

Cuprins

Motivație	2
Introducere	3
1 Descrierea aplicatiei AudIT	5
1.1 Problema adresata	5
1.2 Solutia propusa	6
1.3 Functionalitatile aplicatiei	6
1.3.1 Autentificarea pe platforma	6
1.3.2 Verificarea actiunilor utilizatorilor	7
1.3.3 Pagina de start	8
1.3.4 Gestionarea misinilor de audit public	8
1.3.5 Pagina rezumat misiune de audit	8
1.3.6 Prelucarea pasilor unei misiuni de audit	9
1.3.7 Stabilirea obiectivelor de auditat	11
1.3.8 Vizualizarea si accesarea ricurilor din misiuni anterioare	11
1.3.9 Identificarea si evaluarea riscurilor	11
1.3.10 Vizualizarea obiectivelor	12
1.3.11 Pagina detalii actiune	12
1.3.12 Salvarea documentelor necesare pe platforma	13
1.3.13 Accesarea documentelor necesare pe platforma	14
1.3.14 Completarea documentelor tip sablon	14
1.3.15 Bara de navigare	15
1.3.16 Gestionarea accesului la resurse partajate	16
1.3.17 Gestionarea activitatilor desfasurate	16
1.3.18 Profilul personal	16
1.3.19 Sistemul de notificari	16

2	Arhitectura platformei	17
2.1	Arhitectura generala	17
2.2	Arhitectura serverului	18
2.3	Arhitectura interfatei grafice	23
2.4	Stocarea datelor	24
2.5	Aspecte de securitate	25
	Concluzii	26

Motivație

În contextul actual al digitalizării foarte rapide a instituțiilor statului, vine de la sine nevoia dezvoltării unei soluții accesibile tuturor auditorilor publici din România, o soluție eficientă din punct de vedere al utilizării acesteia în sectorul public, transparentă și în conformitate cu standardele legislative prezente la momentul actual în România.

Soluția propusă presupune o platformă web, accesibilă public, în care atât auditorul public, cât și persoanele reprezentante ale instituțiilor auditate vor avea acces la funcționalitățile oferite de aplicație, adică, ei vor putea să își ușureze semnificativ munca depusă, crescând astfel randamentul activităților desfășurate.

De asemenea, inspirația și un ajutor deosebit în dezvoltarea acestei soluții a venit din partea tatălui meu, auditor public în cadrul Consiliului Județean Vrancea, care mi-a oferit explicații amănunțite și detaliate în ceea ce privește totalitatea procedurilor și reglementărilor legislative din cadrul unei misiuni de audit public.

Introducere

Privind in retrospectiva avansul tehnologiei si impactul pe care aceasta il are asupra noastra in viata de zi cu zi, adaptarea la aceasta noua realitate nu mai este o optiune, ci o necesitate.

Avand acest lucru in vedere, este nevoie acum, mai mult ca niciodata, de introducerea si implementarea a noi solutii tehnologice in cat mai multe domenii si sectoare posibile. Domeniul administratiei publice din Romania este locul perfect pentru imbratisarea acestor schimbari, sperand astfel la imbunatatirea eficientei atributiilor angajatilor.

In acest cadru, solutiile ce vor fi detaliate in documentul ce urmeaza, constituie un instrument esential in arsenalul tehnologic necesar, cum este precizat si mai sus, pentru a raspunde provocarilor contemporane.

De asemenea, integrarea acestei solutii cu alte sisteme informatice deja prezente si folosite la momentul actual, va permite o comunicare eficienta intre diferite agentii si departamente, eliminand astfel din limitarile si erorile umane ce sunt prezente.

Descrierea sumară a soluției

Proiectul propus are in vedere dezvoltarea unei platforme web, ce ofera diferite functionalitati auditorilor publici cat si angajatilor departamentelor audidate. Acestia se vor folosi de platforma pentru a accesa in timp real rezultate atat ale misiunilor de audit care se desfasoara in prezent, cat si ale celor ce au fost efectuate in trecut, resurse care constau in documente si rapoarte oficiale, incarcarea si accesarea dovezilor care demonstreaza implementarea recomandarilor oferite de auditor, cat si un sistem de notificari in timp real, ce permite utilizatorilor sa fie la curent cu cele mai noi informatii prezente pe platforma. De asemenea, utilizatorii se vor folosi si de un sistem de exportare a datelor prezente pe site, atat in formate de fisiere .XLS, .CSV sau .DOCS, cat si autocompletarea unor documente oficiale aflate in procedura de audit public de tip sablon cu datele si informatiile pe care auditorul le-a introdus in aplicatie.

Metodologia folosita

Metodologia de lucru la acest proiect presupune documentarea si analiza procedurilor legislative si a pasilor in desfasurarea unei misiuni de audit public, identificarea problemelor sau a riscurilor ce pot aparea si gasirea solutiilor in materie de functionalitati prezente in aplicatie pentru a le rezolva pe acestea. Dezvoltarea aplicatiei consta in utilizarea tehnologiei .NET Core: C# ASP.NET Core Web API pentru serviciile server-side, ASP.NET Core Blazor pentru serviciile client-side, MySQL pentru stocarea persistenta a datelor respectiv platforma cloud AWS pentru diferite solutii de stocare, S3 Bucket, cat si servicii de trimitere Email.

Capitolul 1

Descrierea aplicatiei AudIT

Adaptarea la noile paradigme tehnologice ne impune tuturor o provocare, mai mult sau mai putin difica, institutiile publice statului confruntandu-se zilnic cu aceasta problema, este nevoie cat mai repede de o solutie eficienta care va rezolva aceasta problema. Prima sectiune a acestei lucrari urmareste sa exploreze in detaliu cum platforma AudIT se aliniaza si contribuie la actiunea de transformare si adaptare digitala in sectorul public, analizand componentele cheie ale aplicatiei in raport cu problemele pe care acestea incearca sa le rezolve.

