

Oferta Tehnică

Dezvoltarea software pentru realizarea si automatizarea fluxurilor de procesare în cadrul procesului de evidență a aeronavelor înmatriculate

RFP NO.

**S&T Mold**

Str. Calea Ieșilor 8
MD-2069, Moldova
t: +373 22 837960
e: office@snt.md
w: www.snt.md

Table of Contents

1. Rezumat	3
2. Detalii tehnice	4
2.1. Principii de arhitectură și proiectare.....	4
2.2. High Level Design	7
2.3. Tehnologii și Platforme.....	8
3. Detalii dezvoltare	9
3.1. Agile/SCRUM	9
3.2. Metoda Waterfall	9
3.3. Tehnici și Abordări pentru Dezvoltarea Software-ului	10
3.4. DevOps instrumente.....	11
3.5. Tehnici și Abordări pentru Asigurarea Calității.....	11
4. Managementul proiectului.....	12
4.1. Inițiere.....	12
4.2. Planificare.....	13
4.4. Execuția Proiectului	13
4.5. Monitorizare și Control.....	14
4.6. Închiderea Proiectului	14
5. Controale Specifice Proiectului	14
6. Executarea Cerințelor Funcționale	15
8. Cerințe pentru instruirea utilizatorilor sistemului	19
9. Backup și recuperarea în caz de dezastru.....	20
10. Testare Automatizată	20
11. Procedura de acceptare a sistemului	20
12. Concluzie	21

1. Rezumat

S&T Mold dorește să își exprime interesul pentru proiect și să prezinte abordarea și expertiza tehnică ce vor fi utilizate pentru executarea și finalizarea cu succes a proiectului.

S&T Mold operează în Moldova din 1995 și are un portofoliu larg de soluții livrate atât pentru sectorul public, cât și pentru cel privat, care include sisteme de tip enterprise critice.

Compania deține competențele și abilitățile necesare pentru finalizarea cu succes a acestui proiect și este pregătită să le utilizeze în modul și stilul descris în continuare în acest document.

2. Detalii tehnice

Abordăm proiectarea și construirea sistemelor informatice prin echilibrarea tehnologiilor fundamentale și dovedite cu metode moderne și inovatoare de dezvoltare și scalare a aplicațiilor software. De asemenea, valorificăm vasta noastră experiență de două decenii în dezvoltarea și livrarea de software pentru organizații mari, lecțiile învățate, abilitățile dobândite, inteligența de cod și experiența considerabilă de colaborare cu echipele donatorilor internaționali și oficialii guvernamentali, care ne-a permis să construim un portofoliu extins de proiecte livrate cu succes.

În capitolele următoare, vom prezenta principiile și tehnologiile pe care intenționăm să le utilizăm în implementarea proiectului actual.

2.1. Principii de arhitectură și proiectare

2.1.1. Arhitectură-Orientată-Servicii

Sistemul va fi proiectat și construit ca o arhitectură orientată spre servicii folosind microservicii.

2.1.2. SOLID

Sistemul va fi proiectat pe baza principiilor SOLID, ceea ce va facilita întreținerea și extensia acestuia.

2.1.3. Domain Driven Design

Vom folosi Domain Driven Design pentru a colabora continuu cu experții în domeniu, cu scopul de a îmbunătăți modelul aplicației și de a abstractiza corect entitățile și procesele.

2.1.4. Microservicii

Arhitectura microserviciilor adoptă principiul responsabilității unice (Single Responsibility Principle) și îl extinde la serviciile slab cuplate care pot fi dezvoltate, implementate și întreținute independent. Microserviciile pot fi scalate independent, spre deosebire de aplicațiile monolitice gigantice care se scală împreună. Aceasta înseamnă că o zonă funcțională specifică, care necesită mai multă putere de procesare sau lățime de bandă de rețea pentru a susține cererea, poate fi scalată, evitând astfel

scalarea inutilă a altor zone ale aplicației. Microserviciile oferă, de asemenea, o izolare îmbunătățită a erorilor, astfel încât, în cazul unei erori într-un serviciu, întreaga aplicație nu trebuie neapărat să înceteze să funcționeze.

2.1.5. Containere (Docker)

Contenerizarea microserviciilor va permite ca acestea să fie livrate împreună cu toate fișierele de configurare, bibliotecile și dependențele necesare pentru ca un microserviciu să funcționeze eficient și corect în diferite medii de calcul.

Avantajul unei abordări orientate pe containere pentru dezvoltare și implementare este că elimină majoritatea problemelor cauzate de configurările inconsistente ale mediilor și problemele asociate acestora. În plus, containerele permit o scalare rapidă a aplicației prin instanțierea de noi containere după cum este necesar.

2.1.6. Orchestrarea containerelor și scalarea automată prin Kubernetes

Vom configura clustere Kubernetes pentru a găzdui și scala automat sistemul. Kubernetes este un sistem open-source pentru automatizarea implementării, scalării și gestionarea aplicațiilor containerizate. Oferă un cadru pentru rularea sistemelor distribuite într-un mod rezilient. Se ocupă de scalarea automată pe baza consumului de memorie și/sau CPU, auto-repararea prin repornirea automată a containerelor care eșuează sau nu răspund la verificările definite de utilizator, nu promovează containerele clienților până când acestea nu sunt pregătite să răspundă, echilibrul automat al traficului (load balancing) și descoperirea serviciilor, și multe altele.

