Ethereum-Based Quiz Application

Nikhita Kokkirala
Gsu department of Computer Science
Blockchain & Applications
Atlanta, United States
nkokkirala1@student.gsu.edu

Michael Narine
Gsu department of Computer Science
Blockchain & Applications
Atlanta, United States
mnarine1@student.gsu.edu

Farnoosh Sharifian Esfahani
Gsu department of Computer Science
Blockchain & Applications
Atlanta, United States
fsharifianesfahani1@student.gsu.edu

***Abstract*—This paper defines a decentralized application based on the Ethereum platform. The DApp is a Quiz Application that starts upon entrance fee paid by the users. It allows the user to participate in or make their own quiz for others to participate in excluding themselves. The entrance fee is submitted into a accruing pool, and the final balance will be the prize awarded to the winner. The winner is decided based on the whoever obtains the perfect score first.**

I. INTRODUCTION

The general idea of the project is to have an electronic quiz system using Ethereum, where the users will have the opportunity to take a quiz. The user will pay with ether as the entrance fee, in order to take the quiz and can only take the quiz once. A user also has the opportunity to create a quiz for other users to participate in. However, the quiz creator is not allowed to participate in their own quiz so that everyone taking the quiz has a fair chance, and no individual has prior knowledge of the correct answers. Entrance fees paid by the users for that particular quiz, will be submitted into an accruing pool, which will be the prize awarded to the winner. The winner is chosen based upon who ever submits the correct answers first. After a winner receives the prize, the pool balance is reset to zero. This defines that no other individual has the possibility of winning a prize after the prize has already been awarded to another individual.

II. SURVEY OF RELATED WORK

There are two pivotal parties in a lottery game: player and lottery provided[2]. And there are essentially four stages: initialization, lottery purchase(in this project this stage will also include the process of users submitting his/her answer choices), closing time, and verifying winning numbers(submitted quiz answers).

A commitment-based lottery system is defined in "Zero-Collateral Lotteries in Bitcoin and Ethereum". The paper discusses a system where users interact with a smart contract by publishing a transaction[1]. The transaction contains a procedure call, the address of the contract, the name of the function(the function that is to be invoked), and any arguments to pass. Whenever a user creates a transaction, he/she pays a "gas" fee. The creation of a transaction acts as an initiation for the user to into the lottery. The lottery contract is defined as a time-based state machine, where there are hardcoded deadlines where users must submit their commitments[1]. This means that the user has a certain time window to make any submissions. In order to define the multi- user capability, a N-Player tree is created in python which helps keep track of the number of players and the time they submitted their commitments[1].

A useful algorithm that helps randomize a given array is the Fisher-Yates where any combination or permutation of the elements of the array must be equally likely[3]. The complexity of the algorithm is O(n). A simple solution takes in an array and creates a temporary(usually a copy of the original array). Then, a random element is selected and removed from the temporary array which replaces the first index of the original array. The index position of the original is incremented. And the process of selecting from the temporary and replacing an element in the original array for n times. The Fisher-Yates algorithm generates a random number O(1) using a function rand(). Starting from the last element, the element in question is swapped with a randomly selected element from the whole array. This process is repeated until the first element is reached.
When designing a lottery system, it imperative to consider the following parameters:

1) Is the drawing secure and random?(Fairness)
2) How is the winner determined?(Transparency)
3) How are the funds allocated and determined? (Security/Privacy)

The paper, "Design of A Blockchain-based Lottery System for Smart Cities Applications" goes over a blockchain-based lottery system called FairLotto and also utilizes smart contracts[2]. FairlLotto contains three layers- Data, Smart Contracts, and Interface Layers. There are four kinds of smart contracts in Fair Lotto: user registration, money transfer, lottery claiming, and payouts[2]. The Interface layer is the connection between the blockchain and the user i.e. the platform that the user actually makes changes to and accesses.

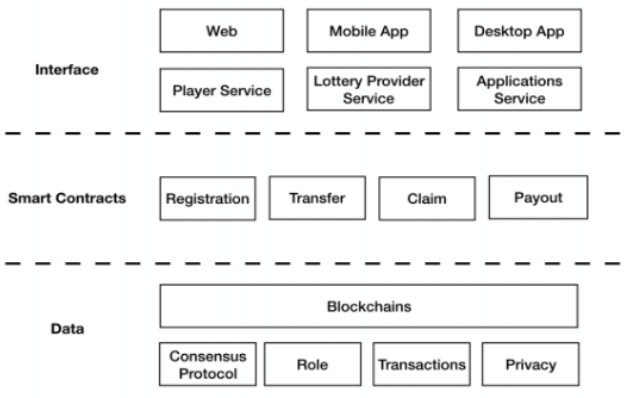


Figure 1. A Depiction of the FairLotto Software Structure.

