République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Fin d'études

En vu de l'obtention du diplôme Master professionnel en Informatique

Option : Génie Logiciel

Thème

Traçabilité des absences par la Blockchain Cas d'étude : Lycée Chouhadaa Annani de Béjaia

Présenté par :

AZOU Cherifa & BAZIZENE Ilham

Devant le jury composé de :

Présidente :	Dr BACHIRI Lina	M.C.A U.A/Mira Béjaia
Examinateur :	M. OUZEGGANE Redouane	M.A.A U.A/Mira Béjaia
Promotrice :	Dr HAMZA Lamia	M.C.A U.A/Mira Béjaia
Invitée :	Mlle BAIR Narimane	Doctorante U.A/Mira Béjaia

Année Universitaire : 2022/2023

Remerciements

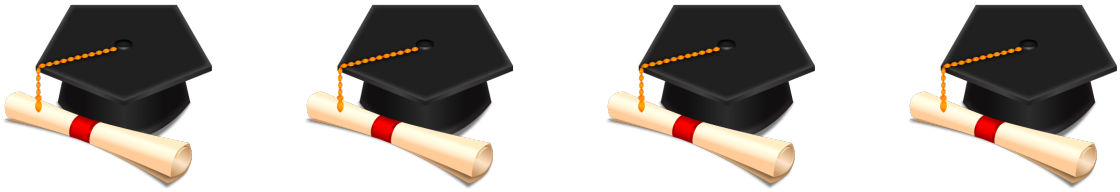
Avant tout, nous remerciant Dieu, pour nous avoir accordé la santé, la volonté et la patience nécessaires pour mener à bien notre formation de master et accomplir ce travail.

Nous désirons adresser nos remerciements sincères à notre promotrice, Mme HAMZA Lamia, pour la confiance qu'elle nous a témoignée tout au long de cette période de travail. Ses précieux conseils, son assistance inestimable et sa patience ont été d'une valeur inestimable pour nous.

Nous sommes extrêmement honorées par la participation du madame BACHRI lyna et monsieur OUZEGGANE Redouane en tant que membres du jury de soutenance, ainsi que madame BAIR Narimane en tant qu'invité. Nous leur sommes profondément reconnaissantes d'avoir accepté d'évaluer notre travail avec leur expertise.

Nous tenons à exprimer notre gratitude envers l'ensemble de nos enseignants et enseignantes, qui ont enrichi notre parcours à l'université de Béjaia de leurs connaissances et de leur dévouement.

Un grand merci est également adressé à nos familles pour leur soutien moral et financier indéfectible ainsi que pour les sacrifices qu'elles ont consentis. Votre appui a été une source d'inspiration et de motivation tout au long de ce parcours.



Dédicace

C'est avec une grande modestie et un immense plaisir que je dédie ce travail :

A mes très chers parents ***Kamal*** *et* ***Ourdia*** *Aucun mot au monde ne peut exprimer l'amour immense que je vous porte ni rendre compte de la gratitude profonde que j'éprouve pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon éducation et mon bien-être. J'espère avoir été à la hauteur des espoirs que vous avez placés en moi. Ce modeste travail est ma manière de vous rendre hommage, une marque de ma reconnaissance éternelle et de mon amour infini.*

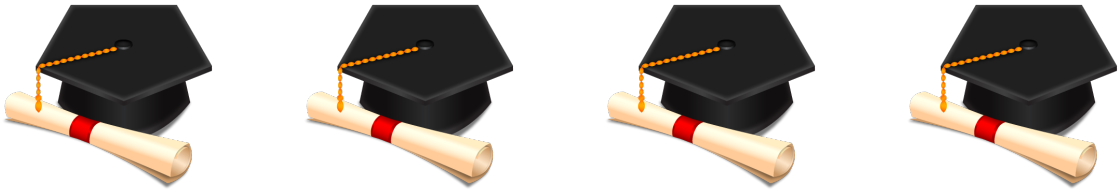
A mon cher frère ***Alilou*** *pour son soutien moral et qui j'aime énormément.*

A mes adorables grands-parents ***yemma aazouzou*** *et* ***jedi*** *et à mes chers oncles et tantes qui m'ont permis de grandir et d'évoluer depuis ma petite enfance dans un climat d'amour et de joie.*

A Mon cher binome ***Ilham*** *complice de réussite, et d'avoir fait de cette aventure une expérience inoubliable..*

À mes chers amis exceptionnels et proches de mon cœur, en particulier Dalila, Manal et Tiziri avec qui j'ai partagé des moments inoubliables et qui sont ma source de bonheur et de joie dans ma vie.

Cherifa



Dédicace

C'est avec une grande modestie et un immense plaisir que je dédie ce travail :

A mes très chers parents ***Abdelhak*** *et* ***Ouahiba*** *Aucun mot au monde ne peut exprimer l'amour immense que je vous porte ni rendre compte de la gratitude profonde que j'éprouve pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon éducation et mon bien-être. J'espère avoir été à la hauteur des espoirs que vous avez placés en moi. Ce modeste travail est ma manière de vous rendre hommage, une marque de ma reconnaissance éternelle et de mon amour infini.*

Mes tendres sœurs ***Narimane*** *et* ***Chaima*** *ainsi que mon beau-frère* ***Babi*** *méritent également ma gratitude pour leur soutien moral constant.*

A Mon cher binome ***Cherifa*** *complice de réussite, et d'avoir fait de cette aventure une expérience inoubliable..*

Je souhaite également adresser mes remerciements à mes amis exceptionnels, ancrés profondément dans mon cœur, en particulier mon ami ***Ahmed****, avec qui j'ai partagé des moments inoubliables et qui sont une source inestimable de bonheur dans ma vie.*

Ilham

Table des matières

Liste des tableaux

Table des figures

Liste des abréviations

2TUP *2 Track Unified Process*

UP *Unified Process*

UML *Unified Modeling Language*

UDS *unité de pistage*

XML *Extensible Markup Languag*

XP *Extreme Programming*

P2P *Réseau paire-à-paire*

SHA *chiffrement asymétrique*

OMG *Objet Management Group*

OO *Orienté Objet*

BDD *Base De Données*

HTML *HyperText Markup Language*

CSS *Cascading Style Sheets*

PHP *Hypertext Preprocessor*

Introduction générale

Avec l'avènement des technologies, la révolution numérique a profondément transformé la façon dont nous traitons et transmettons l'information. L'omniprésence d'Internet a permis aux entreprises et aux établissements de partager leurs systèmes d'information avec leurs partenaires et fournisseurs, une avancée significative pour la collaboration. Cependant, cette ouverture comporte des défis majeurs, notamment la nécessité de maîtriser le contrôle d'accès et de garantir les droits des utilisateurs de ces systèmes d'information.

C'est à ce stade que la Blockchain entre en scène en tant que solution innovante. Elle offre une garantie d'intégrité, de confidentialité, de disponibilité, de non-répudiation et d'authenticité inégalée pour les informations qu'elle héberge. Son utilisation au sein du secteur de l'éducation revêt une importance particulière, car elle peut résoudre un problème essentiel au sein des établissements scolaires : la gestion des absences.

La motivation pour travailler avec la Blockchain dans le contexte d'un lycée est claire. Enregistrer les absences des élèves sous forme de transactions dans des blocs enchaînés, créant ainsi une chaîne de blocs (blockchain), offre un niveau de sécurité et d'intégrité des informations exceptionnel. Cela garantit que les données relatives aux absences des élèves restent immuables et inviolables, tout en facilitant leur accès et leur partage lorsque nécessaire.

Dans les sections à venir, nous explorerons en détail les différents aspects de notre projet, en mettant l'accent sur l'intégration de la technologie blockchain dans le domaine de l'éducation et ses implications potentielles pour la gestion des absences au sein du lycée. Dans ce qui suit nous allons présenté les chapitres suivants :

- Le chapitre 1 : introduction de la technologie de la blockchain dont nous présentons sa définition son historique ainsi que ses concepts de base, ses types et ses domaines d'application y compris notre domaine d'application qui est l'éducation.

- Le chapitre 2 : Blockchain dans les institutions éducatives, dans ce chapitre nous allons présenter notre problématique ainsi que sa solution, en expliquant comment intégrer la blockchain dans le domaine de l'éducation.

- Le chapitre 3 : Analyse et conception, ce chapitre est dédié a la spécification et l'analyse des besoins.

- Le chapitre 4 : Implémentation, ce dernier chapitre porte sur la description des outils et technologies utilisés et la présentation des interfaces réalisées.

Nous terminons ce mémoire par une conclusion générale et quelques perspectives qui peuvent aider à améliorer le système dans le futur.

Chapitre 1

Introduction à la technologie blockchain

1.1 Introduction

Depuis des années les transactions monétaires entre deux entités nécessitent un organisme tiers qui prend le contrôle de toutes les données et informations nécessaires pour ces transactions. la création et le développement de la blockchain nous permet de résoudre cette problématique de gestion centralisé en éliminant l'organisme tiérs. Dans ce présent chapitre nous allons présenté les notions fondamentales de la blockchain : Origine de cette technologie et son évolution, sa structure globale et ces fonctionnalités, ses opérations de base et ses domaines d'application les plus reconnus.

1.2 Définition de la blockchain

La blockchain est une technologie de grande envergure similaire à Internet. Elle permet le stockage décentralisé, sécurisé et immuable des données numériques. Il s'agit d'un registre transparent consultable par tous, mais où les entrées précédentes ne peuvent jamais être modifiées. Ce registre est composé de blocs contenant des centaines de transactions qui s'ajoutent les unes aux autres, formant ainsi une chaîne de blocs, d'où le terme "blockchain". Les transactions peuvent être variées, allant de l'échange d'actifs au vote lors d'un scrutin électoral en passant par la conclusion de contrats numériques. Pour garantir ce processus, des individus ou des entreprises, appelés mineurs, mettent à disposition la puissance de calcul de leurs ordinateurs pour effectuer les calculs nécessaires afin de vérifier la validité de toutes les transactions enregistrées dans la blockchain [1].

1.3 Historique

La technologie blockchain a fait une grande sensation lors de sa création. C'est une nouvelle innovation qui a touché presque tous les secteurs. La notion de la blockchain a commencé au début des années 1990.
-En 1991 : les chercheurs Stuart Haber et W. Scott Stornetta ont proposé une solution calculable afin que personne ne pouvait altérer l'horodatage des documents. Leur système a utilisé une

blockchain cryptée et sécurisée pour stocker les documents horodatés.
- En 1992 le protocole “arbres Merkle“ a pu améliorer l’efficacité du système, et ainsi en permet de collecter plusieurs documents dans un seul bloc, cependant cette technologie tomba dans l’oubli.
-En 1995 : le NY Times met en place la première blockchain dans le journal, cette technologie qui est toujours active, est la plus longue blockchain de l’historique.
-En 2005 : Nick Szabo a développé les concepts de base de cryptomonnaies et lance le projet BitGold, le précurseur de Bitcoin.
-En fin 2008 : la première blockchain est apparue, avec un livre blanc du Bitcoin qui expose les objectifs de la création de cette monnaie électronique, développée par un inconnu sous le pseudonyme de Satoshi Nakamoto.
-En 2009 le logiciel Bitcoin a été mis à la disposition du public, il a été un logiciel open source.
-En 2013 : Vitalik Buterin, programmeur et fondateur du magazine Bitcoin, a fondé Ethereum qui est à la fois une monnaie cryptographique et une plate-forme applicative distribuée, il lança en 2015 Ethereum comme une deuxième blockchain publique, qui peut enregistrer des contrats, des emprunts, etc. La figure 1.1 schématise cette évolution [2, 11].

1.4 Concepts de base

Dans cette section nous présentons les concepts de base da la blockchain.

1.4.1 Réseau paire-a-paire

En anglais peer to peer, souvent abrégé P2P c’est un modèle d’échange en réseau, où chaque entité est à la fois client et serveur, contrairement au modèle client/serveur. Un système pair à pair peut être soit partiellement centralisé, dans ce cas une partie de l’échange passe par un serveur intermédiaire, ou complètement décentraliser, c’est-à-dire les nœuds de ce réseau interconnectent et partagent les ressources entre eux sans avoir recours à un système administratif centralisé [19] comme illustrer dans la figure 1.2.

1.4.2 Consensus

Le consensus est un algorithme très puissant, qui consiste à élaborer un accord au sein d’un groupe de nœuds d’un réseau P2P [3]. L’un des modèle d’algorithme de consensus est la preuve de travail ou prof-of-work en anglais, qui consiste à résoudre le problème de comment savoir que les informations enregistrées n’ont pas été modifiées en interne ou en externe. Ce modèle est utilisé par Bitcoin pour négocier la valeur de son token (jeton) entre les membres de son réseau [8].

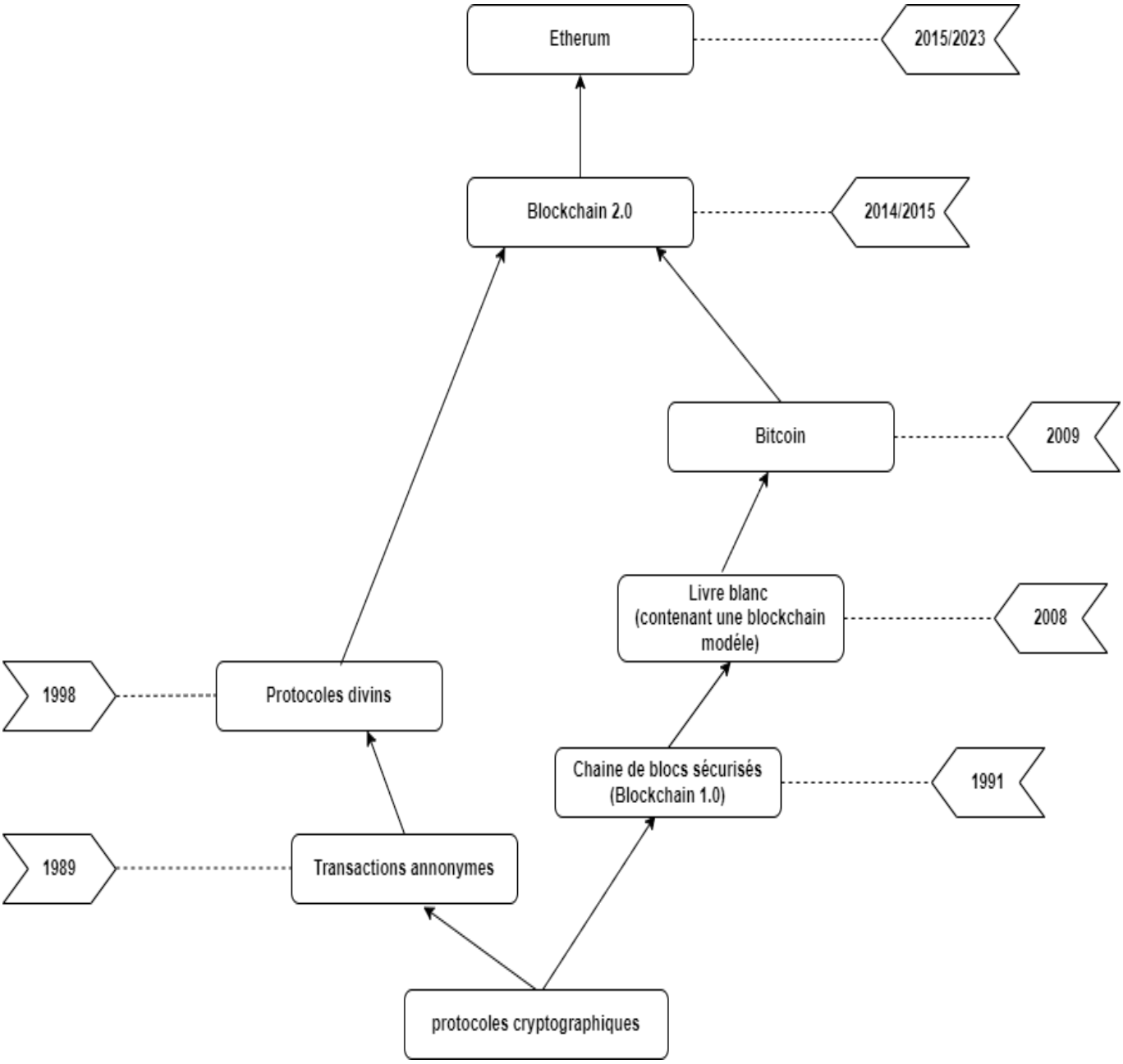


FIGURE 1.1 – schéma illustrant l'évolution de la blockchain en fonction du temps [2, 11].

1.4.3 Mineur

Les mineurs permettent de travailler en groupe et parallèlement pour trouver la preuve de travail et ils se concurrencent pour revendiquer le droit de création d'un nouveau bloc et de sa validation. Ils permettent aussi de vérifier les transactions et les diffuser. Les mineurs assurent aussi la diffusion du nouveau bloc et la confirmation des transactions [3].

- Ajout de blocs (minage) : Les mineurs sont toujours en concurrence pour créer un nouveau

bloc et obtenir la récompense. Le minage est une opération qui implique la résolution de problèmes mathématiques complexes par les mineurs afin de valider les transactions et de créer un nouveau bloc. Une fois qu'un mineur ait résolu le problème, l'annonce sera diffusée sur le réseau et le bloc aussi sera diffusé à tous les nœuds du réseau, et l'autre mineur va vérifier le nouveau bloc. S'ils parviennent à un consensus pour ajouter un nouveau bloc à la chaîne. Celui-ci sera ajouté à leur copie de la blockchain. Ensuite un nouvel ensemble de transactions sera enregistré et confirmé [11].

- Validation des transactions : les mineurs doivent valider les transactions en effectuant des algorithmes de consensus pour s'assurer que toutes les transactions sont valides et conformes aux règles de la blockchain et qu'elles n'ont pas été modifiées. Une fois qu'une transaction est validée, elle est ajoutée à un bloc et ce bloc est ajouté à la blockchain [2].

- Minage : le minage est le processus par lequel les mineurs sont responsables de la création de nouveaux blocs dans la chaîne de blocs et ils sont récompensés pour leur travail de validation des transactions. Les mineurs sont chargés de résoudre des calculs complexes qui nécessitent beaucoup de puissance afin d'ajouter de nouveaux blocs à la chaîne [28].

- Vérification de l'intégrité de la chaîne de blocs : la vérification de l'intégrité de la chaîne de blocs est le processus par lequel les nœuds du réseau s'assurent que la chaîne de blocs est cohérente et qu'il n'y a pas de blocs manquants ou de transactions invalides. Les nœuds vérifient la chaîne de blocs en vérifiant chaque bloc et en s'assurant que les transactions sont valides et que la preuve de travail a été effectuée correctement.

1.4.4 Noeud

Un noeud est une machine qui fait partie d'un réseau P2P, comme illustrer dans la Figure 1.2 Son architecture est symétrique, d'où chaque noeud possède les mêmes avantages que les autres, pas de noeud maître, ou esclave. Le rôle du noeud et de valider les transactions et les blocs [2].

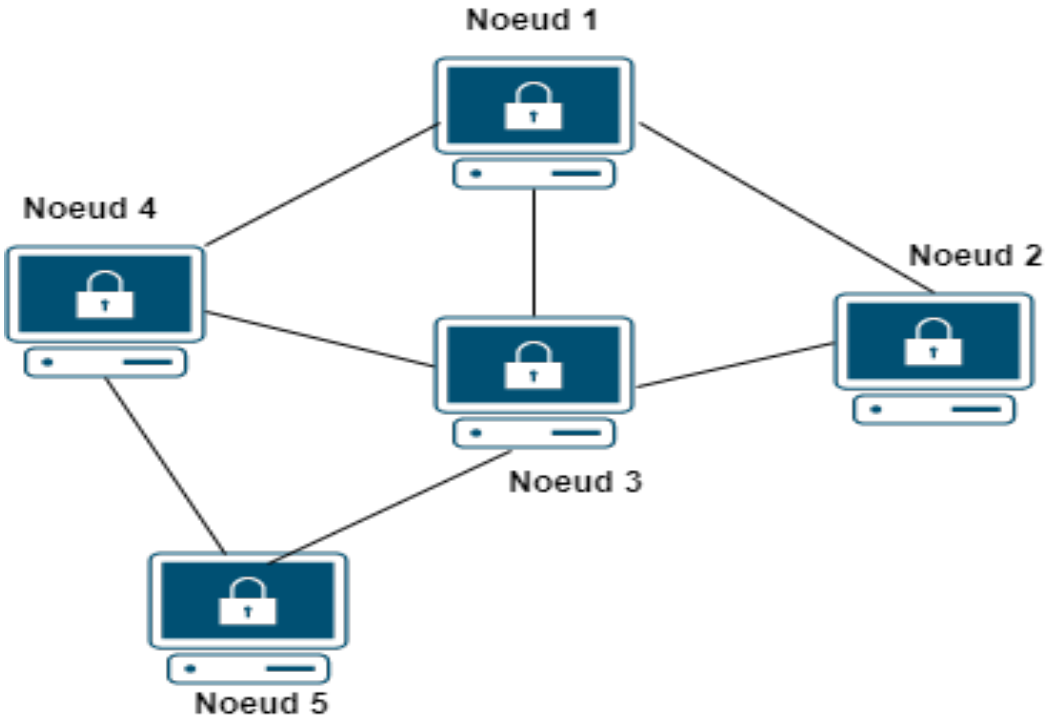


FIGURE 1.2 – R P2P

1.4.5 Bloc

Le bloc constitue l'élément fondamental de la structure de la blockchain. Il englobe une liste de transactions et les distribue à tous les nœuds du réseau une fois validées par des utilisateurs spécifiques appelés mineurs. Ces blocs sont formés de plusieurs transactions signées par des clés publiques, puis ils sont horodatés par les acteurs impliqués. Ce processus est connu sous le nom d'horodatage, et il permet de placer les blocs dans un ordre chronologique, créant ainsi la blockchain où les transactions sont classées successivement. Le bloc se compose de deux parties distinctes : la première est l'entête, une métadonnée qui permet de vérifier sa validité, tandis que la deuxième partie est le contenu, qui contient les transactions sélectionnées par le mineur pour être incluses dans le bloc qu'il a crée. Le nombre maximal de transactions qu'un bloc peut contenir dépend de la taille du bloc lui-même ainsi que de la taille de chaque transaction [3, 10].

