SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER SCIENCE

Etherlotto

BLOCKCHAIN AND DISTRIBUTED LEDGER TECHNOLOGIES

Professors:
Claudio Di Ciccio

Students:
Andre Datchev
Alex Lacueva
Srinjan Ghosh
Mursal Furqan Kumbhar
Waddah Alhajar

Academic Year 2023/2024

Contents

1 Preface

In the realm of decentralized finance (DeFi), EtherLotto stands as a pioneer, leveraging blockchain technology to revolutionize traditional lotteries. This report provides an in-depth exploration of EtherLotto, covering its inception, blockchain fundamentals, software architecture, implementation details, and future prospects.

1.1 EtherLotto - Reinventing Lotteries with Blockchain

EtherLotto is a decentralized application (DApp) redefining the lottery experience by leveraging the transparency, fairness, and security of blockchain technology. Players participate directly through smart contracts, ensuring a tamper-resistant and automated process. EtherLotto embodies the future of lotteries—secure, transparent, and globally accessible.

1.2 Team Members and Main Responsibilities

- **Waddah Alhajar:** Smart contract coding, presentation and report
- **Alex Lacueva:** Presentation and BackEnd
- **Andre Datchev:** BackEnd testing and Report
- **Mursal Furqan Kumbhar:** FrontEnd and UI
- **Srinjan Ghosh:** Frontend and Backend

1.3 Outline

This report delves into the development and implementation of EtherLotto, a groundbreaking DApp transforming lotteries through blockchain. Beginning with an exploration of blockchain fundamentals, the report navigates through the application domain, context, software architecture, and detailed implementation, addressing challenges and future considerations.

2 Background

2.1 Introduction

Blockchain technology has emerged as a transformative force, revolutionizing the way we think about digital transactions and decentralized systems. Its origins can be traced back to the introduction of Bitcoin in 2008 by an individual or group under the pseudonym Satoshi Nakamoto. Bitcoin's primary goal, as outlined in the seminal Bitcoin white paper, was to create a decentralized digital currency that operated on a peer-to-peer network, eliminating the need for central authorities like banks. This groundbreaking concept, detailed in Nakamoto's paper [1], laid the foundation for a new era of digital finance and decentralized applications. A blockchain is an open, decentralized, and distributed ledger that records transactions between two parties in a verifiable and permanent way. The term "blockchain" is derived from its structure, where transactions are grouped into blocks, and each block is linked to the previous one through a cryptographic hash, forming a continuous chain. Each block contains the history of every block that came before it with timestamped transaction data down to the second. Bitcoin's success paved the way for the development of Ethereum, a decentralized blockchain platform founded by Vitalik Buterin in 2015. Ethereum introduced the groundbreaking concept of smart contracts, which are self-executing programs that automate agreements. These contracts run on Ethereum's blockchain, specifically on the Ethereum Virtual Machine (EVM) [2]. Ethereum's blockchain initially operated on a Proof of Work (PoW) consensus mechanism, where miners competed to solve complex cryptographic puzzles to validate transactions and add blocks to the chain [3]. However, Ethereum transitioned to Proof of Stake (PoS) in 2022, a more energy-efficient and scalable consensus mechanism. In PoS, validators are chosen based on the amount of cryptocurrency they hold and are willing to stake, taking turns proposing and validating new blocks [4]. The process for consensus in Ethereum's PoS is two-staged: First, a new block can be proposed at each slot and the new head of the chain is voted through the LMD-GHOST algorithm by a pool of pseudo-randomly selected validators. Then, after a block gets finalized through Casper-FFG [5] votes, it is very expensive (and subject to slashing) to try to rewrite history. This shift to PoS aims to improve energy efficiency and scalability, addressing various concerns associated with PoW such as reducing its electricity consumption and environmental impact.

This transition towards PoS, and the smart contracts on Ethereum, has paved the way for the development of Web3.0 and Decentralized Applications (DApps): applications operating with a blockchain in the back-end, removing the need for centralization.

Blockchains can be categorized into three types: public, private, and consortium blockchains. Public blockchains, such as Bitcoin and Ethereum, are open to anyone and

are maintained by a decentralized network of nodes. Private blockchains are restricted to a specific organization or group and are often used for internal record-keeping. Consortium blockchains are a hybrid model where multiple organizations collaborate on a shared ledger, often used in industries such as finance and supply chain management [6, 7].

Blockchain technology offers several key features that make it a powerful tool for various applications:

• Decentralization: Blockchain eliminates the need for a central authority, empowering users with direct control over their data and assets [8].

• Transparency: All transactions are recorded on the blockchain, making them publicly auditable and increasing trust and accountability.

• Immutability: Once a transaction is recorded on the blockchain, it cannot be altered or deleted, ensuring data integrity and security.

• Security: Blockchain uses cryptography to secure transactions and prevent tampering, making it highly resistant to fraud and manipulation [9].

The Ethereum Virtual Machine (EVM) is a crucial component of the Ethereum blockchain. It's a virtual machine that executes smart contracts written in the Solidity programming language. The EVM ensures that smart contracts are executed consistently and securely across the entire Ethereum network [10]. The combination of blockchain technology, smart contracts, and the EVM has opened up a world of possibilities for decentralized applications (DApps) in various sectors, including finance, supply chain management, healthcare, and social networks.

VRFs are cryptographic functions that generate verifiable random numbers, meaning that the randomness of the output can be mathematically proven. This ensures that the results cannot be manipulated or influenced by any party, including the system operator. VRFs are essential for applications like lotteries, where trust in the randomness of the draw is paramount[11]. To enhance the fairness of lottery draws, EtherLotto plans to integrate Chainlink VRF (Verifiable Random Function). This robust randomness solution ensures that the winner selection is verifiable,unpredictable, and tamper-proof, further reinforcing the commitment to a fair and unbiased lottery experience. Verifiable Random Functions (VRFs) are a crucial component in ensuring fairness and transparency in blockchain applications, particularly those involving randomness, such as lotteries.

EtherLotto's conceptualization arises from a fundamental question: In a world where cryptocurrencies are the norm, how can smart contracts redefine traditional activities? The answer emerged when the Christmas lottery approached—an ideal opportunity to employ Ethereum's capabilities for a decentralized lottery. The vision was to create a platform that leverages the transparency, security, and automation inherent in blockchain and smart contracts.

2.2 Smart Contracts in Action

Central to EtherLotto's functionality are smart contracts—a single, robust lottery contract that governs the entire lifecycle of each lottery instance. These contracts, deployed on the Ethereum blockchain, manage the creation, participation, and conclusion of lotteries. Each smart contract instance can be customized with unique parameters, such as prize pool, ticket price, and duration, providing flexibility for various lottery scenarios.

2.3 Application Domain

The lottery industry plays a significant role in the global gaming market, offering participants the excitement of winning substantial prizes. However, traditional lottery systems often face issues related to transparency, trust, and operational efficiency. These challenges can deter participation and undermine the integrity of lottery operations. In traditional lottery systems, participants must trust operators to conduct fair and unbiased draws. The lack of transparency in the draw process can lead to suspicions of fraud or manipulation. Centralized control over ticket sales, prize distribution, and winner selection poses significant security risks, including data breaches and unauthorized access to sensitive information. Blockchain as a Solution for Decentralized Lottery Operations Blockchain technology presents a transformative solution by introducing a decentralized, transparent, and secure framework for lottery operations. The EtherLotto platform utilizes the Ethereum blockchain to redefine lottery processes, ensuring a fair and trustworthy gaming experience for all participants.