2.1.7. Procesarea fluxurilor de date asincrone

Microserviciile care vor colabora în procesarea datelor vor schimba mesaje prin intermediul fluxurilor de date asincrone persistente, ceea ce va decupla consumatorii și producătorii de date, va permite proiectarea și implementarea ușoară a pipeline-urilor de procesare complexe și va permite sistemului să se scaleze natural prin realocarea resurselor de la pipeline-urile inactivate la cele ocupate.

2.1.8. Limbaje și tehnologii de programare

Vom dezvolta sistemul folosind limbaje de programare consacrate și de top în industrie pentru dezvoltarea aplicațiilor enterprise.

- Java pentru back-end
- JavaScript pentru front-end

De asemenea, toate framework-urile și bibliotecile utilizate vor fi de top în industrie, populare și open-source. Nu vor fi utilizate limbaje sau framework-uri cu o adoptare redusă.

2.1.9. Aplicație cu o Singură Pagină (Single Page Application)

Aplicația cu o singură pagină funcționează în browser și nu necesită reîncărcări de pagină sau timp suplimentar de așteptare. Cele mai multe resurse (HTML/CSS/Scripte) sunt încărcate o singură dată pe parcursul duratei de viață a aplicației. Deoarece aplicațiile cu o singură pagină nu actualizează întreaga pagină, ci doar conținutul necesar, ele îmbunătățesc semnificativ viteza aplicației. Doar datele sunt transmise înainte și înapoi. Aplicațiile web cu o singură pagină se potrivesc perfect pentru construirea de platforme dinamice și aplicații enterprise.

2.1.10. Jurnal de Audit Complet

Jurnalele de audit înregistrează fiecare acțiune a utilizatorului într-un sistem, oferind un istoric complet al operațiunilor din cadrul aplicației. Prin urmare, jurnalele de audit sunt o resursă valoroasă pentru administratori și auditori care doresc să diagrameze și să depaneze probleme sau să examineze activitatea utilizatorului.

Jurnalele de audit contribuie și la securitate, deoarece oferă înregistrări ale tuturor activităților utilizatorilor, inclusiv activități suspecte. Jurnalele de audit pot ajuta la monitorizarea datelor și sistemelor pentru posibile breșe de securitate sau vulnerabilități și la identificarea utilizării incorecte a datelor interne. Ele pot oferi înregistrări care pot servi drept dovezi în cazul unor dispute.

2.1.11. Securizat prin Design

Sistemul va implementa cele mai recente standarde de securitate și va urma recomandările OWASP la toate nivelurile. De asemenea, va utiliza cele mai noi tehnologii și va menține sistemul la zi cu toate patch-urile de securitate emise pentru bibliotecile de bază. Folosind instrumente automate de urmarire a conformității de securitate ca parte a procesului de integrare continuă, echipa de dezvoltare va putea să monitorizeze securitatea aplicației împotriva vulnerabilităților cunoscute și să rezolve problemele înainte de lansarea aplicației.

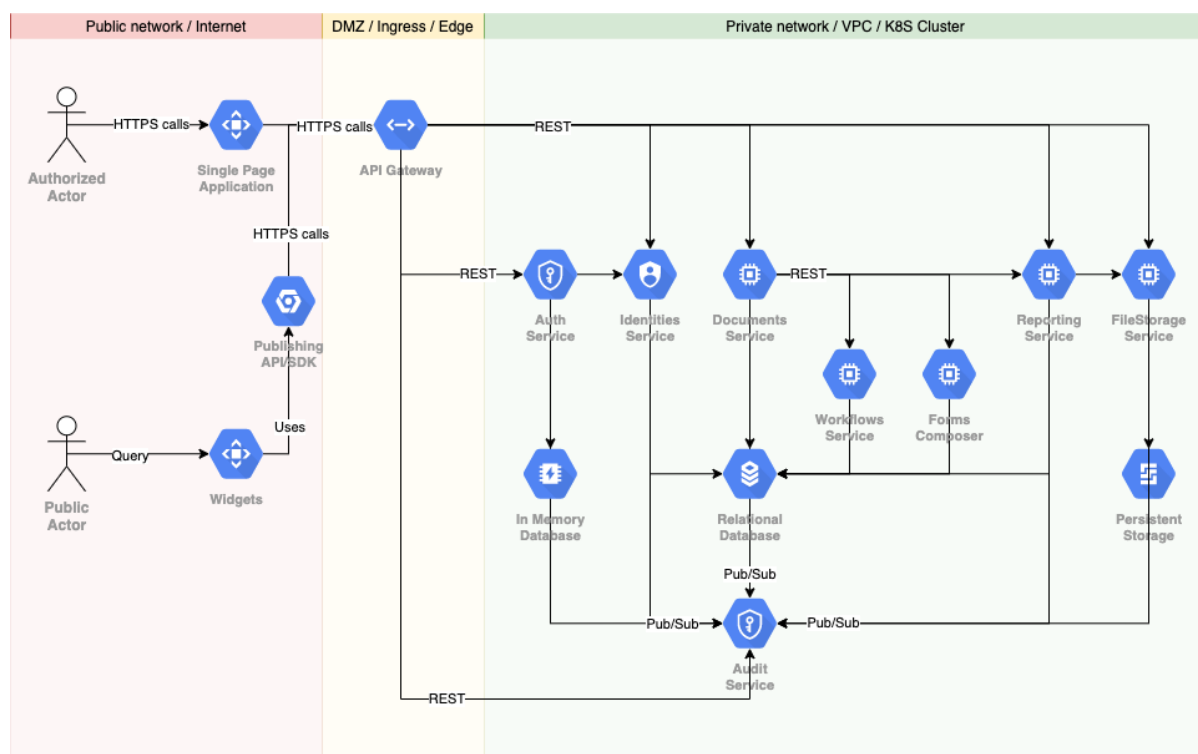
2.1.12. Interconectivitate cu alte sisteme

