smartdata

Specificarea tehnică deplină propusă de către ofertant

Numărul procedurii de achiziție nr. MTender ID ocds-b3wdp1-MD-1770813303659 din “11” 02. 2026

Obiectul achiziției: Servicii de dezvoltarea și implementare a paginii web oficiale a Î.S. „Aeroportul Internațional Chișinău”

1. DECLARAȚIE DE CONFORMITATE ȘI ASUMARE

Operatorul economic declară că soluția tehnică propusă este proiectată pentru a respecta cerințele prevăzute în Caietul de Sarcini aferent procedurii de achiziție privind dezvoltarea portalului web www.airport.md.

Prin prezenta propunere tehnică, ne angajăm să:

- Implementăm integral toate cerințele funcționale (FR) și non-funcționale prevăzute în documentația de atribuire;
- Respectăm termenul maxim de implementare de 12 luni calendaristice;
- Asigurăm perioada de garanție de minimum 12 luni;
- Asigurăm, în condițiile unei infrastructuri dimensionate corespunzător conform recomandărilor tehnice furnizate în cadrul documentației de implementare, respectarea cerințelor de performanță (≤ 3 secunde), scalabilitate (100.000 utilizatori simultan FE) și disponibilitate (minim 99,5%);
- Asigurăm integrarea cu sistemele externe prevăzute (MPay, card bancar, ERP, API zboruri etc.);
- Livrăm toate livrabilele prevăzute în documentația de atribuire.

Confirmăm că soluția propusă respectă integral cerințele din documentația de atribuire și este proiectată pentru operare stabilă într-un mediu operațional 24/7.

2. ÎNȚELEGEREA PROIECTULUI ȘI OBIECTIVELE SOLUȚIEI

2.1 Contextul proiectului

Proiectul presupune dezvoltarea unui portal web complex pentru Aeroportul Internațional Chișinău, care îndeplinește simultan mai multe roluri:

- Platformă publică de informare pentru pasageri;
- Sistem de publicare și administrare conținut;
- Platformă operațională internă (CRM / registre);
- Sistem tranzacțional pentru servicii cu achitare online;
- Interfață de integrare cu sisteme externe operaționale și financiare.

Portalul trebuie să funcționeze într-un context de trafic ridicat, expunere publică semnificativă și risc crescut de atacuri informatice.

2.2 Complexitatea proiectului

Proiectul este caracterizat prin:

- Număr mare de cerințe funcționale (FR);
- Integrare cu sisteme externe critice (MPay, ERP, sistem zboruri);
- Cerințe stricte de performanță (≤ 3 sec);
- Cerințe ridicate de scalabilitate (100.000 utilizatori simultan);
- Necesitatea conformității WCAG;
- Necesitatea auditului și trasabilității operaționale;
- Cerințe stricte de securitate și protecție date.

Această complexitate impune o arhitectură robustă, o planificare etapizată și un control strict al calității.

2.3 Obiectivele soluției propuse

Soluția propusă urmărește următoarele obiective principale:

1. Performanță și scalabilitate Asigurarea timpilor de răspuns conform cerințelor și suportarea traficului ridicat.
2. Securitate și protecția datelor Implementarea unui sistem securizat, auditabil și conform bunelor practici internaționale.
3. Modularitate și extensibilitate Permisiunea extinderii ulterioare fără refactorizare majoră.

4. Operabilitate și eficiență Asigurarea unui sistem intuitiv pentru utilizatori și eficient pentru personalul operațional.
5. Stabilitate pe termen lung Livrarea unei soluții sustenabile, documentate și ușor de întreținut.

---

3. ARHITECTURA SOLUȚIEI

3.1 Viziunea arhitecturală

Soluția este construită pe o arhitectură modulară, containerizată și orientată pe performanță.

Structura generală include:

- Componenta Publică (Portal) – Laravel Blade (SSR)
- Componenta Admin/CRM – Next.js + TypeScript
- Backend aplicație – Laravel (monolit modular)
- Bază de date – PostgreSQL
- Cache – Redis
- Containerizare – Docker
- Distribuție conținut și protecție trafic – CDN sau reverse proxy configurat la nivel de infrastructură (ex.: Nginx / Varnish / echivalent)

3.2 Arhitectură logică

3.2.1 Componenta publică

- Generare server-side pentru SEO optim;
- Cache multi-layer;
- Integrare API zboruri;
- Formular feedback;
- Conținut multilingv.

3.2.2 Componenta administrativă (CRM)

- Aplicație SPA Next.js;

- Comunicare prin API interne Laravel;
- Gestionare registre;
- Workflow operațional;
- Export date;
- Jurnalizare.

3.2.3 Backend Laravel

- Implementare logică business;
- Implementare RBAC;
- Implementare integrări externe;
- Validare input;
- Logging operațional și audit al activităților critice (conform mecanismelor descrise în capitolul 6 – Securitate).
- Layer API pentru comunicarea între frontend, CRM și sistemele externe;
- Mecanisme de procesare asincronă (queue workers) pentru operațiuni care nu necesită execuție imediată, cum ar fi trimiterea notificărilor, actualizarea indexurilor de căutare sau sincronizarea cu sisteme externe;
- Mecanisme de caching pentru reducerea încărcării bazei de date și optimizarea timpilor de răspuns;
- Sistem de logging și audit operațional pentru monitorizarea activităților critice și diagnosticarea incidentelor.

3.3 Justificarea tehnologiilor

Laravel (PHP)

- Conform stivei acceptate;
- Stabilitate și maturitate;
- Ecosistem robust;
- Securitate integrată.

PostgreSQL

- Performanță ridicată;
- Stabilitate;
- Suport tranzacții complexe.

Next.js + TypeScript

- Interfață modernă pentru CRM;
- Scalabilitate UI;
- Tipizare strictă pentru reducerea erorilor.

Redis

- Reducere încărcare DB;
- Cache rapid pentru date dinamice.

Docker

- Portabilitate;
- Reproductibilitate medii;
- Scalare facilă.

3.4 Principii arhitecturale

- Security by Design;
- Performance First;
- API modular;
- Code review obligatoriu;
- Separare medii (Dev/Test/Pre-prod/Prod);
- Observabilitate și monitorizare operațională;

3.5 Scalabilitate și extensibilitate

Arhitectura permite:

- Scalare orizontală a backend-ului;
- Extindere module noi;
- Integrare API suplimentare;
- Actualizare independentă componente.

Arhitectura propusă respectă principiile moderne de proiectare a aplicațiilor web scalabile și securizate și permite evoluția ulterioară a platformei fără refactorizări majore și fără impact asupra continuității serviciilor operaționale.

4. IMPLEMENTAREA DETALIATĂ A CERINȚELOR FUNCȚIONALE

4.1 Principii generale de implementare

Toate cerințele funcționale (FR) prevăzute în Caietul de Sarcini vor fi implementate utilizând următoarele principii:

- Modularitate – fiecare grup de funcționalități este implementat ca modul distinct;
- Reutilizare – utilizarea șabloanelor și componentelor reutilizabile pentru eficiență;
- Separare clară între interfața publică și modulele interne;
- Validare strictă server-side pentru toate operațiunile critice;
- Trasabilitate – fiecare FR este mapat la taskuri și livrabile interne;
- Auditabilitate – toate acțiunile relevante sunt jurnalizate.

4.2 PORTAL PUBLIC

Această secțiune acoperă cerințele FR privind interfața publică a site-ului.

4.2.1 Structura și navigarea portalului

Implementare:

- Meniu principal dinamic gestionat din CMS;
- Suport pentru structură ierarhică (niveluri multiple);
- Breadcrumbs automate;
- Footer configurabil;
- Responsive design pentru desktop/tablet/mobil;

- Switch limbă (RO/RU/EN);
- Social media integration.

Tehnic:

- Blade SSR pentru performanță și SEO;
- Gestionare meniu în baza de date;
- Cache Redis pentru structura meniului.

4.2.2 Management conținut informativ

Toate paginile informaționale (ex. documente de călătorie, servicii aeroport, transport etc.) vor fi implementate pe baza unor șabloane configurabile.

Funcționalități:

- Editor WYSIWYG avansat;
- Inserare imagini, tabele, fișiere;
- Versionare conținut;
- Publicare programată;
- Preview înainte de publicare;
- Gestionare bannere asociate paginilor.

Implementare:

- CMS Laravel modular;
- Stocare versiuni în DB;
- Media management dedicat.

4.2.3 Funcționalitate căutare site

Obiectiv

Implementarea unei funcționalități de căutare rapide și relevante pentru paginile publice, știri, companii aeriene și alte resurse textuale ale portalului, în limbile RO / RU / EN.

Căutarea va permite identificarea rapidă a informațiilor de interes pentru utilizatori și va suporta corectarea automată a unor erori tipografice, filtrarea rezultatelor și evidențierea termenilor căutați.

Soluție propusă

Funcționalitatea de căutare va fi implementată utilizând un motor de căutare open-source optimizat pentru performanță și relevanță, precum Meilisearch, integrat cu backend-ul aplicației Laravel.

Motorul de căutare va indexa conținutul public al portalului (pagini informative, articole, companii aeriene și alte resurse textuale), permițând:

- căutare rapidă în întregul portal;
- sugestii și toleranță la greșeli tipografice;
- filtrarea rezultatelor după tipul de conținut;
- evidențierea termenilor căutați în rezultate.