Transparency and Immutability: The EtherLotto platform records all transactions, including ticket purchases and prize distributions, on the Ethereum blockchain. This immutable ledger ensures that all actions are transparent and verifiable by participants, eliminating doubts about the fairness of the lottery process.

Security and Data Integrity: By leveraging smart contracts, EtherLotto automates and secures the entire lottery lifecycle. Smart contracts ensure that the rules of the lottery are enforced without intermediaries, reducing the risk of human error or fraud. The blockchain protects sensitive participant information from unauthorized access and tampering [2].

Decentralized Control and Fairness: The decentralized nature of the Ethereum blockchain ensures that no single entity controls lottery operations, mitigating manipulation risks and enhancing overall fairness. The Verifiable Random Function (VRF) integrated into EtherLotto ensures that winner selection is tamper-proof and verifiable, providing participants with confidence in the lottery outcomes [12].

Enhancing User Experience and Operational Efficiency: Blockchain technology not only addresses trust and security concerns but also improves operational efficiency and user experience.

Reduced Administrative Overhead: Blockchain's peer-to-peer transaction capability facilitates direct interactions between participants and the lottery system, reducing administrative overhead and streamlining operations. This efficiency allows for faster ticket sales, real-time prize distributions, and seamless transaction processing [13].

Cost Efficiency: The elimination of intermediaries in the lottery process leads to significant cost savings, which can be passed on to participants as lower ticket prices or higher prize pools, enhancing the attractiveness of the lottery [14].

The entire codebase, along with monetary transactions and participant information, is publicly accessible. This design fosters a fair user experience without the need for blind trust. Every aspect is auditable, ensuring the delivery of a secure and trustworthy service. In summary, The application domain of the EtherLotto platform demonstrates the significant advantages of integrating blockchain technology into lottery operations. By ensuring transparency, security, and fairness, EtherLotto addresses the core challenges faced by traditional lotteries. The platform's decentralized nature, coupled with the efficiency of blockchain technology, sets a new standard for lottery systems, promising a secure and enjoyable experience for all participants.

3 Context

3.1 Aim of the DApp

EtherLotto's primary goal is to revolutionize traditional lotteries by introducing a decentralized, blockchain-based solution. This section elucidates the motivations behind choosing blockchain technology and the specific blockchain type employed to ensure secure and transparent lottery operations.

While we don't seek to promote gambling, we acknowledge its prevalence in many cultures and among numerous individuals. Traditionally, people resort to centralized and sometimes questionable systems or companies to participate. We believe that everyone desires confidence in the fairness and genuine randomness of lottery systems when they invest money to take part.

Hence, we introduce EtherLotto as an open-source, fully automated alternative to conventional lotteries. Our aim is to offer people the assurance they seek by placing their trust in the Blockchain instead of relying on closed and centralized services.

3.2 Choosing a Blockchain

We will look at the Wüst & Gervais chart to see if and what type of Blockchain we should use.

To ensure the functionality of EtherLotto, we will record every transaction to maintain a comprehensive record of who purchased which type and quantity of lottery tickets. Our vision is to create a lottery that is openly accessible to all, allowing

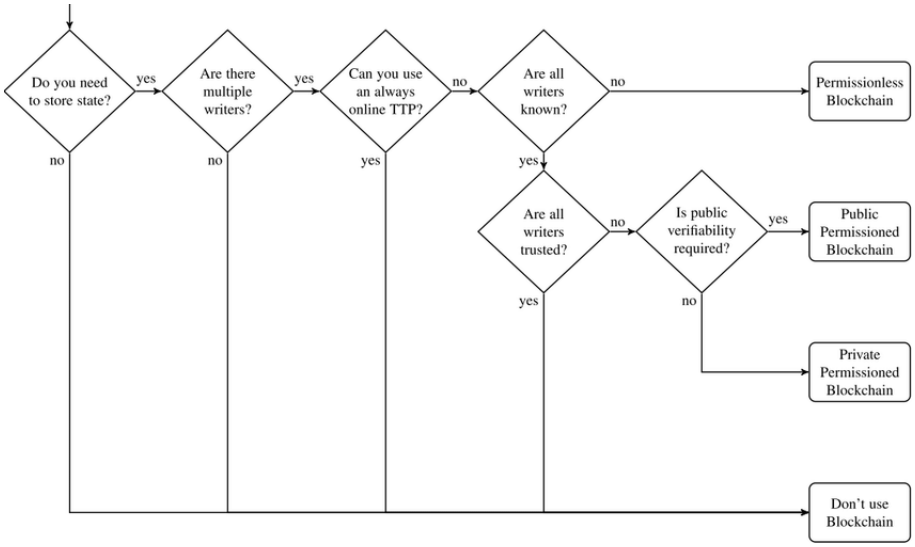

Figure 1: Why a blockchain is needed? according to [15]

multiple unknown writers to participate. As previously suggested, traditional centralized lottery services often lack trustworthiness. For those reasons we chose to use Ethereum as a Permissionless Blockchain, for our project.

3.3 The Advantages of using the Blockchain for this Purpose

3.3.1 Safety and Security

In the lottery industry, various scams have been prevalent. External scams range from attempts to print or steal a winning ticket. Additionally, middle-tier retailers sometimes falsely claim prizes for themselves. By storing transactions on the public, immutable Blockchain, we mitigate external scams, ensuring transparency and enabling verification for all participants.

Another concern involves internal scams, where the lottery service manipulates winning numbers to increase their chances of claiming the prize. Several instances of such misconduct have eroded trust in single centralized institutions. Leveraging the decentralized nature of Blockchain and an open-source smart contract, we effectively eliminate this issue. Moreover, Blockchain's inherent security has demonstrated resilience against hackers and similar threats.

Through EtherLotto, we aspire to instill a sense of trust and confidence among users, allowing them to place their trust freely in the system.

3.3.2 Automation

The whole lottery service will be automated and run by smart contracts. Therefore no intermediates are needed and the costs for them can be safed. Therefore all the money that gets in from ticket sales will be used for the prize pool. Also the prize

money can be transferred to the winner immediately and directly.

3.3.3 Global participation

The accessibility of the Blockchain extends to anyone worldwide with internet access [16]. EtherLotter harnesses this global accessibility, enabling lotteries to engage significantly larger participant pools and consequently, larger prize pools. Simultaneously, participants can maintain anonymity by solely providing their Ethereum address, preserving the integrity of the lottery process.

4 Software Architecture

A visual exploration of EtherLotto's software architecture, featuring detailed diagrams illustrating the relationship between smart contracts, users, and the blockchain network. This section provides an in-depth understanding of the DApp's underlying structure.

