

Correcting The Voting System

Anthony Gonzalez

Biancia Francis

Shunqi Zheng

```
for object to mirror_mod.mirror_object
operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

#selection at the end
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.update()
print("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
= bpy.context.selected_objects[0]
data.objects[one.name].select = 1

print("please select exactly one mirror")

-- OPERATOR CLASSES -----
```

```
types.Operator):
on X mirror to the selected
object.mirror_mirror_x"
mirror_x"
```

Overview

- ❖ The problem statement
- ❖ Proposed Solution
- ❖ Results
- ❖ Smart Contracts
- ❖ ER Diagram of Data Storage and Transfer
- ❖ UML Diagram
- ❖ Cryptophrase Generation
- ❖ UI Representation

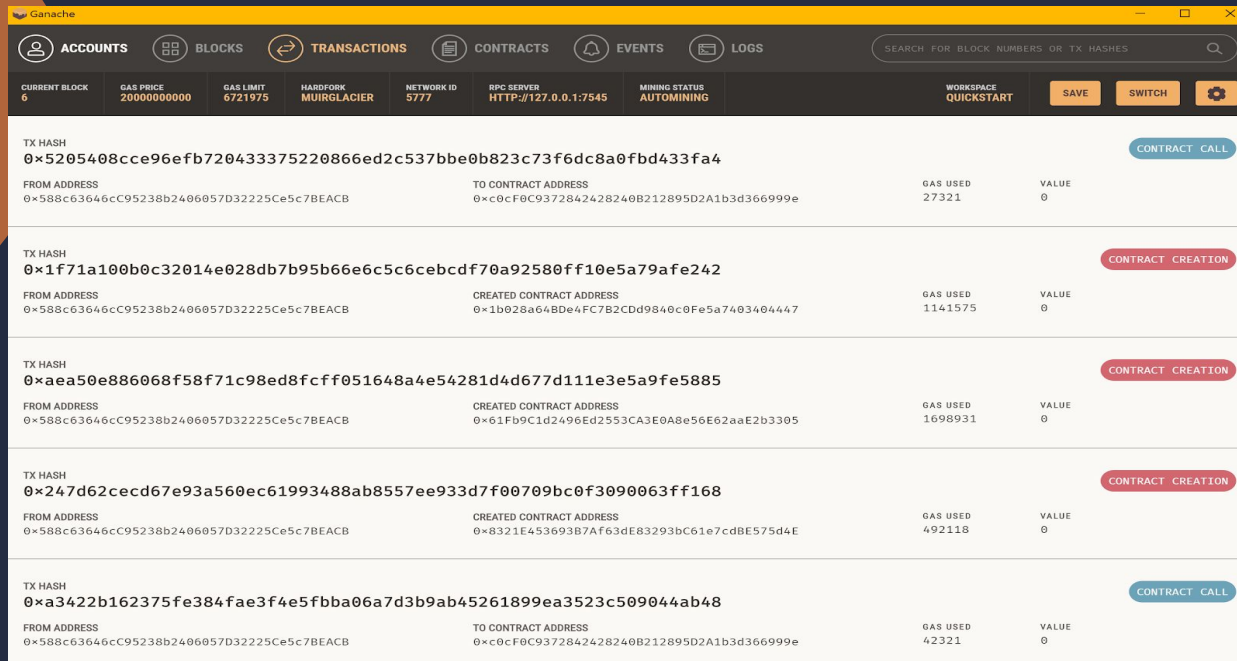
The Problem Statement

- ❖ What our research aimed to resolve was the miscalculation of votes during election periods
- ❖ Voters continue to experience the closed door treatment whereby votes are casted, then authorities alter the received information to benefit a specific candidate
- ❖ The lack of voter visibility and accessibility and there is no room for voters to challenge the final vote count
- ❖ The current voting system is old and dated and as a result, the voting inadequacies continue to occur year over year