III. DETAILED ALGORITHM

A. *Making Quiz Algorithm*

User has the ability to take a quiz and certain requirements are taken in such as: Quiz Name, Entrance Fee, Initial Pool Amount, and a Question. Also, each question has particular properties such as: 1 correct answer choice and 3 wrong answer choices.

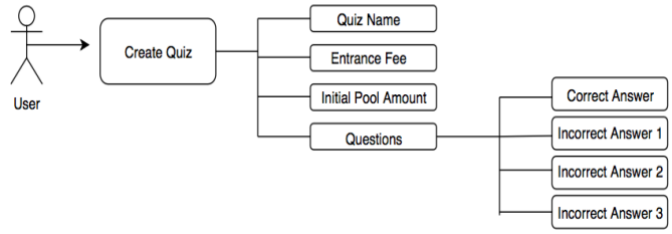


Figure 2: Make Quiz Flow chart

B. *Payment Algorithm*

There are major sectors in terms of payments in this application which are: user paying money, adding amount paid to pool balance, and system giving user money.

In order for the user to pay money, the quiz ID needs to be greater than zero which means that a new quiz needs to be created. Also the quiz ID needs to be less than total number of quizzes because whenever a new quiz is created, the quiz ID is incremented based on the last quiz number. Also the amount the user paid is greater than or equal to the entrance fee set by the creator of the quiz.

After the user pays money and the fact that the user has paid has been confirmed, then that particular amount can be added to the pool.

After the pool has accrued a balance of money(pool balance is based on the money collected by user's entrance fee), and a winner is decided, money can be sent from the system to the user. A winner is decided based on whoever turns in a quiz with a perfect answer choice. In order for the system to give the winner the total pool balance for a particular quiz, the quiz ID and the winner's account address is required. The user must have also paid for the quiz and had taken the quiz. The pool balance is then sent to the winner's account balance and the pool balance is immediately set to zero. By setting the pool balance to zero, it ensures that nobody else can take the quiz or win the pool balance.

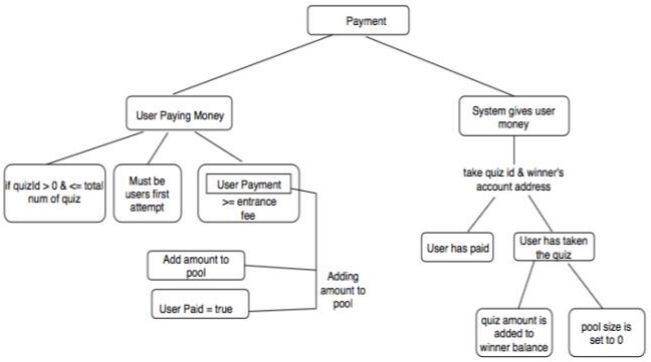


Figure 3: Taking/Scoring Quiz Flowchart

C. *Taking/Scoring Quiz Algorithm*

In order for a user to take a quiz, it must be the user's first attempt and the quiz must be a valid quiz. If these conditions are met, then the quiz and it's answer options are returned. For scoring a quiz, the quiz ID and the user's answer choice is taken in as a parameter. Then, the user's answer is compared to the correct answer, and if the user answered it correctly, then the user will a perfect score for that question, else will not.

