

Ministère de l'enseignement supérieure et de la recherche scientifique

Université Abdelhamid Ibn Badis – Mostaganem –

Faculté des Sciences exactes et de l'informatique

Département de mathématique et informatique

Filière Informatique : Systèmes informatiques



Rapport de Projet de Fin d'Études

Présenté en vue de l'obtention du Diplôme National Master 2 en Informatique

Spécialité : Ingénierie des Systèmes d'Information

Par : BENHAMMOU Taha Houcine

Thème

Systeme de vote en ligne basé sur la blockchain pour une application de crowdfunding

Encadrante académique:	A.MIMI	Maître Assistante
Président de jury:	F. Henni	Docteur
Examineur:	M. Miroud	Maître Assistant

Année universitaire : 2018/2019

Dédicace

Je dédie ce modeste travail à :

Mes chers parents pour tous les sacrifices qu'ils ont consentis pour que je réussisse.

Je le dédie également à :

Mes frères.

Ma famille.

Mes amis.

En un mot à tous les gens qui ont contribué à ma réussite de près ou de loin.

Je remercie également tous mes professeurs de la première année primaire jusqu'au Master 2 universitaire.

Remerciement

Tout d'abord, je tiens à remercier le bon Dieu le tout Puissant de m'avoir donné la force et le courage de mener à bien ce modeste travail, également je remercie infiniment mes parents, qui m'ont encouragé et aidé à arriver à ce stade de ma formation.

Je tiens à remercier tous ceux et celle qui ont contribué à finaliser ce modeste travail.

Mes remerciements vont à madame MIMI Anissa mon encadreur pour m'avoir guidé pour la réalisation de ce projet.

Mes plus grands remerciements vont à toute l'équipage de la Faculté des sciences exactes et de l'informatique FSEI Mostaganem, tous les Enseignants et toutes les Enseignantes sans oublier l'ensemble de personnels administratifs.

Je remercie vivement les étudiants Master 2 Informatique ISI, pour leur aide morale durant toute la période de préparation.

Enfin, je tiens à remercier tous ceux qui m'ont aidé et assisté durant mes études et j'exprime toute ma gratitude à tous ceux qui m'ont formé.

Résumé

Le présent rapport a été élaboré dans le cadre de mon projet de fin d'études à la faculté FSEI Mostaganem.

Ce projet a pour objectif de réaliser un système de vote en ligne basé sur la Blockchain pour une application de crowdfunding. Nous avons utilisé le Framework Ethereum pour mettre en place un réseau Blockchain et pour développer un smart contract qui permet d'exécuter des transactions dans la Blockchain. Puis, nous avons intégré les fonctionnalités développées sur l'environnement "JSE" dans notre site Web développée avec la technologie JEE.

Mots clés : Blockchain, Ethereum, Crowdfunding, Voting, smart contract, l'environnement "JSE", JEE.

Abstract

This report was prepared as part of my graduation project at the FSEI Mostaganem faculty.

This project aims to create an online voting system based on the Blockchain for a crowdfunding application. We used the Ethereum Framework to set up a Blockchain network and develop a smart contract that allows transactions to be executed in the Blockchain. Then, we integrated the functionalities developed on the environment "JSE" in our Web site developed with JEE technology.

Keywords: Blockchain, Ethereum, Crowdfunding, Voting, smart contract, environment "JSE", JEE.

Table des matières

Introduction Générale	1
CHAPITRE I Contexte Générale	
1. Introduction	4
2. L'histoire du crowdfunding	4
3. Les différents types de crowdfunding	5
4. Les acteurs du crowdfunding	6
5. Fonctionnement	6
5.1. La recherche d'une plateforme adéquate	7
5.2. La recherche des fonds	7
5.3. La période post-financement	8
6. Les avantages inattendues du crowdfunding	9
7. Quelles sont les risques du financement participatif ?	9
8. Le vote en ligne	9
8.1 Développement du vote en ligne	10
8.2 Sécurité du vote en ligne	10
9. Le vote en ligne au cœur de deux grandes interrogations	10
10. Chaînes de blocs et vote en ligne	10
11. Conclusion	11
CHAPITRE II Etude préalable sur la blockchain	
1. Introduction	14
2. Concept clés	14
2.1. Bitcoin	14
2.2. Blockchain	14
2.3. Bloc	15
2.4. Peer-to-Peer	16
2.5. Consensus	17

2.6. Smart contract	17
3. Framework blockchain	18
3.1. Hyperledger	18
3.2. Ethereum	18
3.3. Ripple	18
3.4. Bitcoin	18
3.5. Étude comparative	19
3.6. Au cœur d'Ethereum	20
3.7. Les smart contract, un élément central d'Ethereum	20
4. Blockchain sécurisera le crowdfunding?	21
4.1. L'apport de la technologie blockchain	21
4.2. Les bénéfices de la technologie blockchain	21
5. Comparatif des solutions pour le vote en ligne	22
5.1 Technologie Blockchain et vote en ligne	22
5.2 A quoi pourrait servir la Blockchain alors ?	22
5.3 Fonctionnement du vote électronique à l'aide de la blockchain	23
6. Conclusion	24

CHAPITRE III Spécification des besoins et conception

1. Introduction	26
2. Identification des acteurs du système	26
3. Recensement des besoins	26
3.1. Besoins fonctionnels	26
3.2. Besoins non fonctionnels	27
4. Spécification des besoins	28
4.1. Gérer les paramètres de mon compte	29
4.2. Poster son projet	32
4.3. Voter pour le meilleur projet	33

4.4. Gérer tous les comptes par l'administrateur	34
4.5. Consultation des transactions Blockchain	36
5. Diagramme de classe	38
6. Conclusion	39
CHAPITRE IV Implémentation et réalisation	
1. Introduction	41
2. Architecture de la solution	41
2.1. Architecture opérationnel	41
2.2. Architecture applicative	42
3. Déploiement de l'application	43
4. Environnement de développement	44
4.1. Environnement matériel	44
4.2. Environnement logiciel	44
4.3. Environnement de développement Netbeans	44
4.4. Serveur : Apache Tomcat 8.0	45
4.5. Serveur de base de donnée : WampServer	45
4.6. Technologies de programmation	45
5. Principaux développement	45
5.1 Vu générale sur les pages web du site	46
5.2 Configuration du réseau blockchain	54
6. Conclusion	63
Conclusion Générale	64
Bibliographie	65

Table des figures

Figure 1 : Différents types de financement participatif	5
Figure 2 : Principales acteurs du crowdfunding et leurs interactions	6
Figure 3 : Principe du crowdfunding	8
Figure 4 : Processus exécution d'une transaction avec la technologie Blockchain	14
Figure 5 : Structure du registre Blockchain	15
Figure 6 : Model Peer-to-Peer	16
Figure 7 : Exemple d'un smart contract	17
Figure 8 : Comparatif entre les 4 Framework leaders	19
Figure 9 : Système de vote en ligne basé sur Blockchain	23
Figure 10 : Diagramme de cas d'utilisation global	28
Figure 11 : Raffinement de diagramme de cas d'utilisation 'Gérer les paramètres de mon compte' ..	19
Figure 12 : Diagramme de séquence : 'Créer un compte'	30
Figure 13 : Diagramme de séquence 'Annulation d'une transaction'	31
Figure 14 : Raffinement de diagramme de cas d'utilisation 'Poster son projet'	32
Figure 15 : Raffinement de diagramme de cas d'utilisation 'Voter pour le meilleur projet'	33
Figure 16 : Diagramme de séquence : 'Voter pour le meilleur projet'	33
Figure 17 : Raffinement de diagramme de cas d'utilisation 'Gérer tous les comptes'	34
Figure 18 : Diagramme de séquence : 'Ajouter un compte'	35
Figure 19 : Diagramme de séquence : 'Modifier un compte'	36
Figure 20 : Raffinement de cas d'utilisation 'Consultation des transactions Blockchain'	37
Figure 21 : Diagramme de séquence : Consultation des transactions Blockchain	38
Figure 22 : Diagramme de classe	39
Figure 23 : Architecture opérationnelle de notre application.....	41
Figure 24 : Architecture Applicative de notre application.....	42
Figure 25 : Diagramme de déploiement	43
Figure 26 : Logiciels utilisés dans la réalisation de l'application	44
Figure 27 : Page d'accueil su site.....	46

Figure 28 : Page d'accueil su site -suite-	46
Figure 29 : Page d'ouverture d'une nouvelle session ou une session existante.....	47
Figure 30 : Page d'inscription	47
Figure 31 : Formulaire d'inscription pour les porteurs de projet	48
Figure 32 : Formulaire d'inscription pour les investisseurs	48
Figure 33 : Formulaire d'ajout un nouveau projet	49
Figure 34 : Espace réservé pour les investisseurs	49
Figure 35 : Login administrateur	50
Figure 36 : Espace administrateur	50
Figure 37 : Liste des investisseurs.....	51
Figure 38 : Liste des porteurs de projet	51
Figure 39 : Liste des projets	51
Figure 40 : Base de donnée du site web.....	52
Figure 41 : code source de la page d'accueil.....	52
Figure 42 : code source de la session admin	53
Figure 43 : code source de page ajouter projet	53
Figure 44 : Dreamweaver pour améliorer l'interfaces des pages web.....	54
Figure 45 : Installation Node Package Manager	54
Figure 46 : Page d'accueil de Ganache	55
Figure 47 : Extension de Metamask	56
Figure 48 : Sublime Text 3	56
Figure 49 : Notre Smart-Contract	58
Figure 50 : Dossier migrations.....	59
Figure 51 : Dossier test	60
Figure 52 : Fichier index.html	61
Figure 53 : Fichier app.js	61
Figure 54 : Affichage de la partie client pour les investisseurs	62
Figure 55 : Meta-mask pop-up	62

Liste des tableaux

Tableau 4.1.1 : Description textuelle du cas d'utilisation « Créer compte »	30
Tableau 4.1.2 : Description textuelle du cas d'utilisation « Consulter compte »	31
Tableau 4.2 : Description textuelle « Décrire en détail le projet »	32
Tableau 4.3 : Description textuelle du cas d'utilisation « Sélection des contenus »	34
Tableau 4.4.1 : Description textuelle du cas d'utilisation « Ajouter un compte ».....	34
Tableau 4.4.2 : Description textuelle du cas d'utilisation « Modifier un compte »	35
Tableau 4.4.3 : Description textuelle « Consulter comptes »	36
Tableau 4.5 : Description textuelle « Consultation des transactions Blockchain ».....	37

Liste des abréviations

- **HTML** = Hypertext Markup Language
- **HTTP** = Hypertext Transfer Protocol
- **IT** = Technologies de l'information
- **JDBC** = Java DataBase Connectivity
- **JDK** = Java SE Development Kit
- **JEE** = Java 2 Enterprise Edition
- **JSE** = Java Standard Edition
- **JSP** = JavaServer Pages
- **SGBD** = Système de Gestion de Base de Données
- **SQL** = Structured Query Language
- **UML** = Unified Modeling Language
- **XRP** = Ripple Transaction Protocol
- **EDI** = Échange de données informatisé
- **CDDL** = Common Development and Distribution License
- **PHP** = Hypertext Preprocessor
- **NPM** = Node Package Manager
- **XML** = EXtensible Markup Language

Introduction Générale

Le secteur financier possède un poids considérable sur l'économie des pays dans le monde entier. Il peut soutenir les entreprises selon plusieurs axes de développement : création d'entreprises, emploi, etc. Parallèlement, il recommande des mises à jour ciblées du cadre afin qu'il puisse répondre aux développements récents et aux innovations du secteur. Il est ressorti de nombreux modèles de financement que la présence de nouveaux arrivants sur le marché favorise l'innovation dans l'offre de services.

Et parmi les nouveaux modèles de financement innovant qui ont fait le buzz sur l'échelle mondiale : on note le modèle de financement participatif « crowdfunding »

Le crowdfunding est en plein essor : une situation relative à la démocratisation de l'accès à internet, au développement des réseaux sociaux et à la frilosité des banques à accorder des crédits professionnels. Le crowdfunding est un facteur de développement économique, car il permet de créer des projets et des emplois qui n'auraient peut-être pas vu le jour avec un système de financement traditionnel. Il a également un rôle à jouer dans la compétitivité des pays, favorisant l'émergence des projets innovants et à forte valeur ajoutée. La mise en vigueur du décret relatif à la réglementation du crowdfunding a pour objectif de favoriser sa croissance. Conscients de cette opportunité, les acteurs bancaires cherchent à s'implanter sur ce marché, mettant en évidence leur rôle de tiers de confiance.

Donc, Le crowdfunding est une technique de plus en plus utilisée par les créateurs de projet pour financer un projet qu'ils veulent entreprendre. Si certaines personnes utilisent ce moyen pour compléter la somme dont elles ont besoin, d'autres l'utilisent pour récolter la totalité des fonds nécessaires à l'aboutissement du projet.

Mais, Comment obtenir un financement pour son projet ?

Obtenir un financement n'est pas facile, car l'on est en concurrence avec des milliers de personnes. Pour recevoir des fonds via le crowdfunding, il faut présenter un projet attractif et original. Il faut que ce dernier soit également bien développé et bien clair. L'essentiel est de convaincre l'investisseur. Il est primordial d'entamer une étude approfondie du produit que vous allez vendre, du marché que vous voulez toucher ainsi que des techniques que vous allez adopter pour dépasser les concurrents. Une personne réussit à obtenir un financement plus rapidement lorsque : le produit qu'il veut réaliser est innovant, s'il est utile pour le marché, s'il a un aspect communautaire ou protecteur de la nature. Il faut tout de même bien fixer les conditions de prêts ou de dons en avance pour éviter les conflits.

Pour cela, plusieurs plateformes ont vu le jour permettant la gestion de crowdfunding. Dans ce projet il s'agit de permettre à toutes personnes de participer au financement de différents projets axés sur un thème global (thème humanitaire, environnemental, etc.).

Avec cette application, les utilisateurs pourront devenir membres, faire des propositions de projet et faire des dons, Chaque don donne le droit de voter ou non pour chaque projet proposé. Finalement, le projet qui aura le plus de votes sera financé par les dons récoltés.

L'objectif de cette application est d'intégrer la technologie de blockchain afin de tirer profit de ses nombreux avantages en matière de sécurité.

La Blockchain favorise la transparence et la sécurité des transferts d'argent. Cette solution favorise la confiance des donateurs à un moindre coût. Outre ses qualités relatives à la sécurité, cette application est innovante car elle utilise une technologie nouvelle et principalement liée à la finance. Alors notre application propose cette solution qui permettra de contribuer à une juste cause en toute confiance.

C'est dans ce cadre-là que s'inscrit notre projet de fin d'études, Il s'agit de concevoir et de développer un système permettant d'implémenter les principales fonctionnalités du métier de financement participatif en se basant sur la technologie Blockchain.

Le présent rapport s'articule autour de quatre chapitres. Nous présentons dans le premier chapitre intitulé « **Contexte général** » l'essentiel à savoir sur l'investissement participatif et son rapport avec le vote en ligne. Le second chapitre intitulé « **Étude préalable sur la blockchain** » définit les concepts fondamentaux et indispensables de blockchain au services crowdfunding.

En plus de cela, un troisième chapitre intitulé « **Spécification des besoins et conception** » présente l'analyse et la spécification des besoins et détaille les différents cas d'utilisation en illustrant quelques diagrammes UML. La partie conceptuelle fait l'objet du quatrième chapitre intitulé « **Implémentation et réalisation** » Nous présentons dans ce dernier chapitre les différentes étapes de réalisation, les outils utilisés ainsi que les résultats obtenus à travers une sélection d'interfaces de l'application. Enfin, nous faisons une conclusion générale clôturant ce travail.

CHAPITRE I

Contexte générale

Plan

1. Introduction	4
2. L'histoire du crowdfunding	4
3. Les différents types de crowdfunding	5
4. Les acteurs du crowdfunding	6
5. Fonctionnement	6
5.1. La recherche d'une plateforme adéquate	7
5.2. La recherche des fonds	7
5.3. La période post-financement	8
6. Les avantages inattendues du crowdfunding	9
7. Quelles sont les risques du financement participatif ?	9
8. Le vote en ligne	9
8.1 Développement du vote en ligne	10
8.2 Sécurité du vote en ligne	10
9. Le vote en ligne au cœur de deux grandes interrogations	10
10. Chaînes de blocs et vote en ligne	10
11. Conclusion	11

1. Introduction

Faire financer son projet par la foule, ce concept permet à des créateurs de projet de tout horizon de présenter leurs projets à des milliers d'internautes et collecter les fonds nécessaires pour les réaliser. Ce procédé a permis à des dizaines de milliers de projets de voir le jour à travers le monde. L'idée de ce concept est née grâce à un certain nombre d'artistes qui ont rencontré beaucoup de problèmes pour réussir leurs projets. De nombreuses plateformes ont été proposées où il est désormais possible de s'adresser directement au public pour financer ses projets. Nous présentons dans ce chapitre l'essentiel du crowdfunding ainsi son fonctionnement.

2. L'historique du crowdfunding

Pour comprendre son succès actuel, retournons un peu en arrière...

Le crowdfunding trouve sa source dans le concept de crowdsourcing (externalisation ouverte), c'est-à-dire l'usage de la foule pour l'obtention d'idées, de « *feed-back* », et de solutions pour développer les activités de l'entreprise. Le terme de crowdsourcing a été introduit par Jeff Howe en 2006 qui précise son large champ d'application, mettant en avant un des composants fonctionnels de crowdsourcing « le réseau ». Le crowdsourcing a lieu quand une entreprise à but lucratif, externalise des tâches essentielles de fabrication et de vente de produits destinés au grand public (la foule). [1]

Le crowdfunding est souvent considéré comme un « nouveau phénomène ». Toutefois, le crowdfunding a une longue et riche histoire. Il n'est donc pas aussi nouveau que nous pouvons le penser, en tant que concept, il a des racines qui remonte aux années 18ème siècle. Le principe du crowdfunding est appliqué depuis longtemps par les organismes humanitaires, mais la méthode a radicalement changé. [2].

La grande différence du crowdfunding par rapport à ce qui se passait avant, réside dans l'utilisation de plateformes sur internet pour organiser le financement. L'évolution de l'informatique, et des réseaux de communication puissants ainsi que l'apparition de réseaux sociaux au sein d'une grande partie de la population mondiale ont donné une nouvelle dimension au financement participatif. Il est désormais possible de cibler plus facilement un nombre très important de personnes.

Les premières plateformes de crowdfunding sont apparues aux Etats-Unis. Depuis lors, le crowdfunding a connu une croissance incroyable, avec un intérêt toujours grandissant.

La plateforme américaine « *Kickstarter* » est une des premières plateformes de crowdfunding à avoir connu un large succès. Kickstarter a financé plus de 84 000 projets créatifs [3]

3. Les différents types du Crowdfunding

Les différents types de crowdfunding dépendent de ce que l'investisseur reçoit en échange de son investissement.

- **Le don simple** : Le don simple est un acte de générosité, où les gens donnent de l'argent à un projet sans que le demandeur de fonds n'ait une quelconque obligation de donner quelque chose en échange.

- **Le don avec récompense** : Les campagnes de crowdfunding basées sur les récompenses en nature offrent, en échange de contributions, des produits ou des services d'une valeur plus faible. Généralement les récompenses ont une valeur symbolique et sont offertes par les demandeurs de fonds. Parmi ces récompenses on retrouve par exemple le CD ou l'autographe d'un artiste.

- **Le prêt** : Le prêt est un modèle de crowdfunding où les investisseurs ont l'occasion de prêter leur argent à des personnes en quête de financement pendant une période déterminée. Les emprunteurs sont principalement des entrepreneurs ou des start-up qui cherchent à financer différents projets. Le prêt peut avoir lieu avec ou sans intérêt.

- **L'investissement contre prise de participation** : L'investissement contre une prise de participation, également appelé equity crowdfunding, est une forme de financement participatif où en échange d'une contribution, l'investisseur reçoit des parts de l'entreprise, des actions ou des obligations. [4]



Figure 1 : Différents types de financement participatif

4. Les Acteurs du Crowdfunding

Le financement participatif implique l'intervention de trois acteurs principaux, les entrepreneurs d'une part, les investisseurs d'autre part et finalement les plateformes qui vont jouer le rôle d'intermédiaire, comme indiqué dans la Figure 2.

Acteurs et relations du financement participatif

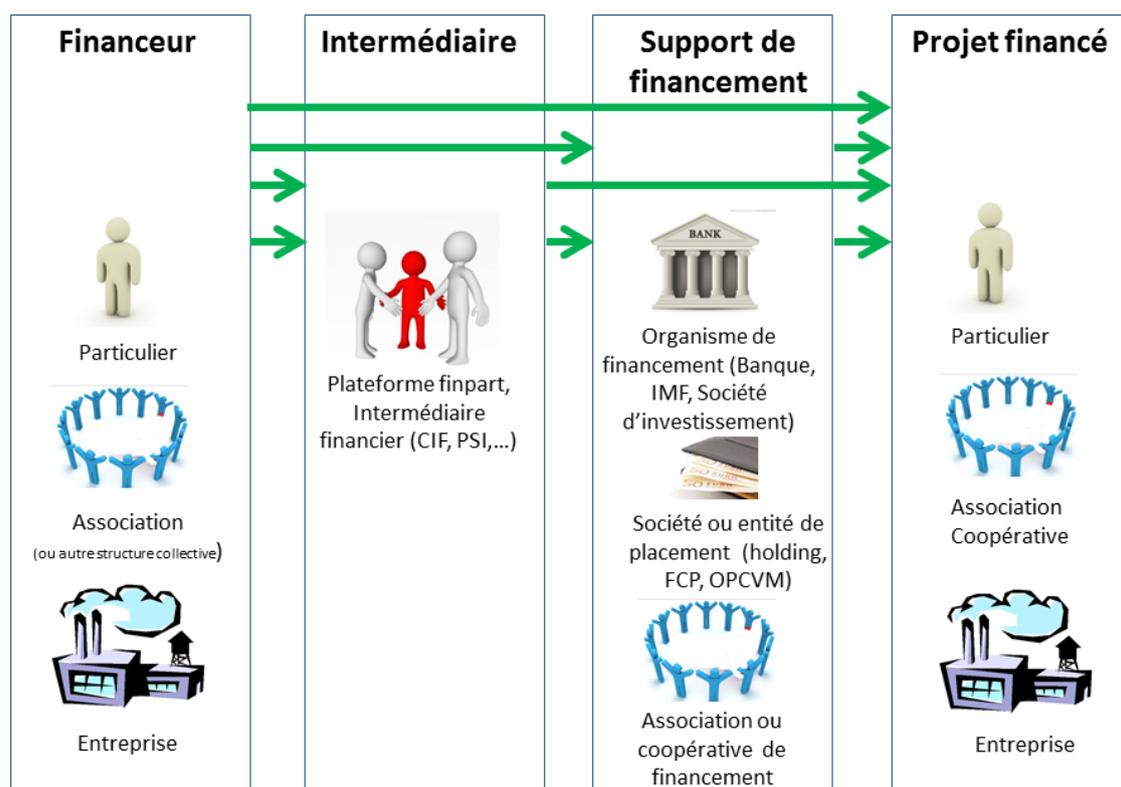


Figure 2 : Principales acteurs du financement participatif et leurs interactions

- Investisseurs : Certaines personnes tiennent à soutenir des projets avec lesquels ils partagent les mêmes valeurs ou des projets locaux qui peuvent contribuer à la création d'emplois. D'autres sont davantage intéressés par les « récompenses » qu'ils vont obtenir en échange de leur contribution.
- Créateur de projet : C'est la personne qui met en pratique une idée qui lui tient à cœur et qu'il souhaite la réaliser et la développer.

5. Fonctionnement

Le crowdfunding repose sur deux principes du web social : la transparence et la participation. La transparence, car les porteurs de projet et les investisseurs sont clairement identifiés. Du côté des investisseurs, ils sont fiers d'avoir contribué à la genèse d'un projet ; en misant sur un projet précis, ils savent exactement où part leur argent.

5.1. La recherche d'une plateforme adéquate

Une fois cette décision prise, il convient de trouver la plateforme qui réponde aux objectifs que l'entrepreneur veut atteindre. Tout dépend de ce que l'entrepreneur est prêt à donner en échange des contributions. Peut-il simplement des donations, ou est-il prêt à emprunter de l'argent en le rémunérant par des intérêts ? il peut même aller plus loin en ouvrant le capital de son entreprise aux investisseurs. Nous reviendrons en détails sur les différents types de *crowdfunding* et de plateformes plus tard dans ce chapitre. Il est également important de s'informer sur la manière dont la plateforme se rémunère. En règle générale elle prélève une commission sur le montant total des fonds alloués à un projet. Le taux de commission peut varier d'une plateforme à l'autre et généralement c'est entre 5% et 10%. D'autres variables sont à prendre en compte, notamment la question de savoir quand l'argent sera-t-il octroyé aux entrepreneurs ? à ce jour, la méthode la plus utilisée est celle du « *all or Nothing* ». Avant toute introduction de projet, les entrepreneurs doivent évaluer le montant nécessaire à la réalisation du projet et le délai qu'ils souhaitent laisser aux investisseurs pour soutenir le projet. Le montant ambitionné sera appelé « l'objectif de financement ». Une fois le délai expiré, les entrepreneurs ne reçoivent l'argent uniquement si leur objectif de financement a été atteint. Dans le cas contraire, les investisseurs se voient tous remboursés de la somme de leur contribution. [5]

5.2. La recherche des fonds

« Ce n'est pas la nature du projet qui détermine celui que va réussir à utiliser correctement l'outil *crowdfunding* ou pas, c'est le créateur lui-même, et la façon dont la collecte va être conduite. » (Vincent Ricordeau, fondateur *KissKissBankBank*)

Dès que la plateforme idéale a été trouvée, l'entrepreneur va essayer de faire connaître son projet au grand public. Pour parvenir à faire cela, il va devoir convaincre trois cercles d'influences, qui sont déterminés sur base de leur distance avec le créateur. Tout d'abord il y a l'entourage proche du créateur, ses amis et sa famille, ensuite il y a le relais d'influence qui contient les amis de ses amis, et finalement le grand public, qui partage la même passion que le créateur.

La mission du créateur est de parvenir à motiver son entourage proche de la validité de son projet. Ensuite, c'est précisément cet entourage qui se mobilisera et créera le « buzz » en attirant le grand public. Un des facteurs de réussite dans la recherche de fonds est celui de la transparence. Il est nécessaire que les entrepreneurs expliquent le mieux possible les risques liés à leur projet.

Ensuite, c'est via la bouche-à-oreille et les réseaux sociaux que l'entrepreneur va tenter de faire connaître son projet. Durant toute la campagne, et même parfois après, il est primordial

que l'entrepreneur informe les investisseurs de manière régulière de l'évolution du projet. Il n'est d'ailleurs pas rare que les entrepreneurs organisent des rendez-vous avec des potentiels investisseurs pour les sensibiliser ainsi que pour partager des idées.

