***Sistem software de vot***

Se consideră un sistem software de vot care implementează interacțiunea dintre cetatean - responsabil judetean – responsabil national în ceea ce privesc voturile din timpul alegerilor.

* *Prezentarea* cerintelor funcționale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura.

1. Se vor implementa 2 aplicatii, o mobile in limbajul android si o aplicatie web in php si html, dar si un server care va primi toate si prelucra datele, tehnologia folosita fiind .NET (constrangere).
2. Baze de date Sql (constrangere);
3. Aplicatia mobile trebuie sa fie functionala pentru orice revizie a sistemului de operare Android (constrangere);
4. Sistemul trebuie să poată fi scalat în ceea ce privește utilizarea aplicatiilor de la 100 de utilizatori aflați în locații geografice diferite la 100.000 de utilizatori fără a crește dificultatea de prelucrare a datelor.
5. Timpul de raspuns trebuie sa fie mai mic de 10 secunde.
6. Termenul de finalizarea a proiectului va fi 1.10.2018.
7. Numarul de utilizatori conectati simultan va fi 500.000. In caz ca acesta creste, sistemul va bloca cererile pana cand acesta devine “liber” pentru utilizatorii respectivi.
8. Sistemul trebuie sa fie usor de modificat in cazul unor noi cerinte fara a influenta arhitectura.
9. In ceea ce priveste portabilitatea, sistemul va fi usor portat dar si testat.
10. Autentificarea: sistemul poate verifica identitatea utilizatorilor, daca acestia sunt in masura pentru a vota.
11. Autorizarea: utilizatorii vor avea acces la optiuniile politice, implicit la detalierea acestora, dar si la numarul de voturi pe care fiecare partid l-a inregistrat pana la acel moment.
12. Criptarea: mesajele trimise de aplicații catre servere sunt criptate;
13. Integritatea: conținutul mesajelor nu este modificat în timpul transmisiei;
14. Fiecare cetatean poate vota doar dupa ce se autentifica pe baza datelor personale, cel mai important fiind CNP-ul.
15. Dupa autentificare si validare, se expima optiunea dorita iar dupa aceasta nu isi mai poate modifica votul. Pentru a nu se anula votul, se poate selecta un singur partid politic.
16. Datele vor fi transmise catre un servere care le vor prelucra si vor realiza diferite statistici (prezenta la vot pe ore, numarul de femei / barbati care au votat, pretenta rural/ urban, numarul de voturi inregistrat de fiecare partid). Aceste statistici vor putea fi urmarite in timp real de fiecare utilizator.

Tehnologiile folosite pot influenta arhitectura, tocmai de aceea aceasta este influentata de cerintele 1, 4, 7, 8, 9 si nu numai.

* *Descompunere in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele.*

Vor exista 10 componente:

1. Cetatean: Va furniza date personale catre aplicatii pentru a se putea autentifica.
2. Aplicatia web/ 3. Aplicatia mobile: Vor prelua date de la utilizatori ce vor fi transmise mai apoi componentei de verificare. Dupa aflarea unui rezultate se vor intoarce catre utilizatori mesaje specifice.

4. Verificarea autenticitatii: Se va face cautarea in baza de date a cnp-ului introdus de catre utilizator, care mai apoi va furniza aplicatiilor rezultatul.

5. Componenta de criptare: Aceasta va cripta datele ce provin de la utilizatori si vor fi transmise criptat catre servere.

6. Server judetean: Acesta va primi datele criptate de la utilizatorii judetului respectiv, iar dupa decriptare si va realiza statisticile solicitate prin componenta de calcul, care mai apoi vor fi transmise catre server-ul national.

7. Componenta de decriptare: Serverele judetene vor da ca date de intrare mesajele criptate si le vor primi mai apoi decriptate.

8. Server national: Va primi statistici de la serverele judetene si va genera statisticile nationale. Toate aceste date vor fi transmise din nou catre cetateni, care pot monitoriza voturile.

9. Calcul statistici: Realizarea efectiva de statistici avand ca date de intrare datele primite de la server-ul judetean/national, iar rezultatele fiind transmise tot catre acestea.

