'inscription pour un institut.

➢ **Page d'inscription pour un client :**

Cette interface permet de s'inscrire en tant qu'un client en remplissant les champs du formulaire correctement.

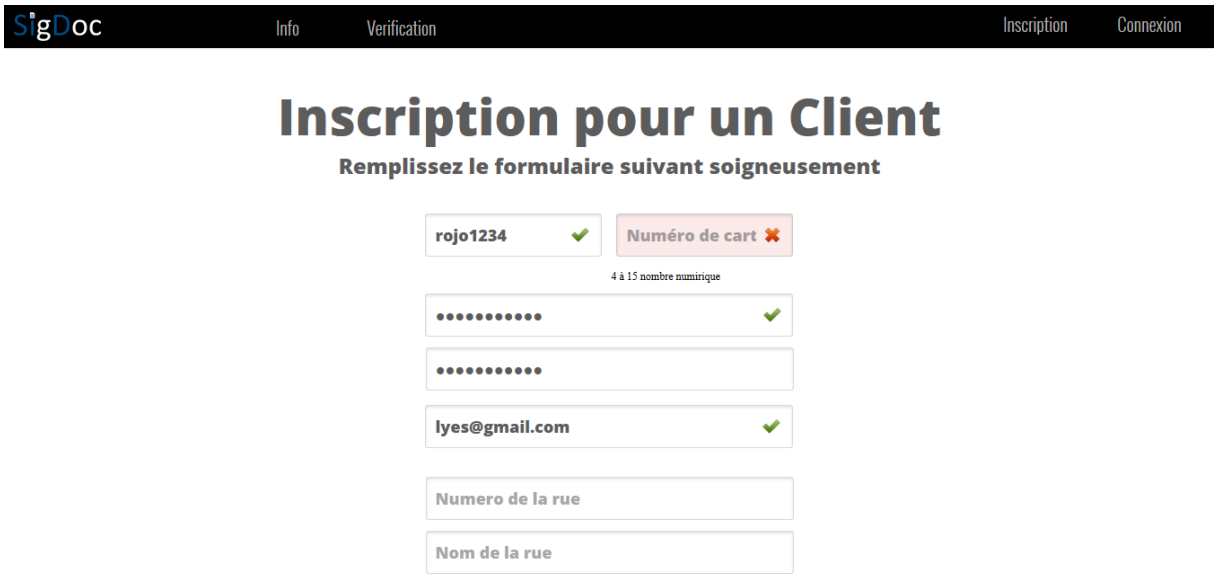


FIGURE 19 : Fenêtre d'inscription pour un client.

➢ **Profil Administrateur :**

Cette interface permet au superviseur d'avoir un contrôle total sur le site, lui permettant ainsi une consultation des demandes reçues et satisfaire les besoins des clients (envoi des documents administratifs).

FIGURE 20 : Profil Administrateur.

➢ **Interface gestion de publicité:**

Cette interface permet le contrôle de la gestion de la publicité de la page d'accueil.

FIGURE 21 : Page de gestion de publicité.

➢ **Profil Institut :**

Cette interface permet à l'institut d'avoir un contrôle total sur ses informations, ainsi que la consultation des demandes reçues et satisfaire les besoins des clients (envoi des documents académiques).

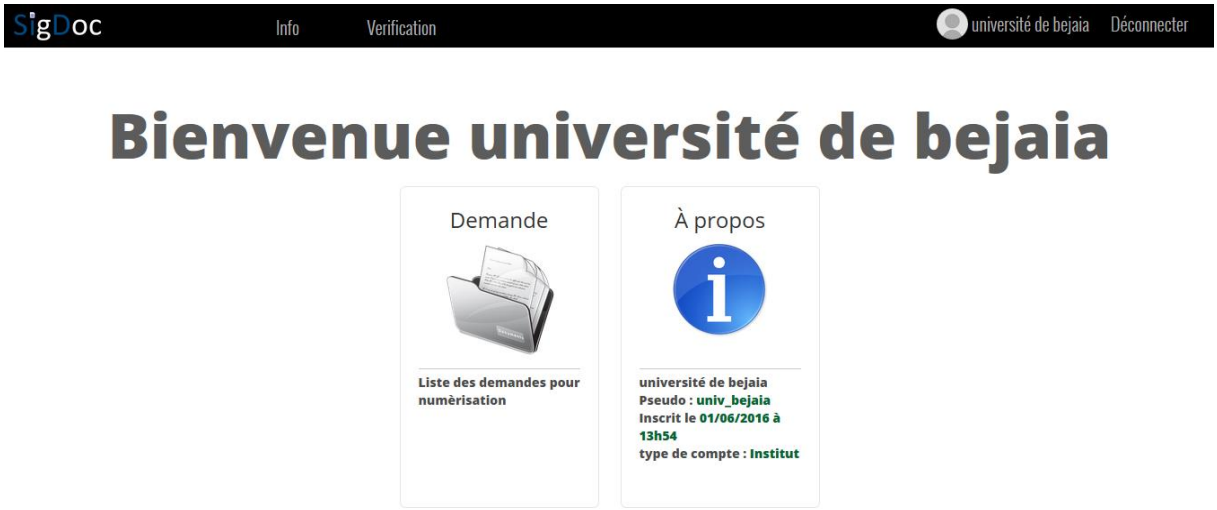


FIGURE 22 : Profil Institut.