Sistemul va fi capabil să comunice ușor cu alte sisteme atât ca și consumator, cât și ca producător de web-servicii, utilizând protocoale standard: REST și SOAP.

De asemenea, echipa noastră are o vastă experiență în integrarea cu MPass, MSign, MLog, MPay, MNotify, MConnect și a realizat integrații cu majoritatea serviciilor MCloud.

2.2. High Level Design

Următoarea diagramă prezintă o vedere de ansamblu la nivel înalt a arhitecturii soluției propuse și modul în care containerele comunică între ele. Fiecare componentă este o unitate containerizată, rulabilă/deployabilă separat, care reprezintă o aplicație cu o singură pagină, un microserviciu, o bază de date etc.



Prin utilizarea serviciilor containerizate, software-ul poate fi implementat în cloud, într-un cluster Kubernetes sau pe masini virtuale (utilizând un runtime de containere).

2.3. Tehnologii și Platforme

Componentă	Descriere	Tehnologii
Single Page Application	Aplicația principală (Back Office)	JavaScript (ECMAScript 2023), ReactJS 18.2.0
Publishing API/SDK	API-ul este organizat în jurul REST/SOAP și este utilizat pentru a prelua date și informații publice (Front Office)	JavaScript (ECMAScript 2023)
API Gateway	Punct de intrare echilibrat la încărcare, responsabil pentru filtrarea interogărilor invalide și redirecționarea cererilor către serviciile necesare	Java 21 / Cloud API Gateway
Auth Service	Serviciul de autentificare a utilizatorilor	Java 21
Identities Service	Gestionarea utilizatorilor	Java 21
Documents Service	Gestionarea documentelor	Java 21
Reporting Service	Serviciul de gestionare a rapoartelor. Responsabil pentru stocarea șabloanelor de rapoarte și generarea rapoartelor pe baza parametrilor de intrare	Java 21
FileStorage Service	Responsabil pentru persistarea fișierelor încărcate și streamingul fișierelor pentru descărcare	Java 21
Workflows Service	Responsabil pentru fluxurile de lucru	Java 21
Audit Service	Responsabil pentru înregistrarea tuturor activităților utilizatorilor	Java 21
In-Memory Database	Stocarea detaliilor fiecărei sesiuni de utilizator	Redis
Relational Database	Stocarea datelor aplicației (detalii despre aeronave, documente, taxonomii, utilizatori etc.)	PostgreSQL
Persistent Storage	Stocarea fișierelor încărcate / generate	Cloud Storage

3. Detalii dezvoltare

Abordarea principală a echipei S&T este să urmeze cele mai bune practici din industrie, metodologiile dovedite și know-how-ul extensiv pe care compania l-a dobândit în proiecte similare de-a lungul anilor, care s-au transformat în principii organizaționale.

3.1. Agile/SCRUM

Plănuim să aplicăm metodologia Agile numită SCRUM pentru execuția acestui proiect, care promovează următoarele principii:

- Toate sarcinile sunt formalizate conform planului de proiect și adăugate în backlog, care reprezintă o listă neordonată a tuturor sarcinilor ce urmează a fi executate.
- Lucrul este împărțit în sprinturi, fiecare sprint având o durată de 2 săptămâni. La începutul sprintului, sunt stabilite obiectivele pentru livrabilele care vor fi predate clientului la sfârșitul celor 2 săptămâni. Sarcinile sunt prioritizate, estimate și mutate din backlog în sprinturi.
- Sarcinile sunt atribuite membrilor echipei.
- În fiecare dimineață au loc întâlniri stand-up, în care fiecare membru al echipei își actualizează colegii cu privire la progresul realizat.
- La sfârșitul sprintului, livrabilele sunt prezentate clientului, iar feedback-ul primit este transformat în sarcini pentru sprintul următor, dacă este necesar.

Această abordare oferă avantaje importante pentru client, principalul fiind că acesta poate monitoriza îndeaproape progresul și poate primi livrabile funcționale la fiecare 2 săptămâni, pe baza cărora poate corecta direcția, dacă este necesar. Acest lucru contrastează puternic cu abordarea iterativă planificată, care are cicluri mult mai lungi și prezintă riscul ca, la final, clientul să obțină un rezultat diferit de cel anticipat.