1.1 Problema adresata

Digitalizarea, potrivit definitiei este procesul de transformare a informatiilor dintr-un format analogic, hartii, intr-un format digital, biti. De fapt, acest procedeu constituie o adevarata noua paradigma in materie de algoritmi administrativi, sensul de derulare al intregului sistem si metodele utilizate de catre factorul uman in dezvoltarea solutiilor.

In decursul discutiilor cu tatal meu, auditor public, am descoperit impreunna numeroase puncte nevralgice in metodele si solutiile folosite de auditorii publici din Romania pentru a duce la capat anumite intrebuintari de serviciu. Acestea pot parea nesemnificative pe moment, dar observand fenomenul la scara larga, de exemplu pe intreg parcurul unei misiuni de audit public, care poate dura pana la 6 luni, constatam faptul ca intreg procesul si eficienta auditorului sunt major incetinite de aceste imperfectiuni.

Una dintre cele mai mari probleme prezente in procesul de audit public, si cel mai probabil in majoritatea institutiilor publice din tara, este nevoia de a folosi si a administra inventarul a multor documente oficiale, pierzand astfel mult timp in identificarea documentului corespunzator actiunii sau activitatii pe care auditorul vrea o sa efectueaze, ulterior

pierzand si mai mult timp in completarea si in comunicarea si transmiterea acestui act catre reprezentantul agentiei sau departamentul auditat. De acest lucru este strans legata si problema comunicarii intre partile care participa la misiunea de audit public, aceasta realizandu-se in majoritatea cazurilor prin intermediul postei iar uneori daca distanta permite chiar prin intermediul unor 'curieri umani'.

Avand in vedere aceste vulnerabilitati din sistemul public de audit, platforma web AudIT a fost conceputa pentru a raspunde nevoii de adaptare si transformare digitala, incercand in acelasi timp sa imbunatateasca protocoalele si procesele interne, astfel permitand factorului uman sa isi indeplineasca sarcinile intr-un mod mult mai usor si rapid.

1.2 Solutia propusa

Platforma Web AudIT este conceputa ca o solutie inovatoare asupra provocarilor datorate digitalizarii in institutiile publice, oferind un set de instrumente si functionalitati care fac mult mai accesibila si fluenta munca auditorului public cat si cea a reprezentantilor institutiilor auditate.

Aplicatie are ca si scop principal cresterea eficientei in procesul de audit public, prin implementarea diferitelor functionalitati care vor imbunatati drastic accesul utilizatorilor la informatii si documente relevante, vor creste nivelul eficientei, auditorii concentrandu-se pe aspecte esentiale ale auditului, fara a-si consuma astfel timpul si energia pe numeroase sarcini care se pot dovedi repetitive, amanuntite si obositoare in final respectiv va facilita un mod de comunicare eficace intre persoanele care iau parte la misiunea de audit.

1.3 Functionalitatile aplicatiei

Subsectiunile care vor urma o sa explice in detaliu functionalitatile de baza ale platformei, modul in care acestea au fost implementate, cat si dificultati si provocari ulterioare in ceea ce priveste facilitatile oferite de acestea.

1.3.1 Autentificarea pe platforma

Prima interactiune a fiecarui utilizator cu platforma web o constituie pagina de autentificare, care asigura faptul ca accesul la functionalitatile aplicatiei este restrictionat doar celor care detin sau doresc sa isi creeze un cont pe aceasta aplicatie.

Procesul de creare a unui cont nou este conceput astfel incat sa se ajusteze pe necesitatile

de securitate de baza ale institutiilor publice.

Presupunand faptul ca fiecare angajat al unui departament dintr-o institutie a statului detine o adresa de email cu domeniul institutiei de care apartine, tot ce trebuie sa faca noul potential utilizator este sa se foloseasca de aceasta adresa de email ca sa isi creeze un cont nou. Contul nou este creat cu drepturi limitate, acesta neavand acces la nici o resursa care apartine de institutia sa pana in momentul cand un reprezentant al institutiei nu ii valideaza contul.

—INSERT PICTURE HERE—

Aceasta metoda de autentificare se bazeaza pe o configurare initiala a unor utilizatori cu drepturi elevate, reprezentantii departamentelor, carora li se ofera capacitatea de a verifica noii utilizatori care se inregistreaza pe platforma utilizand domeniul departamentului in cauza. Fiind pe o parte un mod in plus prin care se limiteaza accesul utilizatorilor la anumite resurse pana cand identitatea acestora este confirmata, este pe de alta parte un pas necesar care nu prezinta momentan un sistem de automatizare a verificarii identitatii utilizatorilor, eliminand astfel nevoia unei configurari initiale a platformei.

1.3.2 Verificarea actiunilor utilizatorilor

In cadrul aplicatiei, accesul la fiecare entitate este protejata prin implementarea unor liste de acces care definesc permisiunile de scriere si de citire asupra respectivei entitati. Acest lucru se asigura ca initial, fiecare utilizator are drept de scriere si de citire doar asupra resurselor create de acesta pe platforma, ulterior acesta avand posibilitatea de a acorda sau a primi acces de scriere sau citire asupra altor resurse aflate pe platforma.

—INSERT PICTURE HERE—

De asemenea, este implementat si un sistem de roluri care restrictioneaza si acestea la randul lor accesul la diferite functionalitati ale aplicatiei, astfel spre exemplu, un utilizator cu rol de reprezentant al unei institutii nu va putea accesa paginile referitoare la crearea sau editarea unei misiuni de audit. In plus, pentru o conformitate si pentru o evidenta sporita asupra actiunilor utilizatorilor asupra resurselor de pe platforma este implementat un sistem de auditare al entitatilor, toate operatiile de creare, modificare si stergere fiind salvate.

1.3.3 Pagina de start

Pagina de start este locul implicit unde un utilizator este redirectionat atunci cand autentificarea sa pe aplicatie este cu succes. Aceasta ii prezinta auditorului ultimele modificari la resursele la care are acces, o lista de notificari pe care acesta le-a primit din partea reprezentantilor institutiilor la care auditorul are misiuni de audit in desfasurare cat si diferite butoane de navigare catre pagini cheie din aplicatie, astfel oferind o interactiune mai usoara cu platforma Audit.

