REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre :RTIC15/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Réseaux et Technologies de l'Information et de la Communication (RTIC)**

Blockchain for the driver's license management

Par :

LEHRAKI DOUAA

Soutenu le . ./. ./…. devant le jury composé de :

	grade	Président
	grade	Rapporteur
	grade	Examinateur

Année universitaire 2020-2021

Abstract

Blockchain is a new technology that allows transactions to be carried out securely and transparently and without interference from third parties.

Its structure is decentralized, which means that the data is distributed among the users of the network, so that the information can never be erased. These features make blockchain technology suitable for managing driving licenses and combating the presence of unlicensed drivers.

In order to ensure that only qualified people who have received proper driver training and have a genuine driver's license are allowed to drive vehicles, we will introduce a decentralized system based on blockchain technology to solve this issue.

key-words : Blockchain, Driver's license , Smart contract, blockchain based license management.

Résumé

Blockchain est une nouvelle technologie qui permet d'effectuer des transactions en toute sécurité et transparence et sans interférence de tiers.

Sa structure est décentralisée, ce qui signifie que les données sont réparties entre les utilisateurs du réseau, de sorte que les informations ne peuvent jamais être effacées. Ces fonctionnalités rendent la technologie blockchain adaptée à la gestion des permis de conduire et à la lutte contre la présence de conducteurs sans licence.

Afin de garantir que seules les personnes qualifiées ayant reçu une formation de conduite appropriée et disposant d'un véritable permis de conduire soient autorisées à conduire des véhicules, nous introduirons un système décentralisé basé sur la technologie blockchain pour résoudre ce problème.

Mots-clés : Blockchain, Permis de conduire, Smart contract, gestion des licences basée sur la blockchain .

DEDICATION

Praise to **ALLAH** the Lord of the worlds, may his blessings and peace be upon the most honored of messengers **(Muhammad)** and upon all his family and companions.

I dedicate my dissertation and the journey that lasted five years for those whom supported me through out my life,my beloved father **Larbi** and the sunshine of my life, my dear mother **Fattoum**.

Also, I dedicate my dissertation to the countries of martyrs, especially **Algeria** and **Palestine** and to my sister's pure soul **Aya**, may God have mercy on her.

Finally,i dedicate it also to my sisters, my brothers, my nephews and my niece may allah protect them, my dear family, especially my little aunt, my relatives, friends ,someone dear to my heart and everyone who made me happy and supported me one day.

ACKNOWLEDGEMENTS

First of all, I thank **ALLAH** God for giving me the courage and patience to complete this humble work.

After that, I would like to thank my supervisor who was more than generous with his expertise and precious time **Dr.Guerrouf Fayçal** also for his patience throughout the entire process.

I would like to express my deepest gratitude and sincere thanks to the members of the jury who will agree to read and evaluate this work.

Finally, I would like to thank all the teachers of our department for teaching me and enlightening my way during these 5 years also all those who have helped and supported me from near or far.

Contents

List of Figures

List of Tables

General Introduction

Among the issues facing transportation systems is road safety which has become a matter of national concern given the terrible rise in road traffic accidents in recent years.

There are numerous factors account for these accidents, like human error, speed,no application of the driving code, incompetent drivers getting licenses through unfair means or carrying a forged license .and given the seriousness of the situation our work will focus on one of the most important parts, namely the driver's license wich is an essential part in transportation for any driver.

In view of this, we are led to ask how we can secure the process of obtaining driver's licenses and to fight the presence of unlicensed drivers ?. This in order to preserving human lives and property.
To achieve this, we will propose a decentralized system based on blockchain technology to manage the driver's license starting from its creation. Where we find that the most important reasons make us use Blockchain are:

The high security that it provide , it is unlikely that Blockchain being the target for tring to hack because it is distributed, so no single point to target it, which make the information more secure and these systems can achieve transparency in evaluation , prevent illegal transactions around information and deliberate manipulation by corrupt authorities.

Our project is organized in parts: concerns the state of the art of our subject and presents the realization of a system for driver's license management using Blockchain .

- The introduction where we are going to begin our thesis with associate introduction to the context of this work, the targeted problem, and also the solution that we propose and why we choise it.

- The first chapter presents the general and important concepts of Blockchain technology, including history,components, types and features of Blockchain . Next, we will indicate how the Blockchain working with its advantages and disadvantages. Finally, we will present the most implementations of Blockchain.
- The second chapter is devoted to describe drive's license and her process of creation as well as problems that it is facing . The chapter ends with Traffic offense penalties.
- The third chapter offers the design of the proposed system and therefore presents the global and detailed architecture and presents implementation of our system.
- Finally, we end our thesis with a general conclusion.

Chapter 1

Blockchain Technology

1.1 Introduction

Blockchain has become very popular in recent years, as it is the latest wave of technical digitization, it has promised to positively alter the existing paradigms of all industries. Blockchain transmits information in a reliable and secure manner in a decentralized system that is difficult to penetrate. The most famous examples for this technology are Bitcoin and Ethereum.

In this chapter, we will cover the basic concepts of this technology.

1.2 History of Blockchain

Blockchain is one of the most important inventions which emerged in the late 1980s and early 1990s [23].

In 1991, a signed chain of information was used for digitally signing documents whereby facilely show no one could varied the signed documents [23].

In 2008 , the person(s) who's working under the pseudonym Satoshi Nakamoto conceptualized the first Blockchain, and described in the paper Bitcoin: A Peer to Peer Electronic Cash System.

In 2009, the Bitcoin cryptocurrency blockchain network was launched and was the first blockchain application [23].

In 2013, Ethereum introduced blockchain in the form of "a decentralized platform that runs smart contracts" wich is the first expansions outside of currency and her officially launched in 2015 [18].

1.3 Definition of Blockchain Technology

There are many definitions of blockchain technology, and they all revolve around being a secure, distributed technology that is implemented without the need for a third party. To get to know this technology more, we can do so through the following definitions:

Blockchain is a distributed digital ledger, in which the storage of all database entries on one computer is replaced with their storage on multiple computers. The stored data is a block, each block related to the previous block [11].

Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous section after validation and a unanimous decision. With the addition of new blocks, modifying old blocks becomes more difficult. New blocks are replicated via ledger copy within the network, and any conflicts are resolved automatically using established rules [23].

Technically a blockchain is defined as: It is a peer-to-peer, distributed ledger that is cryptographically secure, append only, immutable, and updateable only via consensus or agreement among peers [7].

Blockchain is a peer-to-peer,and cryptography is used in the distributed ledger to provide security services that make the ledger safe against tampering and misuse, and data can only be added to the blockchain in a sequential order that is immutable and updateable only by consensus or peer agreement. This is what gives it the power of decentralization [7].

The concept of peer-to-peer, which was previously mentioned, means that all participants talk to each other directly without a third-party involvement [7].

1.4 Components of blockchain

The main components of a blockchain are as follow[23] :

1.4.1 Blocks

The core component of blockchain is block (Fig 1.1)which is a collection of transactions that are stored digitally such as a record of bank transfers or a registry of real estate titles...

A block can be also described as a container that hold different types of information which can be summarized as follow:

1. Block header:

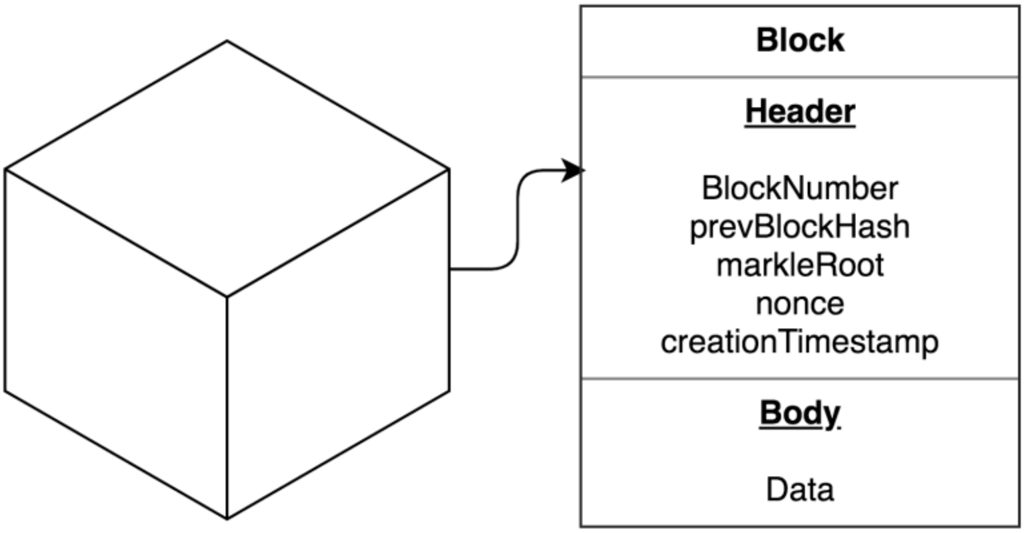


Figure 1.1: Block figure

A block header is the very important portion of a block that contains metadata about the block itself, These metadata are as follows :

(a) The hash of the previous block's header :is contained in the hash of the new block. Without this component, there would be no connection and chronology between each block, its size is 32 bytes.

(b) Timestamp :is the time of block creation and the time of transaction records, its size is 4 bytes.

(c) The nonce value :is the variable needed for consensus process. For blockchain networks which utilize mining, this is a number which is manipulated to solve the hash puzzle ,its size 4 bytes.

(d) Hash of block data(markle root) :is hash of the actual block that different methods can be used to accomplish it, its size is 32 bytes.The name merkle root derives from the fact that Bitcoin uses the Merkle tree to generate the hash from the data in the block.

2. Block data:

   A block data is the other portion of a block that contains a list of validated transactions and have been submitted to the blockchain network.

Validity and authenticity is guaranteed by checking that the transaction is correctly formatted and that the providers of digital assets in each transaction have each cryptographically signed the transaction.

There is a very important and special block in the blockchain called genesis block, let's describe it:

The genesis block also known as Block 0. It is the first block in the blockchain and the basis on which other blocks are added to form a chain of blocks.In technical terms, it means that the Genesis Block has it's "previous hash" value set to 0. Every node in the network can identify the genesis block's hash and structure, the fixed time of creation, and the single transactions within.

1.4.2 Transactions

A transaction is the fundamental unit of a blockchain which represents a transfer of the cryptocurrency between blockchain network users or could be a way of recording activities occurring on digital or physical assets where Transactions between network users are grouped together in a data structure called a block.To add this block to the Blockchain, it must be first validated by specific nodes of the network called miners where their role is to answer a complex crypto mathematical puzzle. Each answer found is specific to only one block which preventing its reuse for the validation of a new block. Once the block is validated by them and approved by the other nodes, it is time-stamped and added to the top of the block chain and all nodes in the network add it to their copies of the chain[9].

1.4.3 Chaining Blocks

Blocks are chained together through each block containing the hash digest of the previous block's header. If a previously published block were changed, it would have a different hash. This in turn would cause all subsequent blocks to also have different hashes since they include the hash of the previous block.figure 1.2

1.4.4 Consensus

Consensus refers to the mechanism of guarantee that a transaction is not fraudulent and is valid through an optional process known as mining, and for achieve consensus, different consensus algorithms are used where it's governed by the type of blockchain in use. let's describe some of these algorithms below:

1. Proof of work(PoW): is the first consensus algorithm created and used by cryptocurrencies where a publishing node wins the right to publish the next block by being the first to solve a computationally intensive puzzle .The puzzle is

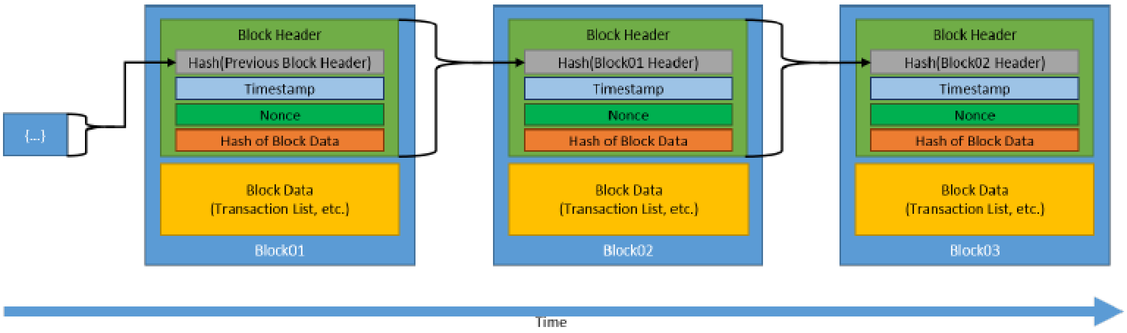


Figure 1.2: Generic chain of blocks

designed such that his solving is difficult but easy to verify .This enables all other full nodes to easily validate any proposed next blocks. Figure 1.3 shown process of proof of work.

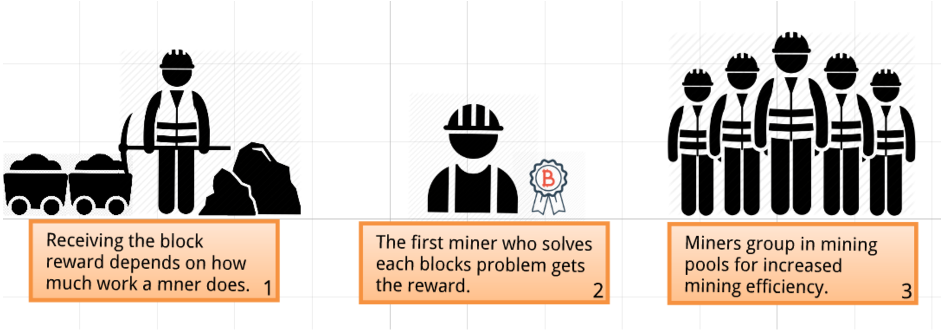

Receiving the block reward depends on how much work a mner does. 1
The first miner who solves each blocks problem gets the reward. 2
Miners group in mining pools for increased mining efficiency. 3

Figure 1.3: Proof of work

2. Proof of Stake (PoS): Was developed in 2011 as an alternative to PoW, as there are some differences and basic characteristics between them, especially while examining new blocks . it is based on the idea that the more stakea user has invested into the system, the more likely they will want the system to succeed, and the less likely they will want to subvert it where the blockchain network is secured by users locking an amount of cryptocurrency into the blockchain network, a process called staking. Figure 1.4 shown process of Proof of Stake.

3. Round robin:is consensus model for permissioned blockchain networks where nodes are pseudo-randomly selected to create blocks, but a node must wait several block-creation cycles before being chosen again to add another new block.

To learn more about these algorithms ,we will be presented them by comparing them in 1.5 and 1.6 figures:

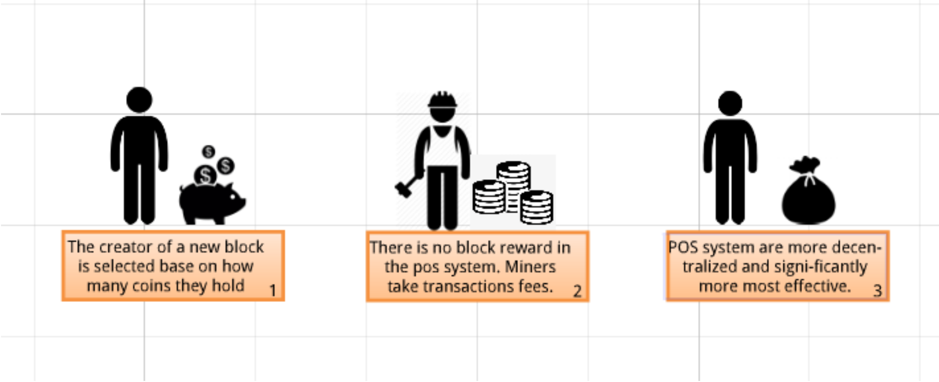

The creator of a new block is selected base on how many coins they hold 1
There is no block reward in the pos system. Miners take transactions fees. 2
POS system are more decen-tralized and signi-ficantly more most effective. 3

Figure 1.4: Proof of Stake

1.5 Functioning of Blockchain

Blockchain protocol is based on a peer-to-peer architecture, each participant constituting a node in the network. These participants store an identical copy of the ledger and then work together in the process of validating and certifying digital transactions, adding new transactions to the ledger.