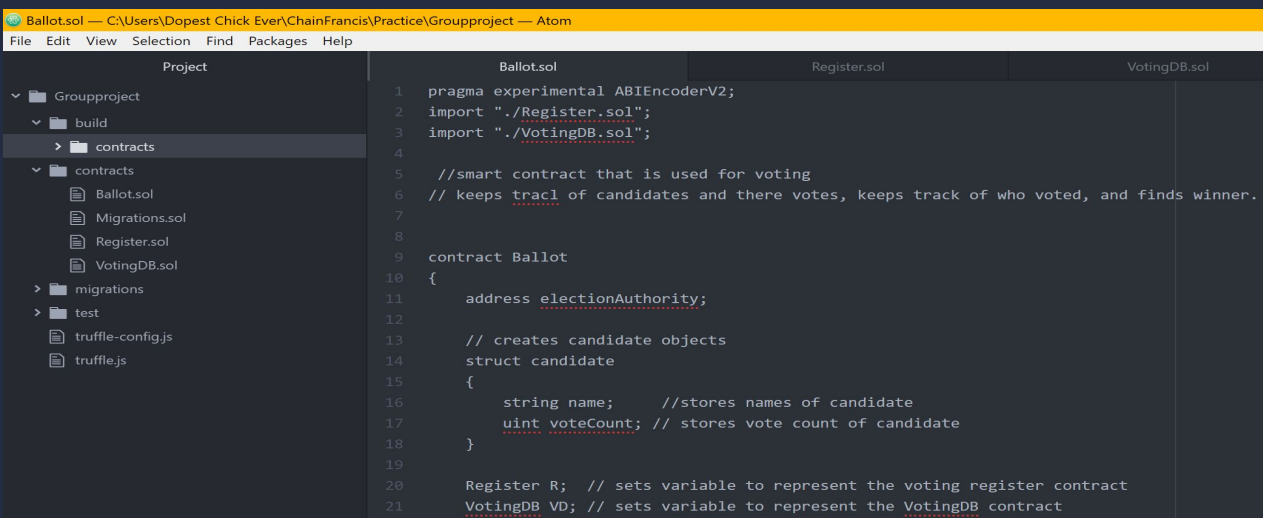
The Proposed Solution

- ❖ We proposed to implement the Blockchain technology of smart contracts, namely; Ballot, Register and Voting, to remove the layer of authorities who are usually responsible for the counting of the ballots
- ❖ Our solution was to designed provide security, increase storage with access to past voting history and for future outlook, provide scalability as well as reliability
- ❖ Crypto keys were also utilized in the process to aid the identification process of each individual re their registered address

Results



TX HASH	FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE	
0x5205408cce96efb720433375220866ed2c537bbe0b823c73f6dc8a0fbd433fa4	0x588c63646cC95238b2406057D32225Ce5c7BEACB	0xc0cF0C9372842428240B212895D2A1b3d366999e	27321	0	CONTRACT CALL
0x1f71a100b0c32014e028db7b95b66e6c5c6cebcdf70a92580ff10e5a79afe242	0x588c63646cC95238b2406057D32225Ce5c7BEACB	0x1b028a648D64FC782CDd9840c0Fe5a7403404447	1141575	0	CONTRACT CREATION
0xaa50e886068f587f1c98ed8fcff051648a4e54281d4d677d11e3e5a9fe5885	0x588c63646cC95238b2406057D32225Ce5c7BEACB	0x61Fb9C1d2496Ed2553CA3E0A8e56E62aaE2b3305	1698931	0	CONTRACT CREATION
0x247d62cecd67e93a560ec61993488ab8557ee933d7f00709bc0f3090063ff168	0x588c63646cC95238b2406057D32225Ce5c7BEACB	0x8321E453693B7AF63dE83293bc61e7cdBE575d4E	492118	0	CONTRACT CREATION
0xa3422b162375fe384fae3f4e5fbba06a7d3b9ab45261899ea3523c509044ab48	0x588c63646cC95238b2406057D32225Ce5c7BEACB	0xc0cF0C9372842428240B212895D2A1b3d366999e	42321	0	CONTRACT CALL



```
1 pragma experimental ABIEncoderV2;
2 import "../Register.sol";
3 import "../VotingDB.sol";
4
5 //smart contract that is used for voting
6 // keeps track of candidates and there votes, keeps track of who voted, and finds winner.
7
8
9 contract Ballot
10 {
11     address electionAuthority;
12
13     // creates candidate objects
14     struct candidate
15     {
16         string name; //stores names of candidate
17         uint voteCount; // stores vote count of candidate
18     }
19
20     Register R; // sets variable to represent the voting register contract
21     VotingDB VD; // sets variable to represent the VotingDB contract
```