Si l'objectif de financement est atteint avant le délai fixé, le créateur du projet recevra les fonds. Dans le cas contraire, il se peut que l'entrepreneur ne reçoive rien si le projet utilise la méthode du « *all or Nothing* ».

5.3. La période post-financement

C'est durant cette période que l'entrepreneur pourra effectivement créer son projet tout en tenant les investisseurs informés du développement de celui-ci. Ici encore, les attentes diffèrent fortement en fonction du type de financement participatif. Si le projet est basé sur un retour financier pour les investisseurs, l'entrepreneur devra faire part de la santé financière du projet. En revanche, s'il s'agit du financement d'un film, les investisseurs voudront par exemple être tenus informés de la date de sortie du film. [5]

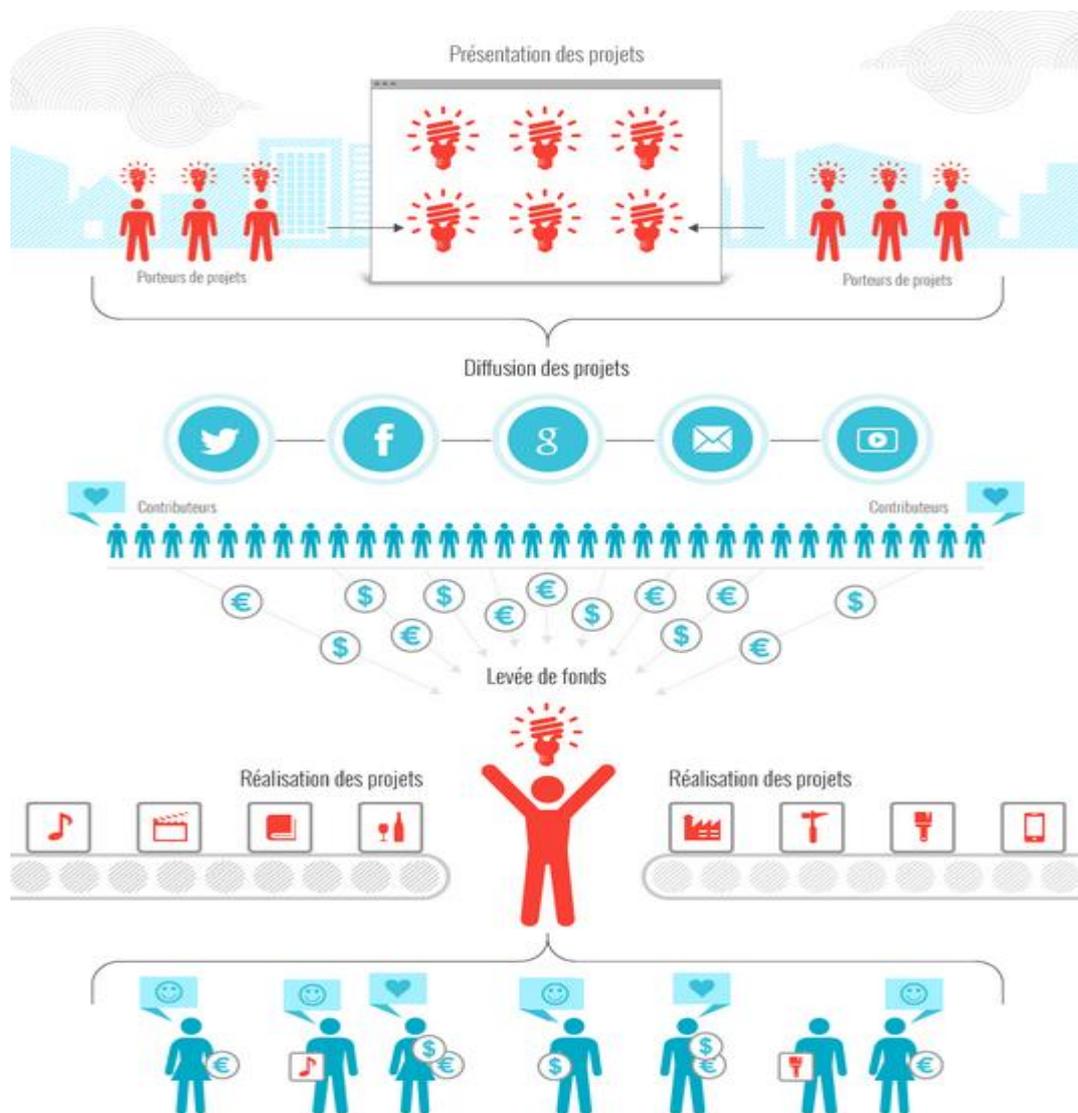


Figure 3 : Principe du crowdfunding

6. Les avantages inattendus du crowdfunding

L'objectif premier du financement participatif est de permettre à des porteurs de projet de réunir simplement et avec l'aide de particuliers la somme nécessaire au lancement de leur entreprise. Avec le développement du crowdfunding, les chefs d'entreprise découvrent des avantages supplémentaires, auxquels ils ne s'attendaient pas. [6]

Dans la grande majorité des cas, les entreprises ayant recours au crowdfunding passent par une plateforme spécialisée dans ce domaine. Ces plateformes, de leur côté, réunissent un public qui peut être d'envergure régionale, nationale, voire internationale en fonction de la catégorie de projets qu'elles acceptent.

- **Une publicité supplémentaire pour l'entreprise** : le projet, grâce à la plateforme, est porté à la connaissance de milliers de personnes. En ayant recours au crowdfunding, l'entreprise bénéficie d'une publicité pouvant impacter réellement son chiffre d'affaires.

- **Le crowdfunding, un test grandeur nature** : le crowdfunding peut être perçu comme un « bac à sable », un endroit où l'on peut faire une étude de marché grandeur nature. Ainsi, un chef d'entreprise présentant son projet est tout d'abord assisté par des experts qui lui diront si son projet est viable ou non. Mais en plus, les investisseurs, en prêtant plus ou moins de l'argent, lui indiqueront si son projet pourra rencontrer, ou non, le succès.

7. Quels sont les risques du financement participatif ?

Comme tout investissement, le financement participatif n'est pas sans risque. Il existe un risque de perte totale ou partielle du capital investi ou des fonds prêtés, notamment en cas de difficulté de l'entreprise émettrice ou emprunteuse, une difficulté en cas de revente des titres car les sociétés ne sont pas cotées.

Aussi, l'absence de garantie quant à l'affectation des fonds collectés au regard du projet initial constitue un risque.

8. Le vote en ligne

Le vote en ligne est la forme de vote électronique la plus numérisée, mais peut recouvrir des modalités technologiques très différentes (et qui déterminent les performances du système de votation) :

- Centralisé vs décentralisé ;
- Protocole : http, mail, ...
- Identification : par toile de confiance, par puce électronique (sur carte ou d'autres supports), ...
- Cryptographique : symétrique vs asymétrique, quantique, post-quantique, ...

8.1 Développement du vote en ligne

Le vote par Internet, évolution la plus récente du vote électronique, représente un changement de paradigme en matière de votations, en rendant possible leur organisation *décentralisée*, que ce soit pour des élections ou des référendums.

8.2 Sécurité du vote en ligne

La sécurité d'un système informatique se définit par la conjonction de :

- La **fiabilité** : capacité d'un système à fonctionner sans erreur et sans tomber en panne, les causes possibles étant de trois types :
 - Défaillances matérielles ;
 - Erreurs de conception ;
 - Erreurs de manipulation.
- La **sûreté** : ensemble des moyens matériels, humains et organisationnels visant à éviter ou contrer toute fraude ou attaque.

9. Le vote en ligne au cœur de deux grandes interrogations

On peut distinguer deux types de problématiques concernant la mise en œuvre du vote en ligne.[7]

La première cherche à évaluer le degré de sécurité susceptible d'être assuré par les nouveaux dispositifs et met en évidence l'ensemble des faiblesses nouvelles d'un système reposant totalement sur le transfert de données par l'internet. Le besoin de sécurité reste cependant relativement tributaire de l'appréhension qu'en ont les usagers, selon leurs habitudes et la confiance qu'ils accordent traditionnellement à leurs responsables publics. [8]

La seconde, plus essentielle, s'interroge sur la nature du « citoyen numérique » et pose la question de l'avenir de l'acte central des démocraties modernes. Dans quelle mesure le développement du vote par l'internet à distance fait peser une menace sur l'un des symboles républicains les plus fondamentaux de la démocratie représentative ? Les débats dans la communauté scientifique sont âpres mais les chercheurs s'accordent à souligner que la mise en place de ces nouvelles modalités de vote comporte en germe des mutations profondes de la notion de citoyenneté.

10. Chaînes de blocs et vote en ligne

Les chaînes de blocs rendent possible un système de votation décentralisé : les votants, après avoir fait leur choix, ont accès à l'immatriculation de leur vote dans la Blockchain (plus de détails sur la blockchain dans le chapitre 2) et peuvent ainsi vérifier qu'il a bien été pris en compte (celui-ci fait alors partie de la base de données, et ne peut être modifié ou supprimé).

II. Conclusion

Nous avons défini dans ce premier chapitre le cadre de notre projet en présentant l'essentiel sur le financement participatif et le vote en ligne. En vue d'avoir une idée plus claire sur le projet, le prochain chapitre sera consacré à une étude préalable afin de comprendre les concepts généraux de blockchain et d'étudier les comparatifs des solutions pour le vote en ligne.

CHAPITRE II

Etude préalable sur la Blockchain

Plan

1. Introduction	14
2. Concept clés	14
2.1. Bitcoin	14
2.2. Blockchain	14
2.3. Bloc	15
2.4. Peer-to-Peer	16
2.5. Consensus	17
2.6. Smart contract	17
3. Framework blockchain	18
3.1. Hyperledger	18
3.2. Ethereum	18
3.3. Ripple	18
3.4. Bitcoin	18
3.5. Étude comparative	19
3.6. Au cœur d'Ethereum	20
3.7. Les smart contract, un élément central d'Ethereum	20
4. Blockchain sécurisera le crowdfunding?	21
4.1. L'apport de la technologie blockchain	21

4.2. Les bénéfices de la technologie blockchain	21
5. Comparatif des solutions pour le vote en ligne	22
5.1 Technologie Blockchain et vote en ligne	22
5.2 A quoi pourrait servir la Blockchain alors ?	22
5.3 Fonctionnement du vote électronique à l'aide de la blockchain	23
6. Conclusion	24

1. Introduction

Afin de réaliser notre projet, il faut passer par une étude des technologies et des concepts clés autour desquels tourne notre solution. Pour se faire, nous présentons dans ce chapitre une étude théorique des concepts du domaine de travail.

2. Concepts clés

Nous nous intéressons, dans cette partie, à définir les notions générales en relation avec notre projet.

2.1. Bitcoin

En 2008, selon le site officiel de bitcoin : une dizaine de personnes connue sous le nom Satoshi Nakamoto ont proposé Bitcoin, un système cryptographique avec un niveau de sécurité très élevé basé sur des preuves mathématiques compliquées. Il permet la gestion des transactions de paiement et le transfert des bitcoins (crypto-monnaie utilisée dans la plateforme Bitcoin) indépendamment de toute autorité centrale et avec des frais très faibles. En effet, ces transactions sont effectuées dans un réseau public de nœuds utilisant la technologie peer-to-peer pour assurer une communication directe entre les nœuds, chacun intervenant dans la validation de la transaction qui, une fois vérifiée, sera enregistrée dans un registre partagée et publique.

2.2. Blockchain

La « Blockchain » est un protocole informatique qui s'apparente à un registre public, sécurisé et partagé entre les participants du réseau [9]. En effet, c'est une infrastructure paire à pair de transactions et de certification cryptographique en temps réel. Elle permet la mise en place d'un réseau composé d'un ensemble d'ordinateurs exécutant des opérations et actualisant le registre partagé en continu. Comme le montre la figure 4, chaque opération envoyée par un utilisateur est ajoutée à un bloc, qui s'ajoute par la suite aux blocs précédents.

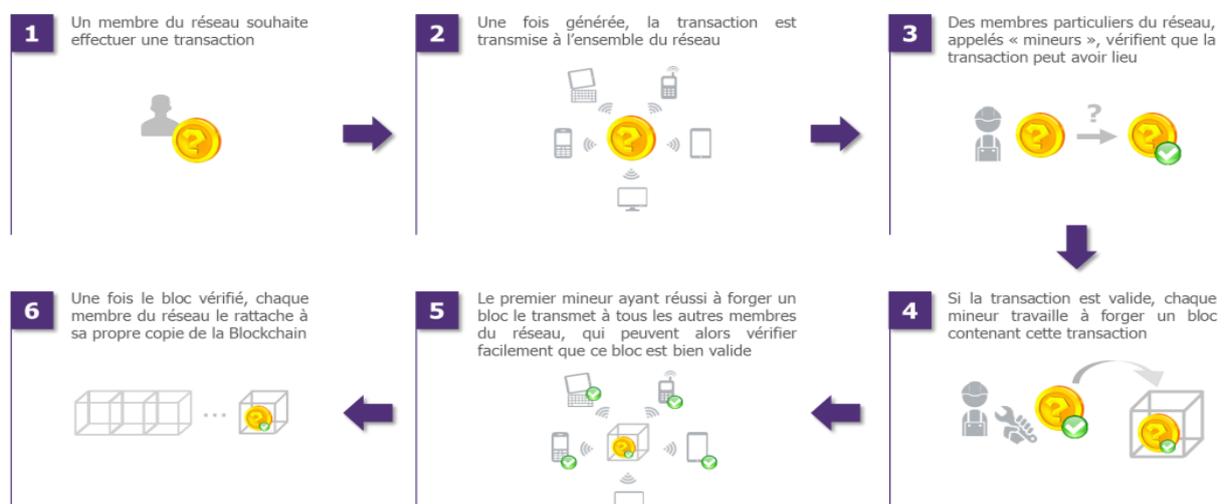


Figure 4 : Processus exécution d'une transaction avec la technologie Blockchain

L'ensemble de ces blocs constitue donc un registre de l'historique des transactions effectuées par les membres du réseau. En effet, une transaction exécutée dans une Blockchain présente toute action d'échange entre deux acteurs. Cet échange peut être un échange de propriété comme des biens, des immobiliers, des titres, etc. Il peut être encore un transfert de droits comme les votes ou les identités. Enfin, il y a un autre type d'échange, qui est l'échange d'engagements, comme la location, le prêt, le partage de matériel, etc. Nous distinguons deux types de Blockchains :

_ Les Blockchains publiques, qui sont des réseaux libres d'accès, dans lesquels les membres peuvent accéder à l'intégralité du registre, effectuer des transactions en toute liberté et participer au processus de leur validation.

_ Les Blockchains privées, dont les informations du registre sont seulement accessibles par des membres qualifiés, de même que la validation des transactions.

2.3. Bloc

La chaîne de blocs se compose de blocs qui contiennent des lots de transactions valides ainsi que les instants de validation de ces transactions-là [10]. Chaque bloc comprend le hash du bloc précédent qui lui relie aux autres blocs. Ces blocs liés forment une chaîne comme le montre la figure 5.

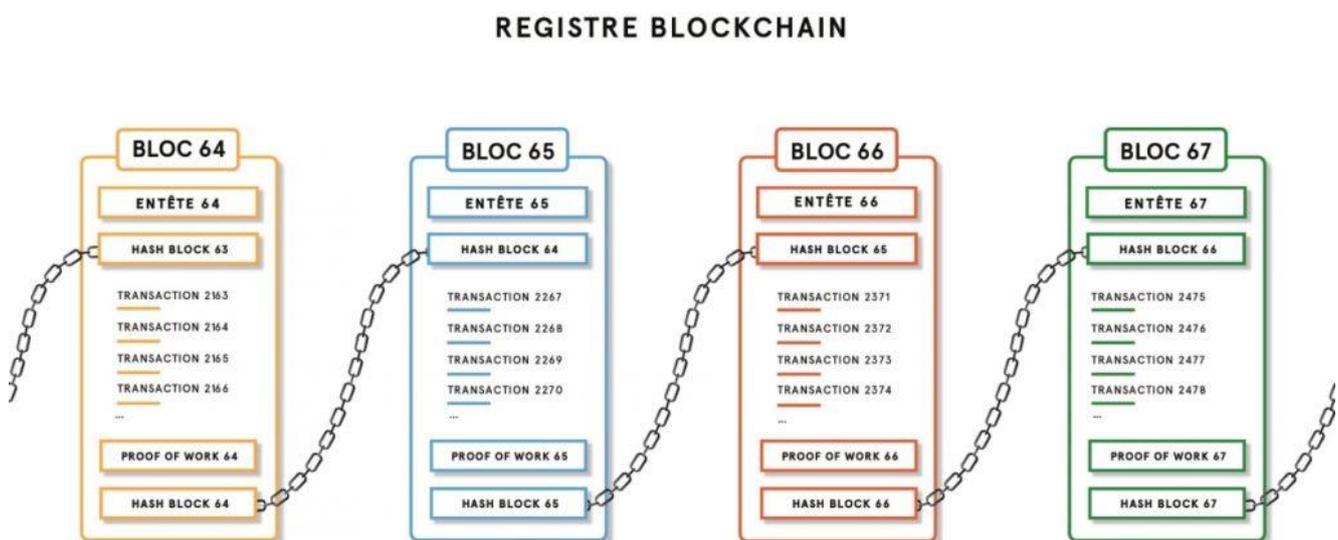


Figure 5 : Structure du registre Blockchain

Le hash (somme de contrôle) est un nombre calculé à partir des informations que contient un bloc par une fonction de hachage unique, ce qui rend impossible sa reconstitution. Dans le cas de Bitcoin, le hash est un nombre de 256 bits. Deux blocs ayant des sommes de contrôle

différentes contiennent nécessairement des contenus différents. Donc, la somme de contrôle d'un bloc peut servir d'identifiant unique au bloc qui lui précède.

La construction d'une Blockchain est composée de deux étapes. La première s'agit de la construction d'un bloc et la deuxième consiste à ajouter ce nouveau bloc à la chaîne de blocs précédents [11].

L'introduction d'un nouveau bloc au système se fait par un nœud du réseau. Le système vérifie que la structure du bloc est correcte et que ce dernier est valide et non contradictoire par rapport aux blocs déjà enregistrés. Si ces tests sont satisfaits, l'ajout du bloc est mis en attente dans une liste locale et diffusé à tous les membres du réseau. Sinon, le bloc sera rejeté.

Dans la deuxième étape, les mineurs du réseau travaillent simultanément et choisissent les transactions à inclure dans leur bloc en construction. Ils ajoutent au bloc un en-tête composé de sa somme de contrôle et l'identifiant du bloc précédent.

2.4. Peer-to-Peer

Un réseau Blockchain est défini par un ensemble de nœuds, appelé chacun Peer [12]. Chaque nœud possède une copie du registre partagé. On distingue essentiellement deux types de nœud dans un réseau Blockchain :

- _ Nœud non mineur : fonctionne en tant que proxy de connexion aux concurrents voisins. Un NVP n'exécute pas les transactions, mais les héberge aux nœuds mineurs,
- _ Nœud mineur : nœud chargé d'exécuter le consensus, de valider les transactions et maintenir le registre partagé.

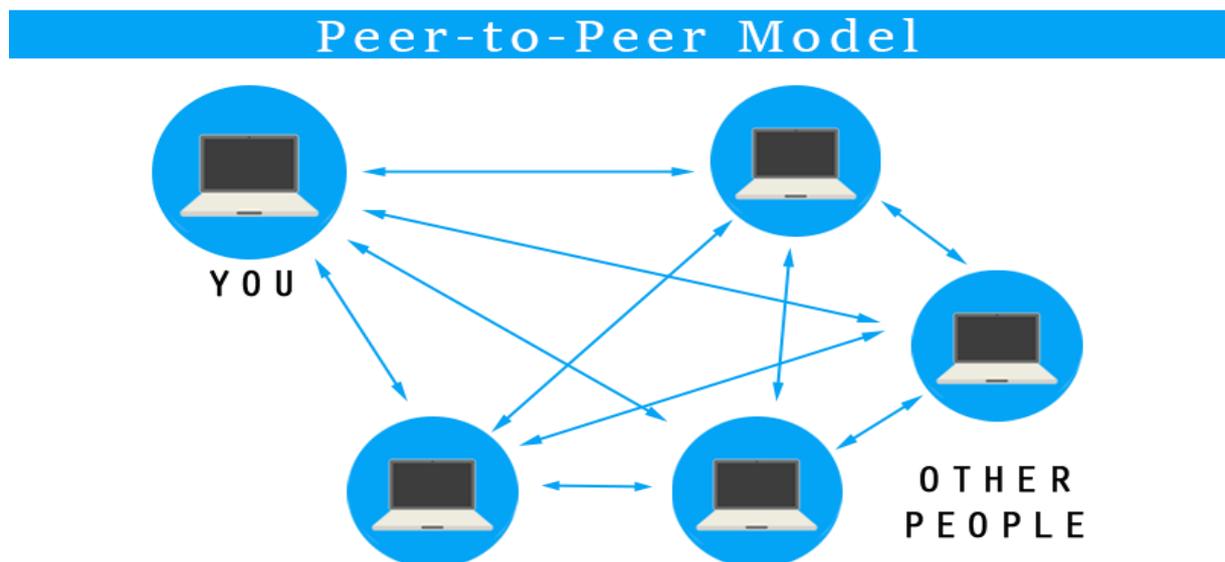


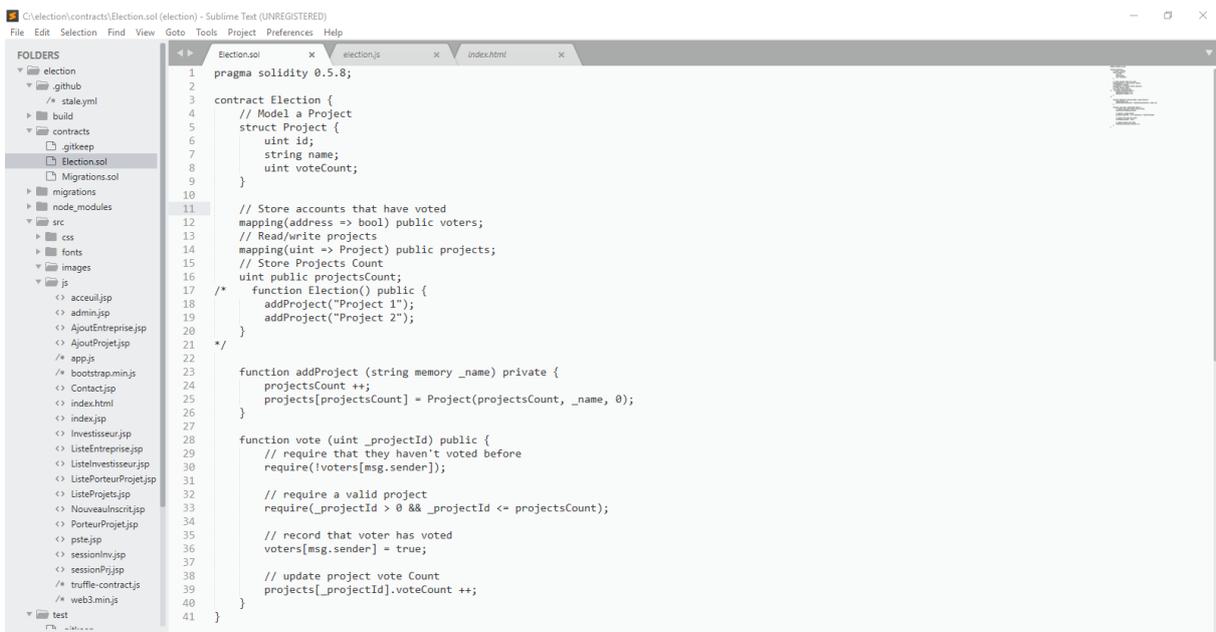
Figure 6: Model Peer-to-Peer

2.5. Consensus

Le consensus consiste dans l'obtention d'un vote entre les membres du réseau, en supposant que les utilisateurs malveillants seront toujours moins nombreux que les utilisateurs honnêtes. A cette fin, les mineurs, des nœuds du réseau ayant un équipement spécifique avec une puissance de calcul énorme, exécutent un problème mathématique afin de valider les transactions. Ce problème permet de déterminer si le membre qui désire exécuter une transaction ne s'agit pas d'un utilisateur mal intentionné. Ceci est réalisé par le fait de vérifier si la série des blocs que possède ce nœud s'agit bien d'une série correcte, c'est à dire qu'elle doit être conforme à la série des blocs que possèdent les membres du réseau. Une fois la majorité des utilisateurs du réseau ont confirmé l'exactitude et la validité de la transaction, le bloc possédant cette transaction sera ajouté à la chaîne des blocs existante.

2.6. Smart contract

Selon Blockchain France, le "smart contract" est un ensemble de programmes autonomes qui exécutent automatiquement des conditions définies au préalable. Ils sont basés sur les instructions conditionnelles de type "if – then" [13]. L'introduction du smart contract se fait soit au moment de la construction de la chaîne, soit après, c'est à dire lorsque la Blockchain est en cours d'exécution, tout dépend de la technologie Blockchain utilisée. Grâce aux procédés cryptographiques utilisés, un smart contract ne pourra pas être modifié et l'ensemble des opérations s'effectuent en toute sécurité, de façon décentralisée et sans aucune autorité centrale de contrôle.



```
1 pragma solidity 0.5.8;
2
3 contract Election {
4     // Model a Project
5     struct Project {
6         uint id;
7         string name;
8         uint voteCount;
9     }
10
11     // Store accounts that have voted
12     mapping(address => bool) public voters;
13     // Read/write projects
14     mapping(uint => Project) public projects;
15     // Store Projects Count
16     uint public projectsCount;
17     /*
18     function Election() public {
19         addProject("Project 1");
20         addProject("Project 2");
21     }
22     */
23     function addProject (string memory _name) private {
24         projectsCount ++;
25         projects[projectsCount] = Project(projectsCount, _name, 0);
26     }
27
28     function vote (uint _projectId) public {
29         // require that they haven't voted before
30         require(!voters[msg.sender]);
31
32         // require a valid project
33         require(_projectId > 0 && _projectId <= projectsCount);
34
35         // record that voter has voted
36         voters[msg.sender] = true;
37
38         // update project vote Count
39         projects[_projectId].voteCount ++;
40     }
41 }
```

Figure 7 : Exemple d'un Smart Contract

3. Framework Blockchain

Techniquement parlant, la Blockchain n'est pas un logiciel, mais un Framework open source de développement informatique, avec un code disponible sur internet et développé par des communautés.

Donc tout d'abord, nous devons choisir le Framework Blockchain que nous utiliserons dans le développement de notre système. Actuellement, il y a une multitude de Framework supportant la technologie Blockchain dont nous exposons dans le paragraphe suivant les leaders de cette technologie.

