

1em

RÉPUBLIQUE TUNISIENNE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DE GABÈS
ÉCOLE NATIONALE D'INGÉNIEURS DE GABÈS

RAPPORT DE PROJET DE FIN D'ÉTUDES

Étude et Mise en Place d'une Solution Basée sur la Blockchain pour la Délivrance des Attestations de Stage

Présenté par : Votre Nom

Soutenu le : [Date de soutenance]
devant la commission de jury :
Mme. [Nom du jury 1], Rapporteur
M. [Nom du jury 2], Examineur

Encadrant académique : [Nom de l'encadrant]
Encadrant professionnel : [Nom de l'encadrant]

Année universitaire : 2024-2025

Dédicaces

À Dieu, le Tout-Puissant, pour m'avoir guidé et donné la force de mener ce projet à bien.

À mes parents, [Noms], dont l'amour inconditionnel et les sacrifices ont été le socle de mes études.

À mes frères et sœurs, [Noms], pour leur soutien et leur inspiration.

À mes encadrants, [Noms], pour leur patience, leurs conseils avisés et leur expertise.

À mes amis et collègues, pour leurs encouragements et leurs discussions enrichissantes.

Ce travail est le fruit de votre présence et de votre soutien.

Remerciements

Nous exprimons notre profonde gratitude à Dieu pour la santé, la persévérance et l'inspiration accordées tout au long de ce projet.

Un remerciement particulier à notre encadrant académique, [Nom], pour son encadrement rigoureux, ses retours constructifs et sa disponibilité sans faille.

Nous remercions également notre encadrant professionnel, [Nom], pour son expertise technique, ses orientations pratiques et son soutien constant.

Nos vifs remerciements vont à [Nom de l'organisme] pour avoir fourni un environnement de travail stimulant et des ressources techniques essentielles.

Nous adressons notre reconnaissance aux membres du jury, [Noms], pour leur temps et leur évaluation approfondie.

Enfin, merci à notre famille, nos amis et tous ceux qui, de près ou de loin, ont contribué à la réussite de ce projet.

Résumé

Ce rapport décrit la conception et les bases théoriques d'une plateforme web basée sur la blockchain pour la délivrance des attestations de stage. Développée dans le cadre d'un stage de quatre mois à [Nom de l'organisme], la solution utilise Polygon pour garantir l'immutabilité et la traçabilité des données, Solidity pour les contrats intelligents, React JS pour une interface utilisateur dynamique, Node.js avec Ethers.js pour le backend, et MySQL pour les données hors chaîne. Les fonctionnalités incluent l'inscription sécurisée via MetaMask, la soumission et validation tripartite des rapports, la génération d'attestations avec QR code, et une vérification publique. Ce document détaille l'étude préliminaire et la conception du système, posant les fondations d'une implémentation robuste et scalable.

Introduction générale

La transformation numérique des processus administratifs dans l'enseignement supérieur est devenue une priorité pour répondre aux exigences d'efficacité, de transparence et de sécurité. Cependant, la gestion des attestations de stage, un document clé pour valider les expériences professionnelles des étudiants, reste confrontée à des défis majeurs. Les méthodes traditionnelles, basées sur des documents papier ou des systèmes centralisés, sont vulnérables à la falsification, entraînent des délais de validation prolongés, et manquent de transparence, ce qui complique la vérification par des tiers tels que les employeurs ou les institutions.

La technologie blockchain, avec ses caractéristiques d'immutabilité, de décentralisation et de traçabilité, offre une solution innovante pour surmonter ces obstacles. En enregistrant les données de manière sécurisée et transparente sur un registre distribué, elle permet d'automatiser les processus, de garantir l'authenticité des documents, et de simplifier leur vérification. Ce projet de fin d'études, réalisé dans le cadre d'un stage de quatre mois à [Nom de l'organisme] en partenariat avec l'École Nationale d'Ingénieurs de Gabès (ENIG), vise à concevoir une plateforme web basée sur la blockchain pour la délivrance des attestations de stage. L'objectif principal est de développer une solution qui assure l'intégrité des données, automatise la validation tripartite (encadrants académique, professionnel, et tiers déblocueur), et permet une vérification rapide et publique via un QR code ou un identifiant unique.