Integrare în arhitectură

Căutarea este integrată cu sistemul CMS al portalului, astfel încât conținutul nou publicat sau modificat să fie actualizat automat în indexul de căutare.

Frontend-ul portalului transmite interogările de căutare către backend, care interacționează cu motorul de căutare și returnează rezultatele optimizate pentru afișare.

Datele dinamice precum informațiile despre zboruri vor continua să fie furnizate direct prin API-urile dedicate și mecanismele de caching ale aplicației.

Performanță și disponibilitate

Motorul de căutare este optimizat pentru răspuns rapid și pentru gestionarea unui volum mare de interogări. Performanța funcționalității de căutare va fi verificată în cadrul testelor de performanță descrise în Capitolul 5.

În situația în care serviciul de căutare este temporar indisponibil, aplicația va utiliza mecanisme alternative de afișare a rezultatelor pentru a menține funcționalitatea portalului.

4.2.4 Afișare zboruri (Arrivals/Departures)

Funcționalități:

- Afișare zboruri pentru intervale definite;
- Filtrare după dată/număr zbor;
- Căutare rapidă;
- Actualizare periodică;
- Fallback în caz de indisponibilitate API.

Implementare:

- Conector API către sistemul de zboruri;
- Cache Redis pentru rezultate;
- Tabel dinamic performant;
- Logging erori integrare.

4.2.5 Formular feedback și solicitări

Funcționalități:

- Formular public;
- Validări obligatorii;
- reCAPTCHA v3;
- Confirmare trimitere;
- Înregistrare automată în registru intern;
- Status solicitare.

Implementare:

- Validare server-side;
- Generare ID unic solicitare;
- Notificare email internă;
- Jurnalizare.

4.2.6 Cookie Consent & Accesibilitate (WCAG AA)

Soluția va fi dezvoltată conform WCAG 2.1 / 2.2 la nivel AA. Vom livra un widget de accesibilitate și un banner cookie conform cerințelor, precum și toate ajustările necesare pentru respectarea succes-criteriilor WCAG AA.

Accesibilitatea va fi integrată de la design la implementare: elemente semantice HTML, navigare completă prin tastatură, indicatori de focus vizibili, aria attributes acolo unde e necesar, contrast adaptat, texte alternative pentru media, formulare accesibile și testare manuală cu cititoare de ecran.

Toate componentele vor trece printr-un set obligatoriu de verificări (automate și manuale) înainte de acceptanța finală. Rezultatele testelor și planul de remediere vor fi livrate ca parte din documentația finală.

Cerinţe tehnice şi de implementare (WCAG AA)

Implementarea interfeței va respecta principiile WCAG 2.1 / 2.2 nivel AA, incluzând:

- utilizarea elementelor HTML semantice și a atributelor ARIA acolo unde este necesar;
- asigurarea contrastului adecvat pentru text și elemente UI;
- navigare completă prin tastatură și focus management corect;
- formulare accesibile și mesaje de eroare lizibile;
- suport pentru tehnologii asistive (screen readers);
- adaptare corectă la redimensionare și zoom.

Validarea conformității WCAG este realizată prin activitățile de design, implementare și testare descrise în capitolele 7 și 10.

4.3 SISTEM DE ADMINISTRARE (CMS)

4.3.1 Gestionare utilizatori și roluri

Implementare RBAC:

Roluri:

- Super Admin
- Admin CMS
- Operator CRM
- Operator financiar

- Utilizator autentificat

Funcționalități:

- Creare/ștergere utilizatori;
- Resetare parolă;
- Blocare cont;
- Limitare IP;
- Permisiuni granulare pe module.

4.3.2 Audit și jurnalizare

- Log autentificări;
- Log modificări conținut;
- Log tranzacții financiare;
- Log schimbare status solicitări;
- Export jurnal activități.

4.4 CRM / MODULE INTERNE

Implementat în Next.js + TypeScript.

4.4.1 Registre interne

Funcționalități:

- Listare solicitări;
- Filtrare avansată;
- Sortare multi-coloană;
- Configurare coloană vizibilitate;
- Paginare;
- Export CSV/XLSX/PDF.

4.4.2 Workflow operațional

- Statusuri configurabile;
- Atribuire responsabil;
- Notificări automate;
- Istoric modificări;
- Comentarii interne.

4.4.3 Dashboard operațional

- Statistici solicitări;
- Volum pe perioade;
- Indicatori operaționali;
- Vizualizări grafice.

4.5 MODUL PLĂȚI ONLINE

4.5.1 Integrare MPay

- Inițiere tranzacție;
- Redirect securizat;
- Confirmare prin webhook;
- Validare semnătură;
- Actualizare status solicitare.

4.5.2 Integrare card bancar

- Procesare plăți securizată;
- Tokenizare;
- Confirmare plată;

- Gestionare erori.

4.5.3 Export ERP

- Export automat tranzacții;
- Format configurabil (CSV/API);
- Programare periodică;
- Log export.

4.6 INTEGRĂRI EXTERNE

4.6.1 API Zboruri

- Conector REST;
- Autentificare API (dacă e necesar);
- Rate limiting;
- Cache;
- Fallback.

4.6.2 Notificări (SMTP / SMS)

- Configurare SMTP;
- Integrare SMS Gateway;
- Șabloane email configurabile;
- Retry logic.

4.6.3 Integrare MPass

Integrarea MPass se va realiza prin SAML 2.0 (SP-initiated SSO). Soluția va valida obligatoriu Assertion semnată (XML Signature), Issuer, Audience/Recipient şi condițiile NotBefore/NotOnOrAfter, va extrage NameID (NameIdentifier = IDNP) şi atributele esențiale (EmailAddress, FirstName, LastName, AdministeredLegalEntity) şi va efectua identificarea utilizatorului preferențial după IDNP (email ca fallback). La primul acces se poate crea un cont parţial urmat de workflow de completare; în conformitate cu politica de securitate, blocarea unui cont local nu va fi ocolită prin SSO. Fluxul va suporta RP-initiated logout şi, în situaţia în care MPass oferă, IdP-initiated SLO; toate endpoint-urile ACS/SLO vor funcţiona exclusiv prin HTTPS, cu protecţii contra replay şi validări CSRF la callback. Livrabilele aferente vor include conectorul SAML configurat, documentaţia mapping atribute şi raport UAT pentru scenarii critice.

4.6.4 Integrare plăți — MPay & card

Pentru procesarea tranzacțiilor online propunem integrarea cu MPay (conector MPay) și suport backend pentru plăți cu cardul (gateway bancar), conform cerințelor AIC. Soluţia include implementarea fluxului de plată securizat, manipularea webhook-urilor de confirmare (idempotent, verificare semnătură), testare în sandbox și reconciliere automată a tranzacțiilor cu registrul intern şi export ERP. Toate comunicațiile sensibile vor fi realizate exclusiv prin canale criptate (TLS), iar datele de card vor fi procesate/primitulate conform cerințelor PCI DSS (în funcție de opţiunea de integrare — tokenizare client sau redirect către gateway) şi politicilor băncii desemnate de AIC. Vom implementa măsuri suplimentare de securitate: validare webhook, control duplicate callbacks, logging auditat şi proceduri de rollback/refund. Livrabile: conector MPay/Payment Gateway configurat, handler webhook securizat, plan teste sandbox & UAT pentru scenarii directe şi de regresie, scripturi de reconciliere şi documentaţie operaţională privind modul de integrare cu banca aleasă de AIC.

4.6.5 Portal API / Standardizare

- Endpoint-uri REST.
- Autentificare utilizând mecanisme standard (OAuth2, JWT sau API Key), în funcție de cerințele fiecărei integrări.
- Documentație OpenAPI.

4.7 MULTILINGVISM

- RO/RU/EN;
- Gestionare traduceri în CMS;
- Fallback limbă implicită;
- URL separat per limbă;
- SEO multi-lingv.

4.8 VALIDARE ȘI ACCEPTANȚĂ

Pentru fiecare modul:

- Testare funcțională;
- Testare integrare;
- Testare regresie;
- Validare împreună cu beneficiarul;
- Confirmare prin UAT formal.

4.9 TRASABILITATE CERINȚE

Pentru fiecare FR din Caietul de Sarcini:

- FR ID → Modul implementare → Componentă → Livrabil → Test asociat

Această matrice va fi anexată separat în document.

---

5. PERFORMANȚĂ, SCALABILITATE ȘI DISPONIBILITATE

5.1 Obiective de performanță

Soluția propusă este proiectată astfel încât, în condițiile unei infrastructuri dimensionate corespunzător conform recomandărilor tehnice furnizate în cadrul documentației de implementare, să poată respecta următoarele obiective de performanță:

- Timp de reacție pentru operațiunile publice ≤ 3 secunde (cu excepțiile agreate la etapa de proiectare);
- Arhitectura aplicației permite suportarea unui volum de până la 100.000 vizitatori simultan pe componenta publică, în condițiile unei infrastructuri dimensionate corespunzător conform recomandărilor tehnice din documentația de implementare;
- Capacitate de suport pentru minim 1.000 utilizatori simultan în modulele interne;
- Funcționare 24/7;
- Disponibilitate minimă anuală de 99,5%;
- Rezistență la perioade de trafic intens.

Aceste obiective sunt abordate printr-o combinație de arhitectură optimizată, caching multi-layer, mecanisme de distribuție a conținutului configurate la nivelul infrastructurii beneficiarului (ex. CDN sau reverse proxy) și testare formală de performanță.

