Blockchain-Powered Voting App

Ionescu Martin
Ioana Teodora Isar

1. Introduction:

In the ever-evolving landscape of democratic practices, the integration of decentralized voting systems has emerged as a game-changer. With a focus on accessibility and security, the marriage of decentralized technologies with Android applications is poised to reshape the way we participate in electoral processes. This essay explores the innovative synergy between decentralized voting systems and Android apps, emphasizing the potential to create a more inclusive, convenient, and resilient democratic experience.

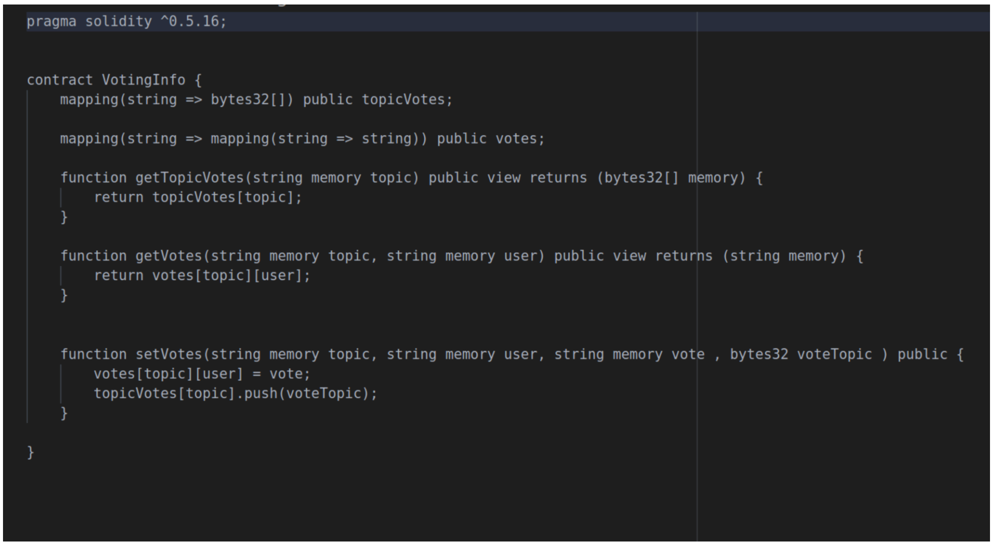
2. Steps towards succes:

We started by documenting the steps that we needed to do to achieve this type of application:

- Create a log-in form into our app
- Create a smart contract that can take votes from users on different topics
- Add the smart contract to our blockchain
- Made the connection between out android app and the blockchain
- Created an ui for our app that interacts with our contract

3. Details for implementation:

- Used ganache local blockchain to simulate oru blockchain for free
- Used https://remix-project.org/ to write our contract that we added in our blockchain.
- Used solc to generate .bin and .abi files that we later use in our android project
- Used web3j to generate a java file using .bin and .abi that we added to our app to make the connectivity with our blockchain



```
pragma solidity ^0.5.16;


contract VotingInfo {
    mapping(string => bytes32[]) public topicVotes;

    mapping(string => mapping(string => string)) public votes;

    function getTopicVotes(string memory topic) public view returns (bytes32[] memory) {
        return topicVotes[topic];
    }

    function getVotes(string memory topic, string memory user) public view returns (string memory) {
        return votes[topic][user];
    }


    function setVotes(string memory topic, string memory user, string memory vote , bytes32 voteTopic ) public {
        votes[topic][user] = vote;
        topicVotes[topic].push(voteTopic);
    }

}
```

- To simulate our voting system we had to improvise our interface for the local blockchain so we have a button that shows the
- contract architecture consists on a double map for filtering: first by topic of voting, and then for the user's vote on the topic.

4. Interface:

- We needed to simulate the balance of the users that are using our apps so we added 2 buttons. One for showing the balance and the other of getting ballance.
- Added button to go to the voting stage where the user is prompted with choices to vote.
- Added a button to show the votings

5. Instructions to run the app:

1. Install and run ganache
2. Find the ip of ganache and replace it in Constants class and in network_security_config.xml
3. Run the app using android studio

6. Resources:

https://docs.web3j.io/4.10.0/transactions/credentials/
https://www.tutorialspoint.com/ethereum/ethereum_ganache_server_settings.htm
https://medium.com/coinmonks/compiling-the-smart-contracts-8dcda8071638
https://docs.soliditylang.org/en/latest/contracts.html