LISBON SCHOOL OF ENGINEERING

Department of Electronical Engineering, Telecommunications and Computers

Ticket management system using Blockchain technology

Rodrigo Filipe Leitão Dias

Bachelor's

Project Work to obtain the masters degree
in Informatics and Multimedia Engineering

Adviser : PhD Carlos Gonçalves

Jury:

President: [Grau e Nome do presidente do juri]

Vogal: [Grau e Nome do primeiro vogal]

September, 2024

LISBON SCHOOL OF ENGINEERING

Department of Electronical Engineering, Telecommunications and Computers

Ticket management system using Blockchain technology

Rodrigo Filipe Leitão Dias

Bachelor's

Project Work to obtain the masters degree
in Informatics and Multimedia Engineering

Adviser : PhD Carlos Gonçalves

Jury:

President: [Grau e Nome do presidente do juri]

Vogal: [Grau e Nome do primeiro vogal]

September, 2024

Abstract

Resumo

Contents

Contents 9

List of Figures 11

List of Tables 13

1 Introduction 15

1.1 Motivation 16

1.2 Objectives 17

1.3 Contributions 17

1.4 Document Structure 17

2 Background and Related Work 19

2.1 Background 19

2.1.1 Interacting with the Blockchain 20

2.1.2 Blockchain 21

2.1.3 Wallets 22

2.1.4 Networks 24

2.1.5 Smart Contracts 25

2.1.6 Token Standards 26

2.1.7 Non-Fungible Tokens (NFTs) 28

2.2 Related Work 28
2.2.1 Traditional Ticket Selling Platforms 28
2.2.2 Application of NFTs 28

3 Requirements Analysis **29**
3.1 Use Cases 29
3.1.1 System owner use cases 30
3.1.2 Organizer use cases 30
3.1.3 Validator use cases 31
3.1.4 User use cases 31
3.2 Functional Requirements 32
3.3 Non-Functional Requirements 33

4 Implementation **35**
4.1 Project Features 35

5 Results **37**
5.1 Limitations 37
5.2 Future Work 37

List of Figures

2.1 Blockchain concepts 20
2.2 How does a blockchain work 21
2.3 Token standards 26

3.1 System owner use cases 30
3.2 Organizer use cases 31
3.3 Validator use cases 31
3.4 User use cases 32

List of Tables

1

Introduction

Concerts and festivals play a big role in peoples life's, allowing them to create memorable experiences watching live performances from their favorite artists. Those are the kinds of memories that last for life, so every event organizer wants to ensure that the whole process works seamlessly, from the end user to the entire background planning of the event.

The process of organizing an event starts with the event organizer, who is responsible for the whole planning of the event, from the venue to the artists, to the marketing and ticketing. The event organizer is the one who takes the risk of organizing the event, and the one who will profit from it, and can be a company, a group of people, or even a single person, whom will hire the artists, the venue, the security, the marketing, and the ticketing.

There is an issue with the ticketing process, which is the scalping. Scalping is the process of buying tickets in bulk, exploiting high demand, and reselling them at significantly inflated prices. This not only disadvantages people that genuinely want to attend but also undermines the integrity of the ticketing system. Traditional ticketing platforms often rely on centralized databases and intermediaries, providing opportunities for scalpers to manipulate the system and engage in fraudulent activities. Moreover, existing ticketing systems frequently encounter issues related to security, trust, and reliability. Centralized databases are vulnerable to cyber attacks, leading to unauthorized access, data breaches, and the manipulation of ticketing information. Trust in the authenticity of tickets and the reliability of transactions is compromised, creating a

pressing need for innovative solutions that can address these inherent challenges.

That's where blockchain comes in. Blockchain technology, renowned for its decentralized and transparent nature, presents a compelling solution to revolutionize the ticketing industry. By leveraging blockchain, it becomes possible to create a secure and tamper-proof ledger of transactions, mitigating the risk of scalping and ensuring the integrity of the ticketing process. The use of smart contracts further automates transactions, reducing the reliance on intermediaries, therefore extra costs, and enhancing operational efficiency. This allows the event organizer to have a more secure and reliable ticketing process, and the end user to have a more transparent and fair ticketing process.

1.1 Motivation

The motivation for this work is primarily to avoid ticket scalping. By using blockchain, it's possible to create a system that prevents any kind of unwanted operations because of the properties of smart contracts that enforce a certain behavior, allowing for users to resell a ticket, but not for a price higher than the original price. This is a way to guarantee that the end user will have a fair and transparent ticketing process. For this to happen, the idea is to have a marketplace where users are able to resell their tickets, enforcing a price cap on them.