- **L'en-tête du bloc**
  Processus de hachage en deux temps : il fait référence à la manière dont cet en-tête est haché pour sécuriser le bloc et lier les blocs entre eux de manière immuable. Voici comment cela fonctionne généralement :

  1. Hachage initia : Les informations de l'en-tête sont concaténées et hachées une première fois à l'aide d'une fonction de hachage cryptographique, généralement SHA-256 (Secure Hash Algorithm 256 bits). Cela produit un hash unique qui résume toutes les

informations de l'en-tête.

2. Minage : Le but du minage est de trouver une valeur de nonce qui, en combinant avec l'en-tête haché, satisfait certaines conditions prédéfinies (comme un certain nombre de zéros au début du hash). Les mineurs ajustent le nonce de manière itérative jusqu'à ce qu'ils trouvent un hash satisfaisant.
3. Hachage final : Une fois qu'un mineur a trouvé un nonce approprié, l'en-tête est à nouveau haché avec ce nonce pour produire le hash final du bloc. Ce deuxième hachage incorpore le résultat du minage dans le hash du bloc, ce qui prouve que le travail a été effectué pour résoudre un problème complexe et ainsi sécuriser le bloc.

L'en-tête d'un bloc dans une blockchain contient plusieurs informations essentielles qui aident à sécuriser, valider et lier les blocs entre eux de manière cohérente tel que illustré dans la figure 1.3, il est haché deux fois pour créer l'empreinte numérique, cette dernière contient des informations sur le bloc lui-même, telles que son numéro de séquence, son horodatage, sa version, son hachage, sa taille et un nonce. Voici les principaux éléments qui sont généralement inclus dans l'en-tête d'un bloc :

- Numéro du bloc
- Les hachages des blocs précédents : chaque bloc de la chaîne de blocs contient un hachage qui identifie le bloc précédent dans la chaîne, et il permet de relier les blocs les uns aux autres.
- Nonce : Un nombre aléatoire utilisé dans le processus de minage, c'est l'une des choses qu'on peut modifier en minant, pour créer différents hashs et trouver le hash adéquat.
- L'horodatage de la création du bloc.
- Racine de Merkle : est un hash unique obtenu en résumant toutes les transactions présentes dans un bloc. L'arbre de Merkle consolide ensuite l'ensemble des transactions en calculant successivement des hashes, jusqu'à ce que la racine de l'arbre soit trouvée.
- Difficulté : Le champ "difficulté" (difficulty en anglais) fait référence à un paramètre qui contrôle la complexité du processus de minage nécessaire pour ajouter un nouveau bloc à la chaîne de blocs. La difficulté est utilisée pour réguler la fréquence à laquelle de nouveaux blocs sont ajoutés, garantissant ainsi un rythme stable et prévisible.

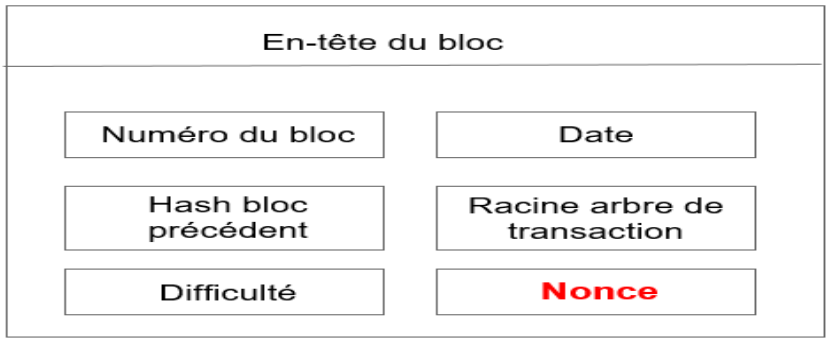


FIGURE 1.3 – En-tête du bloc

- **Le contenu du bloc** est enregistré de manière permanente dans la blockchain et est accessible à tous les nœuds du réseau et il est constitué de la transaction.
- **Les méta-données** chaque bloc peut contenir des méta-données supplémentaires, telles que des identifiants de nœuds, des signatures numériques ou des données de chiffrement.

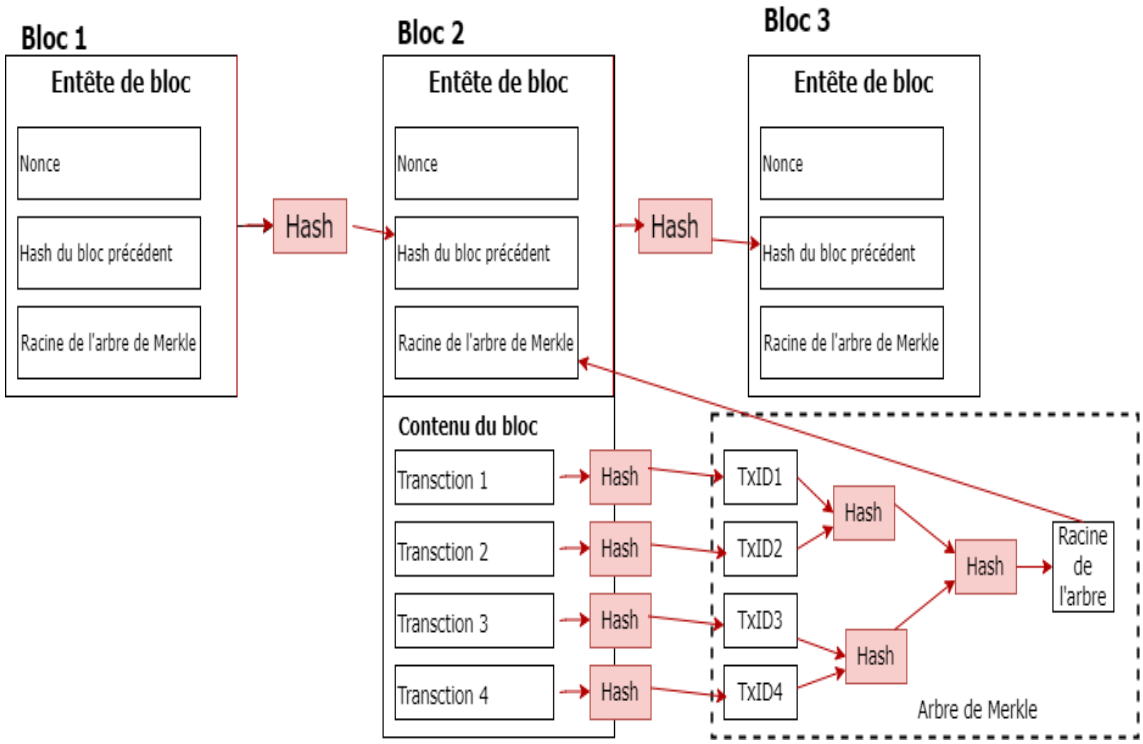


FIGURE 1.4 – forme détaillée d'un bloc

Arbre de Merkle

Arbre de merkle réalisé par 'Ralph Merkle' en 1979, il représente une suite de blocs de données qui comportent chacun le hash du bloc précédent, formant une structure spécifique pour recouvrir

un ensemble de données. Dans le domaine de la cryptomonnaie il représente les transactions de la blockchain, et permet de vérifier l'intégrité et l'état d'une transaction et de simplifier l'accessibilité des données [10].

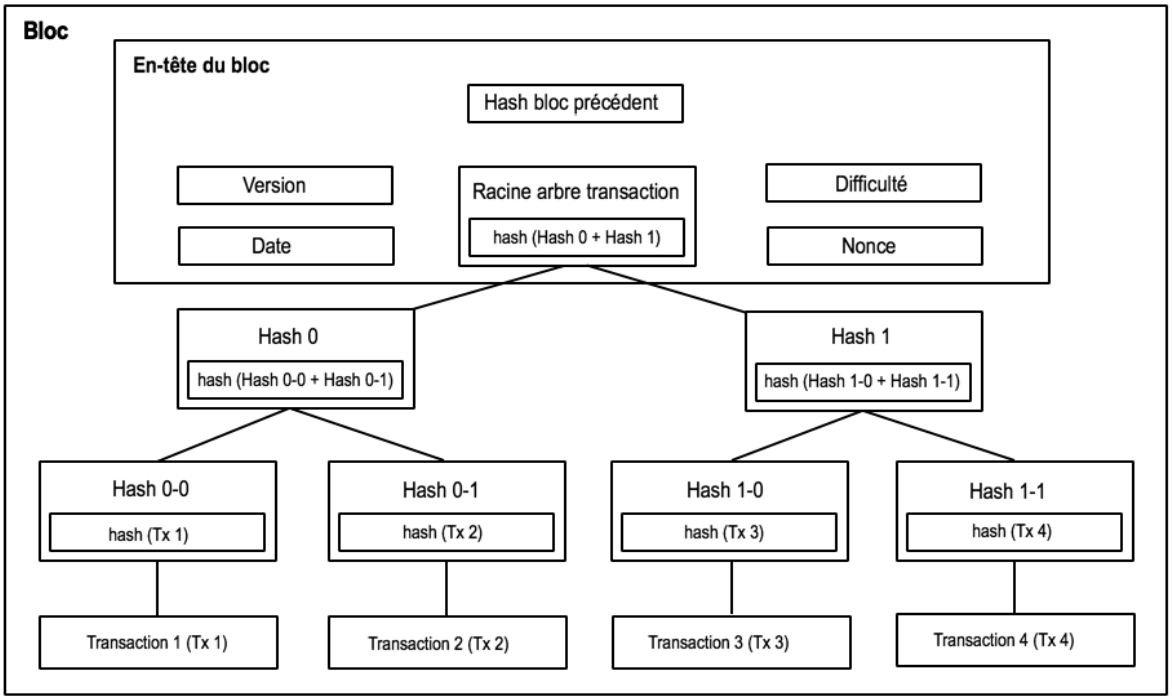


FIGURE 1.5 – schéma Arbre de Merkle

1.4.6 Transaction

Une transaction dans le contexte de la blockchain est le fait qu'un émetteur envoie des données à un récepteur dans le réseau. Quand quelqu'un demande d'effectuer une transaction, cette dernière est diffusée sur un réseau P2P, composé de machines appelées noeuds, chaque noeud héberge une copie de la base de données dans laquelle inscrit l'historique des transactions validées. Avant que la transaction soit reçue, on doit passer par les 5 étapes illustrées dans la figure 1.6. La transaction est enregistrée dans un bloc après validation par les noeuds du réseau qui utilise des algorithmes de consensus pour réaliser cette phase [2].

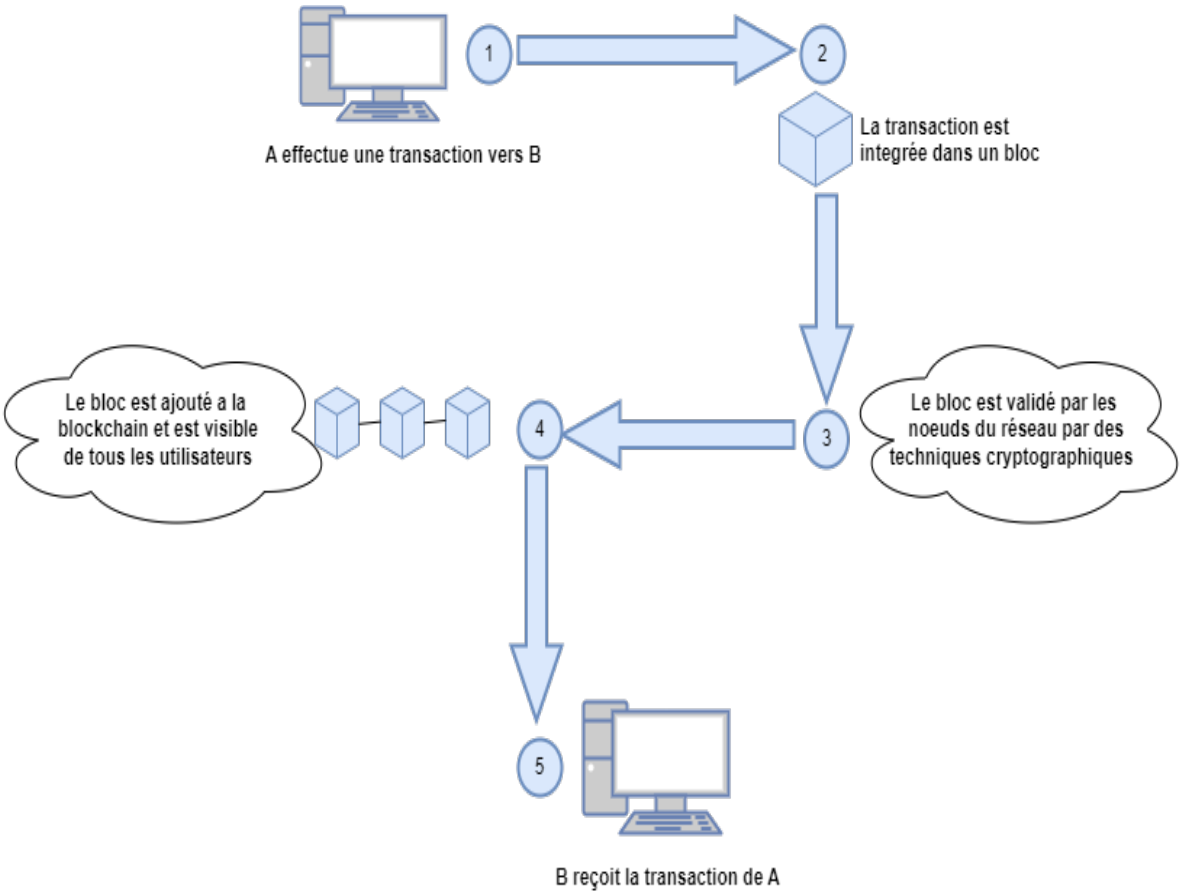


FIGURE 1.6 – Processus d'envoi de la transaction

1.4.7 Adresse

Dans la blockchain bitcoin pour gérer les identités sur le réseau chaque compte dispose d'une clé publique et d'une clé privée, puis on va assimiler la clé publique à l'adresse bitcoin d'une personne ce qui signifie le numéro du compte de la personne mais cela ne permet pas d'identifier la personne physique qui la possède. Ainsi les adresses de blockchain sont utilisées pour envoyer et recevoir des fonds en cryptomonnaies sur une blockchain. Le fonctionnement des adresses de blockchain est relativement simple. Chaque utilisateur de blockchain possède un portefeuille numérique, qui contient une ou plusieurs paires de clés cryptographiques : une clé publique et une clé privée. La clé publique est utilisée pour créer l'adresse de la blockchain, tandis que la clé privée est utilisée pour signer les transactions. Lorsqu'un utilisateur souhaite envoyer des fonds à une autre personne sur la blockchain, il doit saisir l'adresse du portefeuille du destinataire dans le champ approprié. La blockchain vérifie alors que cette adresse existe bien sur la blockchain et qu'elle appartient bien au destinataire avant de permettre la transaction. Les adresses des comptes sont générées à l'aide des paires de clé publique, clé privée, cela se fait aléatoirement en 256 bits et ils utilisent une fonction de hachage qui est appliquée à la clé publique pour obtenir l'adresse du compte, ce qui les rendent unique et ne peuvent pas être modifiées une fois créées [3, 10].

1.4.8 Portefeuille

Dans le contexte de la technologie blockchain, un portefeuille de blockchain, également appelé ”wallet”, est un logiciel ou un dispositif matériel qui permet de stocker, de gérer et d'envoyer des cryptomonnaies en toute sécurité. Les portefeuilles de blockchain sont indispensables pour interagir avec les différentes blockchains et pour effectuer des transactions en cryptomonnaies. Ils permettent notamment de générer des adresses de blockchain uniques pour chaque utilisateur, de stocker les clés privées associées à ces adresses et de signer les transactions [22].

1.5 Types de blockchain

Tous les types de blockchains ont en commun un consensus décentralisé et utilisent un protocole de communication pair-à-pair. Ce qui les distingue, ce sont les règles spécifiques régissant l'utilisation de la blockchain, qui déterminent si une entité agit en tant que validateur ou nœud dans le réseau [21]. On peut distinguer trois grands types :

1.5.1 Public

Dans ce type, tous les nœuds du réseau participent à l'échange de manière égalitaire, sans aucune autorisation préalable requise pour effectuer une transaction. Un exemple de cette blockchain est Bitcoin, souvent appelée la blockchain originale. Elle peut être définie comme une base de données ouverte et accessible à tous, sans restrictions ni identification préalable.

1.5.2 Privée

Ce type de blockchain fonctionne sur un réseau privé, où le gestionnaire peut modifier le protocole à sa guise, et où personne ne peut y participer sans autorisation. Il est particulièrement adapté aux organismes tels que les banques et les entreprises. Son intérêt est limité car il ne permet pas la liaison entre différents acteurs.

1.5.3 Hybride (consortium)

Lorsqu'un ensemble spécifique de nœuds est créé pour contrôler le processus de consensus, on parle de blockchain de consortium. Ce type peut être considéré comme partiellement décentralisé, car l'accès au réseau peut être limité à un certain nombre de participants. La blockchain de consortium, ou hybride, est sous le contrôle d'un groupe d'organisations, où le droit d'accès peut être ouvert à tous ou limité à certains utilisateurs. Ce type de blockchain est souvent utilisé dans des secteurs fortement réglementés, où seul un groupe de nœuds prés sélectionnés participe au processus de consensus.

1.6 Structure d'une blockchain

La structure de la technologie Blockchain est représentée par une liste de blocs avec des transactions dans un ordre particulier. De nombreuses transactions forment un bloc, qui sont reliés entre eux pour former une chaîne, tel que expliquer dans la figure 1.7. La structure de la Blockchain est un élément crucial de son fonctionnement. Elle est conçue pour garantir la sécurité, la transparence et la décentralisation des données stockées dans la Blockchain [3, 22]. Voici les différents éléments qui composent la structure de la Blockchain :

- Les transactions : Les transactions sont des échanges de données entre les portefeuilles numériques dans la Blockchain. Chaque transaction est validée et vérifiée par le réseau de nœuds avant d'être ajoutée à un bloc.
- Les blocs : Les blocs sont les unités de base de la Blockchain. Chaque bloc contient un ensemble de transactions qui ont été validées par le réseau de nœuds. Chaque bloc est également lié au bloc précédent par un code de hachage, formant ainsi une chaîne continue de blocs.
- Le réseau de nœuds : Les nœuds sont des ordinateurs connectés à la Blockchain qui jouent un rôle important dans la validation des transactions et la vérification des blocs. Les nœuds travaillent ensemble pour atteindre un consensus sur l'état actuel de la Blockchain, garantissant ainsi l'intégrité des données stockées.
- Les mineurs : Les mineurs sont des nœuds qui participent au processus de validation des transactions et à la création de nouveaux blocs. Les mineurs résolvent des problèmes mathématiques complexes pour ajouter de nouveaux blocs à la Blockchain.

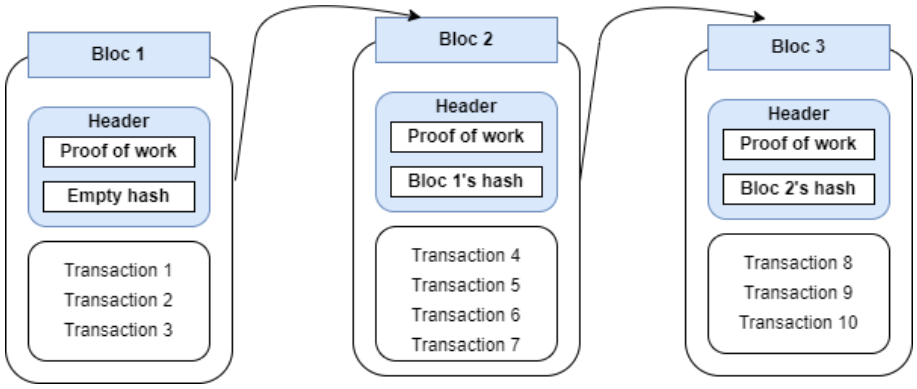


FIGURE 1.7 – la structure de la blockchain

1.7 Fonctionnement de la blockchain

Les blocs d'une blockchain peuvent contenir différent types de données, des enregistrements de transactions (Bitcoin), des images, des textes, des applications, etc. Ces données peuvent être chiffrées ou enregistrées en clair. Les blocs d'information dans la blockchain sont liés en constituant une chaîne. Ce chaînage s'effectue à l'ordre de la signature numérique de chaque bloc qui est transmise au suivant dans la chaîne. Cette signature est appelée Hash qui est chiffrée par un algorithme de chiffrement asymétrique (SHA 256, Keccak-256 [1] [2]). La Blockchain est répliquée chez les utilisateurs et sur toutes les machines. Cet ordre de transaction est représenté par une adresse et qui sera distribué dans le réseau de la blockchain.