➢ **Profil Utilisateur :**

Toute internaute inscrit sur le site aura le droit d'accès à cette interface après authentification bien sûr, cette interface lui permet de vérifier ses informations (nom, prénom, pseudo etc.), et lui permet aussi d'effectuer une demande de document académique ou bien administratif, de visualiser ses documents reçus et les imprimer.

FIGURE 23 : Profil Utilisateur.

- **Interface mes documents:**

Cette interface s'affiche à l'utilisateur, elle lui permet de consulter ses documents académiques et administratifs reçus et authentifier par des instituts, elle lui permet aussi de visualiser imprimer et enregistrer un document au format numérique à tout moment.

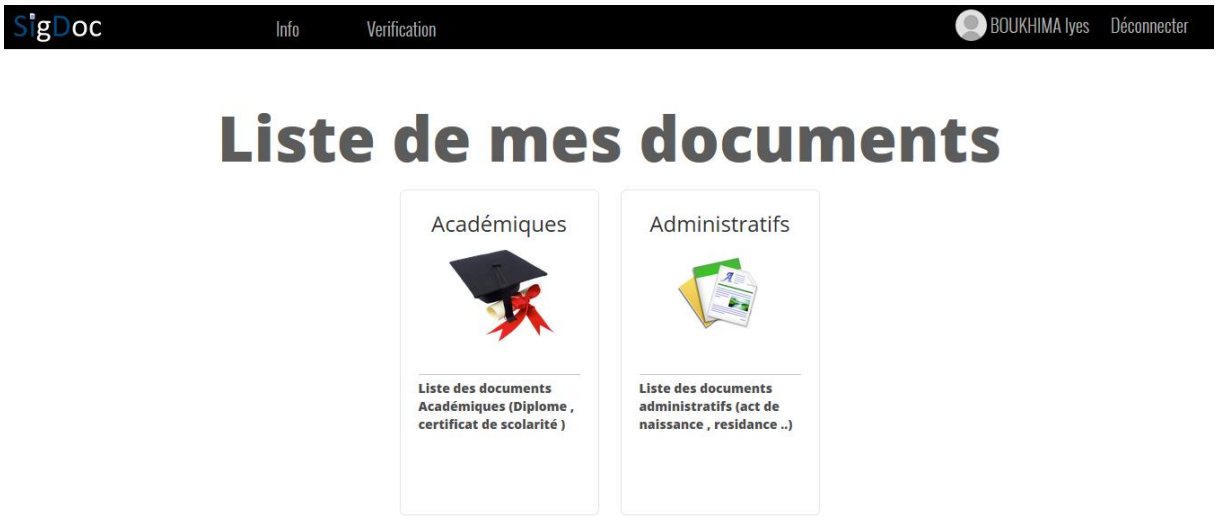


FIGURE 24 : Interface mes documents.

➢ **Document académique signé (Diplôme)**

Après la visualisation de document si l'utilisateur désire l'imprimer. Cette interface lui sera renvoyée, (comme exemple diplôme) contenant les informations de l'utilisateur (nom, prénom, matricule, spécialité.. etc.) et ces information seront affichées sous forme d'un QR code en-dessous de la page à droite, et le document sera signé et affiché sous forme d'un QR code en-dessous de la page à gauche.