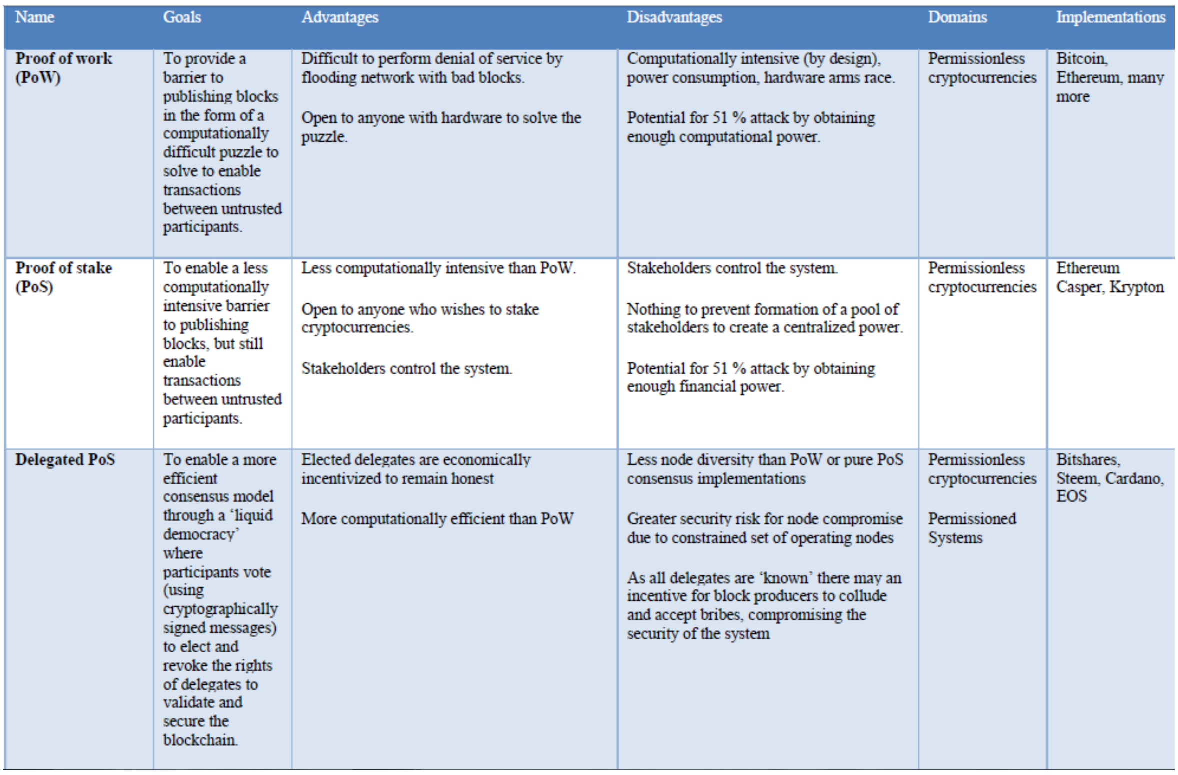
The process of adding transactions involves evaluating the proposed transaction and submitting it to a vote. If the majority of participants believe that the transaction is valid, it is added to the ledger, which links it to the previous transaction. Each transaction that goes through the binding process is grouped into a block, which further contains a cryptographic hash from the previous block, and then added linearly to the ledger in chronological order. Changes to the ledger are replicated across the network, so each participant has a complete copy of the updated ledger. It means that no one participant has the ability to easily attack the entire distributed network[14]. This can be represented with the scheme 1.7.

1.6 Cryptography in Blockchain

Cryptography is a method of developing techniques and protocols to prevent a third party from accessing and gaining knowledge of the data from the private messages during a communication process. In blockchain, cryptography is used for the following two purposes:

1. Securing the identity of the sender of transactions
2. Ensuring that past records cannot be tampered

To understand cryptography in blockchain, we have to understand the types of cryptography.



Name	Goals	Advantages	Disadvantages	Domains	Implementations
Proof of work (PoW)	To provide a barrier to publishing blocks in the form of a computationally difficult puzzle to solve to enable transactions between untrusted participants.	Difficult to perform denial of service by flooding network with bad blocks. Open to anyone with hardware to solve the puzzle.	Computationally intensive (by design), power consumption, hardware arms race. Potential for 51 % attack by obtaining enough computational power.	Permissionless cryptocurrencies	Bitcoin, Ethereum, many more
Proof of stake (PoS)	To enable a less computationally intensive barrier to publishing blocks, but still enable transactions between untrusted participants.	Less computationally intensive than PoW. Open to anyone who wishes to stake cryptocurrencies. Stakeholders control the system.	Stakeholders control the system. Nothing to prevent formation of a pool of stakeholders to create a centralized power. Potential for 51 % attack by obtaining enough financial power.	Permissionless cryptocurrencies	Ethereum Casper, Krypton
Delegated PoS	To enable a more efficient consensus model through a 'liquid democracy' where participants vote (using cryptographically signed messages) to elect and revoke the rights of delegates to validate and secure the blockchain.	Elected delegates are economically incentivized to remain honest More computationally efficient than PoW	Less node diversity than PoW or pure PoS consensus implementations Greater security risk for node compromise due to constrained set of operating nodes As all delegates are 'known' there may an incentive for block producers to collude and accept bribes, compromising the security of the system	Permissionless cryptocurrencies Permissioned Systems	Bitshares, Steem, Cardano, EOS

Figure 1.5: Consensus Comparison Matrix 1

1.6.1 Type of Cryptography

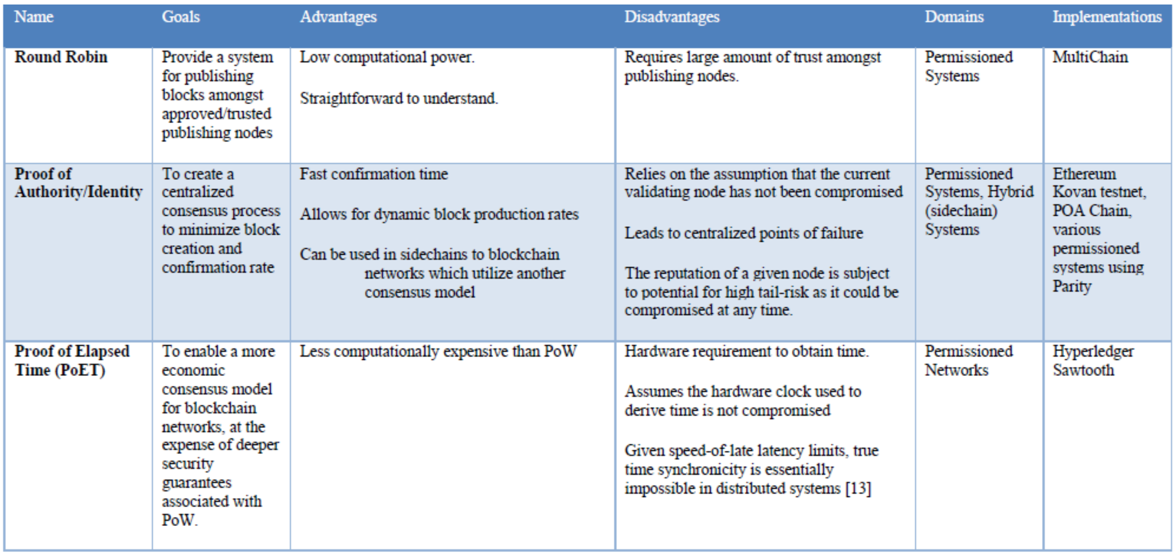
There are mainly three different ways to perform cryptographic algorithms, namely, symmetric-key cryptography, asymmetric key cryptography, and hash functions [20].

1.6.1.1 Symmetric-Key Cryptography

Symmetric-Key Cryptography (also referred to as Secret-Key Cryptography) take a single key into application. This common key is used for the encryption as well as the decryption process. Using a common single key creates a problem of securely transferring the key between the sender and the receiver.

1.6.1.2 Asymmetric-Key Cryptography

Asymmetric-Key Cryptography uses a pair of keys where an encryption with a private key, and a decryption with a public key. The key pair generated by this algorithm consists of a private key and a unique public key that is generated using the same algorithm.



Name	Goals	Advantages	Disadvantages	Domains	Implementations
Round Robin	Provide a system for publishing blocks amongst approved/trusted publishing nodes	Low computational power. Straightforward to understand.	Requires large amount of trust amongst publishing nodes.	Permissioned Systems	MultiChain
Proof of Authority/Identity	To create a centralized consensus process to minimize block creation and confirmation rate	Fast confirmation time Allows for dynamic block production rates Can be used in sidechains to blockchain networks which utilize another consensus model	Relies on the assumption that the current validating node has not been compromised Leads to centralized points of failure The reputation of a given node is subject to potential for high tail-risk as it could be compromised at any time.	Permissioned Systems, Hybrid (sidechain) Systems	Ethereum Kovan testnet, POA Chain, various permissioned systems using Parity
Proof of Elapsed Time (PoET)	To enable a more economic consensus model for blockchain networks, at the expense of deeper security guarantees associated with PoW.	Less computationally expensive than PoW	Hardware requirement to obtain time. Assumes the hardware clock used to derive time is not compromised Given speed-of-late latency limits, true time synchronicity is essentially impossible in distributed systems [13]	Permissioned Networks	Hyperledger Sawtooth

Figure 1.6: Consensus Comparison Matrix 2

1.6.1.3 Hash Functions

Hash Functions uses a cipher to generate a hash value of a fixed length from the plain text where Any changes in the input completely changes the output. It is nearly impossible for the contents of plain text to be recovered from the ciphertext.

Cryptographic hash functions have these important security properties [23]:

1. Input and output are one-way; it is impossible to compute the correct input value given some output value (e.g., given a digest, find x such that hash(x) = digest).

2. Cannot find an input that hashes to a specific output. More specifically, cryptographic hash functions are designed so that given a specific input, it is impossible to find a second input which produces the same output (e.g., given x, find y such that hash(x) = hash(y)).

3. Cannot find two inputs that hash to the same output. More specifically, it is impossible to find any two inputs that produce the same digest (e.g., find an x and y which hash(x) = hash(y)).

1.7 Features of blockchain

Several characteristics are associated with the blockchain: disintermediation,transparency, immutability and security. Let's describe them bellow :

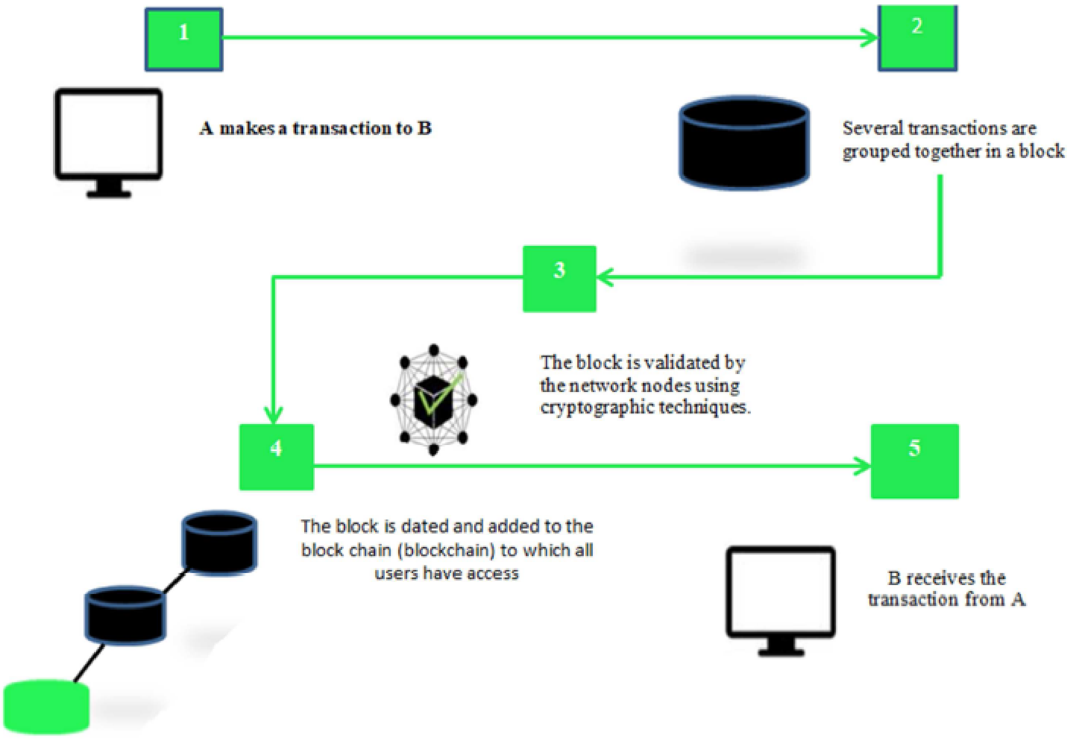

1
2
A makes a transaction to B
Several transactions are grouped together in a block
3
The block is validated by the network nodes using cryptographic techniques.
4
5
The block is dated and added to the block chain (blockchain) to which all users have access
B receives the transaction from A

Figure 1.7: Functioning of Blockchain

1.7.1 disintermediation

Blockchain technology makes it possible to trade without the control of a third party. The validation and the addition of a block result from a consensus between the user-validators, which is based on the possibility of verifying their validation work and which makes it unnecessary to control by a reference institution. Everything is done without the intervention of a central authority, the users operate the surveillance, and control each other, ensuring the certification of the safeguards and their consistency [21].

1.7.2 Immutability

Once data is registered to the blockchain and a enough number of participants have agreed on this state, the information is stored permanently and immutably. Changing the information contained in a particular block would require to also changing all the following blocks to her and this infeasible [9].

1.7.3 Transparency

Once a document is registered on the blockchain, this is enough to prove that it does exist at time T and that it has not been modified. The blockchain is qualified as transparent because anyone can download it in its entirety and check its honesty at any time. All blockchain users can thus see current and past transactions [17].

1.7.4 Security

Decentralized makes blockchain a secure technology which has great resistance, because all the data is copied to the different servers. This makes it resistant to cyber attacks or state control. This offers the blockchain a high level of security. The blockchain is therefore considered to be unassailable and inviolable [17].

1.8 Types of blockchain

Blockchain technologies can be roughly divided into three types:

1.8.1 Public Blockchain

public blockchains are permissionless distributed ledger system. they are open to the public, and anyone can participate as a node in the decision-making process. All users use a distributed consensus mechanism to decide the eventual state of the ledger. their primary advantages of using public blockchain are that it is secure and transparent and the biggest public blockchain examples are the Bitcoin and Ethereum blockchain [7].

1.8.2 Private Blockchain

A private blockchain is permissioned distributed ledger system working only in a locked network where they are open only to a consortium or group of individuals or organizations who have decided to share the ledger among themselves and there are one or more entities which monitoring the network.their primary advantages of using private blockchain are that it is Fast and Scalable and the biggest examples about that is Hyperledger Fabric of Linux Foundation [7].

1.8.3 Semiprivate blockchain (Hybrid)

With semiprivate blockchains, part of the blockchain is private and part of it is public, the private part is controlled by a group of individuals, while the public part is open for participation by anyone. This hybrid model can be used in scenarios where the private part of the blockchain remains internal and shared among known participants, while the public part of the blockchain can still be used by anyone[7].

1.9 Uses of blockchain

There is a lot of wide usage of Blockchain beyond finance, We mention some of them:

1.9.1 Electronic Voting

Blockchain technology provides an electronic voting system that provides security and less cost [15] becouse distributes individual voting information to thousands of computers around the world,that making it impossible to edit or delete votes once they have been cast. The Blockchain protocol would also maintain transparency in the electoral process, reducing the staff needed to conduct elections and providing officials with immediate results [10] .

1.9.2 Supply Chain Management

Blockchain technology provides transparency to the whole supply chain process. It gives businesses the ability to track the goods from the source points to their delivery points. These trackings are done accurately and provide a better way to handle goods and their condition [19].

1.9.3 Healthcare

Blockchain provides a transparent, trust, and immutable system that can change how healthcare can tackle the problems. For exmple, the medicines can be tracked by integrating it into the supply chain. Also, BurstIQ is a security platform that enables companies to securely transfer patient's data from one place to another, that making BurstIQ's proprietary Blockchain platform the leader in health insurance portability and accountability law compliant. to the HIPAA secure data platform [22].

1.9.4 Automobiles

With blockchain technology, the onboard sensors can not only capture the information but also securely and transparently store and distribute it with the authorized entities. These will improve trust and collaboration among the parties involved in the ecosystem including consumers, vehicles, and businesses.The automobiles executives are also accepting that blockchain technology can have its impact felt within the next three years.

1.9.5 Insurance

There are many companies that are actively working to integrate blockchain technology in their insurance organization. Etherisc is one of those companies that is providing an open source development platform for decentralized insurance applications.French insurance company Axa recently launched a policy called Fizzy via the Ethereum Blockchain. The purpose of travel insurance is to insure passengers against delays. The aim is to make the contracts and the settlement of claims more transparent on using Blockchain [5] .

1.9.6 Identity Management

Verifying a person's identity is a challenge in some areas of the business. However, with the help of blockchain technology, people's identities can be identified more securely and faster than ever before. This is based on extensive databases which allow identification and verification. In particular, existing identification documents: driving licenses, passports and identity cards could thus be implemented digitally in complete security. Data loss would also be avoided since the data is stored in a decentralized manner [13].

1.10 Smart Contracts

The theory of smart contracts was first introduced by Nick Szabo in the late 1990s in an article titled Formalizing and Securing Relationships on Public Networks. The true potential and benefits of it were appreciated with the invention of Bitcoin and the development in blockchain technology. Smart contracts are described by Nick Szabo as "A smart contract is a computerized transaction protocol that executes the terms of a contract where the general objectives of it are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement),reduce exceptions both malicious and accidental, and reduce the need for trusted intermediaries"[7].

1.10.1 Definition

A smart contract is computer protocol that is deployed using cryptographically signed transactions on the blockchain network Where it aims to organize and manage agreements between members of the network and it is executed by nodes within the blockchain network. All nodes that execute the smart contract must derive the same results from the execution, and the results of execution are recorded on the blockchain. smart contract is also activated automatically when the conditions previously specified in this contract are met, which makes us dispense with the presence of a third party [23].

a smart contract has the following four properties:

1. Automatically executable
2. Enforceable
3. Semantically sound
4. Secure and unstoppable

In any smart contract, an automatically executable property and an enforceable property must be realized. As for a Semantically sound property and a secure and unstoppable property that is not required or enforceable in some scenarios, for example:a financial derivatives contract does not perhaps need to be semantically sound and unstoppable but should at least be automatically executable and enforceable at a fundamental level. On the other hand, a title deed needs to be semantically sound and complete, therefore, for it to be implemented as a smart contract, the language must be understood by both computers and people [7].