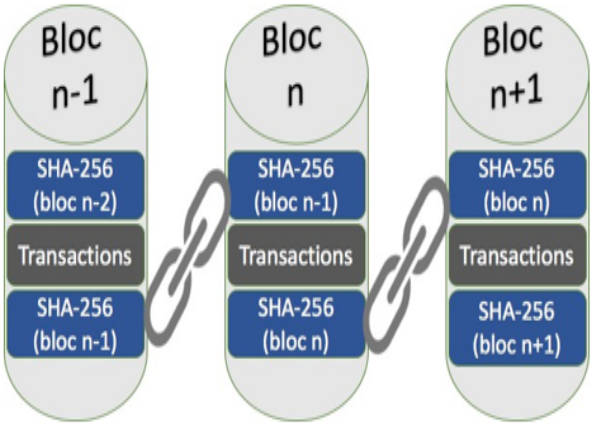


FIGURE 1.8 – Lien entre les blocs [5]

Quand les utilisateurs du réseau veulent effectuer des transactions elles seront regroupées en blocs. Chaque bloc est validé par les mineurs au moyen de techniques cryptographiques. Les blocs de données se fait via un consensus entre les membres du réseau qui le souhaitent (mineurs), qui vont alors entrer en compétition pour valider le bloc, ils vont aussi devoir résoudre un problème mathématique complexe reposant sur un principe de cryptographie, en utilisant les capacités de calcul de leur ordinateurs (Preuve de travail). La première machine à obtenir la solution à ce problème, propose son bloc au réseau et le soumettre à un vote afin de le vérifier et de le valider. Si une majorité de gens approuve ce bloc sera validé, et il sera daté et ajouté et chaîné au bloc précédent de la blockchain et devrais être visibles accessibles par tous les utilisateurs. Si ce n'est pas le cas, le bloc sera alors rejeté. Il n'est pas possible de modifier la chaîne car les blocs sont ordonnés chronologiquement et informatiquement, toutes modifications d'un bloc passé brise cette chaîne. Les informations sont regroupées par paquets qui sont ajoutés à la chaîne, chaque paquet

1. Keccak-256 est une fonction de hachage cryptographique basée sur l'algorithme Keccak.
2. Elle a été conçue pour être résistante aux attaques cryptographiques courantes et offre une sécurité solide.

de données est lié au précédent de sorte à ce qu'une modification d'une données altérée la totalité, ces chaînes de blocs sont par conséquent immuables.
On a 2 types de nœuds sur le réseau blockchain :
- Les utilisateurs qui produisent de l'informations.
- Les mineurs qui sont des particuliers qui permettent de vérifier la validité des transactions bloc par bloc en produisant des preuves de travail ce qui explique la figure [10, 11].

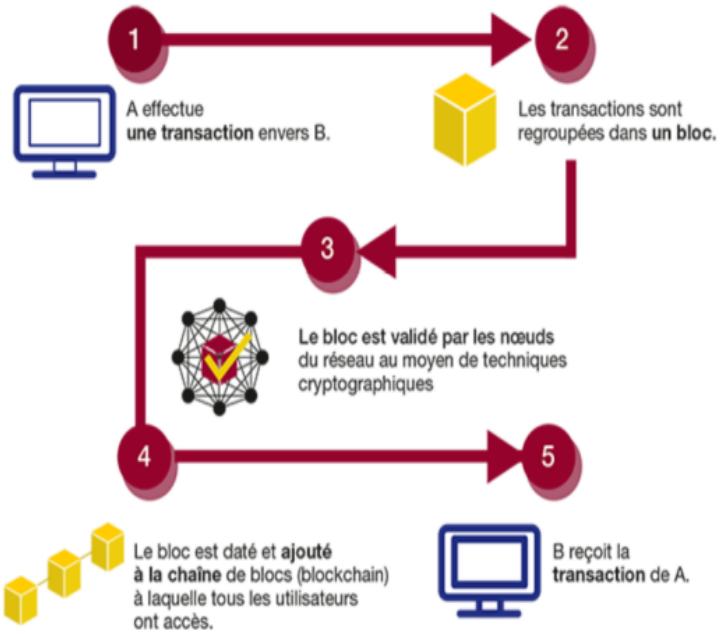


FIGURE 1.9 – Fonctionnement de la blockchain [6]

1.8 Domaines d'application

La blockchain est une technologie innovante qui propose des améliorations dans divers secteurs. De nombreuses applications s'intéressent à cette technologie en raison de sa capacité à sécuriser les données et à accélérer les transactions [2, 12]. Dans ce qui suit, nous présentons les principaux domaines d'application de la b Blockchain :

1.8.1 Banque

La blockchain a particulièrement ciblé le secteur bancaire en vue d'éliminer les intermédiaires et de proposer des solutions permettant des transactions financières rapides, transparentes et sécurisées.

1.8.2 Santé

Dans le domaine de la santé, la technologie blockchain permet le partage et le stockage sécurisé des données cliniques, en identifiant qui a le droit d'y accéder. Cela renforce la confiance des patients dans la protection de leurs informations sensibles.

1.8.3 L'éducation

Dans le secteur d'éducation il y'avait de nombreux cas de fraude de degré qu'on trouve plusieurs étudiants ou élèves qui falsifient leurs diplômes ou leurs relevés de notes. C'est pour cela certaines universités et instituts ont appliqué la blockchain pour contribuer à réduire le nombre de fraudes. La blockchain permet d'accorder et gérer les diplômes des étudiant(e)s et les stockés tout en les sécurisants.

1.8.4 Energie

La blockchain est appliquée dans le domaine de l'énergie pour garantir que l'énergie produite par des sources telles que les panneaux solaires est vendue directement aux acheteurs, sans passer par des fournisseurs traditionnels.

1.8.5 Scrutin électoral (vote)

Un scrutin électoral gérer de manière transparente et équitable garantie la stabilité des institutions et l'épanouissement social d'un pays. La généralisation de l'utilisation de la blockchain dans les scrutins électoraux empêche toute modification ou suppression d'un vote une fois validé, c'est donc une garantie de sécurité et de fiabilité du scrutin qui rassure toute la classe politique dans son ensemble.

1.8.6 Commerce

La blockchain peut contribuer à l'amélioration du secteur du commerce, en permettant le suivi de la chaîne d'approvisionnement dés le début de l'acquisition des marchandises jusqu'à leurs points de distribution et de vente.

1.8.7 Identification numérique

Avec une estimation de plus d'un milliard de personnes dans le monde qui n'ont pas d'identité, Microsoft travaille à la création de cartes d'identité pour donner plus de pouvoir aux personnes pauvres et aux réfugiés. Cela permettra de les connecter au secteur financier formel. Microsoft vise à atteindre cet objectif grâce à son application Authenticator, qui serait basée sur la technologie

Blockchain. Authenticator n'utilise pas seulement des mots de passe, elle utilise plusieurs couches de sécurité qui utilisent un code ou un jeton pour identifier l'utilisateur ou l'appareil qui revient.

1.9 Contrats intelligents

Les contrats intelligents, également connus sous le nom de "smart contracts" en anglais, sont des programmes informatiques autonomes qui exécutent automatiquement les termes d'un contrat dès que les conditions prédéfinies sont remplies. Ils sont souvent associés à la technologie blockchain, bien que leur utilisation ne se limite pas exclusivement à cette dernière. Un contrat intelligent est écrit sous forme de code informatique et stocké sur une blockchain. Il peut contenir des règles, des conditions, des clauses et des actions spécifiques qui doivent être remplies et exécutées par les parties concernées. Une fois que les conditions sont vérifiées, le contrat intelligent se met automatiquement en action, sans nécessiter d'intervention humaine supplémentaire. L'avantage des contrats intelligents réside dans leur automatisation et leur exécution immédiate et précise. Ils éliminent le besoin d'intermédiaires ou de tiers de confiance, car ils reposent sur la technologie blockchain, qui garantit la transparence, l'immuabilité et la sécurité des transactions. Les contrats intelligents ont de nombreuses applications potentielles dans différents domaines, tels que les transactions financières, l'immobilier, la logistique, les assurances, les soins de santé et bien d'autres. Ils offrent la possibilité d'automatiser des processus complexes, de réduire les coûts, d'accélérer les transactions et d'assurer une exécution transparente des accords contractuels. Il convient de noter même si les contrats intelligents présentent de nombreux avantages, ils ne sont pas exempts de risques et de limites. La qualité du code utilisé pour les programmer est cruciale, car une fois déployé sur une blockchain, il est difficile de le modifier. De plus, les contrats intelligents ne peuvent pas interpréter les informations provenant du monde réel de manière autonome, ce qui limite leur applicabilité à des conditions préalablement définies [30].

1.9.1 Structure d'un contrat intelligent

La structure d'un contrat intelligent peut varier en fonction de la plateforme ou de la technologie utilisée pour le mettre en œuvre, voici une structure générale couramment utilisée :

- **Définition des parties :** Le contrat intelligent commence par l'identification des parties impliquées. Il peut s'agir d'individus, d'entreprises ou d'entités juridiques spécifiques.
- **Conditions préalables :** Cette section énonce les conditions qui doivent être remplies avant que le contrat puisse être exécuté. Cela peut inclure des vérifications d'identité, des validations de données ou d'autres exigences spécifiques.
- **Clauses contractuelles :** Les clauses contractuelles définissent les termes et les conditions du contrat. Elles peuvent inclure des déclarations, des obligations, des droits et des responsabi-

lités pour chaque partie impliquée. Les clauses peuvent également spécifier les événements déclencheurs qui activeront l'exécution du contrat.

- **Conditions de déclenchement :** Cette section précise les conditions ou les événements spécifiques qui doivent se produire pour que le contrat soit exécuté. Il peut s'agir de dates, d'horaires, de conditions de paiement ou de tout autre paramètre prédéfini.
- **Actions et exécution :** Une fois que les conditions de déclenchement sont remplies, cette partie du contrat intelligent spécifie les actions à entreprendre automatiquement. Il peut s'agir de transferts de fonds, de l'émission de certificats numériques, de l'activation de fonctionnalités ou de tout autre processus automatisé.
- **Mécanismes de résolution des litiges :** Cette section peut inclure des mécanismes spécifiques pour résoudre les éventuels litiges ou désaccords entre les parties. Cela peut impliquer l'utilisation d'arbitrage, de médiation ou d'autres moyens de résolution alternative des conflits.
- **Conditions de résiliation :** Si nécessaire, cette partie du contrat intelligent énonce les conditions dans lesquelles le contrat peut être résilié avant son expiration normale. Cela peut inclure des pénalités, des frais ou d'autres conséquences liées à la résiliation anticipée.

Il est important de noter que la structure d'un contrat intelligent peut être plus complexe en fonction de la nature et de la complexité de l'accord contractuel. De plus, différentes plateformes ou technologies peuvent avoir leurs propres spécificités et langages de programmation pour la mise en œuvre des contrats intelligents.

1.9.2 Exécution et le déploiement d'un contrat intelligent

L'exécution et le déploiement d'un contrat intelligent impliquent les étapes suivantes [18, 31] :

1. **Écriture du contrat intelligent :** Tout d'abord, le contrat intelligent doit être écrit en utilisant un langage de programmation compatible avec la plateforme blockchain choisie. Par exemple, Solidity est souvent utilisé pour écrire des contrats intelligents sur la blockchain Ethereum. Le contrat intelligent doit être soigneusement conçu pour inclure les termes, les conditions et les actions souhaitées.
2. **Test et vérification :** Avant de déployer un contrat intelligent sur la blockchain en production, il est recommandé de le tester et de le vérifier rigoureusement pour s'assurer de son bon fonctionnement. Des outils de développement et des simulateurs peuvent être utilisés pour exécuter des tests unitaires, des tests de sécurité et des simulations de scénarios.
3. **Déploiement sur la blockchain :** Une fois que le contrat intelligent a été testé et vérifié, il peut être déployé sur la blockchain. Le déploiement consiste à télécharger le code du contrat intelligent sur la blockchain et à l'associer à une adresse unique. Cette adresse servira de référence pour interagir avec le contrat intelligent.

4. **Exécution automatique :** Une fois le contrat intelligent déployé, il est en attente d'exécution. Les actions définies dans le contrat seront automatiquement déclenchées dès que les conditions préalables spécifiées seront remplies. Par exemple, si un contrat intelligent de location stipule que le locataire doit effectuer un paiement mensuel avant une date limite, dès que le locataire effectue ce paiement à la bonne adresse, le contrat intelligent exécute automatiquement l'action de validation du paiement.

5. **Interactions avec le contrat :** Les parties impliquées peuvent interagir avec le contrat intelligent en utilisant des transactions spécifiques sur la blockchain. Par exemple, pour un contrat intelligent de vente, l'acheteur peut envoyer un paiement à l'adresse du contrat intelligent pour déclencher le transfert de propriété.

6. **Conditions de résiliation :** Au besoin, cette section du contrat intelligent spécifie les circonstances dans lesquelles le contrat peut être résilié avant son terme prévu. Ces dispositions pourraient englober des pénalités, des frais ou d'autres répercussions associées à une annulation anticipée.

Les contrats intelligents une fois déployés sont généralement immuables, ce qui signifie qu'il est difficile de les modifier ou de les arrêter une fois qu'ils sont en cours d'exécution sur la blockchain. Il est donc crucial de bien tester et vérifier le contrat avant son déploiement.

1.10 Conclusion

Il semble évident que la Blockchain s'impose naturellement dans les domaines où nous traitons beaucoup d'argent. Puis progressivement, s'imposer à d'autres secteurs, comme l'éducation, que nous verrons au chapitre suivant. Il est encore tôt pour savoir à quel point cette technologie va révolutionner nos sociétés, mais une chose est sûre, la blockchain représente une opportunité inédite pour transformer les systèmes actuels.

Chapitre 2

Blockchain dans les institutions éducatives

2.1 Introduction

Les technologies de la blockchain n'ont pas cessé de conquérir des domaines et des secteurs d'activité divers et variés, à l'instar du secteur de l'éducation (objet de notre présent mémoire). En effet les institutions éducatives ont de plus en plus recours à cette technologie de la blockchain dans leur gestion administrative quotidienne et pour des objectifs divers :

- **Certification et vérification des diplômes :** Les institutions éducatives peuvent utiliser la blockchain pour émettre des certificats et des diplômes de manière sécurisée et vérifiable. Les informations pertinentes sont enregistrées sur la blockchain, ce qui permet aux employeurs ou aux autres parties intéressées de vérifier facilement l'authenticité des diplômes.

- **Partage de ressources éducatives :** La blockchain peut être utilisée pour créer des plates-formes de partage de ressources éducatives entre les institutions. Les enseignants et les chercheurs peuvent partager du contenu pédagogique, des études de cas, des matériaux d'apprentissage, etc., en s'assurant que l'accès est sécurisé et que les droits d'auteur sont respectés.

- **Gestion des identités et des accès :** La blockchain peut servir de base pour la gestion des identités numériques dans les institutions éducatives. Cela permet de gérer de manière sécurisée les informations d'identification des étudiants, des enseignants et du personnel administratif, ainsi que de contrôler l'accès aux ressources et aux services en ligne.

- **Financement et paiement :** La blockchain peut faciliter les transactions financières entre les institutions éducatives, les étudiants et les bailleurs de fonds. Elle permet des paiements sécurisés, traçables et sans intermédiaire, ce qui peut simplifier les processus de collecte de frais de scolarité, de distribution de bourses et de financement de projets éducatifs.

- **Transparence et intégrité de données :** La blockchain permet de créer un registre transparent et immuable des données, ce qui peut être utile pour garantir l'intégrité des informations

académiques telles que les relevés de notes, les calendriers, les programmes d'études, etc. Les étudiants, les enseignants et les administrateurs peuvent avoir accès à ces informations de manière sécurisée.

Ces utilisations de la blockchain dans les institutions éducatives sont encore en développement et peuvent varier en fonction des besoins spécifiques de chaque institution. Cependant, elles offrent des avantages potentiels tels que la sécurité des données, la réduction des fraudes, la facilitation des transactions et la promotion de la transparence dans le domaine de l'éducation.
Dans notre cas, on a pris comme exemple le Lycée Chouhadaa Annani.

2.2 Présentation du lycée Annani de Bejaia

Le lycée Chouhadaa Annani (lieu du déroulement de notre stage pratique) est un établissement d'enseignement secondaire situé au cœur de la ville de Béjaia donnant sur le boulevard Krim Belkacem. Le numéro de l'identifiant national du lycée est le 60017. Fondé en 1971 et ouvert en 1978, le lycée Chouhadaa Annani offre un environnement d'apprentissage stimulant et inclusif, propice à l'épanouissement de chaque étudiant. Sa capacité d'accueil est 1400 places, sa superficie totale est de 34240.0 m2 pour une surface bâtie de 16000m2 et un espace vert de 17040m2 . Le lycée est renommé pour son engagement envers l'excellence académique et sa contribution au développement des jeunes élèves. Le lycée Chouhadaa Annani propose un large éventail de programmes éducatifs pour les élèves, allant de l'enseignement général aux filières technologiques et professionnelles. L'objectif principal de l'école est de fournir une éducation de qualité qui prépare les élèves à réussir dans leurs études supérieures et à s'épanouir dans leur vie professionnelle. Les installations du lycée Chouhadaa Annani comprennent 30 salles de classes modernes et bien équipées, 04 laboratoires scientifiques, une bibliothèque bien approvisionnée, une salle informatique, 02 ateliers une pour génie mécanique et l'autre pour génie électrique, un amphithéâtre, une salle de sport, un stade, une salle de soin et un centre UDS (unité de pistage). L'école met également l'accent sur les activités parascolaires et offre aux élèves la possibilité d'intégrer des clubs sportifs et de participer aux compétitions sportives et aux événements culturels et artistiques.

2.2.1 Organisation pédagogique de l'établissement

- Première année tronc commun :

  1. science :
     - Mathématique
     - Science expérimentale
  2. Gestion
  3. Technique Math :

   - génie électrique
   - génie mécanique

4. Lettre :
   - Lettre et langue étranger (espagnol)
   - Lettre et philosophie

- Deuxième année secondaire :

1. science expérimental
2. Mathématique
3. Gestion économique
4. Technique Math :
   - génie électrique
   - génie mécanique
5. Langue étranger
6. Lettre et philosophie

- Troisième année secondaire : elle poursuit la deuxième année.

2.2.2 Carte pédagogique et administratif

Niveau d'étude	**Spécialité**	**Nombre d'élèves**	**Nombre de division**
Première année	Tronc commun lettre	94	3
	Science et technologie	233	6
Deuxième année	Lettre et philosophie	75	2
	Langues étrangères	42	2
	Science expérimentale	123	3
	Gestion économique	62	2
	Mathématique	32	1
	Technique math	40	2
Troisième année	Lettre et philosophie	61	2
	Langues étrangères	49	2
	Science expérimentale	123	3
	Gestion économique	90	2
	Mathématique	30	1
	Technique math	44	2

Le nombre de postes enseignants ouvert est de 73 postes et l'effectif de l'encadrement administratif et de service est de 63 postes.
En ce qui concerne les résultats scolaires, le lycée Chouhadaa Annani a une histoire d'excellence. Les élèves sont encouragés à se surpasser et à viser l'excellence dans leurs études. L'école met en place des mesures de soutien pédagogique pour aider les élèves en difficulté et organise régulièrement des séances de tutorat et de suivi individualisé.