3.2. Metoda Waterfall

Uneori, metoda Agile de gestionare a proiectelor este dificilă de implementat cu anumiți clienți. Pentru astfel de cazuri, putem implementa un proces Waterfall împreună cu Managerul de Proiect al Clientului/Beneficiarului. Toți pașii și metodologia pot fi aplicate rapid de echipa noastră. Managerii noștri de proiect au pregătire formală oferită de PMI atât în metodele Agile, cât și în cele clasice.

3.3. Tehnici și Abordări pentru Dezvoltarea Software-ului

Practica noastră de dezvoltare software a fost stabilită și îmbunătățită de-a lungul anilor, integrând toate cele mai bune practici relevante și standardele industriei, precum și lecțiile învățate de noi. Acum avem un proces complex și sigur de scriere a software-ului, cu părți mari automatizate complet și numeroase verificări de siguranță implementate pentru a detecta problemele cu codul cu mult înainte ca acestea să ajungă în producție sau să perturbe planurile.

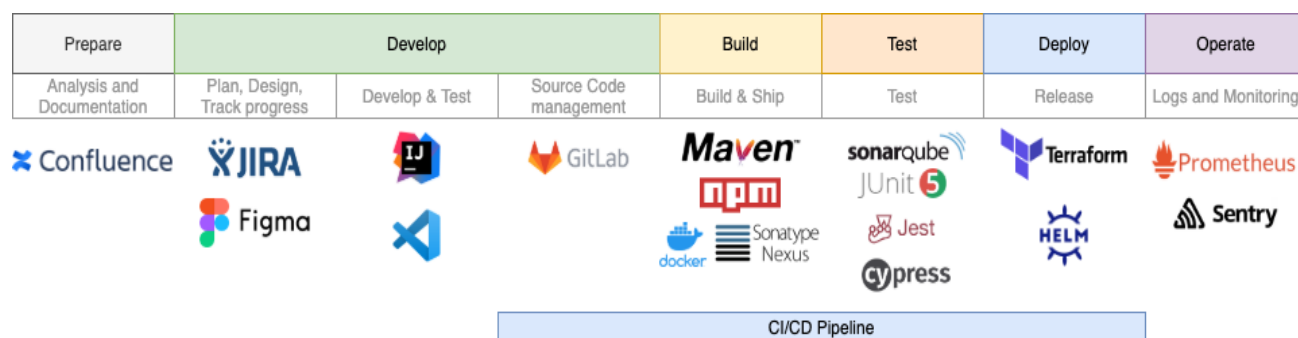
Printre cele mai bune practici pe care le aplicăm se numără următoarele:

1. **Revizuirea codului în mai multe etape.** Niciun cod nu ajunge în repository fără a fi revizuit și îmbunătățit de cel puțin un membru senior al echipei.
2. **Refactorizare continuă**, odată la fiecare 3-5 sprinturi. Periodic, oprim toată munca nouă și îmbunătățim ceea ce a fost deja realizat pentru a preveni anumite probleme și a evita acumularea de datoriile tehnice.
3. **Acoperire cu teste unitare.** Avem praguri obligatorii de acoperire cu teste unitare, iar dacă orice commit nu respectă aceste praguri, nu este acceptat în repository.
4. **Test-Driven-Design.** Aplicăm TDD acolo unde este relevant, garantând astfel că funcționalitățile dezvoltate corespund criteriilor de acceptare respective.
5. Avem configurate pipeline-uri de **Integrare Continuă / Livrare Continuă** pentru fiecare tip de proiect pe care îl dezvoltăm. Acest lucru garantează livrarea sigură a tuturor componentelor sistemului către mediile relevante.
6. Avem o practică de front-end foarte bine stabilită, care se traduce în **UI/UX** excelent, apreciat de clienți. Interfețele noastre sunt concepute pentru a fi intuitive, moderne și receptive. A fost întotdeauna un punct forte care ne-a diferențiat în fața concurenței.
7. Folosim instrumente de **prototipare** foarte puternice și proiectăm interfețele împreună cu clienții noștri, procedând la implementare doar după ce clientul aprobă prototipurile.

8. **Procese QA riguroase** - testarea regresiei, testarea sistemului, testarea de acceptare și testarea de acceptare a utilizatorilor sunt toate procese bine stabilite care ne permit să livrăm produse de înaltă calitate.
9. **Cele mai bune practici de securitate:** urmărim îndeaproape ghidurile OWASP, codul nostru este securizat din design și avem un număr de instrumente automate care rulează verificări de securitate pentru fiecare build și identifică vulnerabilitățile.

3.4. DevOps instrumente

Acestea sunt unele dintre instrumentele pe care le utilizăm în procesele noastre de dezvoltare software. Printre beneficiile utilizării acestor instrumente se numără frecvența îmbunătățită a livrărilor, rata mai mică de eșec a noilor versiuni, timpul mediu mai scurt de recuperare în caz de eșecuri ale livrărilor și un timp mai rapid până la lansare.



3.5. Tehnici și Abordări pentru Asigurarea Calității

Testarea este o parte integrantă și importantă a ciclului de viață al dezvoltării, care asigură funcționarea corectă a întregului produs și că atât cerințele funcționale, cât și cele non-funcționale sunt îndeplinite.

Pentru a asigura cea mai înaltă calitate a produsului și a minimiza timpul petrecut pentru remedieri, echipa implementează controlul calității continuu, mai degrabă decât testarea simplă.