Les objectifs spécifiques du projet incluent :

1. Automatisation des interactions entre les parties prenantes (étudiants, encadrants, tiers) pour réduire les délais administratifs.
2. Développement d'une interface utilisateur intuitive, accessible sur divers appareils, pour une adoption facile.
3. Conformité aux normes de protection des données, notamment le Règlement Général sur la Protection des Données (RGPD) [8], pour garantir la confidentialité des informations.
4. Proposition d'une architecture modulaire et scalable, adaptée à une mise en production à grande échelle.

Ce projet s'inscrit dans une démarche d'innovation technologique, combinant des outils modernes comme la blockchain Polygon, le langage Solidity, le framework React JS, et l'environnement Node.js, tout en intégrant une base de données MySQL pour optimiser la gestion des données hors chaîne. Il répond aux besoins spécifiques des établissements académiques tunisiens, tout en proposant une solution adaptable à d'autres contextes.

Le rapport est structuré comme suit :

1. **Chapitre 1 : Étude préliminaire** : Présente l'organisme d'accueil, analyse la problématique, évalue les solutions existantes, détaille les technologies utilisées, et décrit la contribution proposée.
2. **Chapitre 2 : Analyse et conception** : Identifie les besoins fonctionnels et non fonctionnels, justifie les choix technologiques, et propose une conception détaillée du système à travers des modèles UML.

Une conclusion générale et des perspectives clôturent ce rapport, mettant en lumière les apports du projet et les opportunités d'évolution future.

Table des matières

Dédicaces	2
Remerciements	3
Résumé	4
Introduction générale	5
Table des matières	6
Table des figures	7
Liste des Figures	7
Liste des tableaux	7
Liste des Tableaux	7
1 Étude préliminaire	8
1.1 Introduction	8
1.2 Présentation de l'organisme d'accueil	8
1.3 Problématique	8
1.4 Étude de l'existant	9
1.4.1 Analyse des solutions existantes	9
1.4.2 Limites identifiées	9
1.5 Technologies utilisées	10
1.6 Contribution proposée	11
1.6.1 Description générale de la solution	11
1.6.2 Principe de fonctionnement	11
1.7 Conclusion	11
2 Analyse et conception	13
2.1 Introduction	13
2.2 Analyse des besoins	13
2.2.1 Identification des besoins fonctionnels	13
2.2.2 Identification des besoins non fonctionnels	14
2.3 Étude comparative et choix technologiques	14
2.3.1 Comparaison des solutions blockchain	14
2.3.2 Choix des frameworks	15
2.4 Conception générale du système	15
2.4.1 Architecture globale du système	15
2.4.2 Description des interactions principales	16
2.5 Conception détaillée	16
2.5.1 Diagramme de cas d'utilisation	16
2.5.2 Diagramme de classes	17

2.5.3	Diagrammes de séquences	17
2.5.4	Diagramme de déploiement	17
2.6	Conclusion	18
Conclusion générale et perspectives		19
Références bibliographiques		20
Bibliographie		21
Annexes		22

Table des figures

1.1	Technologies utilisées	10
1.2	Flux de fonctionnement de la plateforme	11
2.1	Comparaison des frameworks frontend	15
2.2	Architecture globale du système	15
2.3	Interactions principales	16
2.4	Diagramme de cas d'utilisation	16
2.5	Diagramme de classes	17
2.6	Diagramme de séquence pour soumission	17
2.7	Diagramme de séquence pour validation	17
2.8	Diagramme de déploiement	18

Liste des tableaux

1.1	Comparaison des solutions existantes	9
2.1	Besoins non fonctionnels	14
2.2	Comparaison des solutions blockchain	14

Chapitre 1

Étude préliminaire

1.1 Introduction

Ce chapitre pose les fondations du projet en explorant le contexte organisationnel et technique, en définissant la problématique, en analysant les solutions existantes, en détaillant les technologies retenues, et en présentant la contribution proposée. Cette étude préliminaire vise à fournir une vision claire des enjeux et des choix stratégiques qui guideront le développement de la plateforme.