Performanța finală depinde de dimensionarea infrastructurii hardware și de configurarea mediului de producție.

Configurația minimă recomandată a infrastructurii (servere aplicație, bază de date, resurse CPU/RAM și mecanisme de distribuție a traficului) va fi definită în documentația tehnică livrată în Faza 1 – Analiză detaliată și arhitectură, pe baza volumului estimat de trafic și a cerințelor operaționale ale beneficiarului.

5.2 Arhitectură orientată pe performanță

5.2.1 Server-Side Rendering pentru portal public

Componenta publică este implementată utilizând Laravel Blade (SSR), ceea ce permite:

- Generarea paginilor pe server cu conținut complet;
- SEO optim;
- Timp redus de încărcare inițială;
- Reducerea dependenței de execuție JavaScript la client.

5.2.2 Caching pe mai multe niveluri

Pentru asigurarea performanței în condiții de trafic ridicat, se utilizează un model de caching pe mai multe niveluri:

Nivel 1 – CDN / Reverse Proxy

- Cache pentru resurse statice (imagini, CSS, JS);
- Cache pentru pagini publice unde este permis;
- Protecție împotriva atacurilor volumetrice prin mecanisme de filtrare trafic;
- Rate limiting configurabil;
- Web Application Firewall configurat la nivel de infrastructură sau reverse proxy.

Nivel 2 – Cache aplicație (Redis)

- Cache pentru rezultate API externe (zboruri);
- Cache pentru structuri meniu;
- Cache pentru pagini generate frecvent;
- Stocare temporară sesiuni utilizator în Redis pentru componentele care utilizează mecanisme de autentificare bazate pe sesiuni.

Nivel 3 – Optimizare bază de date

- Indexare strategică a coloanelor frecvent interogate;
- Optimizare query-uri;
- Evitarea N+1 queries;
- Monitorizare performanță DB.

5.3 Scalabilitate

Soluția este proiectată pentru scalare orizontală în infrastructura beneficiarului, prin replicarea serviciilor containerizate și distribuirea traficului prin mecanisme standard de load balancing.

5.3.1 Containerizare Docker

- Aplicația este livrată în containere Docker;
- Separare componente (frontend public, backend, DB);
- Replicarea instanțelor backend pentru scalare;
- Distribuirea traficului prin mecanisme standard de load balancing.

5.3.2 Scalare orizontală

În cazul creșterii traficului, soluția permite:

- Replicarea instanțelor aplicației backend;

- Distribuirea traficului prin load balancer;
- Ajustarea resurselor (CPU/RAM) în funcție de necesități.

5.4 Gestionarea traficului intens

Pentru situații precum:

- anulări masive de zboruri,
- condiții meteo extreme,
- campanii media,

se aplică următoarele măsuri:

- Cache agresiv pentru pagini publice;
- Cache temporar pentru date zboruri;
- Rate limiting pentru endpoint-uri critice;
- Fallback informativ dacă API extern este indisponibil;
- Monitorizare în timp real a metricilor.

5.5 Testare de performanță (Load Testing)

Testarea de performanță va fi realizată utilizând instrumentul k6.

5.5.1 Tipuri de teste

1. Baseline Load Test
    - Simulare trafic constant;
    - Verificare stabilitate p95 sub 3 secunde.
2. Spike Test
    - Creștere bruscă de trafic;
    - Evaluare comportament sistem.
3. Soak Test
    - Trafic constant pe perioadă extinsă;
    - Detectare memory leaks sau degradare.
4. Stress Test
    - Încărcare progresivă până la degradare controlată;
    - Identificare praguri de capacitate.

5.5.2 Indicatori monitorizați

- p95 / p99 response time;
- Error rate;
- Requests per second (RPS);
- CPU / RAM utilizate;
- Număr conexiuni DB;
- Timp răspuns API extern.

5.6 Disponibilitate și continuitate operațională

Pentru asigurarea unei disponibilități minime de 99,5%, sunt implementate:

- Monitorizare aplicație și infrastructură;
- Alertare automată la depășirea pragurilor;
- Proceduri de restart controlat;
- Proceduri rollback;
- Proceduri backup & restore testate;
- Documentație operare pentru STIT.

Sistemul va include mecanisme de observabilitate operațională, care permit monitorizarea în timp real a indicatorilor tehnici ai aplicației (performanță, erori, utilizare resurse) și diagnosticarea rapidă a incidentelor operaționale.

5.7 Disaster Recovery și Backup

5.7.1 Politica de backup

- Backup periodic bază de date;
- Backup fișiere media;
- Stocare securizată;
- Retenție configurabilă.

5.7.2 Testare restaurare

- Teste formale de restaurare în mediu controlat;
- Validare integritate date;
- Documentare procedură.

5.7.3 RTO / RPO — obiective și proceduri

Pentru asigurarea continuității serviciului și a integrității datelor se propun următoarele obiective și proceduri pentru mediul de producție:

- RPO (Recovery Point Objective)
    - Date critice (transacții de plată, jurnal tranzacții, status zboruri): ≤ 1 oră (backup incremental orar).
    - Date non-critice (conținut public, fișiere media): ≤ 4 ore.
- RTO (Recovery Time Objective)
    - Servicii critice: ≤ 4 ore pentru restaurare completă a serviciilor esențiale (API zboruri, platformă plăți, pagini publice critice).
    - Servicii non-critice: ≤ 24 ore pentru restaurare completă.
- Politici backup & retenție
    - Backup incremental la fiecare oră pentru baze de date critice; backup full zilnic pentru DB.
    - Retenție: backupuri orare păstrate 48 ore; backupuri zilnice păstrate 30 zile; backupuri săptămânale păstrate 90 zile; backupuri lunare păstrate 365 zile.
    - Snapshot-uri la nivel de obiecte pentru media; exporturi periodice ale indexurilor (Meilisearch).
- Proceduri de testare
    - Teste de restaurare trimestriale în mediu pre-prod: restaurare DB critice și verificare integritate + smoke tests UI/API.
    - Teste de restaurare după upgrade-uri majore sau modificări de schemă.
- Măsuri operaționale
    - Scripturi automate de backup și playbook de restaurare documentat.
    - Document DR (runbook) cu responsabilități, pași, contacte și checklist.
    - Raport DR: rezultat, timp efectiv de restaurare, lecții învățate și plan de remediere.

5.8 Capacity Planning

În faza de analiză, se vor defini:

- Specificații minime hardware pentru producție;
- Număr recomandat de instanțe aplicație;
- Estimare resurse DB;

- Strategii de creștere graduală.

Aceste recomandări vor fi incluse în documentația tehnică livrată beneficiarului.

---

6. SECURITATE ȘI PROTECȚIA DATELOR

6.1 Principii generale de securitate

Soluția propusă este proiectată conform principiului Security by Design, ceea ce înseamnă că mecanismele de securitate sunt integrate încă din faza de arhitectură și nu adăugate ulterior.

Principii aplicate:

- Respectarea recomandărilor OWASP Top 10;
- Minimizarea suprafeței de atac;
- Separarea responsabilităților (frontend / backend / infrastructură);
- Validare strictă a datelor la nivel server;
- Principiul „least privilege” pentru acces și permisiuni;
- Auditabilitate completă a operațiunilor critice.

Soluția va fi implementată în deplină conformitate cu cadrul normativ național, incluzând Legea nr. 48/2023 privind securitatea cibernetică și HG nr. 728/2023 privind cerințele minime de securitate cibernetică.

În acest sens, mecanismele de autentificare și gestionare a credențialelor vor fi configurate în conformitate cu politicile de securitate aplicabile beneficiarului (ex.: complexitate parole, expirare periodică, controlul tentativelor de autentificare), iar sistemul va include mecanisme extinse de jurnalizare (logging) care permit auditarea și trasabilitatea operațiunilor critice, în linie cu bunele practici de securitate și cu standardele internaționale relevante (ex. ISO/IEC 27001).

6.1.1 Politică de gestionare a secretelor

Pentru protejarea credențialelor și a altor informații sensibile (chei API, parole, token-uri, certificate), soluția propusă utilizează un mecanism dedicat de gestionare a secretelor (secret vault / secret manager) sau un mecanism securizat echivalent furnizat de infrastructura beneficiarului.

Principiile aplicate sunt următoarele:

Separarea secretelor de codul aplicației

- niciun secret nu este stocat în codul sursă sau în repository;
- credențialele sunt injectate securizat la rularea aplicației.

Controlul accesului

- accesul la secrete este gestionat pe bază de roluri și politici de acces;
- aplicațiile și utilizatorii primesc doar permisiunile strict necesare.

Utilizarea securizată în aplicație

- frontend-ul nu conține secrete sensibile;
- operațiunile critice care utilizează chei sau token-uri sunt executate exclusiv la nivelul backend-ului.

Audit și monitorizare

- accesul la secrete și operațiunile asociate sunt jurnalizate și pot fi auditate;
- mecanismele de rotație și revocare a secretelor sunt gestionate conform politicilor operaționale ale infrastructurii.

Acest model reduce riscul expunerii credențialelor și permite administrarea centralizată și controlată a tuturor secretelor utilizate de aplicație.

6.2 Securitatea comunicațiilor

- Toate comunicațiile vor utiliza protocol HTTPS (TLS 1.2+);
- Certificat SSL valid pentru domeniul public;
- Interzicerea accesului prin HTTP necriptat;
- HSTS activat pentru prevenirea downgrade attack;
- Configurare secure headers (Content-Security-Policy, X-Frame-Options, X-Content-Type-Options etc.).