4.1 Use Case Diagram

The User

- The User logs into the System using MetaMask to authenticate their Identity
- In the user interface he can browse the different available lotteries and choose one to his liking. He can see the details of the lotteries such as: entry price, price pool, expiry date and so on.
- After choosing a lottery he can buy a ticket for it. He should be able to see the bought ticket. He can buy multiple tickets for the same lottery. A user can also buy many tickets from different lotteries.
- After the end of a lottery the user will be able to see if he has a winning ticket

The Manager

- The Manager also logs into the System using MetaMask to authenticate their Identity
- In the Manager Interface he has the Opportunity to create a new lottery within the application.
- After one of his lotteries expired he can start the winner selection.

Note that everyone can also be a Manager. But only the Manager of a lottery can start the lottery's winner selection. Both managers and users interact with the Ethereum blockchain to perform transactions and retrieve data. This ensures that all operations are transparent and verifiable on the blockchain.

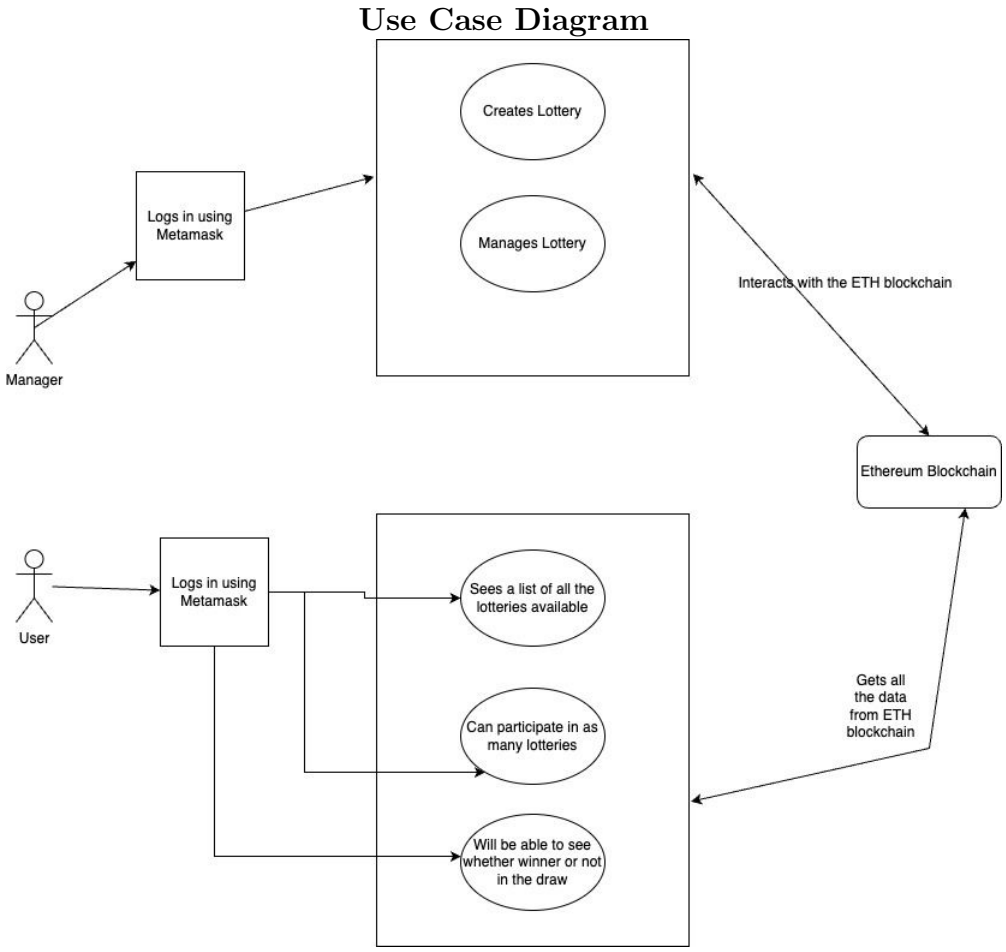

Figure 2: Use Case Diagram

4.2 Components Diagram

The components diagram of the EtherLotto project illustrates the integration of various technologies across the frontend, backend, and blockchain network. The frontend, built with React, React Bootstrap, and Sass, utilizes MetaMask for secure user authentication and interaction with the Ethereum blockchain. This interaction is facilitated through ethers.js, which allows the frontend to communicate directly with the blockchain, calling smart contract functions and reading blockchain data. The backend development process involves Hardhat for compiling, testing, and deploying Solidity smart contracts. These contracts, written in Solidity, are executed on the Ethereum blockchain, ensuring transparency, security, and decentralization in the lottery operations. This diagram highlights the integration and interaction between the frontend, backend, and blockchain, forming the robust architecture of the EtherLotto decentralized application.