1.3.4 Gestionarea misiunilor de audit public

In cadrul procesului de audit public, o gestionare eficienta a misiunilor, atat curente cat si din trecut, este esentiala pentru o experienta cat mai naturala si intuitiva a utilizatorului pe platforma.

Crearea unei noi misiuni de audit este similara cu crearea unui nou proiect, auditorul specificand numele noii misiuni de audit, institutia respectiv departamentul asupra caruia se realizeaza noua misiune de audit.

Dupa crearea noii misiuni, auditorul este redirectionat catre o pagina in care acesta poate vizualiza intr-un tabel toate misiunile de audit la care acesta are acces, cele create de el, dar si cele la care i-a fost oferit accesul. Afisarea intrarilor din tabel este una de tip paginata cu un numar de 7 misiuni pe pagina, astfel incat atentia utilizatorului sa fie concentrata doar pe aceste misiuni, in acest fel eliminand posibilitatea de a nu gasi informatia pe care acesta o cauta datorita unui numar prea mare de linii si informatii.

—INSERT PICTURE HERE —

De asemenea, informatiile afisate in acest tabel pot fi sortate alfabetic dupa numele misiunii de audit, dupa starea in care fiecare dintre acestea se afla, dupa departamentul asupra caruia se desfasoara misiunea sau dupa data ultimei modificari a acesteia.

1.3.5 Pagina rezumat misiune de audit

Pagina de rezumat a unei misiuni de audit are ca scop informarea auditorului asupra unei viziuni de ansamblu asupra misiunii de audit respective. Aceasta contine urmatoarele informatii:

- in partea de sus a paginii, auditorului ii este prezentat sub forma unui lant de pasi,

statusul curent al misiunii, acesta fiind primul lucru pe care privirea utilizatorului il vede;

- o scurta descriere asupra parametrilor misiunii de audit, cum ar fi nume, data ultimii modificari, numele departamentului dar si statusul actual al misiunii. Utilizatorul are optiune de a edita acesti parametri si a salva modificarile aduse;

—INSERT PICTURE HERE—

- in partea dreapta a paginii sunt prezente patru chenare care prezinta cele mai recente modificari si actualizari in materie de : obiective, documente atasate misiunii, fise de identificare a problemei cat si activitati recente asociate misiunii de audit .Utilizatorul are posibilitatea de a naviga apasand pe numele intrarii din lista catre pagina dedicata acesteia, sau la apasarea butonului de 'See more' sa fie redirectionat catre pagina dedicata tuturor entitatilor de acel fel;

—INSERT PICTURE HERE—

- in partea dreapta jos, auditorul are posibilitatea de a selecta misiunea de audit ca fiind misiunea curenta, astfel orice actiune pe care acesta o sa o faca pe platforma o sa ia ca optiune preselectata misiunea selectata de acesta;

—INSERT PICTURE HERE—

d

1.3.6 Prelucarea pasilor unei misiuni de audit

Pentru o reproducere cat mai precisa si cooerenta a stagiilor prin care o misiune de audit trece, platforma permite setarea unui status al fiecarei misiuni de audit, astfel auditorul avand posibilitatea de a-si marca in detaliu progresul pana la momentul curent asupra misiunii de audit. De asemenea, fiecare pas major dintr-o misiune de audit prezinta functionalitati specifice, care vor fi explicate sumar in aceasta subsectiune.

Pregatirea misiunii de audit

Pregatirea misiunii de audit este etapa initiala in care auditorul creeaza misiunea, consulta misiunile anterioare efectuate la acelasi departament, se elaboreaza un plan de

audit, se stabilesc obiectivele, actiunile specifice fiecarui obiectiv respectiv riscurile specifice fiecarei actiuni si se intocmesc o serie de documente oficiale, pentru a tine evidenta activitatilor ulterioare pe care auditorul le va realiza in aceasta misiune de audit.

Interventia la fata locului

Interventia la fata locului este o etapa importanta a procesului de audit public, etapa care implica de cele mai multe ori o deplasare in teren, auditorul efectueaza interviuri, realizeaza esantioane, analizeaza riscurile si obiectivele stabilite la pasul anterior si incearca sa inteleaga intr-un mod cat mai corect si obiectiv activitatile desfasurate de departamentul respectiv. Acest pas consta in esenta in crearea si completarea a multor documente de tip sablon pe care auditorul le va folosi ulterior in pasii ce urmeaza pentru a intocmi un raport final.

Rezultatele Misiunii

Dupa finalizarea pasului anterior, auditorul acum dispune de intreg instrumentalul pentru a intocmi un raport final. Acesta este intocmit pe baza diferitelor intalniri intre auditor si reprezentantul institutiei in care se discuta aspecte legate de constatarile facute in respectiva misiune de audit. Raportul final cuprinde constatarile facute, recomandari sub forma unor Fise de Identificare si Analiza a Problemei respectiv cauze si consecinte ale problemelor.

Acest raport este prezentat partilor participante la misiune pentru a le informa asupra rezultatelor misiunii de audit si pentru a ajunge la o intelegere asupra termenilor de remediere a problemelor pe care acestia trebuie sa le rezolve.

Urmărirea recomandarilor

Urmărirea recomandarilor este pasul final dintr-o misiune de audit in care sunt monitorizate recomandarile oferite de catre auditor si respectarea termenilor limita de implementare a acestora. Reprezentatii institutiilor trebuie sa ia la cunostinta aceste recomandari si sa gaseasca, ajutati de Fisa de Identificare si Analiza a Problemei corespunzatoare fiecarei recomandari, solutii pentru fiecare chestiune in parte respectand totodata si termenul limita impus de aceasta.