1.10.2 Programming language

There is a lot of smart contract programming languages like Solidity, Pact, Liquidity but in this part we will discuss solidity:

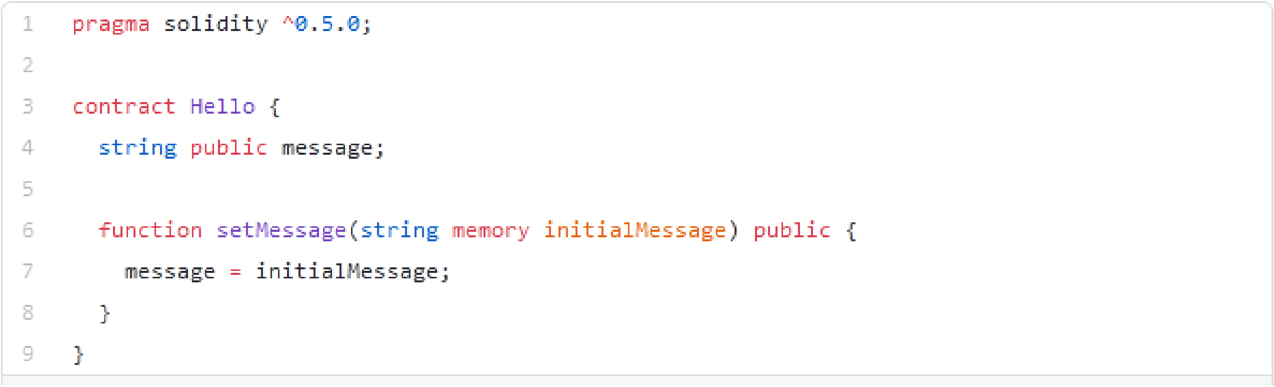
1.10.2.1 solidity

Solidity is a programming language with a similar syntax to JavaScript built for writing smart contracts which is the primary choice language for implementing smart contracts on the Ethereum platform.Solidity was influenced by C++, Python and JavaScript and runs on the Ethereum Virtual Machine (EVM) that is hosted on Ethereum nodes connected to the blockchain.It supports inheritance, libraries and complex user-defined types among other features [16].

With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

1.10.2.2 Contract Implementation

In figure 1.8, we show a smart contract example , we named 'Hello'.



```
pragma solidity ^0.5.0;

contract Hello {
  string public message;

  function setMessage(string memory initialMessage) public {
    message = initialMessage;
  }
}
```

Figure 1.8: Hello implementation in Solidity

1.11 Advantages and Disadvantages of Blockchain

Every technology has it's advantages and disadvantages that we need to know.we present it as follows :

1.11.1 Advantages

1.11.1.1 Decentralized system

It is the main advantage of blockchain technology becouse the system works without intermediary and all participants make the decisions [12].

1.11.1.2 Trusty

Trust is the result of each transaction is recorded to the Blockchain wich are available to every participant and cannot be changed or deleted. Trust can be increased further, because there can be more shared transaction [12]

1.11.1.3 Security

Security is the other main advantage where it becomes harder for any type of attack to be a success with the increase of people work on the network so It is impossible to malicious actors to gain control over the network.

1.11.1.4 High availability

The system becomes highly available becouse it is based on thousands of nodes in a peer-to-peer network where the data is replicated and updated on every node. Even if some nodes leave the network or become inaccessible, the network as a whole continues to work, so making it highly available [7].

1.11.1.5 Faster dealings

Faster dealings: blockchain play a vital role by enabling the quick settlement of trades. Blockchain doesn't demand a lengthy process of verification and reconciliation because a single version of agreed-upon data is already available on a shared ledger between all participant organizations in this blockchain [7].

1.11.2 Disadvantages

1.11.2.1 The high energy consumption

It is the main disadvantage of the Blockchain. The consumption of power is necessary for protection a real-time ledger. The network's miners are trying to solve a lot of solutions per seconds to validate transactions. They are using substantial quantities of computer power and the signature verification is the one of the reasons to the high energy consumption too[12].

1.11.2.2 The high costs

The average cost of the transaction is between 75 and 160 dollars and most of it covers by the energy consumption .this is one of disadvantage of the Blockchain [12].

1.11.2.3 Scalability

This is the most important problem that could mean the difference between wider adaptability of blockchains or limited private use only by consortiums where the general approach toward tackling the scalability issue generally revolves around protocol-level enhancements. Another approach to addressing limitations in blockchains has been recently proposed by Miller and others in their position paper" On Scaling Decentralized Blockchains" [7].

1.12 Blockchain today

1.12.1 Cryptocurrencies

Cryptocurrencies have been in existence since 2009, and they are an alternative to the traditional central banking system where the digital currency is designed to be able to exchange electronic cash using strong encryption to ensure the security of funds and transactions, and there are 1,833 cryptocurrencies listed with a market cap of 200 billion dollar. Let's get to know some of them [11]:

1.12.1.1 Bitcoin

Decentralization of currency was made possible for the first time with the invention of bitcoin and the one that has proven to be extremely secure and stable from a network and protocol point of view. It is the first application of blockchain technology where can be defined as it's a combination of peer-to-peer network. Nodes in this peer-to-peer network talk to each other using the Bitcoin protocol. The double spending problem is resolved in Bitcoin by using a distributed ledger where every transaction is recorded permanently [7].

1.12.1.2 Ethereum

Vitalik Buterin provided a conceptual of Ethereum in November, 2013. Ethereum is a decentralized platform that runs smart contracts to handle funds utilizing blockchain technology to overcome downtime and third-party interference . The first version of Ethereum, was released in May, 2015. It is an applications that run exactly as programmed without any possibility of downtime, censorship or fraud [11].

The Ethereum blockchain platform is one of the best platforms for developers to build decentralized applications. Key features of Ethereum are:

1. open to the public.

2. Proof-of-work system.

3. Application in several languages such as C++ and python.

1.13 Conclusion

During this chapter, we outlined in detail the blockchain, its principles, and its usefulness as it became clear why many people are interested in this new technology and are thinking of developing it in order to achieve efficiency. In the next chapter, we will see an area in which we apply the blockchain, as it is completely different to digital currencies.

Chapter 2

Driver's license

2.1 Introduction

In the previous chapter, we presented the theoretical background of the Blockchain technology, where the latter has developed its applications on a large scale with technological progress, as this technology has proven its importance in many areas, including the domain of transportation, and in this study we will focus on the important part of this domain called the driver's license.

In this chapter, we will cover the basic concepts and everything related to driver's license and drivers.

2.2 Definition of driver's license

A driver's license is an official driving document, received after passing a driving test, that giving permission to drive one or more types of motor vehicles, such as cars, motorcycles, trucks or buses on a public road in a specific geographic area [9].

The successful completion of the driving license examination, the admitted candidate is issued with a temporary driving license called a probationary driving license. This document is valid for a period of two years, during this period, the new driver cannot apply for a new category.

At the end of this two year probationary period, the new driver will apply a demand to get permanent license.Citizens over the age of sixty are required to renew their driving license every two years regardless of the categories obtained.

2.3 Type of driver's license

The driving license types are as follows [9]:

2.3.1 Category A

1. A1: Light motorcycles (cylinder volume from 50 to 80 cubic centimeters) with a maximum power of 11 kW, and tricycles and four-wheelers (cylinder volume equal to or less than 125 cubic centimeters) with a maximum power of 15 kW.

2. A2: Motorcycle (cylinder volume from 80 to 400 cubic centimeters) with a power not exceeding 35 kW and with a power/weight ratio not exceeding 0,2 kW/kg.

2.3.2 Category B

Vehicles with a maximum authorized mass not exceeding 3.5 tons, a B license is sufficient to carry a maximum of 8 people, excluding the driver, a total of 9 people in the same vehicle. A trailer weighing not more than 750 kg can be towed so that the total weight does not exceed 3.5 tons.

2.3.3 Category C

1. C1: Vehicles are allowed to be driven to transport goods whose weight is between 3.5 tons and 19 tons for single vehicles. A trailer with a weight not exceeding 750 kg can also be towed. To obtain it, you must have a driver's license of category B, and age 25 years old or more .

2. C2: Vehicles are allowed to be driven to transport goods whose weight exceeds 19 tons for a single vehicle or whose weight exceeds 12.5 tons for a vehicle running next to a group of vehicles or an articulated vehicle . A trailer weighing no more than 750 kg can also be towed. To obtain it, you must have a category B driver's license, and are 25 years of age or over.

2.3.4 Category D

Public transport vehicles for persons, which include more than 8 seats plus the driver's seat (more than 9 seats). The vehicle of this class exceeds the total licensed gross weight with a payload of 3.5 tons and can tow a trailer whose total licensed weight does not exceed 750 kg. To obtain it, you must have a category B driver's license, and are 25 years of age or over.

2.3.5 Category E

A driver's license of category E is a license to drive vehicles of classes (B, C, and D) that tow a trailer of more than 750 kg. To obtain this license, the candidate must have obtained a driver's license of category B with a precedence of more than two years, and are 25 years of age or over.

2.3.6 Category F

It is a driver's license for people with special needs, where cars of category A1, A2 or B are driven by the defective and specially adapted to take into account their disability.

the duration for the renewal of the driving license is fixed as follows [8]:

1. Category F is valid for two years.
2. Categories C1, C2, D and E are valid for five years.
3. Categories A1, A2 and B are valid for ten years.

over the age of sixty are required to renew their driving license every two years regardless of the categories obtained.

2.4 The process of obtaining a driver's license

Driving license has become a necessity in our daily life and it is impossible to think of getting it in a few days. Here we will explain the steps needed to get it:

1. Step 1: Pass the theorical test:

   From the age of 16 can pass the road code. Once he enroll in the driving school, he will be able to train to be able to pass the theorical code test. he will need to complete a number of training sessions before he can take the test. The candidate must pay a fee to pass the road code, and the test is done as follows:

   The candidate answers a number of questions, which include traffic lights, transit priorities and restrictive laws. The candidate must answer each question correctly to successfully pass the test, thus, he automaticly start the pratical part .once he failed this test , he could retake the test after two weeks.

2. Step 2: Pass the practical test:

   Any candidate for the driver's license test must have completed at least 20 hours of driving with an engineer to be eligible to register for the driving test.

Your training center handles all registration procedures for the driver's license after the candidate pays a fee to pass this test, and you have the right to fail the driver's license exam 4 times in th same year. You will need to take your road code if you do not get your driver's license within the time allowed.

3. Step 3: If the candidate pass all tests ,he will get a temporary license valid for two years, during this period, the license holder has 12 points . In the event of loss of points during this period, the holder of this license shall continue the training at his expense to recover the lost points.
4. Step 4: After two years, the driver obtain a permanent license valid for 10 years, and 24 points are awarded if the license holder has not committed any offense.

Finally, we will resume all the steps of the process of obtain driving license in the scheme 2.1 .

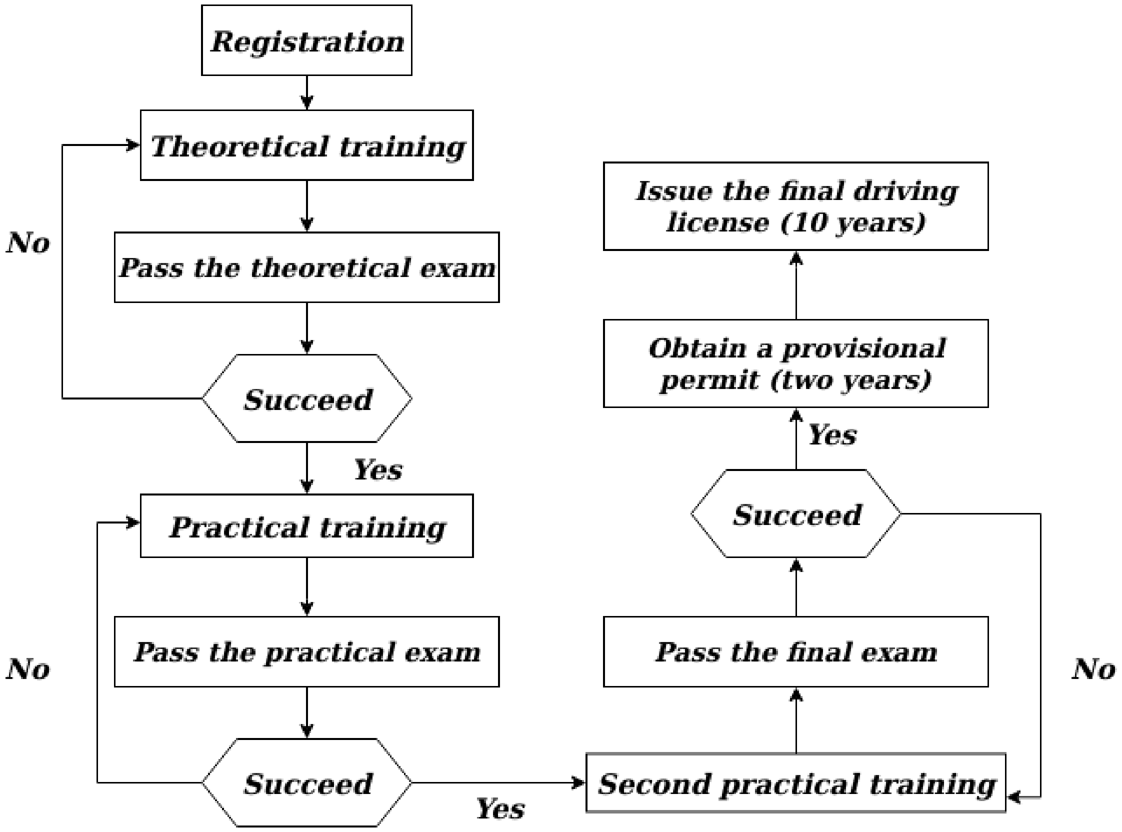

Registration
Theoretical training
Pass the theoretical exam
Succeed
No
Yes
Practical training
Pass the practical exam
Succeed
No
Yes
Second practical training
Pass the final exam
Succeed
No
Yes
Obtain a provisional permit (two years)
Issue the final driving license (10 years)

Figure 2.1: The process of obtaining a driver's license.

2.5 Driver's license problems

The existing system is known that has a lot of loopholes that's what make dishonest people use every possible means to achieve their goals like issue licenses to unqualified drivers in exchange for a bribe. Let's give below some of the major problems with the existing system:

1. Issuing a driver's licence to a person who does not deserve it with the help of people working in the system, either by inserting unauthorized data into the database or by using identity theft.
2. In several cases, traffic police fail to differentiate between an authorized driving license and a forged one.
3. System with security faults and therefore exposed to Attacks, it is possible for fraudsters with computer knowledge to find ways to access the system's database and modify the information.
4. Traffic rules violators arn't held responsible for their actions and any violation is usually not traceable by the system.

2.6 Traffic rules

Everyone must abide by the traffic law to avoid violations, and the most important traffic rules on Algerian roads are the following:

1. Traffic moves on the right side of the road.
2. Minimum driving age is 18 years old.
3. Cell phones are not allowed while driving, even with a hands-free device.
4. Children under 10 years old are not permitted in the front seat.
5. The speed limit for drivers with a new driver's license is 80 km/h.
6. Dark windows are illegal.
7. A solid line can't be skipped.
8. Dangerous change of direction without the use of an order indicator is illegal.
9. Rotation is not permitted on highways.

10. Official processions and military convoys, including ambulances, fire trucks, and funeral cars, always have priority.

11. Vehicles already in the roundabout have priority over oncoming traffic.

12. The permissible level of alcohol in the blood is 0.01

13. The driver and the passenger sitting in the front seat must wear a seat belt.

2.7 Traffic offense penalties

Fines and penalties are imposed on drivers who violate traffic rules. This can be explained briefly as follows [6]:

1. Violation of the first degree:

   One point is deducted. These violations relate to provisions related to lighting, signaling, and curbing bicycles, submitting vehicle documents, non-conformity of vehicle equipment, and wearing seat belts. The fine has been set at 2000 DZD.

2. Violation of the Second degree:

   2 points are deducted and a fine is fixed at 2500 DZD. As these offenses relate to crossing continuous lines, chaotic parking lots, and exceeding the authorized legal speed limit ..

3. Violation of the third degree:

   There are 13 violations, in which 4 points are deducted, and their fine is set at 3000 DZD, as for motorcyclists, among them is not wearing shaft. Transporting children under 10 years old on the front seat...

4. Fourth degree violation:

   There are 30 offenses, in which 6 points are deducted and a fine set at 5,000 DZD. Among them are a violation related to road intersection, traffic priority, violation of prohibited maneuvers in the railways, driving cars without lights, and using cell phones and headphones.

   If drivers do not pay the fines within 45 days, a report will be sent to the Attorney General. The fine will be raised to 3,000 dinars for first-degree offenses, 4,000 dinars for second-degree offenses, 6,000 dinars for third-degree offenses, and 7,000 dinars for fourth-degree offenses.

2.8 Related works

Over the years, several methods of verifying identity documents have been proposed such as signature verification and fingerprints. In our days, the solution proposed is using blockchain technology.

2.8.1 Implementation of the Blockchain in the Optimization of the Security of Transport Documents (Driver's License and Vehicules Registration Cards):

Georges Bell Bitjoka 1 , Pierre Bilong 2 , Moses Macaire Nnanga Edoa 2

This article focuses on Cameroon facing challenges in its licensing system and describes the means by which dishonest drivers obtain forged documents and licenses, such as using specialized software or exploiting weaknesses in the system.

The objective of this work is eliminates the possibility of drivers obtaining licenses without proper training where only qualified persons who have received appropriate driver training and hold a authentic driving license can be authorized to drive vehicles. and allows the traffic authorities to verify the authenticity of their license.

To achieve this, they proposed a framework using Corda, they used a consortium blockchain to set up them system.

They noted that the strengths of this work are:

The operation of the system and its database is decentralized in each node of the blockchain network, the time to obtain a driver's license and vehicle registration is reduced, traceability and transparency during audit and creation of a driving license for a person who has not been accepted The competition for a driving license becomes difficult because all the nodes of the network have results.

2.8.2 A Novel Framework for Blockchain Based Driving License Management and Driver's Reputation System for Bangladesh:

Md. Mumtahin Habib Ullah Mazumder , Tahmidul Islam ,Md. Raiyan Alam

This article focuses on Bangladesh facing challenges in its current system which is notorious for loopholes and inefficiency where this is often abused by the authorities who issue licenses to unqualified drivers in exchange for a bribe.