1.2 Présentation de l'organisme d'accueil

Le projet a été réalisé au sein de [Nom de l'organisme], une entité basée à Gabès, reconnue pour son expertise dans [domaine d'activité, ex. les technologies numériques, l'éducation]. Fondée en [année, si connue], l'organisation joue un rôle clé dans [description du rôle, ex. la formation professionnelle, le développement logiciel], et dispose d'une infrastructure technique robuste, comprenant des serveurs dédiés, des environnements de développement modernes, et une équipe d'experts en informatique. Ce stage de quatre mois, effectué dans le cadre du PFE à l'ENIG, a permis une immersion dans un environnement professionnel stimulant, favorisant l'apprentissage et l'innovation. La collaboration avec des développeurs expérimentés et des gestionnaires de projets a enrichi la conception et la planification de la solution blockchain proposée.

1.3 Problématique

La délivrance des attestations de stage est un processus administratif essentiel, mais souvent inefficace, confronté à plusieurs obstacles :

- **Falsification des documents** : Les attestations papier ou PDF, même signées numériquement, peuvent être modifiées sans trace, compromettant leur authenticité [1]. Par exemple, des outils d'édition permettent de falsifier des signatures ou des contenus, ce qui pose des risques pour les employeurs vérifiant les credentials.
- **Délais de validation prolongés** : La validation manuelle par les encadrants académiques et professionnels peut prendre des semaines, en raison de la coordination nécessaire et des processus bureaucratiques.
- **Manque de transparence** : Les étudiants et les encadrants n'ont souvent pas de visibilité en temps réel sur l'état d'avancement du processus, ce qui génère des frustrations et des retards.

- **Vérification chronophage** : Les tiers, comme les employeurs, doivent contacter directement l'émetteur pour confirmer l'authenticité, un processus long et sujet à erreurs.

Ces défis ont des conséquences significatives : perte de confiance dans les attestations, inefficacité administrative, et obstacles pour les étudiants dans leurs démarches professionnelles. La problématique centrale peut ainsi être formulée : *Comment concevoir une solution numérique qui garantisse l'authenticité, la traçabilité et l'efficacité du processus de délivrance des attestations de stage, tout en automatisant les validations et simplifiant la vérification publique ?*

1.4 Étude de l'existant

1.4.1 Analyse des solutions existantes

Pour contextualiser le projet, une analyse approfondie des solutions existantes a été menée, couvrant trois grandes catégories :

- **Systèmes traditionnels** : Les attestations sont émises sous forme de documents papier ou PDF, souvent signés manuellement ou avec des signatures électroniques simples. Ces systèmes sont largement utilisés dans les universités tunisiennes, y compris à l'ENIG, en raison de leur simplicité. Cependant, ils sont vulnérables à la falsification et nécessitent une coordination manuelle lourde.
- **Plateformes centralisées** : Des portails numériques, comme [exemple fictif : TunisStage ou plateformes universitaires internes], permettent la soumission et la validation en ligne. Ces systèmes offrent une certaine automatisation, mais restent centralisés, ce qui les rend sensibles aux cyberattaques ou aux modifications non autorisées [2]. De plus, ils ne permettent pas une vérification publique directe.
- **Solutions basées sur la blockchain** : Des initiatives comme Blockcerts [?] utilisent la blockchain pour émettre des certificats numériques immuables, principalement pour les diplômes universitaires. Ces solutions garantissent l'intégrité des données, mais se concentrent sur des cas d'usage différents, sans intégrer la complexité de la validation tripartite nécessaire pour les attestations de stage.