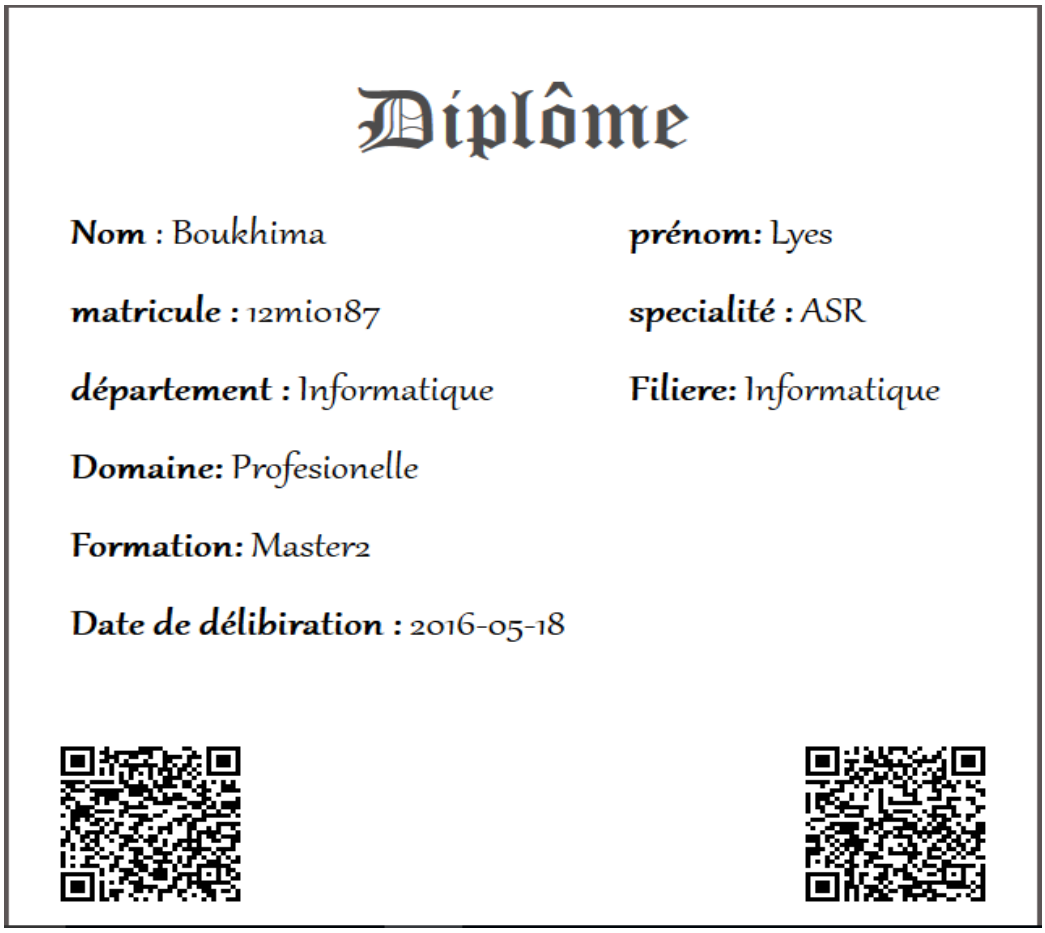


FIGURE 25 : diplôme signé.

➢ **Document administratif signé (Fiche de résidence)**

On prend un autre exemple, un document administratif (fiche de résidence) qui va contenir les informations de l'utilisateur (nom, prénom, wilaya de résidence, numéro de rue.. etc.) et ces information seront aussi affichées sous forme d'un QR code en-dessous de la page à droite, et il sera aussi signé et affiché sous forme d'un QR code au-dessous de la page à gauche.

Fiche de residence

Nom : Boukhima

prénom: Lyes

Degré de parenté : Le pere

wilaya de résidence : Bejaia

NOM De Rue: Imehdeyen

numero de rue: 123254

Preuve fournie: Carte de vote

wilaya de retirement : Bejaia

Date de retirement: 2016-05-11 10:29:13

FIGURE 26 : Fiche de résidence signée.

3.5. La sécurité de SigDoc

La sécurité d'un système informatique fait souvent l'objet de métaphores. En effet, on la compare régulièrement à une chaîne en expliquant que le niveau de sécurité d'un système est caractérisé par le niveau de sécurité du maillon le plus faible. Ainsi, une porte blindée est inutile dans un bâtiment si les fenêtres sont ouvertes sur la rue.

- La sécurité dont dispose notre site « SigDoc» est :

  - Les mots de passe des utilisateurs sont hachés avec l'algorithme MD5 (Message Digest 5) pour la sécurité de leurs comptes SigDoc contre toute attaque.
  - Sécurité contre les injections SQL.
  - Sécurisation des échanges de données avec la protection SSL (Secure Sockets Layer).

Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'un site web.
Dans ce chapitre, nous avons présenté les aspects pratiques liés au site, à savoir le diagramme de déploiement associé à notre système suivi des langages, outils et logiciels de développement nécessaires. En dernier, nous avons illustré les principales interfaces que comprend notre application et les outils assurant la sécurité de SigDoc.

Conclusion Générale

Conclusion Générale

Notre intérêt au domaine de la sécurité et la cryptographie, nous a orienté vers une approche sur les différents moyens de résoudre et apporter des solutions aux problèmes que rencontre tout institut pour assurer l'intimité et l'authentification des documents délivrés. Sur ce chemin, nous nous sommes focalisé sur la signature numérique des documents académiques et administratifs qui ne cessent de subir des falsifications. Notre projet la délivrance de document, apporte une aide considérable au secteur administratif, en ce qui est de gérer au mieux, ainsi que garantir la confidentialité du support numérique le concernant. Ceci dit, notre application web permet à l'utilisateur de bénéficier autant d'un gain de temps, de déplacements que de se voir attribuer un pas de plus vers le monde moderne.

Notre projet a commencé en premier lieu par une étude axée sur les généralités de la sécurité informatique et la cryptographie qui nous ont permis d'approfondir nos connaissances et d'en savoir plus sur ces concepts. Chiffrement RSA, fonction de hachage, signature numérique..etc.

Nous avons établi par la suite, les objectifs de notre application, ainsi que les démarches de développement à suivre durant cette phase. Une étude préliminaire pour identifier les différents acteurs qui interagissent avec le système à réaliser, suivi de la spécification des besoins fonctionnels à travers les diagrammes de cas d'utilisation et de l'analyse des besoins en utilisant les diagrammes de séquence.

La phase de conception suit immédiatement la phase d'analyse, il s'agissait alors d'étendre la représentation effectuée au niveau de l'analyse, en y intégrant les aspects techniques les plus proches des préoccupations des besoins spécifiques. L'élément principal à livrer au terme de cette phase est le diagramme de classe ainsi que le schéma relationnel.

Nous avons ensuite, abordé la réalisation en utilisant les outils d'implémentation appropriés à l'intégration du contenu et le style, la gestion de la base de données (PHP/MySQL,).

Enfin, nous avons pu asseoir nos connaissances à travers la réalisation de notre travail qui se montre fructueux et d'un apport considérable pour nos objectifs ainsi qu'à pallier notre problématique.

Bibliographie

[1] : NADIA BOUNEGTA. Mémoire de Fin d'études. Approche distribuée pour la sécurité d'un réseau de capteur sans fil. Université de Bechar. 2010.