3.1. Hyperledger

Hyperledger est un framework open source. Il s'agit d'un effort collaboratif créé pour supporter la technologie Blockchain en la démocratisant au-delà du secteur de la finance. [14]. Hyperledger a été lancé fin 2015 par la communauté Linux afin de créer des Blockchains privées ou publiques.

3.2. Ethereum

Ethereum est une plateforme open source qui permet de faire des paiements sécurisés et transparents en se basant sur la technologie Blockchain [15]. Elle est basée sur les contrats intelligents permettant la vérification et la validation des transactions effectuées par ses différents utilisateurs.

Elle utilise aussi l'unité ETHER afin de payer les nœuds exécutant les contrats intelligents. Le seul inconvénient de cette technologie, c'est qu'elle ne permet que le transfert de solde avec la technologie Blockchain qui est un domaine restreint par rapport à tous ceux qu'offre cette technologie.

3.3. Ripple

C'est un Framework qui offre des services au profit des institutions financières. [16]. Il permet la gestion des transferts de fond en utilisant sa crypto-monnaie baptisée XRP classée la troisième en termes de valorisation derrière le Bitcoin et Ethereum. Cependant, cette plateforme présente des limites, c'est qu'il y a une possibilité que le XRP change de valeur pendant les 20 secondes que prend une transaction pour être exécutée. Ce qui constitue un risque de conversion entre XRP et les autres devises.

3.4. Bitcoin

C'est la première monnaie électronique créée pour décentraliser les paiements en utilisant un réseau de personne à personne, sans passer par une institution financière [17]. Bitcoin est aussi la première application développée sur une Blockchain en 2009 qui n'a cessé de grandir

et qui assure des paiements rapides et sécurisées. Bitcoin met à la disposition des développeurs un framework open-source à utiliser dans le développement des solutions au profit du secteur bancaire.

3.5. Étude comparative

La figure 9 présente un comparatif entre les 4 Framework les plus populaires et les plus utilisés dans le développement des applications basées sur la technologie Blockchain selon les critères suivants : l'usage du framework, la communauté le supportant, la monnaie, l'état, le consensus sur lequel est basé le framework, la nature du réseau utilisé, le degré de confidentialité et enfin les caractéristiques des contrats intelligents s'ils sont supporté par le framework.

	Hyperedger fabric	Ethereum	Ripple	Bitcoin
Description	<i>Usage général</i>	<i>Usage général</i>	<i>Paiement</i>	<i>Paiement</i>
Communauté	<i>Fondation linux</i>	<i>Les développeurs d'ethereum</i>	<i>Les developpeurs de ripple</i>	<i>Les developpeurs de bitcoin</i>
Monnaie	<i>NON</i>	<i>Ether</i>	<i>XRP</i>	<i>BTC </i>
Récompense des mineurs	<i>NON</i>	<i>OUI</i>	<i>NON</i>	<i>OUI</i>
État (state)	<i>Base de donnée clé-valeur</i>	<i>Données des comptes</i>	<i>NON</i>	<i>Données des transactions</i>
Consensus	<i>Au choix : PBFT</i>	<i>Minage</i>	<i>Le protocole de ripple</i>	<i>Minage</i>
Réseau	<i>Privé</i>	<i>Public ou privé</i>	<i>Public</i>	<i>Public</i>
Confidentialité	<i>Ouvert à confidentiel</i>	<i>Ouvert</i>	<i>Ouvert</i>	<i>Ouvert</i>
Contrats intelligents	<i>Java, Go</i>	<i>Solidity</i>	<i>Pas de contrats intelligents</i>	<i>Possible, mais non utilisables</i>

Figure 8 : Tableau comparatif entre les 4 Framework leaders sur le marché de la Blockchain

Dans le cadre de notre projet, le choix d'un Framework se fait selon des besoins spécifiques pour le développement de notre système, ce Framework doit être open source, doit avoir une implémentation en Java, doit supporter les contrats intelligents et doit avoir un consensus au choix. Selon l'étude comparative réalisée, le Framework qui répond le plus à nos exigences est le projet Ethereum.

Comme on l'avait défini au pare avant Ethereum est considérée comme la blockchain la plus prometteuse en dehors de Bitcoin. Ses créateurs en parlent comme du « premier véritable ordinateur global », qui permet de construire sur sa plateforme des **applications décentralisées**.

3.6. Au cœur d'Ethereum : les applications décentralisées « Dapps »

De façon schématique, le code source Ethereum peut être comparé au système d'exploitation d'un smartphone : à partir de ce code, diverses applications, dites Dapps (applications décentralisées), peuvent être développées. Ainsi :

- **UjoMusic** veut permettre aux artistes de placer leurs titres sur une blockchain, de définir quelle part de chaque morceau doit revenir à tel ou tel contributeur du titre, et de récolter ensuite les droits d'auteur automatiquement et sans intermédiaire, grâce à des smart contracts.
- **Augur** propose une plateforme décentralisée de marché prédictif basée sur la blockchain, utilisant là encore des smart contracts.
- **Transactive Grid** veut permettre à chacun de vendre et d'acheter des crédits d'énergie à son voisin, de façon pair-à-pair.
- **Filecoin** est un service de stockage cloud décentralisé, qui permet de louer à d'autres utilisateurs l'espace libre de son ordinateur, ou inversement d'acheter de l'espace de stockage sur le réseau Filecoin, de façon décentralisée
- **Golem** permettra à chacun de louer la puissance de calcul inutilisée sur ses appareils (ordinateur, smartphone...) pour tout type d'applications (analyses Big Data en recherche médicale, etc.).
- Etc. [18]

Au-delà des applications décentralisées, Ethereum constitue le socle d'une nouvelle méthode de levée de fonds, fonctionnant avec des cryptomonnaies : les Initial Coin Offering (ICO).

3.7. Les smart contract, un élément central d'Ethereum

Les smart contracts sont des programmes autonomes qui, une fois démarrés, exécutent automatiquement des conditions définies au préalable. Ils fonctionnent comme toute instruction conditionnelle de type « if – then » (si – alors). Les bénéfices : vitesse accrue, meilleure efficacité, et confiance dans le fait que le contrat sera exécuté comme convenu. Ces programmes sont capables de surmonter les problèmes d'aléa moral, et de réduire les coûts de vérification, d'exécution, d'arbitrage et de fraude. Ils ont de multiples applications, notamment en assurance.

L'avantage d'implémenter des smart contracts dans une blockchain est la garantie que les termes du contrat ne pourront pas être modifiés. Un smart contract qui ne serait pas dans la blockchain serait un programme dont les termes pourraient être changés en cours d'exécution.

Les smart contracts posent des défis juridiques et éthiques majeurs, qu'il s'agisse de responsabilité légale ou de protection des consommateurs, et pourraient constituer un enjeu clé pour les Etats : il est même possible d'imaginer une collecte d'impôts via des smart contracts, ainsi qu'une automatisation des prestations de protection sociale.

4. Blockchain sécurisera le crowdfunding ?

Dans l'exercice de son activité, une plateforme de financement participatif fait appel à un prestataire de services de paiement agréé pour sécuriser les transactions de paiement en ligne et à un prestataire de services de certification électronique pour attester de la bonne souscription au capital social d'une entreprise. Toutes ces prestations sont aujourd'hui mieux sécurisées grâce à la technologie de la blockchain.

4.1. L'apport de la technologie blockchain

Une communauté dispersée dans le monde enregistre et atteste de l'existence d'un contrat ou d'une transaction entre 2 parties grâce à la blockchain. La technologie de la blockchain facilite au régulateur l'authentification des transactions grâce à leur enregistrement sur un registre archivé et partagé ce qui le rend infalsifiable.

Le « smart contract » garantit la réalisation de transactions traçables et irréversibles par l'émission d'un titre de propriété dématérialisé avec une sécurité supérieure au droit contractuel traditionnel, mais également l'auto-exécution de plusieurs types de clauses contractuelles attachées à cette transaction, comme la distribution d'intérêts ou de revenus. La rédaction de « smart contract », grâce à la technologie de la blockchain, permettrait de digitaliser les bons de souscription pour en faire des actifs digitaux sécurisés. L'usage de la technologie blockchain a abouti à la naissance des crypto-monnaies à la réputation sulfureuse. Mais en réalité, la crypto-monnaie rend très compliquée la fraude, puisque tout est informatisé, archivé, et traçable. A l'instar des autres devises, la contribution en crypto-monnaie ne présentera aucun risque pour une plateforme de crowdfunding parce que le contributeur reste soumis aux mêmes exigences en terme de compliance.

4.2. Les bénéfices de la technologie blockchain

La blockchain favorise la sécurité des investisseurs / prêteurs grâce à la tenue, à la conservation et aux échanges de titres dans des délais quasi immédiats. Grâce aux « smart contract » les actifs digitalisés par l'intermédiaire de la blockchain deviennent facilement cessibles et participent au développement d'un marché secondaire pour les titres financiers crowdfundés (action, obligation et mini-bon) en toute transparence, en les « tokenisant ». Ils sont alors échangeables comme un jeton numérique dans la blockchain. Les crowdfunders du monde entier contribuent directement de façon sécurisée par l'intermédiaire de la blockchain,

notamment en utilisant une crypto-monnaie. La blockchain constitue aussi une solide clause d'extinction pour les plateformes de crowdfunding. En cas de disparition de la plateforme de crowdfunding, le « smart contract » continue de régir les relations entre investisseurs et porteurs de projets selon les modalités initiales négociées. La technologie de la blockchain accroît la confiance des investisseurs envers les plateformes de financement participatif. Le rôle des plateformes de crowdfunding reste inchangé :

- Tiers de confiance, en sélectionnant les projets à financer sur la qualité du « livre blanc » (description du projet) et de ses développeurs.
- Compliance des crypto-contributeurs selon la législation en vigueur.
- Sécurité et transparence des transactions sur la plateforme.

Le financement participatif est l'une des branches d'activité, à laquelle la technologie blockchain apportera plus de confiance à tous les intervenants en sécurisant et en authentifiant les transactions. [19]

5. Comparatif des solutions pour le vote en ligne

5.1 Technologie Blockchain et vote en ligne

Une autre application de la technologie blockchain est le vote. En exprimant des votes sous forme de transactions, nous pouvons créer une blockchain qui enregistre les résultats des votes. De cette façon, tout le monde peut s'accorder sur le décompte final, car il peut compter les votes lui-même et, en raison de la piste d'audit, il peut vérifier qu'aucun vote n'a été modifié ou supprimé et qu'aucun vote illégitime n'a été ajouté.

5.2 À quoi pourrait servir la blockchain alors ?

La révolution qui sera apportée dans ce cas serait d'adosser le système du vote traditionnel à la blockchain. L'objectif ici, est de rendre le scrutin totalement indépendant, libre et transparent. Ceci est possible puisque la blockchain est un système décentralisé, c'est-à-dire qu'il n'est géré par aucun organisme, et dans ce cas par aucune commission électorale. Chaque vote sera vérifié, validé et enregistré par l'ensemble des participants et même par de tierces personnes depuis leurs canapés. [20]

Un autre point important, c'est la sécurité. La solidité de la blockchain face aux attaques informatiques n'est plus à démontrer car elle est un registre distribué. Si vous voulez le corrompre, alors il va falloir le faire sur toutes les machines. Ce qui reste très difficile. C'est comme si tout le monde disposait d'une copie authentique de chacune des urnes dans sa maison. Pour truquer un seul vote, il faudra le truquer sur l'ensemble des copies distribuées.

L'architecture Blockchain offre d'excellentes solutions à nombre de nos problèmes clés:

- Volume élevé: le vote étant décentralisé, les votes peuvent être placés sur des nœuds du réseau, sans nécessiter un serveur central unique prenant toute la charge.

- Merkle trees: permet de s'assurer que seule la plus longue chaîne de votes vérifiés est acceptée, ce qui complique la tâche de voter sans autorisation ou de modifier un vote déjà enregistré.

- Redondance: Il existe plusieurs répliques exactes des résultats du vote, créées au moment du vote. Il serait nécessaire de simuler un vote sur tous les serveurs redondants.

L'implémentation originale de la chaîne de chaînes dans Bitcoin de Satoshi Nakamoto utilisait une technique appelée Preuve de travail pour ajouter de nouveaux blocs à la chaîne. Cette méthode nécessite que tous les mineurs du réseau résolvent un problème mathématique difficile pour obtenir une récompense, tout en ajoutant toutes les transactions précédentes à la chaîne.

5.3 Fonctionnement du vote électronique à l'aide de la blockchain

La première transaction ajoutée au bloc sera une transaction spéciale représentant le projet. Lorsque cette transaction est créée, elle inclut le nom du projet et sert de bloc de fondation, chaque vote de ce projet étant placé au-dessus de-là.

- Chaque fois qu'un investisseur votera, les transactions seront enregistrées et la blockchain sera mise à jour. Pour garantir la sécurité du système, le bloc contiendra les informations de l'électeur précédent. Si l'un des blocs était compromis, il serait facile de le savoir puisque tous les blocs sont connectés les uns aux autres. La blockchain est décentralisée et ne peut pas être corrompue, il n'existe pas de point de défaillance unique. La blockchain est l'endroit où le vote a lieu. Le vote de l'investisseur est envoyé à l'un des nœuds du système, lequel ajoute ensuite le vote à la blockchain. Le système de vote aura un nœud dans chaque district pour assurer la décentralisation du système.

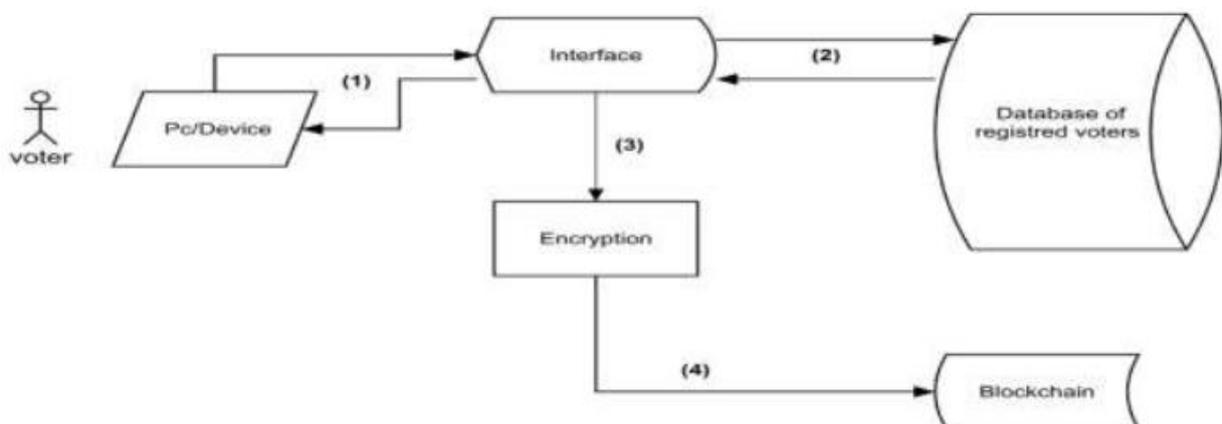


Figure 9 : Système de vote en ligne basé sur la blockchain

6. Conclusion

Dans ce chapitre, nous avons présenté une analyse et une critique sur la blockchain et son impact avec le crowdfunding et ses solutions proposé pour le vote en ligne ainsi que les notions générales en relation avec notre projet.

CHAPITRE III

Spécification des besoins et conception

Plan

1. Introduction	26
2. Identification des acteurs du système	26
3. Recensement des besoins	26
3.1. Besoins fonctionnels	26
3.2. Besoins non fonctionnels	27
4. Spécification des besoins	28
4.1. Gérer les paramètres de mon compte	29
4.2. Poster son projet	32
4.3. Voter pour le meilleur projet	33
4.4. Gérer tous les comptes par l'administrateur	34
4.5. Consultation des transactions Blockchain	36
5. Diagramme de classe	38
6. Conclusion	39

1. Introduction

Dans ce chapitre, nous précisons les acteurs du système à développer et l'ensemble des fonctionnalités qu'il doit satisfaire ainsi que les différentes contraintes auxquelles il doit se soumettre. Ensuite, nous utilisons le langage de modélisation UML et StarUml comme logiciel de modélisation [21] afin de présenter le diagramme de cas d'utilisation globale et de décrire les fonctionnalités majeures à l'aide des diagrammes de séquence système.

2. Identification des acteurs du système

Les Internautes peuvent avoir le statut suivant en fonction de leur activité sur l'application :

- Visiteur : désigne toute personnes physiques ou morales qui se rend sur l'application pour consulter le contenu et les informations publiés sur elle sans y être inscrite ou préalablement à son inscription sur l'application en tant qu'Utilisateur.
- L'administrateur du système : il fait le contrôle du système et la gestion des comptes utilisateurs. Il fournit les conditions et les données nécessaires permettant le bon déroulement des fonctionnalités du système.
- Utilisateur : désigne les internautes personnes physiques ou morales des Porteurs de Projets inscrits sur l'application sous condition d'avoir préalablement adhéré aux Conditions Générales.
- Investisseur : désigne les Utilisateurs ayant investi ou qui sont en cours d'investissement dans un projet ou plusieurs projets proposés via l'application.

3. Recensement des besoins

Dans cette section, nous commençons par la présentation des besoins fonctionnels et non fonctionnels auxquels répond notre système.

3.1. Besoins fonctionnels

Les besoins fonctionnels doivent répondre aux exigences du système en termes de fonctionnalités. Ils constituent une sorte de contrat par rapport au comportement du système cible.

L'application doit mettre à la disposition de l'administrateur les fonctionnalités suivantes :

- Gestion des comptes (Profil : Investisseur/Porteur de projet) :
 - Ajouter un compte,
 - Modifier un compte,
 - Consulter les comptes,
 - Activer et désactiver un compte,

- Effectuer une recherche sur les comptes.
- Gestion des projets :
 - Acquisition du projet,
 - Gérer les projets acquis,
 - Accepter projet,
 - Publication des projets,
- Gestion des entreprises
- Affichage des mouvements effectués par les utilisateurs ainsi que les investisseurs,
- Affichage des transactions sauvegardées dans la Blockchain.

Elle offre au utilisateurs les fonctionnalités suivantes :

- Création d'un compte,
- Consultation du compte,
- Consultation de ses mouvements,
- Annulation d'une transaction,
- Gestion des paramètres de son profil,
- Affichage de ses notifications.

3.2. Besoins non fonctionnels

Les besoins non fonctionnels sont liés à la qualité d'un système. Ils définissent les exigences auxquelles doit répondre ce dernier et le comportement souhaité. Notre système doit offrir à ses utilisateurs les fonctionnalités suivantes :

- Sécurité : l'authentification est nécessaire pour que l'utilisateur puisse utiliser l'application. Il faut aussi tenir compte des droits d'accès et permissions pour protéger les données et surtout lorsqu'il s'agit de l'exécution des transactions financières qui doivent être faites dans un contexte très sécurisé d'où le besoin de protéger le système contre les attaques,
- Ergonomie : Les différentes fonctionnalités du système doivent être compréhensibles et facile à manipuler par l'utilisateur. L'application doit offrir une meilleure expérience utilisateur,
- Modularité et extensibilité : le système doit tenir en compte la possibilité de son extension par l'ajout de nouvelles fonctionnalités,

- **Maintenabilité** : diminuer l'effort nécessaire pour localiser et corriger un défaut dans l'application.

Il faut donc structurer et organiser convenablement le code source.

4. Spécification des besoins

Suite à la définition de nos acteurs, nous décrivons leurs interactions avec notre système à travers un diagramme de cas d'utilisation global qui reflète les fonctionnalités générales. En effet, le diagramme de cas d'utilisation est une représentation intuitive du système et des différents acteurs qui interagissent avec lui sans pourtant se soucier de la manière dont le système va réellement exécuter les fonctionnalités décrites. La figure 10 illustre le diagramme du cas d'utilisation général du système que nous présentons avec plus de détails dans le paragraphe suivant.

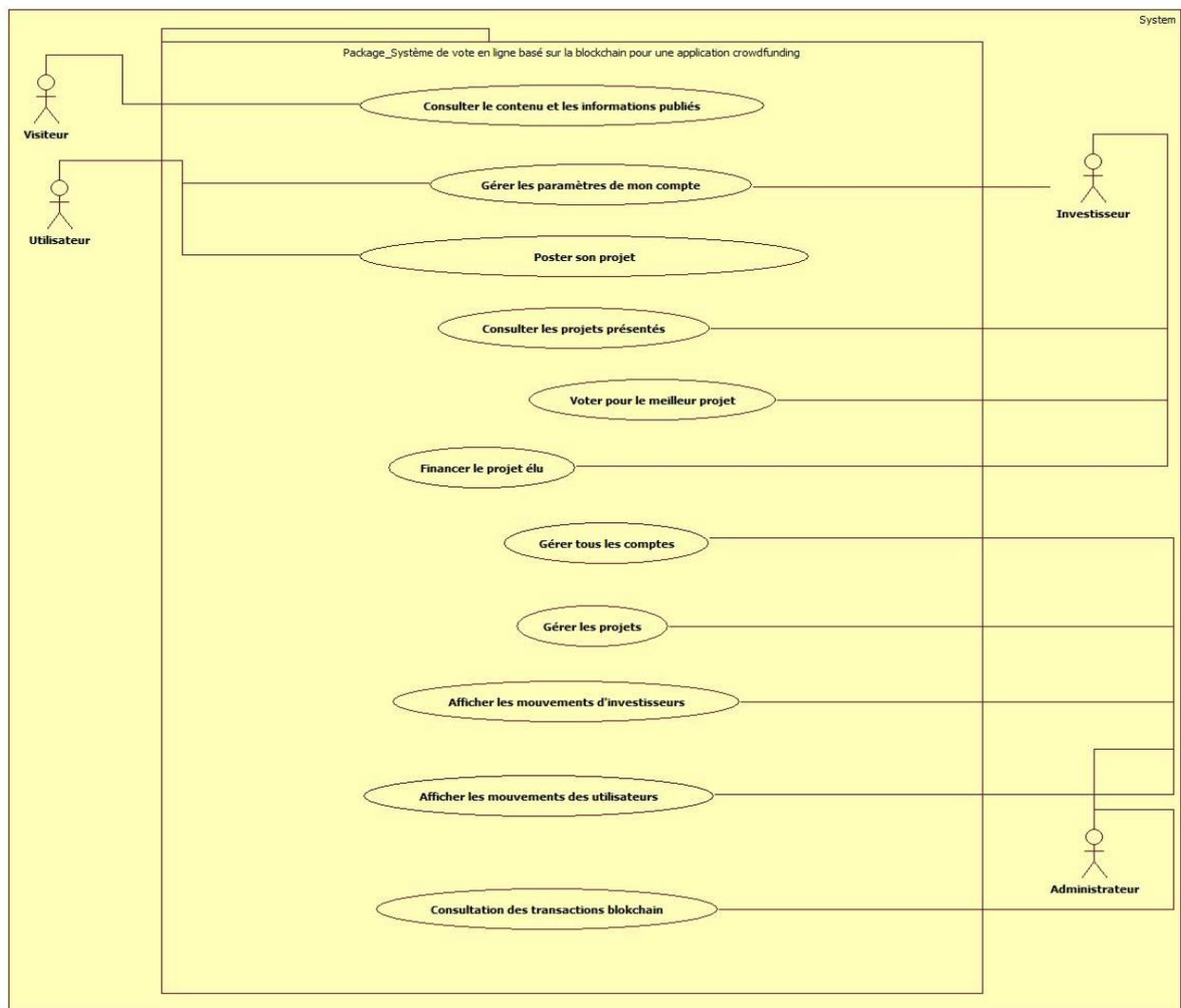


Figure 10 : Diagramme de cas d'utilisation global

Les fonctionnalités offertes à l'administrateur sont les suivants :

- Gestion des comptes.
- Consultation de l'historique des transactions de la Blockchain.
- Consultation des mouvements effectués dans le système.

Quant à l'utilisateur, les modules qui lui sont offerts sont :

- Configuration du compte utilisateur et consultation de ses notifications.
- Consultation des mouvements.

4.1. Gérer les paramètres de mon compte : Cette fonctionnalité, illustrée dans la figure 10, offre aux utilisateurs la possibilité de créer et de consulter des comptes.

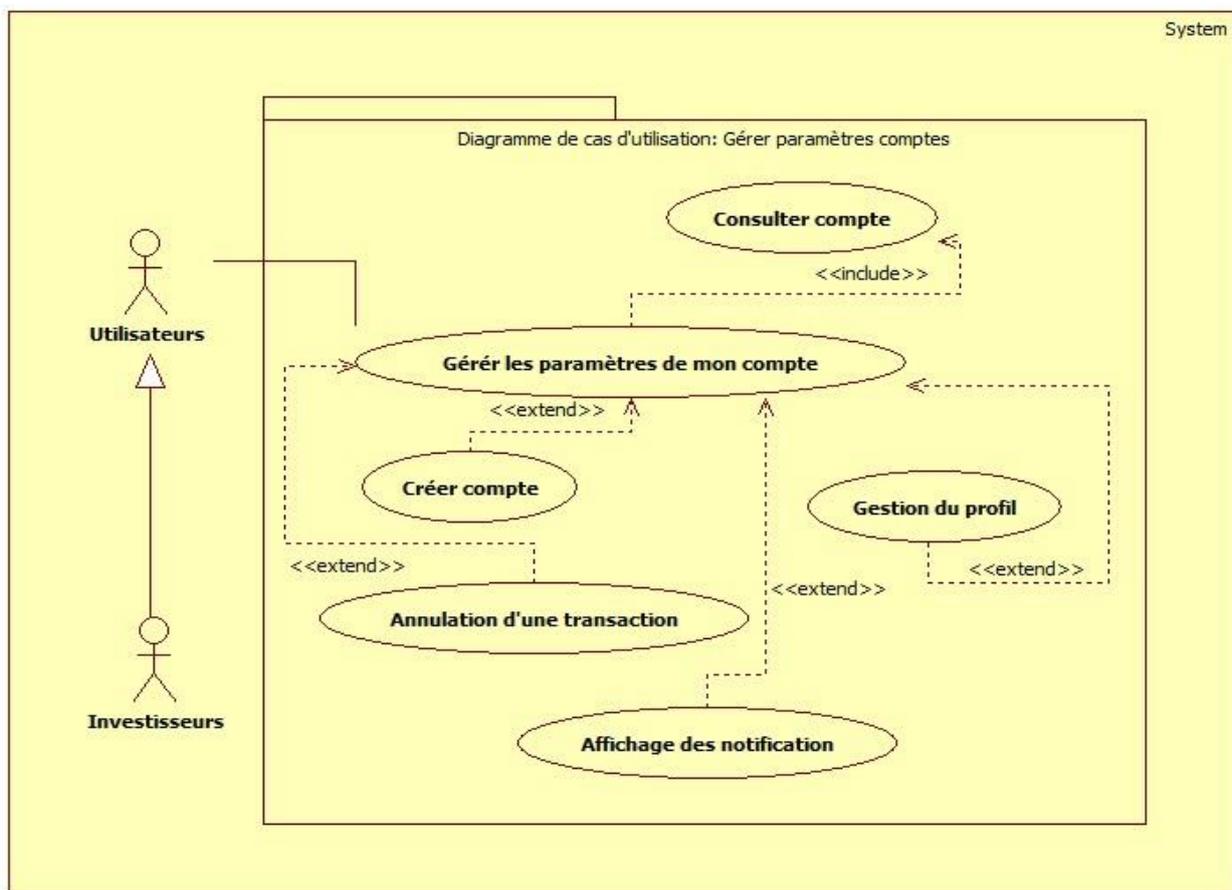


Figure 11 : Raffinement de diagramme de cas d'utilisation 'Gérer les paramètres de mon compte'

- **Créer compte** : Le tableau 4.1 décrit l'enchaînement du cas d'utilisation « Créer compte ».