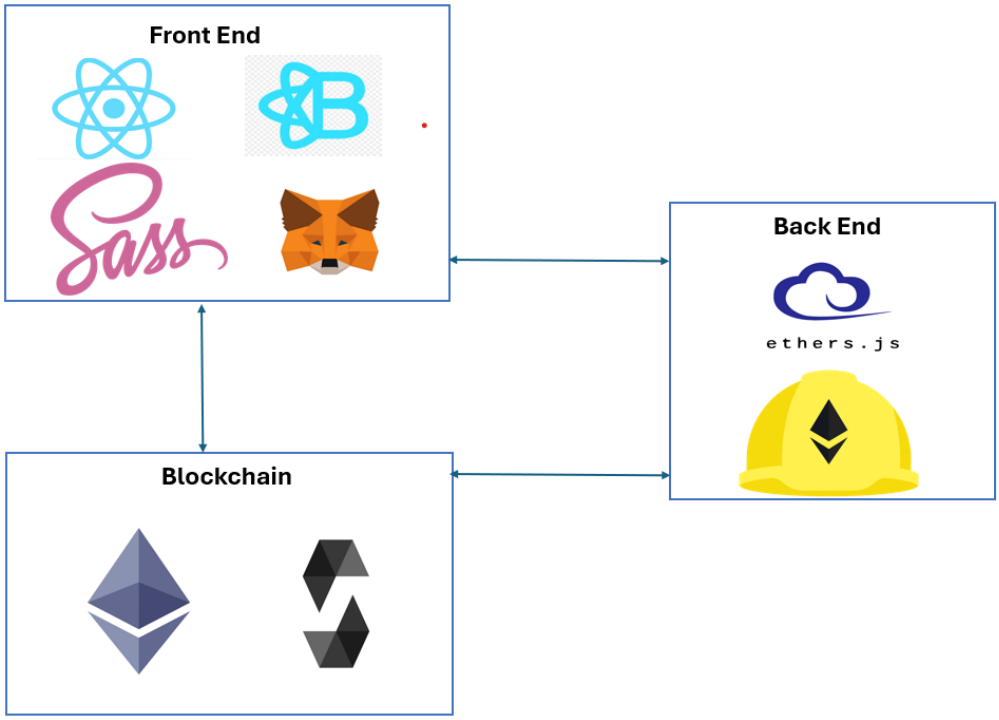

Figure 3: Overall Components of Etherlotto system

4.3 Smart Contract Concept Diagram

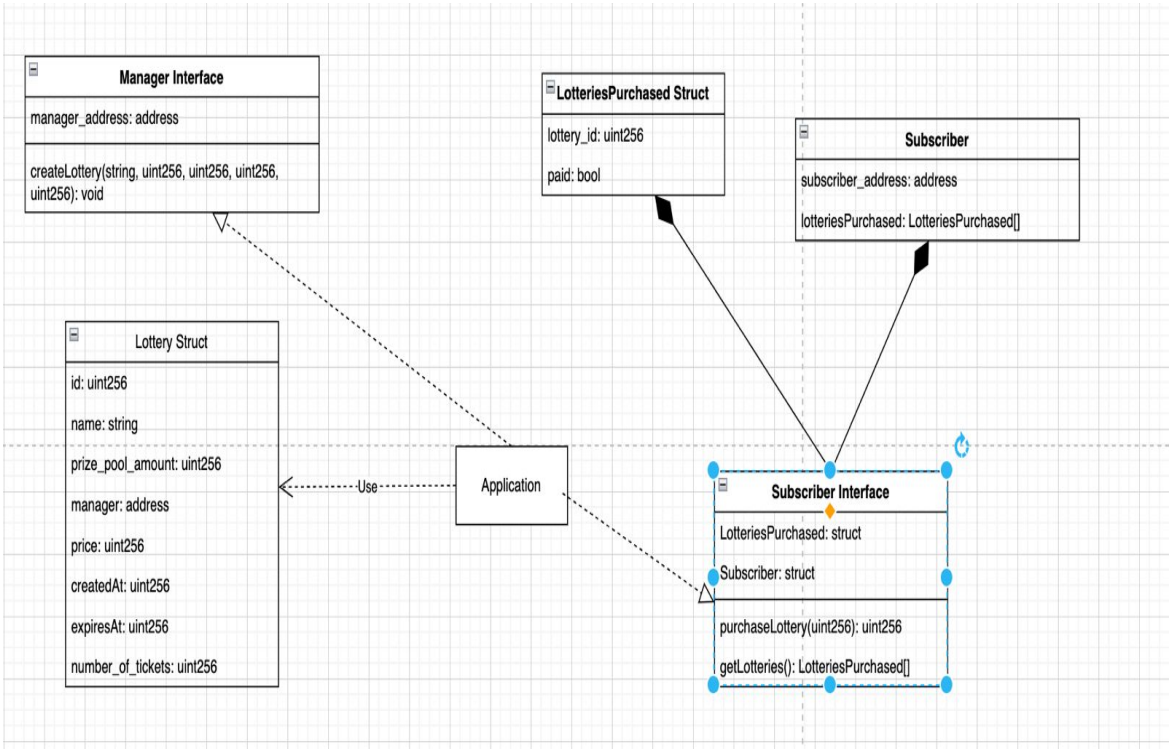

Figure 4: UML Class Diagram

Manager Interface

- manager_address: Holds the Ethereum address of the manager.
- createLottery: A function that allows the manager to create a new lottery by providing necessary details such as name, price, and expiration date.

Lottery Struct

- Contains essential information about a lottery, including ID, name, prize pool amount, manager's address, price, creation, and expiration dates, as well as the total number of tickets.

LotteriesPurchased Struct

- Represents the lotteries that have been purchased by an individual, including the lottery ID and payment status. it has lifecycle dependency on subscriber interface

Subscriber

- subscriber_address: Stores the Ethereum address of a subscriber (user).
- lotteriesPurchased: An array that keeps track of all the lotteries purchased by the user.

Subscriber Interface

- getLotteries: A function that retrieves the list of lotteries available for purchase.

Application

- The central module that uses the Manager and Subscriber interfaces to facilitate the operation of the EtherLotto platform. It interacts with the Manager Interface to create and manage lotteries and with the Subscriber Interface to allow users to purchase lottery tickets and view available lotteries.

The application is structured to ensure secure and transparent transactions on the Ethereum blockchain, with smart contracts serving as the backbone of the EtherLotto platform. The smart contracts enforce the rules of the lotteries, handle the logic for ticket purchases, and the distribution of winnings. This design ensures that once deployed, the EtherLotto application operates in a decentralized and autonomous manner, with minimal need for human intervention.

4.4 Sequence Diagram

The sequence diagram below illustrates the interaction between different components of the EtherLotto system. It shows the sequence of actions taken by both the manager and the user, and how these actions interact with MetaMask, the EtherLotto DApp, and the smart contracts on the Ethereum blockchain.

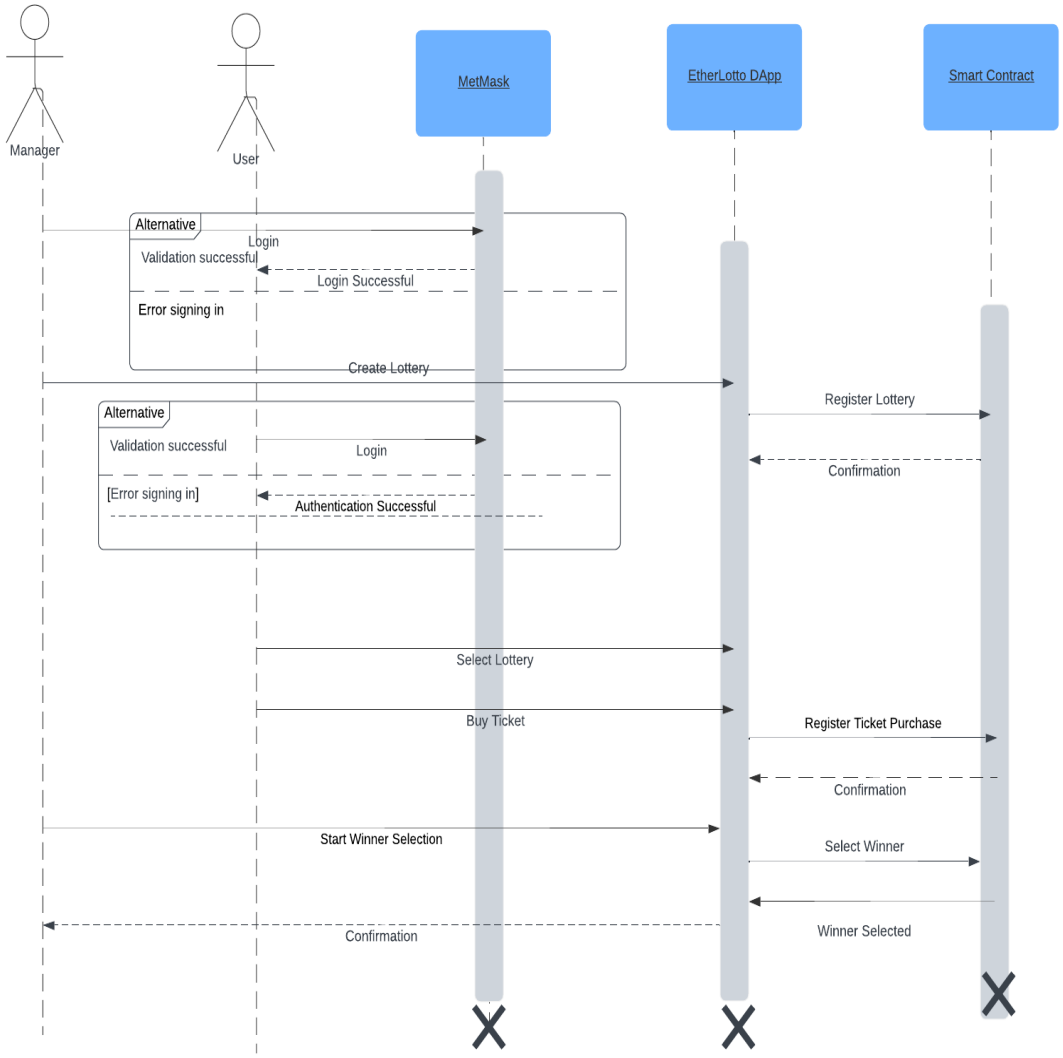

Figure 5: Sequence Diagram for EtherLotto

- **Manager Login Sequence**:
  - The manager logs in through MetaMask.
  - MetaMask authenticates the manager login and confirms the authentication.
- **Creating Lottery Sequence**:
  - The manager creates a new lottery using the EtherLotto DApp.