2.2.3 organigramme administratif du lycée

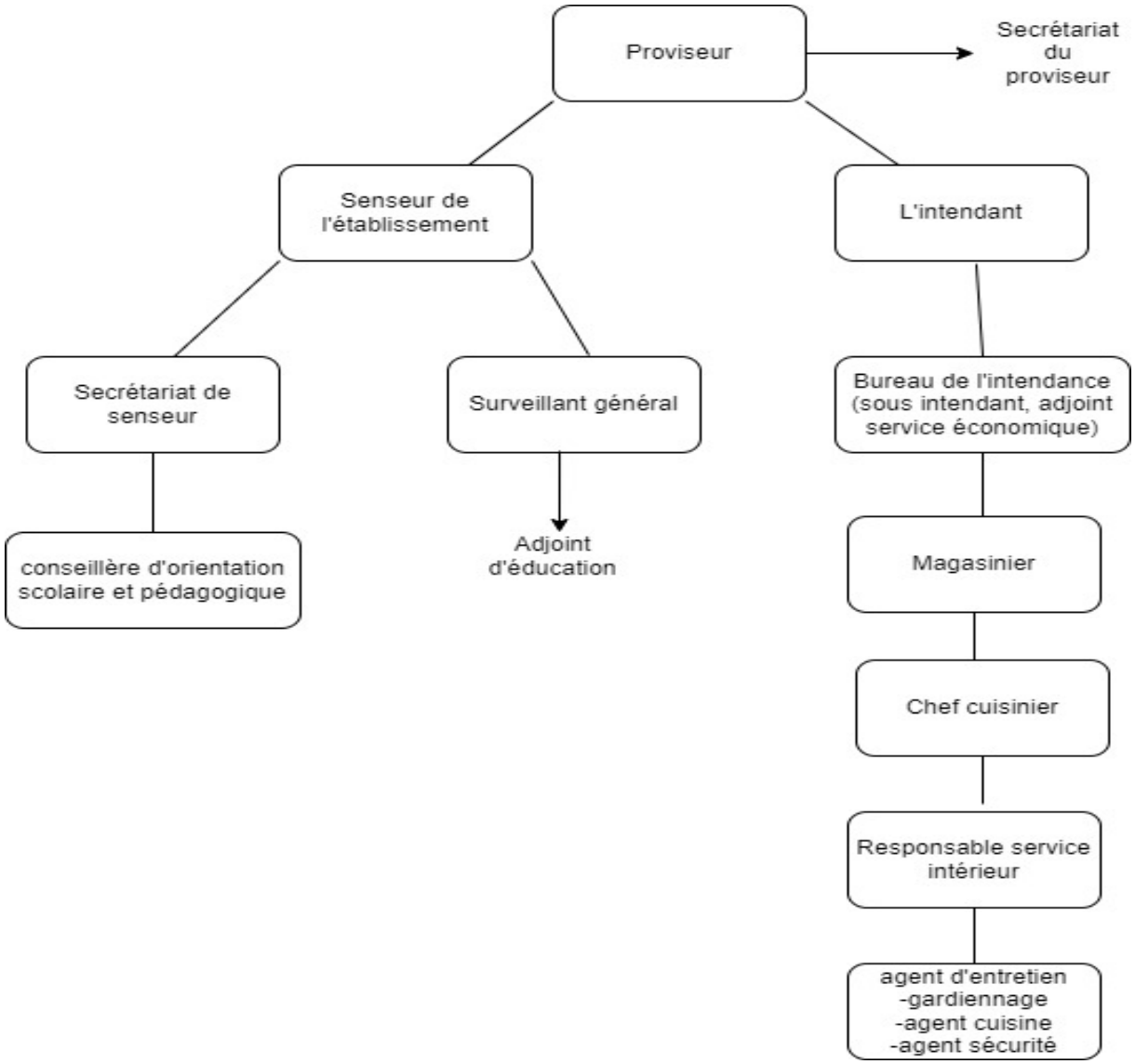


FIGURE 2.1 – Organigramme administratif du lycée

En résumé, le lycée Chouhadaa Annani est un établissement d'enseignement secondaire engagé dans la promotion de l'excellence académique et du développement global des élèves. Il offre un environnement d'apprentissage stimulant, des installations modernes et un corps professoral dévoué. L'école vise à préparer les élèves à réussir dans leurs études supérieures et à devenir des citoyens responsables et engagés.

2.3 Problématique

La difficulté de gestion d'absence des élèves est le problème majeur du lycée Chouhadaa Annani (lieu de stage), car les élèves sont sous la responsabilité de l'établissement pendant les horaires d'enseignement, d'où la sensibilité du problème et l'importance de sa résolution. Comment pourrons nous résoudre ce problème en se basant sur la technologie Blockchain ?

2.4 Solution

On est maintenant au courant que les blocs enregistrent toutes les transactions validées dans le réseau ainsi que toutes ses interactions, ce qui rend la blockchain une technologie de traçabilité, à partir de cette dernière on pourra résoudre le problème de l'absence des élèves dans l'école en créant un système de suivi et de vérification des présences. On considère que c'est l'enseignant qui doit marquer l'absence de ses élèves, et que chaque élève aurait un identifiant unique qui serait associé à une adresse blockchain, l'enseignant va avoir l'accès à un espace là où il insère l'ID de l'élève en question, cette action est donc prise comme une transaction d'où la nécessité de l'enregistrer dans un bloc après avoir été validée, le bloc sera horodaté, visible par tous les autres acteurs et en même temps inchangeable. Des notifications pourraient être envoyées aux parents lorsque leur enfant est absent, ce qui leur permettrait de prendre des mesures pour s'assurer qu'il assiste aux cours de manière régulière.
Toutes les informations sur les absences des élèves seraient stockées de manière décentralisée et sécurisée, ce qui empêcherait toute modification ou falsification des données. De plus, il va y avoir une possibilité de générer des rapports à la fin de chaque mois ou trimestre pour avoir un aperçu global des taux de présence des élèves. Cela permettrait aux enseignants et aux administrateurs de l'école de surveiller les taux de présence à l'échelle de l'école et d'identifier les problèmes lié a cet acte, et donc mettre en place des mesures pour les résoudre.

2.4.1 Architecture globale de notre système

Notre système repose sur un réseau blockchain privé, où chaque nœud possède une copie de la blockchain. Dans ce réseau, l'enseignant joue le rôle central en ajoutant les absences des élèves. Les nœuds élèves autorisent ensuite les nœuds tiers à accéder aux données personnelles des élèves. Pour interagir avec la blockchain, chaque acteur doit posséder un compte dans notre

système et utiliser une application web via un navigateur. Les données personnelles des acteurs sont stockées dans une base de données classique sur le serveur, tandis que les absences des élèves sont enregistrées dans la blockchain.

- Noeuds tierces : dans notre système les noeuds tierces peuvent représentés les éléments suivants : Portefeuille (Wallet), explorateur de blocs, services de stockage (BDD), et les services d'identité. Ces éléments jouent un rôle essentiel en vérifiant les transactions et en validant les blocs, stockage de données authentifiées des utilisateurs, etc.

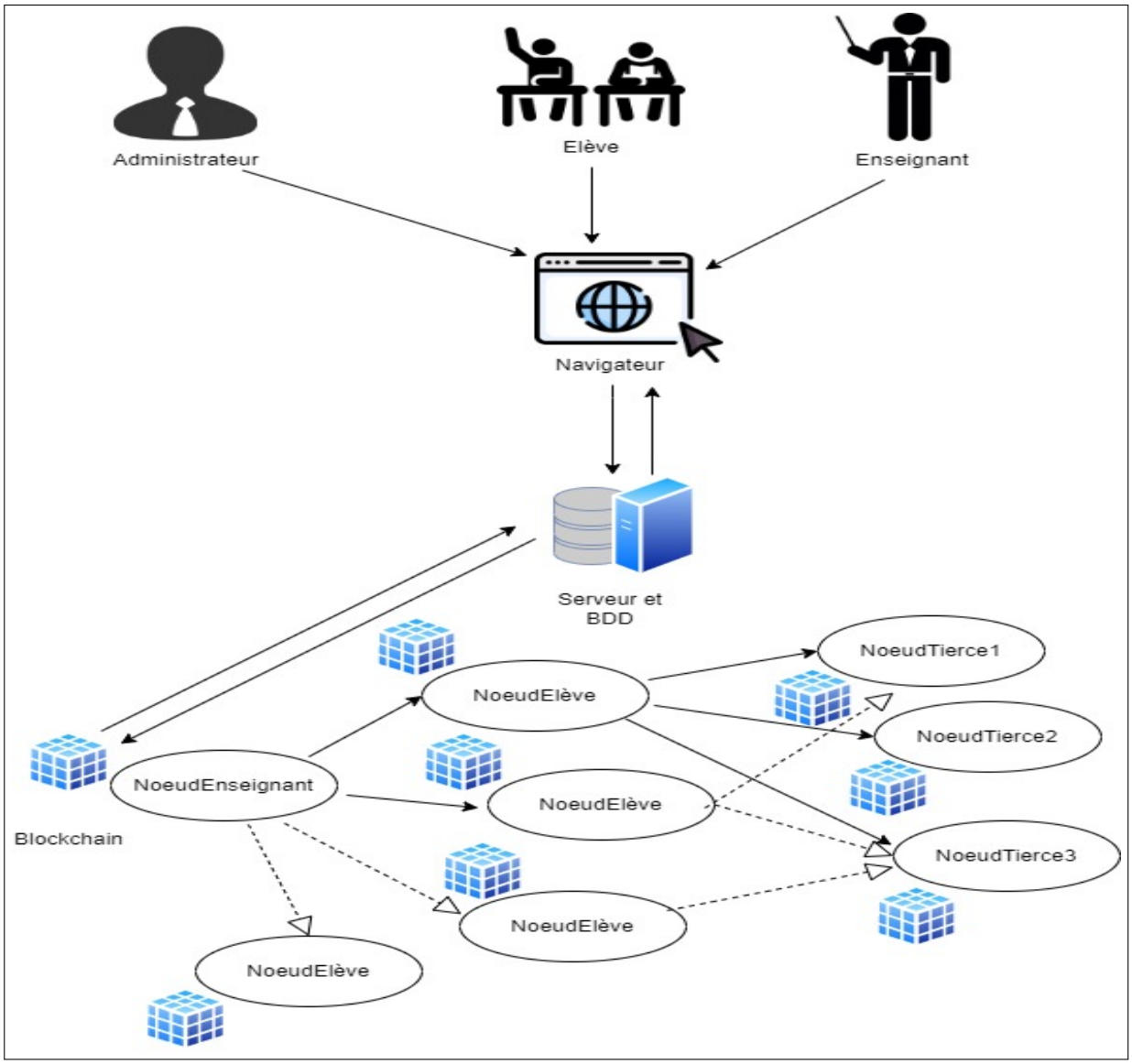


FIGURE 2.2 – Schéma global de notre système

2.4.2 Schéma de solution

Notre système se compose principalement de 4 acteurs, une base de données classique et un réseau blockchain, comme vous pouvez voir dans la figure 2.3 chacun possède un rôle essentiel contribuant au bon fonctionnement de notre application.

1. L'enseignant insert l'ID d'élève qui est absent. Chaque élève doit avoir son identifiant.
2. L'information sera enregistrée dans la Bockchain et validé par un groupe de mineurs en utilisant les algorithmes de consensus.
3. Administrateur va avoir la possibilité de gérer les listes des utilisateurs et leur compte.
4. Les parents assurent que leur enfant assiste à l'école et justifier leur absence.
5. Les élèves sont les principaux acteurs dans la gestion de leurs propres absences. Ils peuvent aussi consulter et poser la justification de leurs absences.

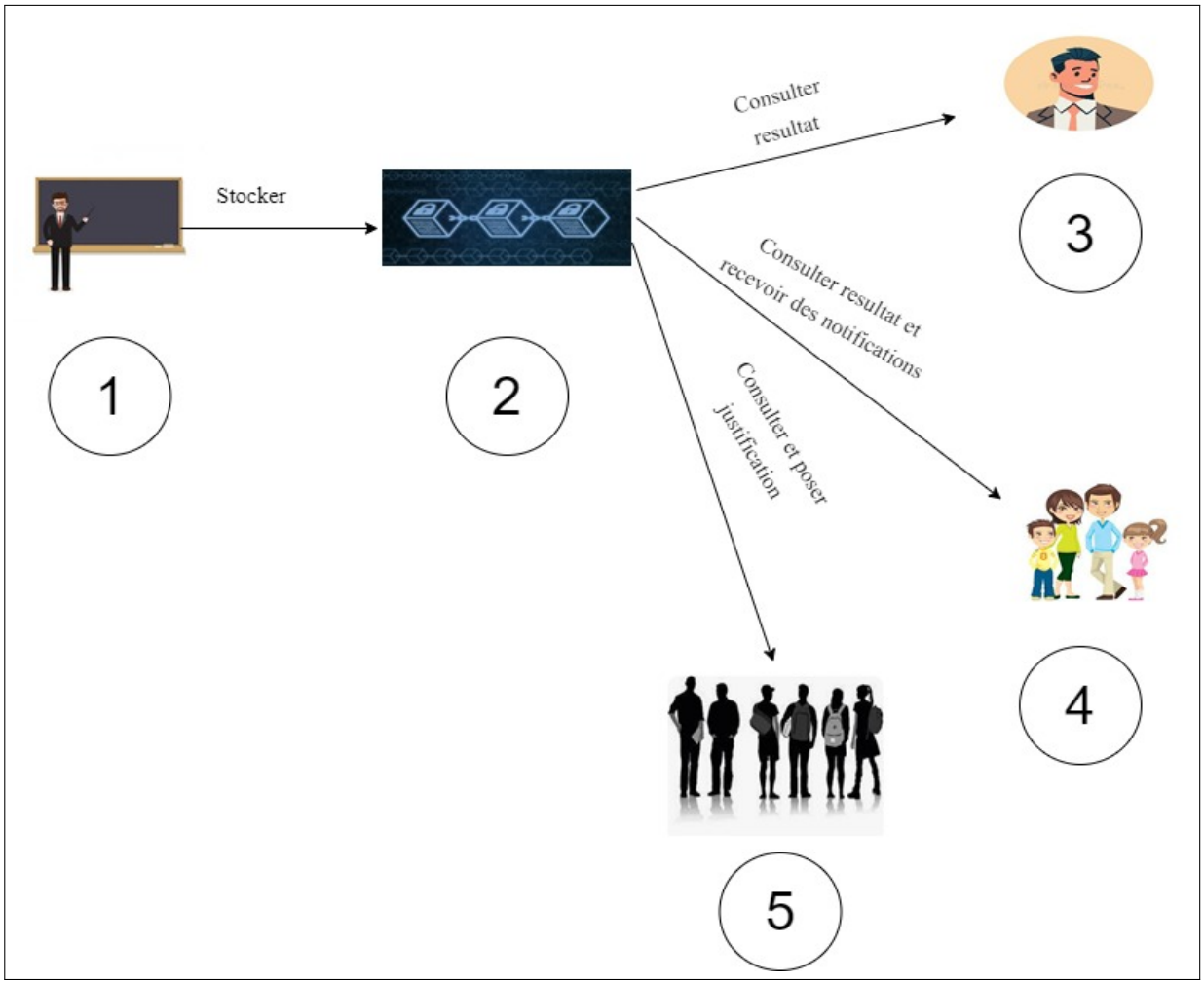


FIGURE 2.3 – Schéma global de notre système

2.5 Conclusion

L'intégration de la blockchain dans les institutions éducatives présente des perspectives prometteuses pour révolutionner la manière dont l'apprentissage et l'administration sont gérés. En exploitant la nature décentralisée, transparente et sécurisée de la technologie blockchain, les écoles peuvent rationaliser les processus administratifs, garantir la bonne gestion des absences, ainsi que promouvoir la confiance entre les parties prenantes.
Cependant, l'adoption de la blockchain dans les institutions éducatives ne se fait pas sans défis. Des problèmes tels que la scalabilité, la réglementation et les coûts initiaux de mise en œuvre doivent être pris en compte. De plus, l'acceptation et la compréhension de cette technologie par tous les acteurs impliqués sont des éléments cruciaux pour son succès.
En somme, la blockchain a le potentiel de transformer en profondeur la manière dont les institutions éducatives opèrent, en améliorant la transparence, la sécurité et l'efficacité. Si les défis sont relevés de manière adéquate et que les avantages sont correctement exploités, la blockchain pourrait ouvrir la voie à une nouvelle ère et passionnante de l'éducation,

Chapitre 3

Analyse et conception

3.1 Introduction

Ce chapitre fournit une description du processus de développement, du langage de modélisation sélectionné, ainsi que des besoins fonctionnels et non fonctionnels de l'application. Un diagramme de cas d'utilisation global est établi, accompagné d'une description détaillée des cas d'utilisation. Enfin, la conception de notre application est présenté par le diagramme de séquence, suivi du diagramme de classe.

3.2 Processus de développement

Il existe plusieurs processus de développement logiciel, telles que UP (Unified Process), RUP (Rational Unified Process), 2TUP (2 Track Unified Process), XP (Extreme Programming), etc. Selon la nature de notre projet, nous avons choisi d'utiliser la méthode UP, qui est la plus adaptée à la réalisation de notre application.

3.2.1 Processus Unifié

Le processus unifié (UP) est un processus de développement logiciel basé sur UML. Il est itératif et incrémental, avec une approche centrée sur l'architecture, les cas d'utilisation et la gestion des risques. Il offre une solution idéale pour résoudre les problèmes fréquemment rencontrés par les développeurs. UP est un modèle de processus qui peut être adapté à une grande variété de systèmes logiciels, à différents domaines d'application, types d'entreprises, niveaux de compétences et tailles d'entreprises [30].

3.2.2 Caractéristiques de Up

1. Itératif

Chaque étape de la méthodologie UP est composée d'itérations, qui représentent un cycle complet de développement logiciel, de la collecte des exigences à la mise en œuvre et aux tests [30].

2. UP est centré sur l'architecture

Dès le début du processus, l'architecture du système est envisagée. Elle est dérivée des exigences métier exprimées par les utilisateurs et d'autres parties prenantes, reflétées par les cas d'utilisation [30].

3. UP est piloté par les cas d'utilisation d'UML

L'objectif principal d'un système informatique est de répondre aux besoins des clients. Le processus de développement est donc axé sur l'utilisateur.
Les cas d'utilisation permettent d'illustrer les besoins, de détecter et de décrire les exigences fonctionnelles du point de vue de l'utilisateur, formant ainsi un modèle de cas d'utilisation qui décrit les fonctionnalités complètes du système [30].

4. Axé sur les risques

Le processus unifié exige que l'équipe de projet se concentre sur la résolution des risques les plus critiques dès le début du cycle de vie du projet. Les livrables de chaque itération, en particulier dans la phase d'élaboration, sont sélectionnés de manière à traiter en premier les risques les plus importants [30].

3.2.3 Phases de UP

La méthode UP se base sur quatre phases :

- Analyse des besoins.
- Élaboration.
- Construction.
- Transition.

Chaque phase est à son tour décomposée en itérations de durée limitée (généralement entre 2 et 4 semaines). Chaque itération produit un système testé, intégré et exécutable.

3.2.4 Activités de UP

Chaque phase est constituée d'une succession d'activités. Les activités du processus UP sont les suivantes :

1. Expression des besoins

Compréhension et expression des besoins fonctionnels et non fonctionnels, ainsi que la création d'une liste d'exigences du Lycée Chouadaa Annani.

2. Analyse

Préparation à la conception, permettant d'obtenir une compréhension des besoins et des exigences liées a la gestion d'absences .

3. Conception

Acquisition d'une compréhension approfondie des langages de programmation utilisés : HTML, CSS, JavaScript pour la création du frontend de l'application, PHP et Solidity pour son backend. Détermination de la manière de résoudre le problème posé, y compris la définition des principales interfaces (connexion Accueil, profil de chaque utilisateur, interface d'ajout d'absences).

4. Implémentation

Construction des programme et des contrats intelligents en utilisant les langages de programmation déja citer dans la phase de conception.

5. Tests

Vérification des résultats de l'implémentation de toutes les exigences, en testant la construction et en assurant une intégration correcte de ces composants logiciels : Metamask et Ganache.

3.3 Langage de modélisation

Comme le processus unifié exige l'utilisation d'UML (Unified Modeling Language), notre modélisation de la solution se fera en utilisant des diagrammes UML.

3.3.1 UML

Le langage de modélisation unifié (UML) est sous l'entière responsabilité de l'OMG (Object Management Group) [32]. UML est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, concevoir des solutions et communiquer des points de vue et il aide à décrire et à concevoir des systèmes logiciels, en particulier des systèmes logiciels fondé sur le style orienté objet(OO), il est dit universel car il est indépendant des langages de programmation. UML2 comporte treize diagrammes. Pour la modélisation de notre système, nous utilisons les trois diagrammes fondamentaux suivants [14] :

- Le diagramme de cas d'utilisation.
- Le diagramme de séquence.
- Le diagramme de classes.

3.4 Spécification des besoins

Les besoins de notre application se divisent en deux types : besoins fonctionnels et besoins non fonctionnels.

3.4.1 Besoins fonctionnels

Le système à réaliser comportera un ensemble de fonctionnalités qui doivent être mises en relation avec un ensemble de besoins utilisateur.
L'application pour la gestion des absences des élèves dans un lycée doit accomplir les traitements suivants :

- **Enregistrement des absences :** Le système doit permettre aux enseignants d'enregistrer facilement les absences des élèves en classe de manière immuable et transparente, afin d'éviter toute falsification ou modification ultérieure. Il s'agit d'une application web, où les enseignants peuvent entrer le nom de l'élève, la date et le motif de l'absence.
- **Notification des absences :** Le système doit être en mesure de notifier automatiquement les parents ou les tuteurs des élèves absents. En utilisant des smart contacts ou les notifications peuvent être envoyées par SMS directement et doivent inclure le nom de l'élève, la date et le motif de l'absence. Il doit être possible pour les parents ou les tuteurs de confirmer ou de justifier l'absence de leur enfant.
- **Consultation des absences :** Le système doit permettre aux utilisateurs de suivre les absences des élèves. Les données collectées doivent être stockées de manière sécurisée et accessible aux personnes autorisées. Les enseignants doivent être en mesure de voir l'historique des absences d'un élève et les problèmes lier à ces absences.
- **Avoir des rapport d'absence :** Le système doit être en mesure de produire des rapports sur les absences des élèves pour les enseignants, l'administration et les parents. Les rapports doivent inclure les tendances d'absentéisme, les absences répétées, les motifs d'absence, le pourcentage des absences, et les mesures prises pour remédier à cette absentéisme.
- **Vérification des justificatifs d'absence :** Le système doit permettre de vérifier l'authenticité des justificatifs d'absence fournis par les parents ou les élèves, en les enregistrant également.
- **Gérer les droits d'accès à l'information :** Le système doit permettre de gérer les droits d'accès à l'information concernant les absences des élèves, en garantissant que seuls les utilisateurs autorisés peuvent accéder à ces informations.