Another important feature is the possibility of having partial refunds. This is something that is not always available in traditional ticketing platforms, and when it is, it's not easy to do. With blockchain, it's possible to have a system that allows for easy and instant refunds, without the need for intermediaries. Enabling a feature like this can actually be advantageous for the organizers, if they're expecting the venue to be full. This way, when users buy their tickets and then realize they can't attend, they have a reason to refund, making that ticket available again for the original price, making a profit for the organizer.

Another point, and the main reason to use blockchain, is to guarantee the user of any operation. In the traditional ticketing system, there can be human errors, or even fraud, and this assures users that any operation defined will never change and will be executed as expected.

1.2 Objectives

We will have a website that will allow Ticketchain to approve organizers into our system to be able create events, so that no random entity can create freely, which would lead to spam. This would also be for the event organizers themselves where, if allowed, they would be able to manage their events, and manage admins and validators associated to them.

There will also be an app for the users to check the events and manage their tickets. The events shown are gonna be the ones stored in our Ticketchain system. They will also have access to the marketplace, where they will be able to resell their tickets for a price no higher than the original one.

For the validators, we will have yet another app that allows them to validate user tickets at the entrance of the venue. These validators are selected by the organizer for the event, and their job is to truthfully verify the tickets, without any chance of fraud. These validators can be either a person with their app to operate or even some automated machine, like a turnstile.

1.3 Contributions

[website] [user app] [validator app]

1.4 Document Structure

2

Background and Related Work

This chapter will present the background and related work for the system. The background section will present the necessary background information for the system. The related work section will present projects with similarities to the system.

2.1 Background

Blockchain is a fairly recent technology and can a lot of times be hard to understand. It leverages cryptographic concepts at its core to make it all work. In this section, we will explain some of the key concepts of blockchain technology and also discuss how wallets work and how they interact with the blockchain. We will then explore some of the most popular blockchain networks and token standards, as well as the emerging trend of non-fungible tokens (NFTs). The Figure 2.1 illustrates the most common concepts of blockchain technology and we'll be mentioning the Wallets (1) in the Subsection 2.1.3, the Networks (2) in the Subsection 2.1.4 and the Smart Contracts (3) in the Subsection 2.1.5.