1.3.7 Stabilirea obiectivelor de auditat

Stabilirea obiectivelor de auditat are loc in faza initiala a pasului pregatirii misiunii de audit, pas in care auditorul stabileste obiectivele principale care vor fi auditate in cele ce urmeaza. Fiecare obiectiv este compus din mai multe actiuni specifice iar acestea contin la randul lor o serie de riscuri identificate. Aceste riscuri identificate de catre auditor sunt ierarhizate pe baza unei formule de calcul care ia in considerare probabilitatea actiunii de a se intampla, impactul pe care aceasta il va avea si riscul final rezultat al inmultirii celor doua.

Platforma web AudIT ofera aceste functionalitati utilizatorului, astfel incat acesta sa respecte in detaliu toti pasii legislativi ai procedurii de audit public. Auditorul poate forma obiective noi si sa le ataseze la misiunea de audit corespunzatoare.

—INSERT PICTURE HERE—

De asemenea, acesta are posibilitatea de a atasa direct din meniul de creare al unui obiectiv, actiuni specifice obiectivului respectiv, precum si documente necesare sau ajutatoare actiunii, astfel usurand semnificativ procesul de inventariere prezent la acest pas.

Tot acest proces de creare a unui nou obiectiv este impartit pe mai multi pasi, astfel incat interactiunea utilizatorului cu aplicatia respectiv cu interfata grafica a acesteia sa fie una cat mai naturala si intuitiva.

—INSERT PICTURE HERE—

1.3.8 Vizualizarea si accesarea ricurilor din misiuni anterioare

Stabilirea obiectivelor de auditat, fiind un pas relativ important in procesul de audit public, o identificare cat mai precisa si corecta a riscurilor actiunilor acestora este cruciala pentru o buna desfasurare dar si pentru rezultate optime ale misiunii de audit.

Aplicatia ofera auditorului acces la un istoric de misiuni de audit public, in care acesta poate filtra doar misiunile de audit asupra departamentului la care se desfasoara si misiunea de audit curenta, astfel avand posibilitatea de analiza a riscurilor ce deja au fost descoperite, ajutandu-l astfel pe acesta sa stabileasca noi riscuri relevante, corecte si in conformitate cu situatia actuala.

1.3.9 Identificarea si evaluarea riscurilor

Platforma AudIT pune la dispozitia utilizatorului un mecanism de stabilire a riscurilor, acestia avand posibilitatea de a atasa noi riscuri identificate la o actiune, de a edita

valorile riscurilor deja prezente sau de a sterge un risc din tabel in cazul in care acesta nu mai este conform sau o greseala in definirea acestuia a fost depistata.

Aceasta functionalitate este implementata prin intermediul unui tabel paginat cu doar 5 riscuri pe pagina, fiecare linie afisand informatii relevante despre risc, cum ar fi probabilitatea, impactul, scorul final (riscul propriu zis) dar si o scurta descriere a acestuia.

—INSERT PIC HERE —

De asemenea, in partea de jos a tabelului este afisat si numarul total de riscuri atasate actiunii respective, cat si scorul total al tuturor riscurilor, scor care il va ajuta ulterior pe auditor in stabilirea selectarii sau nu a obiectivului in auditare.

1.3.10 Vizualizarea obiectivelor

Ulterior crearii unui nou obiectiv al misiunii de audit, utilizatorul este redirectionat catre o pagina in care acesta poate viziona prin intermediul unui tabel toate obiectivele deja stabilite pentru misiunea respectiva de audit intr-un format cat mai intuitiv, usor de folosit si inteles .

Auditorul poate vizualiza toate actiunile specifice obiectivului pe care il analizeaza, avand in plus si informatii asupra numelui, data ultimei modificari/accesari a acesteia, daca este selectat sau nu in procesul de auditare sau optiunea de a inspecta toate riscurile identificate pana la momentul curent, prin apasarea butonului 'More details' din dreptul coloanei 'Action Risks Details'.

—INSERT PICTURE HERE —

De asemenea, utilizatorul poate naviga prin apasarea butonului 'More details' din dreptul ultimii coloane catre pagina dedicata detaliilor actiunii, in care acesta poate gasi mai multe amanunte referitoare la actiunea selectata.

1.3.11 Pagina detalii actiune

Pagina ofera auditorului o imagine de ansamblu asupra actiunii selectata, astfel acesta poate accesa si vizualiza detalii despre actiune cum ar fi:

- un scurt rezumat al acesteia care contine numele, daca este sau nu selectata in procesul de audit, o lista de Controale Interne Asteptate respectiv o lista de Controale Interne Existente;

- un tabel in care sunt afisate intr-un mod paginat riscurile asociate cu actiunea respectiva, afisand informatii despre impact, probabilitate si scorul riscului;
- un tabel care contine informatii despre diferite FIAP-uri (Fisa de Identificare si Analiza a Problemei) care sunt asociate cu actiunea, afisand informatii despre numele FIAP-ului, perioada de start si de sfarsit a interactiunii, problema, cauza si recomandarea oferita de auditor;
- un tabel in care sunt afisate activitatile desfasurate de auditor avand ca motiv actiunea detaliata pe pagina, tabelul prezentand informatii despre numele activitatii, numele departamentului asupra caruia a avut loc dar si tipul actiunii;

—INSERT PIC HERE—

De asemenea, in partea de jos a fiecarui tabel, utilizatorul are optiunea de a adauga o noua entitate prin apasarea butonului 'Add new'. La apasarea acestuia, se deschide un dialog de tip form in care utilizatorul poate completa campurile pentru a initializa o noua intrare in tabel.

—INSERT PIC HERE—

1.3.12 Salvarea documentelor necesare pe platforma

Inventarierea si accesarea documentelor necesare pentru desfasurarea unei misiuni de audit public este o etapa esentiala pentru asigurarea transparentei si eficientei procesului de audit public.

Platforma AudIT ofera aceste functionalitati utilizatorilor acesteia, astfel incat atat un auditor cat si membri ai departamentelor auditate care au acces la misiunea de audit, sa poata incarca si salveze pe platforma orice tip de document ce nu depaseste o anumita marime in dimensiune prestabilita.

