

Vilnius University
Faculty of Mathematics and Informatics
Master's Degree Program of Software Engineering
Course 1, Group 1

Internet Voting System Requirements Specification

Document Authors:

Kęstutis Matuliauskas

Marius Slovikas

Version 1.3

February 10th, 2019

Table of contents

1	Introduction	3
2	Business Level Requirement.....	4
2.1	Why we need to build this system? (Vision statement, product benefits, business mission)....	4
2.2	How we will implement our vision statement / strategic goals? (<i>Product possibilities tree; goal tree, goal model, goal decomposition</i>).....	4
2.3	What objects and information models product (internet voting system) uses, analyses, adopts? (<i>Business objects / informational models, business object relations, business rules</i>)	6
2.4	Who will be the users of the services & features of this product (internet voting system)? (<i>Business partners, buying organization's / company's employees, classes of end-users, their rights & permissions</i>)	7
2.5	Where and how the product (internet voting system) can be accessed to end-users and buying organization after purchase? (Access types, access points, desired devices, desired applications, system distribution & system shredding).....	8
2.6	How fast product must perform to fulfill desired operational (short-term) goals? (<i>User action timing, business transaction's timing, product performance requirements, efficiency requirements, SLA recommendations</i>).....	11
3	End-User requirements.....	12
3.1	Which tasks and why end-user will be performing? (<i>Use cases, tasks</i>).....	12
3.2	How deeply the system has to be digitalized?	13
3.3	What business and informational objects has to exist to perform end-user tasks? (<i>Business objects, informational objects, key database tables</i>)	13
3.4	Who and with what skills will be eligible to use the product? (<i>End-User qualification requirements, UI requirements, usability requirements, security requirements</i>)	13
3.5	Where and which internet voting tasks can be performed by end-user? (<i>Requirement to voting devices and locations</i>).....	13
3.6	How fast end-user tasks has to be performed? (<i>Timing</i>)	14
4	Software System Requirements Level	15
4.1	Software system requirements	15
4.2	Project-bases software system requirements	15
4.3	Implementation requirements for software system.....	15
5	System design sketches	16
5.1	Step of choice.....	16
5.2	Immediate voting process	17
5.3	Process of learning more.....	19
6	Web design proposal for potential buying organizations.....	22

1 Introduction

The requirement specification document consists of sections for requirements, based on modified Zachman-framework table.

	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Objectives / Scope	List of things important to the enterprise	List of processes the enterprise performs	List of locations where the enterprise operates	List of organizational units	List of business events / cycles	List of business goals / strategies
Model of the Business	Entity relationship diagram (including m:m, n-ary, attributed relationships)	Business process model (physical data flow diagram)	Logistics network (nodes and links)	Organization chart, with roles; skill sets; security issues.	Business master schedule	Business plan
Model of the Information System	Data model (converged entities, fully normalized)	Essential Data flow diagram; application architecture	Distributed system architecture	Human interface architecture (roles, data, access)	Dependency diagram, entity life history (process structure)	Business rule model
Technology Model	Data architecture (tables and columns); map to legacy data	System design: structure chart, pseudo-code	System architecture (hardware, software types)	User interface (how the system will behave); security design	"Control flow" diagram (control structure)	Business rule design
Detailed Representation	Data design (denormalized), physical storage design	Detailed Program Design	Network architecture	Screens, security architecture (who can see what?)	Timing definitions	Rule specification in program logic
(Working systems)						
Function System	Converted data	Executable programs	Communications facilities	Trained people	Business events	Enforced rules

2 Business Level Requirement

2.1 Why we need to build this system?

(**Vision statement, product benefits, business mission**)

Business mission – by converting paper voting to internet voting, make elections around the world the election to become more fair, attract more young people to vote and gain more trust & security to the voting process.

Vision statement / strategic goals – internet voting system is dedicated to local governments (city municipalities, county & state governments) around the world to empower them to improve their voting process (make it cheaper to run, quicker to count votes) and advance into digital democracy by using benefits of digital democracy:

- ✓ Involve into voting new generation young citizens, who prefer voting online, and would probably not got to regular voting lines,
- ✓ Addressing global world changes, when many people are travelling abroad so often, and are away during the voting dates,
- ✓ Ensure that disabled people, or those are hard to walk are not going to be affected by postman visiting the voter – as the systems who allows to vote online and are based on “keystroke dynamics” technology (unique key stroke patterns as their digital signature) ensures that exact voter is voting himself only, and because he is doing that online, and only the last vote is counted, nobody can affect his vote.

Product benefits. The main benefits of this system, is that it based on well-known and widely-accepted coding standards and technologies (*i.e. S.O.L.I.D. MVC, PSR-2, PSR-4 Autoloaders coding standards, BCNF database structure, WordPress content management system, PHP programming language, MySQL & MariaDB relational database management systems (RDBMS), Windows & Ubuntu Linux servers with WAMP/LAMP stacks*) – this guarantees the fact of understandability of the system by many people who will vote, as well as it is open-sourced for those who want to have access via code repository (*i.e. GitHub*) or marketplace (*i.e. Envato Market*), so that everybody who wants, can look and inspect the code. Also it has a separate database tables for public and secret voting, where secret tables has only hashes that can be re-generated by the voter only to check the existence of his vote by using his e-signature, as well as that tables does not have any voter 3rd party identification data – no IP address, HOST etc.

2.2 How we will implement our vision statement / strategic goals?

(**Product possibilities tree; goal tree, goal model, goal decomposition**)

Product vision is reachable by implementing all features described in product possibilities tree.

Product possibilities tree:

- I. Empower local governments (city municipalities, county & state governments) around the world to improve their voting process (make it cheaper to run, quicker to count votes) and advance into digital democracy by using benefits of digital democracy that will allow the election to become more fair, attract more young people to vote and gain more trust & security to the voting process.
 - a. Ensure proper voter authentication (a fact that a voter is actually is the person who he is claiming he are).
 - i. Check for voters existence if government citizen’s register (*i.e. via “Centre of Registers”*).
 - ii. Validate voter via mobile carrier issued signature (*i.e. via “T-Mobile”, “Verizon Wireless”, “AT&T”*) **or** validate voter via government organization-issued e-signature (USB key, ID card chip) (*i.e. via “Centre of Registers”*).