Tableau 4.1.1: Description textuelle du cas d'utilisation « Créer compte »

Titre : Créer compte
Acteur : Utilisateur
Précondition : l'utilisateur doit être authentifié en tant que porteur de projet ou investisseur.
Description : Ce cas d'utilisation permet à un utilisateur d'ajouter un compte
Scénario principal : 1. L'utilisateur choisit de créer un compte. 2. Le système vérifie si l'utilisateur peut ajouter un compte. 3. Le système affiche l'interface de création du compte. 4. L'utilisateur saisit l'identifiant et le mot de passe souhaité. 5. L'utilisateur envoie les données saisies dans le formulaire en cliquant sur le bouton « Submit ». 6. Le système vérifie la validité des données saisies par l'utilisateur. 7. Le nouveau compte est stocké dans la base de données.
Scénario alternatif 1 : 1. L'utilisateur ne peut plus ajouter un compte suite à sa possession d'un compte disponible dans le système. 2. Le système affiche un message d'erreur indiquant l'invalidité de l'action.
Scénario alternatif 2 : 1. L'utilisateur saisit un identifiant déjà existant ou veut soumettre un champs vide. 2. Le système affiche un message d'erreur indiquant l'invalidité du code saisi.

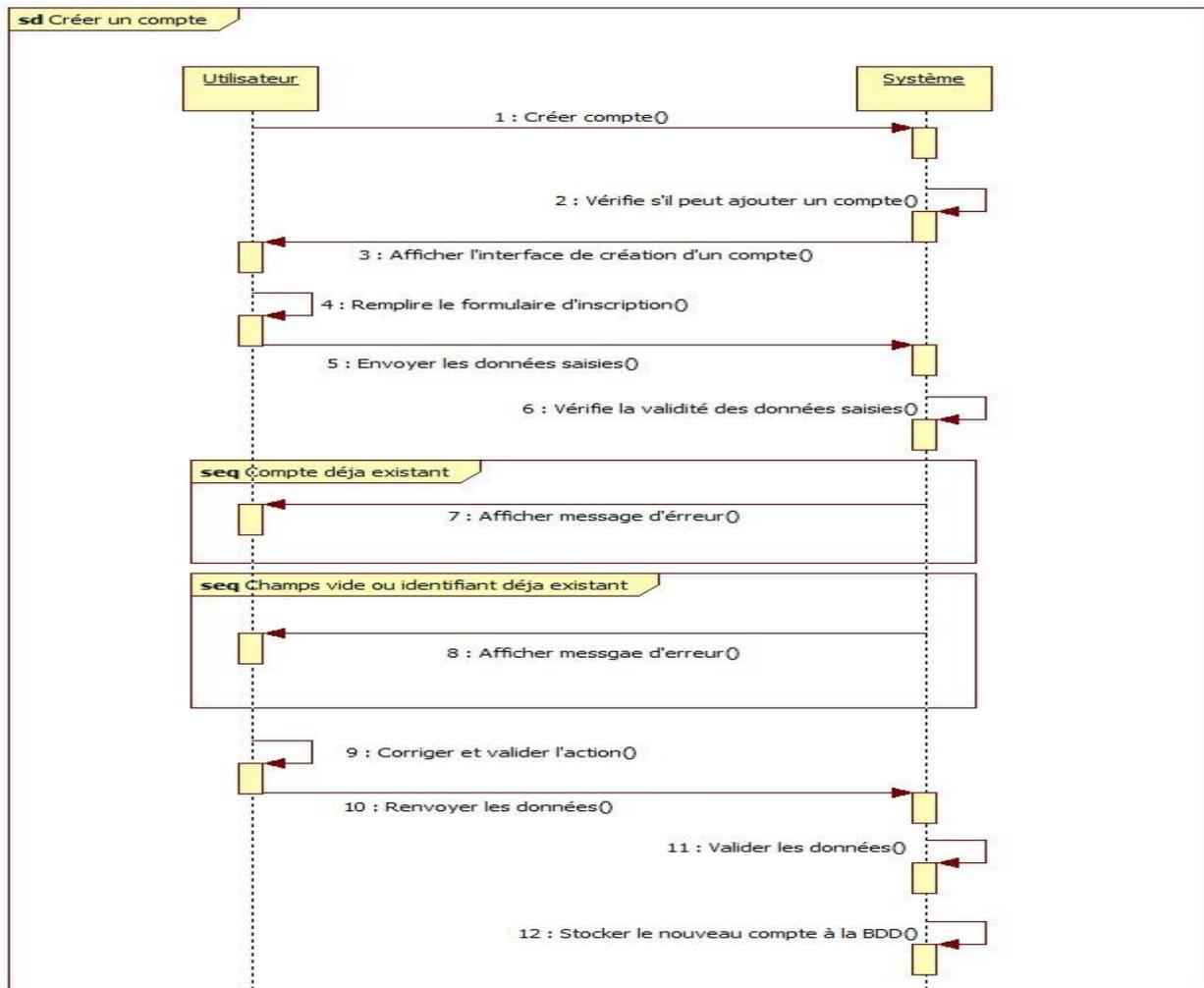


Figure 12 : Diagramme de séquence : 'Créer un compte'

- **Consulter compte** : Le tableau 4.1.2 décrit l'enchaînement du cas d'utilisation « Consulter compte ».

Tableau 4.1.2 : Description textuelle du cas d'utilisation « Consulter compte »

Titre : Consulter compte
Acteur : Utilisateur
Précondition : l'utilisateur doit être authentifié en tant que porteur de projet ou investisseur.
Description : Ce cas d'utilisation permet à un utilisateur d'afficher son compte
Scénario principal :
1. L'utilisateur choisit d'afficher son compte.
2. Le système affiche toutes les informations du compte d'utilisateur : le Nom, Prénom, sexe, code postal et pays de résidence, adresse, mot de passe, adresse mail ex ..

- **Annulation d'une transaction** : La figure 13 présente le diagramme de séquence relatif au cas d'utilisation « Annulation transaction ». En effet, lors de la demande de l'annulation de la transaction sauvegardée dans le système, notre système donne l'ordre à la Blockchain d'annuler la transaction et donc d'annuler tous les mouvements sauvegardés liés à la transaction annulée et de mettre à jour les statuts des titres possédés par les intervenants dans la transaction. Le système vérifie par la suite si ces derniers autorisent l'envoi des notifications par mail. Si c'est le cas, il leurs envoi un mail indiquant l'annulation de la transaction et la raison de cette action. Enfin, le système leurs envoi des notifications indiquant la nature de la transaction effectuée.

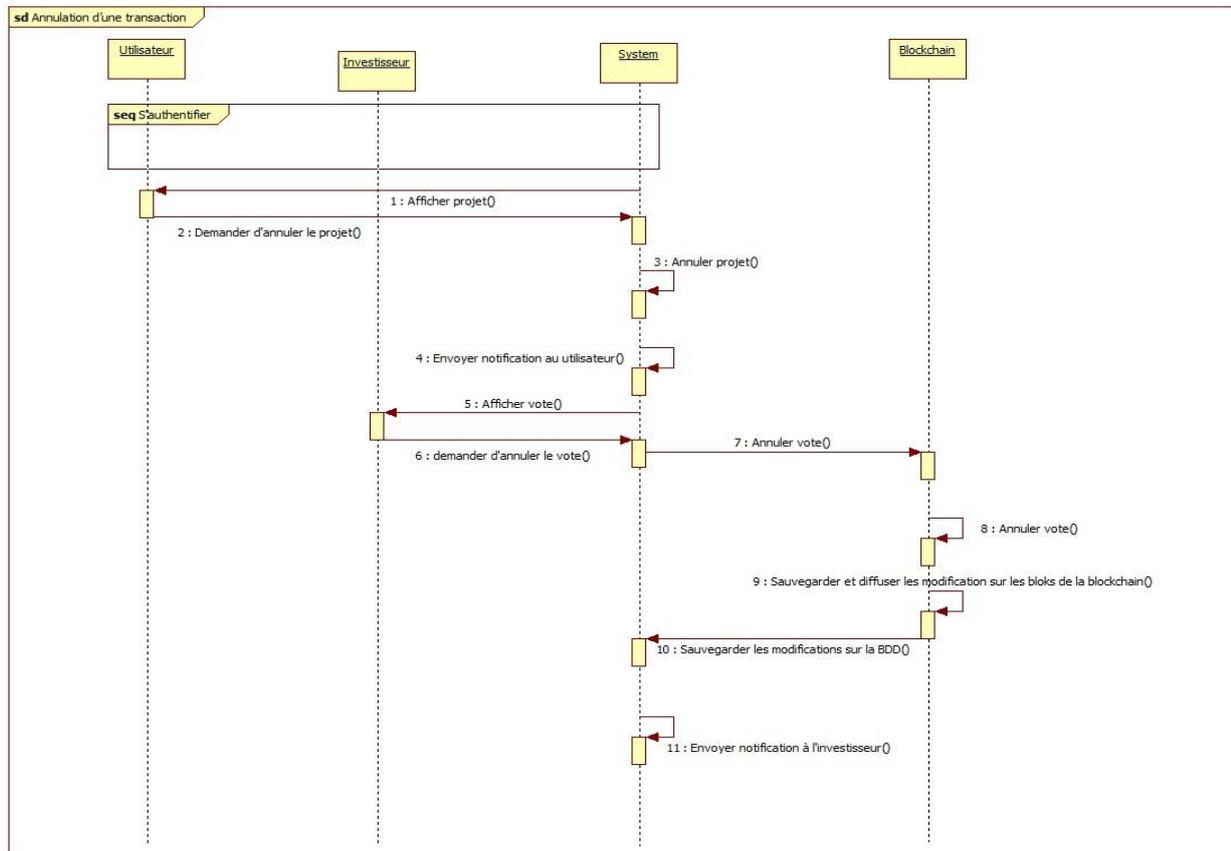


Figure 13 : Diagramme de séquence système relatif au cas d'utilisation 'Annulation d'une transaction'

4.2. Poster son projet : Cette fonctionnalité, illustrée dans la figure 10, offre aux utilisateurs la possibilité de déposer et de publier leur projet sur l'application.

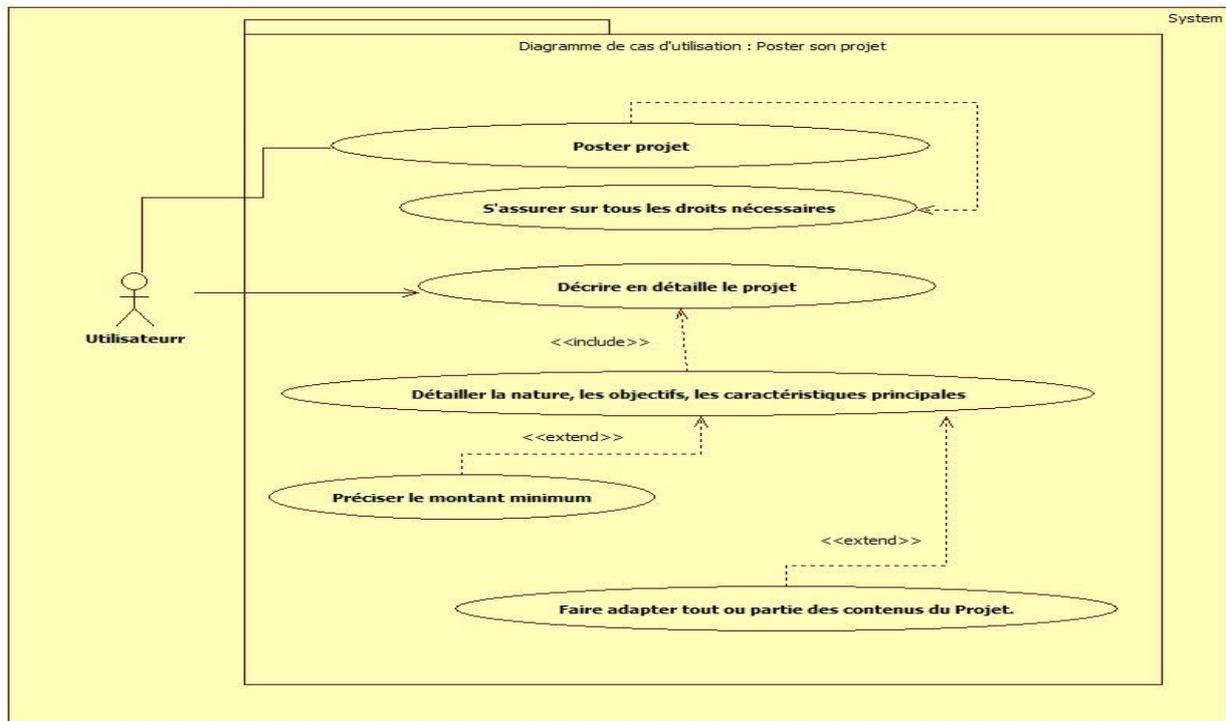


Figure 14 : Raffinement de diagramme de cas d'utilisation 'Poster son projet'

- **Décrire en détail le projet** : Le tableau 4.2 décrit l'enchaînement du cas d'utilisation « Décrire en détail le projet ».

Tableau 4.2: Description textuelle du cas d'utilisation « Décrire en détail le projet »

Titre : Décrire en détail le projet
Acteur : Utilisateur
Précondition : l'utilisateur doit être authentifié en tant que porteur de projet.
Description : Ce cas d'utilisation permet à un utilisateur poster son projet.
Scénario principal :
1. Le Porteur de Projets s'engage à fournir de manière claire et précise, pour tout Projet proposé : la description du Projet, le montant minimum que doivent atteindre les Contributions pour que le Projet soit considéré comme financé, la Période de Collecte ainsi que les Contreparties qu'il propose. 2. Il appartient au Porteur de Projets de s'assurer qu'il a tous les droits nécessaires pour publier le Projet et qu'il est en conformité avec toutes les lois et tous les règlements applicables à la conception et la réalisation du Projet concerné.
Scénario alternatif 1 :
1. Le Porteur de Projets reconnaît que la fourniture d'informations trompeuses, incomplètes ou erronées est susceptible d'engager sa responsabilité à l'encontre de la Société et des Utilisateurs et assume l'entière responsabilité des conséquences résultant de toute omission ou négligence à cet égard.

4.3. Voter pour le meilleur projet : Cette fonctionnalité, illustrée dans la figure 10, offre aux investisseurs l'opportunité de sélectionner le meilleur projet présenté.

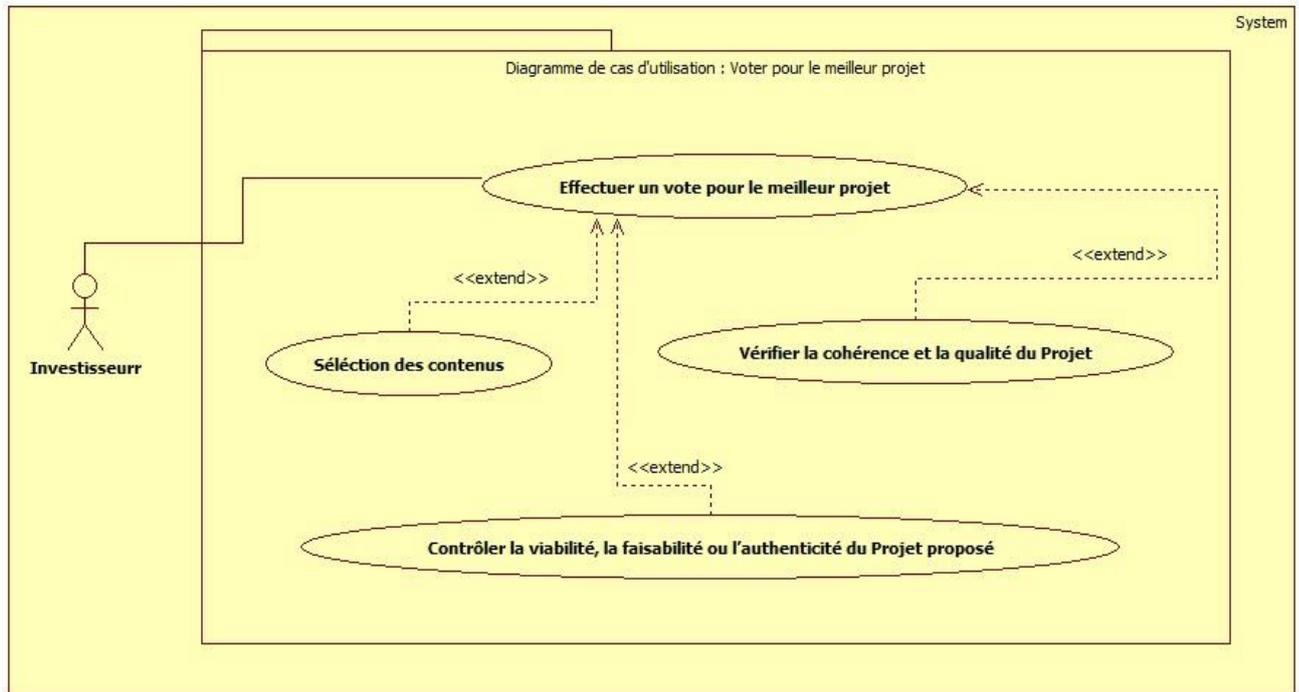


Figure 15 : Raffinement de diagramme de cas d'utilisation 'Voter pour le meilleur projet'

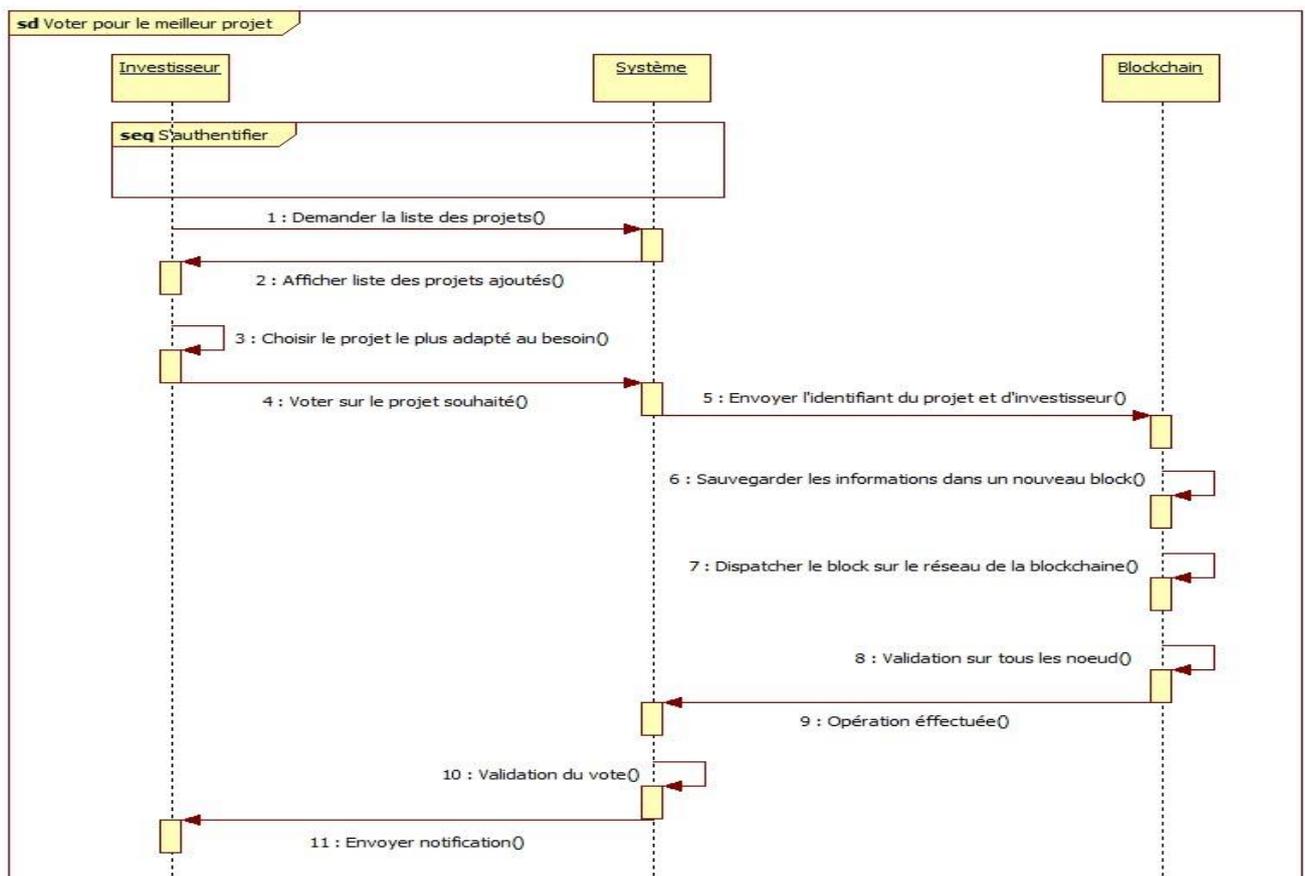


Figure 16 : Diagramme de séquence : 'Voter pour le meilleur projet'

- **Sélection des contenus** : Le tableau 4.3 décrit l'enchaînement du cas d'utilisation « Sélection des contenus ».

Tableau 4.3: Description textuelle du cas d'utilisation « Sélection des contenus »

Titre : Sélection des contenus
Acteur : Investisseur
Précondition : l'utilisateur doit être authentifié en tant qu' investisseur.
Description : Ce cas d'utilisation permet à un investisseur d'aperçu le projet innovant.
Scénario principal :
1. Filtrer les projets proposés par catégorie et par secteur.
2. Appliquer les conditions singé dans le smart contrat.
3. Solliciter le meilleur contenu.

4.4. Gérer tous les comptes par l'administrateur : : Cette fonctionnalité, donne l'avantage à l'administrateur de faire la gestion des comptes utilisateurs ou investisseurs.

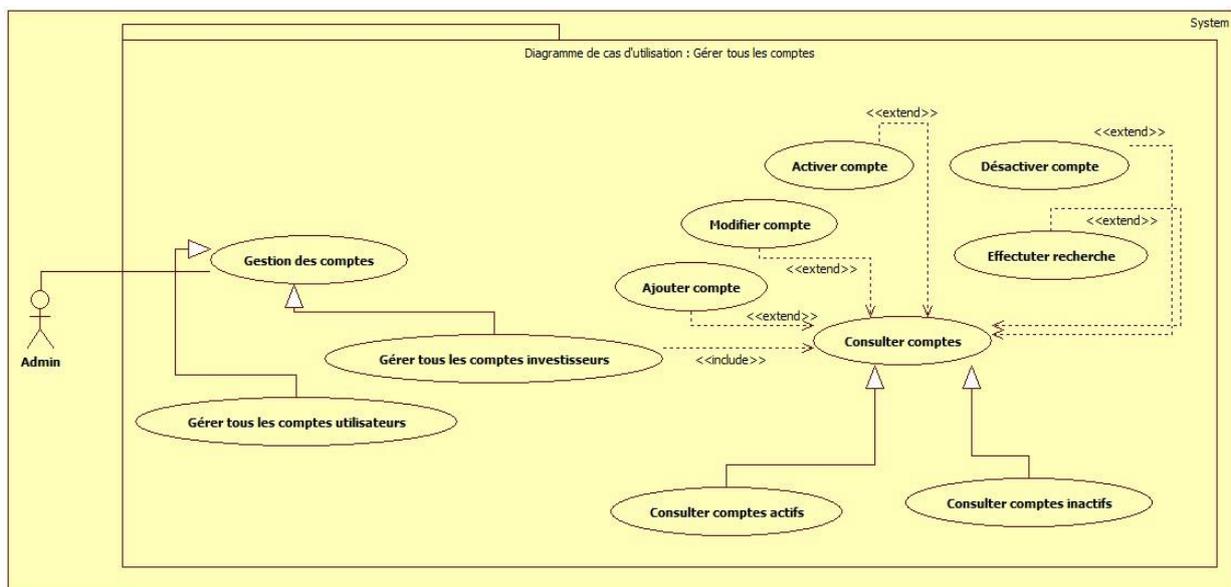


Figure 17 : Raffinement de diagramme de cas d'utilisation 'Gérer tous les comptes'

- **Création d'un compte** : Le tableau 4.4 décrit l'enchaînement du cas d'utilisation « Ajouter un compte ».

Tableau 4.4.1 : Description textuelle du cas d'utilisation « Ajouter un compte »

Titre : Ajouter un compte
Acteur : Administrateur
Précondition : l'utilisateur doit être authentifié en tant qu' administrateur.
Description : Ce cas d'utilisation permet d'ajouter un nouveau compte.
Scénario principal :
1. L'administrateur choisit d'ajouter un compte.
2. Le système affiche le formulaire d'ajout.
3. L'administrateur saisit le nom du client, son adresse mail, son mot de passe, la confirmation du mot de passe et l'adresse IP du nœud se chargeant de l'exécution des transactions envoyées par cet utilisateur-là.
4. L'administrateur envoie les données saisies dans le formulaire en cliquant sur le bouton

« Submit ».

5. Le système vérifie les données saisies par l'administrateur.

6. Le nouveau compte est stocké dans la base de données.

Scénario alternatif :

1. Les champs saisis par l'administrateur sont invalides (adresse email invalide, taille du nom de l'utilisateur est inférieure 6 ou supérieure à 32, taille du mot de passe inférieure à 8 ou supérieure à 32 ou deuxième saisie du mot de passe erronée).