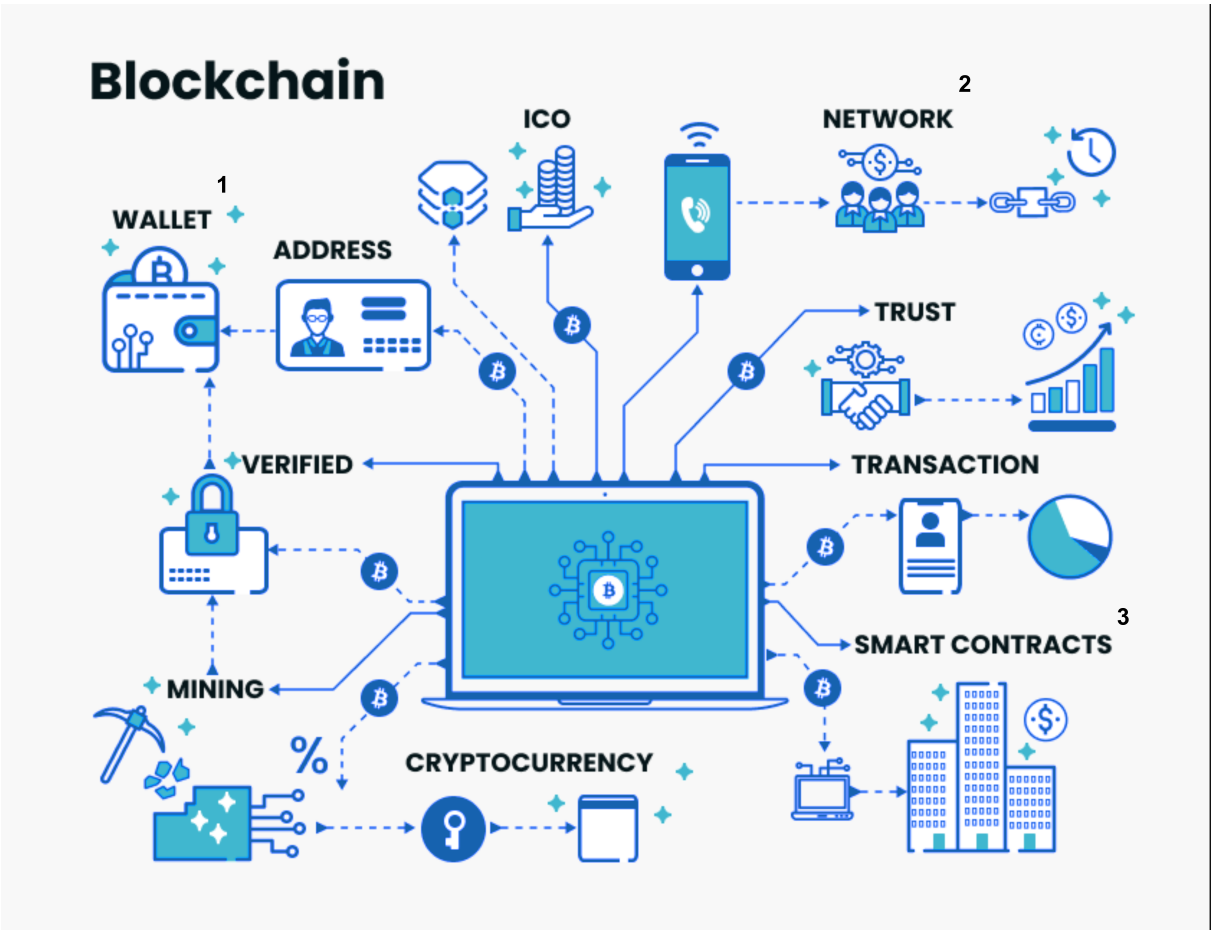

Figure 2.1: Blockchain concepts. Adapted from ...

2.1.1 Interacting with the Blockchain

As we saw, a blockchain works as a decentralized network of nodes, where each node has a copy of the entire blockchain. This means that in order to interact with the blockchain, we need to send transactions to the network, which will then be validated and added to a block by the nodes. To do this, we need to use a wallet, which is an application that allows users to manage their digital assets, interact with smart contracts, and send transactions on the blockchain. Wallets provide a user-friendly interface for accessing the blockchain network, signing transactions with private keys, and viewing account balances and transaction history. To execute a transaction on the blockchain, we need to sign it with our private key, which proves that we are the rightful owner of the assets being transferred. The transaction is then broadcast to the network, where it is validated by network participants and added to a block. Once the transaction is confirmed and included in a block, it becomes part of the immutable blockchain ledger, visible to all participants in the network. This process is ilustrated in the Figure 2.2.

