Blockchain-based Vote System

1. Motivation

In our daily life, we have many votes or questionnaire to finish. Some of them are anonymous, and some are not, as the vote claim. But we can only trust the anonymousness by trusting the host of that vote. If they actually abused the information in your vote, you just have no idea. Sometimes, the people who invoke a vote is different from those who collect the vote. So the vote collectors may change the vote result so to meet their benefits. Now, we introduce a technique called Blockchain Vote System to prevent those situations. Based on blockchain, people can not modify the vote result and the procedure is totally anonymous. And with our blind signature module, vote invoker can control the permission to people who can involve this vote but have no idea who is voting which one the whole procedure.

2. Features and Requirements

1) Web Server

Features:

- Vote

  Vote is an important part of our web app, we provide a series of rich features for vote model.

  I. Create a vote
  User can create a vote by fill in a vote information form.

  II. Participate a vote
  User can participate a vote, as long as he/she is authorized.

  III. Vote information display
  An authorized user can look through vote information which is opened for him or created by him.

  IV. Vote condition real-time update
  Vote condition can be updated in real-time, which means user can see real-time vote number for each candidate.

  V. Vote sharing
  On vote page or after create a vote, user could share vote to others by automatic generation QR code and shared links.

- User management

  For any vote activity, user have different requirement, the quality of service we provided influence user's using experience. We support several functions to increase user experience.

  I. Login
  Before doing operations about vote, User need to login our blockchain vote system.

  II. Logout
  After finishing vote related operations, User could logout.

III. Register
   Allow registration as a member of user.
IV. Guest mode
   Cause some vote is open for all, so this kind of vote allow anyone without account to vote.

- Admin
  As long as users could create and share vote independently, then they could publish something prohibited by law. As a platform, we have duty to audit vote content. Or in some condition, user need adminter's help, so we provided some functions to achieve this requirements.
  I. Delete vote
  II. Delete a vote from vote databases
  III. Alter vote
  IV. Alter vote informations if needed.
  V. Add user
  VI. Create a new user and grant different levels of permission.
  VII. Delete user
  VIII. Delete a user in some special conditions.

Requirements:

- Sponsor
  Sponsors could create a vote and share vote to others. For create a new vote, they could fill in basic vote description, upload candidate informations, set start time and end time, set candidate number, and set vote permissions to participators. Then, they could look through all the vote they created and monitor current vote conditions, such as a certain candidate's vote number.
  At the same time, they are users, they could login, logout, participate other vote activities.
- Participator
  Participators are the main force of vote activities. They could check their vote on blockchain to ensure their vote is not altered by someone. They could see the current and final vote results.They could share a certain vote to their friend with QR code or links. For different vote type, maybe they need to login or not. Anonymous is also needed in some conditions, for instance, they don't want others know their vote results. Also for those participators who join a login-needed vote, if they are not a user before, they could register, if they are, they could login and logout.

2) Block Chain

1. Chain Structure

Features:

- Chain will maintain the result of each vote
- Chain will contain the block information

- Chain will verify whether the block is valid or not
- Chain will store all block and the infomation inside the block (aka. The vote infomation) in database.
- Block is able to verify whether the voteInfo is valid or not.

Requirements:

- Python3.6 or higher
- Sqlite3 database

2. Network Structure

Peer-to-peer (P2P) internet communication is the basic element to implement a block chain-based voting system. Without P2P internet communication, every transaction (voting info in our voting system) and new-packed block will not be transferred. Thus, P2P internet communication is the most fundamental and necessary part in whole voting system architecture.

In P2P internet communication, it should acquire the following feature and requirement:

- In the P2P networks, every node should be an authorized miner and partial of them are connected with the online voting system to receive the vote information from it in time. And every miner node should have a connection with other else miner node.
- In every miner node, a local database is also required for storing current occupied blocks and the un-packed votes.
- Every time a miner receives a vote information, it will broadcast this vote detail to all its one-hop miner neighbors. The receiver should check if this vote info has already existed in its vote pool.
- There is a token passing among all miners. Only the miner who gets the token can pack a block and try to add it to the block chain.
- While a block occupies the token, it gets the right to pack the existing votes in his vote pool to a block, the number of votes in this block depends on a reserved time slot and a vote number threshold. When a block has been packed, token will be passed to next miner and this block will be broadcast to every other miner. Every miner will check the hash-value of this new-arrival block,, if it is valid, then current miner will add it into its own block chain and delete the votes which exist in this block from his own vote pool.
- Every time a new miner logs in, whether it is a new miner or a old miner, it needs to broadcast to each online miner and update his local block chain to the latest one.
- Inter process communication: since the servers will listen given information in other sub-process, and in main-process, we will perform database operations by the data from other process. Thus, we need to use given a inter process communication method to collect all sub-process informations to main-process.