  - The EtherLotto DApp registers the lottery by interacting with the smart contract on the Ethereum blockchain.
  - The smart contract confirms the registration of the lottery.

- **User Login Sequence**:
  - The user logs in through MetaMask.
  - MetaMask authenticates the user and confirms the authentication.

- **Buying a Ticket Sequence**:
  - The user selects a lottery and buys a ticket through the EtherLotto DApp.
  - The EtherLotto DApp registers the ticket purchase by interacting with the smart contract.
  - The smart contract confirms the ticket purchase.

- **Trigger Winner Selection**:
  - The manager initiates the winner selection process through the EtherLotto DApp.
  - The EtherLotto DApp instructs the smart contract to select a winner.
  - The smart contract selects the winner and confirms the selection.

4.5 Activity Diagram

The activity diagram for the EtherLotto project illustrates the flow of activities involved in the system, divided into four swimlanes: Manager, User, DApp and Smart Contract. The process begins with both the manager and the user logging into the system through MetaMask, which can result in either a successful or failed login. Upon successful login, the manager provides the lottery details and creates the lottery, which the DApp registers on the blockchain. The registration is confirmed, and the manager acknowledges this confirmation before initiating the winner selection process. Meanwhile, the user, after a successful login, selects a lottery and purchases a ticket. The DApp/Smart Contract handles the backend operations, including receiving lottery details, registering the lottery, confirming the registration, and executing the winner selection process.

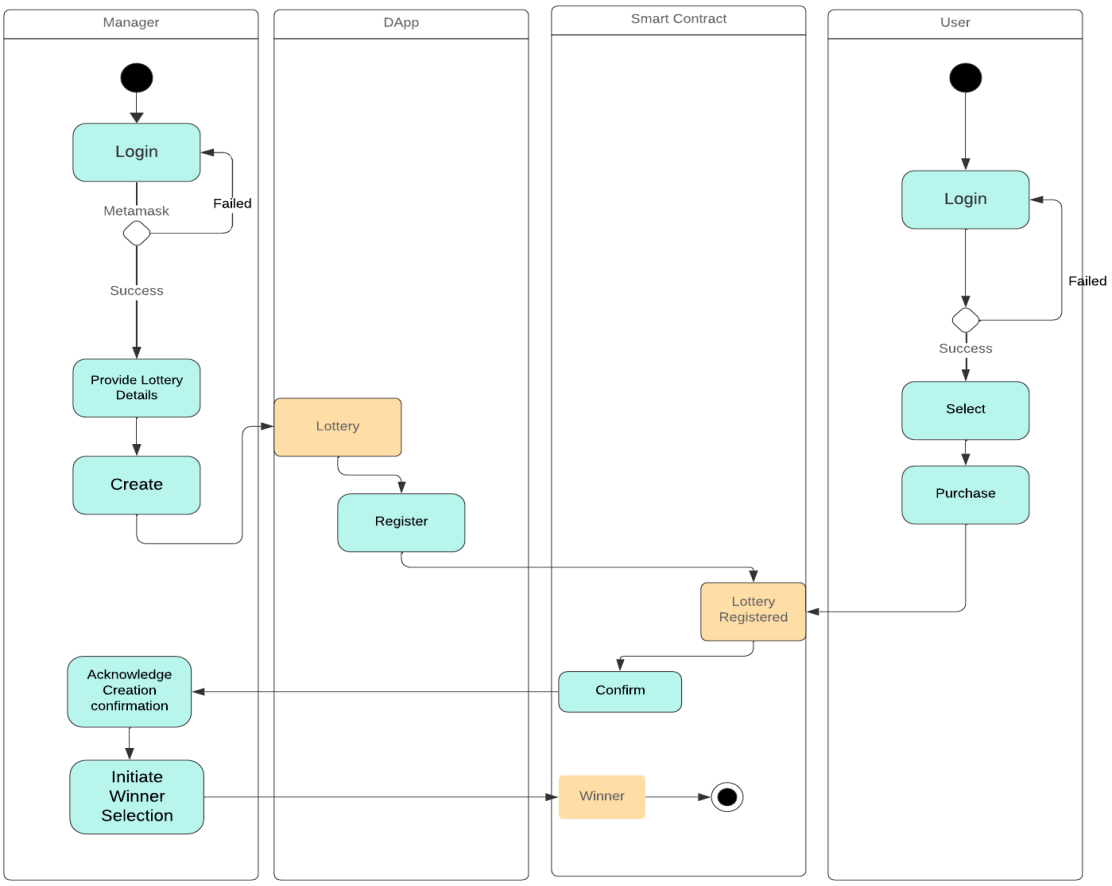

Figure 6: Activity Diagram for EtherLotto

5 Implemtentation

5.1 Tools and Libraries

- Solidity: Solidity is a programming language used for writing smart contracts on blockchain platforms, particularly on Ethereum. It allows developers to define the rules and logic of decentralized applications (DApps).
- React: React is a JavaScript library for building user interfaces. In the context of your project, React is likely used for creating the frontend of your lottery application, enabling dynamic and responsive user interfaces.
- Hardhat: Hardhat is a development environment for Ethereum that facilitates the building, testing, and deployment of smart contracts. It provides a set of tools to streamline the development workflow.
- ethers.js: ethers.js is a JavaScript library commonly used for interacting with Ethereum. It provides utilities for working with wallets, contracts, and transactions.

It's a popular choice for building decentralized applications on the Ethereum blockchain.

- React Bootstrap: React Bootstrap is a library that combines React and Bootstrap, a popular CSS framework. It provides pre-built React components styled with Bootstrap, making it easier to create visually appealing and consistent UIs.
- Sass: Sass is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets (CSS). It extends CSS with features like variables, nested rules, and mixins, allowing for more maintainable and modular stylesheets in your React application.

5.2 Backend

EtherLotto embraces Ethereum's decentralized architecture to ensure transparency and fairness in every facet of its operation. Ethereum's blockchain serves as the immutable ledger, recording every lottery instance, ticket purchase, and prize distribution. The decentralized nature of Ethereum mitigates the risks associated with centralized lotteries, providing participants with an unprecedented level of security. All of the core functionality of the backend is implemented in a smart contract using Solidity, allowing for the creation, management, and execution of lottery events in a decentralized and trustless manner.