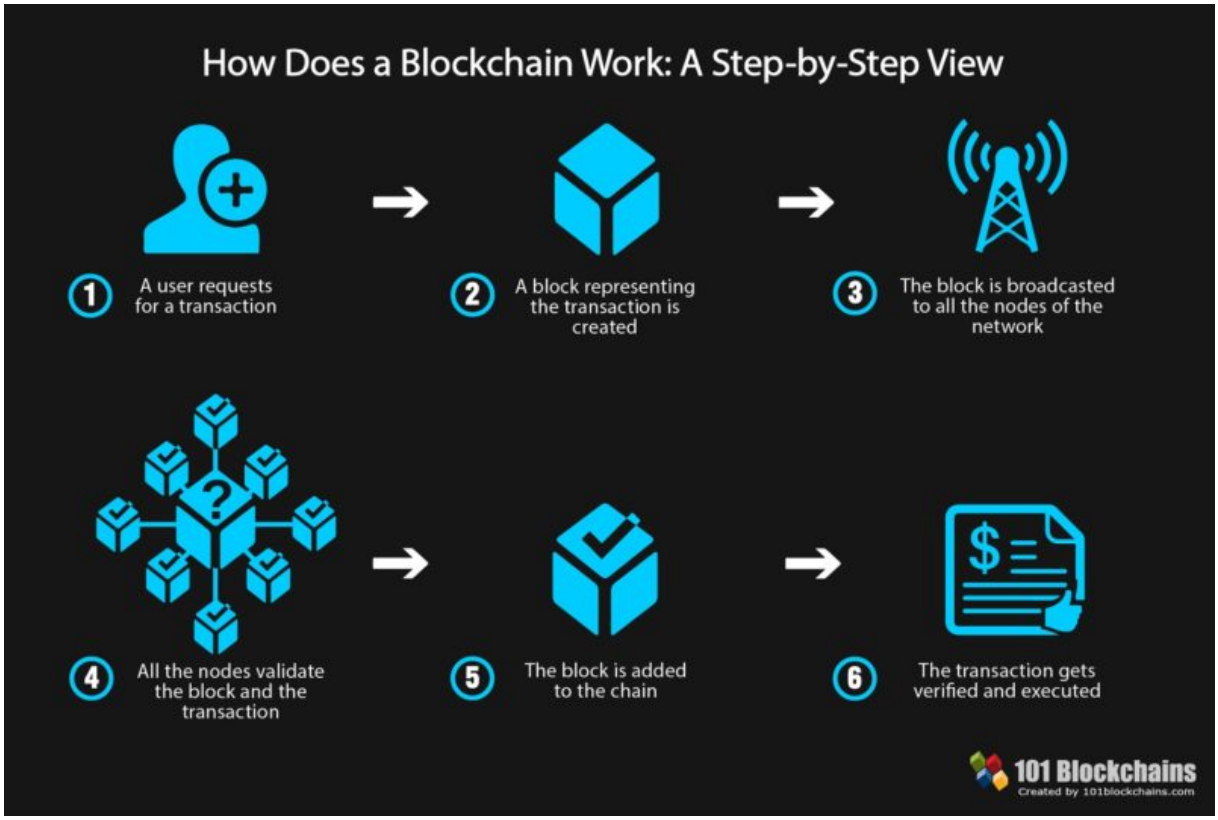

Figure 2.2: How does a blockchain work. Extracted from ...

2.1.2 Blockchain

So blockchain is a decentralized and distributed ledger technology that enables the secure recording and sharing of data across a network of computers. At its core, a blockchain consists of a series of blocks, each containing a list of transactions. These blocks are linked together in a chronological and immutable chain, forming a transparent and tamper-proof record of transactions.

Key characteristics of blockchain technology include:

- **Decentralization**: Unlike traditional centralized systems where data is stored in a single location or controlled by a central authority, blockchain operates on a decentralized network of computers (nodes). Each node maintains a copy of the entire blockchain, ensuring that there is no single point of failure or control.
- **Transparency**: The data recorded on a blockchain is visible to all participants in the network. This transparency fosters trust among users, as they can independently verify the integrity of transactions and the state of the ledger without

relying on intermediaries.

- **Immutability**: Once a transaction is recorded on the blockchain and added to a block, it becomes virtually impossible to alter or delete. This immutability is achieved through cryptographic techniques such as hashing and consensus mechanisms, ensuring that the historical record of transactions remains tamper-proof.
- **Security**: Blockchain technology employs advanced cryptographic algorithms to secure transactions and protect the integrity of the network. Transactions are verified and validated by network participants through a process known as consensus, which prevents fraudulent or unauthorized changes to the ledger.
- **Smart Contracts**: Smart contracts are self-executing contracts with predefined rules and conditions written in code. These contracts automate the execution of transactions and enforce agreements without the need for intermediaries. Smart contracts enable the creation of decentralized applications (DApps) that run on blockchain networks, facilitating a wide range of use cases beyond simple monetary transactions.

Blockchain technology has applications across various industries, including finance, supply chain management, healthcare, and decentralized finance (DeFi). Its potential to revolutionize existing systems by enhancing security, transparency, and efficiency has led to widespread adoption and exploration of its capabilities in solving complex challenges. Some people refer to this ecosystem as Web3, which is a new paradigm for the internet that aims to decentralize control and empower users with greater ownership and privacy over their data and digital assets. A great discussion topic as for how a system like this works is because it leverages the human nature of greed and self-interest to create a system that is secure and reliable. The network of nodes is incentivized to maintain the integrity of the blockchain by rewarding them with cryptocurrency for their efforts. As long as these cryptocurrencies have value, this creates a system where the majority of the network is honest and works together to maintain the integrity of the blockchain, making it resistant to attacks and fraud, as the cost of attacking the network would far outweigh any potential gains.

2.1.3 Wallets

Cryptocurrency wallets rely on cryptographic principles to securely manage and interact with digital assets on blockchain networks. These cryptographic techniques ensure

the security and integrity of transactions while protecting the private keys that control access to cryptocurrency holdings.

Some of the key cryptographic aspects of wallets are:

- **Private and Public Keys**: Cryptocurrency wallets utilize a pair of cryptographic keys: a public key and a private key. The public key, also known as the wallet address, is used to receive funds and is shared publicly. The private key, on the other hand, is known only to the wallet owner and is used to sign transactions and authorize the spending of funds. The relationship between the public key and the private key is based on asymmetric cryptography, where data encrypted with one key can only be decrypted with the other key. This ensures that transactions are secure and that only the rightful owner of the private key can access and control their cryptocurrency holdings.
- **Digital Signatures**: When a transaction is initiated from a cryptocurrency wallet, it is digitally signed using the wallet's private key. This digital signature serves as proof of authorization and ensures that the transaction cannot be tampered with or altered. Digital signatures are generated using cryptographic algorithms such as the Elliptic Curve Digital Signature Algorithm (ECDSA) or the Rivest-Shamir-Adleman (RSA) algorithm, depending on the specific blockchain network and protocol.
- **Hash Functions**: Cryptocurrency wallets use cryptographic hash functions to create a unique representation of transaction data, known as a transaction hash. These hash functions generate fixed-length strings of characters from input data, making it computationally infeasible to reverse-engineer the original data from the hash. Transaction hashes are essential for verifying the integrity of transactions and ensuring that they have not been altered or tampered with during transmission.
- **Seed Phrases and Mnemonic Codes**: Some cryptocurrency wallets use mnemonic codes or seed phrases as a backup mechanism for restoring access to wallet funds in case the original private key is lost or compromised. These seed phrases are generated from a random sequence of words and serve as a human-readable representation of the wallet's private key. They can be used to regenerate the private key and restore access to funds on a new wallet instance.

By leveraging these cryptographic techniques, cryptocurrency wallets provide a secure and reliable means for users to store, manage, and transact with digital assets on

blockchain networks. The robustness of these cryptographic mechanisms ensures the confidentiality, integrity, and authenticity of transactions, safeguarding the value of cryptocurrency holdings against unauthorized access and fraudulent activities.

2.1.4 Networks

This technology has evolved significantly since the inception of Bitcoin in 2009. Numerous platforms have emerged, each offering unique features, capabilities, and use cases.

Some of the most prominent networks that have gained traction in the decentralized ecosystem are:

- **Bitcoin (BTC):** Bitcoin is the first and most well-known cryptocurrency, introduced by an anonymous person or group of people under the pseudonym Satoshi Nakamoto in 2008. It operates on a decentralized network using a Proof of Work (PoW) consensus mechanism to validate transactions and secure the network. Bitcoin is designed as a peer-to-peer electronic cash system, enabling users to send and receive payments without the need for intermediaries. It has gained widespread adoption as a store of value and digital currency, with a fixed supply of 21 million coins and a deflationary monetary policy.

- **Ethereum (ETH)**: Ethereum is a decentralized, open-source blockchain platform that enables the creation and execution of smart contracts and decentralized applications (DApps). It introduced the concept of smart contracts, allowing developers to build a wide range of decentralized applications, from decentralized finance (DeFi) protocols to non-fungible token (NFT) marketplaces. Ethereum operates on a Proof of Work (PoW) consensus mechanism but is transitioning to a Proof of Stake (PoS) consensus model with the Ethereum 2.0 upgrade to improve scalability and energy efficiency.

- **Polygon (MATIC)**: Polygon is a Layer 2 scaling solution for Ethereum, designed to address the network's scalability issues by offering faster and cheaper transactions. It provides a framework for building and connecting Ethereum-compatible blockchain networks, known as sidechains, which leverage the security of the Ethereum mainnet. Polygon aims to enhance Ethereum's capabilities and support the mass adoption of

- **Solana (SOL)**: Solana is a high-performance blockchain platform designed for decentralized applications and crypto-currencies. It uses a unique combination of Proof of History (PoH) and Proof of Stake (PoS) consensus mechanisms to achieve high throughput and low latency, enabling fast transaction speeds and low fees. Solana aims to provide a scalable infrastructure for decentralized finance (DeFi), decentralized exchanges (DEXs), and other high-performance applications.

These are just a few examples of the diverse range of blockchain networks that exist, each offering unique features, capabilities, and use cases. As the blockchain ecosystem continues to evolve, new networks and technologies are constantly being developed, driving innovation and expanding the possibilities of decentralized applications and digital assets.

2.1.5 Smart Contracts

Smart contracts are self-executing contracts with the terms of the agreement directly written in code. These contracts automatically execute and enforce themselves when predefined conditions are met, without the need for intermediaries such as lawyers or notaries. Smart contracts run on blockchain platforms and are stored and executed across a decentralized network of nodes.

Key characteristics of smart contracts include:

- **Autonomy**: Once deployed on the blockchain, smart contracts operate autonomously, executing transactions and enforcing agreements without human intervention. This autonomy ensures that contract terms are upheld impartially and transparently.

- **Trust**: Smart contracts leverage the trustless nature of blockchain technology, meaning that parties can trust the execution of the contract without relying on a trusted third party. The decentralized and immutable nature of blockchain ensures that contract terms are tamper-proof and transparent.

- **Security**: Smart contracts are highly secure due to the cryptographic principles underlying blockchain technology. Once deployed, smart contracts cannot be altered or tampered with, providing a high level of security and reliability.