În timpul fazei de codare, calitatea este asigurată prin aplicarea principiilor TDD, scrierea de teste unitare, revizuire continuă între colegi și sesiuni de refactorizare a codului. Echipa utilizează, de asemenea, instrumente automate de inspectare a codului și de calitate a codului, cum ar fi SonarQube și ESLint.

În timpul fazei de testare, sunt implementate atât strategii de testare manuală, cât și automată, care includ: **testare unitară**, **testare a componentelor**, **testare de**

integrare, testare de acceptare și testare de performanță, pentru a acoperi atât cerințele funcționale, cât și cele non-funcționale.

Testarea manuală acoperă consistența logică și vizuală a produsului și permite explorarea scenariilor de utilizare non-standard.

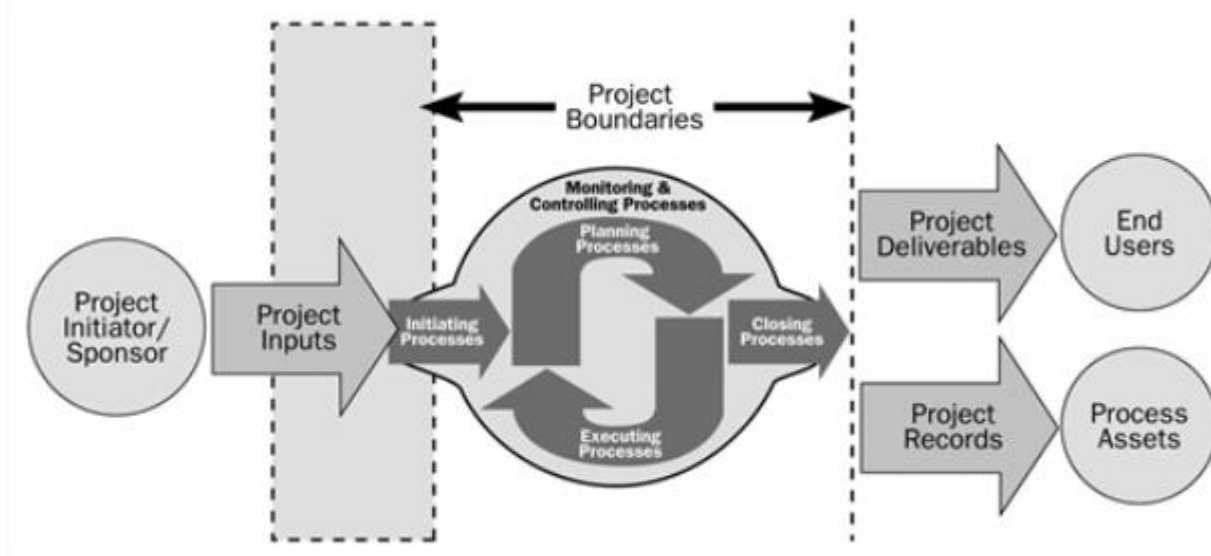
Testarea automată permite executarea rapidă, în paralel, a mai multor scenarii standardizate cu puncte de date variate.

Pentru fiecare proiect, se creează un plan de testare cuprinzător, care acoperă cerințele funcționale și non-funcționale ale aplicației, strategiile de testare, instrumentele, documentele, punctele de date și alte resurse necesare pentru a asigura funcționarea corectă a acestora.

4. Managementul proiectului

Proiectele sunt gestionate în conformitate cu ghidurile și recomandările dezvoltate de The Project Management Institute (PMI) și corpusul său de cunoștințe PMBOK, monitorizând continuu domeniul de aplicare, etapele, costul proiectului, inclusiv rolurile și responsabilitățile părților interesate.

În cadrul unui proiect, toate procesele pot fi împărțite în 5 grupuri distincte: Inițiere, Planificare, Execuție, Monitorizare și Control, Închidere.



4.1. Inițiere

Acest grup de procese include lucrările de bază necesare pentru a crea proiectul și a defini ghidurile și criteriile sub care acesta va funcționa.

În cadrul etapei de Inițiere, are loc semnarea contractului, sunt identificate părțile interesate atât în interiorul, cât și în afara organizațiilor implicate, se creează domeniul de aplicare al proiectului și se alocă resursele financiare necesare pentru începutul proiectului.

În cadrul etapei de licitație, echipa care va livra proiectul este formată și se stabilește data oficială de început a proiectului (întâlnirea de lansare), la care vor participa oficialii cheie de ambele părți.

4.2. Planificare

După autorizarea proiectului, acesta trebuie planificat. Această fază produce Planul de Management al Proiectului. Acesta este documentul principal de planificare care stabilește așteptările părților interesate și clarifică modul în care proiectul va fi gestionat. Ar trebui să contureze domeniul de aplicare al proiectului, costurile, resursele, termenele limită, etapele, nevoile de comunicare și orice alte documente necesare pentru livrarea cu succes a proiectului.

4.3. Prototipizare

Această fază presupune lucrul la prototipuri interactive de înaltă fidelitate împreună cu părțile interesate. Echipa va efectua multiple iterații de prototipizare UI/UX, va prezenta prototipul părților interesate, va colecta feedback și îl va aplica pe prototip. În final, prototipul rezultat va fi o reprezentare de înaltă fidelitate a produsului final, permițând beneficiarului și altor părți interesate să aibă o imagine clară asupra aspectului rezultatului și să aplice eventuale corecții în planul de lucru încă din faza timpurie.