TABLE 1.1 – Comparaison des solutions existantes

Critère	Traditionnel	Centralisé	Blockchain
Immuabilité	Non	Non	Oui
Automatisation	Faible	Moyenne	Élevée
Transparence	Faible	Moyenne	Élevée
Coût	Faible	Moyen	Élevé
Vérification publique	Non	Non	Oui

1.4.2 Limites identifiées

L'analyse révèle plusieurs faiblesses :

- **Manque d'immuabilité** : Les systèmes traditionnels et centralisés ne garantissent pas l'intégrité des données à long terme, car les documents ou bases de données peuvent être altérés.

- **Inflexibilité des solutions blockchain** : Les plateformes comme Blockcerts ne prennent pas en charge des processus complexes comme la validation tripartite, essentielle pour les stages.
- **Ergonomie insuffisante** : Les interfaces des solutions existantes, même numériques, sont souvent peu intuitives, limitant leur adoption par des utilisateurs non techniques, tels que les encadrants académiques.
- **Coûts prohibitifs** : Les solutions blockchain basées sur Ethereum entraînent des frais de transaction élevés, rendant leur déploiement à grande échelle difficile [3].

Ces limites soulignent la nécessité d'une solution hybride combinant la sécurité de la blockchain, l'automatisation des processus, et une interface utilisateur accessible.

1.5 Technologies utilisées

Le choix des technologies a été guidé par des critères de performance, de sécurité, de scalabilité, et de compatibilité avec les standards modernes. Voici une description détaillée des outils retenus :

- **Polygon** : Une blockchain de couche 2 compatible avec Ethereum, offrant des transactions rapides et à faible coût. Le testnet Mumbai est utilisé pour le développement, permettant de tester les contrats intelligents sans frais réels. Polygon réduit les coûts par rapport à Ethereum tout en maintenant une sécurité robuste [3]. Sa compatibilité avec les outils Ethereum (comme MetaMask) facilite l'intégration.
- **Solidity** : Langage de programmation orienté contrat pour développer des contrats intelligents sur Polygon et Ethereum. Sa syntaxe inspirée de JavaScript et sa maturité en font un choix idéal pour coder des logiques complexes, comme la validation tripartite [4].
- **React JS** : Framework JavaScript pour construire des interfaces utilisateur dynamiques et responsive. Intégré avec MetaMask, il permet aux utilisateurs de se connecter à la blockchain via leur portefeuille. React offre une architecture basée sur des composants réutilisables, améliorant la maintenabilité [6].
- **Node.js** : Environnement d'exécution pour le backend, utilisé pour développer des API REST performantes. Avec la bibliothèque Ethers.js, Node.js facilite l'interaction entre le frontend et la blockchain, ainsi que la gestion des requêtes vers MySQL [7].
- **MySQL** : Système de gestion de base de données relationnelle pour stocker les données hors chaîne, comme les informations des utilisateurs et les rapports non sensibles. MySQL est choisi pour sa rapidité, sa fiabilité, et sa facilité d'intégration avec Node.js [?].
- **Hardhat** : Environnement de développement pour compiler, tester, et déployer des contrats intelligents. Il offre des outils de débogage avancés et une intégration avec Polygon [5].
- **Bootstrap** : Bibliothèque CSS pour un design responsive et esthétique, assurant une expérience utilisateur cohérente sur différents appareils.

FIGURE 1.1 – Technologies utilisées

Ces technologies forment un écosystème cohérent, combinant la sécurité de la blockchain, la flexibilité du développement web, et l'efficacité du stockage de données.