- **Efficiency**: By automating contract execution, smart contracts eliminate the need for intermediaries, reducing costs and processing times associated with traditional contract enforcement. Transactions are executed quickly and efficiently, enhancing the overall speed and efficiency of business processes.
- **Versatility**: Smart contracts can be programmed to execute a wide range of functions beyond simple transaction processing. They can facilitate complex conditional agreements, manage digital assets, and even interact with other smart contracts, enabling the development of decentralized applications (DApps) with diverse functionalities.

Smart contracts have numerous applications across various industries, including finance, supply chain management, real estate, healthcare, and more. They are particularly well-suited for scenarios where trust, transparency, and automation are paramount, offering a revolutionary approach to contract execution and enforcement in the digital age.

2.1.6 Token Standards

Token standards play a crucial role in defining the rules and functionalities of digital tokens on blockchain networks. These standards provide a common framework that facilitates interoperability, compatibility, and ease of use for developers and users alike.

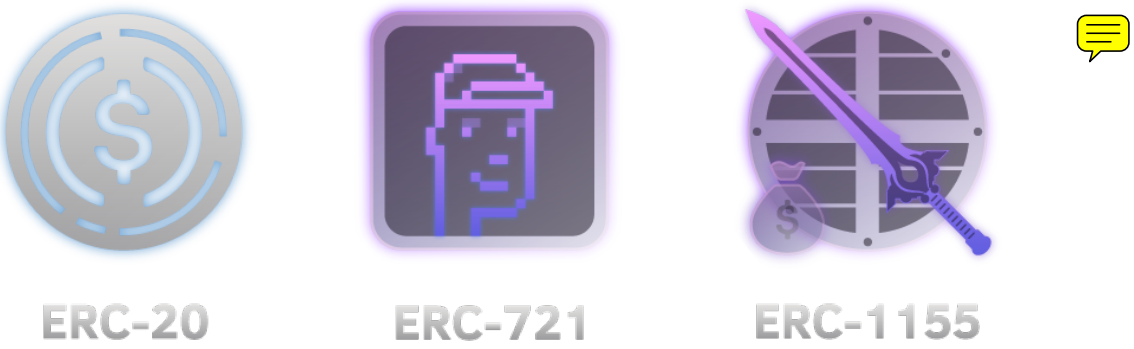

Figure 2.3: Token standards. Extracted from ...

Some of the most widely recognized token standards in the blockchain ecosystem are:

- **ERC-20 (Ethereum Request for Comment 20)**: ERC-20 is the most commonly used token standard on the Ethereum blockchain, governing the creation and implementation of fungible tokens. These tokens are interchangeable and have identical properties, allowing them to be traded on cryptocurrency exchanges seamlessly. ERC-20 tokens adhere to a set of standard functions, including methods for transferring tokens, querying token balances, and approving token transfers on behalf of other addresses. Many of the initial coin offerings (ICOs), token sales, and decentralized finance (DeFi) projects on Ethereum utilize ERC-20 tokens due to their widespread adoption and compatibility with Ethereum wallets and exchanges.

- **ERC-721 (Ethereum Request for Comment 721)**: ERC-721 is a token standard on the Ethereum blockchain that governs the creation and implementation of non-fungible tokens (NFTs). Unlike ERC-20 tokens, each ERC-721 token is unique and indivisible, representing ownership or proof of authenticity of a specific asset. ERC-721 tokens are commonly used to represent digital assets such as digital art, collectibles, virtual real estate, and in-game items. Each token is assigned a unique identifier (token ID), allowing it to be distinguished from other tokens within the same contract. The ERC-721 standard defines methods for transferring tokens, querying token ownership, and managing metadata associated with each token, enabling a wide range of use cases in the burgeoning NFT market.

- **ERC-1155 (Ethereum Request for Comment 1155)**: ERC-1155 is a token standard on the Ethereum blockchain that supports the creation and management of both fungible and non-fungible tokens within the same contract. This allows developers to efficiently manage multiple token types and reduce gas costs associated with deploying multiple contracts. ERC-1155 tokens are highly flexible and versatile, making them suitable for a wide range of applications, including gaming, digital collectibles, and decentralized finance (DeFi). They provide developers with the ability to create tokenized assets with varying degrees of uniqueness and scarcity. The ERC-1155 standard defines methods for transferring tokens, querying token balances, and managing batch transfers of multiple token types, offering enhanced functionality compared to previous token standards.

These token standards represent just a few examples of the diverse range of standards shaping the landscape of tokenization on blockchain networks. As blockchain technology continues to evolve, new standards are likely to emerge, offering innovative solutions and driving further adoption of digital tokens across various industries.