4.4. Execuția Proiectului

Această fază este cea în care se desfășoară lucrările tehnice ale proiectului. Echipa de proiect este constituită și începe lucrul, iar producția livrabilelor proiectului este pusă în mișcare.

Execuția proiectului necesită coordonarea resurselor umane, gestionarea așteptărilor părților interesate și abordarea schimbărilor de proiect. Managerul de proiect trebuie să fie la curent cu problemele care apar, precum și să facă previziuni regulate privind problemele viitoare legate de program și costuri pentru a aborda schimbările cât mai din timp posibil. Cererile de schimbare trebuie gestionate și documentate pe parcursul acestei faze, iar părțile interesate trebuie să fie ținute la curent.

Actualizările de status și alte comunicații ale proiectului sunt trimise părților interesate relevante conform planului de management al proiectului. Documentele sunt stocate și arhivate, iar părțile interesate sunt gestionate conform planului.

4.5. Monitorizare și Control

Pe parcursul proiectului, managerul de proiect trebuie să monitorizeze și să controleze lucrările pentru a se asigura că livrabilele proiectului sunt realizate la timp, în cadrul bugetului și cu o calitate acceptabilă. De asemenea, părțile interesate trebuie menținute satisfăcute și echipa de proiect trebuie să fie menținută motivată și coezivă. Monitorizarea și controlul lucrărilor proiectului se desfășoară simultan cu faza de execuție, astfel încât cele două grupuri de procese se suprapun.

Monitorizarea timpului (termeni limită, repere etc.) și a costurilor se face cel mai frecvent prin Analiza Valorii Efectuate (Earned Value Analysis), care oferă un semnal de alarmă timpuriu pentru abaterile din aceste domenii. Calitatea livrabilelor, comunicarea cu părțile interesate și problemele cu un potențial mare de risc sunt alte domenii de monitorizare regulată. În orice moment, monitorizarea poate duce la modificări ale proiectului.

Dacă sunt necesare modificări în orice parte a proiectului așa cum este documentat în planul de management al proiectului, acestea trebuie documentate și vor rezulta într-un plan actualizat. Aceasta include modificări ale termenelor limită, costurilor, livrabilelor și orice alte schimbări în proiectul așa cum a fost previzionat.

4.6. Închiderea Proiectului

De obicei, există întotdeauna o serie de sarcini implicate în închiderea proiectului și trecerea la etapele următoare, iar acestea sunt de obicei foarte vizibile pentru executivi și sponsorii proiectului. Obligațiile contractuale trebuie finalizate și contractele închise, detaliile finale trebuie trimise, iar cerințele de finanțare trebuie finalizate.

5. Controale Specifice Proiectului

Echipa noastră a înțeles complet și a absorbit necesitatea creării unui nou Sistem Informațional „Registrul Aerian al Republicii Moldova”. Sistemul final furnizat de noi va beneficia de toate avantajele arhitecturii solicitate, cum ar fi scalabilitate extremă, performanță, flexibilitate și securitate.

Următoarele module/funcționalități vor fi analizate împreună cu beneficiarul și va fi implementată o versiune modernă a acestora:

- Crearea unei arhitecturi moderne a aplicației, cu flexibilitatea, fiabilitatea și securitatea necesară
- Crearea dosarului electronic al aeronavelor înregistrate
- Integrare cu instrumente de raportare

- Integrare cu furnizori externi de informații
- Integrare cu MConnect și alte servicii guvernamentale

Toate integrările cu alte entități de stat vor fi realizate prin infrastructura MConnect.

În plus, se va asigura conformitatea cu cerințele pentru Top 10 OWASP.

Performanța sistemului final va depăși următorii parametri:

- apeluri pe zi: ~ 50000;
- apeluri pe ora (în mediu): ~ 2000;
- durată apel (în secunde): ~3.

Se va menționa că performanța sistemului va fi limitată de infrastructura furnizată de Beneficiar.

6. Executarea Cerințelor Funcționale

Această secțiune descrie abordarea noastră pentru implementarea cerințelor funcționale.

a. Dezvoltarea aplicației web (Front Office & Back Office):

- Se va utiliza ReactJS pentru dezvoltarea aplicației web.
- Se va utiliza Fetch API pentru realizarea cererilor HTTP către backend.
- Se vor implementa interfețele utilizator pentru diferite funcționalități ale aplicației folosind componente React (Formular de Înregistrare a Aeronavelor, Formular de Înregistrare a Termenelor, etc).
- Se vor implementa interfețele utilizator pentru căutarea informațiilor pe baza diferitelor filtre. Rezultatele căutării vor fi afișate într-un tabel paginat care va permite navigarea prin date și adăugarea de criterii/filtre suplimentare de căutare.
- Se vor implementa funcționalitățile pentru descărcarea datelor în formate PDF, CSV și XLSX în pagina de rezultate.

b. Dezvoltarea Backendului:

- Se va utiliza Spring Boot pentru backend, care oferă un cadru robust pentru construirea aplicațiilor Java.
- Se va configura o bază de date PostgreSQL pentru stocarea datelor.