2. Le système affiche le message d'erreur correspondant au(x) champs invalide(s) saisi(s) par l'administrateur.

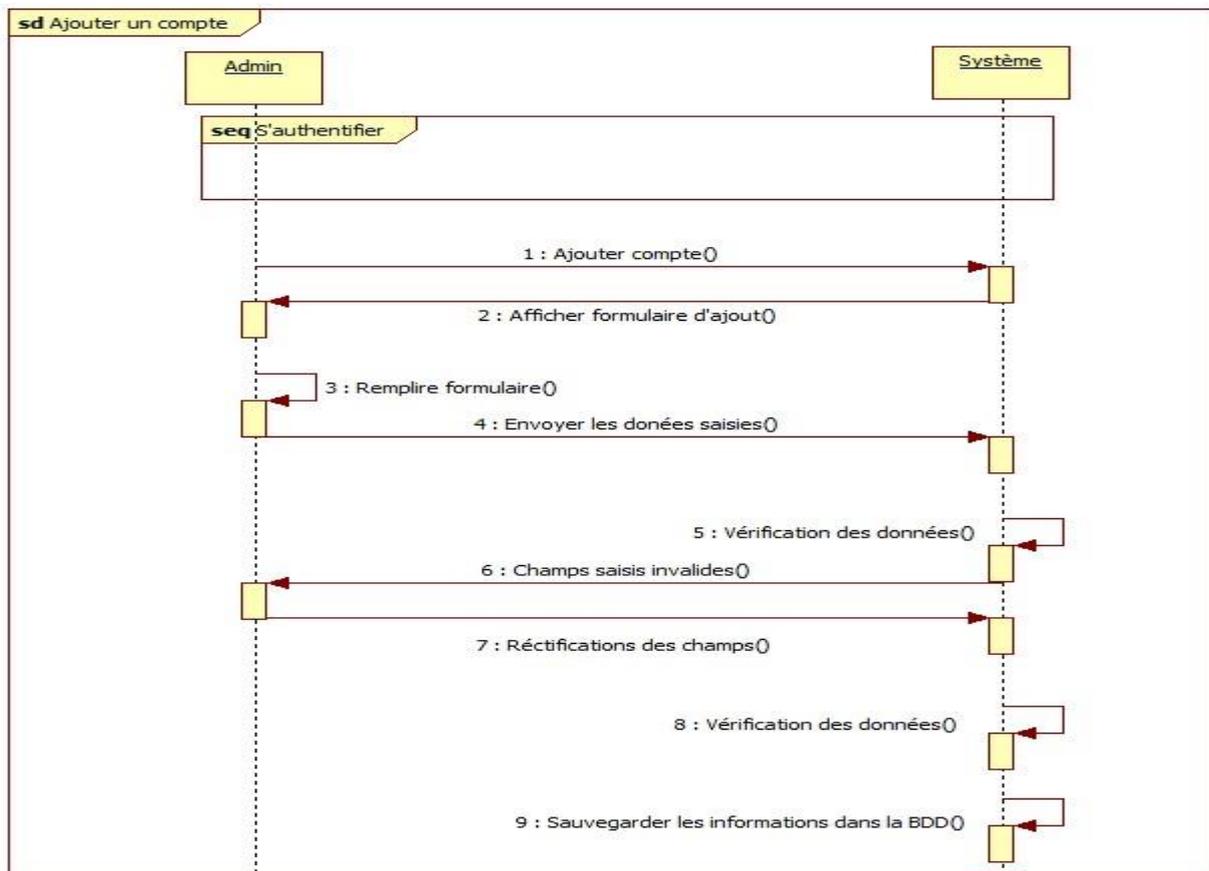


Figure 18 : Diagramme de séquence : 'Ajouter un compte'

- **Modification d'un compte** : Le tableau 4.4.2 décrit l'enchaînement du cas d'utilisation « Modifier un compte ».

Tableau 4.4.2: Description textuelle du cas d'utilisation « Modifier un compte »

Titre : Modifier un compte
Acteur : Administrateur
Précondition : la liste des comptes inactifs doit être affichée.
Description : Ce cas d'utilisation permet de modifier un compte tant qu'il est inactif.
Scénario principal :
1. L'administrateur choisit de modifier un compte.
2. Le système affiche le formulaire de modification à l'administrateur.
3. L'administrateur remplit le formulaire.
4. L'administrateur envoie les données saisies dans le formulaire en cliquant sur le bouton « Submit ».

5. Le système vérifie les données saisies par l'utilisateur.
6. Le système sauvegarde les modifications dans la base de données.

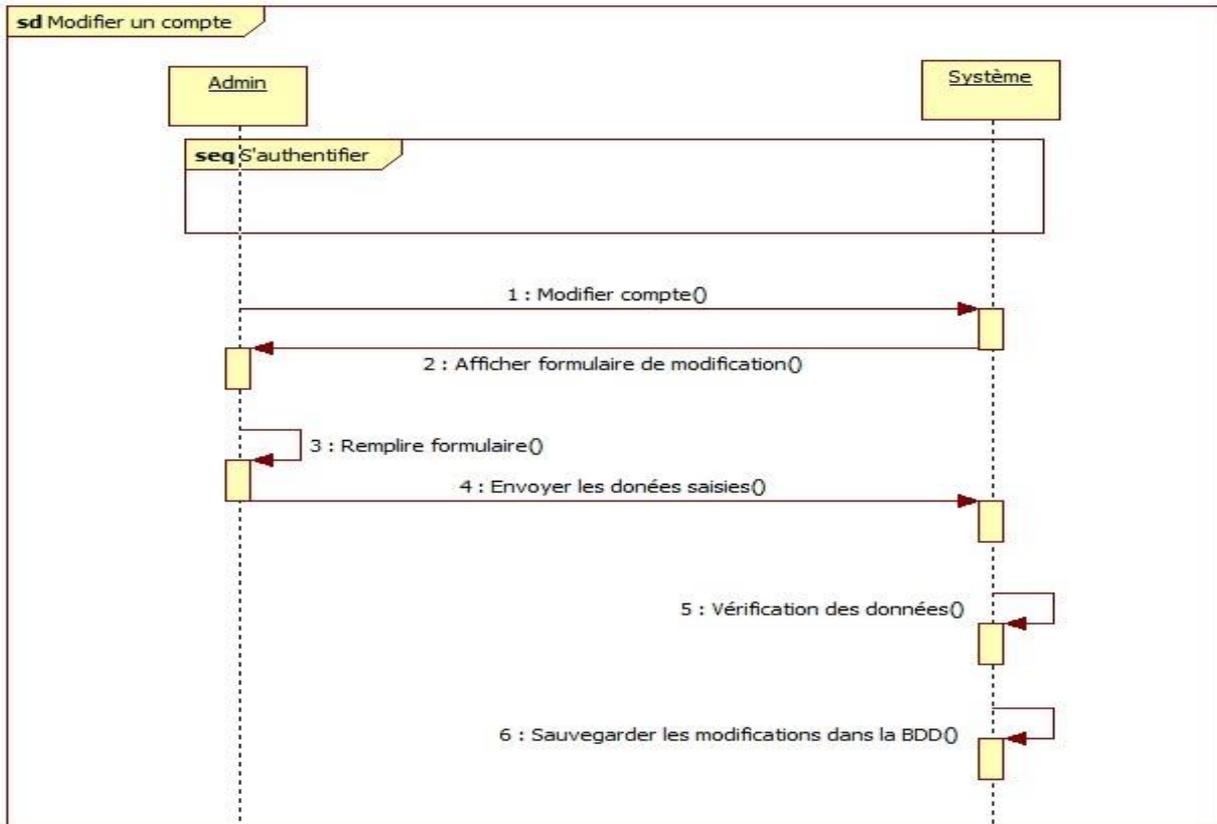


Figure 19 : Diagramme de séquence : 'Modifier un compte'

- **Consulter comptes** : Le tableau 4.4.2 décrit l'enchaînement du cas d'utilisation « Consulter comptes ».

Tableau 4.4.3: Description textuelle du cas d'utilisation « Consulter comptes »

Titre : Consulter comptes
Acteur : Administrateur
Précondition : l'utilisateur doit être authentifié en tant qu' administrateur.
Description : Ce cas d'utilisation permet d'afficher tous les profils actifs des utilisateurs.
Scénario principal :
1. L'administrateur choisit d'afficher les comptes des utilisateurs.
2. Le système affiche la liste des comptes.
Scénario alternatif :
1. L'administrateur choisit de désactiver un compte.
2. Le système sauvegarde le nouvel état du compte.

4.5. Consultation des transactions Blockchain : L'administrateur peut consulter les transactions sauvegardées dans les différents blocs de la Blockchain. Ceci peut être par exemple pour des raisons de contrôle ou de vérification de l'exactitude d'une valeur bien particulière dans la base de données externe.

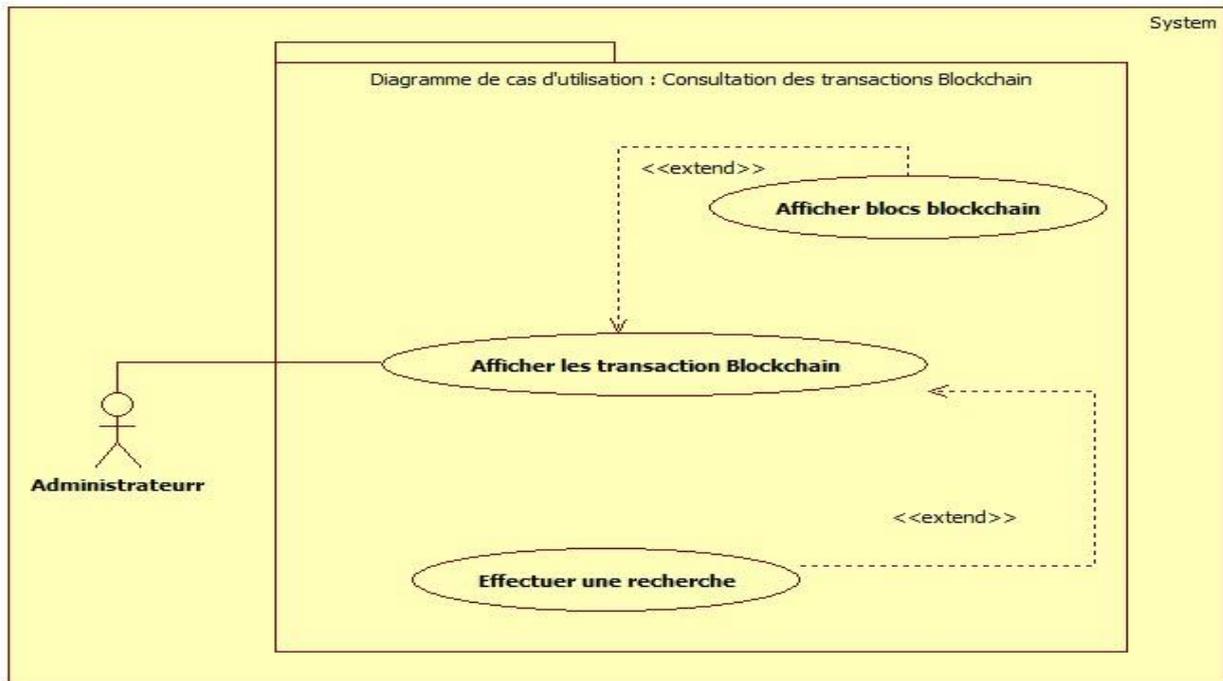


Figure 20 : Raffinement de diagramme de cas d'utilisation 'Consultation des transactions Blockchain'

- **Consultation des transactions Blockchain** : Le tableau 4.5 décrit l'enchaînement du cas d'utilisation « Consultation des transactions Blockchain ».

Tableau 4.5: Description textuelle du cas d'utilisation « Consultation des transactions Blockchain »

Titre : Consultation des transactions Blockchain
Acteur : Administrateur
Précondition : l'utilisateur doit être authentifié en tant qu' administrateur.
Description : Ce cas d'utilisation permet à un administrateur d'afficher toutes les transactions sauvegardées dans la Blockchain.
Scénario principal :
<ol style="list-style-type: none"> 1. L'administrateur choisit d'afficher les transactions. 2. Le système affiche la liste des dernières transactions de la Blockchain.
Scénario alternatif :
<ol style="list-style-type: none"> 1. L'administrateur choisit de consulter le contenu des blocs de la Blockchain. 2. L'administrateur saisit le numéro du bloc qu'il désire consulter. 3. Le système extrait les informations correspondantes au numéro du bloc saisi en cas où ce dernier est correct. 4. Dans le cas contraire, le système affiche un message d'erreur.

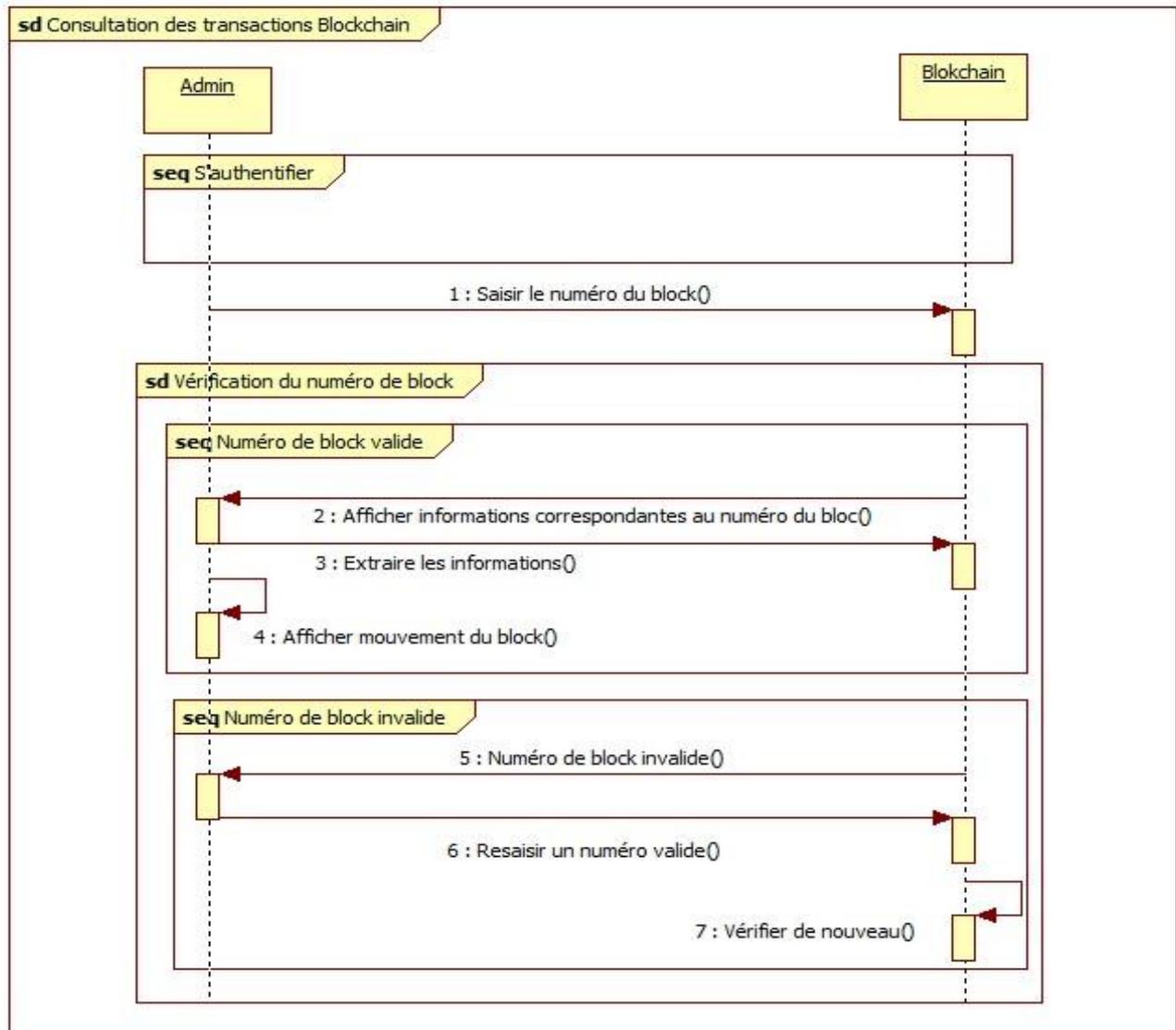


Figure 21 : Diagramme de séquence : 'Consultation des transactions Blockchain'

5. Diagrammes de classe

Nous allons maintenant présenter notre diagramme de classe lequel nous avons fait la conception de notre base de donnée de notre site web, la figure représente les différentes tables utilisées dans la spécification des besoins ainsi que leurs attributs qui décrivent le nature des objets abstraites en classes.

Comme montre la figure 22, l'admin fait la réalisation des comptes (User, Investisseur, Entreprise et Projet)

- Un utilisateur peut ajouter un ou plusieurs projets.
- Un investisseur peut ajouter un ou plusieurs entreprises selon les entreprises qu'il possède.
- Un investisseur peut voter une seule fois sur un seul projet qui existe.

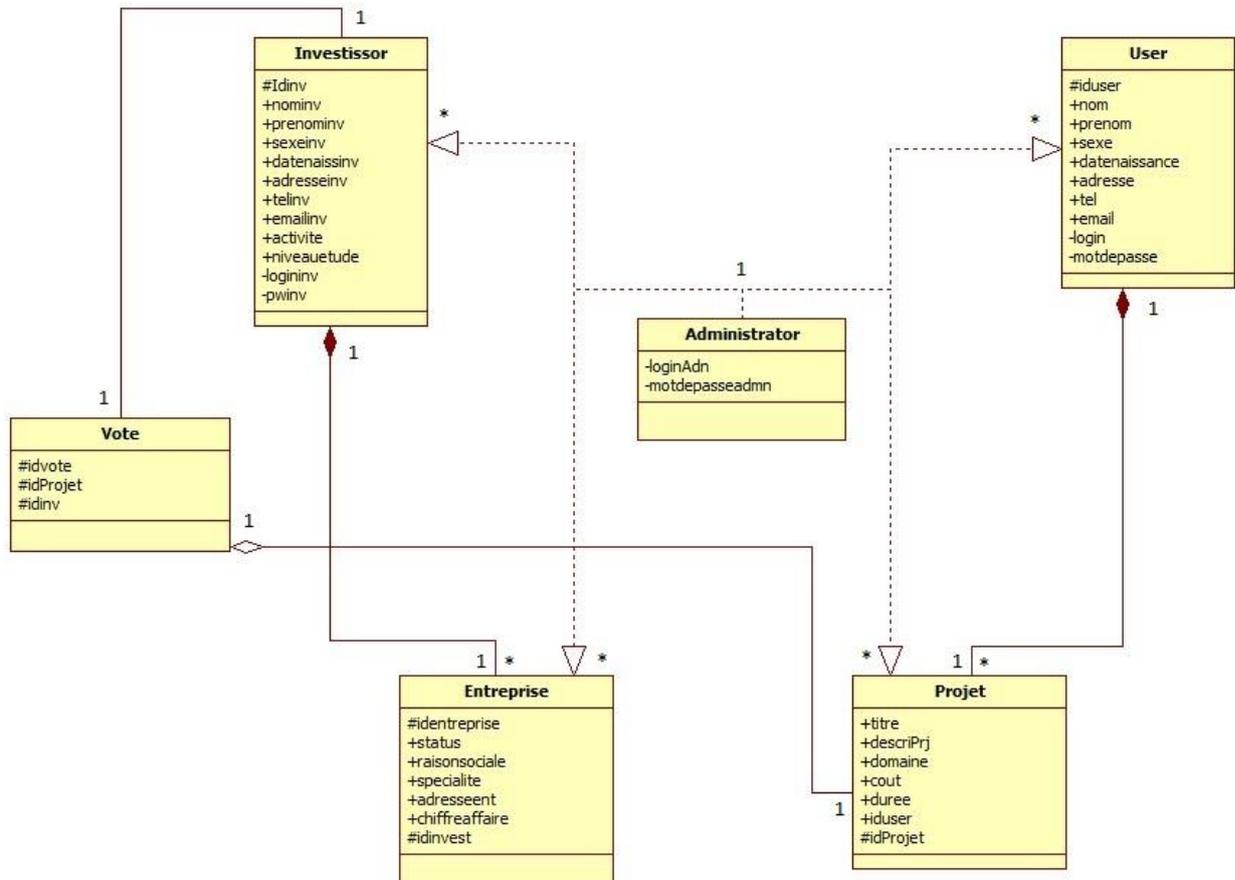


Figure 22 : Diagramme de classe

6. Conclusion

Dans ce chapitre, nous nous sommes intéressés à la spécification des besoins fonctionnels et non fonctionnels de notre solution. Nous avons décelé les cas d'utilisation ainsi que les différents acteurs qui y interviennent. Nous avons présenté et détaillé les cas d'utilisation de notre système en éclairant l'interaction entre les acteurs et le système. Dans le prochain chapitre, nous entamons l'implémentation de notre application.

CHAPITRE IV

Implémentation et réalisation

Plan

1. Introduction	41
2. Architecture de la solution	41
2.1. Architecture opérationnel	41
2.2. Architecture applicative	42
3. Déploiement de l'application	43
4. Environnement de développement	44
4.1. Environnement matériel	44
4.2. Environnement logiciel	44
4.3. Environnement de développement Netbeans	44
4.4. Serveur : Apache Tomcat 8.0	45
4.5. Serveur de base de donnée : WampServer	45
4.6. Technologies de programmation	45
5. Principaux développement	45
5.1 Vu générale sur les pages web du site	46
5.2 Configuration du réseau blockchain	54
6. Conclusion	63

1. Introduction

Nous nous intéressons dans ce dernier chapitre en premier lieu à l'architecture opérationnelle et l'architecture applicative de notre système. Ensuite, nous traitons la partie réalisation de l'application et la mise en œuvre des différents composants décrits au niveau du chapitre précédent. Nous décrivons, dans une première partie, les environnements matériel et logiciel. Dans une seconde partie, nous exposons le travail réalisé par des captures d'écran tout en détaillant les fonctionnalités réalisées.

2. Architecture de la solution

Dans cette section, nous présentons les architectures opérationnelle et applicative de notre application.

2.1 Architecture opérationnelle : Notre application dispose d'une architecture opérationnelle définie principalement par :

- Un client HTTP : Il représente le site web qui permet à son utilisateur d'accéder aux différentes interfaces de l'application,
- Application Web : qui englobe et héberge toutes les pages web de notre application à savoir les composants métiers et les méthodes d'accès à la base de données,
- Un tiers de données : qui est notre base de données « MYSQL » positionné sur le WampServer,
- Un réseau Blockchain : qui fournit un réseau de nœuds qui sont des conteneurs Web équipés des smart contracts déployés dans le réseau Ethereum, du registre partagé qui inclut la chaîne de blocs et la base de données !

Il suit donc l'architecture 4-tiers présentée dans la figure 23

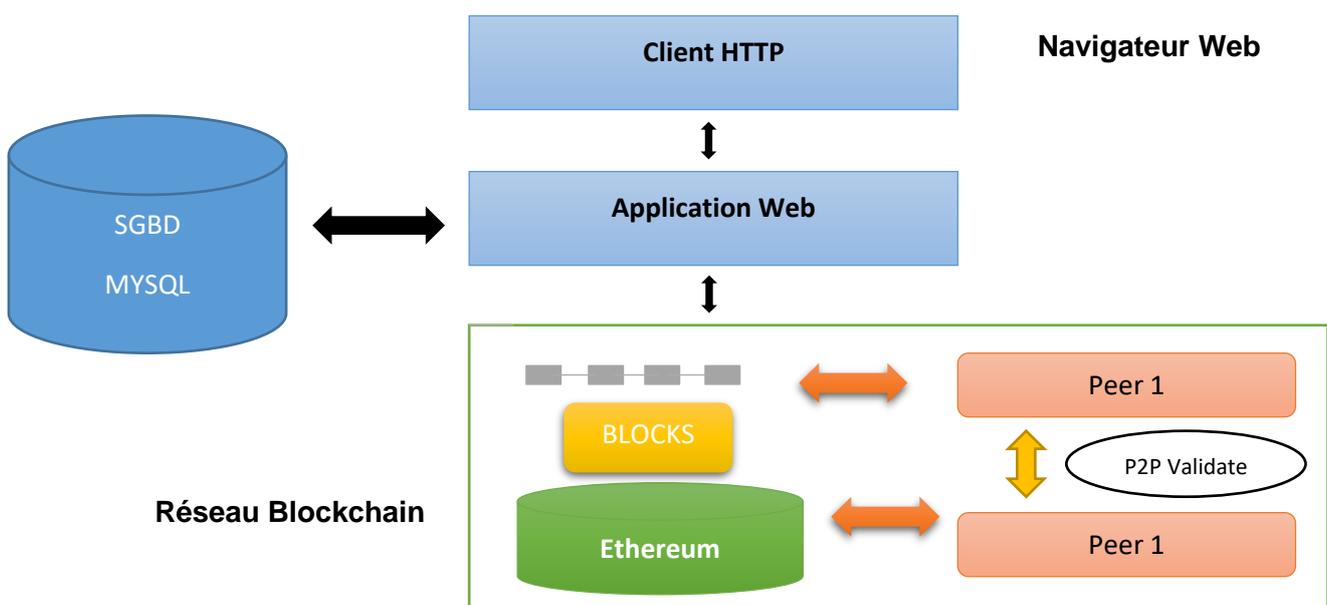


Figure 23 : Architecture opérationnelle de notre application

2.2 Architecture applicative : Notre application est basée, comme le montre la figure 24 sur une architecture de 3 couches.

Ces trois couches sont :

_ **La couche présentation** : permet d'intercepter les requêtes HTTP envoyées par le navigateur pour les traiter au niveau des contrôleurs et fournir les vues correspondantes comme réponse.

Elle se compose de la couche contrôleur qui assure une interaction entre les interfaces utilisateurs et les modèles de données, la couche modèle qui contient les modèles de données nécessaires pour assurer l'interaction entre le coté serveur et le client HTTP, et la couche vue qui rassemble les interfaces interagissant avec la couche métier à travers le contrôleur et les modèles de données.

_ **La couche métier** : correspond à la partie fonctionnelle de notre application. Elle gère le traitement dans le corps du système à savoir la gestion des comptes utilisateurs, la gestion des projets ainsi que l'interaction avec le réseau Blockchain. Ces traitements communiquent leurs résultats avec la couche présentation via le Contrôleur.

_ **La couche d'accès aux données (DAO)** : permet d'interagir avec les données persistantes du système. Elle est implémentée en se basant sur le Framework Truffle qui offre un ensemble de services d'accès aux données de la base *PHP my admin* utilisée.