2.1.7 Non-Fungible Tokens (NFTs)

Since we're talking about a ticketing system for events, we can see a lot of potential in the use of NFTs to represent digital tickets, providing a secure and verifiable means of ticket issuance, transfer, and validation. NFT-based tickets can be associated with unique metadata, such as event details, seat numbers, and access permissions, providing a rich and customizable ticketing experience for event organizers and attendees. NFTs can also be used to create limited edition or VIP tickets, offering exclusive access and additional benefits to holders of these special tickets.

2.2 Related Work

2.2.1 Traditional Ticket Selling Platforms

There are many different traditional ticket selling platforms that are used today. Ticketline and Blueticket are the largest and most reputable Portuguese companies specializing in ticket sales for a variety of events, including concerts, sports games, theater productions, and exhibitions. They also have a wide range of physical outlets, like Worten, Fnac, and El Corte Inglés, making it convenient to buy tickets in person.

They have a website where they list all the events organizers are selling tickets for. The user can then open the event they are interested in and check all the details about it and it allows for the user to buy the tickets online and print them at home, or directs them to a physical outlet where they can buy them in person.

2.2.2 Application of NFTs

A few of the most popular applications of NFTs are in the art and gaming industries. In the art industry, NFTs are used to create digital art pieces that are unique and can be bought and sold. In the gaming industry, NFTs are used to create unique in-game items that can be bought, sold, and traded between players. The most popular project is the Bored Ape Yacht Club which is the famous ape *jpegs* that have been sold for millions of dollars. The project has a community of people who own these apes and they are used to allow access to exclusive events and merchandise.

3

Requirements Analysis

This chapter will present the requirements analysis for the system. The requirements analysis is divided into three sections: use cases, functional requirements, and non-functional requirements. The use cases section will present the use cases for the system, divided into four categories: system owner, organizer, validator, and user. The functional requirements section will present the functional requirements for the system. The non-functional requirements section will present the non-functional requirements for the system.

3.1 Use Cases

This section will present the use cases for the system, divided into four actors: system owner, organizer, validator, and user. All of them have similar use case, the authenticate use case, which is responsible for authenticating the users on the system. On the Subsection 3.1.1 we have the use cases for the system owner, where it mentions the management of the system settings and the control of the organizers that have access to the system. On the Subsection 3.1.2 we have the use cases for the organizer that mention the creation and management of the events, along with the control of the validators for the event. On the Subsection 3.1.3 we have the use cases for the validator, and the only thing he can do is to validate the users' tickets. Lastly, on the Subsection 3.1.4 we have the use cases for the common user, like the purchase, gift, refund and resell of tickets.

3.1.1 System owner use cases

As we can see in the Figure 3.1, the system owner has the specific use case to control event organizers. This is important because the he needs to have control over the organizers that have access to the system. The system owner can also manage the system settings, which is important to control the system's behavior and to adapt it to the organizers' needs.

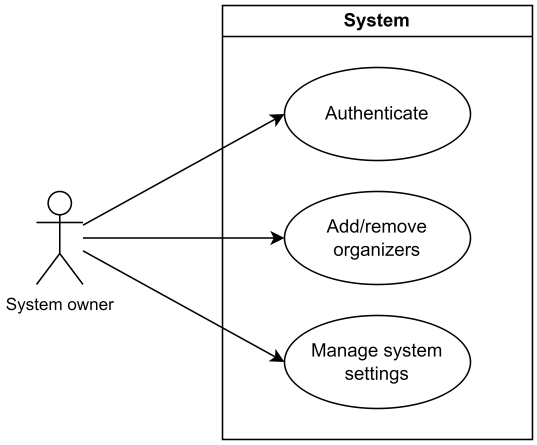

Figure 3.1: System owner use cases

3.1.2 Organizer use cases

The organizer has the use cases to create events, as we can see in the Figure 3.2. He can also control the validators for the event, which is important to select the people that have the authority to validate the tickets for each event. The organizer can also manage the event settings, like updating the event information or cancel it if really needed.