[2] : Doris Baker H.X.Mel. La cryptographie décryptée. CampusPress, 2001.

[3] : Robert Rolland. Formation générale en cryptographie. Institut de Mathématiques de Luminy. Marseille cedex 9. France. 2002.

[5] : Laurent Poinsot. Introduction à la sécurité informatique. Université Paris 13 - Institut Galilée.

[7] : Renaud Dumont. Cryptographie et Sécurité informatique. Université de Liège. Belgique.2010

[8]: Bruce schneier. Cryptographie appliquée, algorithmes, protocoles et code source en C. Vuibert. 1997

[9] : Doris baker H.X.Mel. La cryptographie décryptée. CampusPress, 2001.

[10]: Thomas Peyrin. Thèse doctorat. Analyse de fonctions de hachage cryptographiques. L´Ecole normale supérieure. Paris. Le 3 novembre 2008

[11] : Badja riad,Djabali nacer yacine ,Messoudene Khaled .mémoire de fin de cycle Conception et réalisation d'un site web dynamique dédié aux boutiques en ligne. Université de Bejaïa. 2013.

Webographie

[4] : http://jchambon.fr/Textes/17-COURS_SI_Protection.pdf

[6] : http://www.rederio.br/downloads/pdf/nt00700.pdf

[12]: http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-javascript-509/

[13]: http://www.commentcamarche.net/contents/5-ajax-asynchronous-javascript-and-xml

[14]: http://www.php.net/manual/fr/preface.php

[15]: http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-mysql-4640/

[16]http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-css-4050/

[17]: http://www.techno-science.net/?onglet=glossaire&definition=701

[18]: http://siguillaume.developpez.com

[19]:http://www.01net.com/telecharger/windows/Multimedia/photo_numerique/fiches/33029.html

[20]: http://office.microsoft.com/fr-001/visio/

Remarque: Tous les sites ont été consultés entre le 20/03/2016 et le 02/06/2016.

Annexes

I.1.Les sites web

I.1.1.Définition

Un site web (aussi appelé site internet) est un ensemble cohérent de pages web hyper liées entre elles, conçues pour être consultées avec un navigateur Web, publiées par un propriétaire (une entreprise, une administration, une association, un particulier, etc.).

I.1.2.Les typologies possibles des sites web

Les sites web peuvent être distingués selon différents critères comme suit :

I.1.2.1.Distinction selon le but poursuivi et le contenu

Les Sites catalogue permettent de présenter les produits d'une entreprise et de les mettre en valeur.

Les Sites d'informations sont des sites fournissant des informations particulières à des internautes.

Les Sites institutionnels sont des sites destinés à décrire l'activité d'une organisation, et à donner les informations nécessaires aux clients ou aux bénéficiaires.

Les Sites personnels (parfois pages perso) sont des sites réalisés par des particuliers à titre de loisir, le plus souvent par passion pour un sujet ou une discipline.

Les Sites communautaires sont des sites réunissant des internautes autour d'un intérêt commun.

Les Sites intranet sont des sites accessibles de l'intérieur d'une entreprise ou d'une direction, ayant pour objet la mise à disposition et le partage d'informations professionnelles.

Les Sites vitrine (sites plaquette ou sites identité) sont des sites dont l'objectif est de mettre en avant l'image de marque de la société, en présentant par exemple ses produits ou ses services.

I.1.2.2. Distinction selon les fonctionnalités et les techniques

La Distinction des sites web selon les fonctionnalités et les techniques est représentée ci-dessus :

Le Site statique

Un site web statique est un site où chacune des pages est créée en HTML. Un ordinateur qui se connecte au serveur demande une page, celle-ci lui est directement servie (elle est stockée toute prête sur le serveur).

Site interactif coté client

Un site interactif coté client utilise une technologie Script [JavaScript] qui s'exécute coté client.

Script : Programme informatique qui ne nécessite pas de compilation avant d'être exécuté. Pour fonctionner, les scripts doivent être interprétés par un programme ou un serveur dédié au langage dans lequel ils ont été écrits.

Site interactif coté serveur (site dynamique)

Le Web, ce n'est pas qu'un ensemble de documents HTML statiques ! Mais avec le coté serveur on a la possibilité de permettre à l'utilisateur d'ajouter ou modifier du contenu, de traiter des soumissions de formulaire, d'afficher de manière uniforme l'ensemble des pages d'un site, de proposer des applications interactives …

Site Full Flash

Flash est un système d'animation vectorielle développé par l'éditeur Adobe. Pour pouvoir lire une animation Flash, un plug-in (appelé Player Flash) doit être installé sur le navigateur.

Site dynamique

Même fonctionnement que pour un site interactif coté serveur à part l'existence d'un second serveur de base de données, où le client peut sauvegarder, gérer, modifier des données.

Exemple de langage de gestion des BDD : SQL (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

Site web 2.0

Le Web 2.0 désigne la nouvelle génération de site web ou l'interaction entre les utilisateurs et leur contribution est grandement facilité par l'utilisation de technologies riche comme AJAX. Une page AJAX contient un moteur JavaScript exploitant la classe XML HTTP Request qui permet de récupérer des données sur le serveur en tâche de fond (Asynchrone).