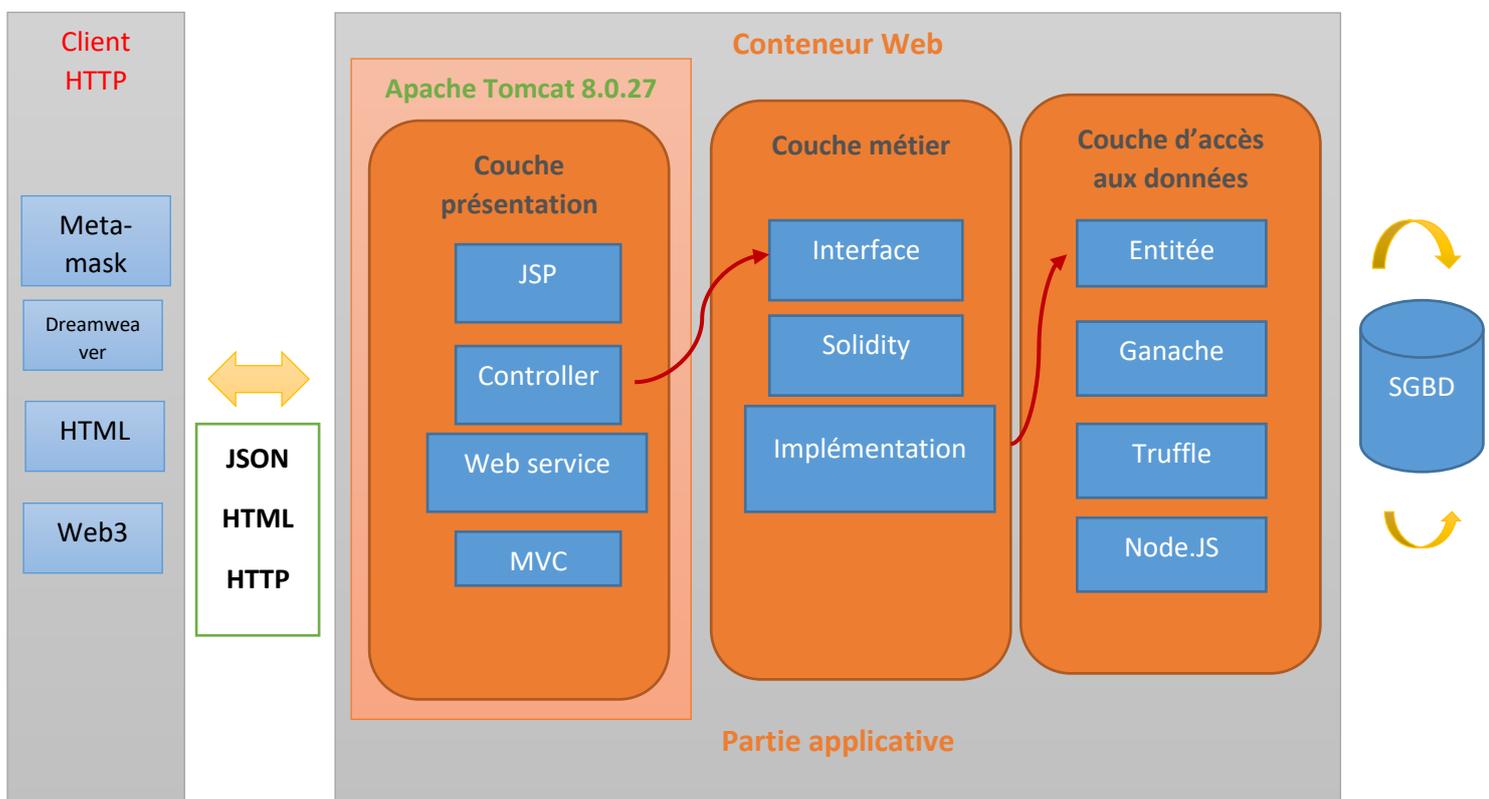


Figure 24 : Architecture Applicative de notre application

3. Déploiement de l'application

La figure 25 décrit la disposition physique du matériel du système et la répartition des entités logiques identifiées.

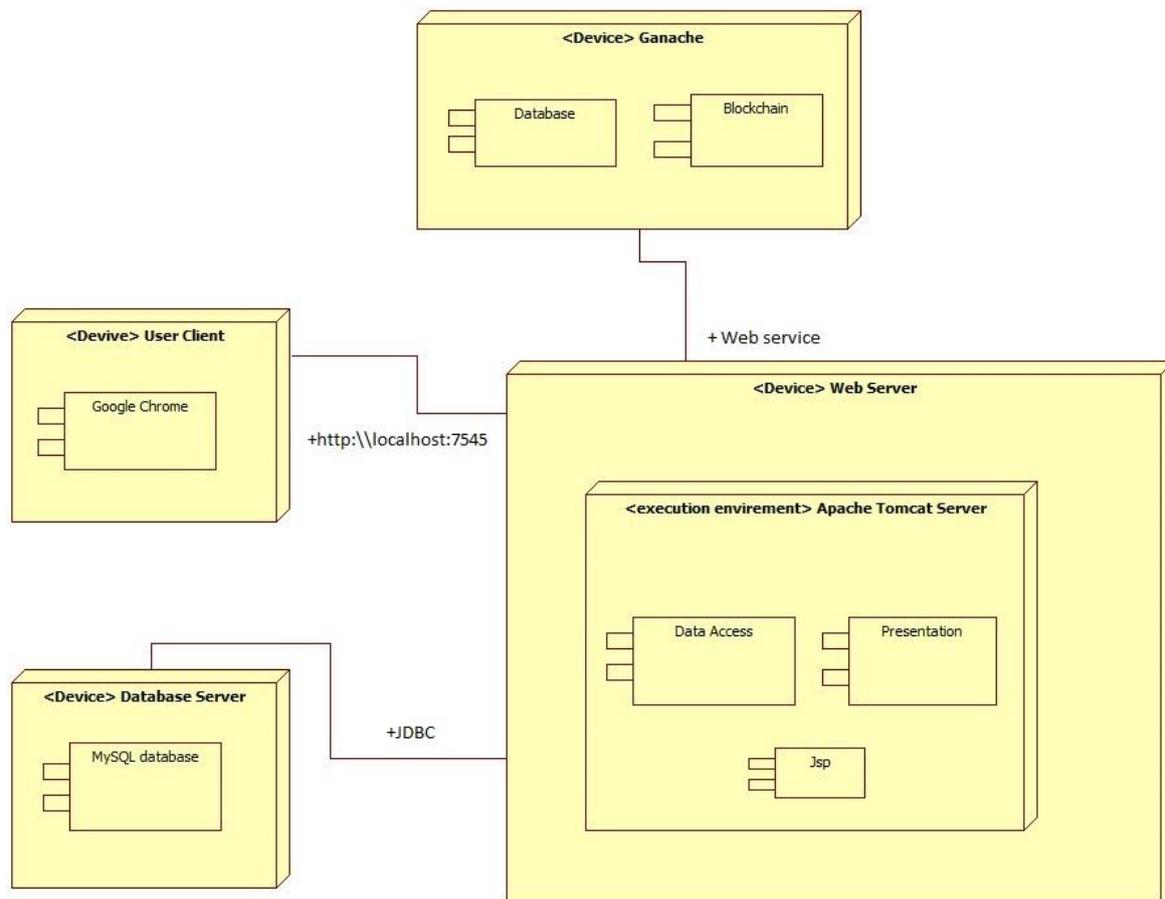


Figure 25 : Diagramme de déploiement de notre application

Notre solution s'exécute dans un serveur Web conteneur de Tomcat 8.0. Les différentes couches, à savoir la couche présentation, la couche métier et la couche d'accès aux données s'exécutent dans le serveur web Tomcat.

Notre application communique avec le réseau Blockchain moyennant le protocole HTTP et plus précisément à travers des Web Services. Concernant la base de données, nous avons choisi MySQL comme serveur de base de données. Quant à la mise en place du réseau Blockchain, nous utilisons Ganache « une blockchain locale en mémoire » et de la base de données décentralisée.

4. Environnement de développement

4.1 Environnement Matériel :

- Marque : ASUS X751M ;
- Processeur : CORE I7 ;
- Capacité de disque dur : 1000 Go ;
- RAM: 8.00 GB.
- Système d'exploitation: Windows 10 Pro x64

4.2 Environnement Logiciel :

La mise en place de ce projet a nécessité un certain nombre de logiciels qui sont récapitulés dans la figure 26.



Figure 26 : Logiciels utilisés dans la réalisation de l'application

4.3 Environnement de développement Netbeans

Netbeans est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, Netbeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de *greffons*. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web)[22].

4.4 Serveur : Apache Tomcat 8

C'est une implémentation open source des spécifications JEE comme la Java Servlet, les JavaServer Pages(JSP), la technologie des Web sockets en java, etc. Il comporte un serveur http et il a été développé avec le langage Java, donc il tire profit de l'aspect de portabilité de langage puisqu'il peut s'exécuter dans n'importe quel système d'exploitation équipé d'une machine virtuelle java. Tomcat est basé sur le moteur de servlet Catalina, le moteur jsp Jasper et le connecteur http Coyote collaborant tous afin d'assurer l'exécution des applications Web [23].

4.5 Serveur de base de données : WampServer

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

4.6 Technologies de programmation

Outre l'environnement matériel et logiciel, l'implémentation de notre application a impliqué l'emploi d'un certain nombre de technologies. Nous avons été confronté donc à sélectionner les langages de programmation les plus adaptées.

- **Java** : Java est un langage typé et orienté objet. Il est compilé et basé sur une architecture logicielle très particulière nécessitant une machine virtuelle Java. Il utilise les notions de la programmation orientée objet. Il fournit un ensemble énorme de bibliothèques standard couvrant plusieurs domaines. Nous avons utilisé le Java Development Kit JDK 8 pour le développement des smart contracts [24].
- **JavaScript** : JavaScript est un langage de script léger, orienté objet, principalement connu comme le langage de scripts des pages Web interactives. Il est incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages Web. Donc le JavaScript va se mettre au service de la page Html pour assurer le dynamisme de la page [25]

5. Principaux développements

Dans cette partie, nous allons présenter à travers un enchaînement de captures d'écran un Aperçu général sur les différentes fonctionnalités de l'application.

5.1 Vue générale sur les pages web du site :

Nous allons présenter en premier lieu la page d'accueil de notre site web :



Figure 27 : Page d'accueil su site



Figure 28 : Page d'accueil su site -suite-

La page d'accueil se compose principalement sur 6 champs comme la figure le montre en haut, le premier champ est « l'accueil du site », ensuite on a « la barre de recherche » pour rechercher un projet sur le site, après il y a le champ « se connecter » qui est réservé pour entrer dans le compte d'utilisateur ou pour faire une nouvelle inscription pour les nouveaux arrivés, le champ « administrateur » comme son nom l'indique est consacré pour faire les

tâches administratives qu'on va les détaillés avec plus d'informations, l'avant dernier champ est « proposer un projet » on a l'ajouter pour faire le dépôt des projet par les porteur de projet, en fin on a le champ « voter » qui est le but de création de ce site qui s'engage à faire le vote en ligne à l'aide de la plateforme blockchain.

Passant maintenant au champ « se connecter » en cliquant sur ce champ-là, on va se rediriger automatique dans cette page :

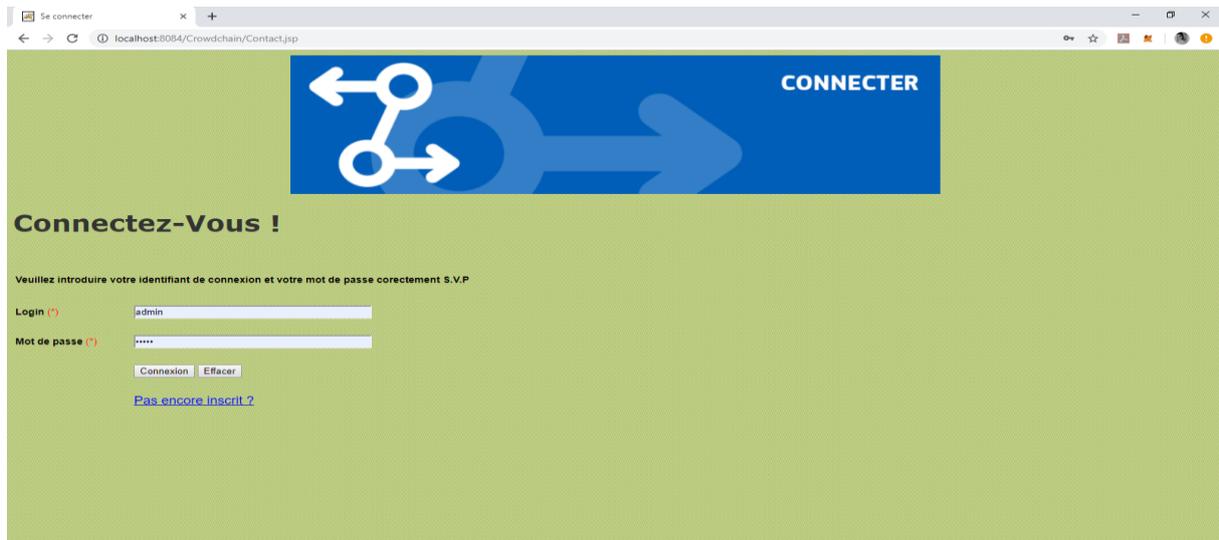


Figure 29 : Page d'ouverture d'une nouvelle session ou une session existante

Si vous possédez déjà un compte, il suffit juste d'entrer votre login et votre mot de passe pour vous redirigez vers votre session, sinon vous devez faire une inscription en cliquant sur le champ Pas encore inscrit ?

Si vous choisissez de faire votre inscription vous devez préciser votre statut, porteur de projet ou investisseur comme l'affiche la figure 30.

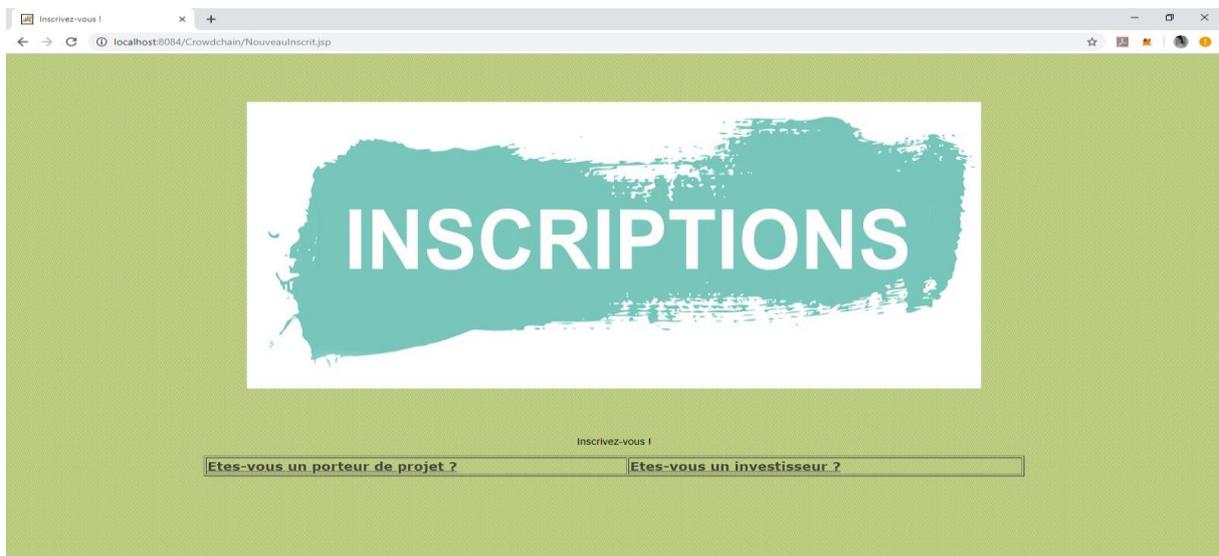


Figure 30 : Page d'inscription

Dans le cas où vous avez choisi Porteur de projet voilà le formulaire qui va s'afficher :

Inscription porteur de projet

localhost:8084/Crowdchain/PorteurProjet.jsp

Formulaire d'inscription pour les porteurs de projet

Instructions :

- 1- Prévoyez 5 minutes pour compléter le formulaire.
- 2- En tout temps, vous pouvez sauvegarder le formulaire. Vous devez cependant avoir une adresse courriel. Un mot de passe vous sera envoyé pour pouvoir poursuivre à un moment qui vous conviendra mieux.
- 3- Les champs marqués par un astérisque **rouge sont obligatoires**.
- 4- Veuillez remplir chacune de section et cliquer sur « Envoyer » lorsque vous avez terminé.

- Nom (*)

- Prénom (*)

- Sexe (*) M ▾

- Date naissance (*)

- Adresse (*)

- Téléphone (*)

- e-mail (*) admin

- Login (*)

- Mot de passe (*)

Envoyer Annuler

Figure 31 : Formulaire d'inscription pour les porteurs de projet

Dans le cas où vous avez Investisseur voilà le formulaire qui va s'afficher :

Inscription investisseur

localhost:8084/Crowdchain/Investisseur.jsp

Formulaire d'inscription pour les investisseurs

Instructions :

- 1- Prévoyez 5 minutes pour compléter le formulaire.
- 2- En tout temps, vous pouvez sauvegarder le formulaire. Vous devez cependant avoir une adresse courriel. Un mot de passe vous sera envoyé pour pouvoir poursuivre à un moment qui vous conviendra mieux.
- 3- Les champs marqués par un astérisque **rouge sont obligatoires**.
- 4- Veuillez remplir chacune de section et cliquer sur « Envoyer » lorsque vous avez terminé.

- Nom (*)

- Prénom (*)

- Sexe (*) M ▾

- Date de Naissance (*)

- Adresse (*)

- Téléphone (*)

- e-mail (*) admin

- Activité (*) Sélectionnez ▾

- Niveau d'étude (*) Sélectionnez ▾

- Login (*)

- Mot de passe (*)

Envoyer Effacer

Figure 32 : Formulaire d'inscription pour les investisseurs

Une fois vous avez terminé votre inscription, vous pouvez se connecter et se rediriger vers la page d'accueil et vous faite votre choix soi de proposer votre projet si vous êtes un porteur de projet comme l'indique la figure 33.

Formulaire d'ajout un nouveau projet

- Titre (*)
- Description du projet (*)
- Domaine (*)
- Cout (*)
- Durée (*)

Envoyer Effacer

Figure 33 : Formulaire d'ajout un nouveau projet

Donc l'utilisateur doit remplir attentivement le formulaire et bien décrire son projet afin d'avoir plus de chance que ce dernier la soit élu.

Si vous n'été pas un porteur de projet, vous devez consulter la liste des projets proposés afin de faire votre vote sur le projet qui vous parait le plus intéressant et le plus adapté à vos besoin en tant qu'investisseur.

Liste des projets proposées :

Idet	Titres	Description du projet	Domaine	Coûts	Durée
10	InnoNautiCs	Innovation pour le Secteur du Développement Nautique dans la Zone Méditerranéenne	Robotique	15690000	17
6	AGRISLES	Avec leurs petites structures traditionnelles, leur environnement préservé, les Ailes de la Méditerranée peuvent répondre aux besoins d'une alimentation saine, d'un développement respectueux de l'environnement, qui sont autant de défis pour l'Europe toute entière. Mais pour cela les acteurs agricoles et ruraux insulaires ont besoin de travailler ensemble.	Automobile	15400000	12
7	EASY FINANCE	Un meilleur accès au financement favorise la viabilité des PME	Services Financiers	14500000	13
5	ACCELMED	ACCELMED vise à capitaliser les résultats des projets MED-KED and MACC-BAM	Industrie	12577000	12
11	KnowInG	Intelligence et Innovation de la Connaissance pour un développement durable	Technologies	1650000	19
8	Flormed	FLORMED est un projet européen de développement de l'horticulture méditerranéenne	Informatique	1590000	12
12	MEDISS	Renforcer les capacités d'innovation de la filière saveurs sur le pourtour Méditerranéen	Services Financiers	1569000	16
9	FREE-MED	Les territoires partenaires du projet présentent des situations diverses	Automobile	1360000	18

Faite votre vote !

N° Projet:

Contactez-nous sur :

Email: crowdchain@gamil.com
 Téléphone: 0770899529/0771343456

© Unlited. All rights reserved | Photos by Google Images | Design by BENHAMMOU

Figure 34 : Espace réservé pour les investisseurs

Parlant maintenant sur l'administrateur, voilà ce qui affiche devant l'admin une fois qu'il veut connecter à sa session, il doit introduire son identifiant et son mot de passe pour entrer dans sa session.

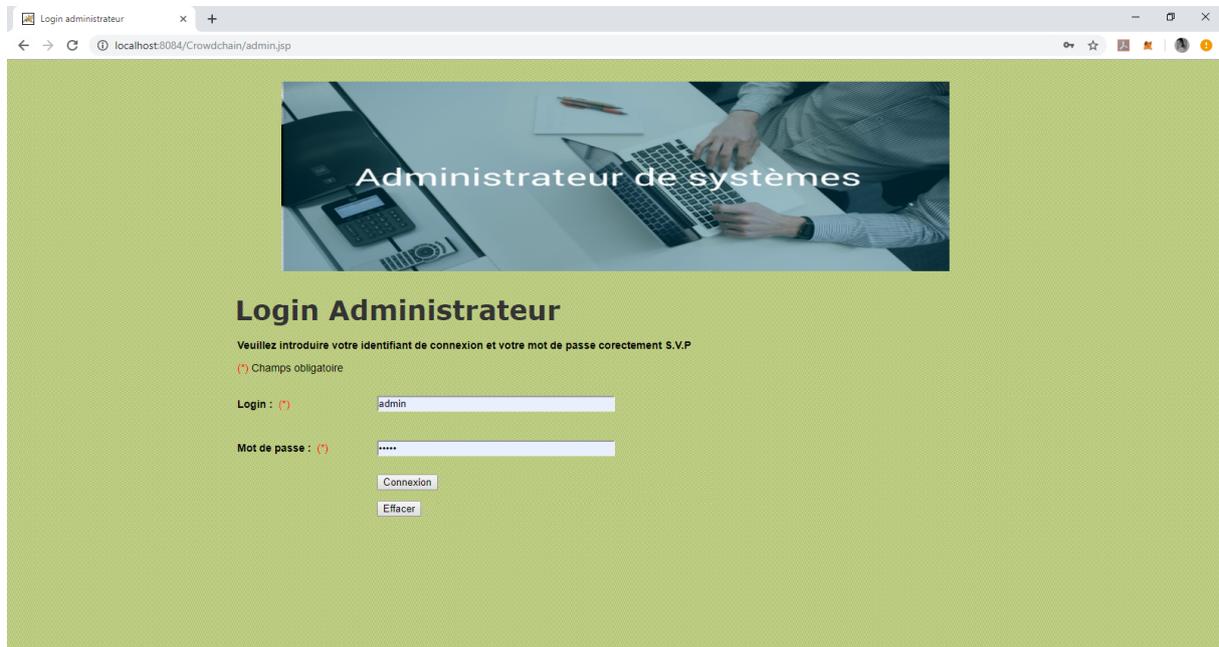


Figure 35 : Login administrateur

Une fois connecté, l'admin va se rediriger vers son espace de travaille comme l'indique la figure 36.

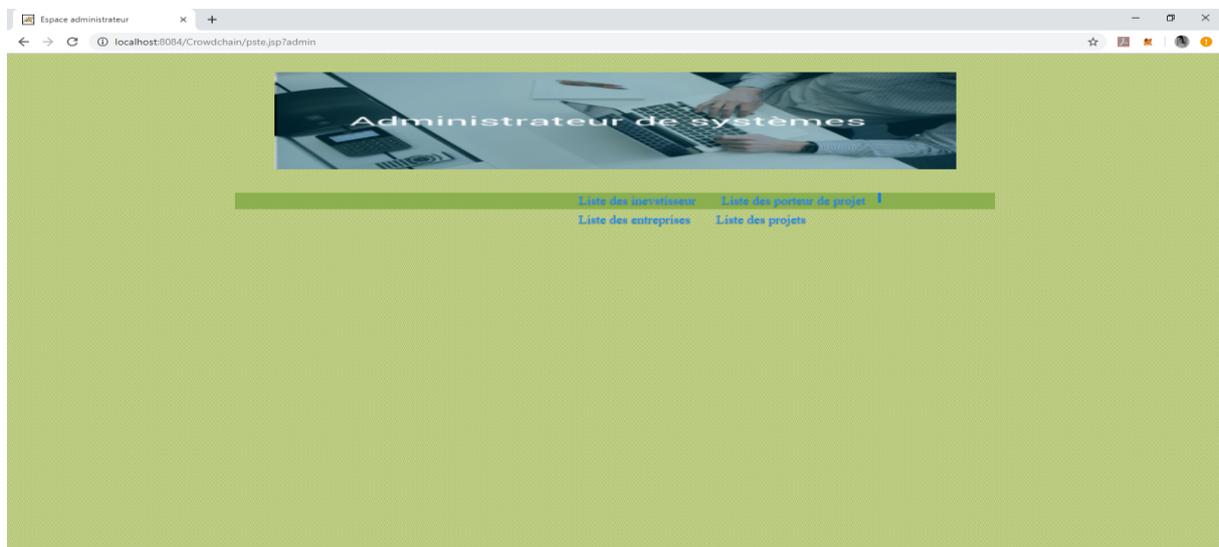


Figure 36 : Espace administrateur

La figure montre 4 champs réservés à l'admin qui se sont : liste porteur de projet, investisseur, entreprise et projet ces champs-là affiche le contenu ajouté par les utilisateurs du site dans la base de donnée.

Chapitre 4. Implémentation et réalisation

Liste des investisseurs



Liste des Investisseurs

ident	Nom	Prénom	Sexe	Date de naissance	Adresse	Téléphone	e-mail	Activité	Niveau d'étude
7	KOUBA	Mehdi	M	1990-05-01	Cité chahid Amirouch Gueima	078965452	di@gmail.com	Officier / Sous-officier	M2 / Bac+5
13	BENSAID	Hadj Ali	M	1990-09-09	Les ammandiers, Oran	0774589652	alibaba@gmail.fr	Ingenierie / Technique	L3 / Bac+3
11	SAFI	Farouk	M	1993-03-01	Alicante, Espagne	0034569874	firouk@hotmail.fr	Informatique	Professeur
14	KASMI	Asma	F	1994-03-08	Glatia, Chier	045879856	asma.fr@gmail.com	Enseignement / Formation	Ingénieur
12	BENAMMAR	Josef	M	1988-06-08	Amsterdam, Holland	00362159874	sefo@gmail.com	Creation - Multimedia	L3 / Bac+3
8	AMARA	Ahlem	F	1998-03-03	Cité salem Kharouba, Mostaganem	0776877459	lem@gmail.com	Commercial - Marketing	M2 / Bac+5
10	HAMMOU	Hakim	M	1990-09-08	Nancy, France	00332547896	kim@gmail.com	Administration / Secretariat	Docteur

Figure 37 : Liste des investisseurs

Liste des porteurs de projet



Liste des Porteurs de projet

ident	Nom	Prénom	Sexe	Date de naissance	Adresse	Téléphone	e-mail
6	MESKI	Kaddour	M	1995-01-03	Sidi lakher, Mostaganem	0778745854	null
4	MADANI	Naima	F	1985-04-06	Biskra, Algérie	0554878965	null
7	BOUZOUR	Amine	M	1980-07-09	Les sablières, Mostaganem	0770125887	null
3	BELHAMITI	AHMED	M	1993-02-09	Cité 3 frères Gheizan	0771458987	null
5	BAKHTI	Sara	F	1989-05-06	Annaba, Algérie	0687898541	null

Figure 38 : Liste des porteurs de projet

Liste des projets



Liste des projets

ident	Titres	Description du projet	Domaine	Coûts	Durée
10	InnoNauTICS	Innovation pour le Secteur du Développement Nautique dans la Zone Méditerranéenne	Robotique	15690000	17
6	AGRISLES	Avec leurs petites structures traditionnelles, leur environnement préservé, les Aôtes de la Méditerranée peuvent répondre aux besoins d'une alimentation saine, d'un développement respectueux de l'environnement, qui sont autant de défis pour l'Europe toute entière. Mais pour cela les acteurs agricoles et ruraux insulaires ont besoin de travailler ensemble.	Automobile	15400000	12
7	EASY FINANCE	Un meilleur accès au financement favorise la viabilité des PME	Services Financiers	14500000	13
5	ACCELMED	ACCELMED vise à capitaliser les résultats des projets MED-KED and MACC-BAM	Industrie	12577000	12
11	KnowInG	Intelligence et Innovation de la Connaissance pour un développement durable	Technologies	16500000	19
8	Flomed	FLORMED est un projet européen de développement de l'horticulture méditerranéenne	Informatique	15900000	12
12	MEDISS	Renforcer les capacités d'innovation de la filière méditerranéenne	Services Financiers	15690000	16
9	FREE-MED	Les territoires partenaires du projet présentent des situations diverses	Automobile	13600000	18

Figure 39 : Liste des projets

Chapitre 4. Implémentation et réalisation

C'était grosse au modo la partie front-end pour les utilisateurs de notre site web, passant maintenant brièvement sur la partie back-end de notre site web, on va donner quelques captures d'écran sur le code source des pages web utilisé sans oublier la base de données lors de la conception du site.