- ❖ Three contracts were deployed using powershell and the ganache platform, as well as atom, as the start of a new phase of vote collection
- ❖ Voters had access to their records via a crypto phrase once they registered
- ❖ Query functions are now set up to assist in the management of the data
- ❖ Ballot contract holds all the relevant information for the election

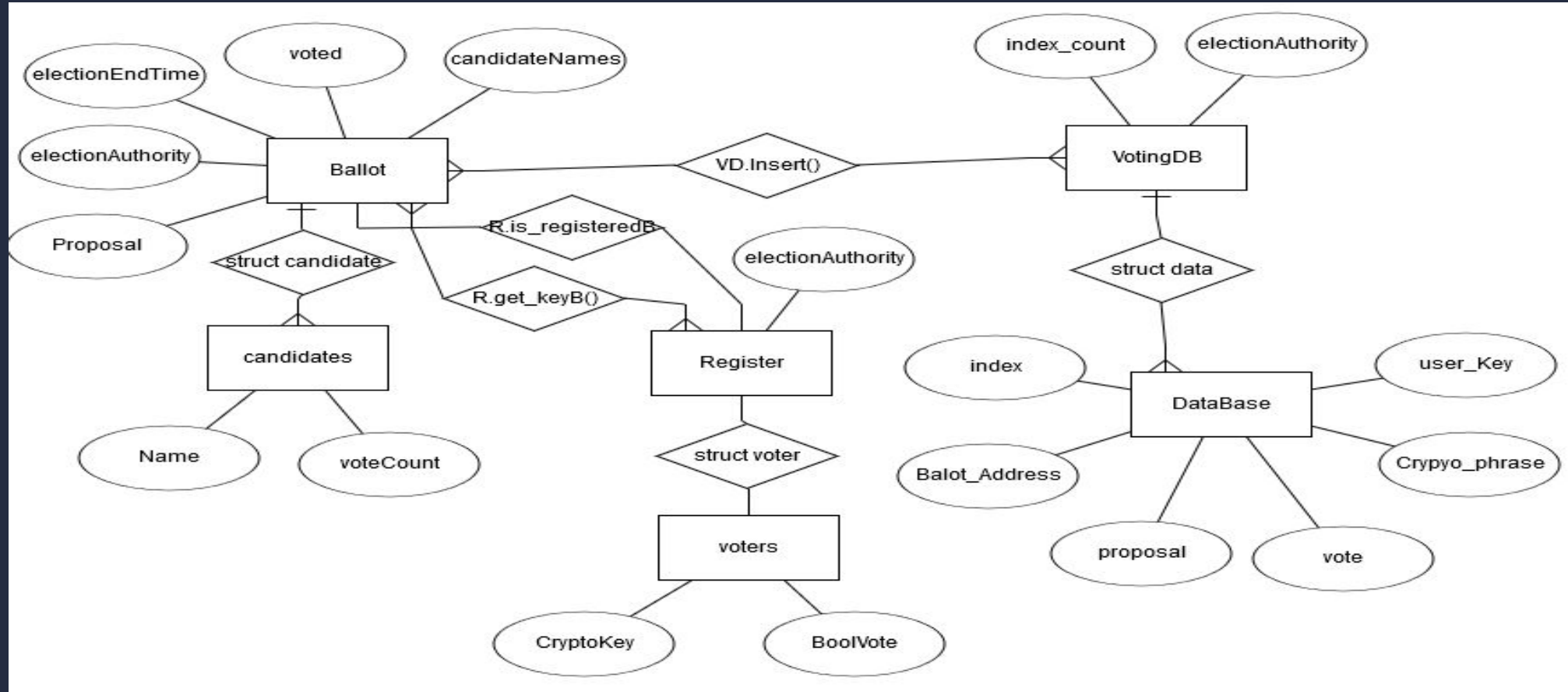
Smart Contracts

To fulfil different aspects such as voter registration, ballot creation/voting, and storage/access of voting history three separate smart contracts were made to help resolve the issues from the problem statement as well as to aid in scalability/reuse.