- iii. Validate voter's certificate via e-signature certification issuance and validation authority (*i.e. Estonian company "AS Sertifitseerimiskeskus"*).
- b. Empower young citizens to vote easily.
 - i. Build user-friendly UI.
 - ii. Follow the UX (User-eXperience) trends of 2019 popular in social networks (Facebook, Linked.in, React-based websites).
- c. Ensure the security of the voting.
 - i. Use at least two different databases
 - ii. Make sure that the "write" happens to at least two locations.
 - iii. Ensure that discs that are used for data-storage uses on of RAID technologies (hardware data cloning).
 - iv. Create a way for anyone who is approved to see the system source.
 - v. Make the system code on interpretable (non-compileable) language.
 - vi. Make sure that all data and code servers stays in the voting country area (not abroad).
- d. Ensure the ability to vote in secret.
 - i. Use trusted hashing algorithms that has not been breached for at least 10 last years.
 - ii. Use only single-way algorithms for critical data hashing.
 - iii. Use hashing algorithms that cannot be decoded with the fastest computers year-by-year in at least 30 years from the election. Use Moore's law for computer speed calculation increase.
 - iv. Separate public and secret votes to a separate database tables.
 - v. Ensure that secret votes database tables does not have any additional voter identification (*i.e. voter's IP address*) besides the voter hash that can be re-generated only by voter.
- e. Ensure the ability for voter to check if his vote is counted.
 - i. Voter should be able, by using his electronic signature, to generate a unique hash, and the system must be able to say if the vote exist.
 - ii. Give a notification / confirmation after casting the vote to at least two devices of voter (*i.e. computer screen and mobile phone*).
- f. Ensure that the system is well understood by a bit part of the voters how it works and how their votes are counted and encrypted.
 - i. Use widely-known and widely-accepted coding standards, web technologies, design patterns.
 - ii. Use trusted code repository to host the code.
Note: as of 2019 stats, mostly trusted code repository in the world is GitHub.
- g. Make system easy to start to use.
 - i. Use most popular content management system in the world.
Note: as of 2019 stats, WordPress runs over 32% of entire internet.
 - ii. Make sure that system can run straight away on standard WAMP/LAMP server with one of popular server management systems / server control panels (*i.e. Plesk, DirectAdmin, CPanel*) without any need for specific server configuration.
- h. Make sure the system is affordable to majority of the world's local governments.
 - i. Focus on the goal that system will have at least 1000 sales, so that it's development price (*work hours x experienced developers & architects work*) could be divided by 1000.
- i. Ensure that everyone who wants to vote and has permit to vote, can vote (system's power, response time).
 - i. Make sure system's response time is no longer than 30 seconds to cast a vote.

- ii. System must be shredded to server-clusters / databases by 500,000 voters per database.
- iii. If all these 500,000 voters would sign it at the same time to vote, their vote has to be counted in no longer than 30 seconds.
 - 1. If during the development process it will be learned that this goal is not reachable or is too expensive to reach for average city government, system has to be re-shredded smaller parts by 100,000 voters (or less) by database.
- iv. The clusters size has to be configurable in via system configuration file.
- j. System must support re-voting.
 - i. Ensure that only last vote is counted.
 - 1. Void all previous votes of that voter.
- k. Ensure that there is no way for voter to vote twice (quality assurance, availability).
 - i. System's availability has to be less important than a completed vote transaction.
 - 1. If vote transaction is not completed, no further write is allowed to any of the servers by that voter.

2.3 **What objects and information models product (internet voting system) uses, analyses, adopts?** *(Business objects / informational models, business object relations, business rules)*

Sub-goals defined in product possibilities tree (see section 2.2) are implemented by using defined business objects (informational models) that are meaningful for internet voting system users. All this has to be done without breaking defined business rules.

Business objects (informational models) meaningful to system users:

- I. Voting organization – a local government, i.e. City municipality, county government, state government.
- II. Product – internet voting system.
- III. Election – a process with specific timeframe when voting happens.
- IV. Voting – a process when voters votes.
- V. Candidate – a person, who is eligible to be elected.
- VI. Voters – citizens, that are eligible to vote.
- VII. Votes – database tables, where the votes are stored.
- VIII. Verification / confirmation devices - mobile devices that has installed applications for vote verification / confirmation number / hash, confirming that the vote was saved to database
- IX. Servers – the place where all hardware and software, as well as database exist.
- X. Trusted data centers – the place where the servers are, the data center must be TIER 1 or higher availability level, located in local government's country.
- XI. Public-Private key encryption devices – mobile SIM cards, or USB keys, or ID cards with electronic signature.
- XII. Voting device – mostly expected this is a personal computer (PC), portable computer (LAPTOP), or tablet (i.e. iPad). Voting is mostly expected to happen via internet browser installed in voting device, but this does not limit to allow to vote via native mobile application (i.e. iOS App, Android App).

Business rules:

- I. Both – public and secret voting must be supported.
 - a. Secret votes must be encrypted with one-way data encryption via trusted encryption algorithm. (i.e. one of SHA2-256, SHA2-384, SHA2-512 algorithms based on loop method (also known as “**key-stretching**“) with ex. 9,563 iterations, or **scrypt**, or **bcrypt**,

- or public-private key algorithm, like **PBKDF2** („*Password-Based Key Derivation Function 2*“), based on RSA cryptography standard).
 - b. There should be no option from secret vote to reveal voter.
 - II. Voter must be able to verify that his vote stored in database by re-generating it’s vote hash.
 - a. Only voter can do that and only for his vote.
 - III. Re-voting must be allowed.
 - a. If voted more than once, only the last vote is counted on that voter, the rest are voided.
 - IV. Nobody can edit (change) existing voter’s vote for that election.
 - a. If vote is detected as illegal vote can only be voided.
 - b. If voter re-votes, his previous vote must be voided by the system.
 - V. Voting servers must be located in local government country only.

2.4 Who will be the users of the services & features of this product (internet voting system)?
(Business partners, buying organization’s / company’s employees, classes of end-users, their rights & permissions)

Features defined as sub-goals in product possibilities tree (see section 2.2) can be used by defined classes of end-users, business partners and buying organization’s (or buying company’s) employees. Each of this system user groups can work with business objects meaningful to system users only in a way, which is allowed to them, and only those actions, which permissions they have granted.