The objective of this work is bring transparency in the evaluation and prevent illegal transactions where every step of the licensing procedure and any violation of traffic regulations will be recorded as transactions on the blockchain. All the associated actors can interact with the Blockchain using User App.

To achieve this, they proposed a framework using Hyperledger Fabric platform wich provides a distributed permissioned blockchain .

Their proposed system makes the licensing process easier, more efficient and secured while respecting privacy of the actors involved. their system eliminates the brokers, third parties, redundant queue for authentication, verification and attestation.

2.9 Conclusion

In this chapter, we learned about everything related to the driver's license and the many challenges it faces, and in the next chapter we will learn about the proposed system to solve these challenges.

Chapter 3

Design and Implementation

3.1 Introduction

After we have chosen the blockchain technology because of its advantages to establish our project, in this chapter, we will present the general structure of our system in addition to its detailed design applied, and we will also present the implementation of the specific system by explaining the hardware and software environment on which our system was developed. Then cite the tools and languages used, then we move on to describe the main components of our system by presenting screenshot.

3.2 Global Architecture

A decentralized application or "Dapp" is an application developed according to the standards of distributed consensus protocols. As a result, these applications are deployed on the Blockchain and shared through peer-to-peer networks.

Dapps are more transparent in their operation and more powerful which aims to eliminate the intermediaries who are omnipresent, and to improve traceability and transparency of information and that what make us propose a decentralized system for managing driver's license starting from its creation, where every steps of the licensing process and traffic violation are recorded as a transaction on the Blockchain. All associated actors can interact with the Blockchain through the interface and the scheme 3.1 will show the overall system workflow.

3.2.1 Interaction of user sing in

Users who want to obtain driver's license need to fill in the form by entering their information through the interface. those information being about the user such as:

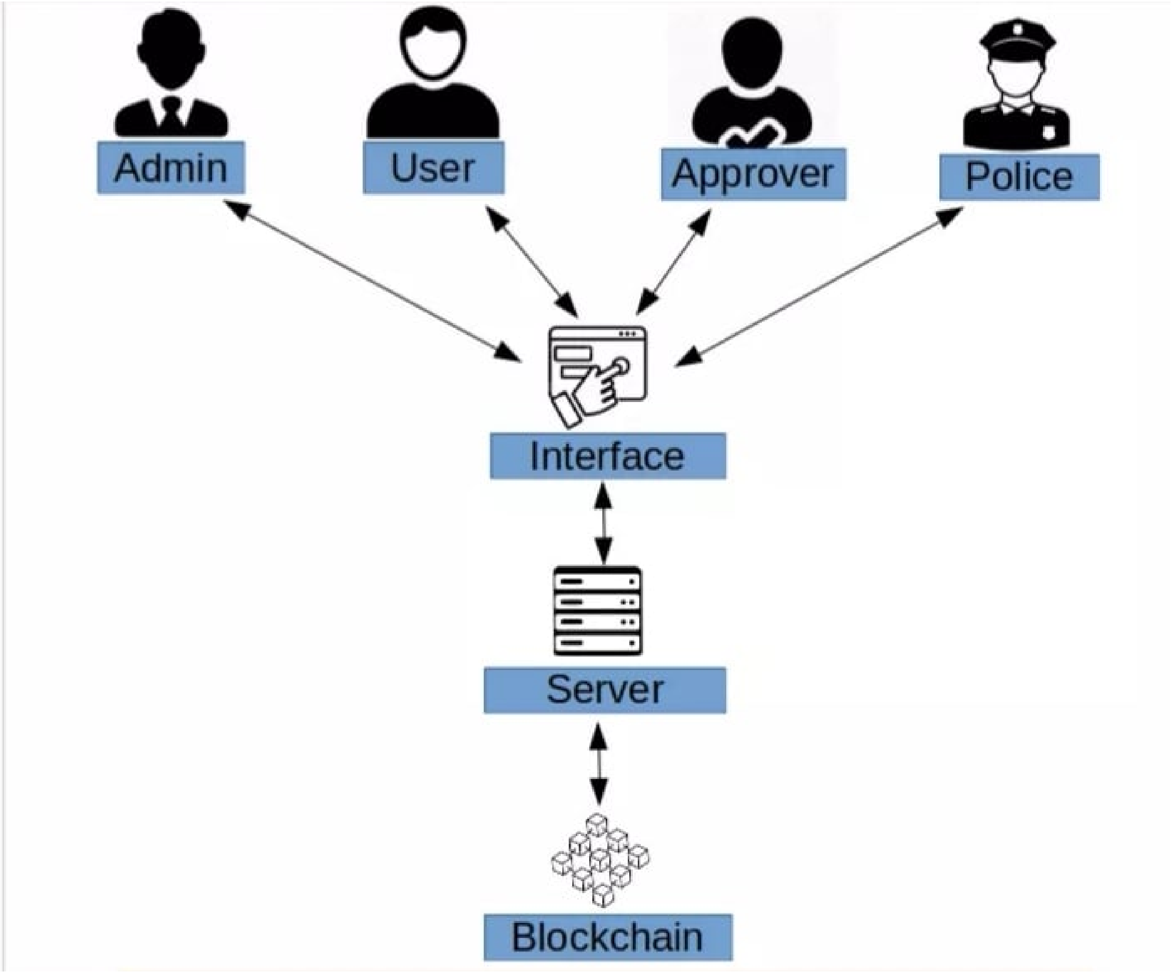


Figure 3.1: Overall system workflow

The full name, number, date of birth,etc. the information is sent to the server, and the server requests the blockchain to check for duplicates and the user fulfills all the necessary conditions, such as reaching the age specified for obtaining a driver's license. In the absence of duplicate accounts , the server collects information about user and creates an account for him and publishes a copy of the the learner's permit which is a limited license issued to anyone undergoing driving training but if a duplicate account exists, the user will be asked from him to login because he already has an account .

3.2.2 Interaction to obtain driver's license

After the user gets an account ,and learn all his lessens , he will get the date of exam. the user goes to pass the exam where his data is extracted from the blockchain by the approver to verify his information.The results of the exam of user are stored in the blockchain for written and Practical test. If the scores are satisfactory after passes the tests his license is activated and the user is then notified of his results by the approver through the interface and the process of obtaining a driver's license is carried out according to the scheme 2.1.

3.2.3 Interaction of police officers with system

Our system will also help police officers so that they can verify the integrity of the license by requesting a blockchain query to find the information of a specific driver. And in the event of a traffic violation, the police file a complaint against this driver, which contains information from the police report such as the complaint id, the license holder's id, the level of the driver traffic violation and points that must be removed. The complaint is then sent and stored in the blockchain, and if the points deduct to zero, the license status will change from active to stopped.

3.2.4 Use case diagram for system

The figure 3.2 is presenting a use case diagram of our system ,thus, it is explaining the role of each actor in the system.

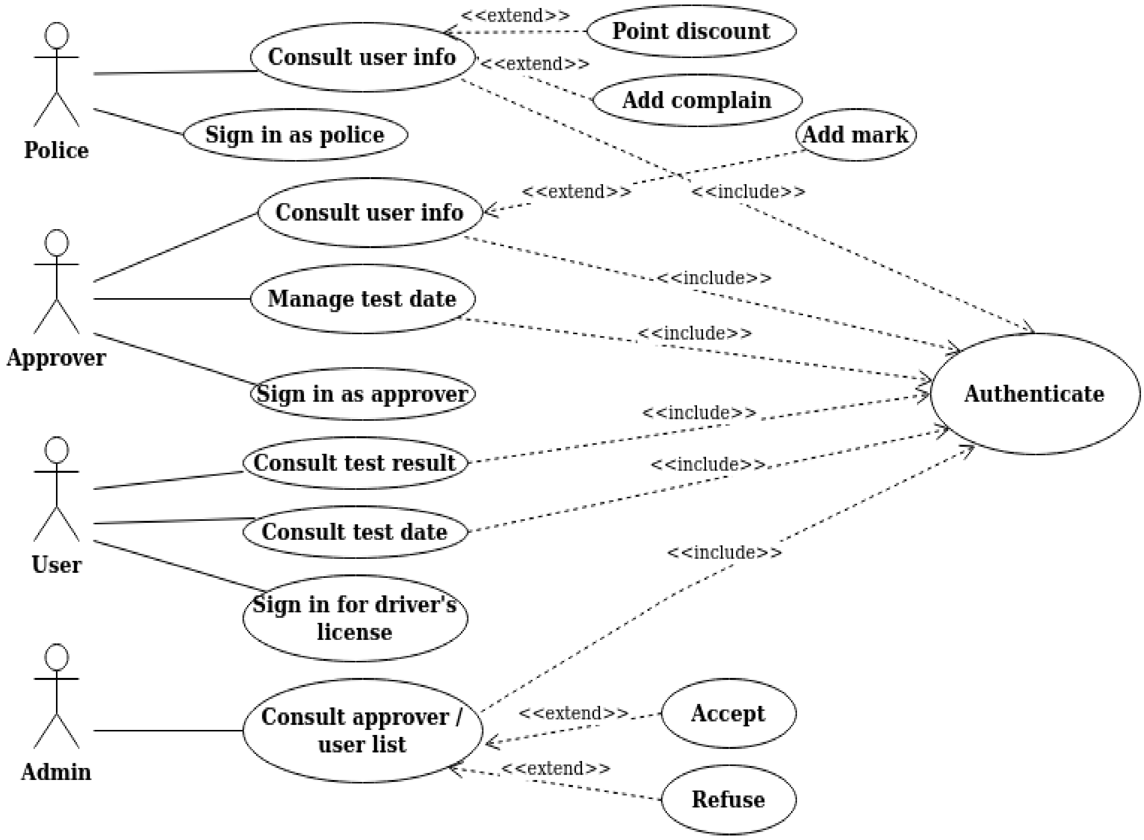


Figure 3.2: Use case of system

3.3 Detailed architecture

In this part, we present the sequence diagrams to define the different interactions between each of the actors and the components of the system.

3.3.1 User Sequence Diagram

the "User" sequence diagram shown in Figure 3.3 present the interaction and all scenarios of user with system where the first one is registration scenario.Once the new user want to has a driving license , he need to filling up a form by inputting credentials such as full name, national identity, blood group, ect through the interface .Our system sends the data to the server wich performs a query on the blockchain by using *searchbyid()* method for duplication check and fulfilling all the necessary conditions, such as reaching the age specified for obtaining a driver's license.If there is no duplicate account and he fulfilling all the conditions to user ,the server collects information about user and creates an account for him using *adddriver()* method and publishes a copy of the the learner's permit which is a limited license issued to anyone under going driving training .In case there is a duplicate account exists, the user will be asked to login because he already has an account.

Second scenario is login, once the user login , he will be able to see his dashboard.if he was a learner , he can see the test date that he will be pass and also he can see his result and history. if he is a driver , he will be able to see in his dashboard all his information and the history of all violations issued against him.

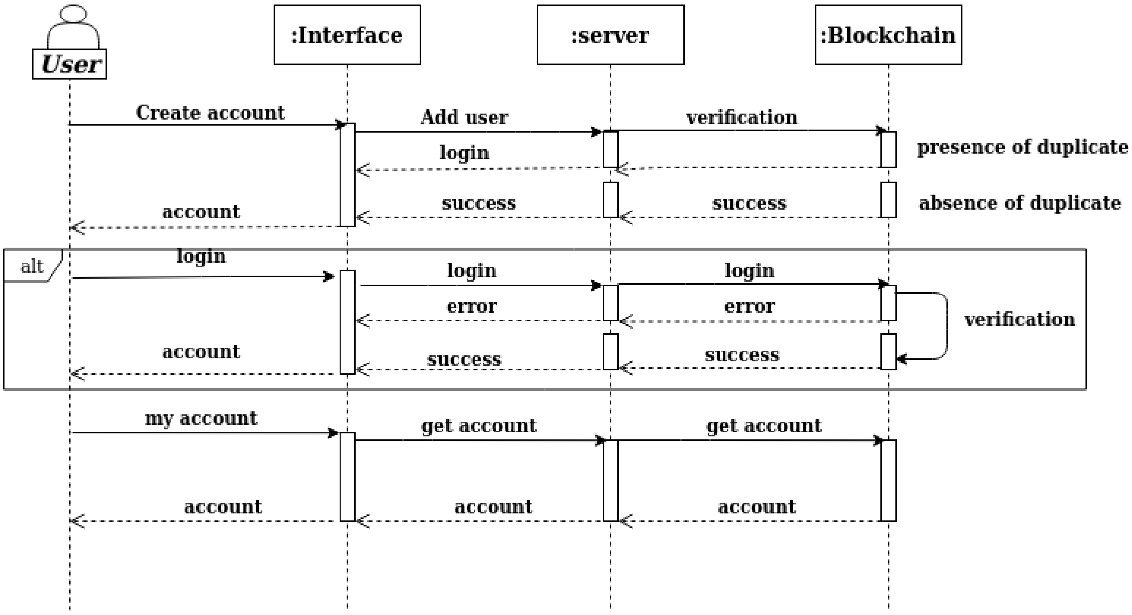


Figure 3.3: "User" sequence diagram

3.3.2 Sequence diagrams of admin

A "Admin" sequence diagram shows admin interactions with our system . It depicts the actors involved in the scenario and the sequence of messages exchanged between the actors needed and admin to carry out the functionality of the scenario.

The administrator first needs authentication to access the system where he will login with his username and password through the interface. our system sends the data to the server wich performs a query on the blockchain for authentification check and there is no error in password or username. if there is no error ,the server collects information and open the admin dashboard , if there is error he will receive message to try again .

The administrator has the authority to access the list of users and their information as well as the list of approvers and their records, and he can accept or refuse both or them, also he can access the Blockchain. In addition, he is the one who must verify the correctness of the information entered by the users who want to sign up .

Scheme 3.4 illustrates the role of the admin.

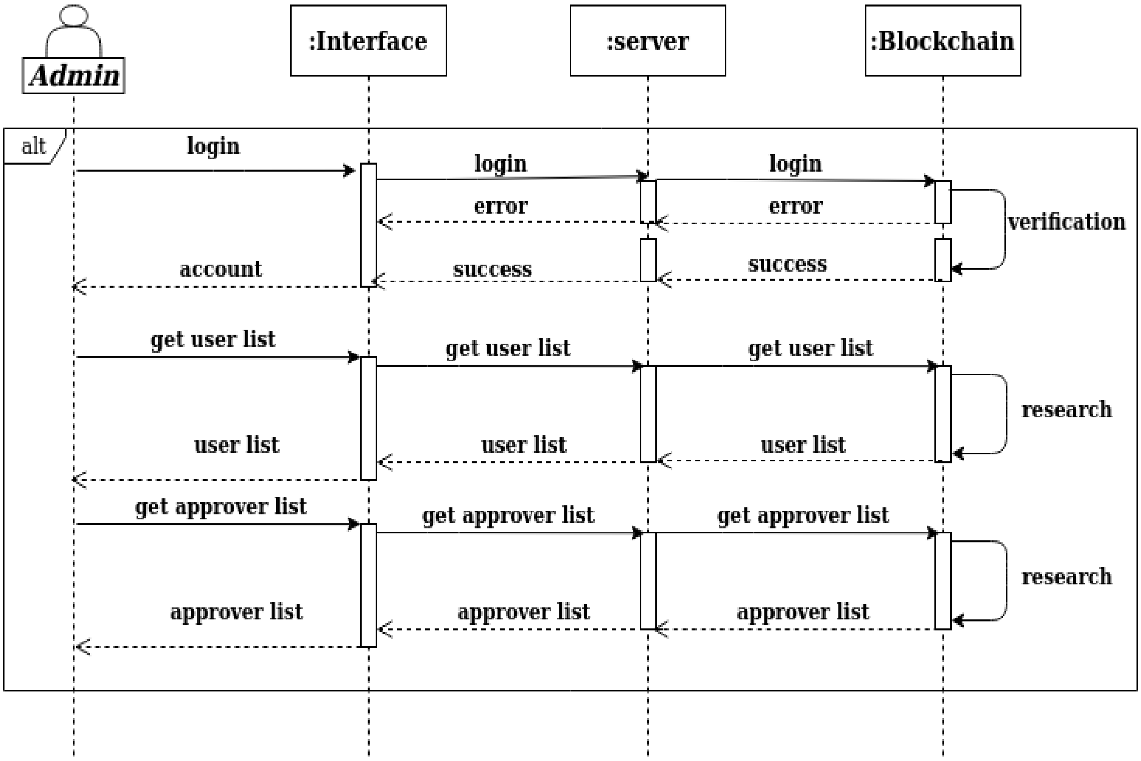


Figure 3.4: "Admin" sequence diagram

3.3.3 Sequence diagrams of police

Interaction of the policeman scenarios are defined in sequence diagram as shown in Figure 3.5,where the first one is registration scenario which lies in his login into the system with his username and password through the interface. our system sends the data to the server wich performs a query on the blockchain for authentification check and there is no error in password or username. if there is no error ,the server collects information and open the policeman dashboard .

After he login and through the interface he could seeing all the information and history of the driver ,thus, he has the possibilite to check driver's identity and the integrity of the license through request for a query to the blockchain by invoking searchbyid() method .

If the policeman find any violation from the driver ,he is filing a complaint against him by filling up a form by inputting id driver, the cause and level of violation and the point reduced through the interface then this complain sent and stored in the blockchain using addcomplain() method. If several police complain and deduce points of driver to zero, the status of the license will be changed from Active to Stalled.