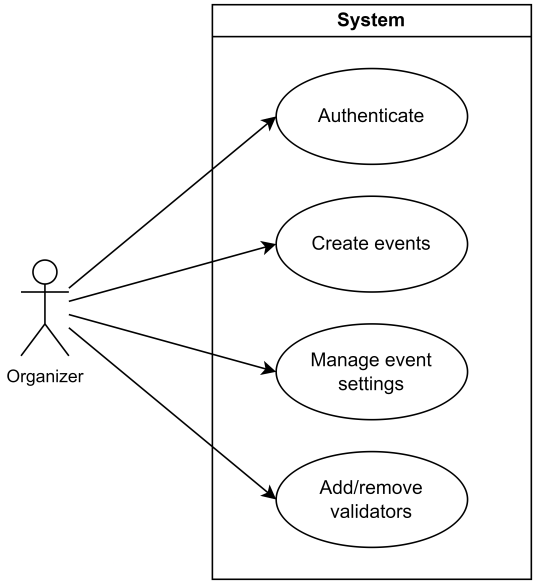

Figure 3.2: Organizer use cases

3.1.3 Validator use cases

For the validators, as we see in the Figure 3.3, the only use case is to validate the users' tickets and to allow them to enter the event. This is a necessary step to avoid users to try to bypass this security measure and to ensure that only the users that have a valid ticket can enter the event.

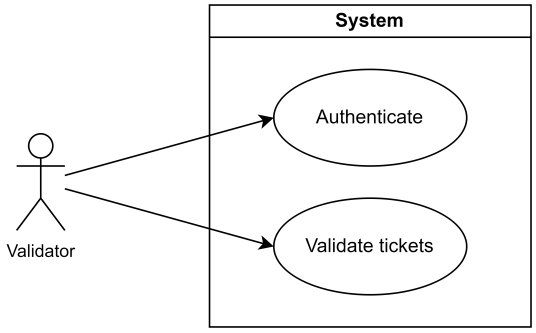

Figure 3.3: Validator use cases

3.1.4 User use cases

In the Figure 3.4 we see the use cases for the common user. The user can purchase tickets for the events, gift tickets to other users, refund tickets if he doesn't want to go

to the event anymore (depending on the configuration of the specific event), and resell tickets if he wants to sell them to other users (with the guarantee that he can't sell at a higher price than the original). All of these use cases are important to give the user the flexibility to manage his tickets and have the freedom to do what he wants with them, within the system's rules, of course

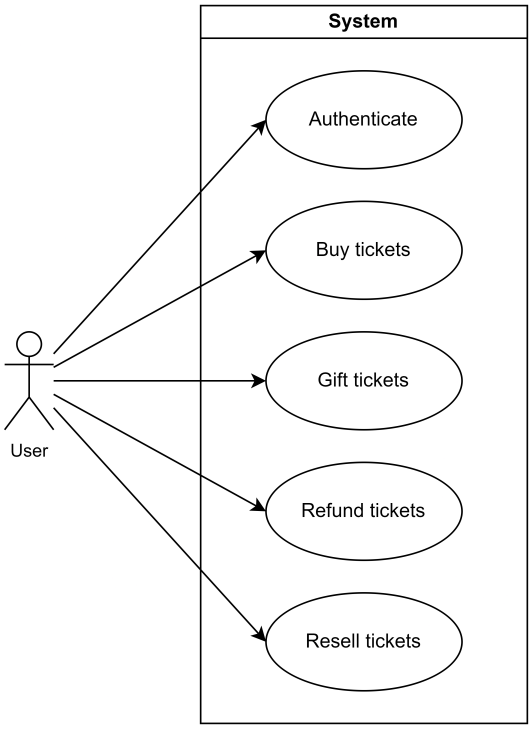

Figure 3.4: User use cases

3.2 Functional Requirements

The functional requirements are the features that the system must have to fulfill the user's needs.

The system must have the following functional requirements:

- **Connect to a wallet software**: The system must be able to connect to a wallet software to interact with the blockchain and to allow the validation of the ownership of the tickets.

- **Interact with the smart contract**: The system has to interact with the smart contract to display the necessary information and to update the state of any event or ticket.

3.3 Non-Functional Requirements

The non-functional requirements are the features that the system must have to fulfill the user's needs, but that are not directly related to the system's functionality.

The system must have the following non-functional requirements:

- **Scalable**: The system must be able to handle a large number of users and events.
- **Low fees**: The system must have low fees for any kind of operation.
- **Fast**: The system must be fast to allow events to have the smoothest experience possible.
- **Secure**: The system must be secure to avoid any kind of fraud.
- **User-friendly**: The system must be user-friendly to allow users to easily buy and sell tickets.

The scalable, low fees and fast requirements are essencially associated with the blockchain network choice. As we saw in the Subsection 2.1.4, different networks have different characteristics and we need to choose the one that best fits our needs. The secure and user-friendly requirements are associated with the system's design and implementation. We need to design the system in a way that is secure, to avoid any kind of fraud, and have a smooth experience for the users.

4

Implementation

[Brief introduction of the chapter]

[todo mention network fees and network choice]

4.1 Project Features

5

Results

5.1 Limitations

5.2 Future Work