Product users:

- I. Voters (end-users).
 - a. Permissions:
 - i. Can only cast a vote.
 - ii. Can re-vote.
 - iii. Can check if his vote exists in database (by re-generating vote hash).
- II. Voting organization employees:
 - a. System administrators at voting organization. Permissions:
 - i. Can void a vote if it is reported as a fake.
 - ii. Are not allowed to edit / change any existing votes.
 - iii. Are not allowed to add new votes.
- III. Voting organization’s partner organization’s employees (i.e. “Transparency International”):
 - a. Employees, that are assigned to monitor selected elections:
 - i. Can inspect real-time on live serves the code of the system and database data of the systems (that includes votes, which, if secret voting is chosen, has encrypted voter’s hash and vote data there only without any other voter’s identification abilities).
 - ii. Can order to voting organization to void a given vote, if they have detected that this is a false vote.
 - iii. Can order to voting organization to halt the voting process (to stop accepting any other votes, as well as to block the ability to void the existing votes) if the system has been compromised.
- IV. Vote verification application (i.e. mobile phone with e-signature based SIM card).
- V. 3rd party applications (i.e. iOS voting app, Android voting app) – if the system would allow to vote by native mobile applications, those application will need to have access to internet voting system via RESTful API to add, void and read votes.
- VI. External communicators (external systems) with callback support (see [Figure 1](#)):
 - a. A citizen’s register access service (i.e. government organization – “Centre of Registers”).

- b. Mobile signature authentication services (i.e. “T-Mobile”, “Verizon Wireless”, “AT&T”)
- c. E-Signature (USB key, ID card chip) authentication service (i.e. government organization – “Centre of Registers”).
- d. E-Signature certification issuance and certification validation authority (i.e. Estonian company “AS Sertifitseerimiskeskus”).

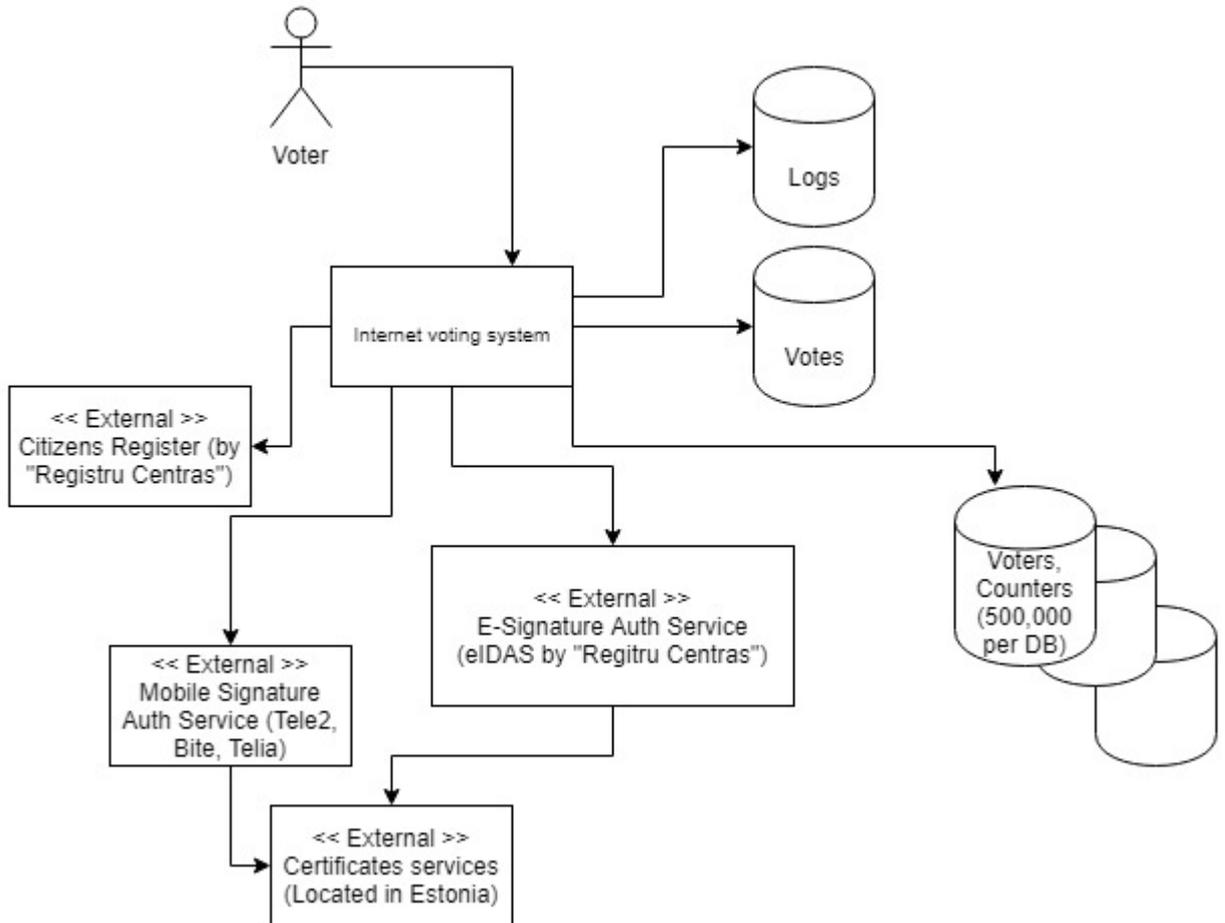


Figure 1. Internet voting system example context view on behalf of external communicators (external systems).

2.5 **Where and how the product (internet voting system) can be accessed to end-users and buying organization after purchase?**

(Access types, access points, desired devices, desired applications, system distribution & system shredding)

The features of the internet voting system described product possibilities tree can be accessed only in desired types of access points via desired devices and desired applications.