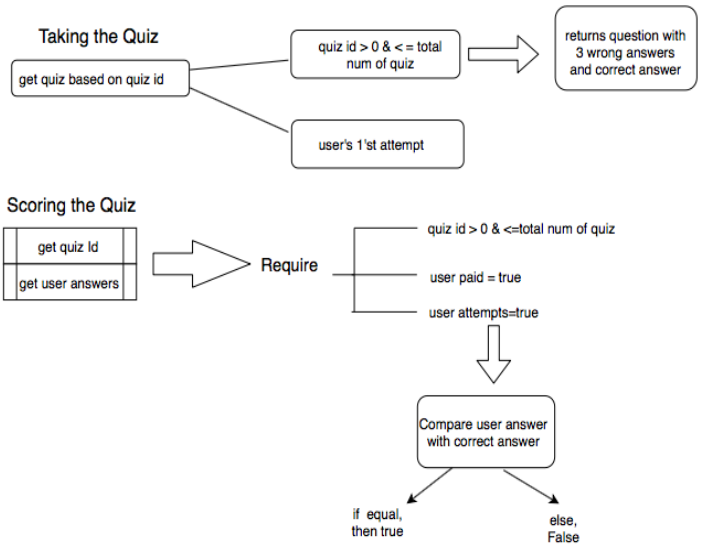
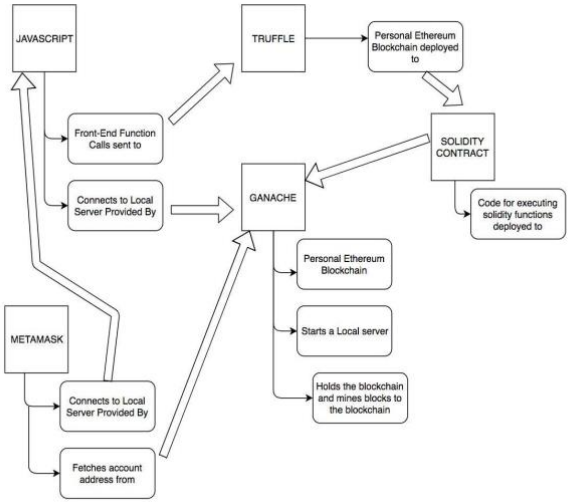


Figure 4: Taking/Scoring Quiz Flowchart

IV. TECHNOLOGY

The major elements for technology and coding is JavaScript, Metamask, Truffle, Solidity Contract, and Ganache. The figure below shows how each element is connected and how this quiz application was able to utilize all these various elements. JavaScript is in charge of all the front-end functionalities and is responsible for user interaction i.e. what the user sees when making/taking a quiz or sending/receiving money. The handshake with Metamask is important as it is responsible for act of sending and receiving money to and from an account address. Ganache is in charge of a starting a local server and mining blocks onto the blockchain. Truffle is the Ethereum testing/developing environment. The Solidity contract is the main sector of this entire web of technology, and is responsible for executing all the solidity functions.



V. IMPLEMENTATION

The versions used all elements of the technology were as follows:

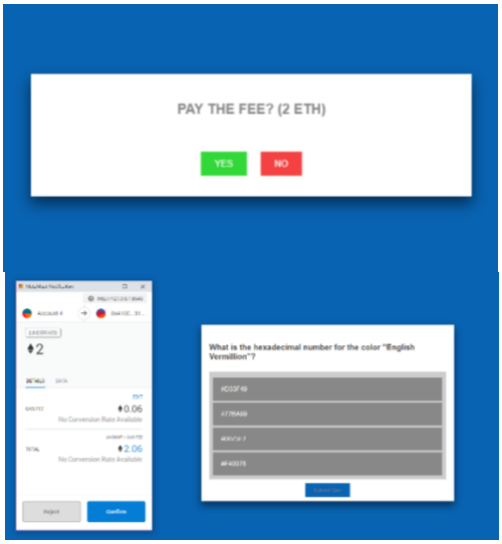
1) *JQuery: 3.3.1*
2) *Web3.js: 1.0*
3) *Metamask: 6.4.0*
4) *Truffle: 5.0*
5) *Ganache: 2.0.0*
6) *Lite-Server: 6.4.1*
7) *Solidity: 0.5.0*

The main interaction for the quiz app was between the Solidity Contract and JavaScript. Whenever a user makes a quiz, takes a quiz, or sends money, that information will be sent from the user interface to the solidity contract. Truffle and Ganache is what allows for blockchain protocol to be implemented and Metamask is pivotal for sending and receiving money to a particular account address.

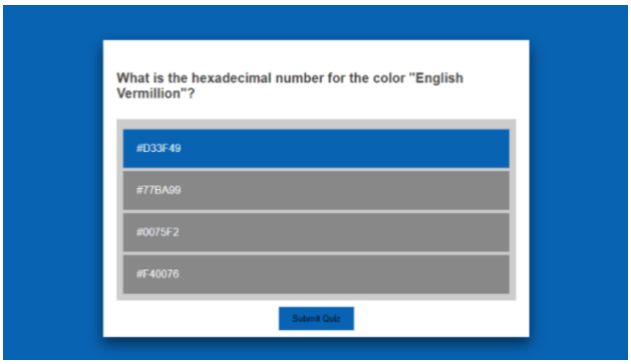
VI. KEY RESULTS/ TEST CASES

Listed below, there are key test cases that represent the functionality of this decentralized application.