1.6 Contribution proposée

1.6.1 Description générale de la solution

La solution proposée est une plateforme web décentralisée conçue pour révolutionner la gestion des attestations de stage. Ses principales caractéristiques incluent :

- **Automatisation** : Soumission, validation, et génération des attestations via des processus numériques, réduisant les interventions manuelles.
- **Immuabilité** : Enregistrement des événements critiques (soumissions, validations, génération) sur la blockchain Polygon, garantissant l'intégrité des données.
- **Vérifiabilité** : Génération d'attestations PDF avec un QR code lié à un hash stocké sur la blockchain, permettant une vérification publique instantanée.
- **Accessibilité** : Dashboards personnalisés pour chaque rôle (étudiant, encadrants, tiers), avec une interface intuitive et responsive.
- **Sécurité** : Authentification sécurisée via MetaMask et conformité RGPD pour protéger les données personnelles.

La plateforme vise à répondre aux besoins des établissements académiques, des entreprises, et des étudiants, tout en offrant une solution adaptable à d'autres types de certificats.

1.6.2 Principe de fonctionnement

Le fonctionnement de la plateforme suit un flux structuré :

1. **Inscription** : Les utilisateurs (étudiants, encadrants, tiers) s'inscrivent via MetaMask, sélectionnant leur rôle et leur organisme. Les adresses de portefeuille sont enregistrées pour l'authentification.
2. **Soumission du rapport** : L'étudiant télécharge son rapport via l'interface web. Le rapport est stocké dans MySQL, et un hash est généré et enregistré sur la blockchain pour référence.
3. **Validation tripartite** : Les encadrants académique et professionnel valident le rapport via des transactions blockchain, signées avec leurs portefeuilles. En cas de retard, un tiers déblocueur intervient pour finaliser la validation.
4. **Génération de l'attestation** : Une fois validé, un PDF est généré, contenant un QR code lié au hash de l'attestation, stocké sur la blockchain.
5. **Vérification publique** : Toute partie peut scanner le QR code ou entrer l'identifiant pour vérifier l'authenticité sur la blockchain, sans accès aux données sensibles.

FIGURE 1.2 – Flux de fonctionnement de la plateforme

Ce processus garantit une gestion fluide, sécurisée, et transparente, tout en réduisant les coûts administratifs et les délais.

1.7 Conclusion

Ce chapitre a établi le cadre du projet en analysant la problématique, en évaluant les solutions existantes, et en proposant une plateforme blockchain innovante. Les technologies choisies

(Polygon, Solidity, React JS, Node.js, MySQL) offrent un équilibre entre sécurité, performance, et accessibilité. Le chapitre suivant approfondit l'analyse des besoins et la conception technique du système.

Chapitre 2

Analyse et conception

2.1 Introduction

Ce chapitre vise à transformer la vision préliminaire en un plan technique détaillé. Il identifie les besoins fonctionnels et non fonctionnels, justifie les choix technologiques à travers une étude comparative, et propose une conception complète du système, incluant l'architecture globale et des modèles UML (cas d'utilisation, classes, séquences, déploiement). Cette étape est cruciale pour garantir une implémentation cohérente et efficace.

2.2 Analyse des besoins

2.2.1 Identification des besoins fonctionnels

Les besoins fonctionnels définissent les interactions principales entre les utilisateurs et la plateforme. Ils ont été identifiés à travers des discussions avec les parties prenantes (étudiants, encadrants, administrateurs) et une analyse des processus existants à l'ENIG. Les fonctionnalités clés sont :

- **Inscription et authentification** : Les utilisateurs s'inscrivent via MetaMask, sélectionnant leur rôle (étudiant, encadrant académique, encadrant professionnel, tiers déblocueur). L'authentification utilise les adresses de portefeuille pour sécuriser l'accès.
- **Soumission des rapports** : Les étudiants soumettent leurs rapports de stage via une interface web sécurisée. Les fichiers sont stockés dans MySQL, et un hash est généré pour la blockchain.
- **Validation tripartite** : Les encadrants académique et professionnel valident les rapports, leurs approbations étant enregistrées comme transactions blockchain. Un tiers déblocueur intervient en cas de retard ou de conflit.
- **Génération des attestations** : Une fois validé, un PDF est généré, incluant un QR code lié au hash de l'attestation, stocké sur Polygon.
- **Vérification publique** : Toute personne peut vérifier l'authenticité d'une attestation en scannant le QR code ou en entrant un identifiant unique, accédant aux données immuables de la blockchain.
- **Dashboards personnalisés** : Chaque rôle dispose d'une interface adaptée (par exemple, l'étudiant voit ses soumissions, l'encadrant ses validations en attente).
- **Notifications** : Alertes en temps réel pour informer les utilisateurs des mises à jour (soumission reçue, validation effectuée).