- **Confidentialité des données :** Il est important de garantir la confidentialité des données des élèves, notamment leurs informations personnelles, leurs justificatifs d'absence et leur historique d'absences, en utilisant des mécanismes de chiffrement.
- **Intégrité des données :** Il est important de s'assurer que les données relatives aux absences des élèves ne sont pas modifiées ou altérées de manière non autorisée.
- **Disponibilité :** Le système de gestion des absences doit être disponible en permanence, afin de permettre une gestion efficace des absences en temps réel.
- **Sécurité :** Le système doit être en mesure d'identifier les élèves absents en cas d'urgence. Les informations sur les absences doivent être accessibles rapidement et facilement accessibles aux enseignants et à l'administration en cas de situation d'urgence, pour s'assurer que tous les élèves sont en sécurité et qu'ils sont pris en charge.

3.4.2 Besoins non fonctionnels

- **Utilisabilité et Convivialité :** Le système de gestion des absences doit être facile à utiliser pour les différentes parties prenantes, notamment les élèves, les enseignants et les parents.
- **Adaptabilité :** Le système de gestion des absences doit être en mesure de s'adapter aux changements réglementaires et aux besoins spécifiques du lycée.
- **Évolutivité :** Le système de gestion des absences doit être en mesure de s'adapter à l'évolution des besoins en matière de gestion des absences, notamment en cas de changement dans la politique du lycée ou de l'évolution des normes en matière de gestion des données.
- **Interopérabilité :** Le système de gestion des absences doit être compatible avec d'autres systèmes utilisés par le lycée, afin de permettre une intégration aisée avec d'autres outils et logiciels de gestion.
- **Performance :** Le système de gestion des absences doit être suffisamment performante pour permettre une gestion en temps réel et une réponse rapide et efficace.
- **Faciliter la collaboration entre les enseignants :** Le système peut faciliter la collaboration entre les enseignants en leur permettant d'accéder aux informations concernant les absences des élèves de manière transparente et sécurisée.

3.5 Analyse des besoins

L'analyse des besoins vise à identifier les exigences ainsi que les acteurs du système et les tâches associées à chacun. On parle également de cadrage de projet.

3.5.1 Identification des acteurs et leur rôle

Un acteur se réfère à une entité externe (comme un utilisateur humain, un dispositif matériel ou un autre système) qui interagit directement avec le système examiné. L'acteur a la capacité de consulter et/ou de modifier directement l'état du système en émettant et/ou en recevant des messages qui peuvent contenir des données [30].
L'étude préliminaire des besoins fonctionnels a révélé la présence des acteurs suivants :
Les élèves :
Les élèves sont les principaux acteurs dans la gestion de leur propre absence. Ils doivent informer leur école de toute absence et ils peuvent également fournir des justificatifs pour leur absence (un certificat médical ou une autre justification valable), qui seront ensuite vérifiés par le personnel administratif.
Les enseignants :
Les enseignants sont responsables de la prise de présence et de l'enregistrement des absences des élèves. Ils peuvent vérifier les justificatifs fournis par les élèves et approuver ou rejeter leur validité et consulter l'historique d'absences des élèves. Ils peuvent également informer les élèves de la politique d'assiduité de l'école et des conséquences d'une absence non justifiée.
Les parents ou tuteurs :
Les parents ou tuteurs sont souvent informés des absences de leur enfant par l'école. Ils peuvent aussi signaler l'absence de leur enfant à l'école ou justifier une absence. Ils peuvent également être informés de la politique d'assiduité de l'école et des conséquences d'une absence non justifiée.
Les administrateurs d'école :
Les administrateurs d'école sont chargés de veiller à l'application de la politique d'assiduité de l'établissement. Leur rôle pourrait impliquer la gestion des dossiers d'absence et la collecte d'informations sur les absences en vue de les documenter, ainsi que la prise en charge des conséquences découlant des absences non justifiées. Ils ont également la possibilité de collaborer avec les enseignants pour élaborer des stratégies visant à réduire les absences et de communiquer avec les parents.

3.5.2 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les utilisations requises d'un système, ou ce qu'un système est supposé faire. La Figure 3.1 présente le diagramme de cas d'utilisation de notre application.

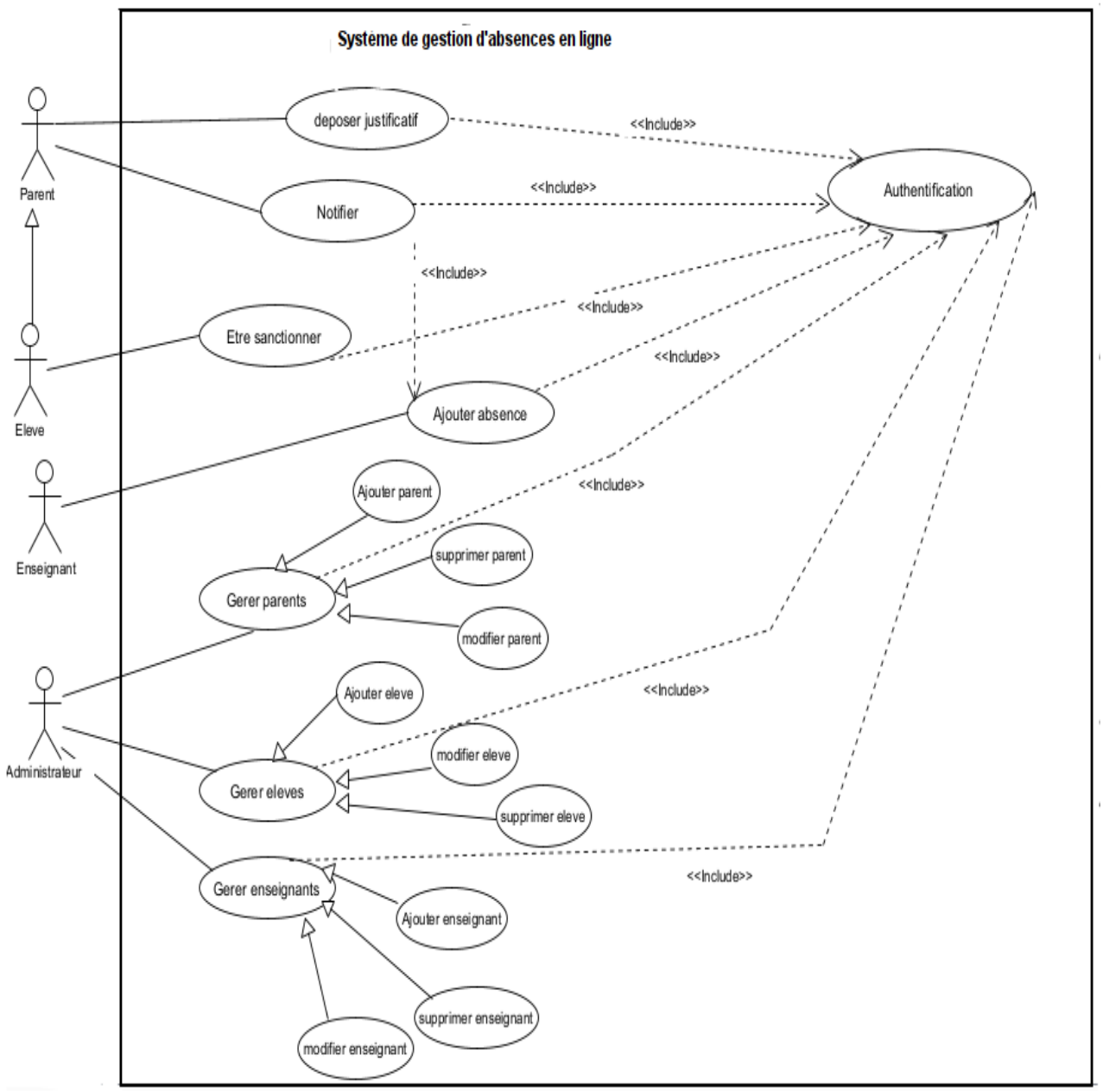


FIGURE 3.1 – Diagramme de cas d'utilisation

3.6 Description textuelle des cas d'utilisation

Afin d'obtenir une meilleure compréhension de notre système et de ses interactions avec les utilisateurs, nous fournirons dans les tableaux ci-dessous une détaillisation des scénarios des cas d'utilisation.

3.6.1 S'authentifier

Nom	S'authentifier
But	Permettre à l'utilisateur de se connecter et d'accéder à son profil
Acteurs principales	Administrateur, élève, parent, enseignant
Séquencement	Le cas d'utilisation commence lorsque l'utilisateur souhaite accéder à son profil
Pré-condition	L'utilisateur possède un compte
Scénario nominal	1-L'utilisateur saisie son Id et son mot de passe et clique sur le bouton **se connecter** 2-Le système vérifie les informations saisies 3-Le système vérifie l'existence du compte dans la BDD 4-Le système affiche le profil d'utilisateur
Enchaînement alternatif	**A1 : le compte n'existe pas dans la BDD** L'enchaînement démarre après le point 3 de la séquence nominale : 4-Le système indique que les informations saisies n'appartiennent à aucun compte enregistré dans la BDD 5-Le système réaffiche de nouveau la page de connexion La séquence nominale reprend au point 1.
Enchaînement d'exception	**E1 : les informations saisies sont incorrectes** L'enchaînement démarre après le point 2 de la séquence nominale : 3-Le système indique que l'Id et/ou le mot de passe saisie est incorrecte en affichant un message d'erreur
Post-conditions	L'utilisateur s'authentifie

TABLE 3.1 – Description du cas d'utilisation (S'authentifier)

3.6.2 Consulter les absences

Nom	Consulter les absences
But	Permettre à l'utilisateur de voir la liste des élèves absents
Acteurs principales	Administrateur, élève, parent, enseignant
Séquencement	Le cas d'utilisation commence lorsque l'utilisateur accède à son profil et clique sur **Liste d'absences**
Pré-condition	L'utilisateur s'authentifie
Scénario nominal	1-L'utilisateur accède à son profile et clique sur le bouton **liste d'absents** 2-Le système affiche la liste des absences enregistrées
Post-conditions	Liste d'absences affichée

TABLE 3.2 – Description du cas d'utilisation (Consulter les absences)

3.6.3 Ajouter absence

Nom	Ajouter absence
But	Permettre à l'enseignant de signaler et enregistrer l'absence d'un élève
Acteurs principales	Enseignant
Séquencement	Le cas d'utilisation commence lorsque l'enseignant souhaite enregistrer une absence
Pré-condition	L'enseignant s'authentifie
Scénario nominal	1-L'enseignant accède à son profile et clique sur le bouton et clique sur le bouton **Ajouter une absence** 2-Le système affiche le formulaire à remplir 3-L'enseignant rempli tous les champs clique sur le bouton **enregistrer** 4-Le système système vérifie la saisie 5-Le système vérifie l'existence de l'élève signalé comme absent dans la BDD 6-Le système enregistre la transaction
Enchaînement alternatif	**A1 : les informations saisies sont incorrectes** L'enchaînement démarre après le point 4 de la séquence nominale : 5-Le système système indique que les informations sont incorrectes ou incohérentes La séquence nominale reprend au point 2 **A2 : L'élève n'existe pas dans la BDD** L'enchaînement démarre après le point 5 de la séquence nominale : 6-Le système affiche un message indiquant que les informations saisies n'existent pas dans la base de données La séquence nominale reprend au point 2
Post-conditions	L'absence enregistrée

TABLE 3.3 – Description du cas d'utilisation (Ajouter absence)

3.6.4 crée un compte

Nom	crée un compte
But	Créer des comptes aux utilisateurs
Acteurs principales	Administrateur
Séquencement	Le cas d'utilisation commence lorsque l'administrateur accède à son profil et clique sur **Ajouter un compte**
Pré-condition	L'administrateur s'authentifie
Scénario nominal	1-L'administrateur accède à son profile et clique sur le bouton **Ajouter un compte** 2-Le système affiche le formulaire à remplir 3-L'administrateur rempli formulaire et clique sur le bouton **valider** 4-Le système vérifie la saisie 5-Le système vérifie si les informations saisies n'appartenant pas à un compte déjà créé 6-Le système mis à jour la BDD en ajoutant le nouveau compte créer
Enchaînement alternatif	**A1 : les informations saisies sont incorrectes** L'enchaînement démarre après le point 4 de la séquence nominale : 5-Le système système indique que les informations sont incorrectes ou incohérentes La séquence nominale reprend au point 2 **A2 : Le compte existe déjà dans la BDD** L'enchaînement démarre après le point 5 de la séquence nominale : 6-Le système système affiche un message indiquant que les informations saisies appartenant à un compte déjà créé La séquence nominale reprend au point 2
Post-conditions	Le compte est ajouté

TABLE 3.4 – Description du cas d'utilisation (Ajouter compte)

3.6.5 Supprimer un utilisateur

Nom	Supprimer
But	Permettre à l'administrateur de supprimer un utilisateur
Acteurs principales	Administrateur
Séquencement	Le cas d'utilisation commence lorsque l'administrateur souhaite supprimer un utilisateur
Pré-condition	L'administrateur s'authentifie
Scénario nominal	1-L'administrateur consulte la liste des utilisateurs en cliquant sur le bouton **liste d'utilisateur** 2-Le système système affiche la liste des utilisateurs 3-L'administrateur choisi l'utilisateur en question et clique sur **Supprimer** 4-Le système affiche une boite de dialogue demandant de valider l'action 5-L'administrateur clique sur **Valider** 6-Le système supprime l'utilisateur
Post-conditions	Utilisateur supprimé.

TABLE 3.5 – Description du cas d'utilisation (Supprimer utilisateur)

3.6.6 Déposer justificatif

Nom	Déposer justificatif
But	Permettre à l'utilisateur de justifier une absence
Acteurs principales	Parent, élève
Séquencement	Le cas d'utilisation commence lorsque l'utilisateur souhaite déposer un justificatif
Pré-condition	L'utilisateur s'authentifie
Scénario nominal	1-L'utilisateur accède à la rubrique **Notifications** 2-Le système affiche l'ensemble des absences associé au propriétaire du compte 3-L'utilisateur choisi l'absence à justifier et clique sur le bouton **joindre une justification** 4-Le système affiche l'interface approprié 5-L'utilisateur ajoute la justification et clique sur **Valider** 6-Le système ajoute le critère **justifiée** à l'absence en question
Enchaînement d'exception	**E1 : la rubrique notifications est vide** L'enchaînement démarre après le point 1 de la séquence nominale : 2-Le système affiche ce message **aucune absence n'est enregistré**
Post-conditions	Absence justifiée

TABLE 3.6 – Description du cas d'utilisation (Déposer justificatif)

3.7 Conception

La conception est un processus créatif et rigoureux qui permet de donner vie à une idée ou à un projet. La conception repose sur la recherche, l'analyse et la réflexion. Elle nécessite une compréhension approfondie des besoins, des contraintes et des objectifs du projet.

3.7.1 Diagrammes de séquence

Un diagramme de séquence offre une représentation visuelle claire et détaillée des interactions dynamiques au sein d'un système. Il capture les différentes étapes d'un processus ou d'un scénario, en illustrant les actions, les messages et les réponses entre les entités impliquées. Il permet de comprendre le flux d'exécution d'un système, d'identifier les interactions clés et de détecter d'éventuels problèmes de synchronisation ou de communication.

1-Diagramme de séquence "Authentification"

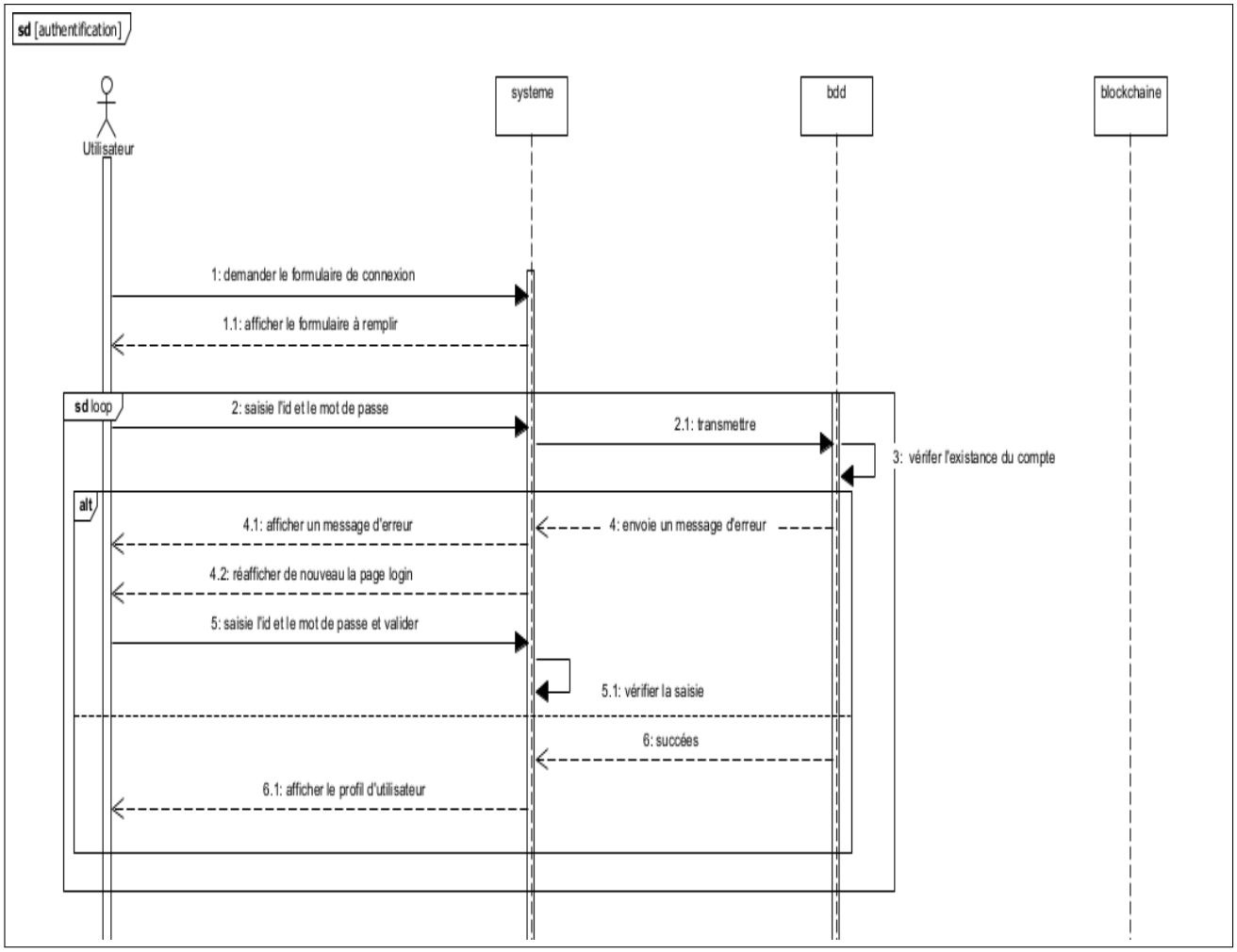


FIGURE 3.2 – Diagramme de séquence (Authentification)

2-Diagramme de séquence ”Consulter les absences”