Desired access types & access points:

- I. Access of admin panel of the internet voting system:
 - a. Access can be done only from white-listed IP addresses. IP white-listing can be done only by having direct (physical) access to the server – meaning, by going to data center and logging-in to data center’s local intranet.
 - b. Advised (but not required), that, if possible, buying organization should configure the access to the server’s with product’s admin panels source code, to be accessed via intranet only (in same location where the servers are located).
- II. Public (Front-end) access:

- a. Accessed over secured HTTPS network with Extended Validation CA (Certification Authority) certificates issued at root (government) access level (top-level HTTPS certificates).
- b. Voter should be able to cast a vote being abroad by using his government-issued ID card with E-Signature chip installed in it, or via government-issued USB key. Also votes abroad may be accepted that are signed by mobile signature if the voter's current country mobile carriers does support same e-signature protocols.

Note: For mobile e-signatures, at this moment it is possible to sign documents between some European Union (EU) countries.

Desired access devices:

- I. Personal computer (PC),
- II. Portable computer (LAPTOP),
- III. Tablet (i.e. iPad).

Desired access applications:

- I. HTML5 & CSS3 standards compliant internet browser (*i.e. Google Chrome, Mozilla Firefox, Opera and Microsoft Edge*), installed in voting device.
- II. (If supported) Native mobile application (*i.e. iOS App, Android App*), installed in voting device, that will use RESTful API to cast a vote on behalf of the voters to the internet voting system server's.

System distribution (recommended configuration):

- I. For security reasons, system should be distributed via at least 3 different data-centers.
- II. Each data-center should be located separate physical locations
- III. Each data-center should be connected on different network line.
- IV. There should be used at least two different configurations (*i.e. WAMP with MariaDB and LAMP with MySQL*) on servers in different datacenters.
- V. Each server should use RAID disks to always write data at the same time to at least two different data disks (SSD or HDD).
- VI. Each server should be connected to back-up server that would perform lazy-clones on the acting databases every 30 minutes (recommended timeframe as a balance between price, stability and scalability, see [Figure 2](#)).
- VII. System has to support staged processing (concurrency) - the system must be based on concurrent process model to allow multiple processes to happen at the same time. The "stoppie" is database write – do not allow the same voter to write more to database, until and started write's by that voter is not marked as completed (see [Figure 3](#)).

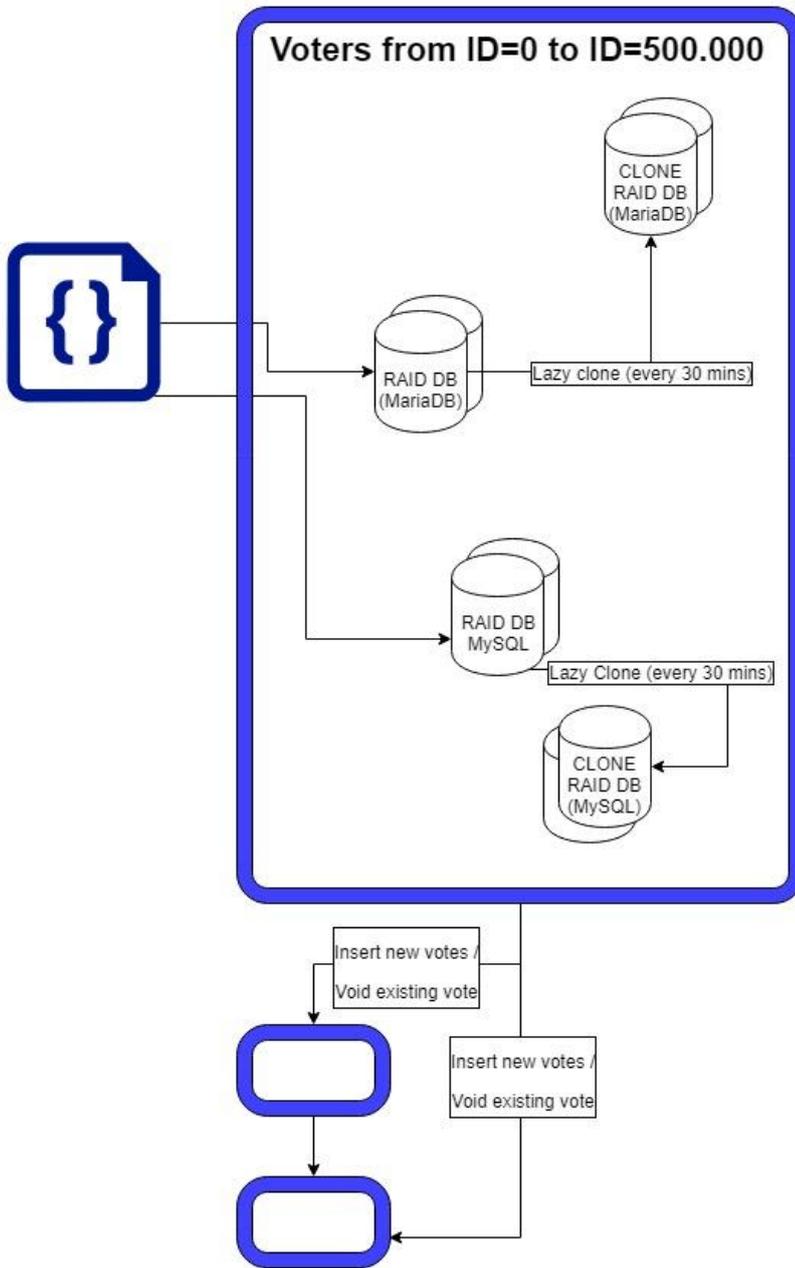


Figure 2. Recommended internet voting system distribution, lazy-cloning & shredding.