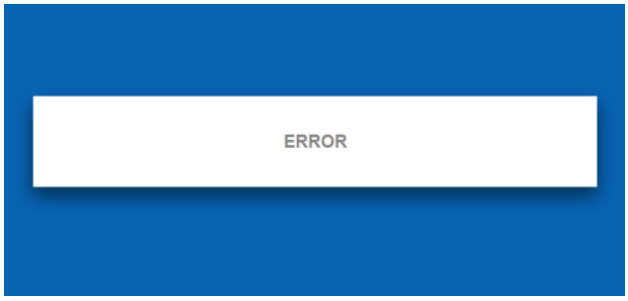
A. Is the user able to submit an entrance fee?



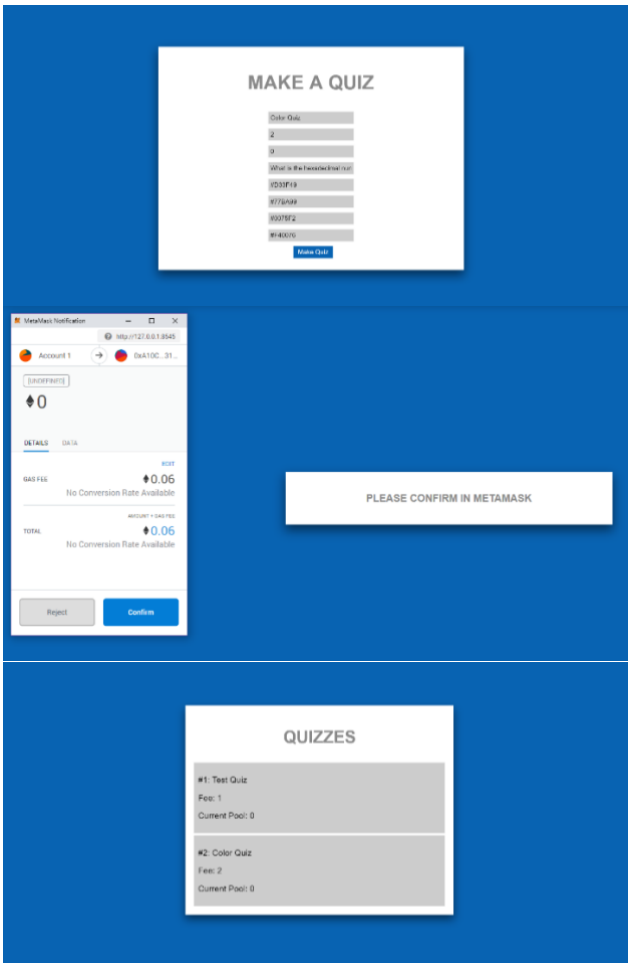
B. Is the user able to retrieve/take a quiz?



C. Is the user able to submit a quiz?



D. Is the user able to successfully make a quiz?



E. Is there a handshake between the Metamask and the decentralized app?

VII. Conclusion

Overall, the quiz decentralized application allows for a user to be able to create or take a quiz. There are three basic segments of this application: Creating a quiz, Taking a Quiz, Sending/Receiving money. These particular algorithms need to have an intricate interaction with all aspects of the technology used(JavaScript, Metamask, Truffle, Solidty, Ganache). In the end, the following aspects of this application worked:

1) *Making a Quiz*
2) *Metamask Connection*
3) *Retrieving Account Information*
4) *Retrieving Available Quizzes*
5) *Selection a particular from a list of quizzes*
6) *Paying a fee to take a quiz*
7) *Retrieving Quiz Question with Answer Choices*

The following are elements that still need to be fixed:

1) *Scoring a Quiz*
2) *Paying the winner*

References

[1] A. Miller and I. Bentov, "Zero-Collateral Lotteries in Bitcoin and Ethereum," 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, 2017, pp. 4-13.

[2] D. Liao and X. Wang, "Design of a Blockchain-Based Lottery System for Smart Cities Applications," 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), San Jose, CA, 2017, pp. 275-282.

[3] T. F. Revano, M. B. Garcia, B. G. M. Habal, J. O. Contreras and J. B. R. Enriquez, "Logical Guessing Riddle Mobile Gaming Application Utilizing Fisher Yates Algorithm," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 2018, pp. 1-4.