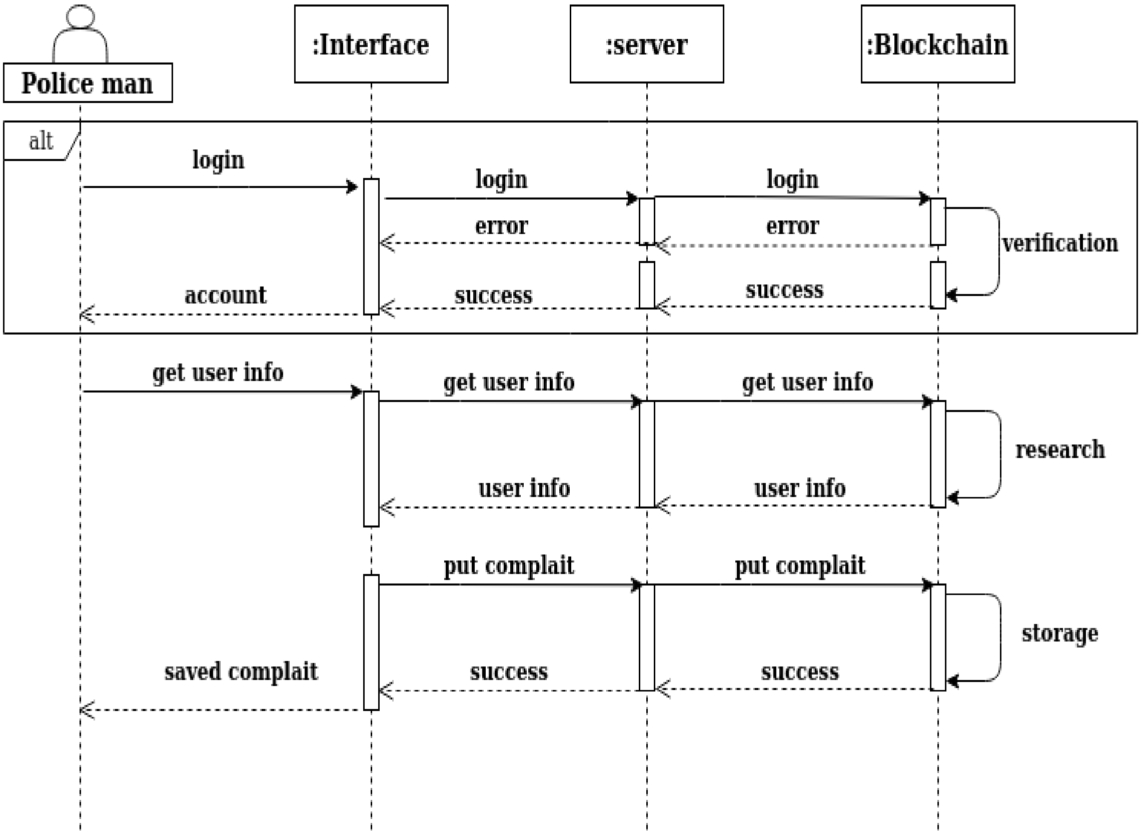


Figure 3.5: "Police" sequence diagram

3.3.4 Sequence diagrams of approver

Scenarios of "Approver" tasks are defined in sequence diagram As shown in Figure 3.6,where the first one is registration scenario. once the approver filling up a form by inputting credentials such as full name,national Identity ,ect through the interface.

Our system sends the data to the server wich performs a query on the blockchain by using searchbyid() method for duplication check and fulfilling all the necessary conditions, such as he able to educating learners. if there is no duplicate account and he fulfilling all the conditions, the server collects information about him and creates an account for him using addapprover() method and open his dashboard. in case there is a duplicate account exists, the approver will be asked to login because he already has an account.

After the approver login in the system where he will login with his username and password through the interface. our system sends the data to the server wich performs a query on the blockchain for authentification check and there is no error in password or username. if there is no error ,the server collects information and open the approver dashboard .

The approver must specify the test date for users and also access their files of course, to verify their information by searchbyid() method. After passing the exam, the approver puts the points in the blockchain through the interface and server and also approver notifies the user of his results through the interface.

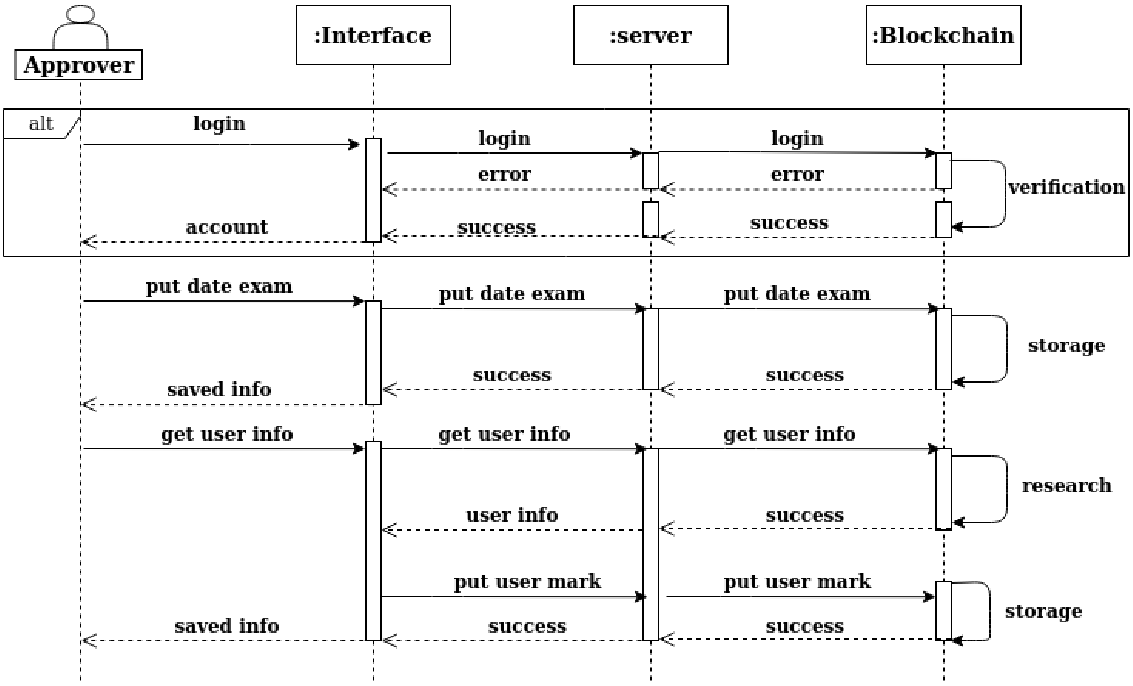


Figure 3.6: "Approver" sequence diagram

3.4 Implementation

3.4.1 Development Tools

Firstly, the main tools that we used for the realization of our system are as follows:

3.4.1.1 System Configuration and Operating System

The project are performed on processeur Intel® Pentium(R) CPU B960 2.20GHz, and 4 Go of memory. We implement the project using ubuntu 18.04.5 LTC.

3.4.1.2 Node.js

Figure 3.7: Nodejs logo

Is an open-source, environment that runs on the V8 engine and executes JavaScript code outside a web browser.it lets developers to write command line tools and for server-side scripting to produce dynamic web page content before the page is sent to the user's web browser. To develop smart contracts, we must configure our environment by installing Node Package Manager (NPM), supplied with Node.js [3].

3.4.1.3 Ganache

Figure 3.8: Ganache logo

Is a personal blockchain that allows you to create an Ethereum blockchain so that you can run tests, execute commands, and inspect status while monitoring the operation of the chain. It gives you the ability to perform whatever actions you would do on the main channel without the cost. Many developers use it to test their smart contracts during development.

Thanks to ganache we can have 10 Ethereum accounts with a balance of 100 ether (fake ether) for each account and even it allows us to examine everything that is happening in this blockchain which is why we have chosen it. All versions of Ganache are available for Windows, Mac, and Linux [2].

3.4.1.4 Metamask

Figure 3.9: Metamask logo

Is a crypto wallet that can be used on Chrome, Firefox browsers. It works as a bridge between normal browsers and the Ethereum blockchain. It is Accessible to everyone, its primary purpose is to make the development of decentralized applications easier.

3.4.1.5 Remix IDE

Figure 3.10: Remix IDE logo

A powerful open source tool that helps you write Solidity contracts right from the browser. Written in JavaScript, Remix supports both in-browser and locally .We chose it because :

1. It is very practical and very relevant to learn to code on Solidity
2. You can access it just by browser and there is nothing to install
3. We automatically have the latest versions of Solidity
4. It allows you to compile and execute smart contracts instantly, in all kinds of blockchains, i.e. you can deploy a smart contract in the real Ethereum blockchain directly from Remix. It is therefore very flexible .

3.4.1.6 Truffle

Figure 3.11: Truffle logo

Is a Framework for developers to launch a smart contract project with one click and provides you with a project structure, files and directories that make it easy to deploy and test. We chose truffle because it is powerful and it helps us and facilitates interaction with our smart contract.

3.4.1.7 web3.js

Is a collection of libraries which allow you to interact with a local or remote ethereum node, using a HTTP connection. It interacts with the Ethereum blockchain and smart contracts, and more [4].

3.4.1.8 React.js

Figure 3.12: React js logo

React or React.js is a free JavaScript library developed by Facebook since 2013. The main purpose of this library is to facilitate the creation of single page web application, through the creation of state dependent and generating components an HTML page at each change of state.React is a library that only manages the interface of the application, seen as the view in the MVC model. thus,It can be used with another library or an MVC framework like AngularJS. The library stands out from its competitors with its flexibility and performance, working with a virtual DOM and only updating the rendering in the browser when needed.

3.4.1.9 Visual Studio Code

Figure 3.13: Visual Studio Code logo

Is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, Java, Python, PHP, Go) and runtimes (such as .NET and Unity) [1].

We chose it to implement the project because free and open-source, also it easy to use.

3.4.1.10 JavaScript

Figure 3.14: JavaScript logo

Is a scripting programming language primarily used in interactive web pages and as such is an essential part of web applications. Along with HTML and CSS technologies, JavaScript is sometimes considered one of the core technologies of the World Wide Web. A large majority of websites use it, and the majority of web browsers have a dedicated JavaScript engine to interpret it, regardless of security considerations that may arise if necessary. It is an object-to-prototype oriented language.

3.4.1.11 Solidity

Figure 3.15: Solidity logo

Is an object-oriented, high-level language for implementing smart contracts wich are programs govern the behaviour of accounts within the Ethereum state.Solidity is influenced by C++, Python and JavaScript, and is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

3.4.2 Environment Configuration

To implemente the system of the driver's license management web application interacting with the blockchain. First, we will create a directory that will contain the files of our project as follow :

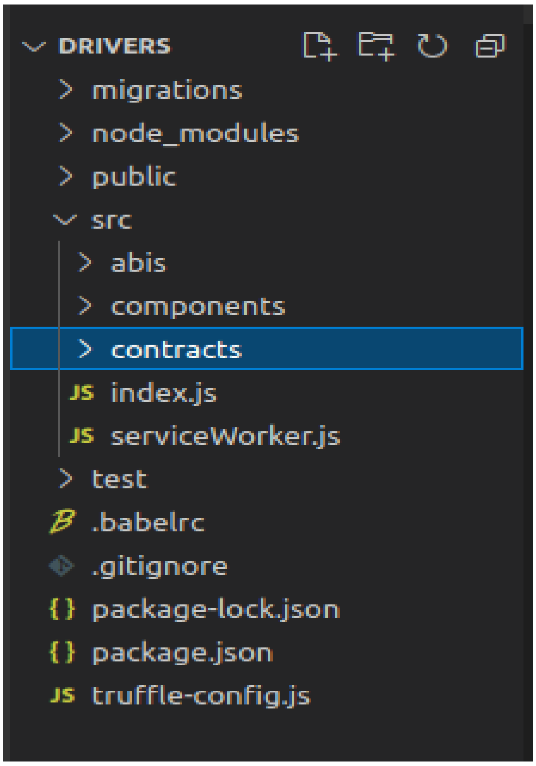
```
$ mkdir drivers
$ cd drivers
```

Now that we are in our file, let's get started building our the project manually. The project directory structure look like at Figure 3.16.

This is a truffle project that I have created to building full blockchain applications, all the the dependencies that we need for project are declared on package.json after that we installed from the command line like this: **$ npm install**

Let's examine the project directory structure that we just created:

- contracts directory: this is where all smart contacts live that handles our migrations to the blockchain.



```
DRIVERS
  > migrations
  > node_modules
  > public
  ⌄ src
    > abis
    > components
    > contracts
    JS index.js
    JS serviceWorker.js
  > test
  .babelrc
  .gitignore
  {} package-lock.json
  {} package.json
  JS truffle-config.js
```

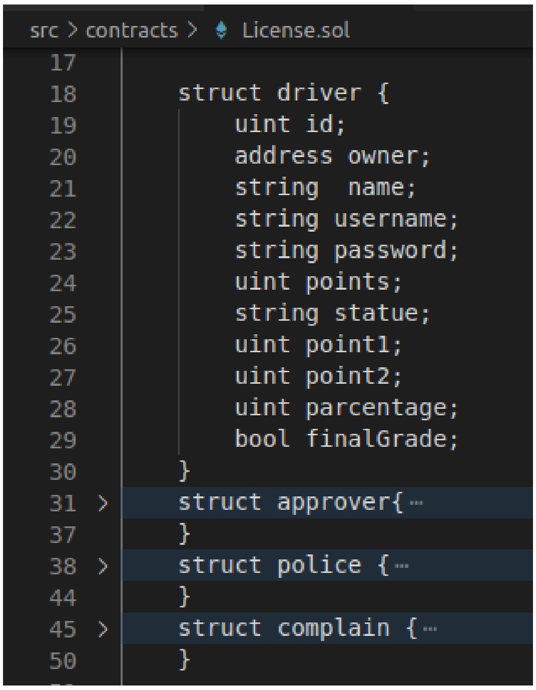
Figure 3.16: Directory structure of project

- migrations directory: this is where all of the migration files live. Whenever we deploy smart contracts to the blockchain, we are updating the blockchain's state, and therefore need a migration.
- node_modules directory: this is the home of all of our Node dependencies we installed.
- test directory: this is where we'll write our tests for our smart contract.
- Components directory: this is where we will develop our client-side application.
- truffle-config.js file: this is the main configuration file for our Truffle project, where we'll handle things like network configuration.

3.4.3 Writing Smart Contract

Inside the contract form, we define seven state variables,one to store the address of owner of contract, which is the account for deploying the smart contract, "driverCount" and " imageCount" to keep track of how many drivers and complains exist in the smart contract, "Result" for result of driver's tests, and other static variables.

We created data structures, with attributes by creating a driver struct, approver struct, police struct, cmplain struct. Structures stored all the attributes that we need and we modeled it like at Figure 3.17.



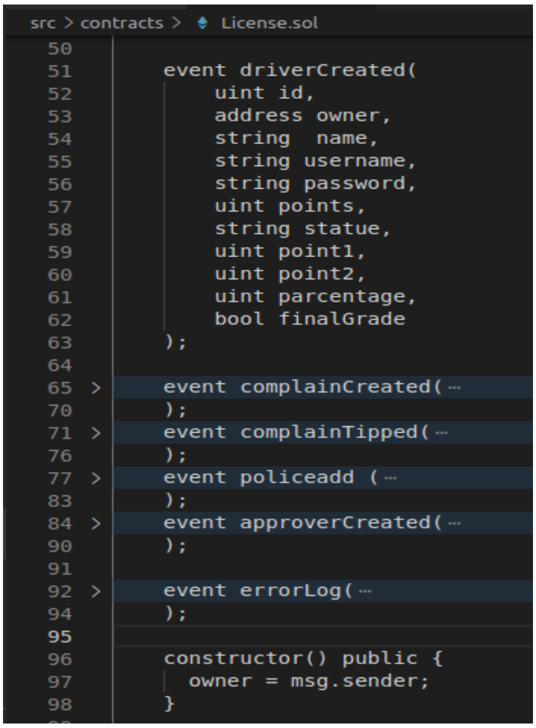
```
    struct driver {
        uint id;
        address owner;
        string  name;
        string username;
        string password;
        uint points;
        string statue;
        uint point1;
        uint point2;
        uint parcentage;
        bool finalGrade;
    }
    struct approver{⋯
    }
    struct police {⋯
    }
    struct complain {⋯
    }
```

Figure 3.17: Struct of project

We created a mapping on Solidity to store driver, approver, complain and police on the blockchain. Mappings have unique keys that return unique values. In our case, we used an id as a key, and the value was a driver struct, approver struct and police struct. This make us essentially to look up a driver by id (have the same work with searchbyid () method) and the same thing with approver, complain and police.