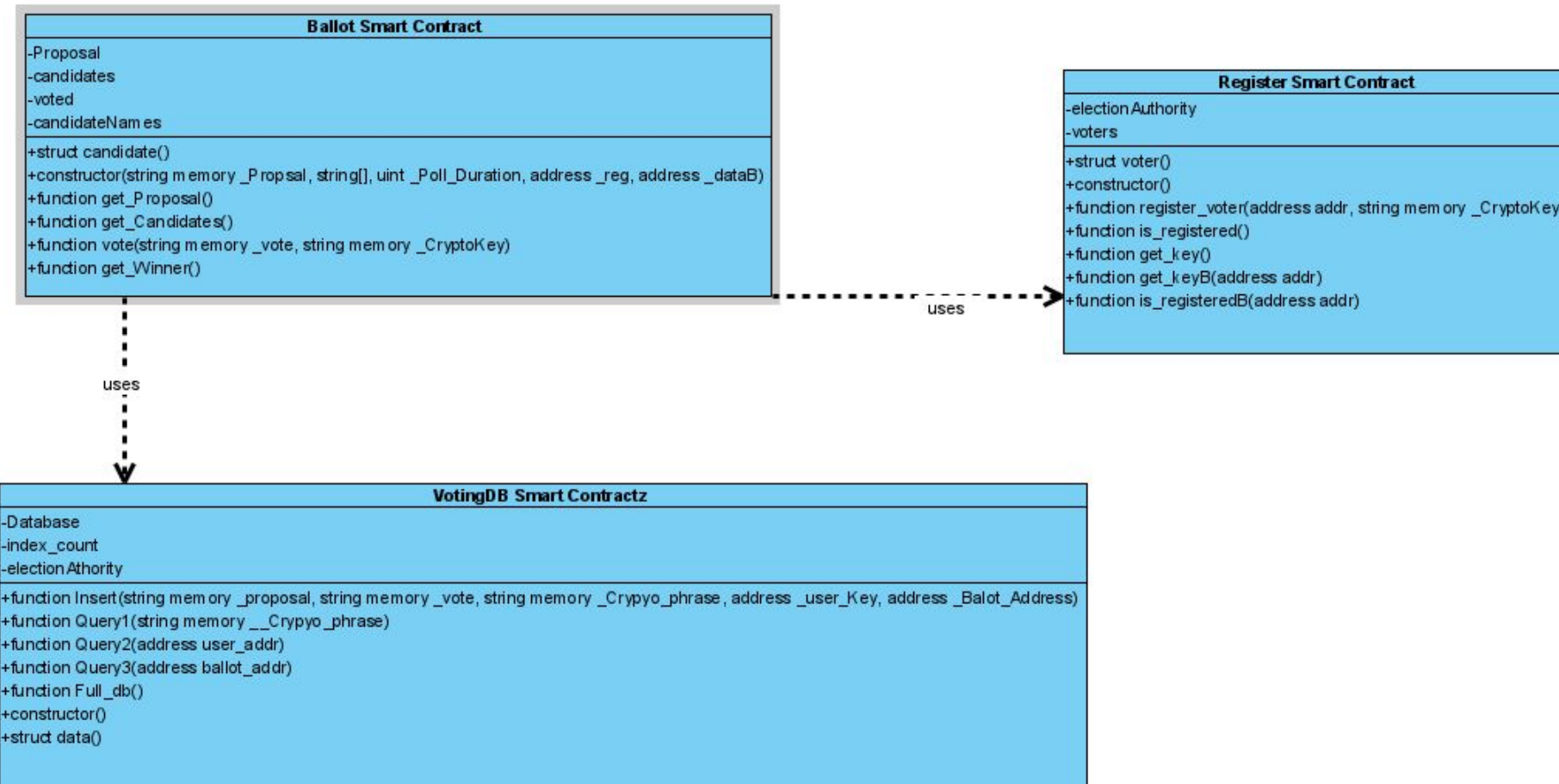
The smart contracts include

- Register.sol
 - Registers voters and stores their voting eligibility and crypto phrase
- Ballot.sol
 - Sets voting parameters, is where voters vote, and counts/sends result.
- VotingDB.sol
 - Stores data from of votes from Ballot.sol and allows users to query the data based on a voters public address, a users crypto phrase, or the address of Ballot.sol

ER diagram of data storage and transfer of smart contracts



UML diagram



- This shows the interactions between the smart contracts
- Ballot.sol uses functions from Register.sol to confirm a voter's credentials then uses a function in votingDB.sol to add votes to the VotingDB.sol database.