La incarcarea unui astfel de document, auditorul are posibilitatea de a alege intre tipul documentului incarcat: document standard (un fisier de sine statator deja completat) sau document sablon (un fisier care necesita completarea acestuia inainte sau ulterior salvarii acestuia pe platforma).

—INSERT PIC HERE —

1.3.13 Accesarea documentelor necesare pe platforma

Dupa crearea unui document si salvarea acestuia pe platforma, utilizatorii au optiunea de a vizualiza documentele elaborate de acestia respectiv cele la care li s-a oferit accesul.

Afisarea documentelor se face prin intermediul unui tabel, unde sunt afisate informatii cum ar fi: numele documentului, tipul documentului, misiunea de audit de care apartine, starea in care acesta se afla, ultima data la care acesta a fost modificat si numele departamentului caruia ii este adresat (in cazul documentelor sablon).

De asemenea, intrarile din tabel pot fi filtrate si sortate dupa diferite criterii, spre exemplu : alfabetic dupa numele sau tipul documentului, dupa starea in care acesta se afla sau dupa numele misiunii de audit de care acesta apartine.

—INSERT PIC HERE—

1.3.14 Completarea documentelor tip sablon

Una dintre sarcinile de baza ale auditorului, dar si un punct nevralgic al sistemului de audit public despre care am discutat anterior, il constituie nevoie de a inventaria si de a completa numeroase documente de tip sablon. Avand in vedere faptul ca numarul acestor documente este uneori de ordinul zecilor intr-o misiune de audit, o functionalitate care ar permite auditorului sa completeze in mediul digital acest tip de documente ar fi bine venita.

Platforma AudIT ofera posibilitatea utilizatorilor sa completeze si sa editeze direct in aplicatie documentele de tip sablon salvate de acestia. Functionalitatea este implementata prin utilizarea unui simplu Markdown editor, in care documentul sablon este incarcat, editat, iar la finalizarea procesului, schimbarile facute sunt salvate.

—INSERT PIC HERE —

1.3.15 Bara de navigare

Pentru facilitarea unui acces cat mai usor la principalele functionalitati ale platformei, utilizatorul se poate folosi de bara de navigare prezenta intotdeauna in partea stanga a aplicatiei.

Aceasta este impartita in sectiuni specifice fiecarei entitati, astfel incat mentionam:

- sectiunea misiunilor de audit unde regasim optiunea de a crea o noua misiune de audit, de a naviga la misiunea de audit selectata curent, vizualiza lista de misiuni de audit dar si o optiune de a cauta in functie de numele misiunii de audit;
- sectiunea obiectivelor in care exista optiunile de creare a unui nou obiectiv, navigare catre pagina tuturor obiectivelor, crearea a noi actiuni specifice unui obiectiv dar si posibilitatea de a cauta un obiectiv al misiunii de audit curente in functie de numele acestuia;
- sectiunea recomandarilor unde auditorul poate adauga o noua recomandare dar si vizualiza recomandarile deja existente;
- sectiunea documentelor unde similar, merg adaugate noi documente si vizualiza documente deja existente pe platforma;
- sectiunea activitatilor unde utilizatorul poate consemna activitati noi desfasurate sau naviga spre cele existente deja;
- sectiunea dedicata exportarii, unde auditorul poate naviga spre diferite pagini de convertire a obiectivelor, actiunilor si a riscurilor in diferite formate dar si auto-completare a unor documente oficiale de tip sablon, cum ar fi Fise de Identificare a Problemei sau Raport de Evaluarea a Riscurilor;
- sectiunea de control al accesului, unde auditorul sau reprezentantii institutiilor auditate pot consulta resursele la care au acces de scriere sau citire respectiv a oferi acces altor utilizatori la resurse personale;
- sectiunea de configurare, unde un utilizator cu drepturi elevate poate configura institutiile respectiv departamentele inregistrate pe platforma, adaugand, editand sau eliminand instante dintre acestea;

1.3.16 Gestionarea accesului la resurse partajate

O componenta cheie in desfasurarea corecta a unei misiuni de audit este colaborarea intre partile participante la misiunea de audit.

Functionalitatea de acordare a accesului la resurse, contribuie la o colaborare cat mai stransa intre membrii partipanti la misiunea de audit, astfel un auditor poate acorda acces de scriere sau citire la diferite resurse create de acesta.

In plus, o lista completa a accesului primit sau oferit se poate vizualiza de catre utilizator pe o pagina dedicata, unde sub forma unui tabel sunt prezentate informatii specifice cum ar fi numele si tipul resursei si email-ul utilizatorului caruia i s-a oferit acces

1.3.17 Gestionarea activitatilor desfasurate

Prin aceasta functionalitate, auditorul poate consemna orice sarcina pe care acesta o efectueaza pe platforma prin intermediul unei activitati. Aceasta cuprinde informatii referitoare la actiunea asupra careia s-a efectuat o activitate, departamentul asociat dar si tipul activitatii care poate fi asociat unei misiuni, unei actiuni sau pur si simplu o sarcina administrativa.

De asemenea , toate activitatile consemnate intr-o misiune de audit, pot fi vizualizate de catre auditor intr-o pagina dedicata, acestea fiind afisate prin intermediul unui tabel care ofera informatii referitoare la numele acesteia, tipul actiunii sau departamentul asupra cauia s-a realizat respectiva sarcina.

1.3.18 Profilul personal

Pagina profilul personal este locul unde utilizatorul poate sa isi verifice informatiile personale care sunt disponibile pe platforma avand posibilitatea de a le edita. Informatiile cuprind detalii de contact, cum ar fi adresa de email, numar de telefon al institutiei, numar de telefon personal, adresa fizica de contact respectiv daca acesta este verificat sau nu.