2.2.2 Identification des besoins non fonctionnels

Les besoins non fonctionnels garantissent la qualité et la viabilité du système :

- **Sécurité** : Immuabilité des données via la blockchain et conformité au RGPD pour protéger les informations personnelles (par exemple, anonymisation des données sensibles) [8].
- **Performance** : Temps de réponse rapide pour les transactions blockchain (moins de 2 secondes sur Polygon) et les requêtes API (moins de 500 ms).
- **Ergonomie** : Interface intuitive, responsive, accessible sur ordinateurs et mobiles, avec une navigation claire.
- **Scalabilité** : Capacité à gérer des milliers d'utilisateurs simultanés, avec une architecture modulaire pour une montée en charge.
- **Maintenabilité** : Code modulaire, bien documenté, avec des tests automatisés pour faciliter les mises à jour.
- **Coût** : Minimisation des frais de transaction grâce à Polygon, rendant la solution viable pour les établissements.

TABLE 2.1 – Besoins non fonctionnels

Critère	Exigence
Sécurité	Immuabilité, RGPD
Performance	<2s (blockchain), <500ms (API)
Ergonomie	Responsive, intuitive
Scalabilité	Milliers d'utilisateurs
Maintenabilité	Code modulaire, tests
Coût	Frais faibles (Polygon)

2.3 Étude comparative et choix technologiques

2.3.1 Comparaison des solutions blockchain

Le choix de la blockchain est critique pour la sécurité et la scalabilité. Une analyse comparative a été menée entre plusieurs options :

TABLE 2.2 – Comparaison des solutions blockchain

Critère	Ethereum	Polygon	Hyperledger	Binance Chain
Coût des transactions	Élevé	Faible	N/A (privé)	Moyen
Scalabilité	Limitée	Élevée	Élevée	Moyenne
Compatibilité Ethereum	Oui	Oui	Non	Partielle
Accessibilité publique	Oui	Oui	Non	Oui
Communauté	Large	Croissante	Limitée	Moyenne
Sécurité	Élevée	Élevée	Élevée	Moyenne

Polygon a été choisi pour :

- **Faibles coûts** : Les frais de transaction sont inférieurs à 0,01 \$ sur Polygon, contre 1-10 \$ sur Ethereum [3].
- **Scalabilité** : Capacité à traiter des milliers de transactions par seconde.
- **Compatibilité** : Support des outils Ethereum (MetaMask, Ethers.js).
- **Communauté active** : Documentation riche et support technique.

2.3.2 Choix des frameworks

Une analyse comparative a également guidé le choix des frameworks pour le frontend, le backend, et la base de données :

- **Frontend** : React JS a été préféré à Angular (trop lourd) et Vue.js (moins adopté) pour sa flexibilité, sa communauté large, et son intégration aisée avec MetaMask [6]. Bootstrap complète React pour un design responsive.
- **Backend** : Node.js avec Express.js est choisi pour sa performance dans les API REST, sa compatibilité avec Ethers.js, et son écosystème riche [7]. Comparé à Django (Python) ou Spring Boot (Java), Node.js offre une meilleure rapidité pour les applications web décentralisées.
- **Base de données** : MySQL est retenu pour sa maturité, sa rapidité dans les requêtes relationnelles, et son intégration avec Node.js [?]. MongoDB (NoSQL) a été écarté, car les données structurées des rapports et utilisateurs conviennent mieux à un modèle relationnel.
- **Développement blockchain** : Hardhat est utilisé pour compiler, tester, et déployer les contrats Solidity, grâce à ses outils de débogage et son intégration avec Polygon [5].