6.3 Autentificare și autorizare

6.3.1 Mecanism de autentificare

- Autentificare bazată pe sesiuni securizate pentru modulele interne;
- Parole stocate hash-uit (algoritmi moderni – ex. bcrypt/argon2);
- Politici de complexitate parolă;
- Expirare sesiune configurabilă;
- Blocare cont după tentative repetate eșuate.

6.3.2 Model RBAC (Role-Based Access Control)

Permisiunile sunt gestionate granular, în funcție de rol.

Exemple de roluri:

- Super Administrator
- Administrator CMS
- Operator CRM
- Operator financiar
- Utilizator autentificat

Pentru fiecare rol:

- Acces limitat la modulele relevante;
- Control granular pe acțiuni (vizualizare, creare, editare, ștergere);
- Audit al modificărilor efectuate.

6.4 Protecție împotriva vulnerabilităților aplicațiilor web

Soluția propusă implementează mecanisme de protecție împotriva principalelor categorii de vulnerabilități ale aplicațiilor web, în conformitate cu bunele practici de securitate și cu recomandările OWASP.

Principalele măsuri implementate includ:

Protecție împotriva SQL Injection

- utilizarea ORM și a mecanismelor de prepared statements;
- interzicerea execuției interogărilor dinamice nesecurizate;
- validarea strictă a datelor de intrare.

Protecție împotriva Cross-Site Scripting (XSS)

- escaparea automată a datelor afișate în interfața aplicației;

- sanitizarea inputului utilizatorilor;
- configurarea politicilor de securitate la nivel browser (Content Security Policy).

Protecție împotriva Cross-Site Request Forgery (CSRF)

- utilizarea token-urilor CSRF pentru formulare și operațiuni critice;
- validarea token-urilor la nivel server pentru fiecare cerere relevantă.

Protecție împotriva atacurilor de tip session hijacking și brute force

- cookie-uri marcate Secure și HttpOnly;
- regenerarea ID-ului de sesiune după autentificare;
- limitarea numărului de încercări de autentificare și blocarea temporară a conturilor după tentative repetate eșuate.

Aceste mecanisme sunt completate de procesele de testare de securitate și audit descrise în secțiunea 6.8.

6.5 Protecție împotriva atacurilor externe

6.5.1 DDoS și atacuri volumetrice

- Utilizare mecanisme de protecție trafic la nivel de infrastructură (CDN, reverse proxy sau firewall aplicativ);
- Web Application Firewall (WAF);
- Rate limiting;
- Filtrare trafic malițios.

6.5.2 Limitare IP și blacklist

- Posibilitate blocare manuală IP;
- Listă neagră configurabilă;
- Monitorizare activitate suspectă.

6.6 Securitatea integrărilor externe

Pentru integrările MPay, card bancar, ERP, API zboruri:

- Comunicare securizată prin HTTPS;
- Validare semnătură digitală webhook (unde este cazul);
- Logare tranzacții;
- Retry logic controlat;
- Validare input extern;
- Izolare logică a conectorilor API.

6.7 Jurnalizare și audit

Sistemul va păstra loguri pentru:

- Autentificări și tentative eșuate;
- Modificări conținut;
- Modificări status solicitări;
- Tranzacții financiare;
- Modificări drepturi utilizatori;
- Erori aplicație și integrare.

Logurile:

- Sunt stocate securizat;
- Pot fi exportate;
- Sunt accesibile doar rolurilor autorizate.

6.8 Vulnerability Assessment și Penetration Testing

Înainte de punerea în producție, vor fi realizate:

6.8.1 Vulnerability Assessment

- Scanare automată aplicație;
- Identificare vulnerabilități cunoscute;
- Clasificare pe nivel criticitate.

6.8.2 Penetration Testing intern

- Testare manuală scenarii atac;

- Simulare SQL injection;
- Simulare XSS;
- Testare autentificare;
- Testare acces neautorizat;
- Testare configurare securitate.

Pentru fiecare vulnerabilitate identificată:

- Se va aplica plan de remediere;
- Se va revalida fix-ul implementat.

6.9 Protecția datelor cu caracter personal

Soluția va respecta principiile GDPR:

- Minimizarea colectării datelor;
- Colectare doar pentru scopuri legitime;
- Posibilitate anonimizare (unde aplicabil);
- Protecție acces neautorizat;
- Audit acces date sensibile.

6.10 Backup și Recuperare (Security & Integrity)

- Backup periodic bază de date;
- Backup fișiere media;
- Testare restaurare;
- Verificare integritate date;
- Procedură documentată.

6.11 Politici operaționale

- Acces la producție limitat;
- Acces bazat pe roluri;
- Trasabilitate modificări cod;
- Deploy controlat prin CI/CD;

- Separare medii Dev/Test/Pre-prod/Prod.

---

7. PLAN DE IMPLEMENTARE

7.1 Abordare generală

Implementarea soluției se va realiza etapizat, pe o perioadă maximă de 12 luni calendaristice, astfel încât să asigure livrări incrementale, validări intermediare și controlul riscurilor. Modelul de lucru propus este iterativ (Agile), cu sprinturi de 2 săptămâni, demo la final de sprint și livrări intermediare în mediu de pre-producere.

Principii-cheie:

- livrare incrementală și validare frecventă cu Beneficiarul;
- „security by design" și testare continuă (unit, integrare, automată);
- quality gate: fiecare feature story trebuie să treacă DoD (inclusiv teste accesibilitate automate + smoke manual);
- rollback & proceduri de urgență documentate;
- trasabilitate FR → implementare → test → livrabil.

7.2 Faze, obiective și PD alocate

Volumul total alocat proiectului este 700 Persoană-Zile (PD). În tabelul de mai jos este distribuţia pe faze, cu PD dedicate şi livrabilele principale.

Faza	Durată	PD alocate	Obiective & livrabile principale
Faza 1 – Analiză detaliată & Arhitectură	Luna 1–2	**80 PD**	Analiză completă FR, model de date, RBAC, specificaţii integrare (MPay, carduri, ERP, API zboruri), document arhitectură, backlog detaliat, diagramă logică/infra.

			Livrabile: ● Document arhitectură aprobat ● Backlog detaliat mapat la cerințe FR ● Model de date preliminar ● Specificații integrare externe
Faza 2 – Design UX/UI (WCAG AA)	Luna 2–3	**70 PD**	Wireframes & prototipuri, Design System (tokens, componente), audit UX accesibilitate, validare WCAG AA la nivel de design, prototip interactiv. Livrabil: Prototip aprobat + Design System accesibil.
Faza 3 – Dezvoltare Portal Public & CMS	Luna 3–6	**220 PD**	Implementare Blade SSR, meniuri, pagini, editor WYSIWYG, media management, search (Meilisearch), căutare, zboruri (conector + cache), migrare conținut public, integrare CDN/Redis. Livrabil: Portal public MVP în Pre-prod.
Faza 4 – Dezvoltare CRM / Module interne (Next.js)	Luna 5–8	**160 PD**	Implementare Admin/CRM (Next.js+TS), registre, workflow, filtre, exporturi (PDF/XLSX), jurnalizare, RBAC intern. Livrabil: Modul CRM complet în Pre-prod.
Faza 5 – Integrări & Funcționalităţi tranzacţionale	Luna 6–9	**110 PD**	Integrare MPay, plăţi cu card, webhook securizat, export ERP, SMTP/SMS, testare end-to-end pentru fluxuri de plată și integrări, fallback logic. Livrabil: Flux tranzacţional complet testat.
Faza 6 – Testare avansată, Securitate & Go-Live	Luna 9–12	**60 PD**	Testare performanţă k6 (baseline, spike, soak, stress), VA + PenTest intern, recovery testing, UAT, training AIC, stabilizare post go-live. Livrabil: Rapoarte testare + Proces-verbal Go-Live.

TOTAL	12 luni	**700 PD**

Distribuția PD pe faze include contribuția cumulată a tuturor rolurilor implicate în proiect (analiză, arhitectură, dezvoltare backend și frontend, DevOps, QA, securitate și management de proiect).

Distribuția PD pe roluri prezentată în Capitolul 8 reflectă efortul individual estimat al fiecărei categorii de specialiști, iar suma acestora corespunde volumului total de 700 Persoană-Zile alocat proiectului.

Observaţie: activităţile privind accesibilitatea (WCAG 2.1 / 2.2 nivel AA) sunt integrate transversal în fazele principale ale proiectului: Faza 2 (design), Faza 3 (implementare) şi Faza 6 (testare). Alocarea detaliată a efortului aferent accesibilităţii este prezentată în secţiunea 7.7.

7.3 Milestone-uri şi calendar

Milestone-uri majore:

- EOP (End of Phase) 1 – Arhitectură aprobată: sfârşitul lunii 2
- EOP 2 – Design aprobat (WCAG incl.): sfârşitul lunii 3
- EOP 3 – Portal public (MVP) în Pre-prod: sfârşitul lunii 6
- EOP 4 – CRM complet în Pre-prod: sfârşitul lunii 8
- EOP 5 – Integrări complete şi teste E2E: sfârşitul lunii 9
- EOP 6 – Testare finală, UAT, Go-Live: sfârşitul lunii 12

Fiecare milestone include lista de livrabile asociate și criterii de acceptanță clar documentate.