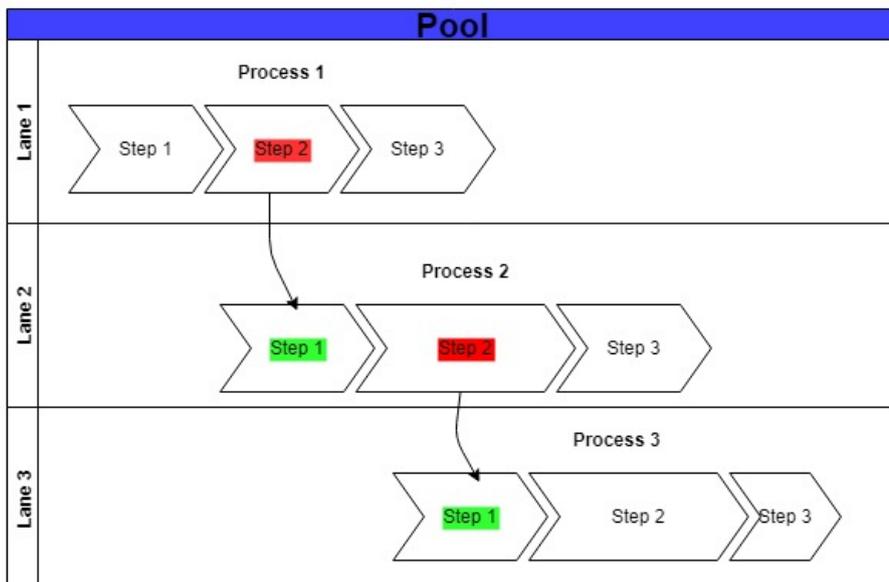


Figure 3. Concurrent system model with the database write as “stoppie” for each voter.

System shredding (recommended configuration):

- I. Logs are in separate database server.
- II. Guests are processed via separate database and code server.
- III. Logged-in voters are shredded in chunks by 500,000 voters per database. The chunk size has to be configurable via system's configuration file (see Figure 2).

2.6 How fast product must perform to fulfill desired operational (short-term) goals?

(User action timing, business transaction's timing, product performance requirements, efficiency requirements, SLA recommendations)

System user (voter, buying organization system administrator, partner organization, external communicator / external service, 3rd party application) has to be able to use each of internet voting system features defined in product possibilities tree in a time, which is not longer than defined in product performance requirements.

Internet voting system performance requirements:

- I. From entering the website, logging-in to getting confirmation that vote is saved to database, should take no more than 5 minutes for whole process.
 - a. Voter has to be able to cast his vote after vote button click in no longer than 30 seconds period on the worst scenario.
 - b. Worst scenario is maximum supported connections to database server (by default this is 500,000 voters).
- II. If the internet voting system is down, voter must get information to his computer screen in no longer than 30+1 second.
- III. If the authentication service (i.e. "Centre of Registers") is down, voter must be informed via his voting device or device used for e-signature authentication in no longer than 60 seconds period.
- IV. For system administrators testing, system must support performance monitoring via performance counters – an on/off screen output of how many milliseconds it took to load each website section, and how many overall queries we ran. This information can be logged to file if total load was longer than 5 seconds, and will include all SQL queries that took longer than 1 second.

SLA (Service-Level Agreement) recommendations between selling organizations and buying organization, buying organization and end users.

- I. Support recommendations during elections period:
 - a. 24/7 support via e-mail with at least 1 technical specialist (server administrator) online. Totally 4 full-time employees will be needed for this.
 - b. Technical specialist phone number will be known to management of main customer (i.e. "by San Francisco State's Government's Head of IT")
 - c. 24/7 support via phone & e-mail of 2 non-technical specialists. Totally 8 full time employees will be needed for this.
- II. Alerting recommendations during elections period:
 - a. "Uptime Robot" (<https://uptimerobot.com/>) software should do pings to servers.
 - b. The e-mails has to be send if server is down, got back up. The e-mail has to be configured by technical administrator to receive by his phone every 5 minutes. Also the "iPad" application that will send a sound on down has to be installed on technical administrators "iPad" and notifications & notification sound must never be disabled for this application if he is on-duty at that time (during his working hours). On non-working hours only the sound can be disabled, notification still has to pop-up in the screen.

3 End-User requirements

3.1 **Which tasks and why end-user will be performing?**

(Use cases, tasks)

Product has to be developed so that end-user could accomplish tasks reasonable to him that allows to use system features defined in product vision statement.

Priority levels of end-user tasks for implementation requirement:

- I. Top – product cannot be launched without this feature.
- II. High – important to have, but product’s beta can be launched without this feature.
- III. Moderate – should be implemented in the system, but probably in later versions.
- IV. Low – optional tasks.

End-User use cases:

- I. Task #1 – cast a vote online:
 - a. Motivation: To participate in elections
 - b. Priority: Top.
 - c. Steps:
 - i. Open HTTP5 & CSS3 compliant internet browser.
 - ii. Open voting website.
 - iii. Login to the system via trusted authority.
 - iv. Select an election.
 - v. Select that we are voting for a first time.
 - vi. Select a candidate for which voter wants to vote.
 - vii. Click on “cast a vote with e-signature”.
 - viii. Sign the vote with e-signature.
 - ix. Get a confirmation code to mobile device and computer screen that the vote has been counted. The code has to match on mobile device and on computer screen.
 1. If code does not match, voter should void his vote and report the government organizations.
 2. If code does match, voter should confirm his vote via secondary device (mobile phone app).
- II. Task #2 – re-vote online:
 - a. Motivation: To change opinion, or to change vote after pressure to vote for somebody by someone.
 - b. Priority: High.
 - c. Steps:
 - i. Same as I.c., except that we have to select that we are “re-voting”.
- III. Task #3 – physically vote.
 - a. Motivation: Some people does not trust the internet.
 - b. Priority: Top.
 - c. Steps:
 - d. Go to election building after online voting is finished.
 - e. Cast a vote.
 - i. If regular vote had be casted, the online vote is voided.

3.2 How deeply the system has to be digitalized?

(End-user operational needs)

End-user operational needs:

- I. End-user may be abroad during election, so he has to be able to vote from abroad. That requires full digitalization.
- II. End-user may still want to go to local government election building physically and vote by hand, so the system has to support the regular voting after online voting is finished.

3.3 What business and informational objects has to exist to perform end-user tasks?

(Business objects, informational objects, key database tables)

Key informational objects:

- I. Voters database table (address, citizenship, eligibility to vote)
- II. Secret votes database table (secret votes hash, vote)
- III. Public votes database table (voter id, vote)
- IV. Candidates database table (address, citizenship, eligibility to be elected)
- V. Election list database table (election id, election start & end dates, voting area of eligible voters)
- VI. External government citizens register (unique person id, address, citizenship)
- VII. Certification authority for voter's e-signature certificates issuance and validation.