FIGURE 2.1 – Comparaison des frameworks frontend

2.4 Conception générale du système

2.4.1 Architecture globale du système

La plateforme repose sur une architecture modulaire à quatre couches, conçue pour séparer les responsabilités et faciliter la maintenabilité :

FIGURE 2.2 – Architecture globale du système

- **Couche présentation (Frontend)** : Développée avec React JS et Bootstrap, elle offre une interface utilisateur dynamique et responsive. Les utilisateurs interagissent via des composants comme les formulaires de soumission, les dashboards, et les interfaces de vérification.
- **Couche applicative (Backend)** : Implémentée avec Node.js et Express.js, elle gère les API REST pour les interactions entre le frontend, la blockchain, et la base de données. Ethers.js facilite la communication avec les contrats intelligents.

- **Couche blockchain** : Les contrats intelligents, codés en Solidity et déployés sur Polygon, gèrent les validations et les hashes des attestations. Ils garantissent l’immuabilité et la traçabilité des données critiques.
- **Couche données** : MySQL stocke les données non sensibles (profils utilisateurs, rapports) et les métadonnées des transactions blockchain, assurant une récupération rapide.

Cette architecture hybride combine les avantages de la blockchain (sécurité, transparence) et des systèmes centralisés (performance, stockage), optimisant l’expérience utilisateur et les coûts.

2.4.2 Description des interactions principales

Les interactions principales entre les acteurs et le système sont :

- **Inscription** : L’utilisateur se connecte via MetaMask, son adresse de portefeuille est vérifiée, et son rôle est enregistré dans MySQL.
- **Soumission** : L’étudiant télécharge un rapport, qui est haché (SHA-256) et stocké sur la blockchain via un contrat intelligent. Le fichier est sauvegardé dans MySQL.
- **Validation** : Les encadrants signent des transactions blockchain pour approuver le rapport. Le tiers déblocueur agit si nécessaire.
- **Génération** : Le backend génère un PDF avec un QR code pointant vers the hash blockchain.
- **Vérification** : Une interface publique permet de vérifier l’authenticité en comparant le hash du PDF avec celui stocké sur Polygon.

FIGURE 2.3 – Interactions principales

2.5 Conception détaillée

2.5.1 Diagramme de cas d’utilisation

Le diagramme de cas d’utilisation décrit les interactions des acteurs avec le système :

FIGURE 2.4 – Diagramme de cas d’utilisation

Les acteurs sont :

- **Étudiant** : S’inscrit, soumet des rapports, consulte ses attestations.
- **Encadrant académique** : Valide les rapports, consulte les soumissions.
- **Encadrant professionnel** : Valide les rapports, confirme l’expérience.
- **Tiers déblocueur** : Intervient en cas de retard ou conflit.
- **Utilisateur public** : Vérifie les attestations via QR code.

Les cas d’utilisation incluent l’inscription, la soumission, la validation, la génération, et la vérification, chacun lié à des fonctionnalités spécifiques.

FIGURE 2.5 – Diagramme de classes

2.5.2 Diagramme de classes

Le diagramme de classes structure les entités du système et leurs relations :

Les entités principales sont :

- **Utilisateur** : Attributs (adresse portefeuille, rôle, organisme) et méthodes (s’inscrire, se connecter).
- **Rapport** : Attributs (ID, hash, date soumission, statut) et méthodes (soumettre, valider).
- **Attestation** : Attributs (ID, hash, QR code, date émission) et méthodes (générer, vérifier).

Les relations incluent des associations (un utilisateur soumet plusieurs rapports) et des dépendances (une attestation dépend d’un rapport validé).

2.5.3 Diagrammes de séquences

Les diagrammes de séquences détaillent les interactions dynamiques pour les processus clés :

FIGURE 2.6 – Diagramme de séquence pour soumission

FIGURE 2.7 – Diagramme de séquence pour validation

Ces diagrammes illustrent :

- **Soumission** : L’étudiant envoie un rapport, le backend le stocke, et le contrat intelligent enregistre le hash.
- **Validation** : Les encadrants signent des transactions, le contrat met à jour le statut, et le tiers intervient si nécessaire.