7.4 Metodologie de lucru

- Sprinturi: 2 săptămâni, cu backlog prioritizat; demo & retrospective la final de sprint.
- Daily stand-up: 15 min zilnic (echipa de proiect).

- Gating: Release în Pre-prod se face după trecerea check-list-ului de quality gates (tests automate + checklist accesibilitate + code review).
- Issue tracking & QA: toate defectele și cerințele noi sunt înregistrate într-un sistem de management al taskurilor și incidentelor (issue tracking system), cu trasabilitate la FR ID.
- Acces Beneficiar: Beneficiarul are acces în Pre-prod pentru review și UAT; sesiuni de validare la capetele de fază.

7.5 Controlul calității și Definition of Done (DoD)

Definition of Done pentru story/feature:

- cod compilat și testat (unit & integration);
- code review aprobat;
- build CI verde;
- toate testele automate asociate trecute (inclusiv Axe pentru accesibilitate fără erori critice);
- smoke manual (keyboard / screen reader) pe componenta relevantă;
- documentaţie minimă de utilizare / note ARIA incluse;
- demo public & acceptare PM/BA.

Calitate: testare continuă (CI), code review obligatoriu, pair-programming recomandat pentru module critice.

7.6 Managementul riscurilor şi buffer

Am planificat buffer operaţional în ultimele 3 luni (lunile 9–12) pentru a gestiona întârzierile la integrări externe, ajustări de performanţă, remediere vulnerabilităţi şi UAT. Dacă apar cereri importante de modificare (change requests), acestea vor fi tratate conform procedurii (analiză impact, estimare PD, accept de ambele părţi).

Mecanisme de monitorizare:

- registru riscuri actualizat lunar;
- review riscuri la fiecare milestone;
- plan de continuitate resurse (înlocuire resurse cu profil echivalent).

7.7 Accesibilitate (WCAG) — alocare PD și timeline

Pentru a asigura conformitatea cu WCAG 2.1 / 2.2 nivel AA, activităţile dedicate de accesibilitate sunt introduse explicit în program şi sunt acoperite din volumul total de 700 PD.

Total alocat pentru accesibilitate: 12 Persoană-Zile, repartizate astfel:

- Design & audit UX (Faza 2 – Luna 2–3): 5 PD
    - Audit design (wireframes/prototip), definire Design System accesibil (tokens culori, fonturi, spacing), specificaţii ARIA, prototipuri testabile. Livrabil: Raport audit + Design System accesibil.
- Implementare & integrare (Faza 3 – Luna 3–6): 4 PD
    - Implementare componente accesibile, focus management, integrare teste automate (axe), documentaţie pentru dezvoltatori. Livrabil: Biblioteca componente accesibile + config CI axe.
- Testare, user testing & remediere (Faza 6 – Luna 9–12): 3 PD
    - Teste manuale (keyboard, screen readers NVDA/VoiceOver), sesiuni user testing cu persoane reale, remediere şi raport final. Livrabil: Raport testare accesibilitate + Accessibility Conformance Report (WCAG-EM).

Acceptance criteria (WCAG):

- paginile și fluxurile critice respectă criteriile WCAG AA, validate prin testare automată și verificări manuale.
- funcționalităţile esențiale sunt utilizabile exclusiv prin tastatură;
- verificări cu screen readers pe paginile critice fără blocker-e;
- rapoarte manuale şi user testing confirmate înainte de semnarea proces-verbalului de recepţie.

Definition of Done include cerinţele de accesibilitate pentru stories care afectează UI.

7.8 Livrabile pe fază (sumar)

- Faza 1: Document arhitectură, backlog detaliat, specificaţii integrare.
- Faza 2: Prototip UI aprobat, Design System accesibil, raport audit design.
- Faza 3: Portal public MVP în Pre-prod, CMS, Search integrat, migrare conținut public.
- Faza 4: Modul CRM complet funcțional.
- Faza 5: Fluxuri tranzacţionale integrate (MPay/card), export ERP, SMTP/SMS funcţional.
- Faza 6: Rapoarte testare performanţă, securitate, accesibilitate; UAT complet; Go-Live; documentaţie finală şi training.

7.9 Acceptare, predare și predare operațională

- La finalul fiecărei faze se va întocmi un raport de progres şi livrabilele aferente vor fi puse în Pre-prod pentru validare.
- UAT formal se desfăşoară în luna 11; remediile se finalizează în luna 12.
- La semnarea proces-verbalului de recepţie finală începe garanţia de 12 luni.
- Se predă documentaţia tehnică completă (arhitectură, deploy, proceduri DR, manuale utilizator, Accessibility Conformance Report) şi se organizează sesiuni de training pentru echipa AIC (STIT).

7.10 Observaţii finale şi condiţii

- Any dependency (ex.: documentaţia API zboruri, sandbox MPay, traduceri) se consideră responsabilitate a Beneficiarului – aceste dependențe sunt enunţate în Capitolul 13 (Ipoteze şi dependenţe). Întârzierile în furnizarea acestor elemente pot necesita replanificare.
- Orice cerere de schimbare va fi tratată prin procedură formală change-request, cu estimare PD suplimentare şi acord scris.

---

8. ORGANIZAREA ECHIPEI ȘI RESPONSABILITĂȚI

8.1 Structura echipei de implementare

Pentru realizarea proiectului în termenul contractual de maximum 12 luni și în limita bugetului maxim admisibil de 700 PD, operatorul economic va aloca o echipă multidisciplinară, cu competențe complementare, capabilă să acopere integral domeniile funcționale, tehnice, de securitate și operaționale prevăzute în Caietul de Sarcini.

Structura echipei propuse este următoarea:

- Project Manager
- Business Analyst
- Arhitect / Tech Lead
- Backend Developers (minim 2 specialiști)
- Frontend Developer (React / Next.js + TypeScript)

- QA / Test Engineer
- DevOps Engineer
- Security Specialist
- UX/UI Designer

Această structură asigură:

- separarea clară a responsabilităților;
- control tehnic permanent;
- acoperirea cerințelor de performanță și securitate;
- respectarea calendarului de implementare.

8.2 Roluri și responsabilități

8.2.1 Project Manager (PM)

Responsabilități:

- Coordonarea generală a proiectului;
- Planificarea și monitorizarea activităților;
- Gestionarea riscurilor și dependențelor;
- Organizarea ședințelor de lucru și a milestone-urilor;
- Raportarea progresului către beneficiar;
- Coordonarea procesului de UAT și Go-Live;
- Asigurarea respectării termenului de 12 luni și a bugetului de 700 PD.

Implicare estimată: 60 PD pe durata proiectului.

8.2.2 Business Analyst (BA)

Responsabilități:

- Analiza detaliată a cerințelor funcționale (FR);
- Elaborarea specificațiilor funcționale și a fluxurilor operaționale;
- Definirea proceselor CRM și tranzacționale;
- Clarificarea cerințelor de integrare externă;
- Validarea implementării împreună cu beneficiarul;
- Suport în etapa de UAT.

Implicare estimată: 30 PD.

8.2.3 Arhitect / Tech Lead

Responsabilități:

- Definirea și validarea arhitecturii soluției;
- Stabilirea standardelor de dezvoltare;
- Supervizarea implementării tehnice;
- Code review continuu;
- Validarea deciziilor tehnice majore;
- Suport în integrarea sistemelor externe;
- Supervizarea optimizării performanței;
- Implicare în fazele de testare avansată și Go-Live;
- Validarea măsurilor de securitate la nivel arhitectural.

Implicare estimată: 60 PD, distribuite pe întreaga durată a proiectului, cu implicare intensivă în fazele 1–3 și în fazele de integrare și optimizare finală.

Această alocare asigură control tehnic permanent și reducerea riscurilor arhitecturale.

8.2.4 Backend Developers

Responsabilități:

- Implementare aplicație Laravel (portal public + API interne);
- Dezvoltare CMS modular;
- Implementare module CRM și logică business;
- Integrare MPay, plăți card, ERP, API zboruri;
- Implementare RBAC și jurnalizare;
- Optimizare performanță backend;
- Remediere defecte identificate în testare.

Implicare estimată: 200 PD.

8.2.5 Frontend Developer (React / Next.js + TypeScript)

Responsabilități:

- Implementare interfață publică (Blade + Tailwind);
- Implementare aplicație Admin/CRM în Next.js;
- Integrare cu API backend;
- Implementare componente reutilizabile;
- Asigurare conformitate UX/UI;
- Asigurare accesibilitate (WCAG);
- Optimizare experiență utilizator.

Implicare estimată: 165 PD.

8.2.6 QA / Test Engineer

Responsabilități:

- Elaborarea planului de testare;
- Testare funcțională și integrare;
- Testare regresie;
- Testare performanță (k6);
- Validare cerințe non-funcționale;
- Suport în UAT;
- Elaborare rapoarte testare.

Implicare estimată: 75 PD.

8.2.7 DevOps Engineer

Responsabilități:

- Containerizare Docker;
- Configurare CI/CD;
- Configurare medii Dev/Test/Pre-prod/Prod;
- Configurare monitorizare;
- Implementare proceduri backup & restore;
- Suport pentru punerea în producție;
- Optimizare infrastructură pentru performanță.

Implicare estimată: 50 PD.