1. **mapping(uint => driver) public drivers;**
2. **mapping(uint => complain) public complains;**
3. **mapping(uint => police) public polices;**
4. **mapping(uint => approver) public approvers;**

The constructor function(Fig 3.18) is a special function that gets called whenever the smart contract is created for the first time, thus, deployed to the blockchain and usually defines the initial contract behavior. Our constructor is defined that the sender's address must be as owner and we set it to msg.sender, which is the represente the global variable of the address that calls the function.Also we created events (Fig 3.18), they are an important concept in smart contracts,wich make every one can listen for this event to verify that a driver, approver, complain and police were created on the blockchain and errorLog informs us that there is an error during registration.



src > contracts > License.sol

```
    event driverCreated(
        uint id,
        address owner,
        string  name,
        string username,
        string password,
        uint points,
        string statue,
        uint point1,
        uint point2,
        uint parcentage,
        bool finalGrade
    );

    event complainCreated(⋯
    );
    event complainTipped(⋯
    );
    event policeadd (⋯
    );
    event approverCreated(⋯
    );

    event errorLog(⋯
    );

    constructor() public {
      owner = msg.sender;
    }
```

Figure 3.18: Events and constructor of project

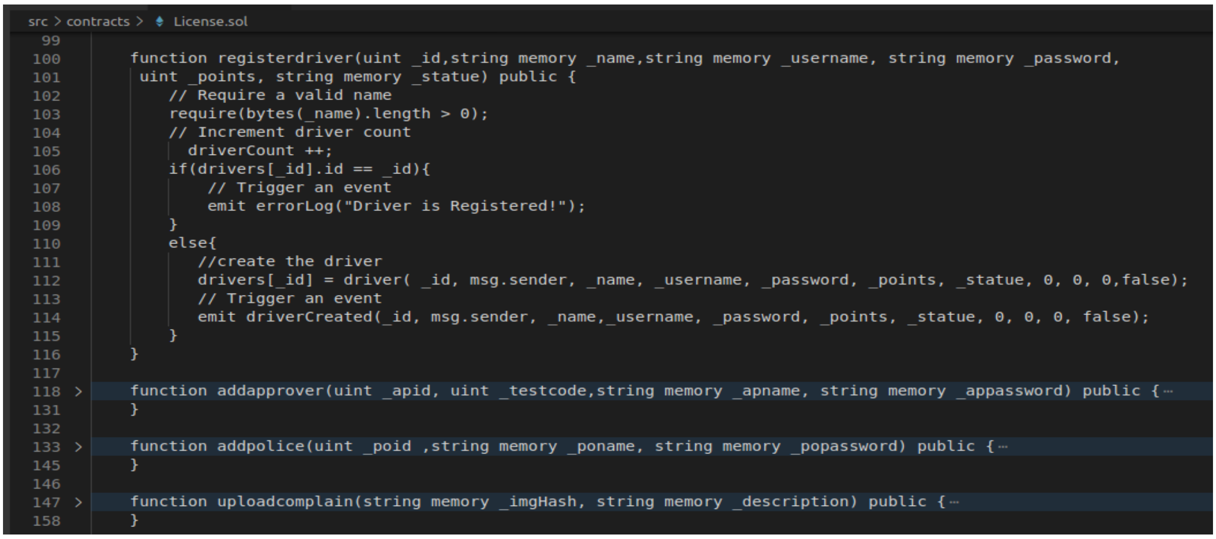
We created a functions for driver’s license management .Let’s explain the work for each one of them:

- **The Register Functions:**

  We set a function registerdriver() (Fig 3.19)that allows a user(learner, driver) to signup which accept six argument. We set the function visibility to public so that it can be called outside of the smart contract, like in the console, or from the client side for example. Next, we add requirement that must be satisfied before the function continues execution. Also we increments the user count to have possibilite to know how mutch users we have in system.

  After that , we put a condition that if id duplicated, trigger an event that this id is already registered . Else we create a new user(learner, driver) and add it to the mapping. Note that msg.sender is the address of the admin accepting the register of user. Finally, we trigger an event to let everyone know that the user(learner, driver) was created sucesfully.

  The same scenario with addapprover(), addpolice(), uploadcomplain() there is a little difference between them lies in difference of attributes of each struct.



src > contracts > License.sol

```
    function registerdriver(uint _id,string memory _name,string memory _username, string memory _password,
     uint _points, string memory _statue) public {
        // Require a valid name
        require(bytes(_name).length > 0);
        // Increment driver count
          driverCount ++;
        if(drivers[_id].id == _id){
            // Trigger an event
            emit errorLog("Driver is Registered!");
        }
        else{
           //create the driver
           drivers[_id] = driver( _id, msg.sender, _name, _username, _password, _points, _statue, 0, 0, 0,false);
           // Trigger an event
           emit driverCreated(_id, msg.sender, _name,_username, _password, _points, _statue, 0, 0, 0, false);
        }
    }

    function addapprover(uint _apid, uint _testcode,string memory _apname, string memory _appassword) public {…
    }

    function addpolice(uint _poid ,string memory _poname, string memory _popassword) public {…
    }

    function uploadcomplain(string memory _imgHash, string memory _description) public {…
    }
```

Figure 3.19: The Register Functions of project

- **The settestresult() Functions:**

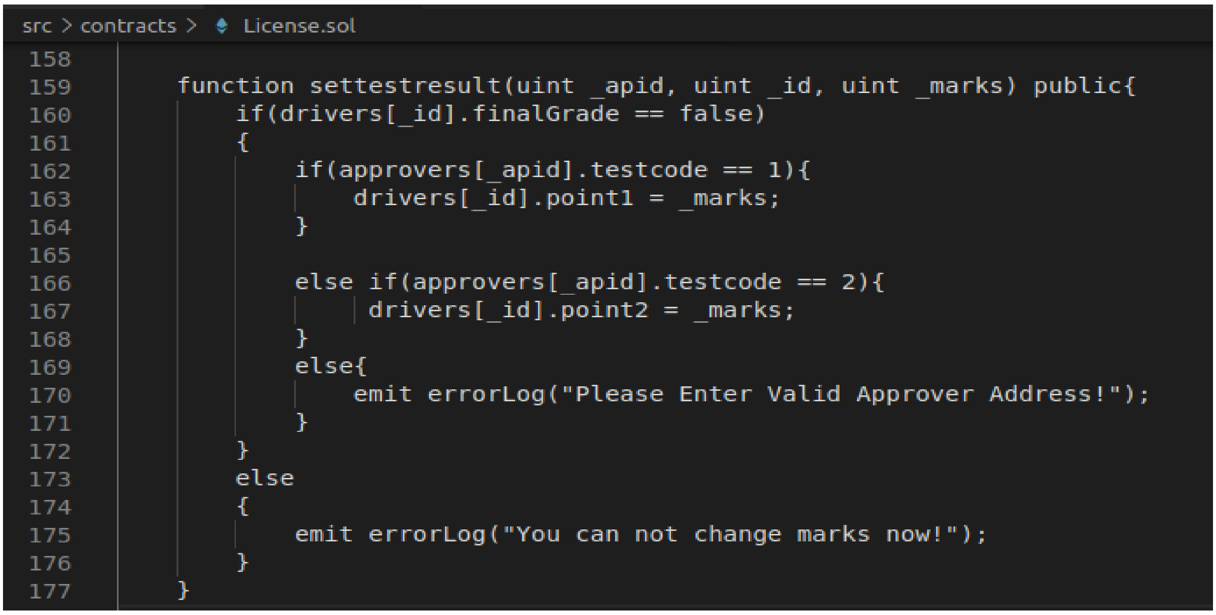
  We set a function settestresult() (Fig 3.20) that allows approver to put learner's test result which accept three argument, they are driver id and approver id and test mark. We set the function visibility to public so that it can be called outside of the smart contract, like in the console, or from the client side for example. After that , we put a condition that if finalGrade was false (that's mean user is not a driver) and the approver test code was 1 or 2, put the mark of learner, else will asks for enter valid approver id. In the case finalGrade was true, we can not change marks.

- **The upgrade() Functions:**

  We set a function upgrade() (Fig 3.21) that allows upgrade learner after he succeed in all tests to be a new driver , which this function accept one argument, it is driver id. We set the function visibility to public so that it can be called outside of the smart contract, like in the console, or from the client side for example. After that , we check the result of learner. Depending on how much his result, he will be a new driver or still learner. If the result was more then 6, we update the learner final Grade to true and also update his statue to new driver and his points to 12.

- **The reducepoints() Functions:**

  We set a function reducepoints() (Fig 3.22) that allowed just for police man to use. This function useful when the police man find a traffic violation from driver.We set the function visibility to public so that it can be called outside



```

    function settestresult(uint _apid, uint _id, uint _marks) public{
        if(drivers[_id].finalGrade == false)
        {
            if(approvers[_apid].testcode == 1){
                drivers[_id].point1 = _marks;
            }

            else if(approvers[_apid].testcode == 2){
                drivers[_id].point2 = _marks;
            }
            else{
                emit errorLog("Please Enter Valid Approver Address!");
            }
        }
        else
        {
            emit errorLog("You can not change marks now!");
        }
    }
```

Figure 3.20: The Set result function of project

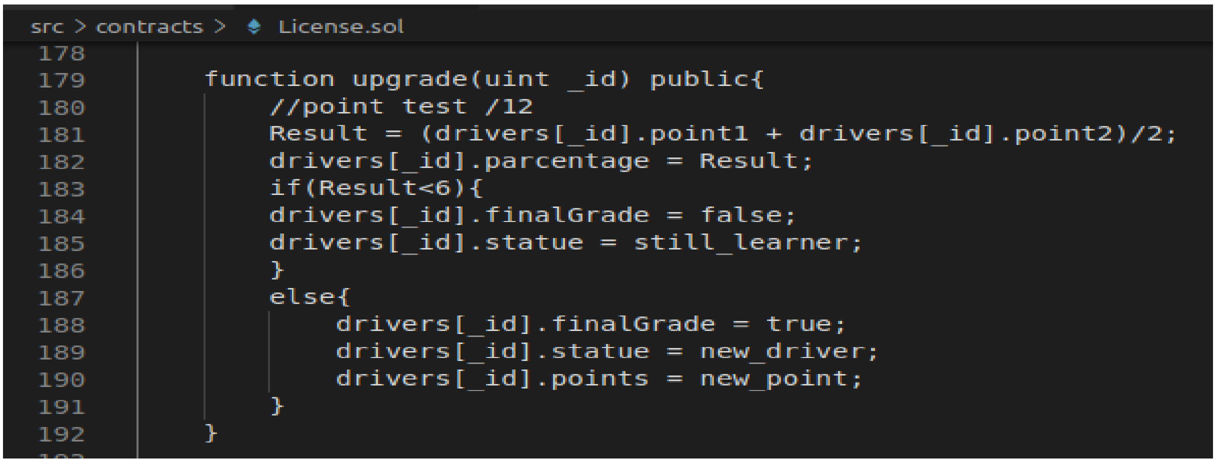
of the smart contract. In the body of function, we initially check whether the total of the points of driver is higher than 0 using **require(drivers[_id].points > 0);**. Depending on that we reduce points from his points depending degree of traffic violation.Afterwards, we update the driver's points in our mapping.

3.4.4 Compiling Smart Contract

We compile the smart contract to make sure that everything worked and there are no errors. After compiling ,a new file was generated **('./src/abis/License.json')**. This file is the smart contract ABI file wich contains a JSON representation of the smart contract functions that can be exposed to external clients, like client-side JavaScript applications.The figure 3.23 shows that we do it successfully.

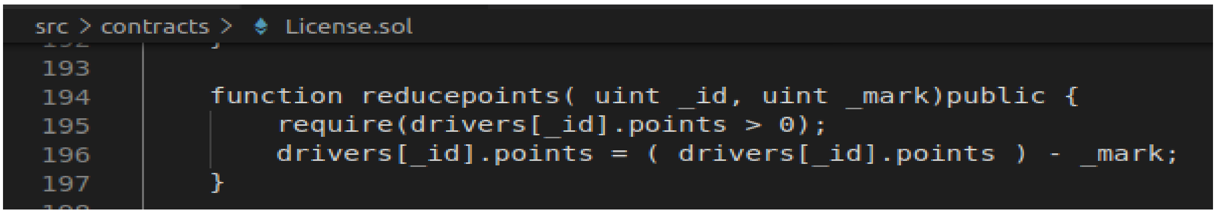
3.4.5 Deploying Smart Contract

We deploy our smart contract to the personal blockchain network by creating a migration script inside the migrations directory meant to alter the state of our application's contracts, moving it from one state to the next by issue **$ truffle migrate** command. In case that we successud the ganache shows that the state of the blockchain has changed and the figure 3.24 representing that.



```
    function upgrade(uint _id) public{
        //point test /12
        Result = (drivers[_id].point1 + drivers[_id].point2)/2;
        drivers[_id].parcentage = Result;
        if(Result<6){
        drivers[_id].finalGrade = false;
        drivers[_id].statue = still_learner;
        }
        else{
            drivers[_id].finalGrade = true;
            drivers[_id].statue = new_driver;
            drivers[_id].points = new_point;
        }
    }
```

Figure 3.21: The up grade function of project



```
    function reducepoints( uint _id, uint _mark)public {
        require(drivers[_id].points > 0);
        drivers[_id].points = ( drivers[_id].points ) - _mark;
    }
```

Figure 3.22: The reduce points function of project

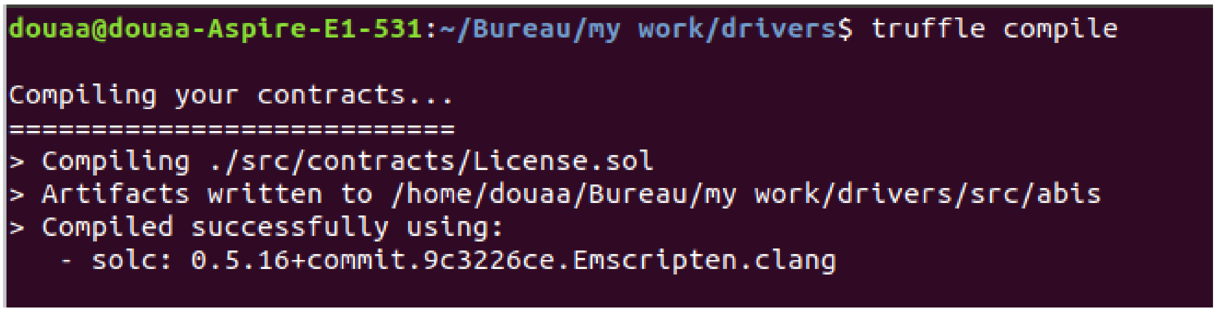
3.4.6 Testing Smart Contract

We test our smart contract to ensure that all function of contract works properly. There is two ways to do that :

First, testing our contract with create a file called License.test.js and creating test using JavaScript to simulate client-side interaction with our smart contract and the figure 3.25 shows the result of some test that we have create. Second, testing our contract with Remix IDE and the figure 3.25 shows the most functions of our smart contract after the deployment using Remix IDE.

3.4.7 Front End

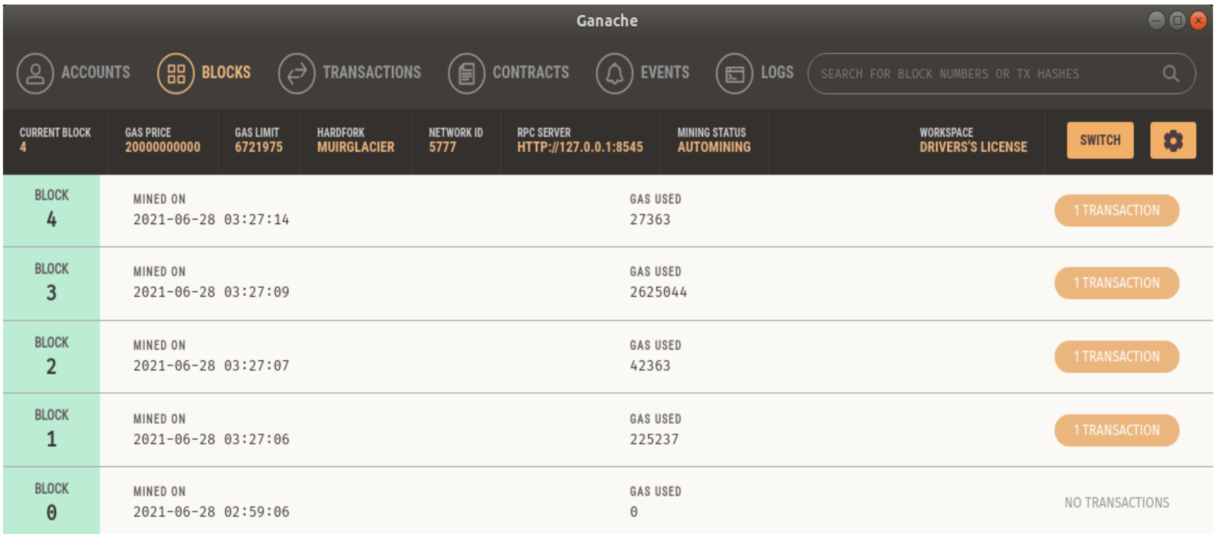
After all past steps that we have done. We build also the client side to interact with driver's license management smart contract.We connected our web browser to the blockchain using Metamask which connected to our Ganache personal blockchain and import some accounts from Ganache into Metamask so that we can act on their



```
douaa@douaa-Aspire-E1-531:~/Bureau/my work/drivers$ truffle compile

Compiling your contracts...
===========================
> Compiling ./src/contracts/License.sol
> Artifacts written to /home/douaa/Bureau/my work/drivers/src/abis
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

Figure 3.23: Result of compiling smart contract



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  HARDFORK     NETWORK ID  RPC SERVER              MINING STATUS   WORKSPACE
4              20000000000  6721975    MUIRGLACIER  5777        HTTP://127.0.0.1:8545   AUTOMINING      DRIVERS'S LICENSE

BLOCK  MINED ON                 GAS USED
4      2021-06-28 03:27:14      27363       1 TRANSACTION
BLOCK  MINED ON                 GAS USED
3      2021-06-28 03:27:09      2625044     1 TRANSACTION
BLOCK  MINED ON                 GAS USED
2      2021-06-28 03:27:07      42363       1 TRANSACTION
BLOCK  MINED ON                 GAS USED
1      2021-06-28 03:27:06      225237      1 TRANSACTION
BLOCK  MINED ON                 GAS USED
0      2021-06-28 02:59:06      0           NO TRANSACTIONS
```

Figure 3.24: Blockchain state after contract deployment

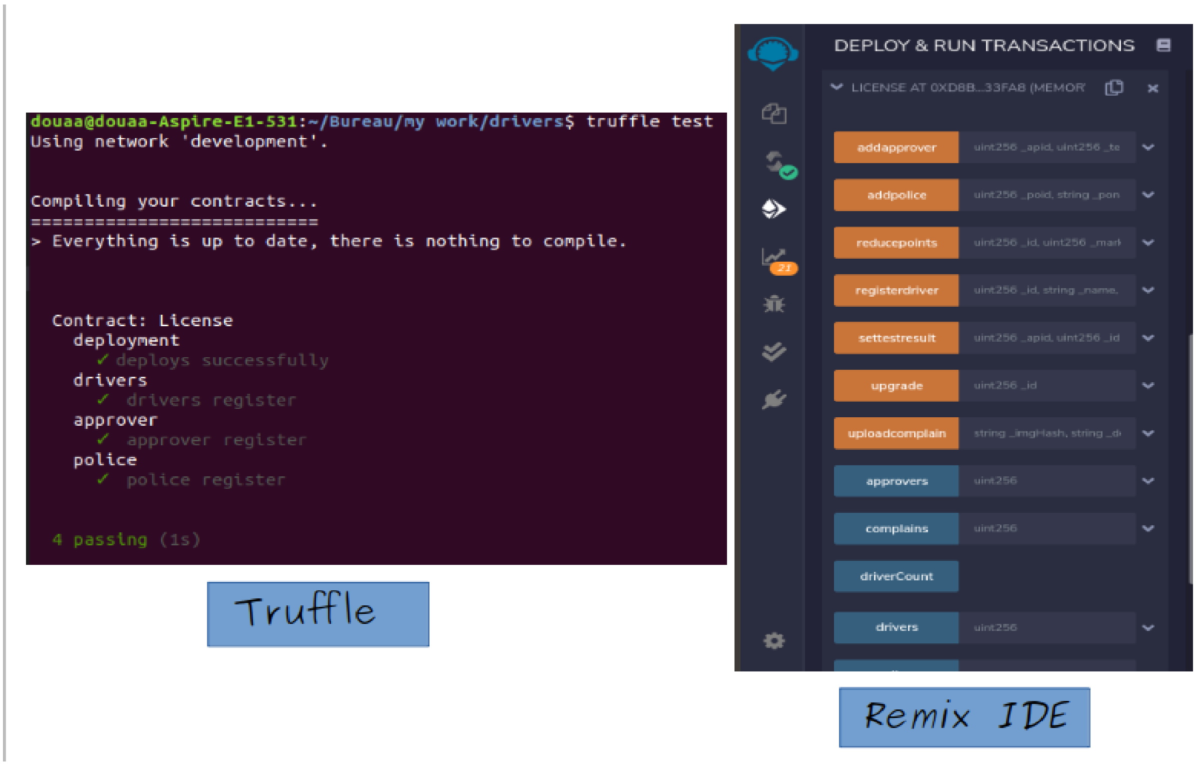
behalf as users of our application.