3) Blind Signature

We want

For achieve a totally anonymous voting, we employ a technique named blind signature. The flow is following:

1. The first thing to be clear is that the system must have a centralized verification agency to verify that whether the voter has the right to vote. We call the agency C.
2. Voter A prepares a pair of public and private keys, with private key is s and public key is p.
3. A turns the public key p into p' through a blinding function and sends p' to the central authority C.
4. C checks A's voting eligibility, then signs p' with its private key, gets s', and returns it to A.
5. After A gets s', it can get s through the anti-blind function.
6. It can be proved that s is the signature of the public key p.
7. Now A can vote anonymously with the public keys p and s. Anyone can test that p is a voter who has been verified by the Central Organization C.

The key to this technique is that although the central agency has verified A, he only sees that p' and never seen p. Additionally C has not way to map p from p'.
In the final ballot box (or voting server), only p appears. No one knows who is the public key of p, and can only test that p is a qualified person.
The central authority only signs A once, so A cannot get multiple valid public keys and cannot vote again.
The above blinding and anti-blinding procedures are applicable to commonly used asymmetric cryptosystems such as RSA and elliptic curves. We implement elliptic curves blind signature in this project.

3. Design and Implementation

1) Web Server

- Design: Use mvc frame to achieve our features.
  On front end, We design 9 pages:
    1. Index page as our welcome page, which link to login page,form page,care page.
    2. Login page for user login
    3. Register page for new user join
    4. Form page for fill in vote informations and post
    5. Card page for user to check all the vote belong to himself/herself or have permission to vote. We should each vote as a card.
    6. Share page for user to share certain vote’s QR code and link.
    7. Vote page for user to vote and show current vote results.
    8. Block informations page to show current block details.
    9. Admin page for administrator to manage User, Vote databases.

On back end, we set corresponding handler functions for each page, keep user's log informations on session for better user experience. We use relational data model to store all the data we need.

- Implementation:
    - Framework: Use Django as our mvc frame.
    - Front End: Bootstrap for user-friendly interface. JavaScript and JQuery for front end operations, such as form checking.
    - User Management: Use Django build-in user management model manage user. For users, we use session to record user information for better user experience.
    - Response Functions: For each front page, we set a corresponding response functions.
        - Login functions: return login pages, verify login informations, if login success, redirect to card page.
        - Register functions: return register pages, verify registration validity. If success, redirect to login page.
        - Form functions: return form pages, verify form validity && store vote form in databases. If success, redirect to share pages.
        - Share functions: return share pages, generate share QR code and link which direct to vote pages.
        - Card functions: return card pages, query all vote entry related to current login users and calculate this vote's condition for current user such as "END", "IN PROGRESS" or "NOT START", then return to front end render to card.
        - Vote functions: return vote pages, query all informations about a certain vote and render to front end dynamically, include each candidate's vote number,this vote-user's public key, blind signature.After a user submit his/her vote, the vote submit will be send to minor nodes in this function.
        - Block-infor functions: return block informations pages, query all block's infor and send back to front end.
    - Database: Use relational data model,build three tables:
        - Vote to store vote basic informations.
        - User to store user basic informations.
        - Entry to store vote-user relations, which help us query a user's related vote and a vote's related participator.
        - Selection to store candidates informations. Candidates are the most important part of a certain vote, we use table Selection to store candidate entry's informations.

2) Block Chain

1. Chain Structure
    - We defined four classes in this system, which are Chain, Block, VoteBlock, VoteInfo.
    - Block:

  - Every block is fixed-size at the set up. If without set-up, it is 1MB.
  - Every block contains four fields:
    - Block id: It identifies the block.
    - Pre-hash filed: It stores the hash value of it's previous value.
    - Hash field: It store the hash value of the info field.
    - Info field: it can basically store any kind of data.
  - The hash method can be user-defined before the chain is setup. Once a chain is setup, the hash method is fixed so that miner can verify the validation of each block.
- VoteBlock
  - VoteBlock is inherited from Block, but the info inside VoteBlock can only be VoteInfo objects.
  - The default hash function for VoteBlock is sha256. User can modify it before the chain is created. As with Block, once the chain is created, the hash function cannot be modified.
- VoteInfo
  - VoteInfo has 7 field:
    - Voteinfo id: it identifies whether the voteInfo id in database.
    - Block id: it identifies which block this voteInfo is belonged to. If the voteInfo has not been added to a block, the fied is NULL as default.
    - Timestamp: this field records when did the voteInfo created. This timestamp is collected from timestamp server.
    - Target: this field identified which vote is this voteInfo belonged to. Since there may be multiple vote ongoing within our system. We use this field to distinguish which voteInfo is for which vote.
    - Pubkey: this field is the public key of the voter.
    - Sign: this info is the ellipse signiture of the info field. We use this field to verify whether the sign is signed by the vote by checking the public key.
    - Info: this field identifies which question and which option did the voter choose.
    - Status: this fied identifies whether the vote is added to a block. 1 means yes and 0 means no.
- Chain
  - Basically chain is a collection of blocks. We implemented this chain on VoteBlock.
  - Chain is also responsible for maintaining all the information in database.
  - The database has for sheets:
    - Block: maintains the information with every block.
    - Vote: maintains the information with every voteInfo.
    - Result: maintains the result information of every vote.