Crypto phrase generation

Crypto Phrase Generator

Address

☐ Text ☒ Hexadecimal

Crypto Phrase

Phrases: Words: Bits:

☒ Number ☐ upper case ☐ Include signatures

by John Walker
December, 2005
Updated: March, 2018

For the purposes of this project each voter address was entered into the generator and produced their own unique crypto phrase.

This phrase is then entered into a register voter function with the voters public key.

this is currently done outside this page, but in future iterations it would ideally be done with in the same UI space

This crypto phrase generator utilizes the same backend and a modified front end of John Walker's "Pass Phrase Generator".

which is a public domain, open source project that generates unique pass phrase based on hexadecimal values or string inputs.

UI: Homepage and Sign up

The image shows a web application for an "Election Voting System". The background is a dark blue gradient with a large orange circle on the left. The main content area is a light gray rectangle. At the top left of this area is the text "Election Voting System". To the right are three buttons: "Sign up" (highlighted), "Projects", "About", and "Contact". Below this is a large heading "Election Voting System" and three lines of text: "Easy to use, transparent, fair way", "This is a new technology, a blockchain", and "Secure Online Voting- Government". A "Learn more" button is at the bottom left. A white "Sign up" modal is open in the center. It has a close button (X) in the top right. Below the title are two tabs: "For voters" (active) and "For authority". The form fields are: "Authority name*" (with a red asterisk), "Enter your user name", "Password (only shown once)*" (with a red asterisk), "Enter your password", and "Business Crypto phase: <-----to be continued for phrase----->". At the bottom right of the modal are "Submit" and "Close" buttons. The footer of the page is a dark gray bar with the text "Election Voting System - © 2020 - Blockchain Projects and Anthony Gonzalez Shunqi Zheng Bianca".

Election Voting System

Sign up Projects About Contact

Sign up

[For voters](#) [For authority](#)

Authority name*

Enter your user name

Password (only shown once)*

Enter your password

Business Crypto phase: <-----to be continued for phrase----->

Submit Close

Learn more

B

✓

Grid icon

Clock icon

Election Voting System - © 2020 - Blockchain Projects and [Anthony Gonzalez](#) [Shunqi Zheng](#) [Biancia](#)

UI: For individual

Election VotingSystem

Sign up

Projects


About


Contact


Election Voting System

Your user name:	Edward
Your Crypto phase:	vegetate thurifer heaume misspend heed glassine
Your vote eligibility:	allowed
Your proposal:	Obama
Your poll duration left:	10 hours