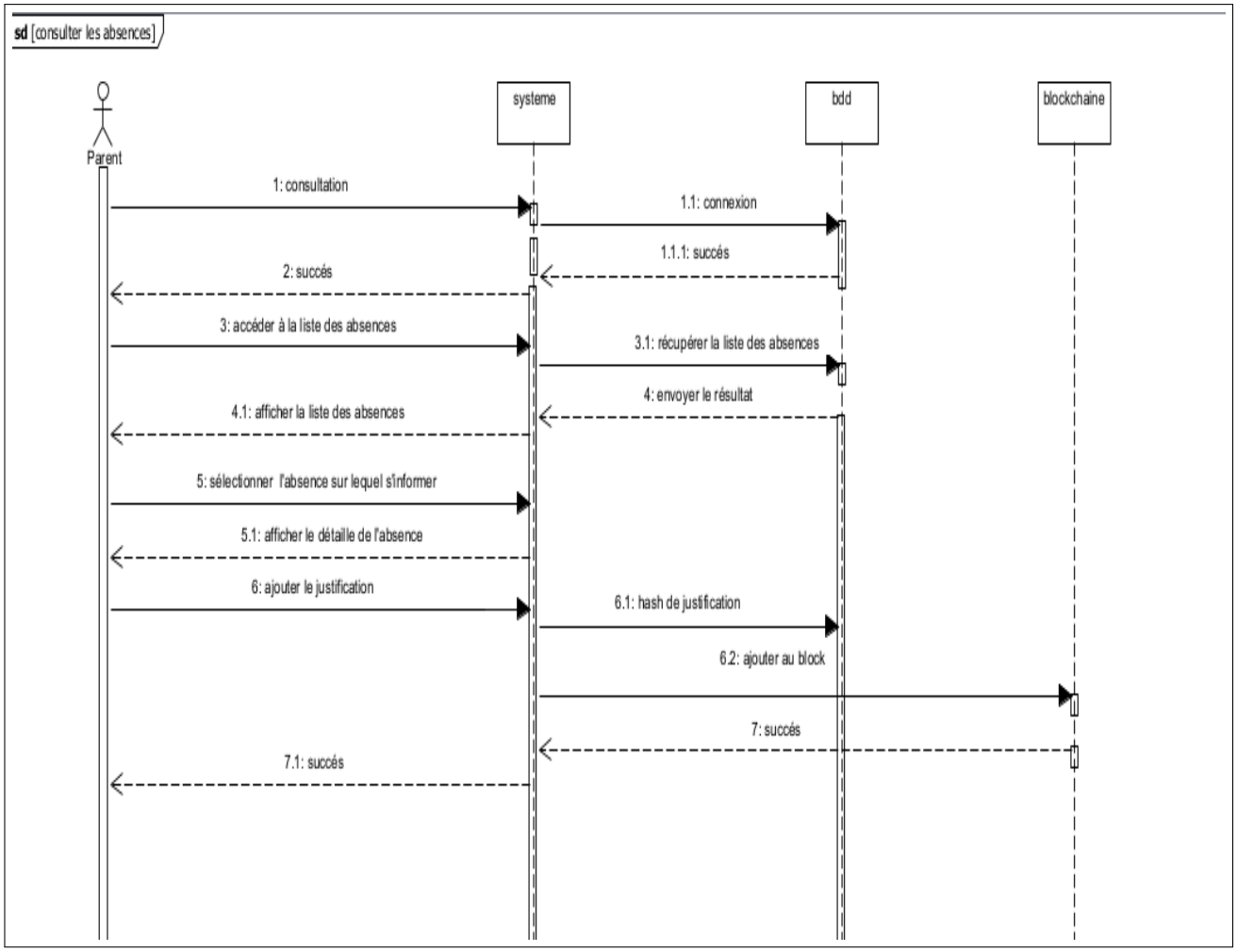


FIGURE 3.3 – Diagramme de séquence (Consulter absences)

4-Diagramme de séquence ”Ajouter absence”

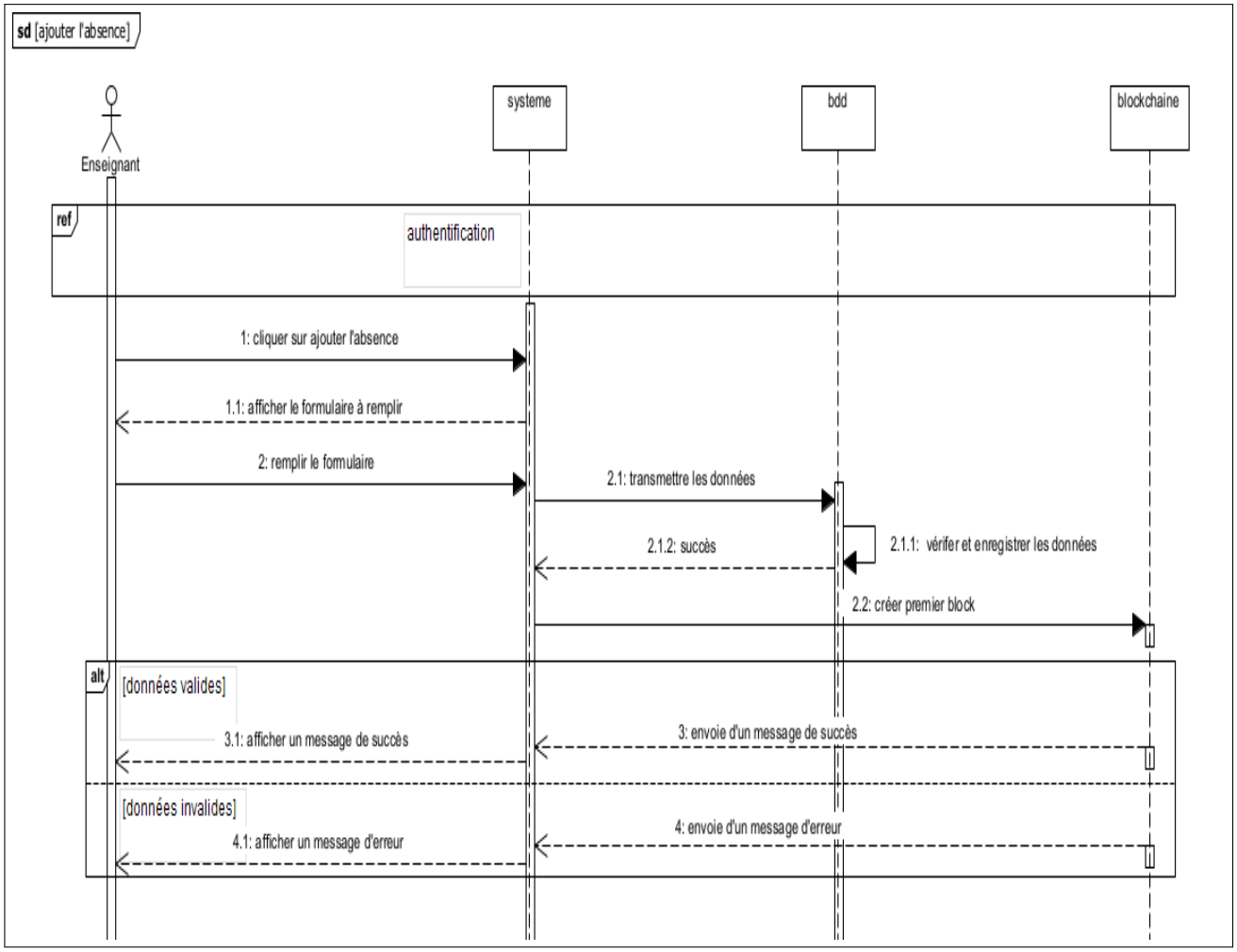


FIGURE 3.4 – Diagramme de séquence (Ajouter absences)

5-Diagramme de séquence ”Crée un compte”

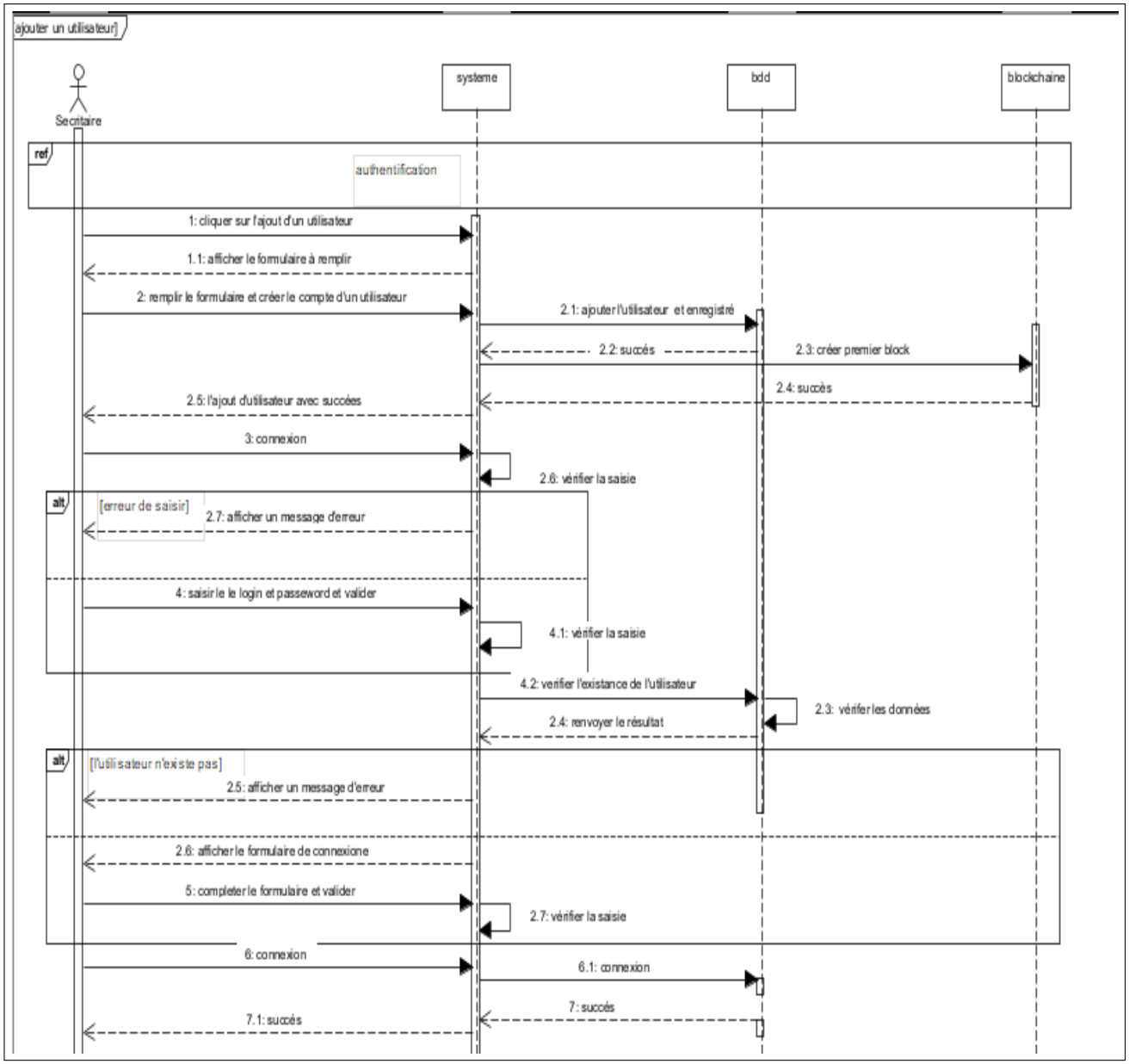


FIGURE 3.5 – Diagramme de séquence (Crée un compte)

6-Diagramme de séquence ”Supprimer un utilisateur”

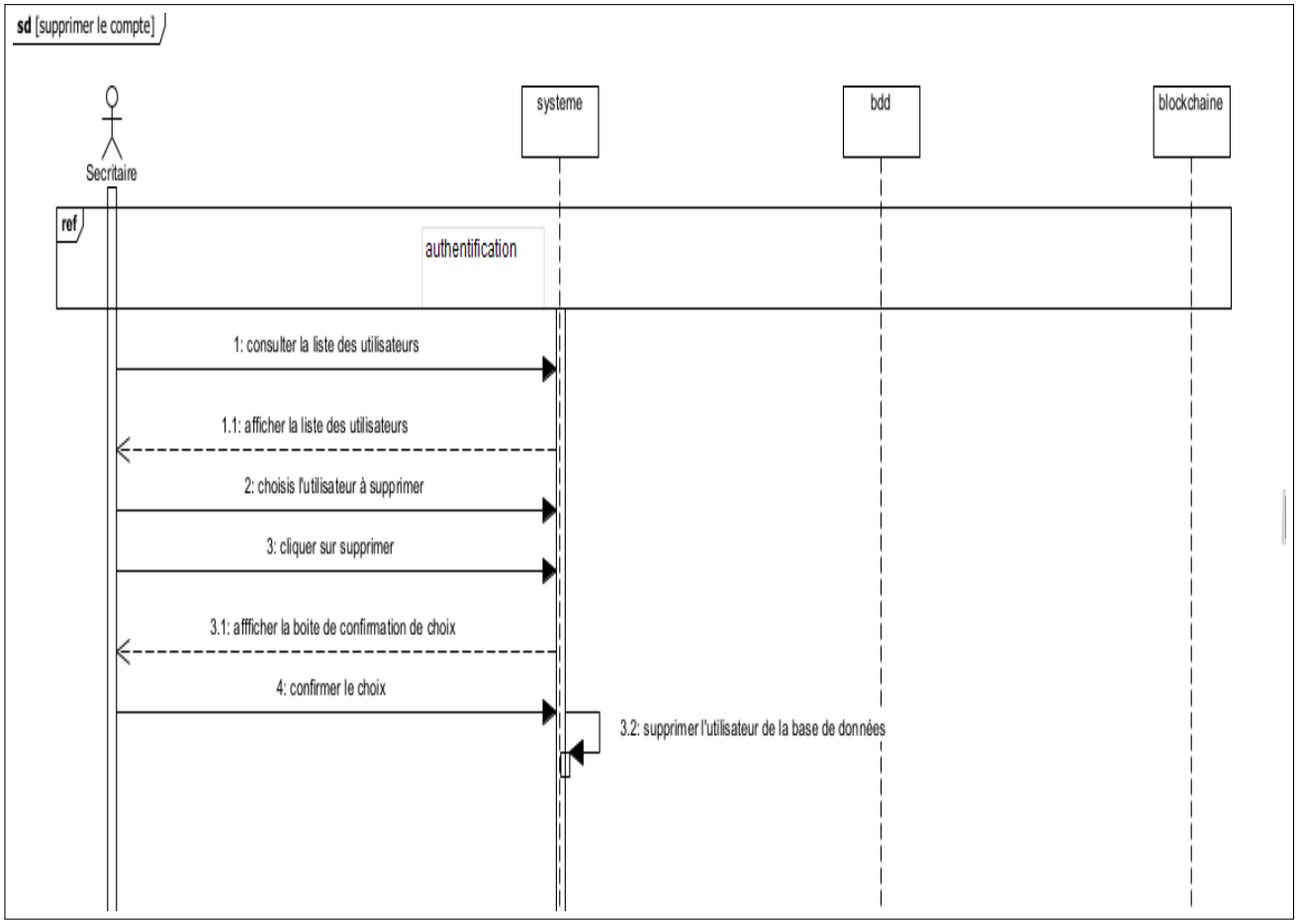


FIGURE 3.6 – Diagramme de séquence (Supprimer un utilisateur)

7-Diagramme de séquence "Déposer justificatif"

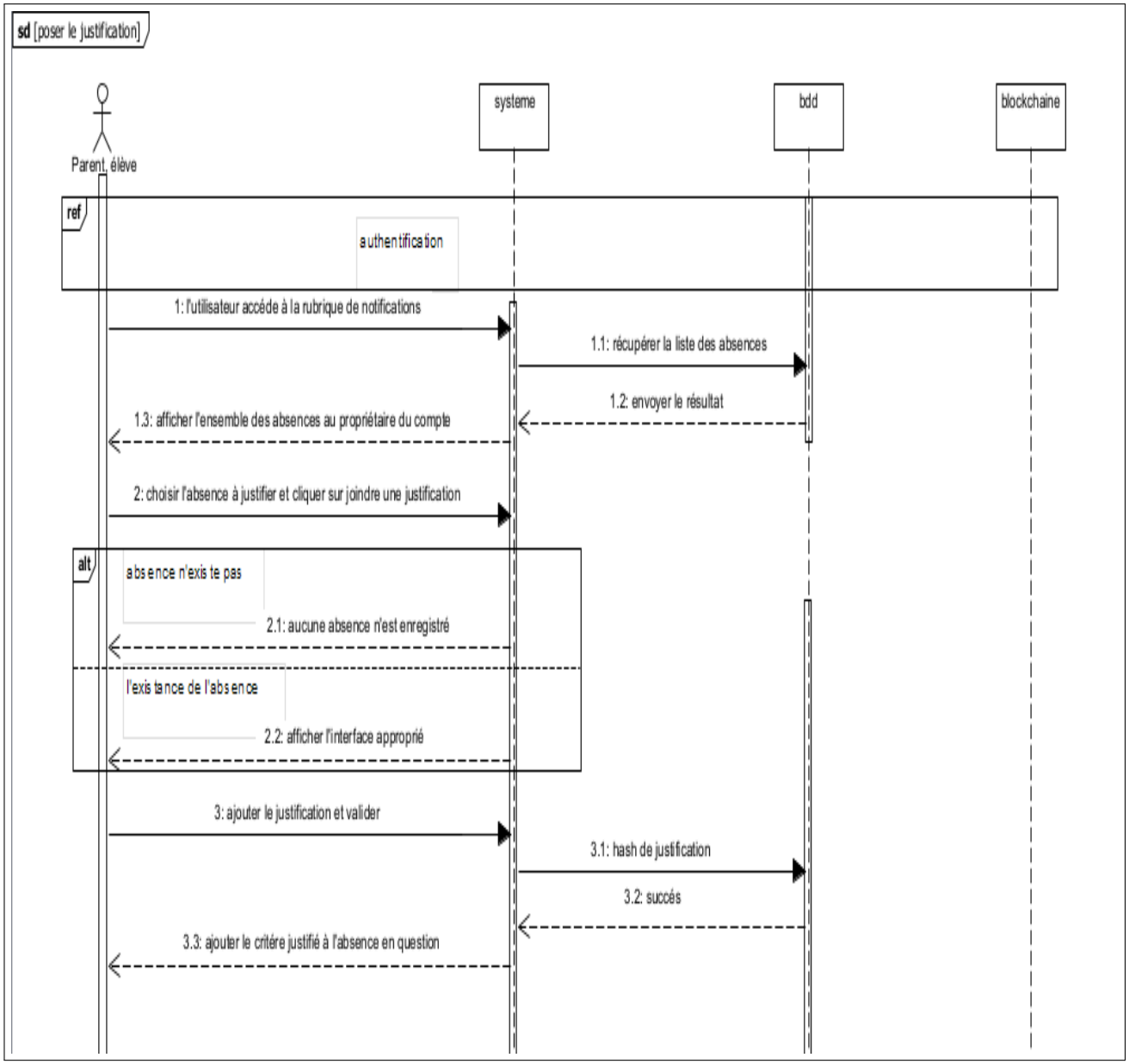


FIGURE 3.7 – Diagramme de séquence (Déposer justificatif)

3.7.2 Dictionnaire de données

Le tableau 3.2 d'écrit et explique toutes les données relatives aux classes de notre application.

Classe	Attributs	Type	Désignation	Méthodes	Responsabilité
Utilisateur	id	int	ID de l'utilisateur	Consulter_Absence()	
	nom	string	le nom de l'utilisateur		
	prenom	string	prenom de l'utilisateur		
	tel	int	numéro de téléphone de l'utilisateur		
	AdressU	string	Adresse de l'utilisateur		
	MotDePasse	string	Mot de passe de l'utilisateur		
	Email	string	L'email de l'utilisateur		
Parent				Depo_justificatif()	
Administrateur				gere_comptes()	gere_parents gere_eleves gere_enseignants
Enseignant	module	string	Le module enseigné par cet enseignant	Ajouter_absence()	
	numero_salle	int	numéro de la salle ou se déroule la séance		
	id_elev	int	ID de l'élève		
Eleve	Niveau_etude	string	Niveau d'étude de l'élève	Etre_sanctionne()	
	date_naissance	date	date de naissance de l'élève		
	specialite	string	La spécialité de l'élève		
	sexe	Genre	sexe de l'élève		
	classe	string	La classe a la quelle appartiens l'élève		
	id_Abs	int	ID de l'absence de cet l'élève		
Salle	numero_salle	int	Numéro de salle ou se déroule la séance		
	type_salle	Type	Type de salle		
Absence	id_Abs	int	ID de l'absence	Liste_Absences	
	id_elev	int	ID de l'élève		
	id_ens	int	ID de l'enseignant		
Marquer	Date	date	Date de l'absence		
	Heure	string	heure de l'absence		

3.7.3 Diagramme de classes

Un diagramme de classe est un outil de modélisation utilisé pour représenter la structure statique d'un système logiciel. Le diagramme de classes occupe une place centrale dans le développement orienté objet pour visualiser et définir la structure du système. En phase de conception, le diagramme de classes représente la structure d'un code orienté objet ou, à un niveau plus détaillé, les modules du langage de programmation. Ce diagramme met en œuvre des classes qui contiennent des attributs et des opérations, et les relie entre elles par le biais d'associations ou de relations de généralisation. Il facilite la communication entre les concepteurs, les développeurs et les parties prenantes en fournissant une représentation graphique claire [30].
la Figure 3.8 illustre le diagramme de classes de notre système.