2.5.4 Diagramme de déploiement

Le diagramme de déploiement décrit l’infrastructure physique :

Le système est initialement déployé localement :

- **Client** : Navigateur avec MetaMask pour interagir avec le frontend.
- **Serveur web** : Node.js hébergeant les API et le frontend.
- **Base de données** : Serveur MySQL pour les données hors chaîne.
- **Blockchain** : Nœud Polygon (testnet Mumbai) pour les contrats.

Une future mise en production envisagerait un hébergement sur AWS (EC2 pour le backend, RDS pour MySQL) et le mainnet Polygon.

FIGURE 2.8 – Diagramme de déploiement

2.6 Conclusion

Ce chapitre a fourni une analyse approfondie des besoins fonctionnels et non fonctionnels, justifié les choix technologiques à travers des comparaisons rigoureuses, et conçu une architecture modulaire soutenue par des modèles UML. Ces éléments constituent une base solide pour l'implémentation, garantissant que la plateforme répondra aux exigences d'authenticité, d'efficacité, et de scalabilité.

Conclusion générale et perspectives

Ce projet de fin d'études a permis de concevoir les bases théoriques et techniques d'une plateforme blockchain innovante pour la délivrance des attestations de stage. Le premier chapitre a contextualisé la problématique, analysé les solutions existantes, et proposé une solution adaptée aux besoins des établissements académiques. Le second chapitre a détaillé les besoins, justifié les choix technologiques, et fourni une conception complète à travers une architecture modulaire et des modèles UML.

Les principaux apports incluent une automatisation des processus administratifs, une garantie d'authenticité via la blockchain, et une simplification de la vérification publique. La solution proposée combine des technologies modernes (Polygon, Solidity, React JS, Node.js, MySQL) pour offrir une plateforme sécurisée, scalable, et accessible.

Perspectives d'évolution :

- **Déploiement réel** : Migration vers le mainnet Polygon pour une utilisation en production.
- **Fonctionnalités avancées** : Intégration de paiements en cryptomonnaie pour les frais administratifs, ou ajout de notifications push pour une meilleure communication.
- **Extension** : Adaptation de la plateforme pour d'autres certificats, comme les diplômes ou les certificats de formation.
- **Intelligence artificielle** : Analyse des rapports avec des outils d'IA pour détecter les incohérences ou automatiser les pré-validations.
- **Interopérabilité** : Intégration avec d'autres blockchains ou systèmes universitaires pour une adoption plus large.

Ce travail ouvre la voie à une modernisation des processus administratifs dans l'enseignement supérieur, avec un potentiel d'impact significatif en Tunisie et au-delà.

Références bibliographiques

Bibliographie

- [1] Nakamoto, S. (2008). Bitcoin : A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>.
- [2] Buterin, V. (2014). Ethereum White Paper. <https://ethereum.org/en/whitepaper/>.
- [3] Polygon Technology. (2023). Polygon Developer Guide. <https://docs.polygon.technology/>.
- [4] Solidity Documentation. (2023). Solidity. <https://docs.soliditylang.org/>.
- [5] Hardhat Documentation. (2023). Hardhat. <https://hardhat.org/docs/>.
- [6] React JS Documentation. (2023). React. <https://reactjs.org/docs/>.
- [7] Node.js Documentation. (2023). Node.js. <https://nodejs.org/en/docs/>.
- [8] Regulation (EU) 2016/679 (GDPR). (2016). <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.

Annexes

- Annexe A : Diagrammes UML complets (cas d'utilisation, classes, séquences, déploiement).
- Annexe B : Spécifications techniques des contrats intelligents (structure et fonctions principales).
- Annexe C : Maquettes de l'interface utilisateur (captures d'écran des dashboards).