Ideas for voting in an association or club.

Exercise : a vote must be taken on a board or subject.
Suppose : 100 members.
How can this be done anonymously?
Assuming everyone has email.
Voting is done on a web page.

The simplest solution.

- Generate 100 unique tokens in a table in a database with this command :
- `$token = bin2hex (openssl_random_pseudo_bytes ( 10 )`);
- Send each member an email with a random unique token.
- So member A doesn't get token 1, member B doesn't get token 2, etc.
- But member A gets token 43, member B gets token 87, etc.
- There is no link between the member table and database table with tokens.
- Consider separate systems: member database <> voting database.
- The e-mail contains a link to the webpage where you can vote :
- `http://www.domein.nl/stemmen.php?token=c3e0814b8fb61203117b`

On the webpage, the token is checked in the database.
If the token does not exist (or does not match) one cannot vote.
As long as the ballotbox is open, you can vote.
You can vote as often as you want.
Every time a vote is cast, the vote is saved.
The last vote is decisive and counted.
Voter sees votinghistory with every new votingattempt or if checked.
After the ballot box has been closed, the votes are counted.
Voter can then only see votinghistory.
The "turnout" is determined by counting the tokens voted on.

Other way

As above, but token is generated during email creation.
Thus, a table of tokens is not generated in advance.
During voting the token is stored and just like before one can vote as many times as one wants to until the ballotbox closes.
The disadvantage is that an miscreant can create and use any token that looks real.
Solution is to encrypt token and use encrypted token in link.
See this example :

```
http://bytes.egestas.net/company/
```

Result:

```
http://bytes.egestas.net/company/receive/?login=YkRlblpvbndaQ2RFeCtpY09QS09BQT09
```

On the web page, in this case, the login decrypts and original token is used to vote and the vote is saved.

Determining the result

After closing the ballotbox, all tokens that have been voted on are added up.
Only the last vote cast is counted.

Example MySQL query in PHP :

```
$query = "CREATE TEMPORARY TABLE IF NOT EXISTS voting result AS (SELECT
token, vote, max (date), count (vote), hash FROM ballot group by token
DESC order by vote);";

$ result = $ db-> query ($ query);
$ query = "SELECT vote, count (vote) AS number FROM VOTE RESULT GROUP BY
vote ORDER BY number DESC, vote; "; $ result = $ db-> query ($ query);
```

The number of votes cast can never be more than the number of members. Check.
Turnout can be determined by comparing the number of votes cast with the number of members.
Or by comparing used tokens with the number of generated tokens.
Statistics can also be used to count how many times have been voted.
Extra options can be added to 'ballot paper' such as province, m/f,
agecategory, etc., so that it can also be used to determine how people voted.
With few members, this kind of information can lead to identification.

Other

Extensive testing in advance prevents errors and malfunctions during actual tuning.
Members, especially the elderly, must be prevented from making mistakes and complaining.
So :

- No computer? It is 2020. Ask the neighbors.
- E-mail not received: unfortunate. Look at the spam.
- E-mail with link lost / deleted: unfortunate. Look at deleted e-mail.
- Link that does not work: too bad. Try a different browser.
- Page not working in browser: unfortunate. Try a different browser.

Basically, the problem on the user side is his / her problem.

Because the person entitled to vote and the token are not linked, and it is therefore unknown who is whose token.

No new token can be issued.

Undoubtedly it is possible to see in a log of sent e-mails who has received what.

You can then resend the token (unwanted).

Assign token to someone else. (Voters are going to complain about a vote they didn't cast.)

You can then see, with some searching, who voted what.

The solution is then to separate member database, email logs and voting database with tokens.

Initially, it is based on honesty.