1.3.19 Sistemul de notificari

Functionalitatea permite vizualirea notificarilor in ceea ce priveste crearea de noi resurse, primirea accesului la o anumite resursa sau notificari in ceea ce priveste actualizarea sau implementarea unor solutii la recomandarile impuse de auditor din partea reprezentantilor institutiei asupra careia are loc misiunea de audit.

Capitolul 2

Arhitectura platformei

Fiind pus în fața cu provocarea de a elabora o platformă web destinată atât auditorilor publici cât și reprezentatilor instituțiilor din administrația publică, a fost nevoie de o arhitectură dinamică, modulară și ușor de marit în cazul adăugării a noi funcționalități. De asemenea, un număr mare de funcționalități necesită implementarea și a diferite *design pattern-uri*, care se asigură că modulele definite comunică într-un mod cât mai eficient între ele și promovează reutilizarea codului deja scris.

Acest capitol încearcă să descrie motivatia pentru alegerile făcute în materie de tehnologii folosite pentru a dezvolta componentele cheie ale aplicației: frontend, backend stocarea datelor și câteva aspecte legate de securitatea de bază a platformei; cât și o comparație sumară între potențialii competitori ai alegerilor făcute.

2.1 Arhitectura generală

Arhitectura generală a fost gândită de la început într-un mod care permite adăugarea de noi funcționalități în aplicație fără a deteriora structura sau funcționalitățile deja implementate.

Acestea fiind zise, proiectul este împărțit în două mari componente (client și server) care comunică între ele prin intermediul *request-urilor HTTPS*, și anume:

- componenta de server, este alcătuită din diferite *endpoint-uri* Web API care oferă funcționalități clienților săi, aceasta comunicând și cu mediul de stocare al datelor, o bază de date MySQL;
- componenta client, sau interfața grafică a platformei, este o aplicație web interactivă de tipul SPA (*Single page application*), oferind funcționalitățile descrise prin

intermediul *request-urilor* catre server, urmand ca mai apoi sa fie afisate pe pagina web prin intermediul *WebAssembly*;

2.2 Arhitectura serverului

Arhitectura *monolith*

Arhitectura generala a serverului este una de tip *monolith* traditional, impartita pe module, cu dependente slabe intre ele, care comunica intre prin intermediul unor contracte (interfete) bine definite.

Alegerea acestui tip de arhitectura a fost motivata de mai multe avantaje cheie ale acesteia:

- simplitatea dezvoltarii in cadrul arhitecturii de tip monolith permite lucrul pe o singura baza de cod , ceea ce simplifica major procesul de dezvoltare, testare si de depistare a erorilor, fiind esentiala in fazele de inceput al unui proiect;
- performanta sporita in cadrul unui aplicatii care raspunde *request-urilor*, astfel un singur *API* poate raspunde la toate cererile, eliminand nevoia de a activa si alte *API-uri* externe pentru indeplinirea sarcinii, ca in cazul arhitecturii de micro-servicii;
- usurinta testarii de tip *unit-testing* cat si *integration-testing* intrucat toate modulele sunt in acelasi loc;
- depistarea erorilor si rezolvarea lor este mult mai rapida;

Ca in orice alegere pe care o facem, trebuie sa punem in balanta avantajele si dezavantajele pe care aceasta le ofera si sa le comparam strict cu nevoile si problemele pe care incercam sa le rezolvam. Privind in ansamblu si pe termen lung, arhitectura de tip *monolith* prezinta si ea unele dezavantaje :

- dezvoltarea incetinita in momentul in care functionalitatile pe care dorim sa le implementam cresc ca si numar, intrucat toate modulele sunt comasate intr-un singur loc;
- *scalabilitate redusa* datorita stranseii legaturi dintre componentele prezente in aplicatie;
- orice schimbare adusa in materie de noi functionalitati necesita lansarii intregii aplicatii, nu doar a unui singur modul;

Pentru a diminua efectele negative pe care aceste dezavantaje le au asupra intregului proces de dezvoltare a aplicatiei, am incercat implementarea diferitelor solutii in materie de arhitectura, *desing pattern-uri* cretionale, arhitecturale cat si a numeroase practici bune comune in scrierea si mentenanta codului.

Arhitectura *Clean Code*

Arhitectura *Clean Code* este bazata pe ideea principala precum ca stratul de logica interna (*business layer*) este situat central in diagrama circulara, astfel acesta este protejat de schimbari externe. Aceasta proprietate poate fi reformulata astfel incat se defineste **Regula Dependintei** care presupune ca dependintele pot sa fie orientate doar inspre interiorul cercului, astfel niciunul din modulele interioare nu ar trebui sa fie legat in orice fel de un modul exterior acestuia.

—INSERT PIC HERE —

Adoptarea acestui tip de arhitectura impreuna cu cea de tip *monolith* include mai multe beneficii cum ar fi:

- mentenanta sporita datorata faptului ca modulele sau straturile principale ale aplicatiei si logica ce le faciliteaza comunicarea eficienta sunt separate, incurajand astfel si o intelegere mai detaliata si simplificata a intregului sistem;
- flexibilitate din punct de vedere al schimbarii tehnologiilor exterioare, stratul de logica interna este independent de ceea ce se intampla in exteriorul sau;
- testarea componentelor se poate face atat individual cat si in relatie cu alte module, astfel eliminand nevoia de testare a intregii aplicatii;
- stratul de logica interna este independent de baza de date folosita, astfel serverul de stocare al datelor poate fi schimbat cu usurinta;

Modul in care arhitectura de tip *Clean Code* a fost implementata in aceste proiect consta in separarea straturilor aplicatiei, astfel incat avem:

- *Core Layer* fiind structura principala ce confera logica interna a aplicatiei. Acesta cuprinde *Domain* unde sunt modelate entitatile aplicatiei respectiv *Application* unde

este definita toata logica interna a serverului, de la declaratiile abstracte ale interfetelor la definirea serviciilor proprii de care se vol folosi ulterior straturile externe ale arhitecturii;

- *API Layer* este partea structurala care defineste *endpoint-urile* aplicatiei prin diferite *controllere*, expunand astfel functionalitatile aplicatiei la internet;
- *UI Layer* defineste structura de prezentare a aplicatiei si este formata din parte de *Frontend* a aplicatiei;
- *Infrastructure Layer* in care gasim logica ce se ocupa de comunicarea cu serviciile externe cum ar fi baza de date, AWS sau servicii de identitate;

***Design pattern-uri* utilizate**

Design pattern-urile, potrivit definitiei, sunt solutii generale si reutilizabile asupra problemelor comune ce pot aparea in decursul dezvoltarii unei aplicatii software.