8.2.8 Security Specialist

Responsabilități:

- Implementare politici securitate aplicație;
- Vulnerability Assessment;
- Penetration Testing intern;
- Analiză și remediere vulnerabilități;
- Validare conformitate OWASP;
- Suport în configurarea mecanismelor de protecție.

Implicare estimată: 15 PD.

8.2.9 UX/UI Designer

Responsabilități:

- Elaborare wireframes;
- Prototip interactiv;
- Design responsive;
- Elaborare design system;
- Validare conformitate WCAG;
- Ajustări pe baza feedback-ului beneficiarului.

Implicare estimată: 45 PD.

8.3 Distribuția totală a efortului pe roluri

Rol	PD
Project Manager	60

Business Analyst	30
Arhitect / Tech Lead	60
Backend Developers	200
Frontend Developer	165
QA / Test Engineer	75
DevOps Engineer	50
Security Specialist	15
UX/UI Designer	45
TOTAL	**700 PD**

Distribuția este echilibrată și asigură:

- control tehnic continuu;
- acoperirea cerințelor complexe de integrare;
- suport pentru performanță și securitate;
- livrarea în termenul de 12 luni.

8.4 Model de colaborare

- Sprinturi de 2 săptămâni;
- Demo la final de sprint;
- Backlog priorizat;
- Sistem formal de issue tracking;
- Acces beneficiar la mediul de pre-producere;
- Ședințe regulate de progres.

8.5 Continuitatea și disponibilitatea echipei

Operatorul economic se angajează să asigure:

- Alocarea resurselor pe întreaga durată contractuală;
- Înlocuirea resurselor, atunci când situația o impune, cu personal cu competențe echivalente sau superioare;
- Documentarea continuă a codului și arhitecturii;
- Continuitatea livrării fără impact asupra termenului contractual.

9. MANAGEMENTUL RISCURILOR

9.1 Abordare generală

Operatorul economic aplică un proces formal de management al riscurilor pe întreaga durată a proiectului, începând cu faza de analiză și continuând până la finalizarea perioadei de stabilizare post Go-Live.

Procesul include:

- Identificarea riscurilor;
- Evaluarea impactului și probabilității;
- Definirea măsurilor de prevenție;
- Definirea planului de mitigare;
- Monitorizarea continuă în cadrul ședințelor de progres;
- Actualizarea registrului de riscuri pe durata proiectului.

Riscurile sunt clasificate în:

- Riscuri tehnice;
- Riscuri operaționale;
- Riscuri de integrare;
- Riscuri de performanță;
- Riscuri de securitate;
- Riscuri de calendar.

9.2 Matricea principală de riscuri

9.2.1 Riscuri tehnice

Risc	Probabilitate	Impact	Măsuri preventive	Plan de mitigare
Complexitate ridicată cerințe funcționale	Medie	Mare	Analiză detaliată în Faza 1	Ajustare backlog și reprioritizare iterativă
Subestimare efort integrare externă	Medie	Mare	Clarificări tehnice timpurii	Utilizare buffer proiect
Probleme performanță DB	Mică	Mare	Indexare și testare timpurie	Tuning DB și scalare orizontală

9.2.2 Riscuri de integrare

Risc	Probabilitate	Impact	Măsuri preventive	Plan de mitigare
Întârziere furnizare API zboruri	Medie	Mare	Definire timpurie specificații	Mock API și fallback temporar
Probleme integrare MPay / card	Mică	Mare	Testare sandbox timpurie	Testare incrementală + suport furnizor
Probleme ERP export	Mică	Mediu	Definire format clar	Ajustare mapare date
Dependenta de disponibilitatea serviciului MPass	Mică	Mediu	Testare timpurie a integrării SAML și	Cooperare tehnică cu administratorul serviciului MPass

	fallback autentificare locală

9.2.3 Riscuri de performanță

Risc	Probabilitate	Impact	Măsuri preventive	Plan de mitigare
Trafic mult peste estimări	Medie	Mare	CDN + cache multi-layer	Scalare aplicație
Degradare performanță în timp	Mică	Mediu	Soak testing	Optimizare și tuning
Degradare performanță API zboruri extern	Medie	Mediu	Cache aplicație	Fallback temporar + notificare operațională

9.2.4 Riscuri de securitate

Risc	Probabilitate	Impact	Măsuri preventive	Plan de mitigare
Vulnerabilități aplicație	Mică	Mare	Code review + VA	Remediere rapidă
Atac DDoS	Medie	Mare	WAF + CDN	Filtrare trafic
Tentative brute force	Medie	Mediu	Rate limiting	Blocare IP/cont

9.2.5 Riscuri organizaționale

Risc	Probabilitate	Impact	Măsuri preventive	Plan de mitigare
Întârzieri în validare din partea beneficiarului	Medie	Mediu	Planificare milestone clare	Ajustare plan și buffer
Schimbări de cerințe	Medie	Mediu	Backlog controlat	Gestionare prin change request

9.3 Riscuri calendaristice

Proiectul include buffer operațional în ultimele 3 luni pentru:

- Integrare finală;
- Testare performanță;
- Securitate;
- Ajustări UAT.

Această planificare reduce riscul depășirii termenului de 12 luni.

9.4 Mecanisme de control și monitorizare

- Registru formal de riscuri actualizat lunar;
- Revizuire riscuri în fiecare ședință de progres;
- Escaladare rapidă către stakeholderi;
- Raportare periodică către beneficiar.

9.5 Continuitatea resurselor

Pentru reducerea riscului de discontinuitate:

- Documentare continuă a codului;
- Code review obligatoriu;
- Înlocuire resurse cu profil echivalent în situațiile în care continuitatea proiectului o impune;
- Alocare resurse suplimentare în caz de risc major.

---

10. PLANUL DE TESTARE

10.1 Obiective

Planul de testare are ca obiectiv asigurarea faptului că:

- Toate cerințele funcționale (FR) din Caietul de Sarcini sunt implementate corect și complet.

- Cerințele non-funcționale (performanță, securitate, accesibilitate, disponibilitate) sunt respectate.
- Integrările externe (MPay, carduri, ERP, API zboruri, SMTP/SMS etc.) funcționează corect în condiții reale.
- Sistemul este stabil și pregătit pentru operare 24/7.
- Acceptanța finală (UAT) se realizează fără blocker-e și cu remedierea defectelor critice.

10.2 Principii și strategie generală

Testarea se realizează pe mai multe nivele și este integrată în procesul de dezvoltare iterativă (Agile), cu sprinturi de 2 săptămâni.

Nivelurile principale de testare sunt:

1. Testare la nivel dezvoltare – unit tests și verificări statice.
2. Testare de integrare – verificarea comunicării dintre componente și API-uri.
3. Testare funcțională sistem – validarea implementării cerințelor funcționale.
4. Testare regresie – verificarea stabilității sistemului după modificări.
5. Testare performanță – evaluarea comportamentului sistemului sub sarcină.
6. Testare securitate – identificarea vulnerabilităților și verificarea configurației de securitate.
7. Testare accesibilitate – verificarea conformității cu WCAG AA.
8. Testare recovery – verificarea procedurilor de backup și restaurare.
9. User Acceptance Testing (UAT).

Pentru fiecare cerință funcțională din Caietul de Sarcini se va asigura trasabilitatea către implementare și către cazurile de test asociate.

10.3 Organizare, responsabilități și medii

Activitățile de testare sunt realizate de echipa de proiect descrisă în Capitolul 8, cu implicarea directă a rolurilor tehnice relevante (QA/Test Engineer, dezvoltatori, DevOps, specialist securitate și management de proiect).

Responsabilitățile principale în procesul de testare includ:

- planificarea și coordonarea activităților de testare;
- elaborarea și executarea cazurilor de test;

- monitorizarea defectelor și verificarea remedierilor;
- realizarea testelor de performanță și securitate;
- suport pentru desfășurarea User Acceptance Testing (UAT) împreună cu beneficiarul.

Medii de testare

Pentru desfășurarea testării sunt utilizate următoarele medii:

Dev mediu utilizat pentru dezvoltare și rularea testelor automate în cadrul pipeline-ului CI.

Test (QA) mediu dedicat testării funcționale și testelor de regresie.

Pre-prod mediu care reproduce cât mai fidel mediul de producție și este utilizat pentru testele de performanță, securitate și pentru UAT.

Prod mediul de producție în care se realizează doar verificări operaționale limitate (ex. smoke tests post-deploy) și monitorizare.

Mediile sunt configurate astfel încât să permită testarea controlată a funcționalităților, fără impact asupra utilizatorilor finali.

10.4 Testare funcțională

Scopul testării funcționale este verificarea implementării corecte a cerințelor funcționale definite în Caietul de Sarcini.

Activități principale:

- Elaborarea cazurilor de test pentru fiecare cerință funcțională;
- Executarea testelor funcționale asupra modulelor portal, CMS, CRM, plăți și integrări externe;
- Verificarea fluxurilor operaționale critice (ex.: înregistrare solicitare, procesare plată, confirmare tranzacție, export ERP);
- Testarea scenariilor negative și a validărilor de input;
- Verificarea corectitudinii rolurilor și permisiunilor utilizatorilor.
- Rezultatele testării sunt documentate și utilizate pentru remedierea eventualelor defecte identificate.

10.5 Testare de integrare

Scope

- Integrarea cu sisteme externe: MPay, carduri bancare, ERP, API zboruri, SMTP, SMS, MPass, altele.