5.2.1 The Smart Contract

```
contract Lottery {

    /* Out Lottery Struct that represents a lottery
    * id is used to identify and address the lotteries
    * lottery_name is the name of the lottery given by the manager
    * current_prizePool tracks how much tickets have been bought and what will be the payout
    * manager stores the address of the manager that creteded (only he can start the select_winner process)
    * createdAt is the time the lottery was created
    * epxiresAt is the time when the lottery expires and when the winner can be selected
    */
    struct LotteryStruct {
        uint256 id;
        string lottery_name;
        uint256 current_prizePool;
        address manager;
        uint256 price;
        uint256 createdAt;
        uint256 expiresAt;
    }
```

In the `Lottery.sol` file we implement our smart contract. There is a lottery struct (see picture). Each new created Lottery is represented by an instance of this struct. The lotteries are addressed by their Id which is given to them in ascending order of creation. In the `lastTicketsArray` mapping we store how many Tickets for each Lottery have been bought and with that also, what would be the number for the next ticket, since their also given out in ascending order. Because we also need to remember

who bought which ticket we have the `ticketToAddress` mapping. Other than getter and setter we have 3 main functions.

To create a lottery a manager calls **createLottery** and has to input the name, the ticket price the creation date and the expiry date. After checking the values to be correct a new lottery struct is created with the current price pool of 0, the manager being the message sender and the id equal to the lottery counter. Before that the lottery counter is being increased and with that the contract knows a new lottery.

The **purchaseLottery** function enables users to join a specified lottery. Participants acquire lottery tickets by sending a specified amount of Ether, representing the ticket price, to the smart contract. The function ensures the validity of the transaction, checking factors like the existence of the selected lottery, available tickets, and correct Ether value. Upon successful validation, the participant's address is linked to a ticket number, and the lottery's data is updated.

The **selectWinner** function orchestrates the random selection of a lottery winner within a decentralized lottery system on the Ethereum blockchain. Triggered by the lottery manager, it verifies that the lottery is valid and has expired, ensuring a fair winner selection. The function utilizes a pseudo-random process, incorporating additional factors, to determine the winning ticket. The prize pool is being send to the winners address and an event is triggered.

5.2.2 Testing

We test the 3 named function for their correctness. With `ether.js` and `hardhat` we can write unit test in the test folder of the project and simply execute them with: `npx hardhat test`. It then shows us which test are completed and also which ones took a longer time. Our test check that the function works the way it should by simulating the Blockchain with truffle, deploying the contract and executing the functions on it. We then check that all the values are correct and also that invalid transaction are being reverted (e.g. wrong purchasing a ticket with wrong price).

```
Compiled 1 Solidity file successfully (evm target: paris).


  Creating Lotteries
    ✔ Should create a new lottery with valid parameters (55ms)
    ✔ Should increase the lottery counter after creating a new lottery
    ✔ Should increase the lottery counter after creating many new lotteries (2711ms)
    ✔ Should not create a new lottery after trying to create a few falseful lotteries

  Buying Lottteries
    ✔ Should allow a player to purchase a lottery ticket
    ✔ Should not buy a single ticket after trying to buy false tickets (65ms)

  Lottery Contract - Selecting Winner
    ✔ Should allow the owner to select a winner (3037ms)
    ✔ Should not allow a non-owner to select a winner


  8 passing (7s)
```

5.3 Frontend

The front end consists of the following pages with screenshots below:

Manager Dashboard Interface: The EtherLotto Manager Dashboard, displaying the current prize pool and ticket price for a lottery named 'Checking', with an expiration date highlighted. There's a section to create new lotteries, emphasizing the user-friendly management tools provided by EtherLotto for lottery administrators.

Player Dashboard Interface: EtherLotto Player Dashboard with a minimalist design. The dashboard provides a personalized experience for users to manage their lottery tickets.

Lottery Timing Details: detailed information about a specific lottery within the EtherLotto system. It provides the current date and time, the next draw date, and a countdown to the next lottery draw, offering players real-time updates and enhancing the anticipation and excitement of the draw.

Lottery Ticket Purchase Interface: The ticket purchasing interface for a lottery. It displays the prize pool, ticket price, and expiration date, with a 'Buy' button, demonstrating the ease of participating in the lottery through the EtherLotto platform. A player can view their 'Checking' lottery tickets. It shows an increased prize pool and provides an option to view the lottery draw, showcasing the transparent and interactive features of the platform.