The main steps to connect our client side application to the blockchain is importing web3 into our App.js and created a new function which it instantiates web3 and function which detects the presence of an Ethereum provider in the web browser(**loadWeb3()**). Also we created a function that loads data from the blockchain named **loadBlockchainData()**.

3.4.8 Presentation of the system interfaces

First, we running our development server by **$ npm run start** to automatically open the website in our browser on home page (Fig 3.26) of system where making the client register and search and login if he alredy signup in system.

Access to the system:



```
douaa@douaa-Aspire-E1-531:~/Bureau/my work/drivers$ truffle test
Using network 'development'.


Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.



  Contract: License
    deployment
      ✓ deploys successfully
    drivers
      ✓ drivers register
    approver
      ✓ approver register
    police
      ✓ police register


  4 passing (1s)
```

Truffle
DEPLOY & RUN TRANSACTIONS
LICENSE AT 0XD8B...33FA8 (MEMOR
addapprover
uint256 _apid, uint256 _te
addpolice
uint256 _poid, string _pon
reducepoints
uint256 _id, uint256 _mark
registerdriver
uint256 _id, string _name,
settestresult
uint256 _apid, uint256 _id
upgrade
uint256 _id
uploadcomplain
string _imgHash, string _d
approvers
uint256
complains
uint256
driverCount
drivers
uint256
21
Remix IDE

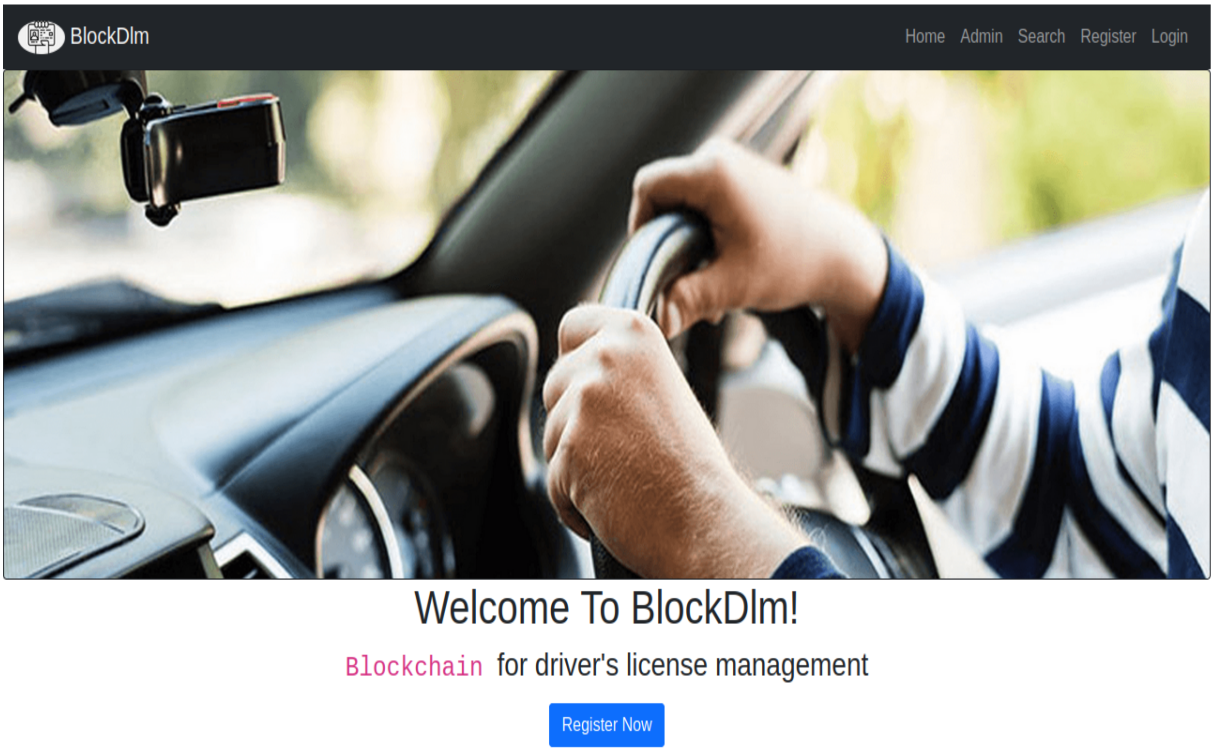
Figure 3.25: Test result with Truffle

When the client want to access to the system, he will have two options:

- Either he already have an account, and in this case he can access to the system by login interface (Fig 3.27), thus, when the enter information is reight, the client will be directed to his profile.

- Either you don't have an account, and therefore you must click on the "Register" in the navbar. That will take you to the registration form .When the client full the form and click on register button , will showing him metamask window to confirm the transaction as shown in Figure 3.29. in case that every thing right, he will receive window of confirmed transaction (Fig 3.28) mean that his account is created, after that he converts to his profile.

Admin and client profiles



Welcome To BlockDlm!

Blockchain for driver's license management

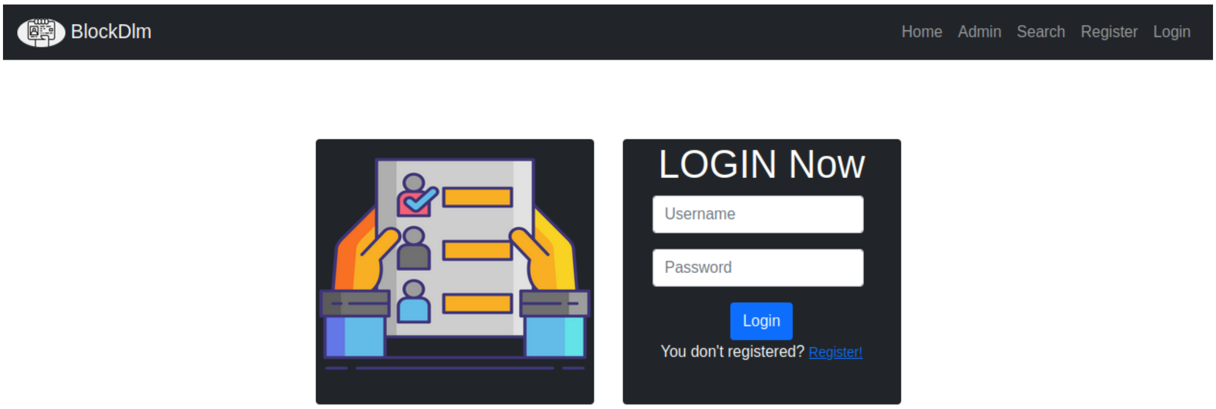
Figure 3.26: Home page

- **Admin profile:**

  If you are an admin, and after you go through the authentication step , you can view your system, can have the list of users , also can have the list of approvers and complain . as the Figure 3.30 shows.

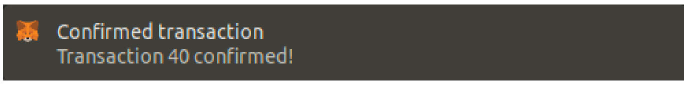
- **Approver profile:**

  After the authentication step, the approver can access to his profile which can upload mark by full the form and click on upload mark where that action form a transaction and the figure 3.33 shows that. Also he can see the liste of users in system and Upgrade user as the figure show.



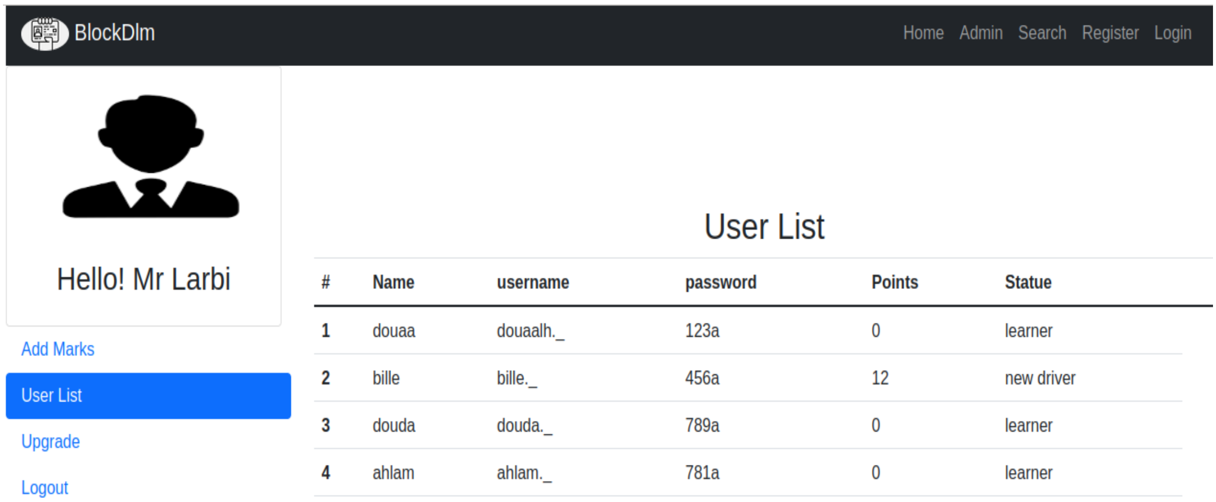
You don't registered? Register!

Figure 3.27: Login page



Confirmed transaction
Transaction 40 confirmed!

Figure 3.28: confirmed transaction

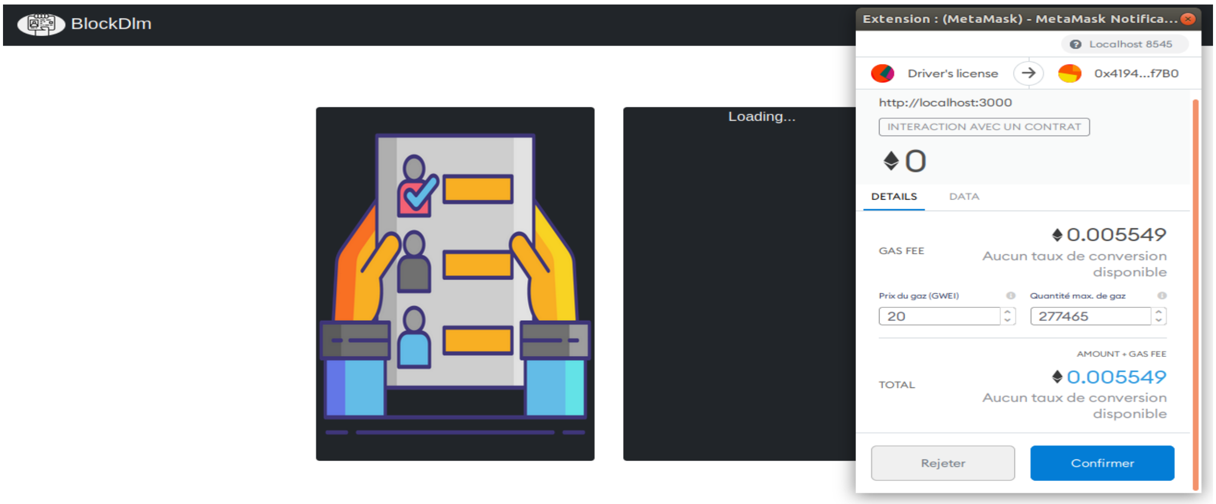


Hello! Mr Larbi

User List

#	Name	username	password	Points	Statue
1	douaa	douaalh._	123a	0	learner
2	bille	bille._	456a	12	new driver
3	douda	douda._	789a	0	learner
4	ahlam	ahlam._	781a	0	learner

Figure 3.31: User list in approver profile



BlockDlm

Loading...

Extension : (MetaMask) - MetaMask Notifica...

Localhost 8545

Driver's license → 0x4194...f7B0

http://localhost:3000

INTERACTION AVEC UN CONTRAT

♦0

DETAILS DATA

GAS FEE ♦0.005549 Aucun taux de conversion disponible

Prix du gaz (GWEI): 20

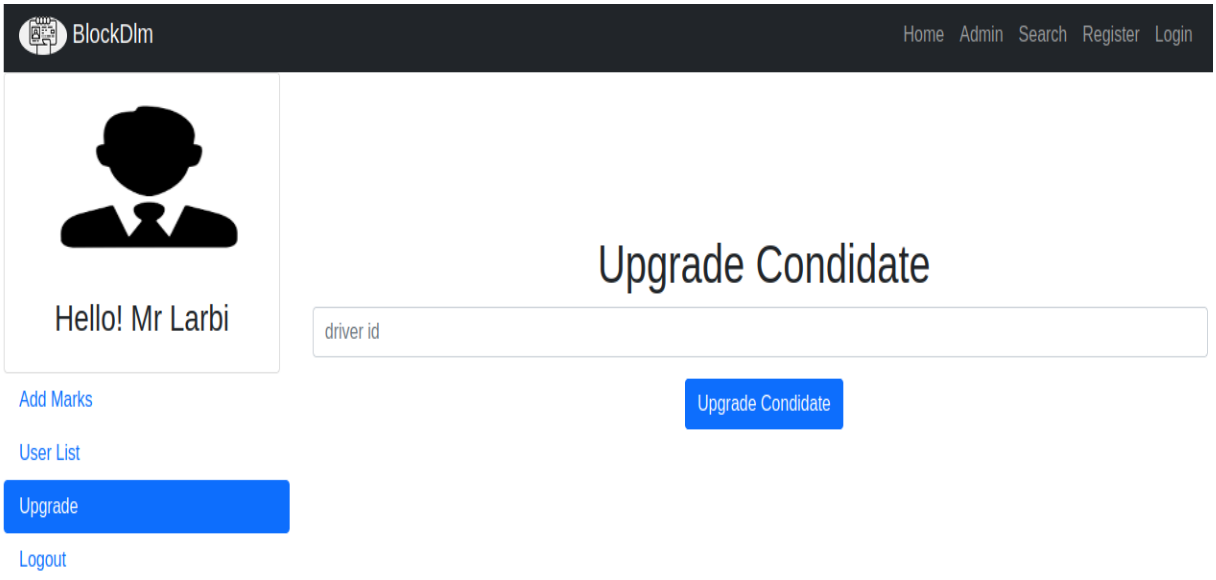
Quantité max. de gaz: 277465

AMOUNT + GAS FEE

TOTAL ♦0.005549 Aucun taux de conversion disponible

Rejeter Confirmer

Figure 3.29: Register to system



BlockDlm

Home Admin Search Register Login

Hello! Mr Larbi

Add Marks

User List

Upgrade

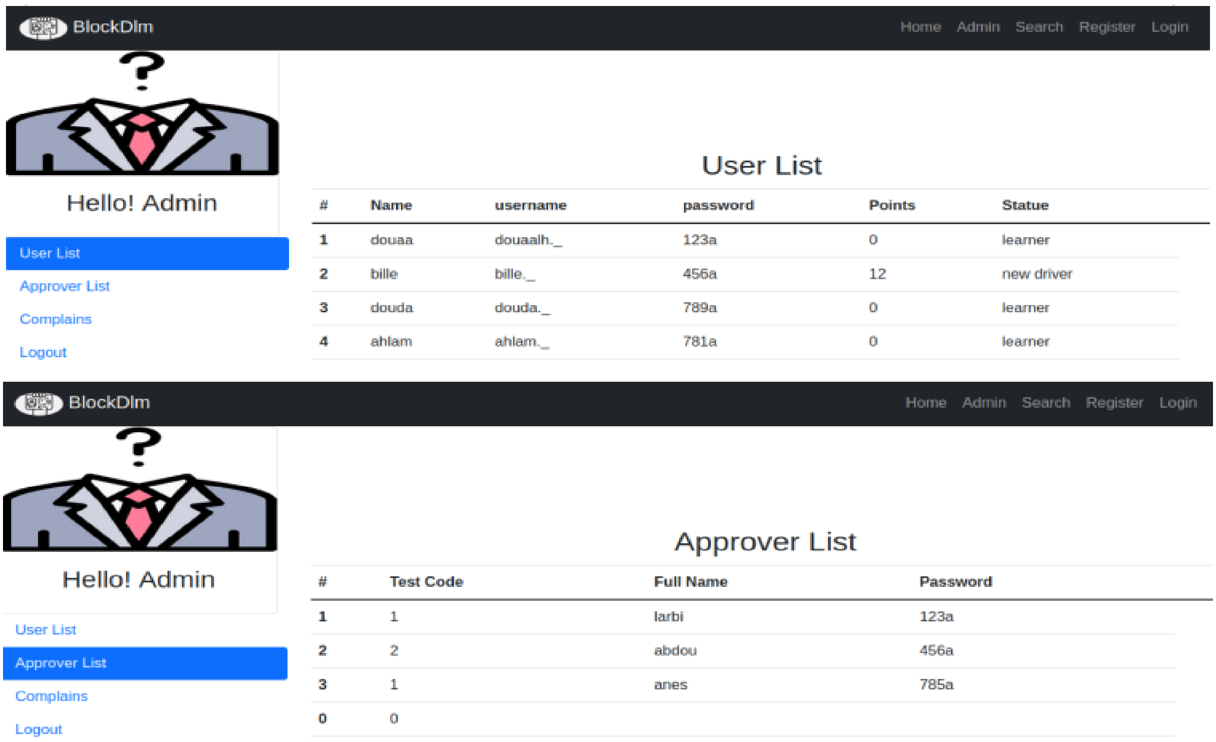
Logout

Upgrade Condidate

driver id

Upgrade Condidate

Figure 3.32: Approver upgrade condidate



User List

#	Name	username	password	Points	Statue
1	douaa	douaalh._	123a	0	learner
2	bille	bille._	456a	12	new driver
3	douda	douda._	789a	0	learner
4	ahlam	ahlam._	781a	0	learner