- Se vor implementa API-uri RESTful folosind Spring MVC pentru comunicarea între frontend și backend.
- Se va utiliza JPA (Java Persistence API) cu Hibernate pentru operațiuni cu baza de date, pentru a simplifica accesul la date.
- Se va utiliza Elasticsearch pentru îmbunătățirea funcționalității de căutare și pentru facilitarea recuperării eficiente a datelor.
- Se va utiliza biblioteca JasperReports pentru generarea rapoartelor și documentelor. Se vor crea șabloane configurabile pentru documentele și rapoartele care trebuie generate.
- Documentele și rapoartele generate vor fi stocate într-o locație sigură în cadrul sistemului IS RA. Se vor oferi mecanisme pentru recuperarea și vizualizarea acestor documente.
- Toate activitățile utilizatorilor vor fi înregistrate folosind serviciul de Audit.

c. Internaționalizare (i18n):

- Se va utiliza biblioteca React-Intl pentru internaționalizare în frontend.
- Se va implementa funcționalitatea de schimbare a limbii pentru a suporta limbile română, engleză și rusă (sau alte limbi, dacă va fi necesar).

d. Autentificare și autorizare:

- Pentru autentificarea utilizatorilor se vor folosi tokenuri de securitate.
- Atunci când un utilizator se conectează, va fi generat și emis de serviciul de autentificare un token opac (un token care nu conține informații despre utilizator). Numai cu acest token opac utilizatorul poate accesa serviciile IS RA.
- Atunci când se inițiază o solicitare către serviciile de backend, API Gateway va identifica detaliile utilizatorului pe baza tokenului opac. API Gateway va folosi aceste detalii pentru a valida solicitarea și a determina permisiunile și nivelurile de acces corespunzătoare.

e. Gestionarea utilizatorilor, rolurilor și drepturilor de acces:

- Se va proiecta un meniu lateral cu linkuri către diferite secțiuni ale panoului de administrare, inclusiv 'Gestionare Utilizatori', 'Gestionare

Roluri'. Numai 'Super Admin' poate accesa secțiunile panoului de administrare.

- Se vor defini tabele pentru stocarea informațiilor despre utilizatori, roluri și drepturile de acces asociate acestora.
- Se vor implementa API-uri RESTful pentru gestionarea profilurilor utilizatorilor, rolurilor și drepturilor de acces.
- Se va implementa logica pentru atribuirea rolurilor utilizatorilor și definirea drepturilor de acces pe baza apartenenței la roluri.
- Se vor proiecta interfețe utilizator pentru gestionarea profilurilor utilizatorilor, rolurilor și drepturilor de acces.
- Se vor include formulare pentru adăugarea/editarea utilizatorilor, atribuirea rolurilor și gestionarea drepturilor de acces.
- Se vor defini drepturi de acces pentru diferite părți ale IS RA pe baza rolurilor.

f. Configurarea Sistemului:

- Se vor defini tabele pentru stocarea parametrilor de configurare.
- Parametrii de configurare vor fi încărcăți dinamic, fără a necesita repornirea sistemului.
- Se va dezvolta o interfață de utilizare (UI) pentru administratori, pentru a vizualiza și modifica parametrii de configurare.
- Se vor implementa mecanisme de control al accesului pentru a restricționa accesul la UI-ul de configurare în funcție de rolurile utilizatorilor.
- Se va implementa jurnalizarea auditurilor pentru a urmări modificările parametrilor de configurare.
- Se vor înregistra utilizatorul care a efectuat modificarea, timpul modificării și valorile vechi și noi ale parametrului de configurare.

g. Monitorizare Operațională, Diagnostiche

- Vor fi utilizate instrumente de monitorizare, cum ar fi Prometheus și Grafana, pentru a colecta și vizualiza parametrii de funcționare, precum metricile de performanță ale sistemului, utilizarea resurselor și jurnalele aplicației.
- Vor fi implementate metrici personalizate pentru a monitoriza aspecte specifice ale operațiunilor IS RA.

- Se va implementa monitorizarea și înregistrarea evenimentelor critice și a erorilor pentru a identifica și diagnostica rapid problemele tehnice.
- Vor fi folosite instrumente de agregare a jurnalelelor, cum ar fi ELK Stack (Elasticsearch, Logstash, Kibana), pentru jurnalizare și analiză centralizată.
- Vor fi dezvoltate scripturi de diagnosticare pentru a automatiza sarcinile comune de depanare.
- Se va utiliza o platformă de raportare, cum ar fi Jasper Reports, pentru a crea rapoarte statistice predefinite privind evenimentele de operare ale IS RA.
- Vor fi implementate mecanisme de control al accesului pentru a restricționa accesul la instrumentele de monitorizare și diagnosticare în funcție de rolurile și responsabilitățile utilizatorilor.

h. Interoperabilitate și schimb de date:

- Se va utiliza platforma de interoperabilitate guvernamentală MConnect pentru a consuma date din sistemele IT externe.
- Se va crea un serviciu de ieșire pentru a expune datele din IS RA către sistemele externe. Serviciul de ieșire va fi integrat cu MConnect.
- IS RA va fi integrat cu serviciile guvernamentale comune (MPass, MSign, MNotify, MLog etc.) utilizând API-urile acestora.
- Se va folosi MSign pentru semnarea digitală a documentelor și formularelor din IS RA.
- Se va utiliza MNotify pentru trimiterea notificărilor către utilizatori și administratori.
- Se va implementa logarea și monitorizarea utilizând MLog pentru urmărirea evenimentelor din sistem și scopuri de audit.
- Se vor implementa criptarea și protocoale securizate pentru schimbul de date.
- Vor fi efectuate audite de securitate regulate și evaluări ale vulnerabilităților.

i. Înregistrarea evenimentelor:

- Se va utiliza un framework (Log4j pentru Java) pentru a înregistra evenimentele generate de aplicație. Înregistrarea se va face asincron pentru a evita impactul asupra performanței principalei aplicații.