        - Chain maintains the result information of a specific vote in database. If a user conducts a multiple-vote, by default the last one will be accepted.
- Verification
    - When a new block is to add to chain, chain will verify whether its hash is corresponding to its info according to its hash function. It will also verify whether the prehash is the same as last block's hash.
    - When a new voteInfo is to added to voteBlock, the voteBlock will verify whether the voteInfo is correctly signed.

2. Network Structure

The overall P2P network requires two servers: first is to receive vote information from http; second is to communicate with other miners, including broadcast votes and blocks.

- Inter process communication: We apply the producer-consumer model to achieve inter process communication. In the producer end, i.e., server's sub-process, the vote information and block information will pass into a queue. In the consumer end, i.e., main process, if we want to do database operations, we only need to get all elements in the queue.
- Http server: we use package socktio() to listen http request and extract the information in it. The website will issue a "vote" http request to server, which contains the voter's pubkey, selection ad sign through json. We decode the http request and verify whether the vote information is valid. If so, we added to our database with status to be 0, and broadcast to other miners. If not, we drop it.
- The inter-miner communication message format: <operation type><sender address><detail information>
- Inter-miner server: We use package TCPServer() to communicate with other miners. In TCPServer(), there is a handler() to do assigned operations while received data from other miners. In handler(), it will handle three types of messages. First is the request of latest block chain, the handler will check its own miner latest block id via inter-process communication and compare it with the required id; Second is receiving broadcast vote, the handler first check if this vote has been sent already and then will pass it to main thread and check the existence of this vote. Third is receiving the broadcasted block, handler will pass it to main-process and its legality will be checked there.

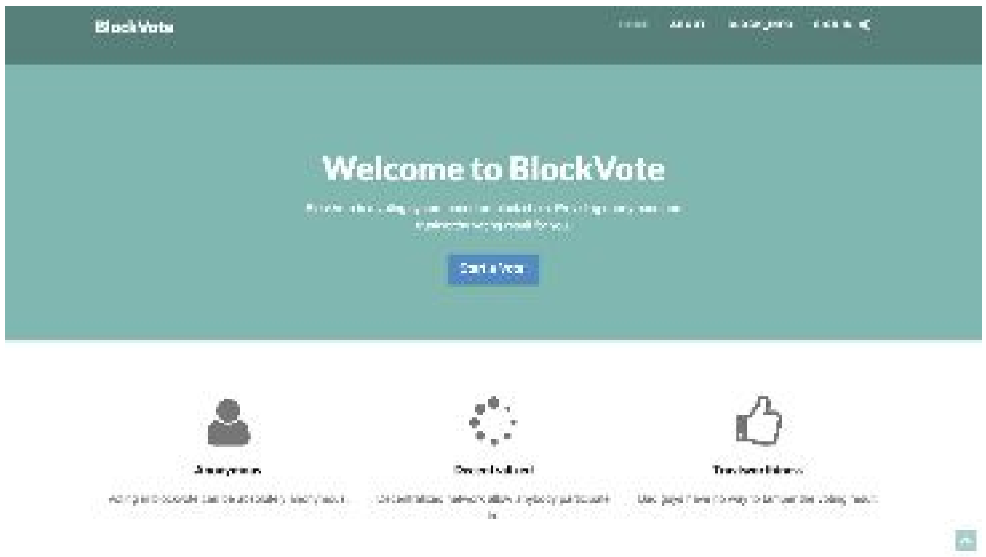
3) Blind signature

4. User Interface

We designed user interfaces for both PC and mobile users. The main six user interface designs are listed as follows.