Approver List

#	Test Code	Full Name	Password
1	1	larbi	123a
2	2	abdou	456a
3	1	anes	785a
0	0		

Figure 3.30: Admin profile

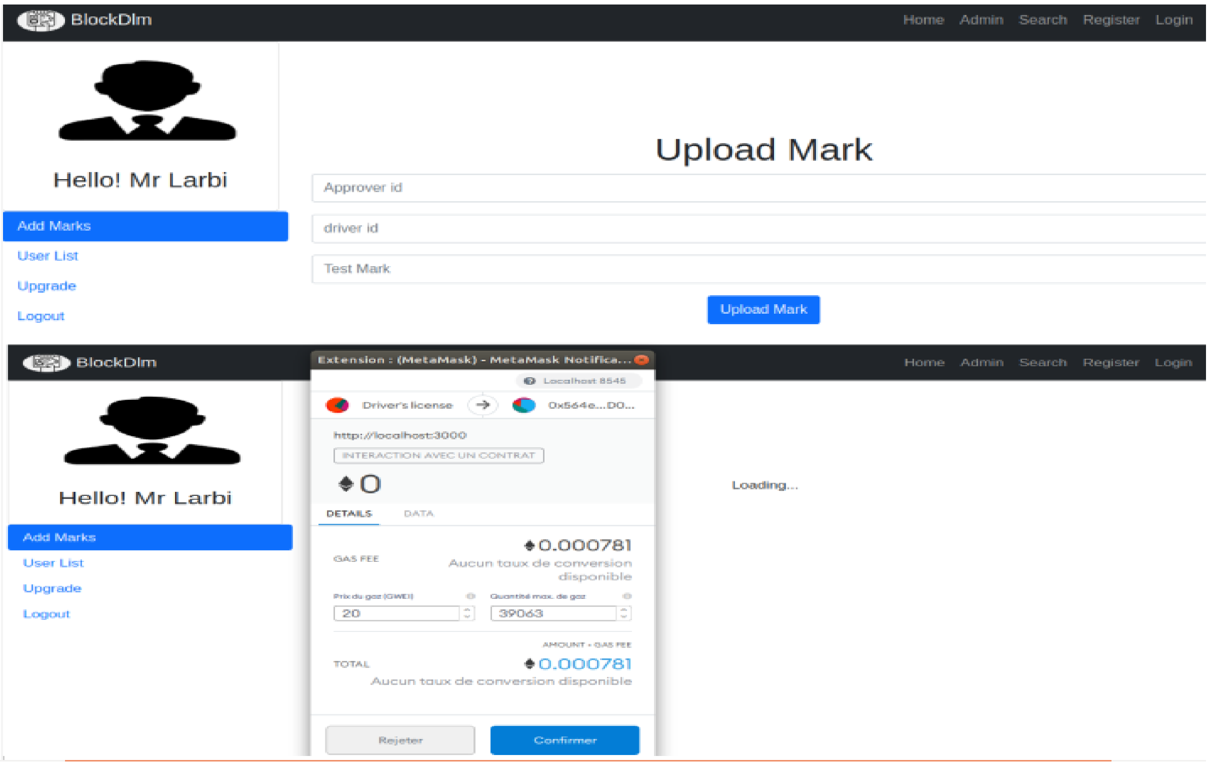
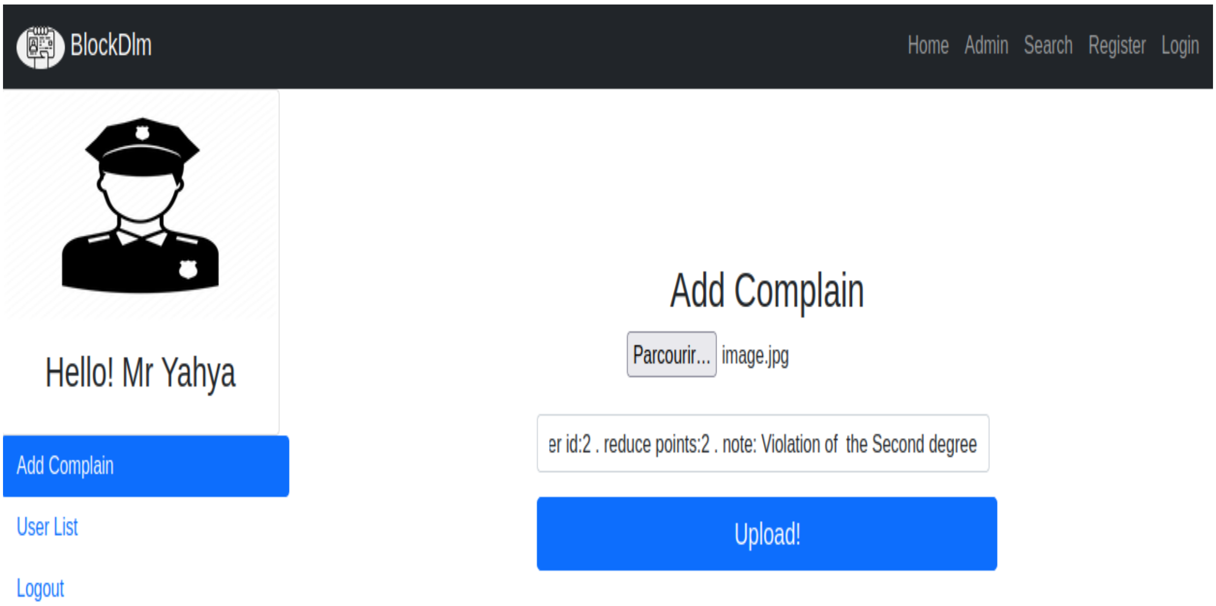


Figure 3.33: Approver upload mark

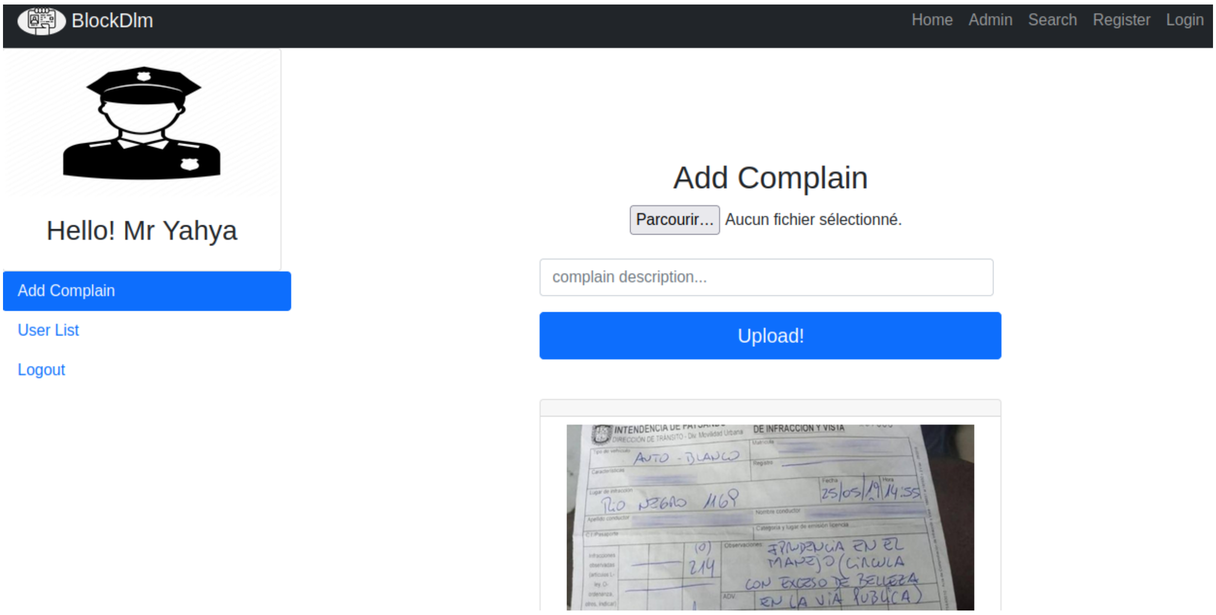
- **Police profile:**

  The police access to his profile which can add complain with picture and description,this action of click on upload button form a transaction and the figure 3.34 shows that . Also he can see the liste of users in system and has a botton of logout.



er id:2 . reduce points:2 . note: Violation of the Second degree

Figure 3.34: Police profile



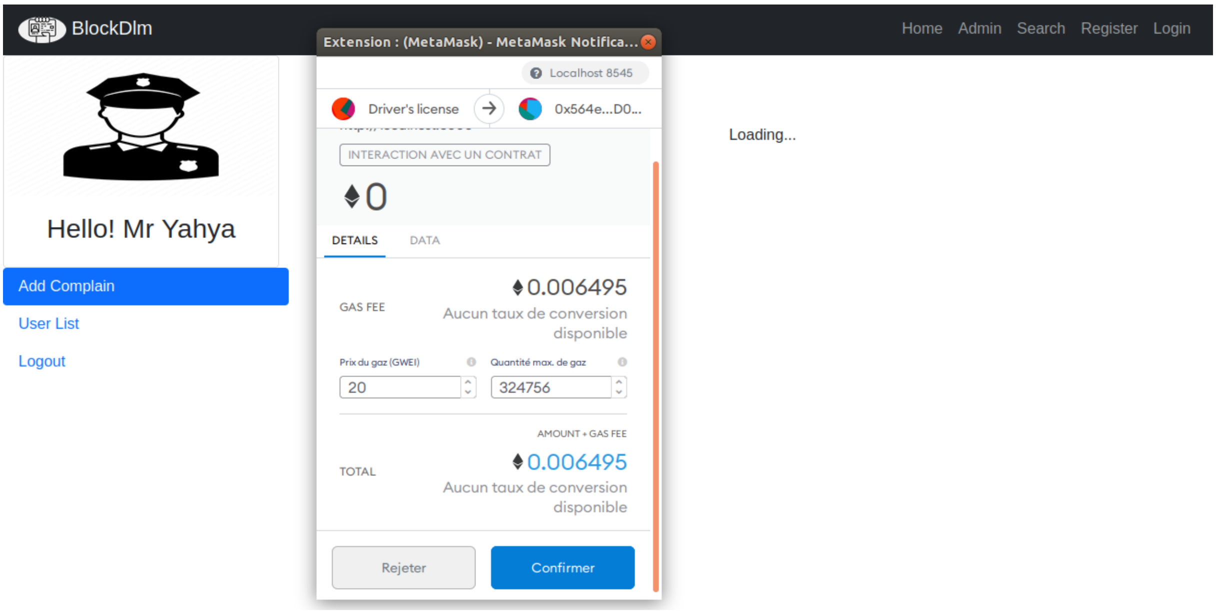
Hello! Mr Yahya

Add Complain

Aucun fichier sélectionné.

complain description...

Figure 3.35: Result of transaction



Hello! Mr Yahya

Extension : (MetaMask) - MetaMask Notifica...

Localhost 8545

Driver's license → 0x564e...D0...

INTERACTION AVEC UN CONTRAT

0

GAS FEE 0.006495
Aucun taux de conversion disponible

Prix du gaz (GWEI): 20

Quantité max. de gaz: 324756

AMOUNT + GAS FEE

TOTAL 0.006495
Aucun taux de conversion disponible

Loading...

Figure 3.36: Add complain

3.5 Conclusion

In this chapter, we introduced the design of our system (global and detailed architecture), by adopting a Blockchain for using driver's license, and we shows actors and their interactions in the system.Also we introduced the development tools we used as well as the most important interfaces of our application through a set of screenshots.

General Conclusion

The promising innovative technology of the Blockchain that has proven its effectiveness in the field of security and decentralization in various sectors has made reliance on the use of data collected from traditional systems unpopular as there is no guarantee that the data will not be tampered in some way.

In this work, our solution to the driver's license management problem addresses (like the lack of integrity of users and service providers) using blockchain technology:

- Transparency that ensures that they brings multiple benefits to all entities in the system .
- Trace-ability which is the main feature to avoid forgery and dishonesty , it enables us to find out who is eligible from whom not.

These two characteristics are among the most prominent characteristics of this technology, which also has many benefits.

We can cite points among the strengths of this work that using blockchain technology, the information will be present and not falsifiable as long as the blockchain exists and also since the blockchain is a large distributed record that is copied in several places this makes it impossible to lose the driver's license and all the recorded information in system.

Bibliography

[1] Documentation for visual studio code. `https://code.visualstudio.com/docs`. [Online; accessed 13-Juin-2021].

[2] Ganache overview. `https://www.trufflesuite.com/docs/ganache/overview`. [Online; accessed 13-Juin-2021].

[3] Node.js - introduction. `https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm`. [Online; accessed 13-Juin-2021].

[4] web3.js - ethereum javascript api. `https://web3js.readthedocs.io/en/v1.3.4/index.html`. [Online; accessed 13-Juin-2021].

[5] Blockchain. `https://www.ionos.fr/digitalguide/web-marketing/vendre-sur-internet/blockchain/`, 2018. [Online; accessed 15-Juin-2021].

[6] Zuaiter Asim. *Mechanisms to reduce traffic accidents according to Law 05-17*. PhD thesis, University Mohamed Boudiaf - M'sila, 2018. [Translated from Arabic].

[7] Imran Bashir. *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained*. Packt Publishing Ltd, 2018.

[8] Ayoub BILEM and Ilyes BRAHAMI. *Application de Gestion des retraits de permis de conduire*. PhD thesis.

[9] Georges Bell Bitjoka, Pierre Bilong, and Moses Macaire Nnanga Edoa. Implementation of the blockchain in the optimization of the security of transport documents (driver's license and vehicules registration cards). *American Journal of Computer Science and Technology*, 3(3):57–67, 2020.

[10] Kevin Curran. E-voting on the blockchain. *The Journal of the British Blockchain Association*, 1(2):4451, 2018.

[11] Elad Elrom. *The Blockchain Developer*. Springer, 2019.

[12] Julija Golosova and Andrejs Romanovs. The advantages and disadvantages of the blockchain technology. In *2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE)*, pages 1–6. IEEE, 2018.

[13] Ori Jacobovitz. Blockchain for identity management. *The Lynne and William Frankel Center for Computer Science Department of Computer Science. Ben-Gurion University, Beer Sheva*, 2016.

[14] Philippe Marrast. Blockchain: Éléments d'explication et de vulgarisation, pourquoi s' intéresser à la blockchain aujourd'hui? In *Blockchain et Santé: Perspectives d'applications et enjeux juridiques (Séminaire IFERISS)*, 2018.

[15] Jeffrey Owens. Blockchain 101 for governments. *Vienna: Wilton Park. Retrieved from https://www. wiltonpark. org. uk/wp*, 2017.

[16] Reza M Parizi, Ali Dehghantanha, et al. Smart contract programming languages on blockchains: An empirical evaluation of usability and security. In *International Conference on Blockchain*, pages 75–91. Springer, 2018.

[17] Marion PIGNEL and Denis STOKKINK. La technologie blockchain une opportunité pour l'économie sociale?

[18] Lewis Popovski, George Soussou, and PB Webb. A brief history of blockchain. *Patterson Belknap Webb & Tyler, New York, NY, USA, Tech. Rep*, 2014.

[19] Sara Saberi, Mahtab Kouhizadeh, Joseph Sarkis, and Lejia Shen. Blockchain technology and its relationships to sustainable supply chain management. *International Journal of Production Research*, 57(7):2117–2135, 2019.

[20] Mayank Sahu. Cryptography in blockchain: Types & applications. `https://www.upgrad.com/blog/cryptography-in-blockchain/`, 2021. [Online; accessed 4-Jan-2021].

[21] Bikramaditya Singhal, Gautam Dhameja, and Priyansu Sekhar Panda. *Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions*. Springer, 2018.

[22] Gautam Srivastava, Reza M Parizi, Ali Dehghantanha, and Kim-Kwang Raymond Choo. Data sharing and privacy for patient iot devices using blockchain. In *International Conference on Smart City and Informatization*, pages 334–348. Springer, 2019.

[23] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. *arXiv preprint arXiv:1906.11078*, 2019.