Activități

- Teste pe sandbox-uri ale furnizorilor (MPay sandbox, bancă sandbox).
- Simulare callback/webhook (duplicități, timeout, erori).
- Teste pentru export ERP (format, consistență, scheduling).
- Testarea retry/backoff si fallback-uri.

Criterii

- Fluxuri integrate funcționează end-to-end în mediu pre-prod; erori de interfațare documentate și rezolvate.

10.6 Testare regresie

- Teste automate rulate la finalul fiecărui sprint major.
- Teste manuale pentru elemente critice și cazuri nou introduse.
- Regresia include testele de accesibilitate automate/run smoke.
- Raport regresie înainte de UAT și Go-Live.

10.7 Testare performanță

Testarea de performanță are ca obiectiv validarea comportamentului sistemului în condiții de trafic ridicat și verificarea respectării indicatorilor de performanță definiți în cerințele non-funcționale.

Scenarii principale:

1. Baseline Load Test – simularea unui trafic constant în condiții normale.
2. Spike Test – simularea unor creșteri bruște de trafic.
3. Soak Test – verificarea stabilității sistemului în condiții de încărcare susținută.
4. Stress Test – identificarea limitelor de capacitate ale sistemului.

Testele sunt realizate în medii controlate, iar rezultatele sunt analizate pentru identificarea eventualelor optimizări necesare.

10.8 Testare securitate (VA + PenTest intern)

Activități

- Vulnerability Assessment: scanare automată (OWASP, dependency checks).
- Penetration Testing intern: testare manuală pentru SQLi, XSS, CSRF, autentificare, elevare privilegii.
- Testare DDoS controlată în colaborare cu infrastructura de securitate a beneficiarului sau cu mecanismele de protecție configurate la nivel de infrastructură.
- Revizuire recordurilor de jurnalizare și mecanisme de protecție.

Criterii

- Defectele critice de securitate remediate și re-testate înainte de UAT/Go-Live.

Livrabile

- Raport VA (clasificare severitate), Raport PenTest (scenarii, PoC), plan remediere și dovadă retest.

10.9 Testare accesibilitate (WCAG 2.1 / 2.2 AA)

Testarea accesibilității are ca obiectiv verificarea conformității interfeței cu cerințele WCAG nivel AA pentru paginile și fluxurile critice ale portalului.

Strategia de testare include:

- verificări automate ale componentelor UI;
- verificări manuale privind navigarea prin tastatură, focus management și utilizarea cu tehnologii asistive;
- validarea contrastului și a structurii semantice a paginilor;
- verificări de utilizabilitate pentru fluxurile principale.

Rezultatele testării sunt documentate într-un raport de accesibilitate, iar eventualele neconformități sunt remediate înainte de acceptanța finală.

10.10 Testare recovery / backup & restore

Activități

- Test de backup & restore al bazei de date şi fişierelor media;
- Teste de restaurare în mediu controlat;
- Verificare integritate date şi timp estimat de restaurare.

Criterii

- Restaurare completă funcţională într-un interval acceptabil (documentat), date verificate.

Livrabile

- Procedură backup & restore, raport test restaurare, recomandări DR.

10.11 User Acceptance Testing (UAT)

Plan

- Mediu pre-prod pus la dispoziție beneficiarului;
- Scenarii UAT negociate și aprobate (fluxuri de business principale);
- Echipa AIC execută UAT împreună cu echipa Smartdata;
- Defectele raportate în sistemul de ticketing; severități, prioritizare și plan remediere.

Acceptanță

- Acceptanţa se semnează după remedierea tuturor defectelor critice și după confirmarea conformității cu criteriile de acceptanță (funcționale, performanță, accesibilitate, securitate).

10.12 Traceability & Matrice FR→Test

- Pentru fiecare FR din Caiet se va crea o intrare în matricea de trasabilitate: FR ID → Modul implementare → Test(s) asociat(e) → Rezultat.
- Matricea va fi livrată ca anexă (format tabelar) și actualizată pe parcursul proiectului.

10.13 Defect management și raportare

- Defecte sunt clasificate: Critical / Major / Medium / Minor.
- Flux: detectare → creare ticket → triere → remediere → retest → închidere.
- Raportare: rapoarte săptămânale QA, raport special la final de fază (test summary).
- SLA remediere: conform Capitolul 11 (Garanție & SLA) pentru defecte în garanție; pentru defecte în UAT se aplică termenele convenite în planul de remediere.

10.14 Automation & CI

- Testele unitare și integrare rulează în CI per PR;
- Testele de accesibilitate automate (axe) rulează pe CI pentru pagini critice;
- Pipeline CI include joburi pentru execuție k6 pe mediu pre-prod (la cererea QA/PM).

10.15 Livrabile asociate testării

- Plan de testare formal (acest document).
- Matrice FR → Test → Rezultat (anexă).
- Rapoarte test funcțională (pe faze).
- Rapoarte regresie.
- Raport performanță (k6).
- Raport Vulnerability Assessment & Penetration Testing.
- Raport accesibilitate (axe + manual + user testing).
- Raport recovery.
- Proces-verbal UAT și proces-verbal Go-Live.

10.16 Estimare efort testare (orientativ)

- Testare funcțională & integrare: ~40 Persoană-Zile (inclusiv automatizare)
- Testare performanță: ~10 Persoană-Zile
- Testare securitate (VA + PT intern): ~10 Persoană-Zile
- Testare accesibilitate (automat + manual + user testing): ~10 Persoană-Zile

- Testare recovery & UAT: ~5 Persoană-Zile

Total efort QA/Test: 75 Persoană-Zile, aliniat cu distribuția efortului pe roluri prezentată în Capitolul 8.

10.17 Procedură de semnare și acceptanță

- După finalizarea UAT și remedieri, se pregătește proces-verbal de recepție;
- Echipa AIC semnează acceptanța pe livrabilele convenite și pe raportul final de testare;
- Începerea perioadei de garanție de 12 luni la data semnării proces-verbalului.

11. GARANȚIE ȘI SLA

11.1 Perioada de garanție

Operatorul economic oferă o perioadă de garanție de 12 luni de la data semnării procesului-verbal de recepție finală și punere în producție a sistemului.

Perioada de garanție include:

- Remedierea defectelor de natură tehnică apărute în exploatarea normală a sistemului;
- Corectarea erorilor de implementare;
- Remedierea problemelor de securitate identificate ulterior punerii în producție;
- Suport tehnic pentru incidentele raportate.

Garanția nu include dezvoltări noi sau modificări funcționale care depășesc cerințele din Caietul de Sarcini, acestea putând face obiectul unui contract separat de dezvoltare continuă.

11.2 Definirea incidentelor

Incidentele sunt clasificate pe niveluri de severitate, în funcție de impactul asupra funcționării sistemului:

Severitate Critică (S1)

- Sistem indisponibil total;

- Modul plăți nefuncțional;
- Blocare acces utilizatori;
- Vulnerabilitate majoră de securitate.

Severitate Majoră (S2)

- Funcționalitate esențială afectată;
- Impact semnificativ asupra proceselor operaționale;
- Erori recurente în module critice.

Severitate Medie (S3)

- Funcționalitate secundară afectată;
- Impact limitat;
- Workaround disponibil.

Severitate Minoră (S4)

- Probleme de afișare;
- Ajustări minore;
- Erori fără impact operațional.

11.3 Timpi de răspuns și remediere (SLA)

Conform cerințelor din documentația de atribuire, operatorul economic va asigura următorii timpi:

Severitate	Timp de răspuns	Timp de remediere
S1 – Critic	max. 15 minute	max. 4 zile lucrătoare
S2 – Major	max. 2 ore	termen agreat în funcție de complexitate
S3 – Mediu	max. 1 zi lucrătoare	inclus în următorul ciclu de mentenanță
S4 – Minor	max. 2 zile lucrătoare	inclus în următorul ciclu de mentenanță

Pentru incidentele critice:

- Se va transmite raport de progres periodic (orar sau conform agreat);
- Se va menține comunicare permanentă până la rezolvare.

11.4 Modalitate de raportare incidente

Incidentele vor fi raportate prin:

- Sistem de ticketing dedicat;
- Email oficial;
- Canal de comunicare stabilit contractual.

Fiecare incident va primi:

- Număr unic de identificare;
- Clasificare severitate;
- Status (Open / In Progress / Resolved / Closed);
- Istoric intervenții.

11.5 Suport tehnic

Pe perioada de garanție, operatorul economic asigură:

- Helpdesk în intervalul orar agreat;
- Asistență în limba română, engleza și rusă;
- Suport pentru echipa tehnică AIC (STIT);
- Consultanță tehnică pentru operare.

11.6 Monitorizare și prevenție

Pentru reducerea incidentelor, vor fi implementate:

- Monitorizare aplicație și infrastructură;
- Alertare automată la degradare performanță;
- Verificări periodice de securitate;
- Actualizări de securitate pentru componentele software utilizate.

11.7 Excluderi din garanție

Nu fac obiectul garanției:

- Modificări legislative care impun dezvoltări noi;
- Extinderi funcționale nespecificate în Caietul de Sarcini;
- Intervenții neautorizate asupra sistemului;
- Probleme generate de infrastructura externă (hardware, rețea) dacă nu sunt cauzate de aplicație.

11.8 Dezvoltare continuă post-garanție

Operatorul economic propune, opțional, încheierea unui contract de mentenanță și dezvoltare continuă după expirarea perioadei de garanție, care poate include:

- Actualizări evolutive;
- Extinderi funcționale;
- Optimizări performanță;
- Audituri periodice de securitate;
- Suport extins.