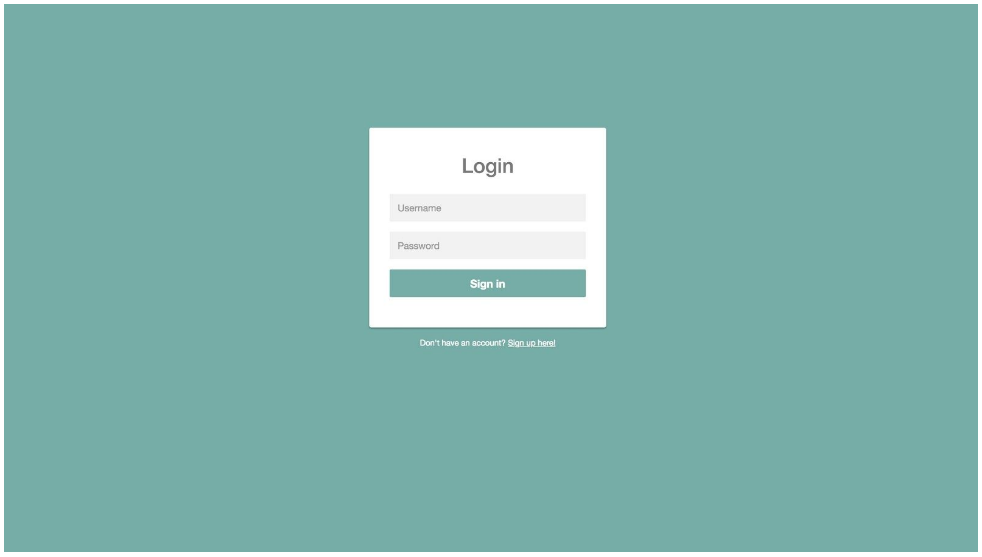
1. For PC Users
    - Homepage

      The homepage gives the brief introduction of our system - a voting system based on blockchain, providing anonymous and trustworth voting result for users. It describes the three feathers: anonymous, decentralized and trustworthiness, and list the developers of the system.

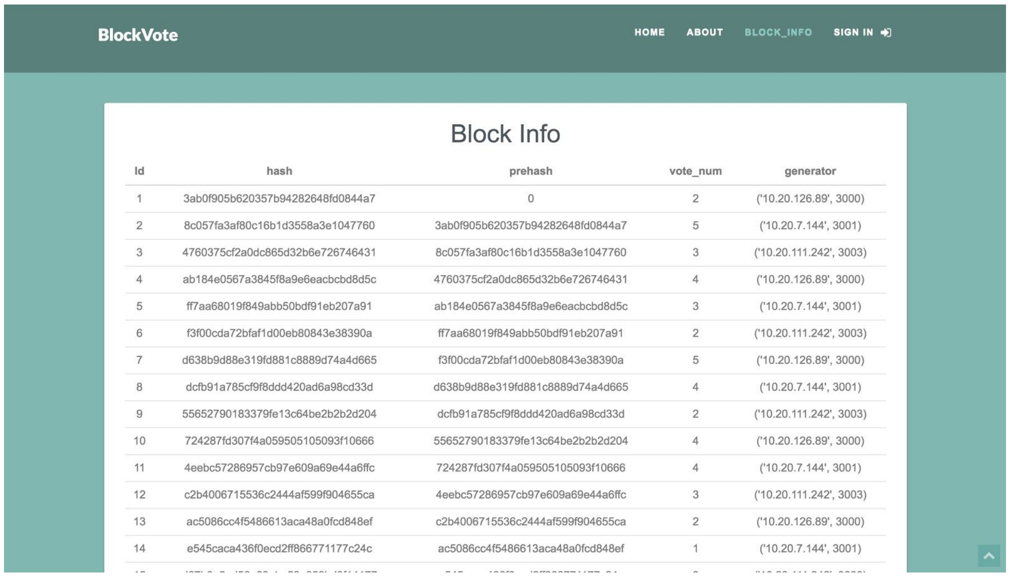
    - Login Page

      If a user want to participant in a vote which is open to everyone, he/she does need to login；If a user would like to participant in a privileged vote, he/she need to login to ensure its privilege; If a sponsor want to start a new vote, he/she need to login.



- Block Information Page

  This page give the basic information of blocks in blockchain. The basic information includes id, hash, prehash, vote number and generator machine of a block. By clink on one block, we can get the specific information of every voting recorded in this block.