XML HTTP Request est un objet ActiveX ou JavaScript qui permet d'obtenir des données au format XML, JSON, mais aussi HTML, ou encore texte simple à l'aide de requêtes HTTP. On explique le succès récent de l'objet et la très grande utilisation qui en est faite actuellement (parfois au détriment de l'accessibilité d'un site) par la simple création du nom AJAX.

XML (entendez eXtensible Markup Language et traduisez Langage à balises étendu, ou Langage à balises extensible) est en quelques sortes un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises (markup).

JSON (JavaScript Object Notation – Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou généré par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript.

I.2. La sécurité des sites internet

I.2.1. Définition

La sécurité sur Internet est une branche de la sécurité informatique spécifiquement liée à internet, impliquant souvent la sécurité du navigateur, mais aussi la sécurité du réseau à un niveau plus général, car il s'applique à d'autres applications ou systèmes d'exploitation sur un ensemble. Son objectif est d'établir des règles et des mesures visant à être utilisées contre les attaques sur Internet.

L'internet représente un canal non sécurisé pour l'échange d'informations conduisant à un risque élevé d'intrusion ou de fraude, tels que le phishing. Différentes méthodes ont été utilisées pour protéger la transmission de données, y compris le cryptage.

I.2.2. Les failles de sécurité web les plus courantes

Voici quelques-unes des failles de sécurité web les plus courantes :

L'injection : la plus connue est évidemment l'injection SQL, mais elle peut concerner d'autres langages (Shell, LDAP …etc). C'est une des plus vieilles failles connues mais c'est également la plus dangereuse.

Gestion de l'Authentification et de la Session: Est-il possible de voler une session ou récupérer un mot de passe ? C'est un sujet vaste et épineux, et il est plutôt recommandé de se reposer sur des Framework et outils pour cette partie (serveurs d'application, spring security).

Cross-Site Scripting: le fameux XSS ! Il est classé troisième à cause de sa fréquence et sa facilité de mise en œuvre, son exploitation est rarement grave. Il consiste la plupart du temps à faire exécuter du JavaScript non prévu par votre application.

Insecure Direct Object References : Cas classique : l'id de la donnée actuellement visualisée est un paramètre de la requête HTTP. En changeant cet ID, il est possible de consulter toutes les données si aucun contrôle d'accès à la donnée n'est effectué. Cela peut être très problématique sur les données utilisateurs ou pire, sur un site d'une banque !

Security Misconfiguration: Nous utilisons toujours beaucoup d'outils pour nos projets (serveur web, application, base de données, Framework…), combien de fois

laissons nous les valeurs par défaut de la configuration de ces outils ? Il est important d'étudier la configuration de chacun et de désactiver tout ce dont le projet n'a pas besoin.

Sensitive Data Exposure: Protection des données sensibles, les mots de passes, les numéros de carte de paiement, les données santés ou personnels doivent être cryptés et leurs confidentialité doit être maintenues sur toute la chaîne avec l'utilisation du SSL.

Missing Function Level Access Control : Contrôler l'accès aux fonctionnalités (pages) proposées par l'application, un utilisateur lambda ne doit pas pouvoir accéder et utiliser les fonctionnalités d'administration. Généralement un contrôle sur l'url ou sur un paramètre (action par exemple) permet de résoudre ce souci.

Cross-Site Request Forgery (CSRF) : Faille assez complexe, elle consiste à forcer un utilisateur à exécuter une requête à son insu. Par exemple, un frame ou une image d'un site tiers a pour source une URL de votre application. Si cette url est celle correspondant à une demande de virement bancaire et que l'utilisateur est déjà connecté au sein du même navigateur, la requête est lancée et interprétée par vote application en visitant simplement le site tiers.

Using Components with Known Vulnerabilities : Semi nouveauté de cette mise à jour du top 10, elle est un peu liés à la 5, OWASP met en garde sur l'utilisation de composants tiers dans nos développements, notamment les composants open source. Il est important de se documenter sur les failles connues des outils que nous utilisons, même CXF ou Spring ont eu leur failles critiques, il faut rester à jour.

Unvalidated Redirects and Forwards : Faille classique et simple à corriger, nous avons souvent besoin d'implémenter des redirect ou des forward génériques dans nos applications. Il faut vérifier la destination de ces redirections avant de les effectuer, ne pas accepter la redirection n'allant pas sur votre site, Le mieux étant de ne pas utiliser les redirections.

I.3.les fonctions de hachage les plus connues

MD4

Inventée par Rivest en1990 pour les laboratoires RSA, MD4 est la plus ancienne fonction de la famille MD-SHA, c'est donc naturellement aussi la plus simple. La fonction de compression produit des hachés de 128 bits.

MD5

En réponse aux attaques sur des versions réduites de MD4, une nouvelle version plus complexe fut créée. Comparée à celle de MD4, la fonction de compression de MD5 possède un tour de plus, de nouvelles fonctions booléennes, une diffusion accrue dans la fonction d'étape, des constantes définies pour chaque étape, etc. Les paramètres généraux restent inchangés : des hachés de taille de 128 bits

RIPEMD-0

RIPEMD-0 est l'une des primitives recommandées en 1992 à l'issue d'une étude d'un consortium dans le cadre du projet européen RACE Integrity Primitives Evaluation (RIPE) sur les primitives permettant de garantir l'intégrité Originalement nommée RIPEMD, la fonction de compression se compose de deux branches parallèles, chacune quasiment identique à la fonction de compression de MD4. Les deux lignes parallèles de calcul ne diffèrent que par l'emploi de constantes différentes. Les paramètres de chaque branche sont donc égaux à ceux de MD4, mais l'ordre d'introduction des mots du bloc de message étendu et les longueurs de rotation lors des étapes sont différentes de ceux de MD4. Les hachés sont de taille de 128 bits.