---

12. LIVRABILE FINALE

12.1 Principii generale

La finalizarea proiectului, operatorul economic va livra beneficiarului toate componentele necesare pentru exploatarea completă și sustenabilă a sistemului informațional.

Livrabilele vor fi structurate astfel încât să permită operarea, mentenanța și dezvoltarea ulterioară a sistemului fără dependență operațională de operatorul economic.

Pachetul de livrabile include atât componentele software ale soluției, cât și documentația tehnică, procedurile operaționale și materialele de instruire necesare utilizării și administrării sistemului.

Toate livrabilele vor fi predate în format electronic, iar acolo unde este necesar și relevant pentru exploatarea sistemului, și în format editabil.

12.2 Livrabile software

12.2.1 Cod sursă complet

- Codul sursă al aplicației Laravel;
- Codul sursă al aplicației Admin (Next.js);
- Scripturi de configurare;
- Scripturi migrare bază de date;
- Fișiere de configurare relevante (fără date sensibile).

Codul va fi livrat într-un format organizat și documentat.

12.2.2 Imagini Docker

- Dockerfile-uri pentru toate componentele.
- Fișiere docker-compose utilizate pentru rularea și configurarea mediilor de dezvoltare și testare.
- Instrucțiuni pentru build și rulare.
- Configurații pentru medii diferite (Dev/Test/Prod).

12.2.3 Baza de date

- Structură bază de date;
- Scripturi creare schema;
- Documentație model date;
- Instrucțiuni restaurare backup.

12.3 Documentație tehnică

12.3.1 Documentație arhitecturală

- Diagramă arhitecturală logică;
- Diagramă infrastructură;
- Descriere componente;

- Fluxuri principale (plăți, zboruri, CRM);
- Model RBAC.

12.3.2 Documentație instalare și configurare

- Pași instalare aplicație;
- Pași configurare mediu producție;
- Configurare conexiuni externe;
- Proceduri de restart și rollback.

12.3.3 Documentație API

- Listă endpoint-uri;
- Parametri de intrare;
- Exemple request/response;
- Mecanism autentificare;
- Standard OpenAPI.

12.3.4 Proceduri operaționale

- Procedură backup;
- Procedură restore;
- Procedură deploy;
- Procedură gestionare incidente;
- Procedură actualizare sistem.

12.4 Documentație utilizator

12.4.1 Manual utilizator CMS

- Gestionare pagini;
- Editor conținut;
- Gestionare bannere;

- Publicare conținut;
- Gestionare traduceri.

12.4.2 Manual utilizator CRM

- Gestionare solicitări;
- Workflow statusuri;
- Export date;
- Filtrare și căutare;
- Vizualizare rapoarte.

12.4.3 Ghid utilizator operațional

- Gestionare plăți;
- Monitorizare tranzacții;
- Interpretare notificări.

12.5 Livrabile asociate testării

- Plan de testare final;
- Matrice trasabilitate FR → implementare → test;
- Raport testare funcțională;
- Raport testare performanță;
- Raport Vulnerability Assessment;
- Raport Penetration Testing;
- Raport testare Recovery;
- Proces-verbal UAT.

12.6 Instruire și transfer de cunoștințe

Operatorul economic va asigura:

- Sesiuni de instruire pentru utilizatori CMS;

- Sesiuni de instruire pentru utilizatori CRM;
- Sesiuni tehnice pentru echipa STIT;
- Materiale suport (prezentări, ghiduri).

Activitățile de instruire vor asigura transferul complet de cunoștințe către echipa tehnică a beneficiarului, astfel încât sistemul să poată fi operat și administrat în mod independent după finalizarea proiectului.

12.7 Punere în producție

Livrabile asociate Go-Live:

- Pachet livrare producție;
- Configurare finală infrastructură;
- Validare post-deploy;
- Raport stabilizare inițială.

12.8 Predare și recepție

La finalizarea implementării:

- Se va întocmi proces-verbal de recepție;
- Se va confirma funcționarea conform cerințelor;
- Se va transfera oficial responsabilitatea operațională către beneficiar;
- Va începe perioada de garanție.

---

13. IPOTEZE, DEPENDENȚE ȘI DELIMITĂRI

13.1 Scopul capitolului

Prezentul capitol definește ipotezele de implementare, dependențele externe și delimitările proiectului, în vederea asigurării unei înțelegeri comune între operatorul economic și beneficiar privind responsabilitățile, condițiile de livrare și factorii externi care pot influența execuția proiectului.

Aceste ipoteze stau la baza estimării de 700 PD și a termenului de implementare de 12 luni calendaristice.

13.2 Ipoteze privind infrastructura

Se consideră că:

- Infrastructura hardware necesară operării sistemului (servere, rețea, firewall, load balancer) este asigurată de beneficiar sau de un furnizor desemnat de acesta, conform practicilor uzuale pentru sistemele informatice instituționale. Operatorul economic va furniza recomandările tehnice necesare pentru dimensionarea și configurarea infrastructurii.
- Există posibilitatea rulării containerelor Docker în mediul de producție;
- Există posibilitatea configurării unui mecanism de distribuție conținut (CDN sau reverse proxy) la nivelul infrastructurii beneficiarului;
- Domeniul www.airport.md este disponibil și gestionat de beneficiar;
- Certificatul SSL/TLS este furnizat de beneficiar sau poate fi emis prin infrastructura disponibilă.

Operatorul economic va furniza specificații minime și recomandări pentru dimensionarea infrastructurii. Aceste recomandări vor include cerințe minime privind resursele hardware, configurația serviciilor și parametrii necesari pentru atingerea obiectivelor de performanță definite în prezentul document.

13.3 Ipoteze privind integrarea cu sisteme externe

Implementarea integrărilor este condiționată de furnizarea de către beneficiar a documentației tehnice complete și a accesului la medii de test (sandbox).

Se consideră că:

- Documentația API pentru sistemul de afișare zboruri este disponibilă și actualizată;
- Specificațiile tehnice pentru integrarea cu MPay sunt furnizate;
- Specificațiile pentru integrarea plăților cu card bancar sunt furnizate de banca desemnată de beneficiar;
- Specificațiile pentru exportul către ERP (format fișier sau API) sunt furnizate în timp util;
- Accesul la medii de test pentru aceste integrări este asigurat în fazele 1–3.

Întârzierile în furnizarea acestor elemente pot influența calendarul implementării.

13.4 Ipoteze privind conținutul și traducerile

Se consideră că:

- Conținutul textual pentru toate paginile publice este furnizat de beneficiar;
- Traducerile în limbile RO/RU/EN sunt furnizate de beneficiar;
- Conținutul este livrat în format digital editabil;
- Materialele media (imagini, documente PDF etc.) sunt furnizate de beneficiar.

Operatorul economic va asigura migrarea conținutului public existent în limita volumului actual al site-ului, cu ajustări minore de structură conform noului design.

Conținutul intern nepublic (arhive interne sau sisteme externe) nu face obiectul migrării în cadrul prezentei oferte, decât dacă este specificat explicit prin cerințe funcționale.

13.5 Ipoteze privind utilizatorii și procesele operaționale

Se consideră că:

- Beneficiarul va desemna persoane responsabile pentru validarea cerințelor și UAT;
- Beneficiarul va furniza în timp util feedback pentru livrările intermediare;
- Procesele operaționale interne (workflow-uri, statusuri, registre) sunt clar definite în faza de analiză;
- Structura organizațională relevantă pentru definirea rolurilor este stabilă.

Întârzierile în validări pot influența planificarea sprinturilor ulterioare.

13.6 Delimitări ale proiectului

Prezenta ofertă include:

- Dezvoltarea unui portal web nou conform cerințelor din Caietul de Sarcini;
- Implementarea modulelor CRM și tranzacționale;
- Integrarea cu sistemele externe menționate;
- Testare, audit securitate, documentație și instruire;
- Perioadă de garanție de 12 luni.

Nu fac obiectul prezentei oferte:

- Dezvoltări suplimentare care nu sunt prevăzute în cerințele funcționale din Caietul de Sarcini;
- Migrarea altor sisteme informatice existente ale beneficiarului, cu excepția conținutului public al portalului actual, conform celor precizate în prezentul document;
- Administrarea și operarea pe termen lung a infrastructurii hardware (servere, rețea, echipamente);
- Achiziția de licențe software comerciale externe, în cazul în care beneficiarul optează pentru astfel de componente;
- Furnizarea serviciilor de hosting sau infrastructură cloud, dacă acestea nu sunt solicitate explicit prin documentația de atribuire.

13.7 Gestionarea modificărilor

Orice solicitare de modificare a cerințelor inițiale (change request) va fi:

- Analizată din punct de vedere tehnic;
- Estimată în Persoană-Zile suplimentare atunci când modificarea depășește domeniul cerințelor inițiale din Caietul de Sarcini;
- Integrată în plan doar cu acordul ambelor părți;
- Documentată formal.

Această procedură asigură transparență și control asupra impactului în termen și buget.

Data completării 05 martie 2026

Digitally signed by Martînenco Andrei
Date: 2026.03.06 15:00:16 EET
Reason: MoldSign Signature
Location: Moldova
MOLDOVA EUROPEANĂ

Cu stimă,

Smart Data SRL

Reprezentat autorizat Andrei Martinenco