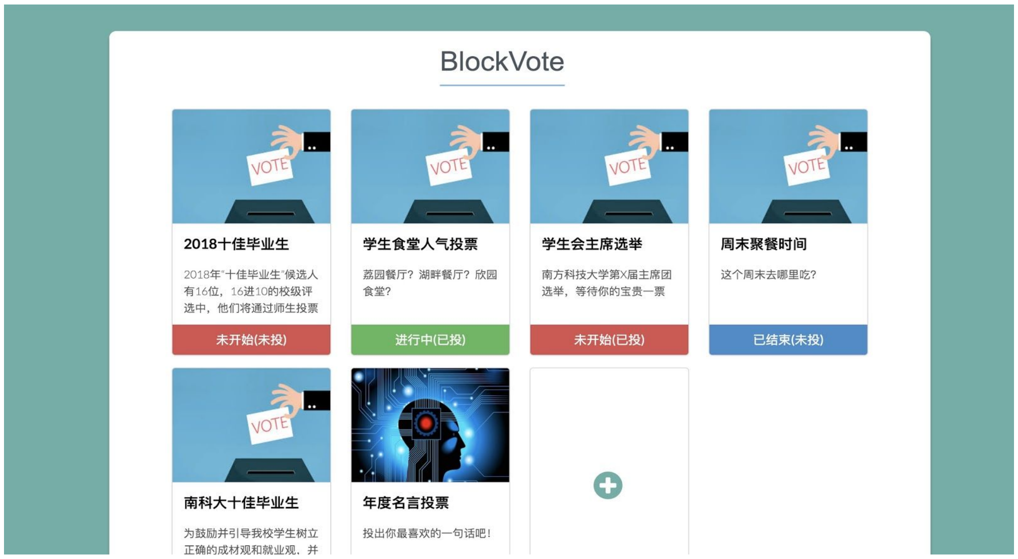
Block Info

Id	hash	prehash	vote_num	generator
1	3ab0f905b620357b94282648fd0844a7	0	2	('10.20.126.89', 3000)
2	8c057fa3af80c16b1d3558a3e1047760	3ab0f905b620357b94282648fd0844a7	5	('10.20.7.144', 3001)
3	4760375cf2a0dc865d32b6e726746431	8c057fa3af80c16b1d3558a3e1047760	3	('10.20.111.242', 3003)
4	ab184e0567a3845f8a9e6eacbcbd8d5c	4760375cf2a0dc865d32b6e726746431	4	('10.20.126.89', 3000)
5	ff7aa68019f849abb50bdf91eb207a91	ab184e0567a3845f8a9e6eacbcbd8d5c	3	('10.20.7.144', 3001)
6	f3f00cda72bfaf1d00eb80843e38390a	ff7aa68019f849abb50bdf91eb207a91	2	('10.20.111.242', 3003)
7	d638b9d88e319fd881c8889d74a4d665	f3f00cda72bfaf1d00eb80843e38390a	5	('10.20.126.89', 3000)
8	dcfb91a785cf9f8ddd420ad6a98cd33d	d638b9d88e319fd881c8889d74a4d665	4	('10.20.7.144', 3001)
9	55652790183379fe13c64be2b2b2d204	dcfb91a785cf9f8ddd420ad6a98cd33d	2	('10.20.111.242', 3003)
10	724287fd307f4a059505105093f10666	55652790183379fe13c64be2b2b2d204	4	('10.20.126.89', 3000)
11	4eebc57286957cb97e609a69e44a6ffc	724287fd307f4a059505105093f10666	4	('10.20.7.144', 3001)
12	c2b4006715536c2444af599f904655ca	4eebc57286957cb97e609a69e44a6ffc	3	('10.20.111.242', 3003)
13	ac5086cc4f5486613aca48a0fcd848ef	c2b4006715536c2444af599f904655ca	2	('10.20.126.89', 3000)
14	e545caca436f0ecd2ff866771177c24c	ac5086cc4f5486613aca48a0fcd848ef	1	('10.20.7.144', 3001)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

- Vote List Page

  After user login, this page shows all the votes related to this users. They may even be started by the user. There are three state of the vote: not started, ongoing, have finished. User can distinguish the three state easily be their colors.

Also, user can start a new vote by clicking the add icon.



BlockVote

2018十佳毕业生
2018年"十佳毕业生"候选人有16位，16进10的校级评选中，他们将通过师生投票
未开始(未投)

学生食堂人气投票
荔园餐厅？湖畔餐厅？欣园食堂？
进行中(已投)

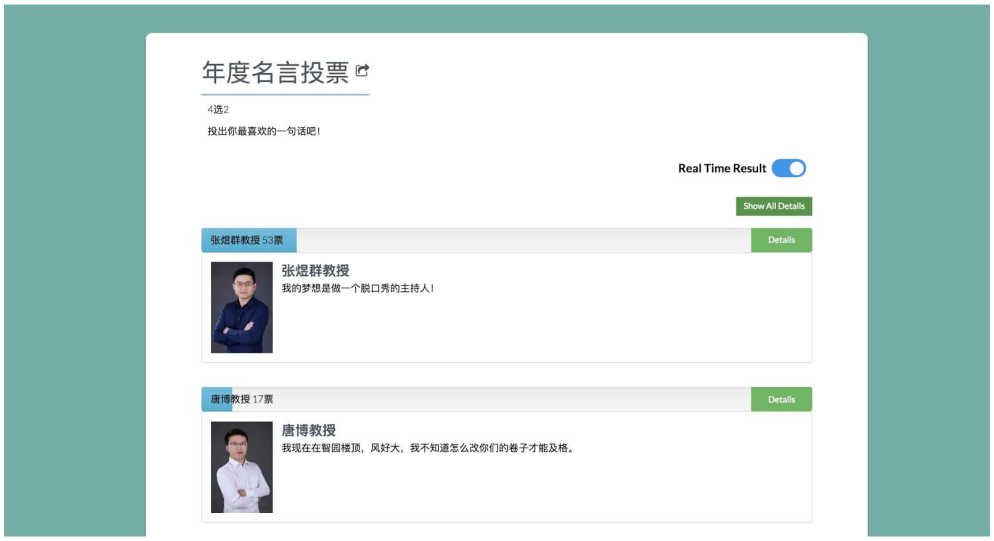
学生会主席选举
南方科技大学第X届主席团选举，等待你的宝贵一票
未开始(已投)

周末聚餐时间
这个周末去哪里吃？
已结束(未投)

南科大十佳毕业生
为鼓励并引导我校学生树立正确的成材观和就业观，并

年度名言投票
投出你最喜欢的一句话吧！

- Voting Page
  This is the page for voting. It can be shared by a QR code, so that the participants can vote easily. Besides, for each option, we give a bar to present the current votes number of it vividly, and user can view the real time result of the vote. In addition, we give the user the public key, private key and blind public key (used in totally anonymous vote) for this vote. Therefore, if user want to check whether their vote is recorded in the blockchain successfully, their can verify it by their public key.