- Pentru înregistrarea acțiunilor utilizatorilor în IS RA se va folosi biblioteca SpringAOP. Aceste jurnale vor fi stocate în Elasticsearch pentru a îmbunătăți funcționalitatea de căutare.
- Evenimentele de afaceri critice și sensibile vor fi, de asemenea, înregistrate în serviciul guvernamental MLog.
- Se va crea un API RESTful care va curăța și arhiva jurnalele mai vechi decât o anumită perioadă configurabilă.
- Pentru monitorizarea utilizării spațiului pe disc și degradarea performanței se va folosi instrumentul de monitorizare - Zabbix.

7. Instalare

Sistemul de producție va fi instalat pe un cluster Kubernetes. În plus, va fi configurat un cluster de dezvoltare/UAT în MCloud sau pe cloud-ul S&T pentru scopuri de testare/demo.

Toate instalările vor fi realizate prin intermediul pipeline-urilor automate folosind o abordare de infrastructură ca cod (infrastructure-as-code).

Pentru acest proiect, vom avea nevoie de servere cu următoarele cerințe tehnice:

- Kubernetes cluster: 3 servere (un master și două workere). Configurație Master: CPU 2 core, RAM 4GB, HDD 100GB. Configurație Worker: CPU 4 core, RAM 6GB, HDD 100GB.
- NFS server (pentru stocarea fișierelor): CPU 2 core, RAM 4GB, HDD 500GB.
- Server bază de date: CPU 4 core, RAM 16GB, HDD 500GB.

8. Cerințe pentru instruirea utilizatorilor sistemului

În cadrul implementării sistemului, vor fi dezvoltate și furnizate materiale de instruire. Instruirea pentru utilizatorii cheie va fi efectuată pe baza materialelor dezvoltate. Procesul de instruire va acoperi subiectele necesare și suficiente pentru ca utilizatorii și administratorii să își îndeplinească funcțiile.

Se va dezvolta și se va agreea un Plan de Instruire cu Beneficiarul. Planul de Instruire va include toate acțiunile necesare, programele, scenariile de instruire și alte cerințe.

9. Backup și recuperarea în caz de dezastru

Se așteaptă ca operatorii și administratorii sistemului să fie instruiți specific pentru a asigura întreținerea de bază a sistemului. Ca parte a transferului de cunoștințe, toate informațiile necesare pentru recuperarea rapidă din situații de eșec vor fi furnizate. Va fi oferit un scurt ghid de instrucțiuni și proceduri de operare.

Procedurile de recuperare vor include atât software-ul aplicației, cât și sistemele de gestionare a bazelor de date.

10. Testare Automatizată

Toate cerințele pentru sistemele de testare sunt asigurate de procesul nostru de dezvoltare. Detalii complete sunt prezentate în capitolul 3.5 „Tehnici și Abordări pentru Asigurarea Calității”.

11. Procedura de acceptare a sistemului

Echipa noastră va prelua conducerea organizării acceptării sistemului. Va fi creat un comitet de acceptare a sistemului, care va include reprezentanți ai Clientului și ai Contractorului.

Software-ul va fi furnizat împreună cu codul sursă, iar Clientul/Beneficiarul va avea dreptul de a modifica și de a angaja alți furnizori pentru a modifica sistemul. Cu toate acestea, este necesar să se acorde o mare atenție în perioada de garanție, și se așteaptă ca, dacă alți furnizori modifică software-ul, aceștia să aibă calificările și abilitățile necesare.

Testele de acceptare vor fi efectuate pe baza unui protocol care va fi convenit cu Beneficiarul.

Următoarea lista de documente va fi furnizată odată cu livrarea sistemului:

- Descrierea funcțională a sistemului;
- Manual de utilizare și funcțiile acestuia;
- Manual pentru administratori de sistem;
- Diagrama fluxului de date + structura bazei de date + diagrama ER;
- Documentația de design a software-ului (SDD);
- Documentația arhitecturii software-ului;
- Instrucțiuni pentru implementare, actualizare și recuperare;
- Act de transfer și acceptare;
- Program și metode de testare.

12. Concluzie

Echipa noastră este pregătită să preia această sarcină și garantează livrarea unui produs de înaltă calitate care va ajuta Beneficiarul să-și automatizeze și să-și modernizeze procesele.

Pentru orice întrebări sau clarificări privind detaliile abordării noastre tehnice pentru acest proiect, vă rugăm să nu ezitați să ne contactați.

Semnat: _____

Andrei Cojocari

Chief Commercial Officer

S&T MOLD SRL

Republic of Moldova, Chisinau city, Calea Ieșilor 8 street, MD-2069