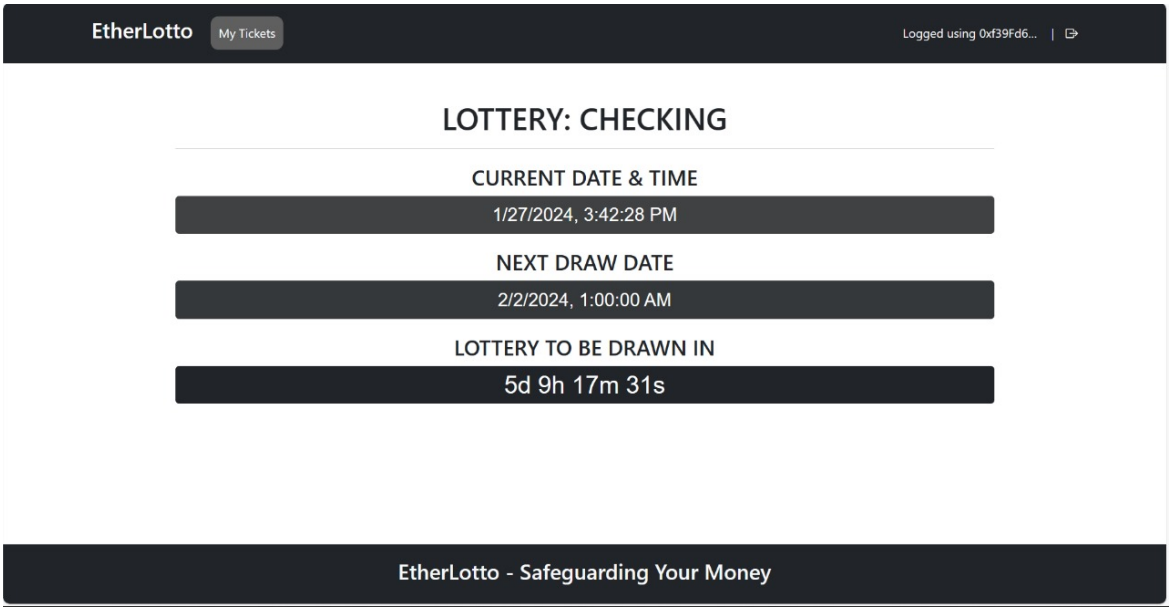

Figure 7: Frontend Image 1

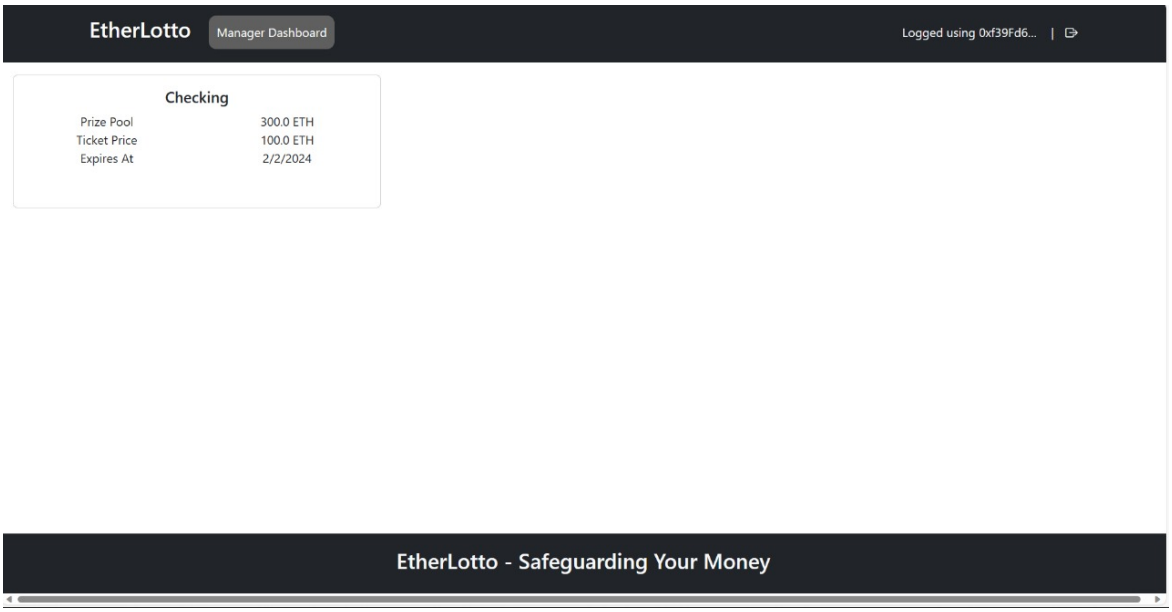

Figure 8: Frontend Image 2

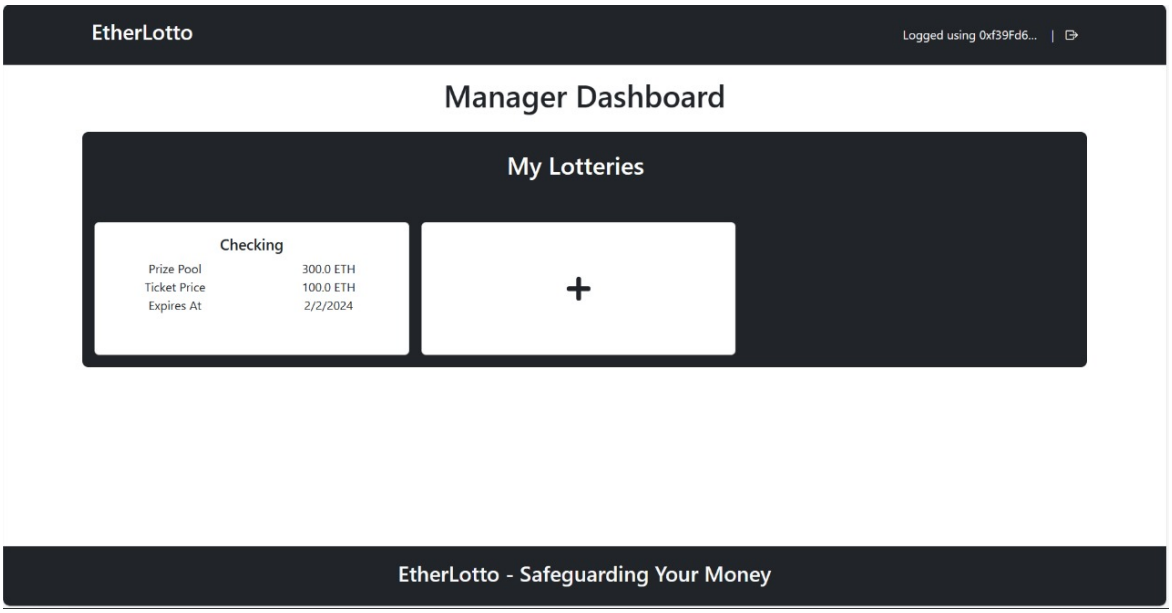

Figure 9: Frontend Image 3 Manager Dashboard

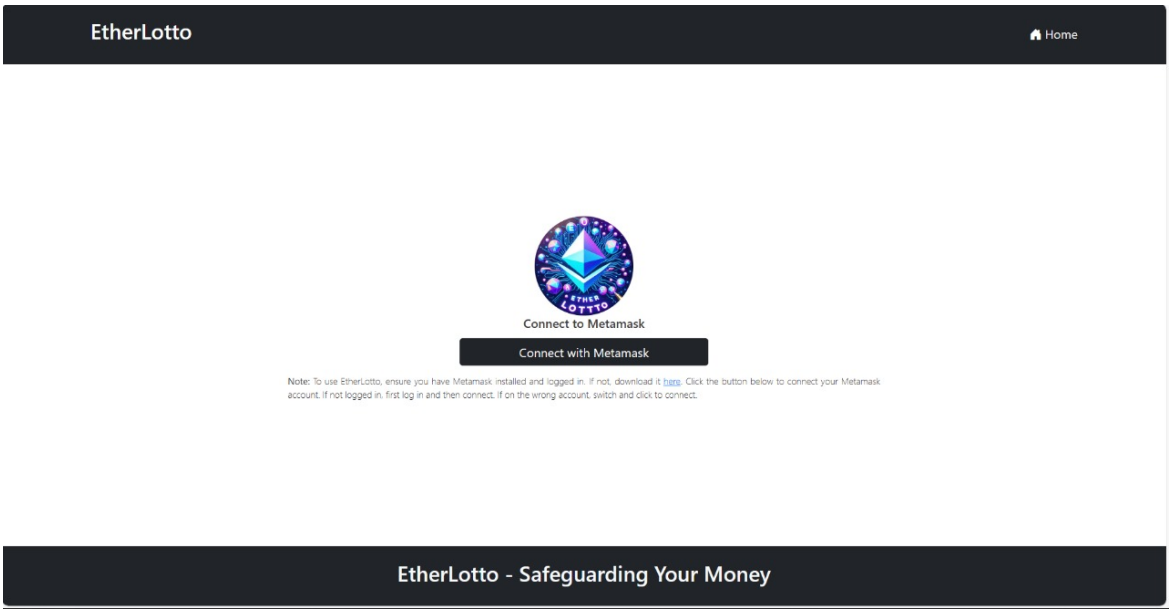

Figure 10: Frontend Image 4

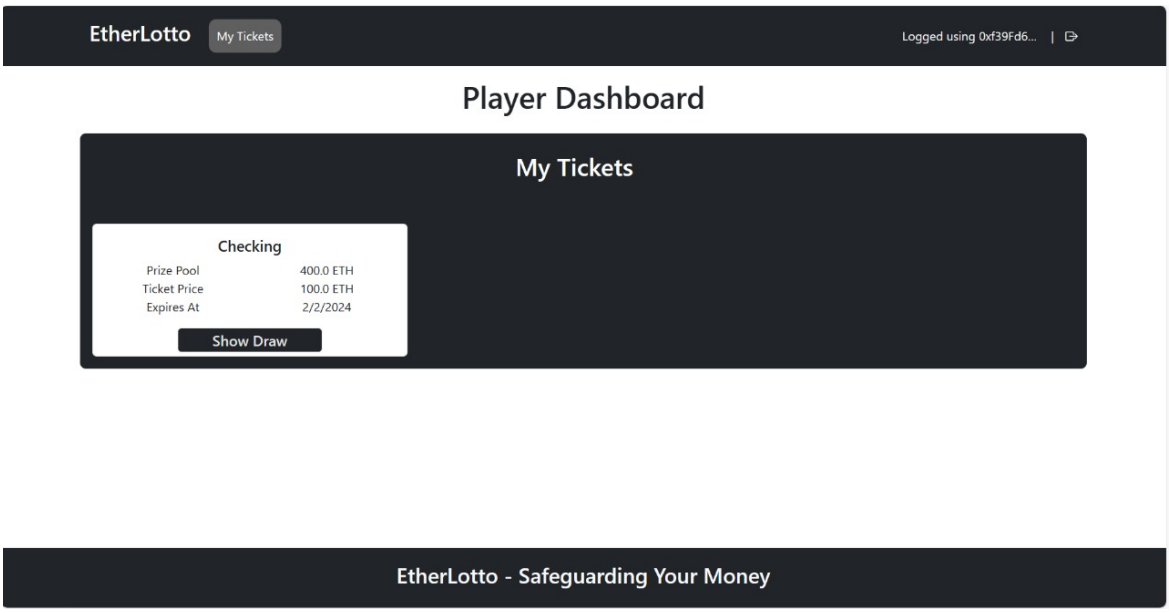

Figure 11: Frontend Image 5 Player Dashboard

6 Known Limitations

6.1 Limitations of Solidity

Cryptographic Randomness Verifiable Random Function (VRF): One notable challenge revolves around the reliance on cryptographic randomness for the winner selection process. While the initial implementation incorporates a robust Verifiable Random Function, the inherent limitations of generating truly random numbers on a blockchain without external tools or paid services pose a challenge. This limitation may impact the absolute unpredictability of lottery outcomes.

Lottery Triggering The initiation of a new lottery and the triggering of a winner demand an actual agent to "press the button". This introduces a centralized aspect in the otherwise decentralized application. Future iterations may explore automated triggering based on predefined conditions, enhancing the DApp's autonomy. However this is not native to the language solidity or to the blockchain.

6.2 Regulatory Hurdles and Taxes

The complex legal landscape surrounding blockchain and cryptocurrencies introduces regulatory challenges. While EtherLotto maintains a commitment to user anonymity, participants may still be subject to tax liabilities based on local regulations. Navigating these legal intricacies and finding a balance between anonymity and tax compliance poses a notable challenge for the widespread adoption of the platform.

6.3 User Adoption

Convincing traditional lottery players to embrace a blockchain-based system may encounter resistance due to the novelty and unfamiliarity of decentralized technologies. Building trust in EtherLotto's transparency, fairness, and security features requires strategic education and outreach efforts. Overcoming the inertia of existing player behaviors in favor of a blockchain-based system is a significant hurdle that necessitates targeted marketing and awareness campaigns.

Addressing these known issues and limitations is crucial for EtherLotto's evolution. Future development efforts will focus on refining the VRF implementation, automating lottery triggering processes, navigating regulatory complexities, and implementing effective strategies to encourage user adoption. The iterative nature of blockchain technology allows for ongoing enhancements to address these challenges and deliver an increasingly robust and user-friendly decentralized lottery experience.

7 Conclusion

In conclusion, this report provides an exploration of EtherLotto, a decentralized application (DApp) that we believe can change the landscape of traditional lotteries through blockchain technology. Beyond the scope of this project, we think EtherLotto can popularize the following way of thinking: If cryptocoins were the norm, what would there be? EtherLotto was one of such answers. We thought that lotteries in the future would use Ethereum's decentralized infrastructure and smart contracts to ensure transparency, fairness, and security in all operations.

In the report we talk about the background of the Ethereum blockchain and the creation of EtherLotto, emphasizing from the very beginning our three foundational principles: fairness, transparency, and security. The utilization of smart contracts and the planned integration of Chainlink VRF show the work we did to ensure an unbiased lottery experience.