年度名言投票
4选2
投出你最喜欢的一句话吧！

Real Time Result

Show All Details

张煜群教授 53票 Details

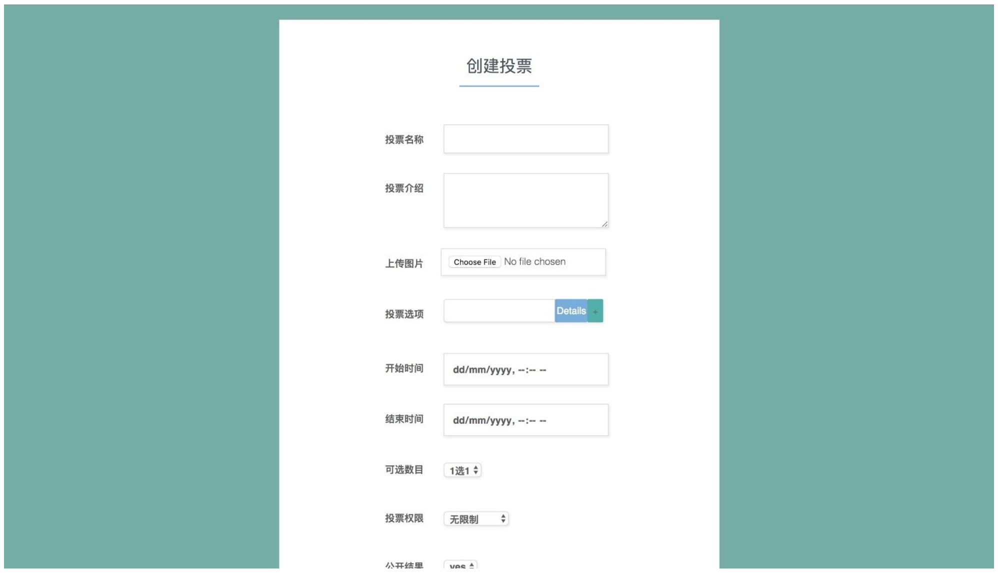
张煜群教授
我的梦想是做一个脱口秀的主持人！

唐博教授 17票 Details

唐博教授
我现在在智园楼顶，风好大，我不知道怎么改你们的卷子才能及格。

- Start Vote Page
  In this page, user can start a new vote. There are many information need to fill:

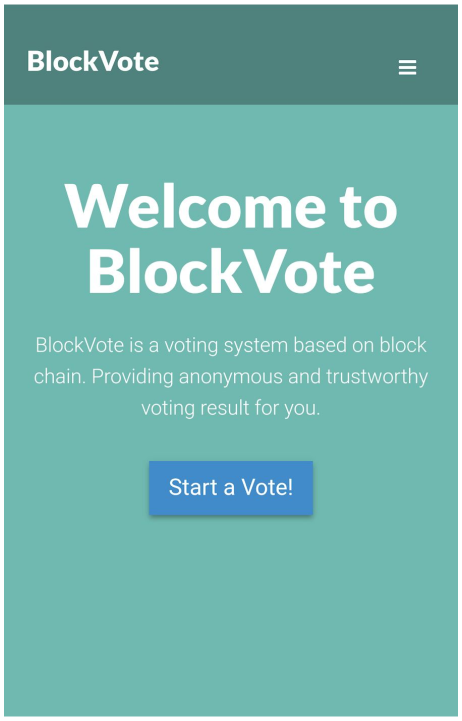
the vote name, the vote description, the vote options, vote date range and so on. It is noted that the anonymous and the privilege of a vote can be customized by the user.



创建投票

投票名称	
投票介绍	
上传图片	Choose File No file chosen
投票选项	Details +
开始时间	dd/mm/yyyy, --:-- --
结束时间	dd/mm/yyyy, --:-- --
可选数目	1选1
投票权限	无限制
公开结果	yes

2. For Mobile Users
   All the website page interfaces are also adjusted in mobile device, and the main features and functions are maintained perfectly.
   - Homepage & Login Page