B

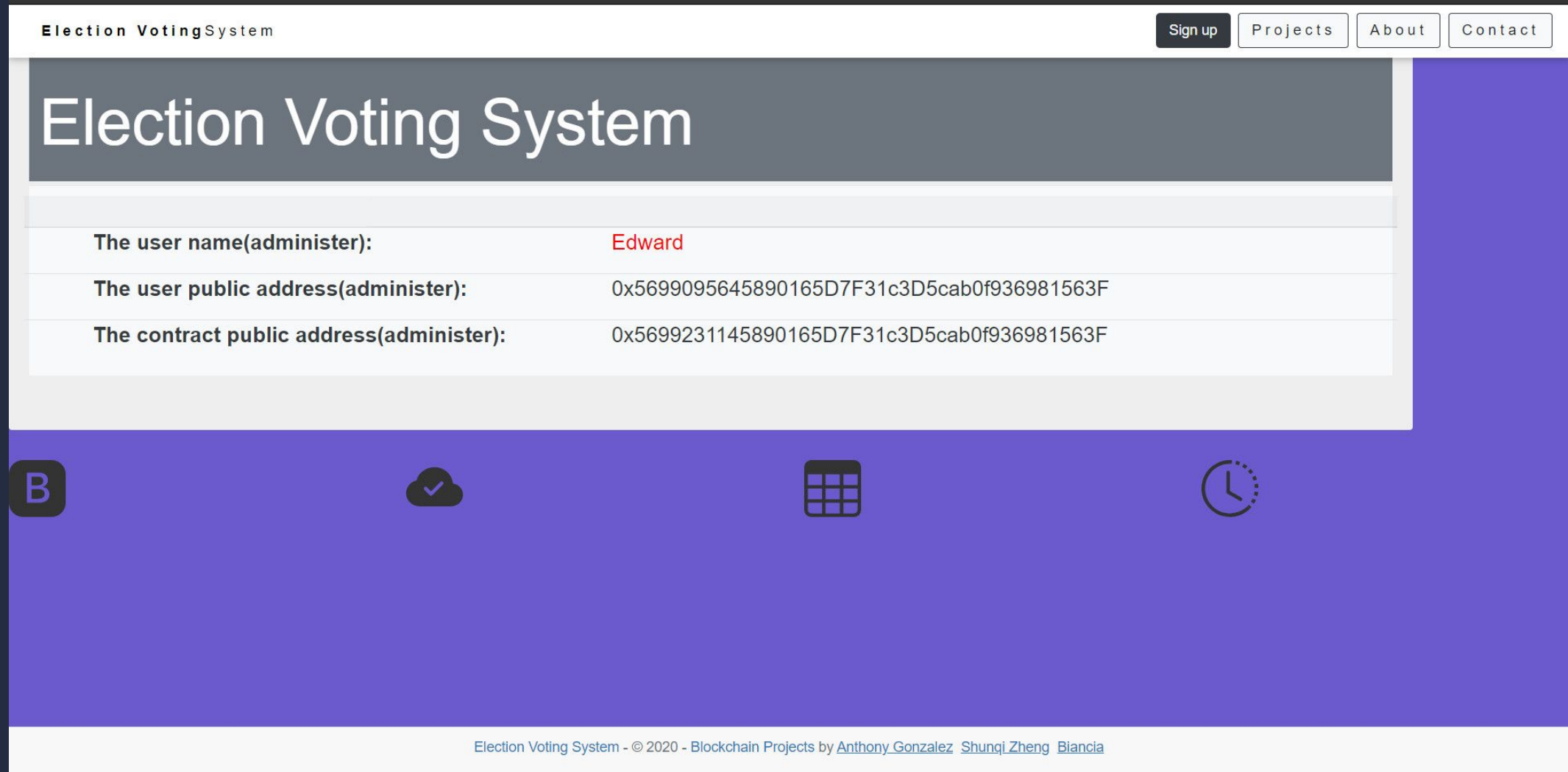






Election Voting System - © 2020 - Blockchain Projects by [Anthony Gonzalez](#) [Shunqi Zheng](#) [Biancia](#)

UI: Voting Database



Conclusion

- ❖ The problem statement
- ❖ The Proposed Solution
- ❖ Results
- ❖ Smart Contracts
- ❖ ER Diagram of Data Storage and Transfer
- ❖ UML Diagram
- ❖ Cryptophrase Generation
- ❖ UI Representation

References

Walker, J. (n.d.). Pass Phrase Generator. Retrieved August 05, 2020, from https://www.fourmilab.ch/javascript/pass_phrase.html

Gürsoy, G., Brannon, C.M. & Gerstein, M. Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts. BMC Med Genomics 13, 74 (2020).
<https://doi.org/10.1186/s12920-020-00732-x>

Rumeysa Bulut, Alperen Kantarcı, Safa Keskin, Şerif Bahtiyar
Blockchain-Based Electronic Voting System for Elections in Turkey
[https://www.researchgate.net/publication/337527299 Blockchain-Based Electronic Voting System for Elections in Turkey](https://www.researchgate.net/publication/337527299_Blockchain-Based_Electronic_Voting_System_for_Elections_in_Turkey)



QUESTIONS?