The report also explains the software architecture of EtherLotto, using diagrams to illustrate the relationships between smart contracts, users, and the Ethereum blockchain. This part shows all the initial development and planning we did; it's the backbone of the project.

Despite our best efforts for making a well rounded project, we also honestly acknowledged the known issues and limitations. Challenges such as cryptographic randomness for winner selection, centralized aspects in lottery triggering, regulatory hurdles, and user adoption resistance are identified. This is the roadmap in case we ever want to keep working on the project, as we targeted the biggest obstacles for future development.

In essence, EtherLotto is a DApp with the potential to redefine the lottery industry. By combining blockchain technology, smart contracts, and a commitment to core principles, EtherLotto works to provide a secure, transparent, and globally accessible alternative to traditional lotteries.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. https://bitcoin.org/bitcoin.pdf.

[2] Vitalik Buterin. Ethereum white paper: A next generation smart contract decentralized application platform, 2013.

[3] Ethereum: A secure decentralised generalised transaction ledger. 2019.

[4] Vitalik Buterin, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Y. Wang, and Yan X Zhang. Combining ghost and casper. *arXiv preprint arXiv:2003.03052*, 2020.

[5] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. Technical report, Ethereum Foundation, 2017.

[6] Christian Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, page 310, 2016.

[7] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *IEEE International Congress on Big Data (BigData Congress)*, pages 557–564, 2017.

[8] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldowich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th ACM Symposium on Operating Systems Principles*, SOSP '17, pages 51–68, New York, NY, USA, 2017. Association for Computing Machinery.

[9] Ethereum Foundation. What is a blockchain? `https://ethereum.org/en/developers/docs/blockchain/what-is-a-blockchain/`, 2023.

[10] Ethereum Foundation. Ethereum virtual machine (evm). `https://ethereum.org/en/developers/docs/evm`, 2023.

[11] Chainlink. Verifiable random function (vrf). `https://chain.link/education-hub/verifiable-random-function-vrf`, 2023. Accessed: 2023-10-27.

[12] Joseph Poon and Vitalik Buterin. Plasma: Scalable autonomous smart contracts. `http://plasma.io/plasma-deprecated.pdf`, 2017. Accessed: 2024-06-03.

[13] Xiwei Xu, Ingo Weber, and Mark Staples. *Architecture for Blockchain Applications*. Springer International Publishing, Cham, Switzerland, 2019.

[14] John Francis. The impact of blockchain technology on financial transactions. *Journal of Financial Transformation*, 51:123–136, 2020.

[15] K. Wüst and Arthur Gervais. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54, 2018.

[16] Chipin. Fire lotto ico: The blockchain lottery. `https://www.chipin.com/fire-lotto-ico-blockchain-lottery/`, 2018. Accessed: 2024-06-03.

[17] Arihant Duggar, Divyanshu Gupta, Royal, and Mohan C.G. Secured lottery system using smart contract and blockchain technology. *International Journal for Research in Applied Science & Engineering Technology (IJRA)*, 10(IV), April 2022.

[18] Aarju Dixit, Aditya Trivedi, and W. Wilfred Godfrey. Blockchain based secure lottery platform by using smart contract. In *2022 IEEE 6th Conference on Information and Communication Technology (CICT)*, pages 1–5, 2022.