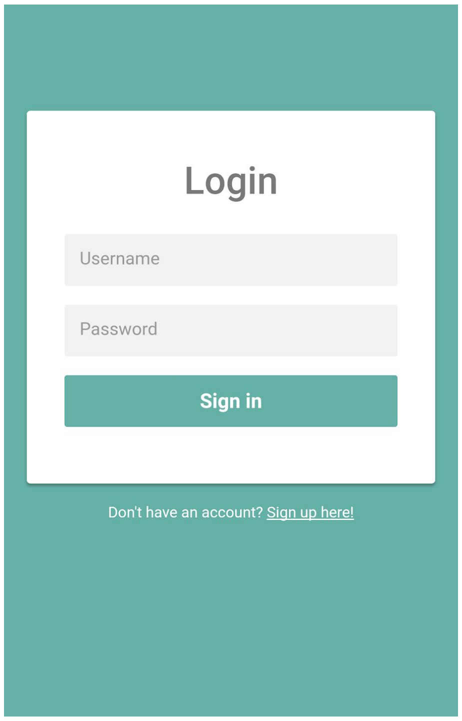


BlockVote ≡

Welcome to BlockVote

BlockVote is a voting system based on block chain. Providing anonymous and trustworthy voting result for you.

Start a Vote!



Login

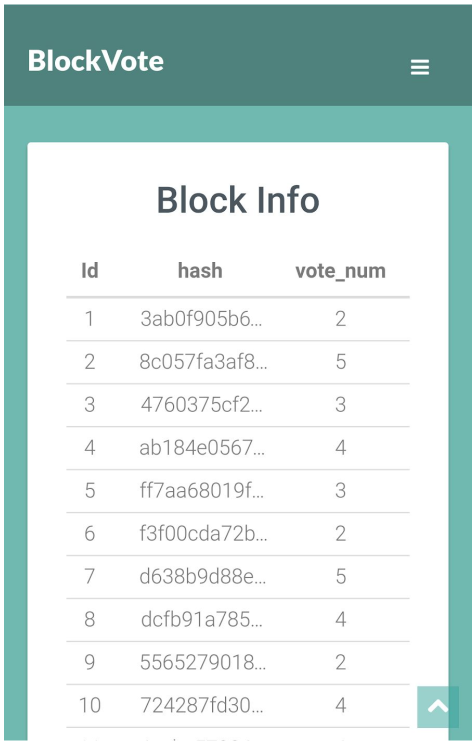
Username

Password

Sign in

Don't have an account? Sign up here!

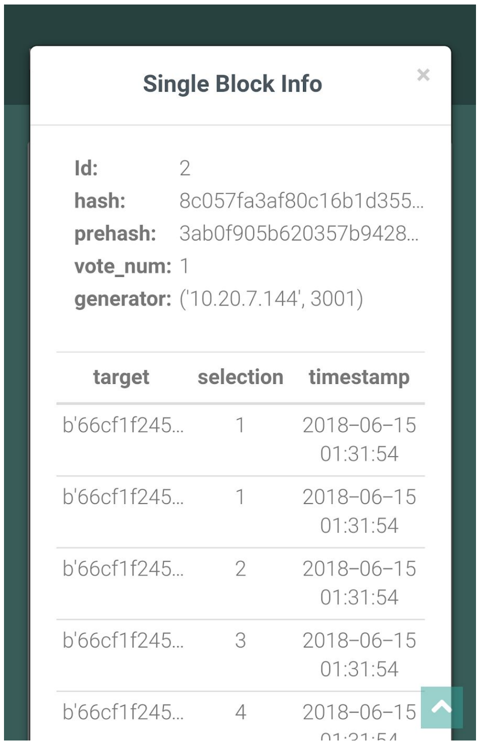
- Block Information Page



BlockVote

Block Info

Id	hash	vote_num
1	3ab0f905b6...	2
2	8c057fa3af8...	5
3	4760375cf2...	3
4	ab184e0567...	4
5	ff7aa68019f...	3
6	f3f00cda72b...	2
7	d638b9d88e...	5
8	dcfb91a785...	4
9	5565279018...	2
10	724287fd30...	4

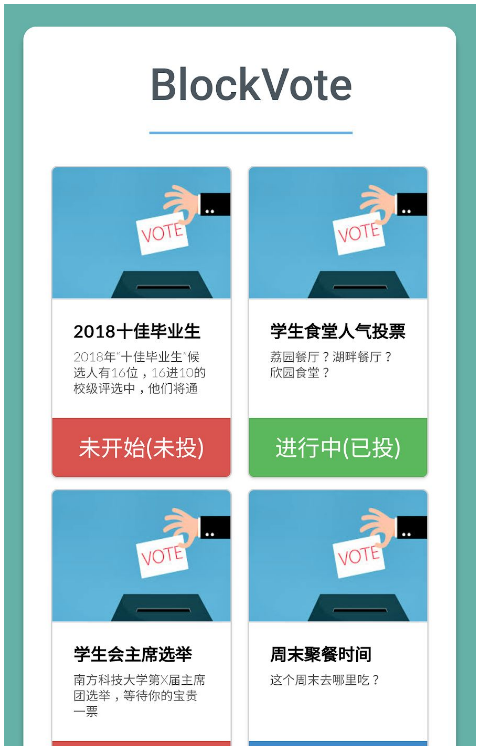


Single Block Info

Id:	2
hash:	8c057fa3af80c16b1d355...
prehash:	3ab0f905b620357b9428...
vote_num:	1
generator:	('10.20.7.144', 3001)

target	selection	timestamp
b'66cf1f245...	1	2018-06-15 01:31:54
b'66cf1f245...	1	2018-06-15 01:31:54
b'66cf1f245...	2	2018-06-15 01:31:54
b'66cf1f245...	3	2018-06-15 01:31:54
b'66cf1f245...	4	2018-06-15