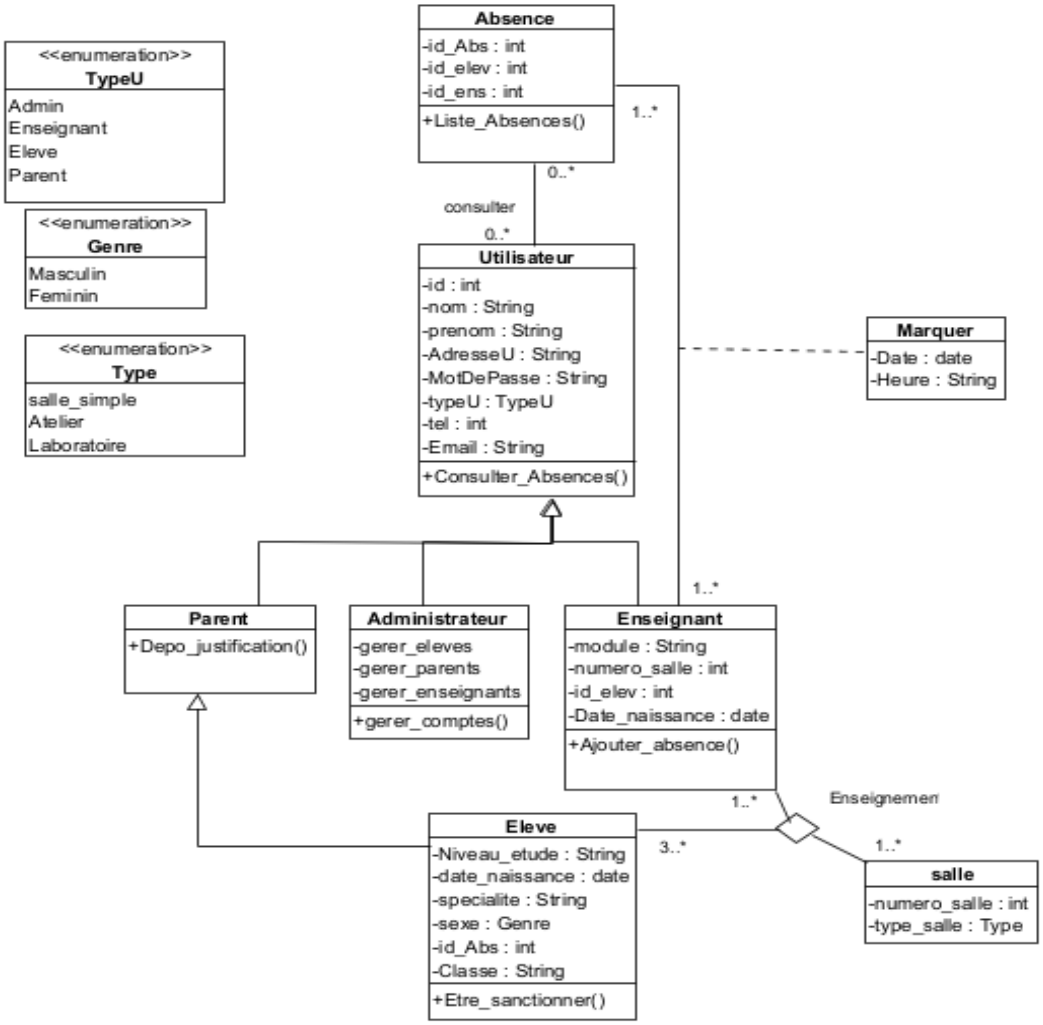


FIGURE 3.8 – Diagramme de classe

3.8 Modèle relationnel

Le modèle relationnel est un modèle de données qui organise les informations dans une base de données en utilisant des relations, également connues sous le nom de tables. Chaque relation est composée de plusieurs attributs, qui représentent les différentes caractéristiques des entités. Les relations sont liées entre elles par des clés primaires et des clés étrangères, permettant ainsi de définir des associations et des dépendances entre les entités. Ce modèle repose sur les principes de l'algèbre relationnelle et permet de représenter et de manipuler les données de manière structurée et cohérente.
Afin de procéder à l'implémentation d'une base de données relationnelle, nous avons appliqué les règles de passages sur notre diagramme de classes, ce qui a abouti au modèle relationnel suivant :

- Absence ($\underline{id\_abs}, \underline{\#id\_elev, \#id\_ens, \#numero\_salle}$)
- Enseignant ($\underline{id\_ens}, Nom, Prenom, Date\_naissance, AdresseU, MotDePasse, Tel, Module, \underline{\#id\_abs, \#numero\_salle}$)
- Marquer ($Date, Heure, \underline{\#id\_abs, \#id\_ens}$)
- Administrateur ($\underline{id\_ad}, Nom, Prenom, AdresseU, MotDePasse, Email, Tel$)
- Parent ($\underline{id\_p}, Nom, Prenom, AdresseU, MotDePasse, Tel, Email$)
- Eleve ($\underline{id\_elev}, Nom, Prenom, AdresseU, MotDePasse, Tel, Niveau\_etude, Date\_naissance, Email, Specialite, classe, Sexe, \underline{\#id\_abs}$)
- Salle ($\underline{numero\_salle}, Type\_salle$)
- Enseignement ($\underline{\#id\_elev, \#id\_ens, \#numero\_salle}$)
- Consulter ($\underline{\#id\_elev, \#id\_ens, \#id\_abs, \#id\_p, \#id\_ad}$)

3.9 Conclusion

Dans ce chapitre, nous avons introduit les exigences fonctionnelles et non fonctionnelles de notre application, ainsi que le langage de modélisation UML et notre approche de développement. Ensuite, nous avons entrepris la phase de spécification et d'analyse des besoins, ce qui nous a permis d'identifier les acteurs de notre application et les différents cas d'utilisation, suivis de descriptions textuelles détaillées. Enfin, nous avons abordé la phase de conception de notre application en présentant le diagramme de séquence et le diagramme de classes, ainsi que son dictionnaire de données.

Dans le prochain chapitre, nous passerons aux phases d'implémentation et de réalisation de notre système.

Chapitre 4

Implémentation

4.1 Introduction

Ce chapitre est consacré à la partie pratique de la mise en œuvre de notre application web assurant la gestion d'absence des élèves du lycée. Il comprend une description des outils de développement utilisés, tels que l'environnement de travail Visual studio code avec les langages de programmation HTML CSS PHP JavaScript et Solidity. Enfin, nous présenterons quelques interfaces de notre application.

4.2 Application web

Une application web est un logiciel ou un programme informatique accessible via un navigateur web. Contrairement aux applications traditionnelles qui nécessitent d'être installées sur un ordinateur ou un appareil mobile, une application web fonctionne directement à travers un navigateur internet. Elle utilise les technologies web telles que HTML, CSS et JavaScript pour fournir une interface utilisateur interactive et permettre aux utilisateurs d'accomplir diverses tâches en ligne

4.3 Environnement de développement logiciels

Les logiciels utilisés pour la réalisation du projet sont les suivants :

Visual studio code

Visual Studio Code (VS Code) est un éditeur de code source léger et puissant, développé par Microsoft. Il est largement utilisé par les développeurs pour écrire, modifier et déboguer du code dans divers langages de programmation. Doté d'une interface utilisateur intuitive et extensible, VS Code offre des fonctionnalités telles que la coloration syntaxique, l'autocomplétion intelligente, la gestion des versions grâce à l'intégration avec des systèmes de contrôle de version, ainsi que des fonctionnalités avancées comme le débogage interactif et la navigation au sein du

code. Son écosystème d'extensions permet aux développeurs d'ajouter des fonctionnalités et des langages supplémentaires, ce qui en fait un outil polyvalent pour répondre aux besoins variés des programmeurs [38].

Ganache

Ganache est un environnement de développement personnel et de test conçu spécifiquement pour les développeurs travaillant avec la technologie blockchain Ethereum. Il fournit une plate-forme locale de chaîne de blocs qui permet aux développeurs de créer et de déployer des contrats intelligents, d'exécuter des transactions et de simuler des interactions sur la blockchain Ethereum sans avoir besoin de déployer des contrats ou de dépenser de la vraie crypto-monnaie. Ganache facilite le processus de développement en offrant un environnement sandbox où les développeurs peuvent tester leurs applications et contrats intelligents en toute sécurité, en observant les effets de leurs actions en temps réel. Cela accélère le processus de développement, car il permet aux développeurs de détecter et de résoudre les problèmes plus rapidement tout en bénéficiant d'une expérience de développement réaliste et sécurisée [21].

4.3.1 Langage de programmation

Afin de développer notre système, nous avons choisi les langages suivants :

HTML

HTML, ou HyperText Markup Language (langage de balisage hypertexte), est le langage de base utilisé pour créer et structurer le contenu des pages web. Il s'agit d'un langage de balisage, ce qui signifie qu'il utilise des éléments appelés "balises" pour définir la structure et la présentation du contenu sur une page web. Les balises HTML sont utilisées pour indiquer au navigateur comment afficher différents types de contenu, tels que du texte, des images, des liens, des vidéos et plus encore [35, 36].

CSS

CSS, ou Cascading Style Sheets (feuilles de style en cascade), est un langage de programmation utilisé pour définir la présentation et la mise en forme visuelle des pages web écrites en HTML. Plutôt que de spécifier directement comment chaque élément doit être affiché, CSS permet de séparer le contenu et la mise en forme, ce qui facilite la personnalisation et la cohérence du design sur l'ensemble d'un site web. En utilisant des sélecteurs et des règles, les développeurs peuvent appliquer des propriétés telles que la couleur, la taille, la police, les marges et les espacements à différents éléments HTML [35].

JavaScript

JavaScript est un langage de programmation polyvalent largement utilisé pour rendre les pages web interactives et dynamiques. Intégré directement dans les navigateurs, JavaScript permet aux développeurs de créer des fonctionnalités telles que les effets visuels en temps réel, les animations, la validation de formulaires et les interactions utilisateur avancées. En complément du HTML et du CSS, JavaScript ajoute une couche de logique et d'interactivité aux sites web en permettant l'exécution de scripts côté client. Il permet également d'effectuer des requêtes vers des serveurs, ce qui permet aux applications web de récupérer et de manipuler des données sans avoir besoin de recharger la page web moderne [36].

PHP

PHP, acronyme de "Hypertext Preprocessor", est un langage de programmation côté serveur conçu spécifiquement pour le développement web. Il est principalement utilisé pour générer des pages web dynamiques en combinant du code PHP avec du HTML, permettant ainsi la création de sites web interactifs et basés sur des bases de données. PHP peut être intégré directement dans le code HTML pour traiter des données, effectuer des opérations de base de données, générer du contenu personnalisé en fonction des utilisateurs, et bien plus encore. En tant que langage serveur, PHP est exécuté sur le serveur web, ce qui signifie que les visiteurs du site ne voient que le résultat généré, sans avoir accès au code source PHP lui-même [33, 34].

Solidity

Solidity est un langage de programmation spécialement conçu pour écrire des contrats intelligents sur la plateforme Ethereum. Ce langage permet aux développeurs de définir les règles et le comportement de ces contrats intelligents, en spécifiant comment ils devraient réagir aux actions et aux événements. Il est basé sur la syntaxe de programmation de JavaScript et C++, et il est utilisé pour créer des applications décentralisées (DApps) et des protocoles financiers déployés sur la blockchain Ethereum [37].

Truffle

truffle est un framework qui fournit une suit d'outils permettant de développer des smart contracts Ethereum en utilisant le langage de programmation Solidity. Il joue un rôle crucial en facilitant l'interaction avec nos contrats intelligents et il permet d'effectuer des tests. De plus, Truffle facilite le développement de l'interface utilisateur côté client pour nos contrats, il offre la capacité de déployer ces smart contracts sur divers réseaux Ethereum [30].

Node.js

Node.js est un environnement de développement open-source basé sur le moteur JavaScript V8 de Google et une plateforme d'exécution côté serveur qui permet de créer et d'exécuter des applications web et des services réseau en utilisant JavaScript.// remarque : Pour développer des smart contracts nous devons configurer notre environnement par l'installation de Node Package Manager (NPM), fourni avec Node.js [12].

Web3

Web3, également appelé "Web3.js", est une bibliothèque JavaScript populaire utilisée pour interagir avec la blockchain Ethereum et d'autres blockchains compatibles avec Ethereum. Cette bibliothèque facilite la communication avec les nœuds Ethereum, ce qui permet aux développeurs de créer des applications décentralisées et des services qui exploitent les fonctionnalités de la blockchain. Celle-ci permet d'entrer l'adresse d'un Smart Contrat et d'appeler les fonctions qu'il contient, en passant éventuellement les paramètres nécessaires [21].

4.4 Quelques interfaces de l'application

Voici les interfaces principales de notre application nommée "Educhaine " (qui est une concaténation des deux termes éducation et Blockchain) :

Accueil

La Figure 4.1 représente l'interface Accueil qui est la premier page chargée lors de l'accès a l'application

FIGURE 4.1 – Interface d'accueil

Connexion

La Figure 4.2 représente la page de connexion qui permet aux utilisateurs d'accéder a leur profils

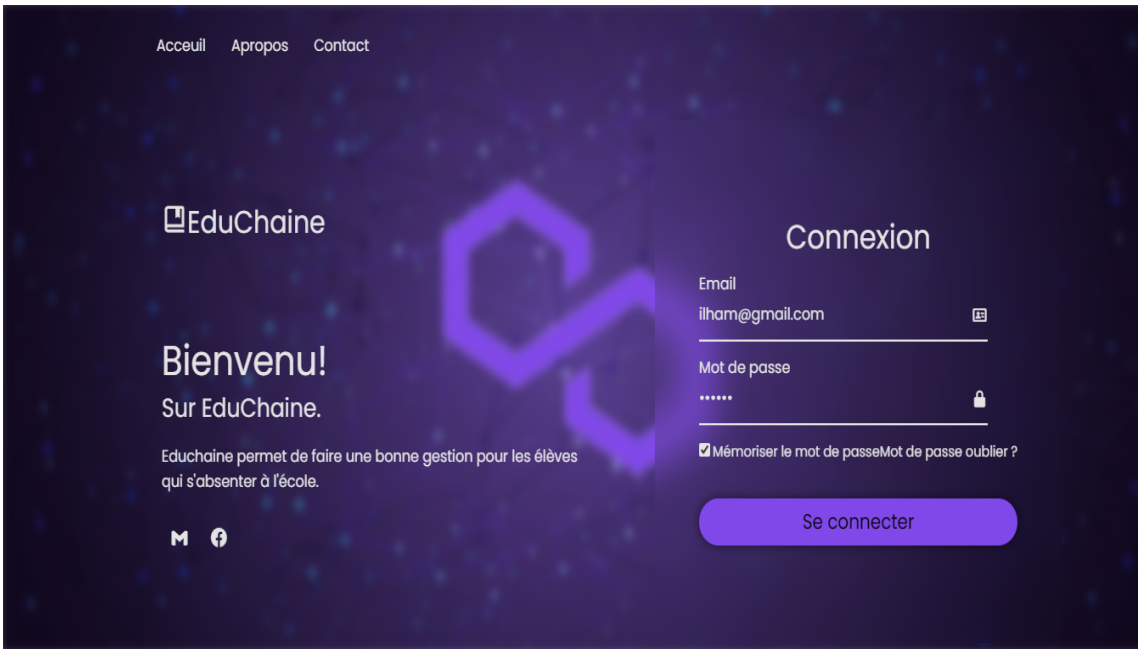


FIGURE 4.2 – Interface de connexion

Profile d'élève

La Figure 4.3 représente l'interface du profil de l'élève, elle lui permet de consulter l'ensemble des absences enregistrées dans le réseau

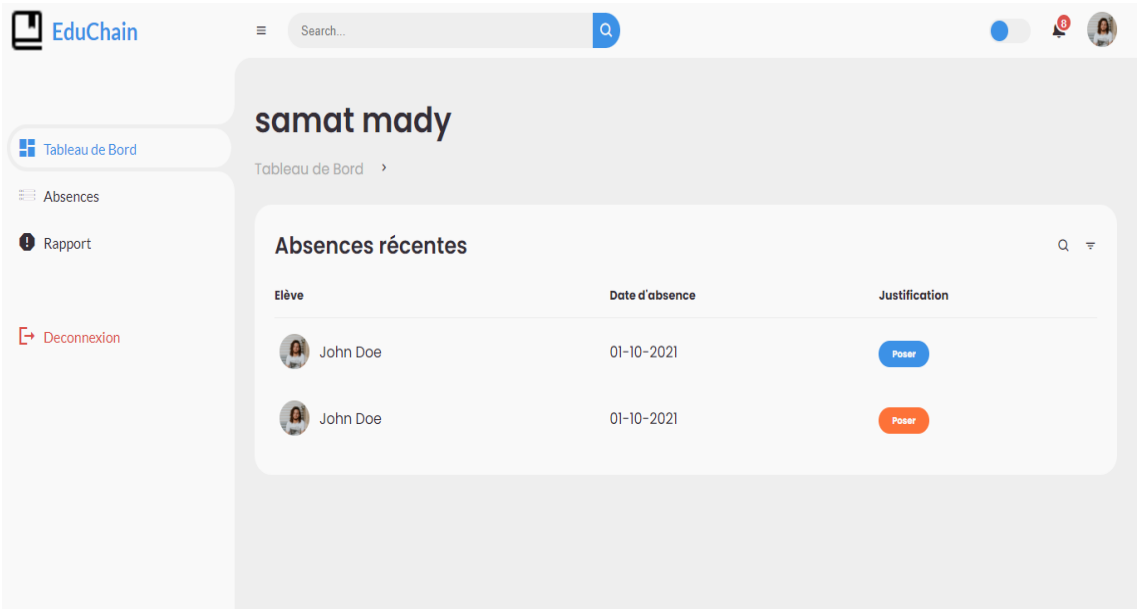


FIGURE 4.3 – Profil élève

Profile d'administrateur

La Figure 4.4 représente l'interface du profil de l'administrateur en mode sombre qui est applicable pour tous les autres profils, elle lui permet de consulter l'ensemble des absences enregistrées dans le réseau et de faire la gestion des utilisateurs (ajout, suppression et affichages des listes).

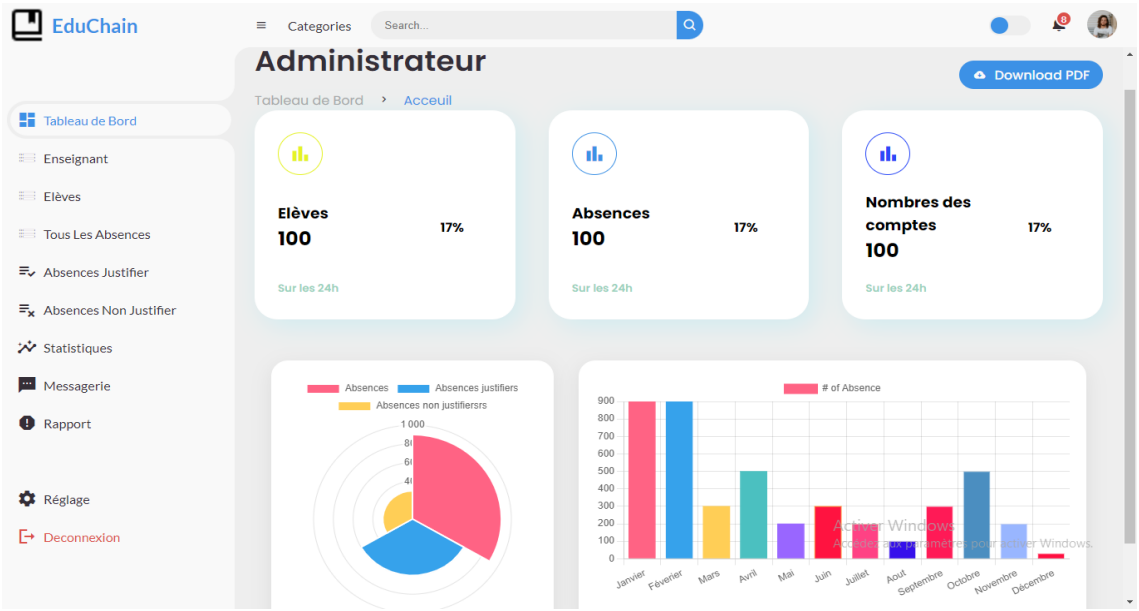


FIGURE 4.4 – Profil administrateur

Interface d'ajout d'utilisateur

La Figure 4.5 représente l'interface d'ajout d'utilisateur (exemple d'élève), elle permet de enregistré l'ensemble d'informations nécessaire.

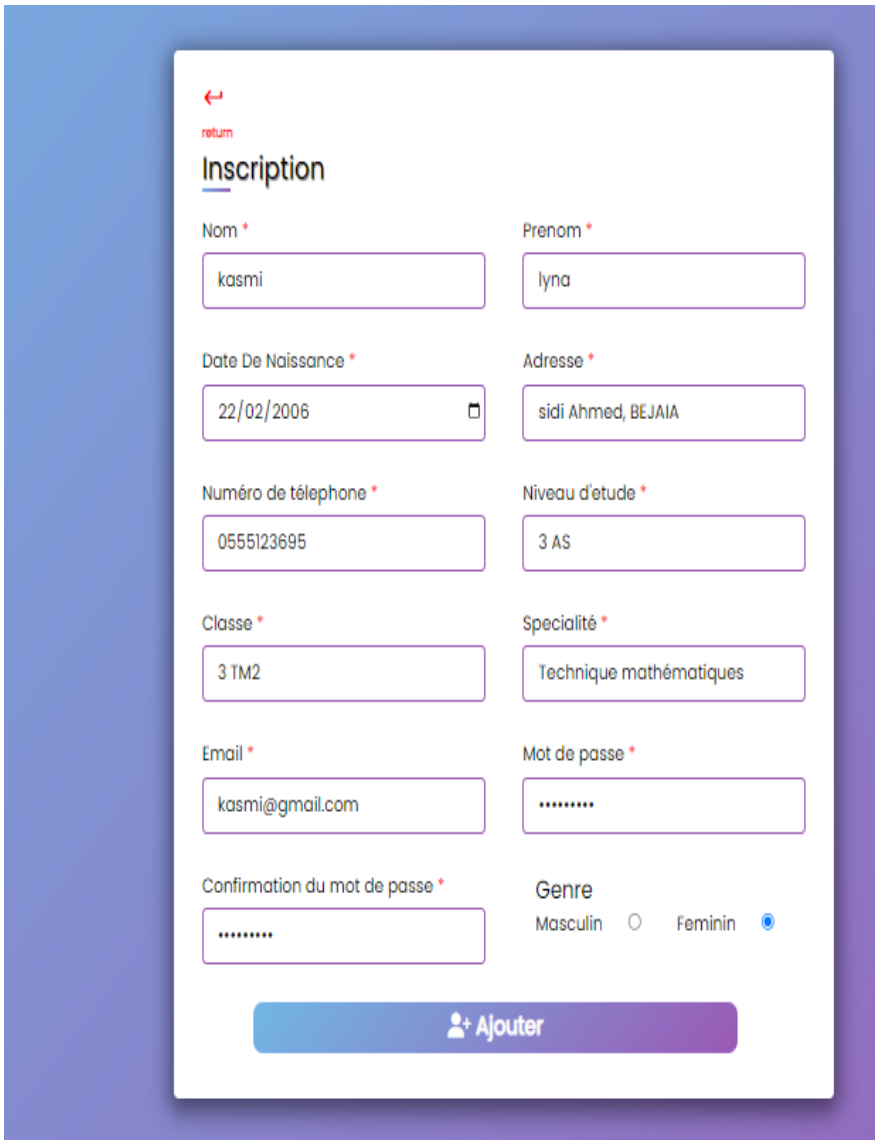


FIGURE 4.5 – Interface inscription

Profile de l'enseignant

La Figure 4.9 représente l'interface du profil de l'enseignant, elle lui permet d'ajouter des absences et de consulter l'ensemble des absences enregistrées dans le réseau.