Utilizare lor conduce la o buna mentananta a codului scris, posibilitatea de a reutiliza module deja scrise, imbunatateste comunicare si legaturile dintre module si incurajeaza un stil de cod cat mai elegant si usor de inteles.

In implementarea aplicatiei au fost folosite *desing pattern-uri* din diferite categorii astfel incat in aceasta subsectiune se vor discuta cateva exemple utilizate.

Optional pattern

Acesta este un model de proiectare care ajuta la gestionarea valorilor care pot sau nu fi prezente, astfel avand valoarea *null*. *Pattern-ul* se asigura ca este eliminata valoarea *null*, care de cele mai multe ori este o sursa comuna in erori la rularea codului (*runtime*). Solutia prezentata de acest model este crearea unei clase *template* care incapsuleaza valoarea propriu zisa a entitatii pe care o construim. Spre exemplu, in cazul in care vrem sa cream o noua entitate de tipul *User* dar la *runtime* apare o eroare, executia programului nu se va opri, iar valoarea entitatii va fi incapsulata intr-un tip *Result<User>*

cu parametrul *Succes* setat pe fals, indicand astfel ca procesul de creare a esuat.

–INSERT PIC HERE –

Mediator pattern

Acest *design pattern* este unul de tip comportamental si se asigura ca nu exista dependinte haotice intre entitatile/clasele din codul scris. Modelul restrictioneaza comunicarea directa intre obiecte si le obliga sa interactioneze doar prin intermediul unui *mediator*.

Acest model este implementat prin utilizarea unei interfate sau clase abstracte care stie toate referintele la componentele care vor sa comunice. In acest mod, in loc sa trimita solicitari directe, un obiect comunica prin intermediul mediatorului, acesta stiind unde sa redirectioneze cererea primita.

–INSERT PIC HERE –

Command pattern

Acest *design pattern* este de asemenea unul de tip comportamental si se utilizeaza partial de *design pattern-ul Mediator*, transformand astfel o comanda, spre exemplu o cerere de creare a unei noi entitati, intr-un obiect independent, acesta ulterior fiind trimis catre mediator si executat in *handler-ul* corespunzator acestuia, de obicei numit *receiver*. In contextul dezvoltarii partii de server a aplicatiei AudIT, acesta este utilizat, impreuna cu modelul Mediator pentru a delega orice comanda (*request*) primita catre obiectul care stie sa o execute. In acest mod, se elimina dependinte stranse intre obiecte, promovand un cod cat mai bine organizat si elegant.

–INSERT PIC HERE –

Repository pattern

Folosit in special in dezvoltarea aplicatiilor web, acest *design pattern* separa logica interna a aplicatiei de accesul direct la date (baza de date). Acesta se utilizeaza de interfate pentru a crea un strat separator intre declararea abstracta a acestor structuri si implementarea concreta a functiilor care acceseaza datele la nivel de baza.

Prin acest model, comunicarea dintre module se realizeaza prin intermediul contractului foarte bine stabilit, astfel eliminand posibilitatea ca un modul abstract sa acceseze direct

un modul de acces de date.

–INSERT PIC HERE –

Tehnologii utilizate

În dezvoltarea partii de server s-a utilizat *framework-ul* ASP.NET Core împreună cu limbajul C#.

Alegerea făcută se bazează pe faptul că *framework-ul* este unul de tip *open-source*, dezvoltat în principal de Microsoft, un actor important pe piața actuală IT, *framework-ul* oferind funcționalități robuste și eficiente pentru dezvoltarea aplicațiilor web, acestea putând fi rulate pe Windows, Linux cât și MacOS.

De asemenea, C# este un limbaj de programare multi-paradigma *high-end* matur, care oferă diverse funcționalități și soluții foarte bine puse la punct atât din punct de vedere al eficienței cât și al sustenabilității codului.

Mai mult de atât, integrarea celor două tehnologii cu alte servicii Microsoft este una foarte ușor de realizat, acest lucru aducând un motiv în plus în alegerea făcută, pe lângă ecosistemul bogat din care acestea fac parte, oferind suport pentru o gamă largă de biblioteci și unelte deja integrate în funcționalitățile limbajului și a *framework-ului*.

În plus, s-au utilizat diferite biblioteci pentru dezvoltarea anumitor servicii, cum ar fi :

- AWS SDK.Core pentru integrarea serviciilor AWS de trimiterea a unui email sau de stocare a datelor și a fișierelor în S3 Bucket;
- MediatR care oferă funcționalitățile *design pattern-ului* Mediator;
- ASP.NET Identity, serviciu pentru integrarea funcționalităților de autentificare și autorizare în aplicație, facilitând accesul și autorizarea utilizatorilor pe platformă;
- AutoMapper este o bibliotecă externă utilizată pentru a transforma diferitele tipuri de obiecte între ele, eliminând astfel codul *boilerplate* necesar pentru a copia datele de la o entitate la alta;
- OpenXML este o bibliotecă *open-source* care permite manipularea fișierelor Office de tipul .XLSX sau .DOCS. Este integrată pentru a oferi utilizatorilor funcționalitatea de a putea edita sau exporta pe platformă documente oficiale tip sablon;

2.3 Arhitectura interfatei grafice

Interfata grafica dezvoltata in acest proiect este realizata in *framework-ul* .NET Blazor, mai exact o aplicatie de tipul *WASM (Web Assembly)* care ruleaza in *browser-ul* utilizatorului. Alegerea a fost facuta intrucat acest nou model arhitectural permite rularea codului direct in *browser*, nemaifiind nevoie de librarii sau extensii suplimentare necesare.