- Vote List Page & Voting Page



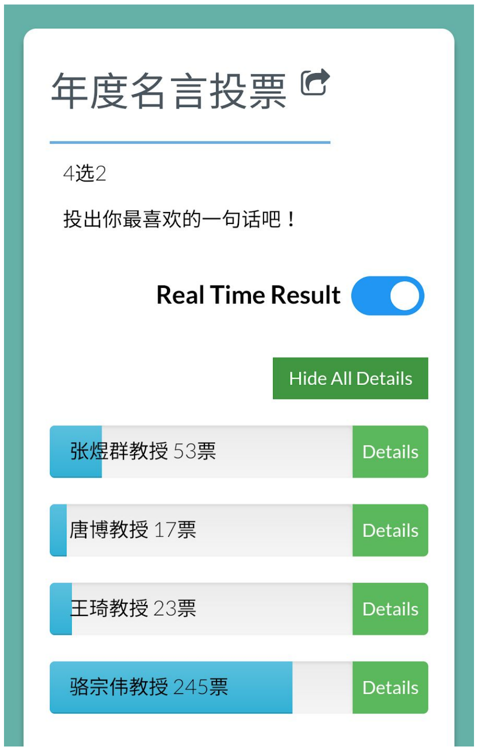
BlockVote

2018十佳毕业生
2018年"十佳毕业生"候选人有16位，16进10的校级评选中，他们将通
未开始(未投)

学生食堂人气投票
荔园餐厅？湖畔餐厅？欣园食堂？
进行中(已投)

学生会主席选举
南方科技大学第X届主席团选举，等待你的宝贵一票

周末聚餐时间
这个周末去哪里吃？



年度名言投票

4选2

投出你最喜欢的一句话吧！

Real Time Result

Hide All Details

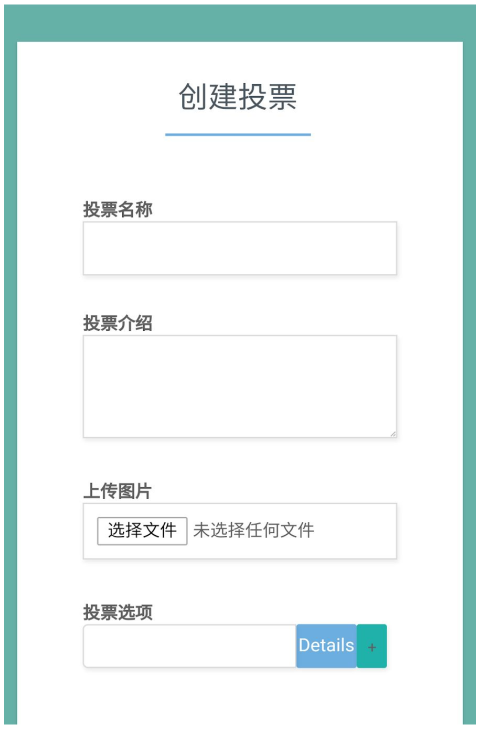
张煜群教授 53票 Details

唐博教授 17票 Details

王琦教授 23票 Details

骆宗伟教授 245票 Details

- Start Vote Page & Share QR Code



创建投票

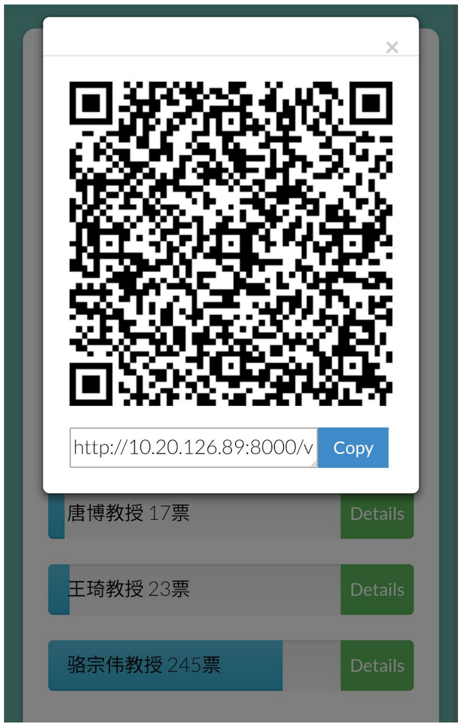
投票名称

投票介绍

上传图片

选择文件 未选择任何文件

投票选项



http://10.20.126.89:8000/v

唐博教授 17票

王琦教授 23票

骆宗伟教授 245票