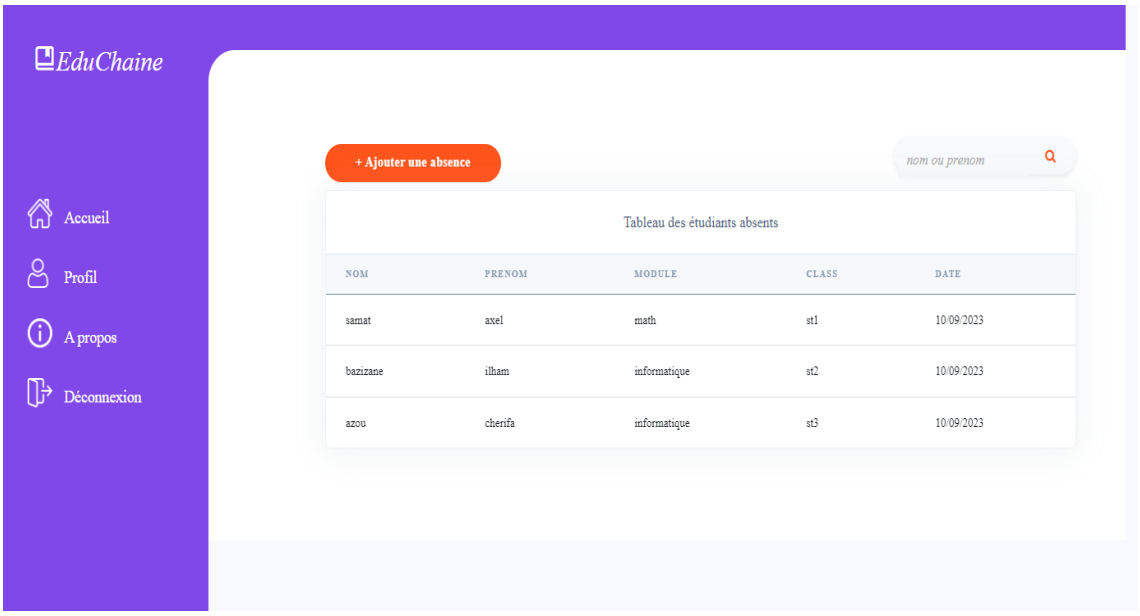


FIGURE 4.6 – Profil de l'enseignant

Profile de parent

La Figure 4.9 représente l'interface du profil de parent, elle lui permet de recevoir des notifications et de consulter l'ensemble des absences enregistrées dans le réseau.

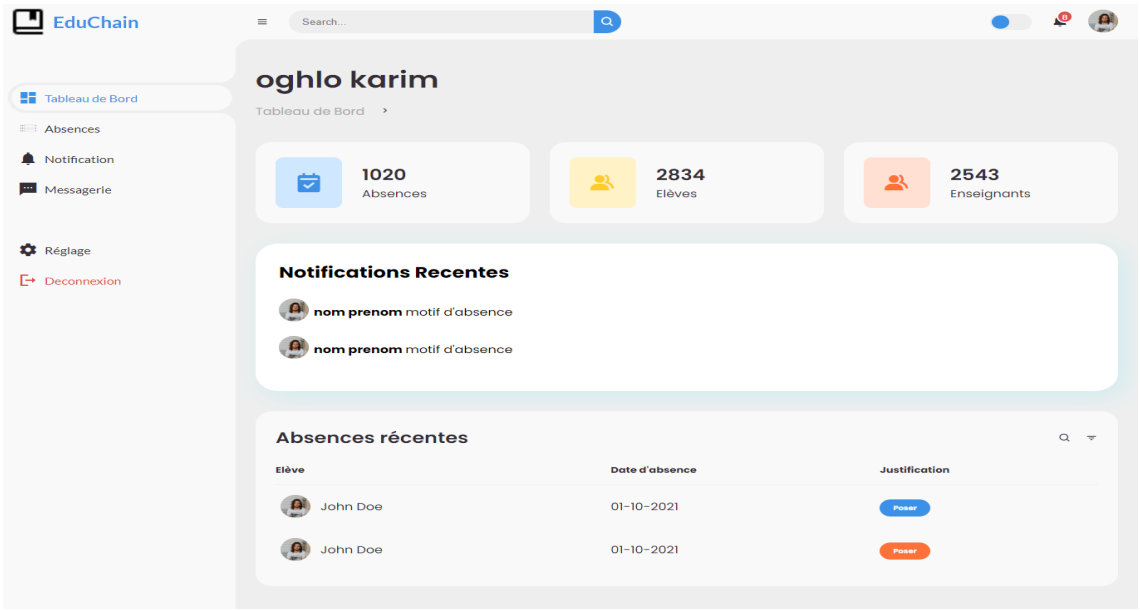


FIGURE 4.7 – Profil de parent

Interface d'ajout d'absences

La Figure 4.8 représente l'interface d'ajout d'absences, elle permet de enregistré le nom, le prénom, la classe de l'élève absence ainsi que le module enseigné lors de son absence, cette dernière sera enregistrée comme transaction horodaté dans l'un des blocs de la blockchain.

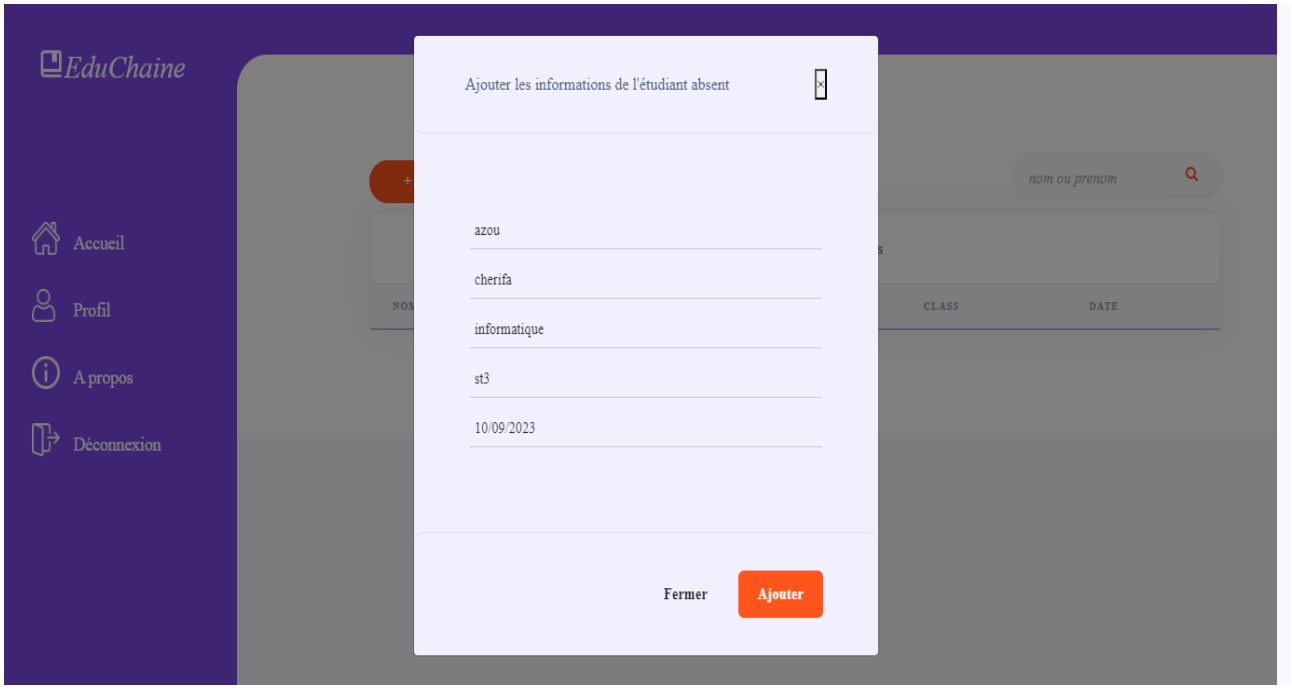


FIGURE 4.8 – Interface ajout absence

la page de A propos

La Figure 4.9 représente la page A propos de notre application elle explique les fonctionnalités de notre application.

FIGURE 4.9 – page A propos

4.5 Conclusion

En conclusion de ce chapitre, il ressort clairement que la mise en œuvre réussie d'un projet ou d'un système repose sur une planification minutieuse, une gestion efficace des ressources et une exécution précise des étapes techniques. Ce chapitre a mis en lumière l'importance de choisir les technologies appropriées, de développer un code propre et bien documenté, et de suivre des pratiques de programmation cohérentes. De plus, une collaboration étroite entre les membres de l'équipe et une évaluation continue sont essentielles pour résoudre rapidement les problèmes émergents et garantir la qualité du produit final. En comprenant les défis potentiels de l'implémentation et en adoptant des approches flexibles, il est possible de créer des solutions robustes et fonctionnelles qui répondent aux besoins spécifiques du projet.

Conclusion générale

La technologie de la blockchain a clairement démontré son efficacité dans le stockage et la transmission sécurisée et transparente d'informations dans divers domaines. Elle a introduit une nouvelle manière d'opérer en éliminant les intermédiaires et en maintenant la confiance des clients. La sécurité et la transparence des systèmes sont maintenues à travers différents concepts fondamentaux.

Dans notre cas, nous avons étudié la blockchain et exploité ces caractéristiques de décentralisation, de transparence et de sécurité pour répondre à notre besoin de développer une application web destinée à résoudre un problème majeur : la traçabilité des absences, une préoccupation commune à de nombreux établissements éducatifs et parents. C'est ainsi que le stockage et le partage des données relatives aux absences des élèves permettent une gestion sécurisée de cette problématique. En générant des rapports détaillés sur les taux d'absence, offrant ainsi une vue globale de la présence des élèves,tandis que les parents reçoivent des notifications en temps réel et peuvent justifier les absences de leurs enfants de manière transparente.Cette application assure que les élèves ne manquent pas de cours à l'insu de leurs parents, qui restent informés de toutes les activités de leurs enfants à l'école.

En somme, ce que nous avons mis en place représente simplement les prémices du développement de la technologie blockchain, qui marque le début d'une véritable révolution, notamment dans le secteur éducatif. La blockchain deviendra un élément indispensable pour l'éducation. Ses opportunités sont multiples et incitent le système éducatif à adopter cette technologie, car elle offre des solutions à de nombreux problèmes auxquels ce secteur est confronté. Parmi ces solutions, l'intégration de la blockchain dans la gestion des absences ouvre de nouvelles perspectives pour une gestion plus efficace, transparente et sécurisée de la présence des élèves dans les établissements éducatifs. Aussi la sécurisation des données des élèves en empêchant toute falsification ou suppression grâce à une base de données distribuée et décentralisée. La technologie blockchain pourrait contribuer à une transformation positive de l'éducation en renforçant la confiance et en améliorant la qualité de l'interaction entre les parties prenantes.

Bibliographie

[1] Taiana Laurenc. *Blockchain for Dummies*, Wiley Publishing Inc, first edition, Italie, (2018).

[2] Mancer M'hamed. *Conception et réalisation d'un modèle de Blockchain intelligent*, Master académique en informatique, université Mohamed Khider Biskra, Algérie, (2020).

[3] Ana Bakhoum. *La Blockchain pour la sécurisation des E-livret scolaire*, Master en informatique, université Assane seck de Ziguinchor, Sénégal, (2019).

[4] Marion Pignel. *La technologie Blockchain une opportunité pour l'économie sociale*, collection 'Rapport technique', économie sociale, sous la direction de Denis Stokkink, (2019).

[5] Sylvain Tessier. *Fonctionnement de la Blockchain et son intérêt pour le monde pharmaceutique*, Thèse de docteur en pharmacie, université de Bordeaux, (2019).

[6] Rahmani Rokia. *Les dossiers médicaux sur Blockchain*, Master académique en informatique, université Mohamed Khider – Biskra, Algérie, (2021).

[7] Reddaf Nassim, Raouache Yakoub. *Protection des données dans l'Internet des objets*, Master en informatique, université A/Mira de Béjaia, (2021).

[8] Bazizi Sonia, Beldjoudi Chabha. *Conception et réalisation d'une Blockchain cas d'étude : gestion du dossier de santé électronique*, Master en informatique, université A/Mira de Béjaia, (2020).

[9] Guillaume Buffet. *Comprendre la Blockchain, Livre blanc édité par U • uchange.co*, (janvier 2016).

[10] Valéria Faure-Muntian, Claude de ganay, députés, et Ronan le gleut. *L'Office parlementaire d'évaluation des choix scientifiques et technologiques sur les enjeux technologiques des Blockchains*, Rapport enregistré à la présidence de l'assemblée nationale et du Sénat, (20 juin 2018).

[11] Philippe Marrast. *Blockchain et Santé : perspectives d'applications et enjeux juridiques*, MCF Sciences de l'information et de la communication, Phd en Informatique, Toulouse, (2018).

[12] Sabrina Dellys, Sofia Benbouabdellah. *Applications de la technologie Blockchain*, Master en informatique , université Akli Mohand Oulhadj de Bouira, (2020).

[13] Léo Besançon. *Interopérabilité des systèmes Blockchains*, Thèse de doctorat en Informatique, université de Lyon, (2021).

[14] Pascal Roques. *Les cahiers du programmeur UML2 modéliser une application web*, 4°edition , université de Lyon, (2010).

[15] Sandrine Favre. *Une Blockchain pour les compétences et connaissances certifiées ouvertes et libres*, Maîtrise universitaire en sciences et technologies de l'apprentissage et de la formation, université de Genève, (2021).

[16] Castagnuozzo Marco. *Implémentation d'une Blockchain et d'un écosystème de smart contracts dans le domaine de l'éducation*, travail de Bachelor réalisé en vue de l'obtention du titre informaticien de gestion, université de Genève, (2021).

[17] Ferkal Karim, Chaibi Yassmina. *Conception et réalisation d'une application Web Service pour la gestion d'un cabinet medical*, Master en informatique, université A/Mira de Béjaia, (2017).

[18] Sarminto Mariana. *La technologie Blockchain : une opportunité pour l'atteinte des objectifs de développement durable*, Maîtrise en science politique, université du Québec à Montréal, (2021).

[19] Masseport Samuel. *Consensus Blockchain : incitation des utilisateurs d'un réseau à la participation et à la loyauté*, Thèse de docteur en informatique, université de Montpellier, (2021)

[20] Saba Saif. *Analysis of Blockchain consensus mechanisms*, Master of science, university of Turku, (2022).

[21] Guani Amina, Hammou Hadjar. *La Blockchain et son utilisation dans les chaînes d'approvisionnement*, Matser en informatique, université Abdelhamid Ibn Badis Mostaganem, (2021).

[22] Adada L'yasmine, Hamidache Nesrine. *Déanonymisation de clients dans le réseau Bitcoin à l'aide de l'apprentissage automatique*, Matser en informatique, université Mouloud Mammeri Tizi-Ouzou, (2020).

[23] Bouchaour Abdelhamid Nabil, Benyahia Anes Zakarya. *Le vote éléctronique basé sur la Blockchain*, Matser en télécommunication, université Aboubakr Belkaid Tlemcen, (2021).

[24] Laurent Dehouck. *Les risques des Blockchains*, Article n°164, (2017).

[25] Bellanger Michel. *Les Blockchains de la théorie à la pratique, de l'idée à l'implémentation*, Editions ENI, 2e édition, France, (2019).

[26] Naomi Bakary. *Blockchain et secteur bancaire. La Blockchain est-elle une opportunité ou une menace pour l'industrie bancaire*, Master en finance, 2e édition, France, (2019).

[27] Ghoggali Brahim El Khalil. *Systéme des credits bancaire basé sur la technologie Blockchain*, Master en informatique, universite Mouhammed Khider de Biskra, (2020).

[28] Karl Ulrich. *Minage : de la théories à la pratique*, Secrétariat d'état à la formation, à la recharche et à l'innovation(SEFRI), Suisse, (2013).

[29] Kbelhoul Lydia, Lalaoui Feriel. *Anonymat et vie privée dans la Blockchain*, Master recherche en informatique , université Abderrahmane Mira de Bejaia , (2021).

[30] P. Roques. *UML 2 par la pratique : Etudes de cas et exercices corriges*. Eyrolles, 5ème édition, 2006.

[31] Bennanni Sid Ahmed. *Implémentation d'un smart contract sous la plateforme Ethereum : vote électronique*, Master recherche en informatique , université saad dahlab de Blida , (2019).

[32] https ://www.omg.org/. *OMG | Object Management Group.* Consulté le 10/08/2023.

[33] Jean-Marie Defrance. *JQuery-Ajax avec PHP*, Eyrolles, quatrième édition, France, (2013).

[34] Jean Engels. *PHP5*, Eyrolles, troisième édition, France, (2013).

[35] Raphaël Goetter. *CSS avancé vers HTML5 et CSS3* , Eyrolles, deuxième édition, France, (2012).

[36] Samuel Ronce. *Developper des jeux en HTML5 et JavaScript* , Eyrolles, France, (2013).

[37] Ritesh Modi. *Solidity Programming Essentials*, Packt Publishing, première édition, (2018).

[38] https ://code.visualstudio.com/. *Visual Studio Code.* Consulté le 28/08/2023.

Résumé

La Blockchain, un sujet au cœur de l'actualité, transcende divers secteurs. Cette technologie révolutionnaire introduit de nouvelles approches basées sur la cryptographie, permettant ainsi d'éliminer les intermédiaires de confiance traditionnels. Structurée en un réseau peer-to-peer (P2P) de nœuds interconnectés, elle repose sur l'utilisation de chiffrement asymétrique pour garantir la sécurité des données et sur le consensus pour éliminer les risques de fraude et instaurer la confiance au sein du système. Cette méthodologie permet la transmission et le stockage d'informations, lesquelles sont ensuite structurées en blocs liés les uns aux autres. Ces données sont partagées entre plusieurs nœuds du réseau, les rendant immuables et inviolables.

Elle offre plusieurs avantages au secteur public, notamment en matière de stockage sécurisé et transparent des données, facilitant ainsi la transition vers l'éducation numérique. Dans notre projet, nous avons mené une étude approfondie sur le système blockchain et ses concepts fondamentaux. Nous avons abordé un problème spécifique, à savoir la traçabilité des absences des élèves, qui exige transparence et fiabilité. C'est dans ce contexte que nous avons proposé une solution basée sur la blockchain, reposant sur des contrats intelligents. Cette solution se traduit par la création d'une application décentralisée garantissant la sécurité et la gestion des absences, tout en préservant la traçabilité et en facilitant le partage d'informations entre les parties prenantes concernées.

Mots clés : Blockchain, Contrat intelligent, Application Web, Éducation, Gestion des absences.

Abstract

The Blockchain, a topic at the forefront of current events, transcends various sectors. This revolutionary technology introduces new approaches based on cryptography, allowing the elimination of traditional trusted intermediaries. Structured as a peer-to-peer (P2P) network of interconnected nodes, it relies on asymmetric encryption to ensure data security and on consensus to eliminate fraud risks and establish trust within the system. This methodology enables the transmission and storage of information, which is then structured into linked blocks. These data are shared among multiple network nodes, making them immutable and inviolable.

It offers several advantages to the public sector, particularly in terms of secure and transparent data storage, thereby facilitating the transition to digital education. In our project, we conducted an in-depth study of the blockchain system and its fundamental concepts. We addressed a specific problem: student absence tracking, which requires transparency and reliability. In this context, we proposed a blockchain-based solution built on smart contracts. This solution involves creating a decentralized application that ensures the security and management of absences while preserving traceability and facilitating information sharing among relevant stakeholders.

Keywords: Blockchain, Smart contract, Web Application, Education, Absence Management.