RIPEMD-128

En 1996, Hans Dobbertin, Antoon Bosselaers et Bart Preneel proposèrent une versionrenforcéedeRIPEMD-0 pour contrer les premières cryptanalyses de MD4 et de RIPEMD-0 qui apparaissaient. De plus, une version 256 bits a aussi été définie, mais pour une sécurité équivalant à une fonction de 128 bits

RIPEMD-160

RIPEMD-160 est une fonction de hachage de 160 bits qui fut publiée en même temps que RIPEMD-128 et qui représente une version plus robuste en raison de sa taille de sortie plus grande, et aussi grâce à sa fonction de compression un peu plus complexe : un tour et un registre interne par branche sont rajoutés. Comme pour RIPEMD-128, une version doublant la taille de sortie est définie, permettant des hachés de 320 bits pour une sécurité de 160 bits

SHA-0

Publié en 1993, SHA-0 est le premier membre de la famille Secure Hash Standard, les fonctions de hachage standardisées par le NIST. Très inspirée de celles de la famille MD, la fonction de compression de SHA-0 n'en diffère quasiment que par l'utilisation d'une expansion de message novatrice : au lieu d'utiliser des permutations des mots de message pour chaque tour, les mots de message étendu sont obtenus par un procédé récursif initialisé par les mots du message d'entrée. Pour permettre une longévité suffisante de l'algorithme quant à l'augmentation de la puissance de calcul, SHA-0 produit des hachés de 160 bits.

SHA-256

Publiée en 2002, SHA-256 fait partie des derniers membres en date de la famille MD-SHA. Outre sa taille de sortie, elle contient plusieurs nouveautés par rapport à ses prédécesseurs. Par exemple, l'expansion de message est beaucoup plus complexe et corrige les précédentes erreurs de SHA-0 ou SHA-1 Comme son nom l'indique, SHA-256 produit des hachés de 256 bits, mais il existe aussi une version 224 bits introduite 2 ans plus tard.

SHA-512

SHA-512 fut publiée en même temps que SHA-256 et représente son équivalent pour les processeurs 64 bits, qui vont progressivement remplacer ceux de 32 bits dans les ordinateurs. Les mots traités seront donc de taille 64 bits pour profiter pleinement de cette nouvelle architecture. Les autres différences par rapport à SHA-256 concernent la taille de sortie, qui est doublée pour obtenir une fonction viable sur le très long terme, et le nombre d'étapes qui est augmenté. Ainsi, SHA-512 produit des hachés de 512 bits, mais une version 384 bits fut aussi introduite en même temps.

I.4. Le chiffrement de RSA

Toutes les opérations du crypto-système RSA se passe dans un ensemble de nombre entiers. Soient p et q deux nombres premiers assez grands. On note N = pq. Le nombre N est appelé module RSA. Supposons que deux personnes A et B veulent communiquer de façon sûre en utilisant le crypto-système RSA. Pour cela, ils doivent, chacun de son côté préparer un module RSA, deux clés e et d, exécuter une procédure de chèrement et de signature et une procédure de déchirement et de vérification de la signature.
L'algorithme se déroule comme suit :

- Choisir deux grands nombre premiers p et q :
  $n=p\times q$
  $\phi(n)=(p-1)\times(q-1)$
- Choisir $e < \phi(n)$ tel que $pgcd(e, \phi(n))=1$
- Calculer d tel que $e\times d \bmod \phi(n)=1$
  Clé publique $\Rightarrow$ (e, n)
  Clé privée $\Rightarrow$ (d, n)
- Chiffrement $\Rightarrow C=M^e \bmod n$
- Déchiffrement $\Rightarrow M=C^d \bmod n$

I.5. Présentation générale d'UML

Le génie logiciel et la méthodologie s'efforcent de couvrir tous les aspects de la vie du logiciel. Issus de l'expérience des développeurs, concepteurs et chefs de projets, ils sont en constante évolution, parallèlement à l'évolution des techniques informatiques et du savoir-faire des équipes.

Comme toutes les tentatives de mise à plat d'une expérience et d'un savoir-faire, les méthodologies ont parfois souffert d'une formalisation excessive, imposant aux développeurs des contraintes parfois contre-productives sur leur façon de travailler. Avec la mise en commun de l'expérience et la maturation des savoir-faire, on voit se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes.

UML, qui se veut un instrument de capitalisation des savoir-faire puisqu'il propose un langage qui soit commun à tous les experts du logiciel, va dans le sens de cet assouplissement des contraintes méthodologiques.

- UML signifie Unified Modeling Language. La justification de chacun de ces mots nous servira de fil conducteur pour cette présentation.