Web assembly

Cum ne putem da seama si din numele pe care aceasta tehnologie o poarta, este vorba despre *byte-code* (cod-masina) care este executat de *engine-ul* JavaScript direct in *browser*.

Wasm nu este un limbaj, ci mai degraba produsul compilarii codului scris intr-un limbaj de programare in cod-masina executabil. Majoritatea limbajelor de programare moderne suporta compilarea codului direct intr-un fisier binar de tip *.wasm* . Dupa o compilare cu succes, fisierele binare *.wasm* sunt incarcate in *browser* unde sunt executate de catre motorul JavaScript specific intr-o instanta virtuala izolata si securizata.

—INSERT PIC HERE —

Un avantaj cheie al utilizarii acestei tehnologii il constituie viteza si eficienta de care da dovada. Codul ruleaza cat se poate de aproape de limitarile *hardware* ale computer-ului, astfel rezultand in performante crescute si un nivel al utilizarii memoriei mai mic.

Pe de alta parte, exista si dezavantaje in utilizarea acestuia, intrucat momentan nu este implementat un sistem de *garbage collector* care sa elimine din memorie functiile si instantele care nu mai sunt folosite in firul executiei.

.NET Blazor

Pentru dezvoltarea paginilor web, s-a utilizat tehnologia .NET Blazor, un *framework* relativ nou aparut, dar care promite performante crescute alaturi de un mediu de lucru bine pus la punct si eficient din punct de vedere al productivitatii dezvoltarii interfatelor interactive si bogate.

Paginile web create in acest mod combina codul C# cu HTML pentru a crea componente reutilizabile ce sunt afisate si actualizate in DOM-ul virtual din *browser*.

–INSERT PIC HERE–

O functionalitate foarte importanta pe care o ofera, este utilizarea asa ziselor componente, parti fundamentale care impreuna alcatuiesc pagina web afisata utilizatorului. O componenta poate fi vazuta ca o piesa esentiala de puzzle care contribuie la afisarea finala a unei pagini web, piesa de puzzle care la randul sau poate fi creata din mai multe astfel de componente, si asa mai departe. Un avantaj pe care Blazor il ofera dezvoltatorilor este acela ca elimina pe cat posibil utilizarea de functii si *script-uri* JavaScript in crearea paginilor web. Blazor se foloseste de cod scris in C# impreuna cu HTML si CSS pentru a elabora pagini si componente web interactive si fluide.

Radzen Components

De asemenea, pentru o stilizare si o reutilizare a componentelor, s-a folosit biblioteca Radzen Components pentru a folosi controale si componente cum ar fi : tabele, butoane stilizate, teme grafice de inalta calitate, *form-uri* si multe altele.

Libraria Radzen Components este una de tip *open-source* si ofera suport dedicat din partea comunitatii active, impreuna cu documentatie detaliata asupra tuturor functionalitatilor oferite.

2.4 Stocarea datelor

Baza de date MySQL

MySQL este un sistem de gestionare a bazelor de date relationare open-source si reprezinta alegerea pentru care am optat a fi folosita pentru stocarea datelor in acest proiect. Acest sistem este una dintre cele mai populare solutii cand vine vorba de stocare intr-un mod relational al datelor, astfel incat acesta ofera diferite beneficii:

- simplitatea cand vine vorba de utilizarea acestuia, folosindu-se de un dialect comun in interogările sale, este foarte usor de utilizat;
- securitatea datelor oferita de MySQL prin integrarea unui sistem solid de privilegii si de restrictionarea a accesului bazat pe roluri;

- flexibilitatea si functionalitatile pe care acesta le ofera sporesc productivitatea dezvoltarii aplicatiilor;

Entity Framework Core

Entity Framework Core este o tehnologie dezvoltata de Microsoft in cadrul *framework-ului* .NET Core care permite dezvoltatorilor sa interactioneze cu entitatile si tabelele din baza de date prin intermediul obiectelor, eliminand astfel necesitatea de a scrie interogari clasice pentru a comunica facil cu baza de date si cu obiectele sale.

Aceasta tehnologie dispune de o serie de caracteristici si functionalitati care o fac o alegere cruciala in ceea ce priveste comunicarea intr-un mod eficient cu baza de date:

- sistemul de *migratii* similar unui *version-control* al versiunii bazei de date si a relatiilor acesteia, permite dezvoltatorilor sa tina evidenta versiunii bazei de date, eventual existand posibilitatea in cazul unor erori sa se intoarca la ultima versiune stabila;
- *code-first* este functionalitatea ce permite actualizarea modelului din baza de date pe baza schimbarilor din codul scris si a modificarilor din entitatile declare in codul sursa;
- suportul pentru majoritatea bazelor de date existente in momentul de fata, primind constant actualizari;
- performanta interogarilor, astfel incat sistemul este optimizat sa colecteze intr-un mod eficient rezultatele interogarilor;

2.5 Aspecte de securitate

adasd

Concluzii

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nunc mattis enim ut tellus elementum sagittis vitae et. Placerat in egestas erat imperdiet sed euismod. Urna id volutpat lacus laoreet non curabitur gravida. Blandit turpis cursus in hac habitasse platea. Eget nunc lobortis mattis aliquam faucibus. Est pellentesque elit ullamcorper dignissim cras tincidunt lobortis feugiat. Viverra maecenas accumsan lacus vel facilisis volutpat est. Non odio euismod lacinia at quis risus sed vulputate odio. Consequat ac felis donec et odio pellentesque diam volutpat commodo. Etiam sit amet nisl purus in. Tortor condimentum lacinia quis vel eros donec. Phasellus egestas tellus rutrum tellus pellentesque eu tincidunt. Aliquam id diam maecenas ultricies mi eget mauris pharetra. Enim eu turpis egestas pretium.