3.4 Who and with what skills will be eligible to use the product?

(End-User qualification requirements, UI requirements, usability requirements, security requirements)

End-User qualification requirements for voting online:

- I. End-user has to have basic computer usage skills.
- II. End-user has to know how to sign a document with digital signature.

3.5 Where and which internet voting tasks can be performed by end-user?

(Requirement to voting devices and locations)

Voting locations:

- I. Vote can be casted from anywhere in the world.

Requirement for voting devices:

- I. HTTP5 & CSS3 compliant web browser
- II. Secure-enough mobile device that can run compatible apps – iOS or Android, and has up-to-date mobile OS.

3.6 How fast end-user tasks has to be performed?

(Timing)

End-user tasks timing:

- I. Vote has to be signed with e-signature by mobile device in less than 60 seconds.
- II. Vote has to be saved in to database in no longer than 30 seconds after the vote has been signed.
- III. Voter has to be able to perform full voting process (from entering the website till confirmation code about saved vote) in no longer than 5 minutes.

4 Software System Requirements Level

Requirements for market type systems of system built for the market for all three views – system analyst view, software system engineer view and developer view – can, and should be understood and extracted from architectural description of software system. The requirement specification has to help to build the system, not to make it much more complex via overload of text which the development needs may change really quickly in changing markets, so it has to be minimal, and focused only to what is mandatory. For that there is architectural description with its views and architectural decision records (ADR).

4.1 Software system requirements

Not included in this document. See architectural description document.

4.2 Project-bases software system requirements

Not included in this document. See architectural description document.

4.3 Implementation requirements for software system

Not included in this document. See architectural description document.

5 System design sketches

5.1 Step of choice



Figure 4. Selection window for experienced voters and undecided voters who need simple guidance.

5.2 Immediate voting process

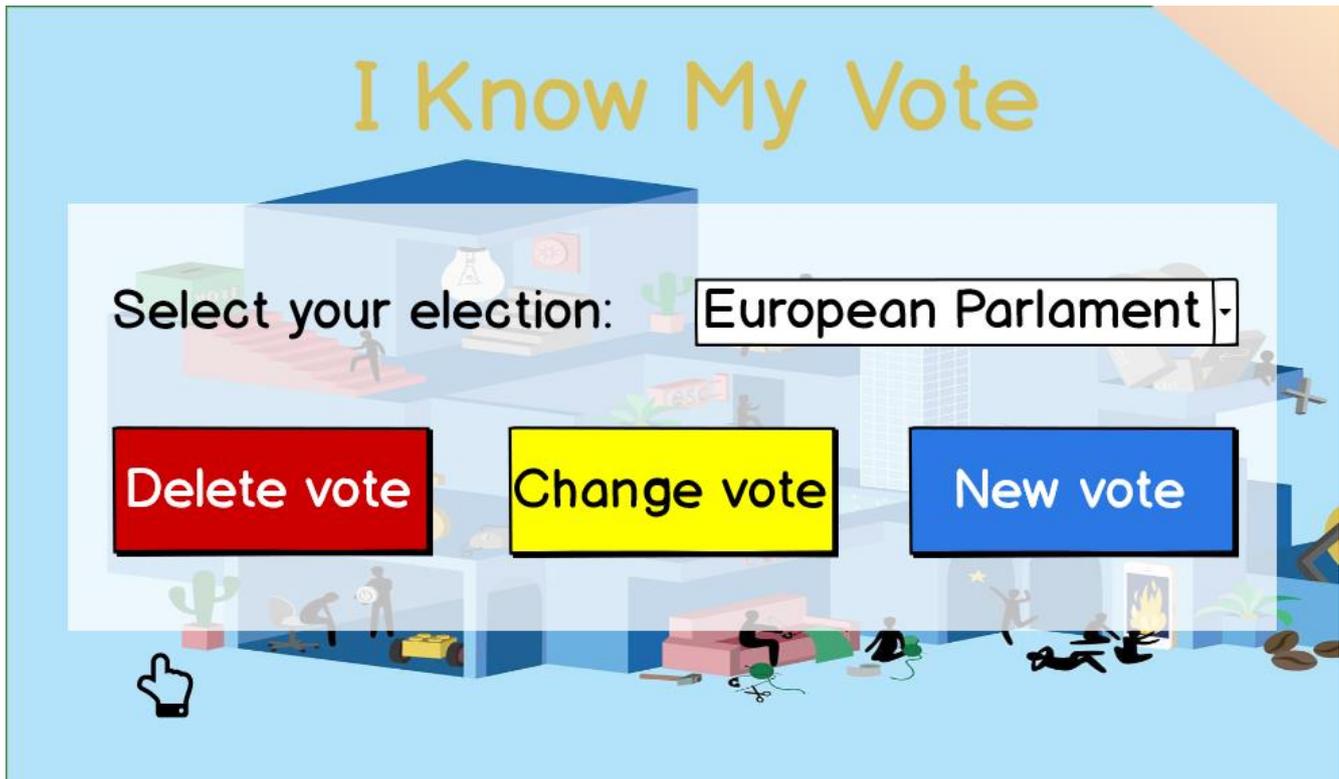


Figure 5. Multiple choices of what you plan to do with your vote.

	Nr	Name	Lastname	Political party
<input type="radio"/>	1	Rosie	Ward	Liberal Party
<input checked="" type="radio"/>	2	Aisha	Morris	Democratic Party
<input type="radio"/>	3	Noah	Miller	Conservative party
<input type="radio"/>	4	Harry	Smith	Democratic Party
<input type="radio"/>	5	Sophie	Taylor	Radical party
				VOTE

Figure 6. List of possible candidates.

Choose your identification Type

The figure shows three separate login panels. The first panel is for 'Cryptographic USB', with instructions to 'Connect USB to your computer' and a 'Login' button. The second panel is for 'Mobile signature' using a personal ID, with a text input field containing a blank space and a 'Login' button. The third panel is for 'Mobile signature' using a chip card, with instructions to 'Connect chip card to your computer' and a 'Login' button.

Figure 7. Ways of voter's identification.