I.5.1. Unified : historique des méthodes de conception

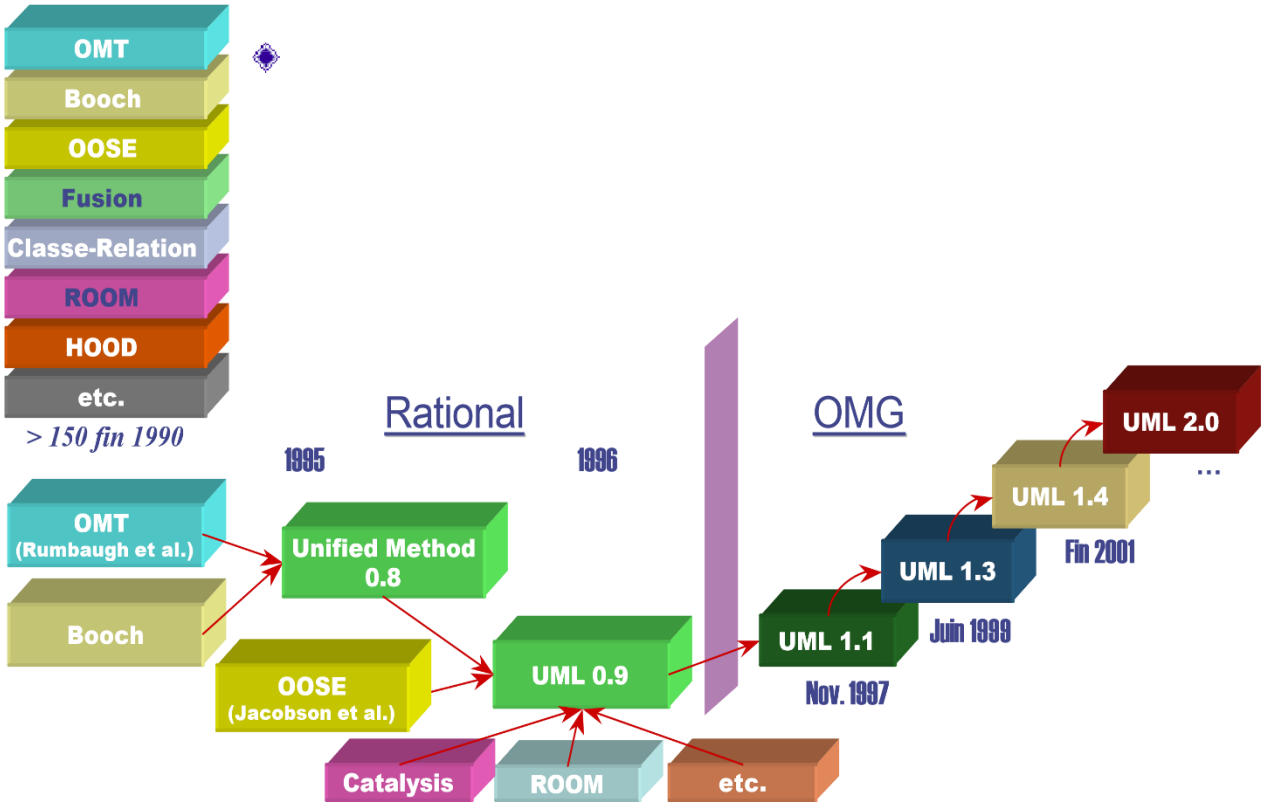


FIGURE 1- Historique de la constitution d'UML.

À chacune des différentes phases de la conception d'un logiciel correspondent des problèmes ou des contraintes différentes. Naturellement, ces niveaux ont fait l'objet de recherches méthodologiques considérables depuis les années 80. Il en résulte que de nombreuses méthodes de développement ou d'analyse de logiciel ont vu le jour, chacune plus ou moins spécialisée ou adaptée à une démarche particulière, voire à un secteur industriel particulier (bases de données, matériel embarqué, ...). Celles-ci ayant été développées indépendamment les unes des autres, elles sont souvent partiellement redondantes ou incompatibles entre elles lorsqu'elles font appel à des notations ou des terminologies différentes, voire à des faux amis.

De plus, à chaque méthode correspond un ou plusieurs moyens (plus ou moins formel) de représentation des résultats. Celui-ci peut être graphique (diagramme synoptique, plan physique d'un réseau, organigramme) ou textuel (expression d'un besoin en langage naturel, jusqu'au listing du code source). Dans les années 90, un certain nombre de méthodes orientées objets ont émergé, en particulier les méthodes :

– OMT de James RUMBAUGH

– BOOCH de Grady BOOCH

– OOSE (Object Oriented Software Engineering) de Ivar JACOBSON à qui l'on doit les Use cases).

En 1994, on recensait plus de 50 méthodologies orientées objets. C'est dans le but de remédier à cette dispersion que les « poids-lourds » de la méthodologie orientée objets ont entrepris de se regrouper autour d'un standard.

En octobre 1994, Grady Booch et James Rumbaugh se sont réunis au sein de la société RATIONAL dans le but de travailler à l'élaboration d'une méthode commune qui intègre les avantages de l'ensemble des méthodes reconnues, en corrigeant les défauts et en comblant les déficits. Lors de OOPSLA'95 (Object Oriented Programming Systems, Languages and Applications), la grande conférence de la programmation orientée objets), ils présentent UNIFIED METHOD V0.8. En 1996, Ivar Jacobson les rejoint.

Leurs travaux ne visent plus à constituer une méthodologie, mais un langage.