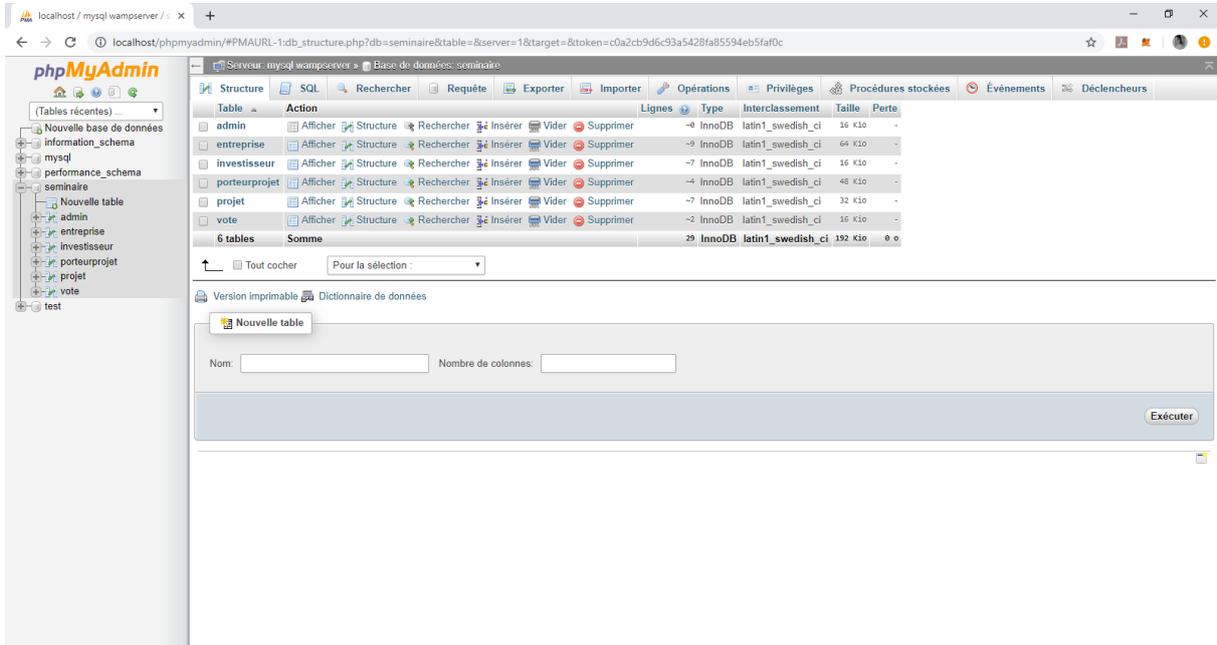


Figure 40 : Base de donnée du site web

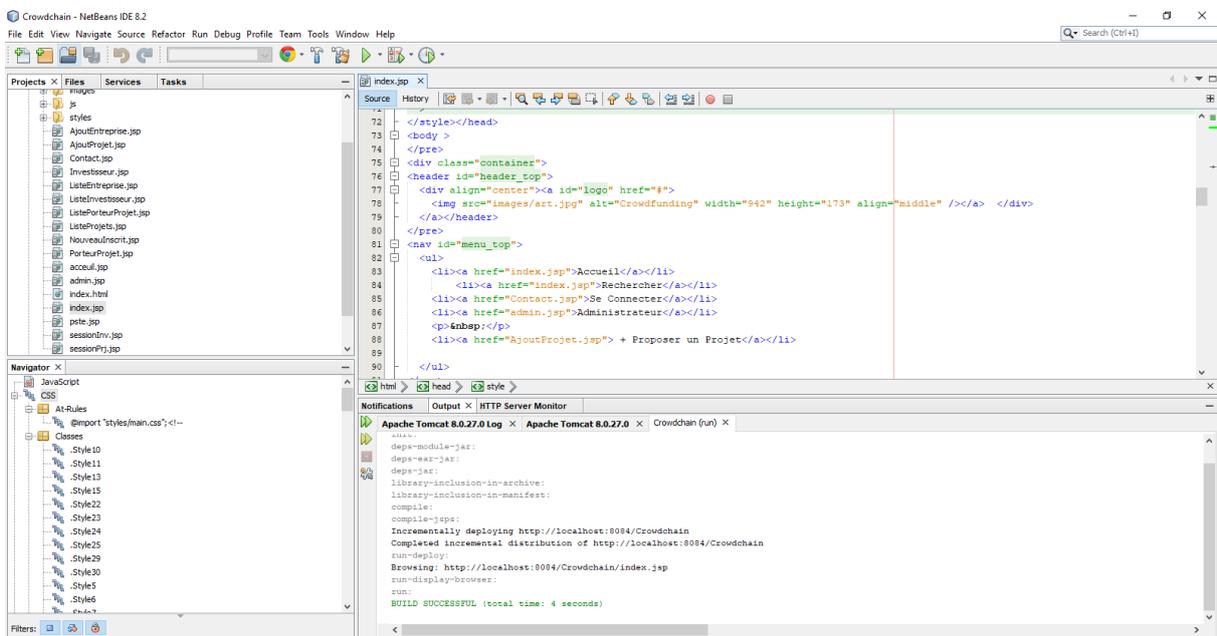


Figure 41 : code source de la page d'accueil

Chapitre 4. Implémentation et réalisation

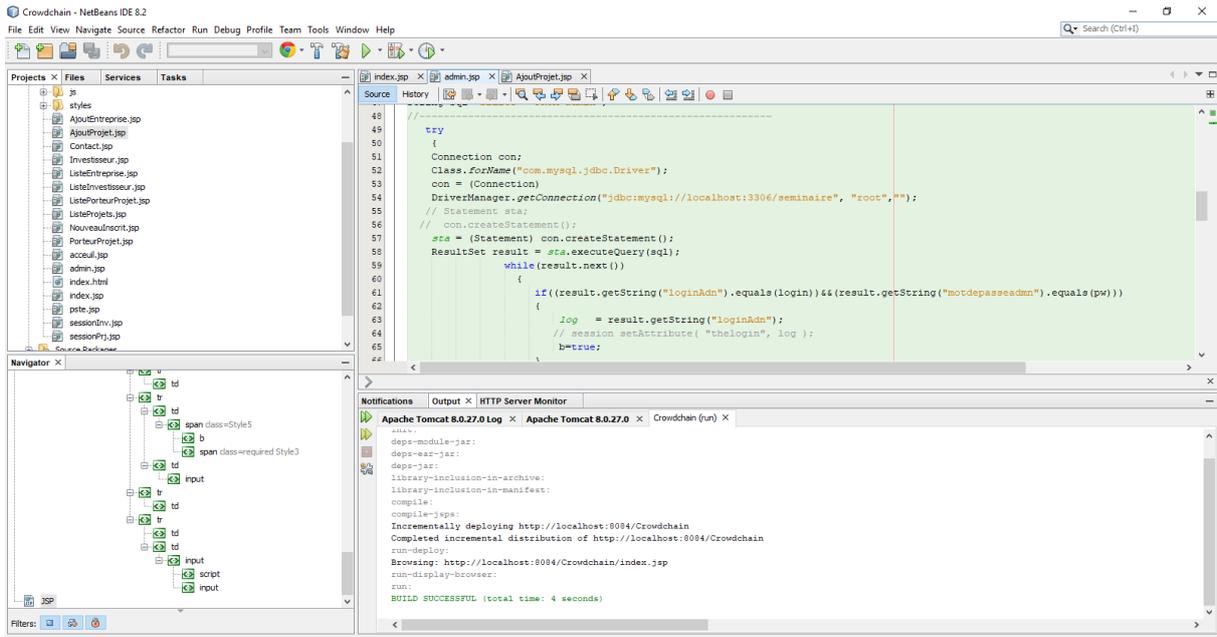


Figure 42 : code source de la session admin

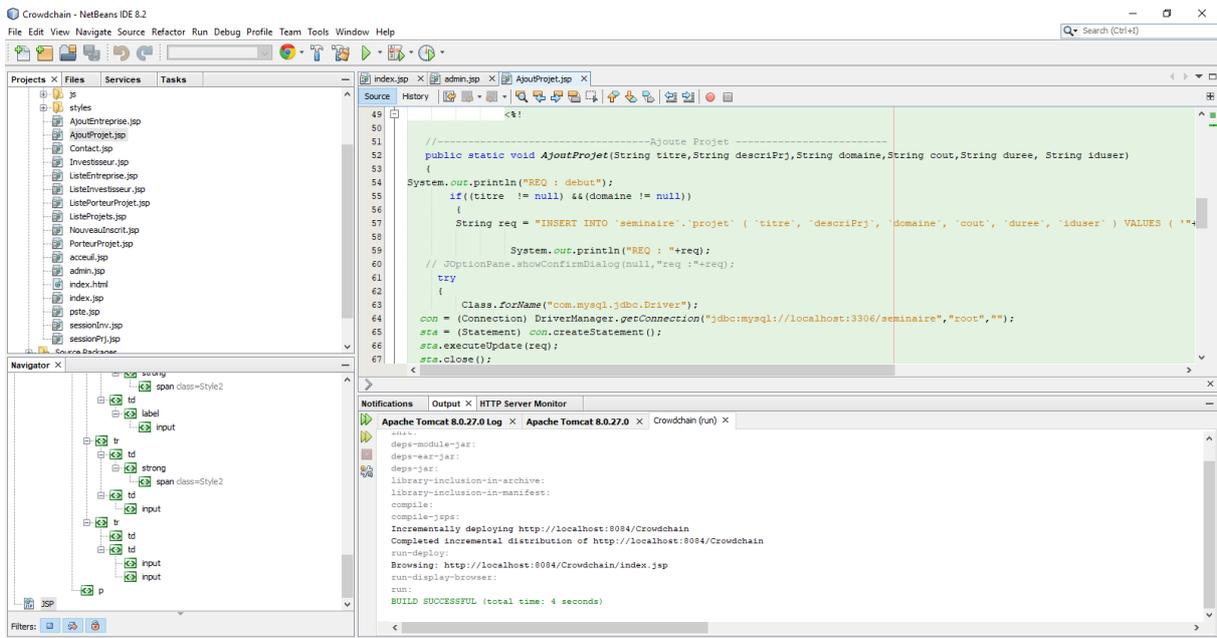


Figure 43 : code source de page ajouter projet

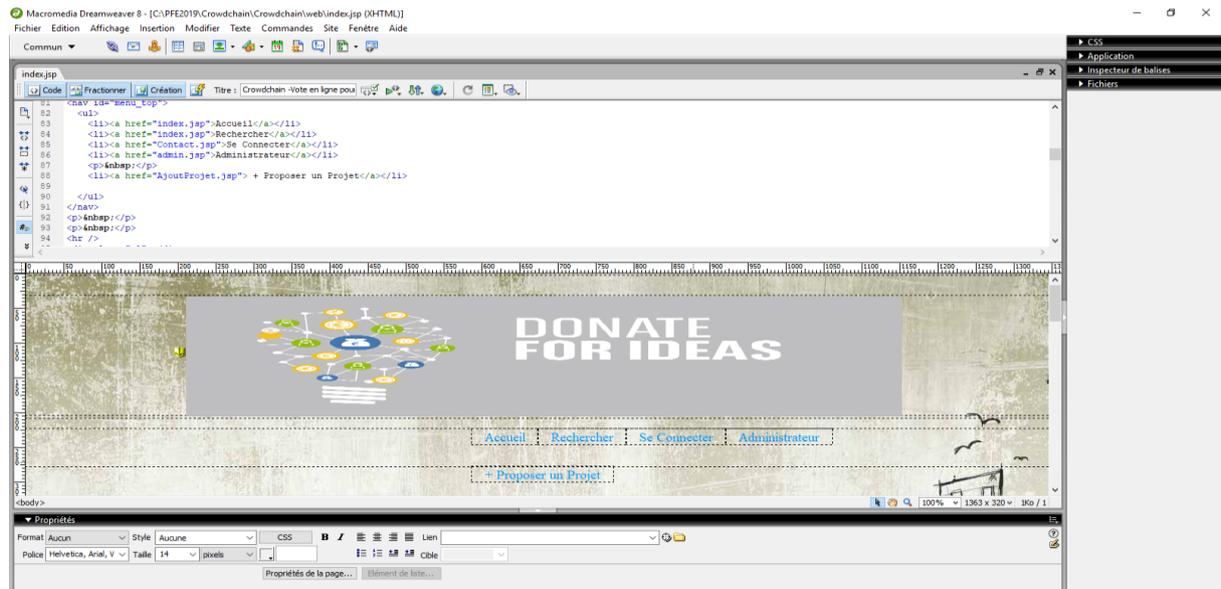


Figure 44 : Dreamweaver pour améliorer les interfaces des pages web

5.2 Configuration du réseau Blockchain :

- Ce que nous allons faire : Nous allons construire une application côté client qui communiquera avec notre contrat intelligent sur la blockchain. Cette application côté client aura un tableau des projets qui répertorie l'identifiant, le nom et le nombre de votes de chaque projet. Il y aura un formulaire où l'utilisateur pourra voter pour son projet souhaité. Il indique également le compte auquel nous sommes connectés à la blockchain sous "votre compte".

Afin de construire notre solution, nous avons d'abord besoin de quelques dépendances.

Installation de dépendances :

1- Node Package Manager (NPM) : La première dépendance dont nous avons besoin est Node Package Manager, ou NPM, fourni avec Node.js.

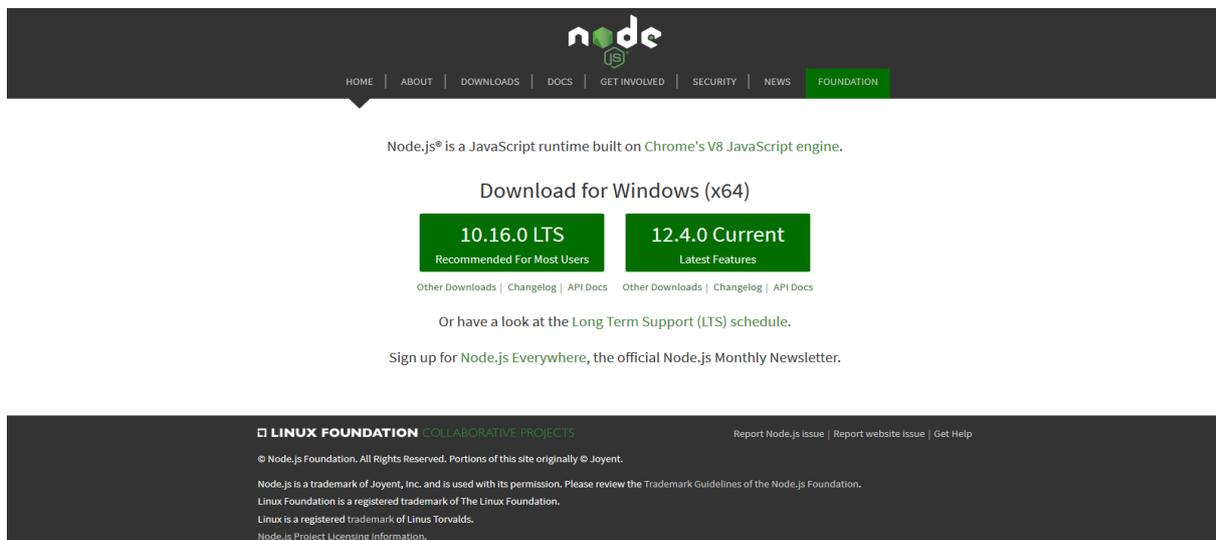


Figure 45 : Installation du Node Package Manager

Remarque : NPM est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, NPM fait partie de l'environnement et est donc automatiquement installé par défaut. NPM fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.js disponibles sur le dépôt NPM.

2- Truffle Framework : La deuxième dépendance est le Framework Truffle, ce qui nous permet de construire des applications décentralisées sur la blockchain Ethereum. Il fournit une suite d'outils nous permettant d'écrire des contrats intelligents avec le langage de programmation Solidity. Cela nous permet également de tester nos contrats intelligents et de les déployer dans la blockchain. Cela nous donne également un endroit pour développer notre application côté client.

Remarque : Solidity est un tout nouveau langage de programmation natif d'Ethereum, la deuxième plus grande crypto-monnaie par capitalisation boursière, publié initialement en 2015. Ethereum n'est pas seulement une crypto-monnaie capable de stocker de la valeur ou d'effectuer des paiements, mais également une plate-forme à part entière pour la création de ce qu'on appelle une puce. -Contrat-

À toutes fins pratiques, un contrat intelligent est une sorte d'entièrement programmable, un intermédiaire indépendant ou un juge équitable capable de gérer une transaction financière entre différentes parties et d'arbitrer de manière autonome un différend.

L'installation de ce Framework se fait avec NPM dans la ligne de commande comme ceci :

```
$ npm install -g truffle
```

3- Ganache : La dépendance suivante est Ganache, une blockchain locale en mémoire. Nous avons installé Ganache en le téléchargeant à partir du site Web de Truffle Framework. Il nous a donné 10 comptes externes avec des adresses sur notre blockchain Ethereum locale. Chaque compte est pré chargé avec 100 faux éther.

Remarque : Ganache est une blockchain personnelle pour le développement Ethereum que vous pouvez utiliser pour déployer des contrats, développer vos applications et exécuter des tests.

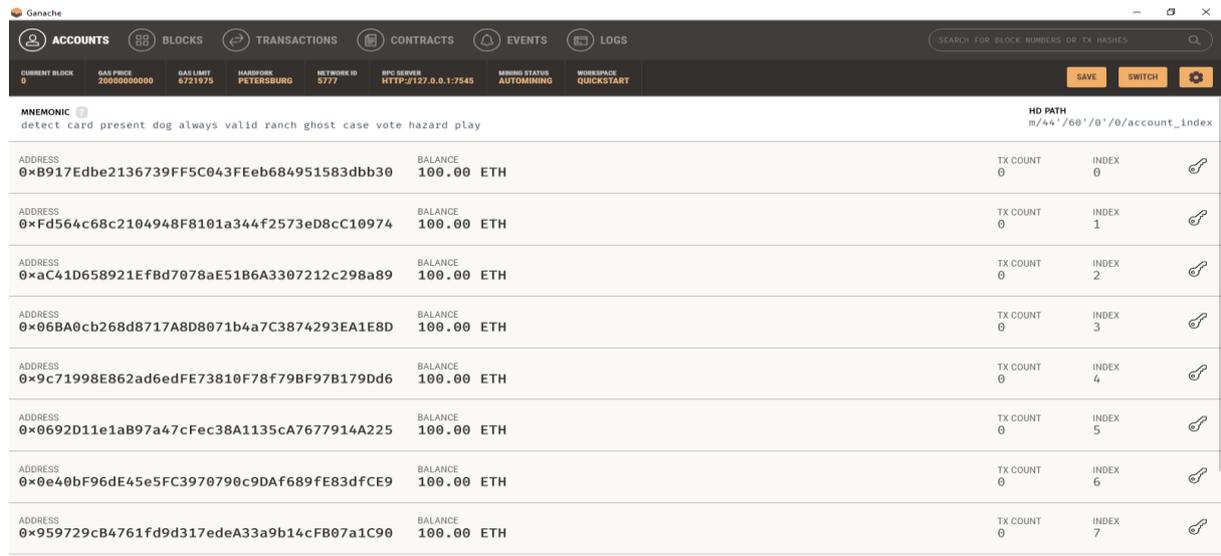


Figure 46 : Page d'accueil de Ganache

4- Metamask : La prochaine dépendance est l'extension Metamask pour Google Chrome. Pour utiliser la blockchain, nous devons nous y connecter. Nous devons installer une extension de navigateur spéciale pour pouvoir utiliser la chaîne de blocs Ethereum. C'est là que le méta masque entre en jeu. Nous pourrions nous connecter à notre blockchain Ethereum locale avec notre compte personnel et interagir avec notre contrat intelligent. Nous allons utiliser l'extension Metamask chrome.

Remarque : **Metamask** est une extension permettant d'accéder aux applications distribuées compatibles Ethereum ou aux "Dapps" de votre navigateur Chrome normal ! L'extension injecte l'API Ethereum web3 dans le contexte JavaScript de chaque site Web, afin que les dapps puissent lire à partir de la blockchain. MetaMask permet également à l'utilisateur de créer et de gérer ses propres identités

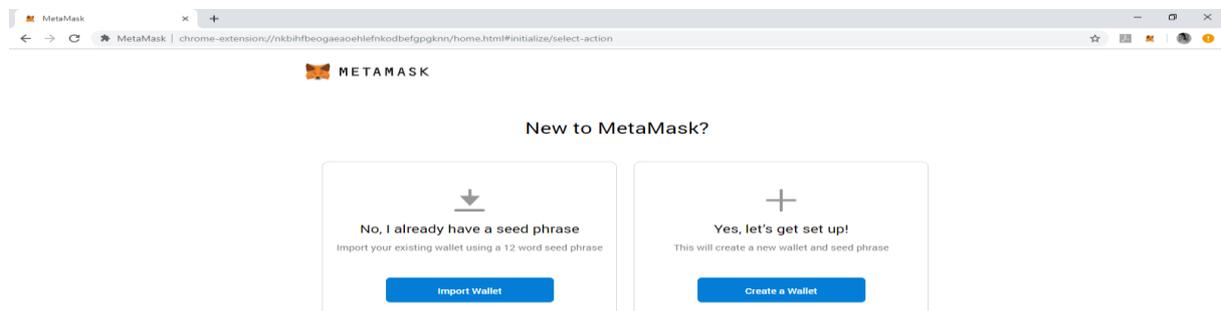


Figure 47 : Intégration d'extension Metamask

5- Syntax Highlighting : La dépendance est facultative, mais recommandée pour la coloration syntaxique du langage de programmation Solidity. La plupart des éditeurs de texte et des IDE ne disposent pas de la syntaxe surlignée pour Solidity prête à l'emploi, nous devons donc installer un package pour prendre en charge cette opération.