The figure illustrates a two-factor authentication process. At the top, a box displays 'Mobile signature' with a help icon and a verification instruction: 'Verify that this session nr is shown on your phone: 3059'. Below this are three login panels. The middle panel, for 'Mobile signature' using a personal ID, has a text input field containing the number '1234567894' and a 'Login' button being clicked by a mouse cursor. To the right, a smartphone screen shows the text: '09:52 AM', signal strength, Wi-Fi, and battery icons at the top; and 'Register center Authentication to your mobile signature Session code: 3059 Connect?' in the center.

Figure 8. Two-Factor vote confirmation steps.

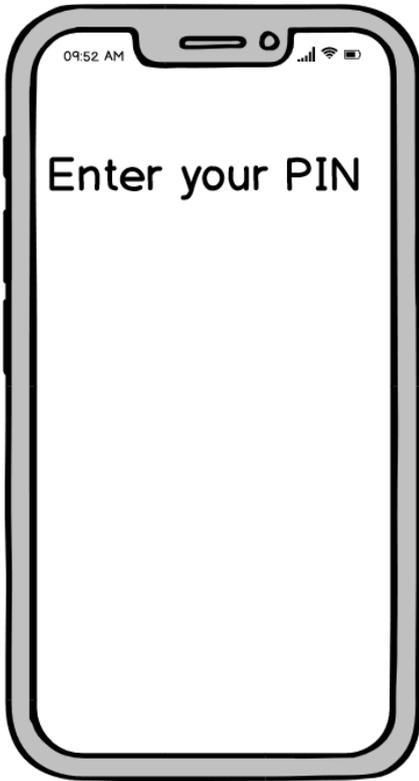


Figure 9. Process of entering voter's PIN code and vote confirmation page.

5.3 Process of learning more

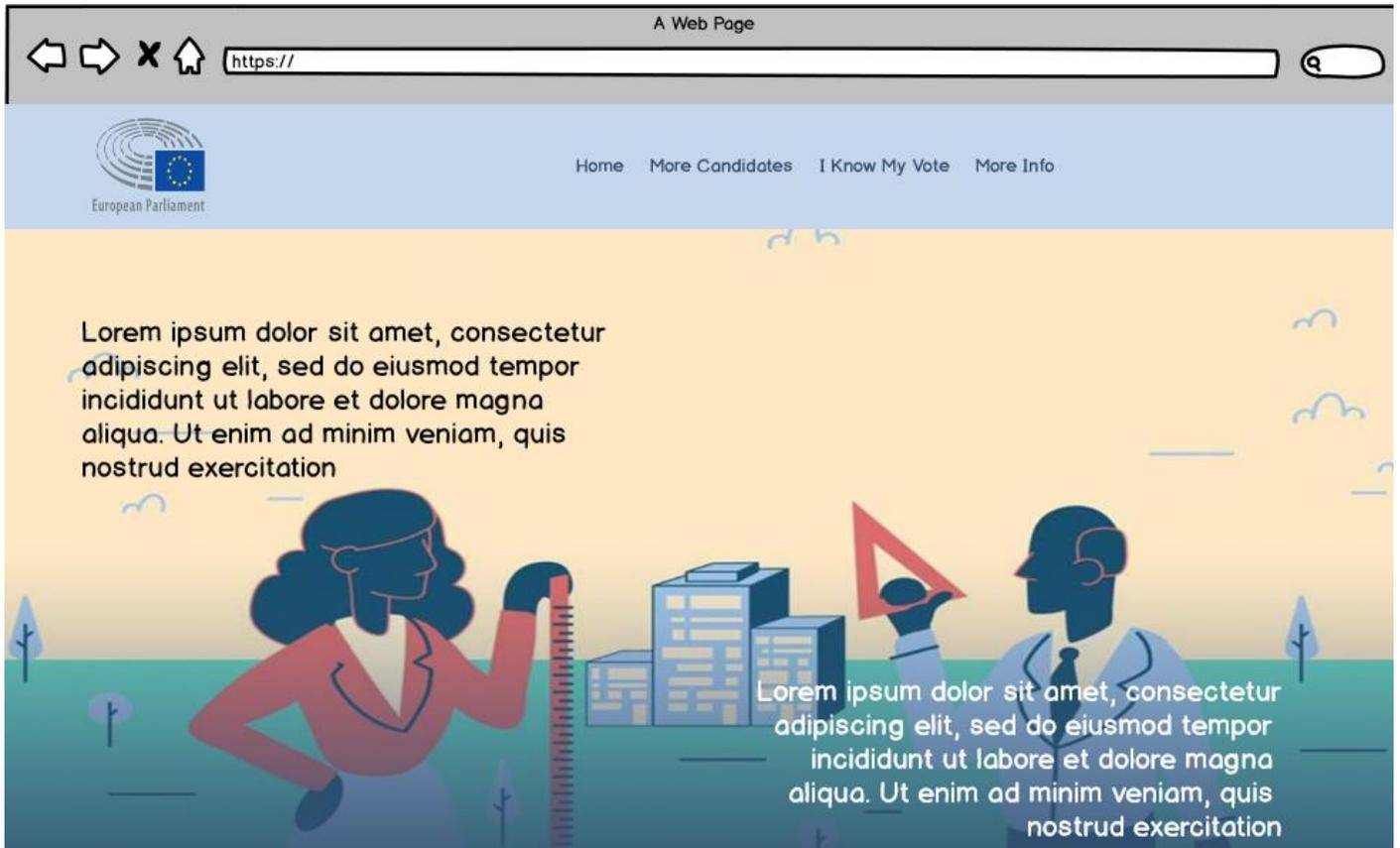


Figure 10. Introduction about the upcoming elections.

Why Vote?



Figure 11. Section about the importance of voting.