Leur initiative a été soutenue par de nombreuses sociétés, que ce soit des sociétés de développement (Microsoft, Oracle, Hewlet-Packard, IBM – qui a apporté son langage de contraintes OCL –, ...) ou des sociétés de conception d'ateliers logiciels. Un projet a été déposé en janvier 1997 à l'OMG 3 en vue de la normalisation d'un langage de modélisation. Après amendement, celui-ci a été accepté en novembre 97 par l'OMG sous la référence UML-1.1. La version UML-2.0 est annoncée pour la fin 2004.

I.5.2 Modeling : analyse et conception

Une bonne méthodologie de réalisation de logiciels suppose une bonne maîtrise de la distinction entre l'analyse et la conception, distinction que nous exposons dans le polycopié complémentaire. Le lecteur verra qu'en pratique, le respect d'une distinction entre des phases d'analyse et de conception rigoureusement indépendantes n'est pas tenable, mais il est important d'avoir en tête la différence lorsqu'on s'apprête à réaliser un logiciel. Encore une fois, il est important de garder à l'esprit qu'UML n'offre pas une méthodologie pour l'analyse et la conception, mais un langage qui permet d'exprimer le résultat de ces phases.

Du point de vue des notations employées en UML, les différences entre l'analyse et la conception se traduisent avant tout par des différences de niveau de détail dans les diagrammes utilisés. On peut ainsi noter les différences suivantes :

- Dans un diagramme de classes d'analyse, les seules classes qui apparaissent servent à décrire des objets concrets du domaine modélisé. Dans un diagramme de classes de conception, par opposition, on trouve aussi toutes les classes utilitaires destinées à assurer le fonctionnement du logiciel.
- Dans un diagramme de classes d'analyse, on peut se contenter de faire apparaître juste la dénomination des classes, avec parfois le nom de quelques attributs et méthodes quand ceux-ci découlent naturellement du domaine modélisé.
- Dans un diagramme de classes de conception, par opposition, tous les attributs et toutes les méthodes doivent apparaître de façon détaillée, avec tous les types de paramètres et les types de retour.
- Dans un diagramme de séquence d'analyse, les communications entre les principaux objets sont écrites sous forme textuelle, sans se soucier de la forme que prendront ces échanges lors de la réalisation du logiciel. Dans un diagramme de séquence de conception, par opposition, les échanges entre classes figurent sous la forme d'appels de méthodes dont les signatures sont totalement explicitées.

Les étapes permettant de passer de diagrammes d'analyse à des diagrammes de conception et les motivations de la formalisation progressive que cela entraîne sont traitées dans le polycopié complémentaire.

I.5.3 Language : méthodologie ou langage de modélisation ?

Il est important de bien faire la distinction entre une méthode qui est une démarche d'organisation et de conception en vue de résoudre un problème informatique, et le formalisme dont elle peut user pour exprimer le résultat.

Les grandes entreprises ont souvent leurs propres méthodes de conception ou de réalisation de projets informatiques. Celles-ci sont liées à des raisons historiques, d'organisation administrative interne ou encore à d'autres contraintes d'environnement (défense nationale, …) et il n'est pas facile d'en changer. Il n'était donc pas réaliste de tenter de standardiser une méthodologie de conception au niveau mondial.

UML n'est pas une méthode, mais un langage. Il peut donc être utilisé sans remettre en cause les procédés habituels de conception de l'entreprise et, en particulier, les méthodes plus anciennes telles que celle proposée par OMT sont tout à fait utilisables.

D'ailleurs, la société RATIONAL (principale actrice de UML) propose son propre processus de conception appelé OBJECTORY et entièrement basé sur UML.

Ainsi, UML facilite la communication entre clients et concepteurs, ainsi qu'entre équipes de concepteurs. De plus, sa sémantique étant formellement définie 4 dans [JBR97b] (sous forme de diagramme UML), cela accélère le développement des outils graphiques d'atelier de génie logiciel permettant ainsi d'aller de la spécification (haut niveau) en UML vers la génération de code (JAVA, C++, ADA, …). De plus, cela autorise l'échange électronique de documents qui deviennent des spécifications exécutables en UML.

UML ne se contente pas d'homogénéiser des formalismes existants, mais apporte également un certain nombre de nouveautés telles que la modélisation d'architectures distribuées ou la modélisation d'applications temps-réel avec gestion des multitâches, dont l'exposé dépasse le cadre de ce document.

Résumé

Ce mémoire de fin d'études ayant pour thème conception et réalisation d'une application web pour la délivrance et l'authentification des documents académiques et administratifs qui repose sur les techniques de la cryptographie. Ce système est basé à la fois sur l'algorithme RSA, la fonction de hachage et la signature numérique.

La conception de ce système est réalisée en utiliser UML, ce que nous a permis de bien modéliser les activités de l'utilisateur et les activités du système.

Pour la réalisation de l'application nous avons utilisé des différents outils comme aptana studio et l'environnement Appache/MySQL/PHP.

Mots clés: signature numérique, sécurité, Cryptographie, fonction de hachage, UML, UP.

Abstract

This research paper which has as a topic Conception and realization of a Web application for the issuing and the authentication of academic and administrative documents, lies on cryptography's technics. This system is based, at the same time, both on RSA algorithm, hash function and digital signature. The conception of this system in question is executed using UML, which allowed us to model swiftly the user's activities and the system's ones as well.

Concerning the realization of the application, we have used different tools such as aptana studio and Appache environment /MySQL/PHP.

Key words: digital signature, security, cryptography, hash function, UML, UP.