10. Baze de date aferente judetelor (3 judete in acest caz).

* *Identitifcarea celor mai importanți 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumetarea alegerii.*

1. Consider ca cel mai important indicator de calitate este cel de autorizare, deoarece doar persoanele valide ar trebui sa aibe acces la prcesul de votare. Printr-o metoda de cautare in baza de date a fiecarui judet se va decide daca un cetatean/utilizator este eligibil pentru a vota.
2. Un alt indicator este cel de criptare. Printr-un sistem de criptare datele nu vor putea fi alterate sau “supravegheate” de catre inamici. Acesta l-am ales deoarece se pot crea presiuni asupra cetatenilor.
3. Timpul de raspuns este foarte important pentru ca altfel utilizatorii si-ar putea pierde rabdarea, iar vizionarea in timp real al statisticilor ar fi greoaie.

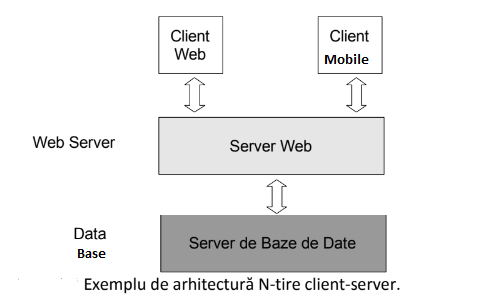
* *Identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor.*

Tehnologiile middleware folosite sunt:

* Tehnologii middleware bazate pe mesaj, ce fac parte din nivelul de transport. Am ales aceasta pentru ca schimbul de date intre componentele software să fie foarte ușor de implementat.
* Tehnologia .NET este folosita pentru servere, deoarece se pot dezvolta aplicatii pe toate platformele de Windows si nu doar atat, dar si pentru securitate.
* SonicMQ este folosit pentru a procesa rapid mesajele si deoarece dispune de instrumente ce permit manipularea mesajelor între diferitele componente ale unui sistem software.
* *Identificarea pincipalelor modele și stiluri arhitecturale folosite, argumentarea alegerilor.*

Am ales modelul N-Tire Client-Server deoarece este in concordanta cu sistemul software ce se doreste proiectat. De asemenea toate proprietatiile modelului sunt necesare acestui sistem, si anume: separarea responsabilităților, comunicarea sincronă si flexibilitate la instalare.

Un alt motiv pentru care am ales acest model de arhitectura este acela ca acest tip se foloseste atunci când exista un număr mare de utilizatori și un număr mare de conexiuni concurente.



* *Prezentarea scenariilor de validare a arhitecturii*

In acest pas se face testarea pas cu pas a arhitecturii pe baza cerintelor si a eventualelor cerinte.

Deoarece nu exista un sistem software “fizic” verificarea se va face prin testarea manuală utilizând scenarii, respectiv validare prin construirea unui prototip.

1. scenarii: Pentru a validarea arhitectura se vor definii stimuli care să aibă efect asupra arhitecturii. Apoi se face o analiză pentru a se determina care va fi răspunsul arhitecturii la un astfel de scenariu. Dacă răspunsul este cel dorit atunci se consideră că scenariul este satisfăcut de arhitectură. Dacă răspunsul nu este cel dorit sau este greu de calificat atunci s-a descoperit o zonă de risc în arhitectură.

2. prototip :

- proof of concept;

- proof of technology;

Odată ce prototipul a fost implementat și testat, răspunsul arhitecturii la stimulii prevăzuți în scenarii se poate obține cu un înalt grad de certitudine.

Un scenariu de testare ar fi acela in care se face validarea pe baza de CNP. Acesta se poate realiza prin implementarea unei metode de cautare si verificarea acesteia.

Un alt scenariu este cel in care criptarea nu se realizeaza pentru toti utilizatorii, iar daca se intampla acest lucru cum se poate intervene in arhitectura.

* Rezentarea sistemului software din doua perspective (o diagram pentru fiecare perspectiva + explicatii)

1. Perspectiva logică: va descrie elementele principale ale arhitecturii și relațiile dintre ele.

2. Perspectiva de dezvoltare: surprinde organizarea internă a componentelor software.