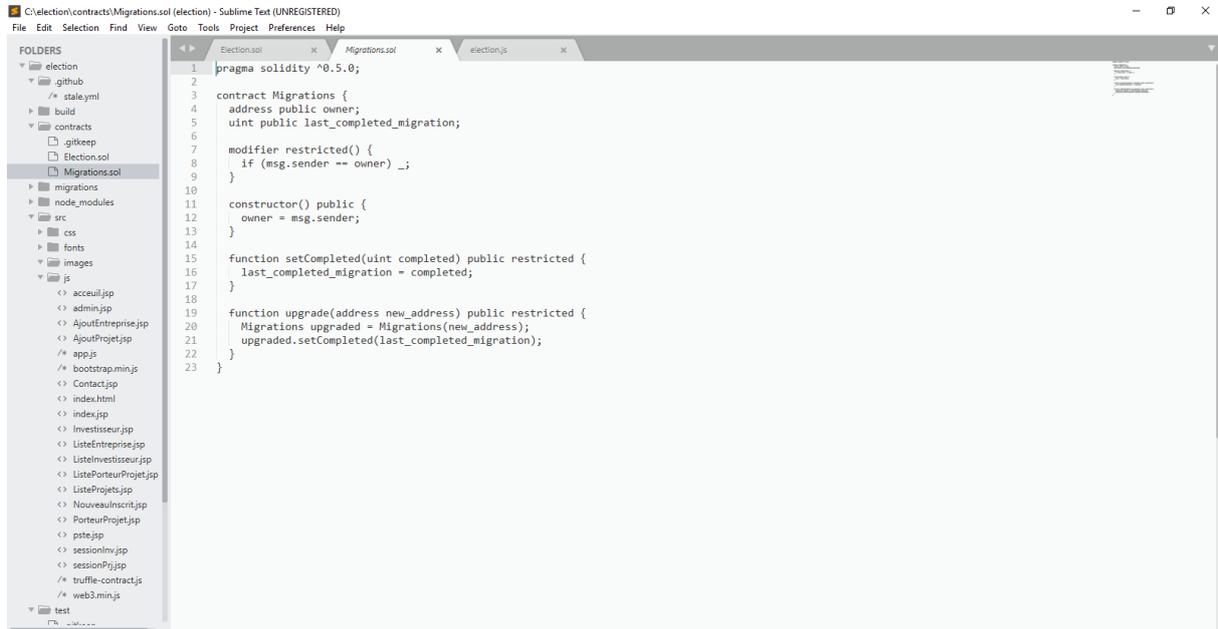


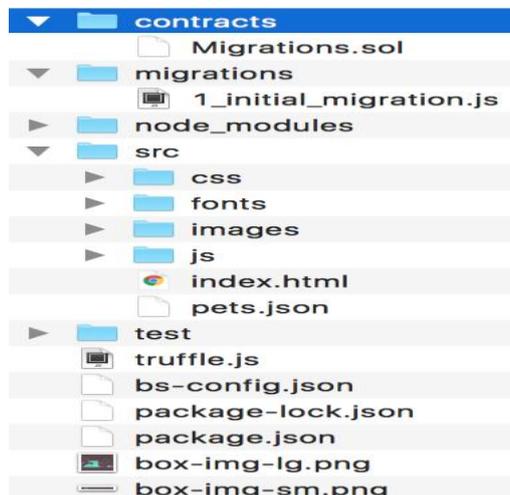
Figure 48 : Sublime Text 3

Après avoir déployer les dépendances, on va créer maintenant un répertoire de projet pour notre solution dans la ligne de commande comme ceci :

```
$ mkdir election
$ cd election
```

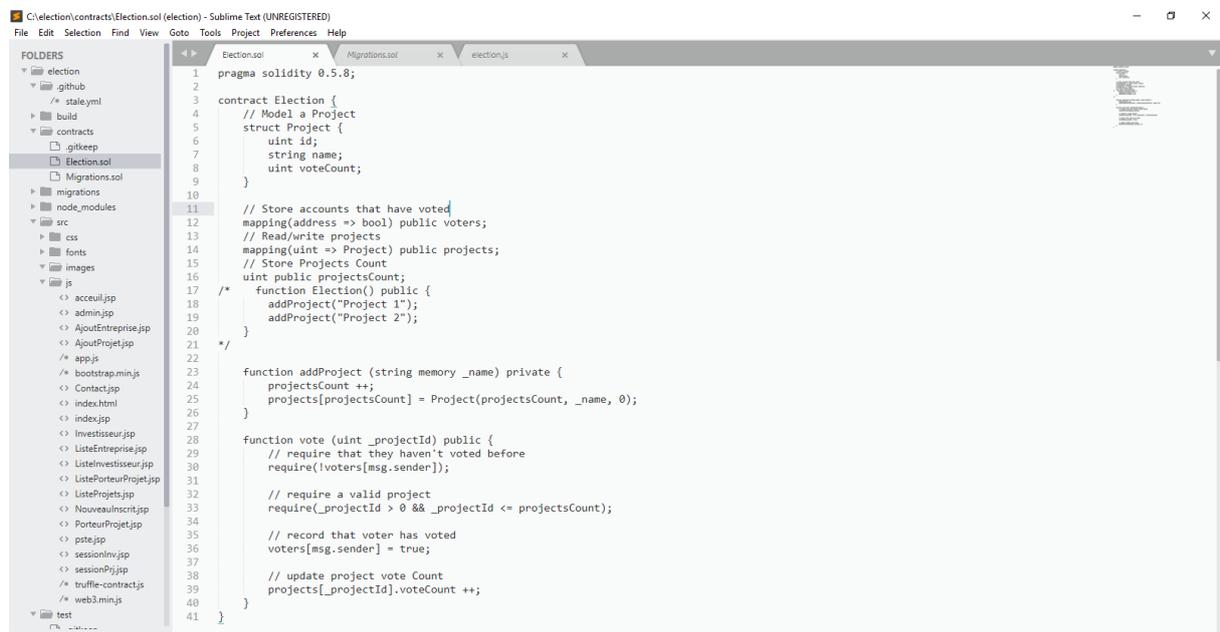
Maintenant que nous sommes dans notre projet, installons la boîte pour animaux de compagnie à partir de la ligne de commande, comme suit :

```
$ truffle unbox pet-shop
```



- **Contracts directory** : c'est ici que vivent tous les contrats intelligents. Nous avons déjà un contrat de migration qui gère nos migrations vers la blockchain.
- **Migrations directory** : C'est ici que résident tous les fichiers de migration. Ces migrations sont similaires aux autres infrastructures de développement Web qui nécessitent des migrations pour modifier l'état d'une base de données. Chaque fois que nous déployons des contrats intelligents sur la blockchain, nous mettons à jour l'état de la blockchain et nous avons donc besoin d'une migration.
- **Node_modules directory** : c'est la maison de toutes nos dépendances de nœuds.
- **Src directory** : C'est là que nous développerons notre application côté client.
- **Test directory** : C'est là que nous rédigerons nos tests pour nos contrats intelligents.
- **Truffle.js file** : c'est le fichier de configuration principal de notre projet Truffle

Le répertoire de projet Election va contenir essentiellement 4 dossier, le premier dossier intitulé « contracts » ce dossier va contenir notre fichier Election.Sol qui est notre contrat-intelligent, Ce contrat intelligent contiendra toute la logique de notre solution. Il sera en charge de la lecture et de l'écriture dans la blockchain Ethereum. Cela nous permettra d'énumérer les projets qui se présenteront aux élections et de garder une trace de tous les votes et investisseurs. Il va également régir toutes les règles de l'élection, comme imposer aux comptes de ne voter qu'une seule fois.



```
1 pragma solidity 0.5.8;
2
3 contract Election {
4     // Model a Project
5     struct Project {
6         uint id;
7         string name;
8         uint voteCount;
9     }
10
11     // Store accounts that have voted
12     mapping(address => bool) public voters;
13     // Read/write projects
14     mapping(uint => Project) public projects;
15     // Store Projects Count
16     uint public projectsCount;
17     /*
18     function Election() public {
19         addProject("Project 1");
20         addProject("Project 2");
21     }
22     */
23     function addProject (string memory _name) private {
24         projectsCount ++;
25         projects[projectsCount] = Project(projectsCount, _name, 0);
26     }
27
28     function vote (uint _projectId) public {
29         // require that they haven't voted before
30         require(!voters[msg.sender]);
31
32         // require a valid project
33         require(_projectId > 0 && _projectId <= projectsCount);
34
35         // record that voter has voted
36         voters[msg.sender] = true;
37
38         // update project vote Count
39         projects[_projectId].voteCount ++;
40     }
41 }
```

Figure 49 : Notre Smart-Contract

Donc, comme vous voyez dans la figure 49 notre smart-contract se base principalement sur la fonction Vote qui prend en paramètre l'identifiant du projet.

Cette fonction est conçue pour faire essentiellement 4 fonctionnalités lequel se repose notre solution pour avoir un vote valide afin de le déployer sur la blockchain locale, ces fonctionnalités sont :

```
// l'exigence que les investisseurs n'ont pas voté avant  
require(!voters[msg.sender]);  
// l'exigence d'un projet valide  
require(_projectId > 0 && _projectId <= projectsCount);  
// Compter le vote des voters  
voters[msg.sender] = true;  
// mettre à jour le nombre de votes du projet  
projects[_projectId].voteCount ++;
```

Après avoir créé et édité le smart-contract dans l'éditeur Sublime Text 3, on a migrer notre smart-contract vers la blockchain locale avec l'aide de la ligne de commande :

\$ truffle migrate

Une fois l'immigration terminé, on a déployé notre fichier Election. Sol dans la blockchain ce qui nous a donné le deuxième dossier « migrations » constitué de deux fichier : 1_initial_migration.js et 2_deploy_contracts.js comme le montre la figure 50.

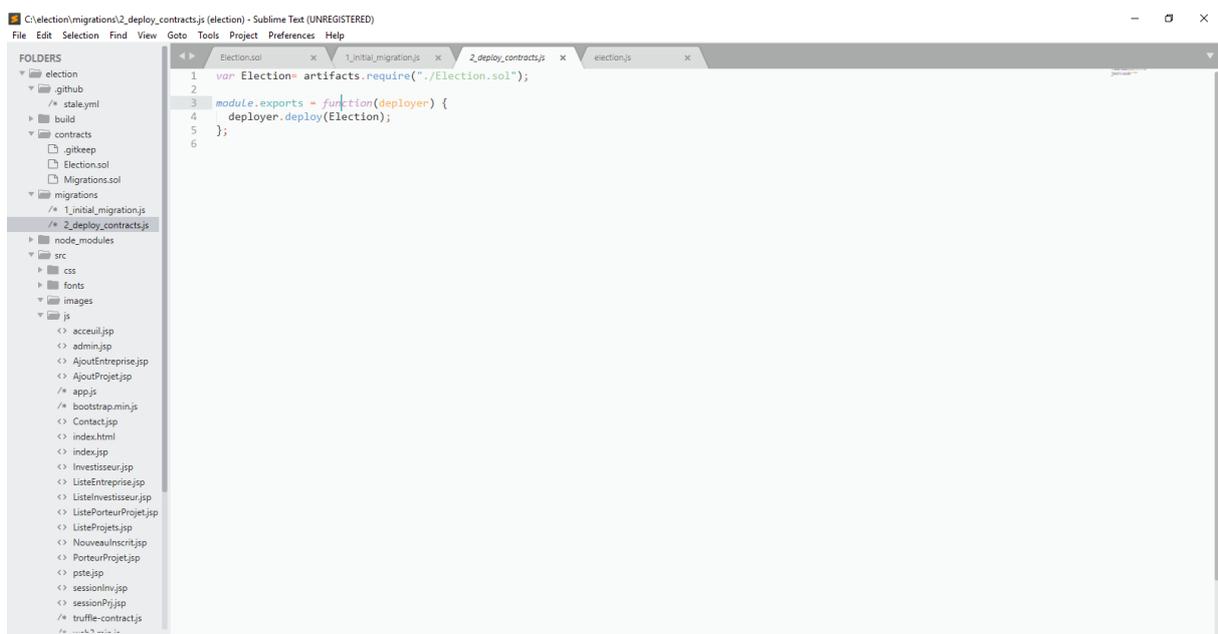
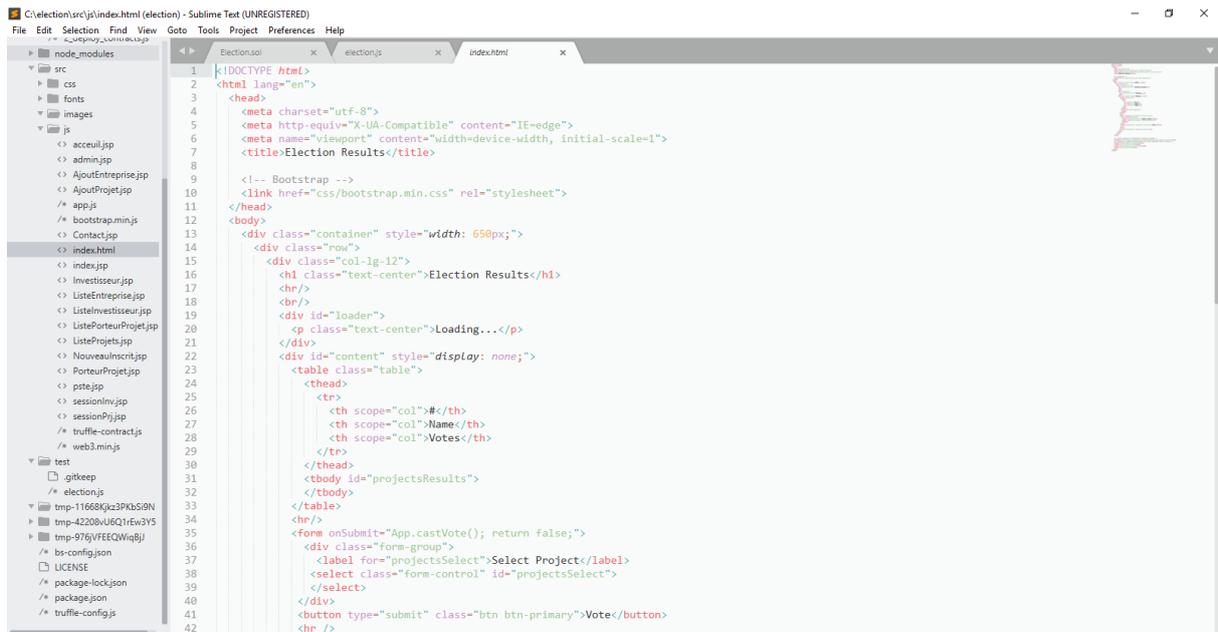


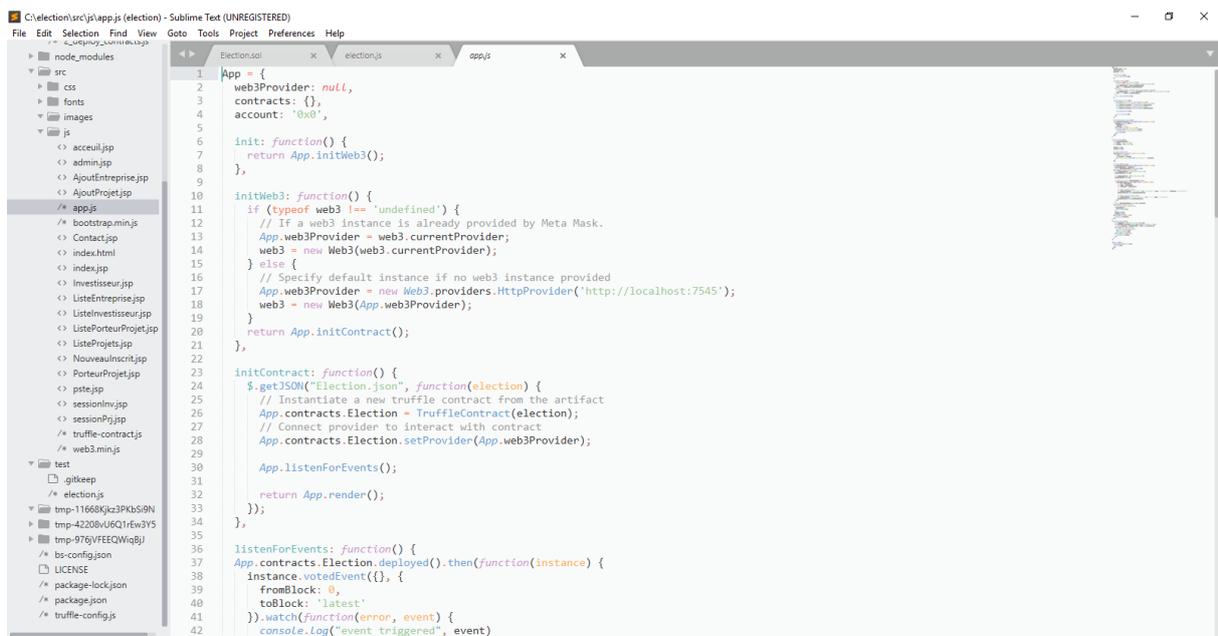
Figure 50 : Dossier migrations

Chapitre 4. Implémentation et réalisation



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <title>Election Results</title>
8
9 <!-- Bootstrap -->
10 <link href="css/bootstrap.min.css" rel="stylesheet">
11 </head>
12 <body>
13 <div class="container" style="width: 650px;">
14 <div class="row">
15 <div class="col-lg-12">
16 <h1 class="text-center">Election Results</h1>
17 <hr/>
18 <br/>
19 <div id="loader">
20 <p class="text-center">Loading...</p>
21 </div>
22 <div id="content" style="display: none;">
23 <table class="table">
24 <thead>
25 <tr>
26 <th scope="col">#</th>
27 <th scope="col">Name</th>
28 <th scope="col">Votes</th>
29 </tr>
30 </thead>
31 <tbody id="projectsResults">
32 </tbody>
33 </table>
34 <hr/>
35 <form onSubmit="App.castVote(); return false;">
36 <div class="form-group">
37 <label for="projectSelect">Select Project</label>
38 <select class="form-control" id="projectSelect">
39 </select>
40 </div>
41 <button type="submit" class="btn btn-primary">Vote</button>
42 <hr />
```

Figure 52 : Fichier index.html



```
1 App = {
2   web3Provider: null,
3   contracts: {},
4   account: '0x0',
5
6   init: function() {
7     return App.initWeb3();
8   },
9
10  initWeb3: function() {
11    if (typeof web3 !== 'undefined') {
12      // If a web3 instance is already provided by Meta Mask.
13      App.web3Provider = web3.currentProvider;
14      web3 = new Web3(web3.currentProvider);
15    } else {
16      // Specify default instance if no web3 instance provided
17      App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
18      web3 = new Web3(App.web3Provider);
19    }
20    return App.initContract();
21  },
22
23  initContract: function() {
24    $.getJSON("Election.json", function(election) {
25      // Instantiate a new truffle contract from the artifact
26      App.contracts.Election = TruffleContract(election);
27      // Connect provider to interact with contract
28      App.contracts.Election.setProvider(App.web3Provider);
29      App.listenForEvents();
30    });
31    return App.render();
32  });
33  },
34
35  listenForEvents: function() {
36    App.contracts.Election.deployed().then(function(instance) {
37      instance.votedEvent({}, {
38        fromBlock: 0,
39        toBlock: 'latest'
40      }).watch(function(error, event) {
41        console.log("event triggered", event)
42      });
43    });
44  }
45 }
```

Figure 53 : Fichier app.js

Lorsqu' on lance l'exécution sur le serveur de développement avec la commande : `$ npm run dev`, on obtient l'affichage suivant :

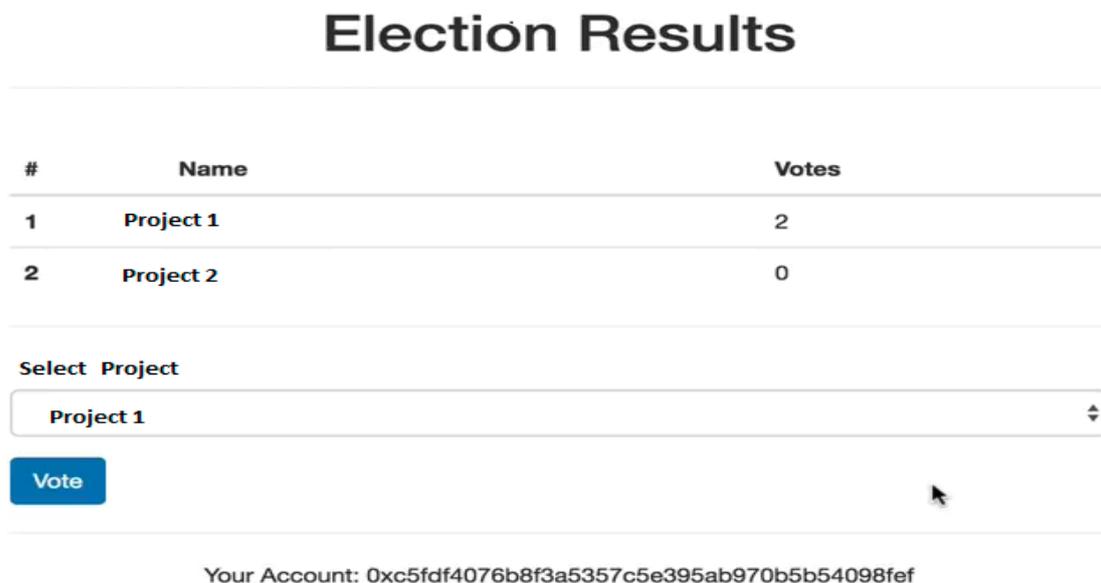


Figure 54 : Affichage de la partie client pour les investisseurs

Comme vous voyez sur la figure 54, la page se compose d'un identifiant de projet et son nom et le nombre de vote sur lui ainsi qu'une barre de sélection de projet et un bouton de vote, lorsqu'on clique sur ce dernier bouton on va voir une fenêtre de confirmation de méta-masque comme se suit dans la figure 55

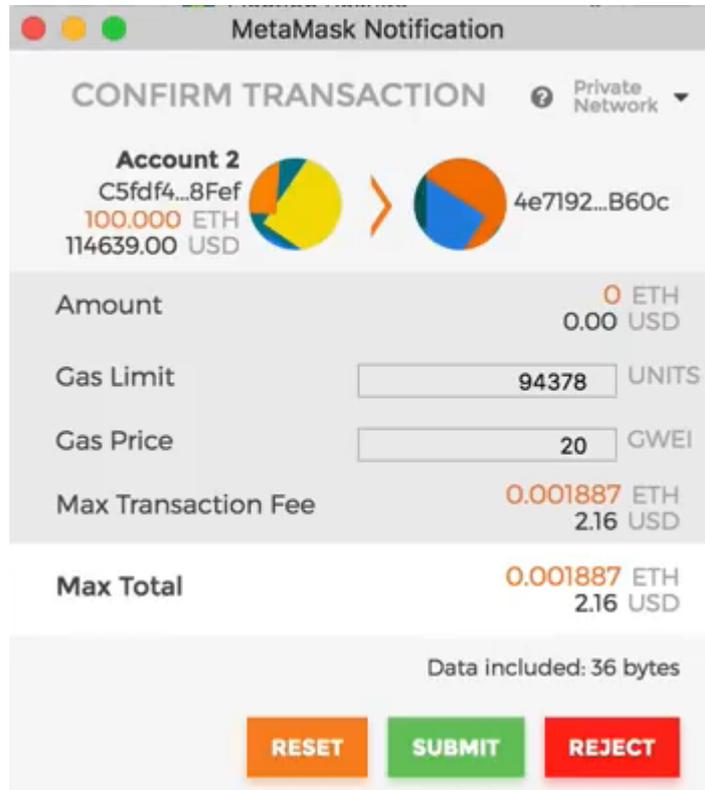


Figure 55 : Meta-mask pop-up

Une fois on clique sur le bouton Submit, le vote sera ajouté est déployés sur la blockchain locale.

6. Conclusion

Nous avons consacré ce dernier chapitre à présenter l'environnement matériel et logiciel du développement de notre application et à la partie réalisation à travers un ensemble d'interfaces.

Conclusion Générale

Ce projet de fin d'études consiste à concevoir et implémenter un système permettant à ses utilisateurs d'effectuer un vote en ligne en se basant sur la technologie Blockchain.

Pour s'y faire, nous avons suivie 3 étapes durant la réalisation de notre projet, la configuration d'un réseau Blockchain, le développement d'un smart contract présentant les différentes opérations qui peuvent s'exécuter sur la Blockchain et enfin le développement d'un site web qui communique avec la Blockchain à travers le Framework Ethereum.

Pour atteindre les objectifs initialement fixés, nous avons commencé par l'étude préalable qui nous à permettre de comprendre les principaux concepts autour desquels tourne notre projet ainsi que les différents choix technologiques adoptées. Puis, nous avons fait une étude des besoins afin de recouvrir les exigences détaillées. Ensuite, nous avons défini l'architecture utilisée, pour aboutir par la suite à la conception, qui présente l'aspect dynamique et statique du système. Enfin, nous avons traité l'étape de réalisation de notre application.

Ce travail nous a été très enrichissant de point de vue des thèmes abordés et technologies utilisées. Il nous a bénéficié d'une opportunité pour aborder la technologie Blockchain, qui est une technologie très intéressante et innovante et qui nous a fait découvrir les techniques de sécurisation avancées des systèmes informatique. Nous avons pu aussi découvrir le métier de la finance et toucher de près le processus d'exécution des transactions financières et les acteurs intervenant dans ces transactions dans le monde réel.

Enfin, ce sujet nous a permis de renforcer nos connaissances dans le développement Java/JEE ainsi que de découvrir des Framework facilitant la tâche de configuration d'un projet pour pouvoir se concentrer plus dans le métier de l'application.

Bibliographie

[1] Le crowdfunding. Les clés du financement participatif -Par Rony Germon et Adnane Maalaoui- Edition : Paris : Ellipses, 2014

[2] <http://gensol.arte.tv/blog/2013/01/10/plateformes-dechanges-citoyens-et-sites-de-crowdfunding/>

[3] <http://www.lefigaro.fr/secteur/high-tech/2015/05/12/01007-20150512ARTFIG00291-la-plateforme-de-crowdfunding-kickstarter-debarque-en-france.php>

[4] <https://www.smartangels.fr/crowdfunding/equity>

[5] Le Crowdfunding : bilan et perspectives -Michaël Riga-

[6] <https://www.abe-infoservice.fr/epargne/financement-participatif/le-financement-participatif-lessentiel>

[7] Voir www.youthevote.net.

[8] Voir notamment Dictson D. et Ray D., de la société SecurePoll.com, « The Modern Democratic Revolution : An Objective Survey of Internet-Based Elections », janvier 2000. Disponible sur leur site. Voir aussi Adler J., « On line Voting Security », www.VoteHere.net, 2000.

[9] C. J. JARRY. (Avril. 2016). Qu'est-ce que la blockchain ? [Accès le 5-Mars-2019], adresse : <http://www.frenchweb.fr/quest-ce-que-la-blockchain/237579>.

[10] G. DREAN. (Octobre. 2016). La « blockchain » pour les nuls. [Accès le 12-Avril-2019], adresse : <https://www.contrepoints.org/2016/10/02/267509-la-blockchain-pour-les-nuls>.

[11] G. DREANNE. (Septembre. 2015). Qui gère le bitcoin ? [Accès le 20-Mai-2019], adresse : <https://www.contrepoints.org/2015/09/16/221958-qui-gere-le-bitcoin>.

[12] S. Ravens. (Mars. 2017). Understanding peer-to-peer, blockchain, and transactive energy.[Accès le 8-Mai-2019], adresse : <https://www.navigantresearch.com/blog/understandingpeer-to-peer-blockchain-and-transactive-energy>.

[13] A. DECAZES. (Janvier. 2017). Les 'smart contract' dans la blockchain : quézako ? [Accès le 19-Mai-2019], adresse : <http://www.usine-digitale.fr/article/les-smart-contractdans-la-blockchain-quezako.N494204>.

[14] R. THOOMU. (Septmeber. 2016). Hyperledger protocol specification. [Accès le 15-Avril-2019], Communauté Linux, adresse : <http://hyperledger-fabric.readthedocs.io/en/v0.6/>.

- [15]** S. PEYROTT. (Mars. 2017). [Accès le 10-Juin-2019], adresse <https://auth0.com/blog/anintroduction-to-ethereum-and-smart-contracts-part-2/>.
- [16]** A. FREDOUELLE. (Septembre. 2017). Ripple, la blockchain qui secoue la finance mondiale. [Accès le 13-Avril-2019], adresse : <http://www.Journaldunet.Com/economie/finance/1184734-ripple/>.
- [17]** J. LUC. (2016). Bitcoin c'est quoi ? [Accès le 11-Mars-2019], adresse : <https://bitcoin.fr/Quest-ce-que-Bitcoin/>.
- [18]** <https://blockchainfrance.net/2016/03/04/comprendre-ethereum/>
- [19]** G. UPADHYAYA. (Janvier. 2017). Hyperledger vs-ethereum-vs-ripple-vs-bitcoin. [Accès le 1-Mars-2019], adresse : <https://goupadhyblog.wordpress.com/2017/01/16/hyperledgervs-ethereum-vs-ripple-vs-bitcoin/>.
- [20]** <https://dev.to/prnthh/muchain---designing-a-voting-system-for-1-billion-on-the-blockchain---part-1-4ab1>
- [21]** <http://inf1410.telug.ca/telugDownload.php?file=2014/01/INF1410-PresentationStarUML.pdf>
- [22]** <https://fr.netbeans.org/> [Accès le 22-avril-2019]
- [23]** G. community. (2015). [Accès le 16-Mai-2019], adresse : <https://github.com/apache/tomcat>.
- [24]** C. DUVALLET. (2011). Oracle 11g installation et administration. [Accès le 18-Février-2019], adresse : <http://litis.univlehavre.fr/~duvallet/enseignements/Cours/Oracle11g/COURS-ORACLE-4p.pdf>.
- [25]** J. PARDANAUD. (Avril. 2017). Dynamisez vos sites web avec javascript ! [Accès le 3-Mai-2019], adresse : <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript>.