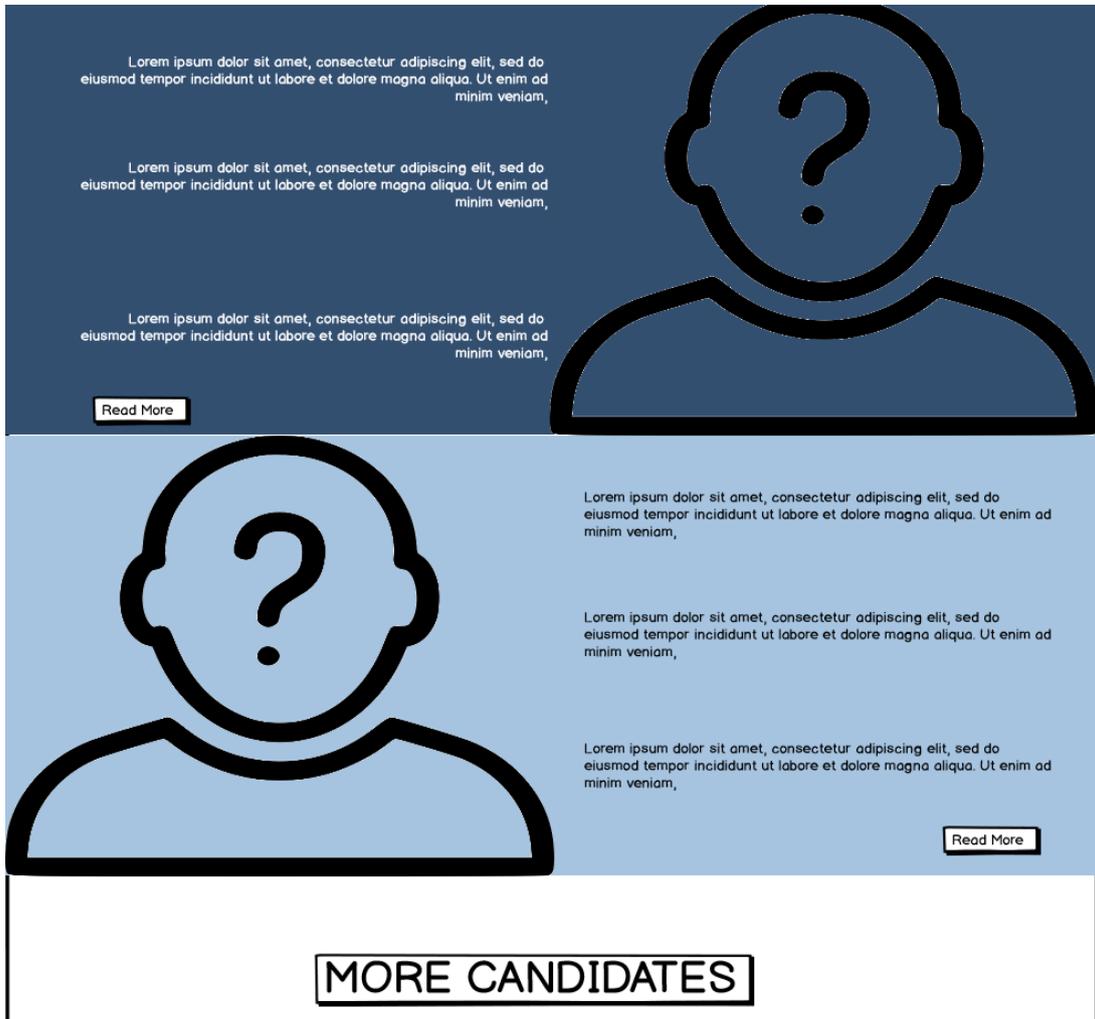


Figure 12. Randomly picked information about candidates and ability to see more.

6 Web design proposal for potential buying organizations



Figure 13. Proposed web page part 1.

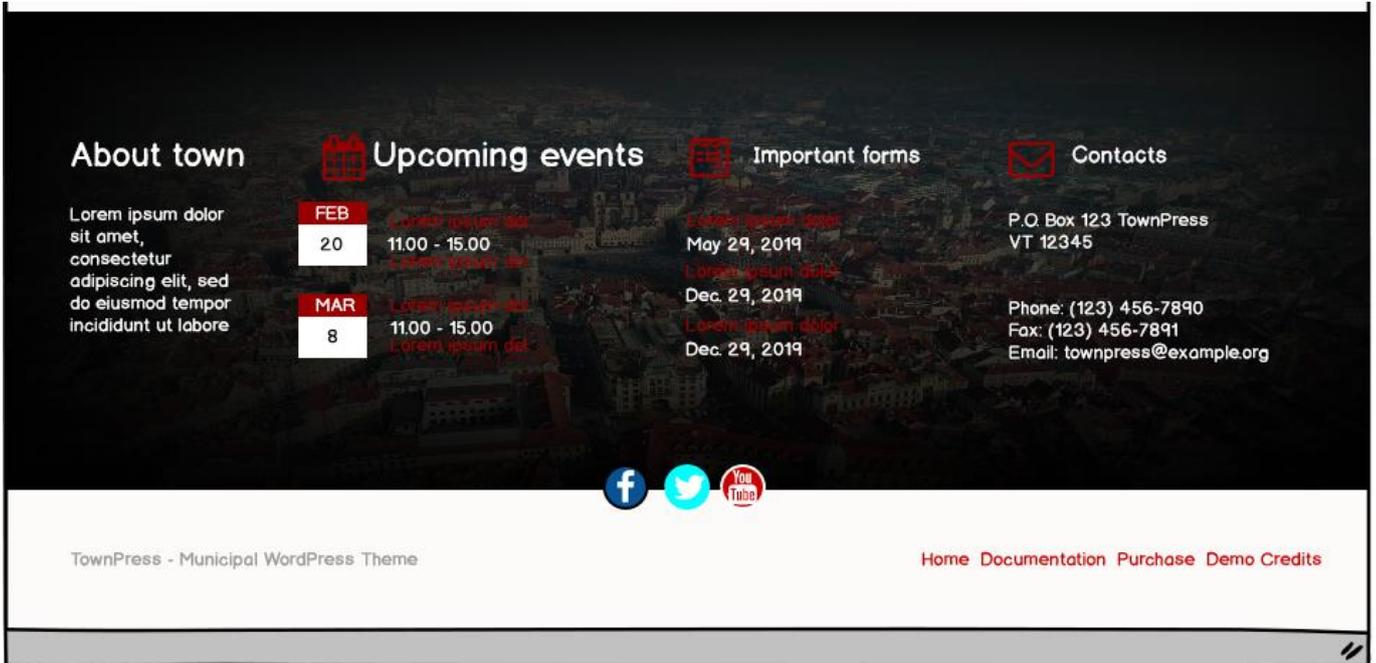


Figure 14. Proposed web page part 2.