

# DONATOR DE SANGE

## Raport de testare

### 1. Scopul documentului

Acest raport are rolul de a detalia activitatile legate de testarea aplicatiei Donator de Sange, dezvoltata in cadrul primului proiect MPS 2018. Documentul prezinta in detaliu tipurile de testare efectuate, tehnologii folosite, dezvoltarea cazurilor de testare (rezultat asteptat si rezultat actual), precum si informatii legate de acoperirea cu teste a produsului software.

### 2. Overview

“Donator de sange” este o aplicatie software medicala ce are ca scop imbunatatirea procesului de donare de sange existent. Aceasta scurteaza multe din procesele ce trebuiau facute direct la punctul de recoltare si ofera donatorilor informatii online. Fiecare user poate completa un chestionar online pentru a se verifica daca este eligibil pentru a dona sange.

Aplicatia a fost implementata sub forma unei aplicatii web. Printre tehnologiile folosite de catre echipa de dezvoltare se numara Angular (pentru interfata utilizator) si Node JS, Python, Docker (pentru implementarea partii backend).

**Testarea proiectului** a avut loc in doua etape.

In prima etapa s-a urmarit functionarea corespunzatoare a API-ului pus la dispozitie de catre server. Pentru acest lucru a fost realizata o suita de **teste functionale**, utilizand limbajul **Python** si, implicit, modulul **unittest**. Suita contine o serie de teste automate, cazurile testate de catre acestea fiind dezvoltate in sectiunea corespunzatoare.

In a doua etapa s-a urmarit verificarea unor scenarii in care se actioneaza asupra interfetei utilizator. Aceasta etapa consta in verificare tuturor elementelor care compun interfata grafica, pentru a asigura faptul ca aplicatia functioneaza conform specificatiilor. Din aceste cazuri de testare am dezvoltat o serie de **teste de acceptare**.

### 3. Teste functionale (backend)

In dezvoltarea testelor automate pentru verificarea functionalitatii API-ului, am tinut cont de forma cererilor, asa cum a fost stabilita in specificatiile initiale, cat si de forma raspunsurilor serverului. Testele au verificat accesul corespunzator la endpoint-uri, precum si consistenta datelor transferate.

Nr. crt.	Descriere caz de testare	Rezultat asteptat	Rezultat obtinut (DA/NU/alte observatii)
1.	Cerere POST /register	Cererea contine parametri necesari autentificarii unui utilizator. Serverul declanseaza o comanda de inserare in tabela donor, doctor sau employee si raspunde cu un id valid	DA
2	Cerere POST /register cu parametri invalizi	Serverul raspunde cu un mesaj de identificare a parametrului pentru care s-a trimis valoarea invalida	DA
3	Cerere GET /register	Serverul raspunde cu un cod de eroare invalid (404 sau 405) pentru a se arata invaliditatea cererii	HTTP 404
4	Cerere PUT /register	Serverul raspunde cu un cod de	HTTP 404

		eroare invalid (404 sau 405) pentru a se arata invaliditatea cererii	
5	Cerere DELETE /register	Serverul raspunde cu un cod de eroare invalid (404 sau 405) pentru a se arata invaliditatea cererii	HTTP 404
6	Cerere POST /login	Serverul primeste un tuplu de credentiale de login valide si intoarce un token de autentificare valid	DA
7	Cerere POST /login cu parametri invalizi	Serverul primeste un tuplu de credentiale de login invalide si intoarce un mesaj de eroare pentru a semnala invaliditatea unuia dintre parametri	DA
8	Cerere GET /login DELETE /login PUT /login (3 cazuri)	Serverul raspunde cu un cod de eroare invalid (404 sau 405) pentru a se arata invaliditatea cererii	HTTP 404
9	GET /donor, /doctor, /employee cu token de autentificare valid (3 cazuri)	Serverul intoarce informatii din baza de date legate de tipul de utilizator descris de fiecare endpoint	DA
10	GET /donor, /doctor, /employee fara token de autentificare in headerul cererii (3 cazuri)	Serverul raspunde cu un mesaj care semnaleaza incercarea accesarii unei resurse protejate	HTTP 401
11	GET /donor, /doctor, /employee cu ID invalid transmis ca parametru cererii (3 cazuri)	Se intoarce un mesaj care semnaleaza pasarea prin endpoint a unui parametru de interogare invalid (ID)	DA
12	DELETE /donor, /doctor, /employee cu specificare ID valid	Serverul elimina intrari din tabelele donor, doctor sau employee corespunzatoare ID-ului	DA, HTTP 200
13	DELETE /donor, /doctor, /employee cu specificare ID invalid	Se intoarce un mesaj care semnaleaza pasarea prin endpoint a unui parametru de interogare invalid (ID)	DA
14	POST /donor, /doctor, /employee	Serverul raspunde cu un cod de eroare invalid (404 sau 405) pentru a se arata invaliditatea cererii	HTTP 404
15	PUT /donor, /doctor, /employee	Serverul raspunde cu un cod de eroare invalid (404 sau 405) pentru a se arata invaliditatea cererii	HTTP 404
16	Cereri pentru /hospital	Serverul raspunde corespunzator cererilor de tip GET, POST, DELETE actionand asupra bazei de date	DA
17	Cereri pentru /transfusionCenter	Serverul raspunde corespunzator cererilor de tip GET, POST, DELETE actionand asupra bazei de date	DA

Suita de teste functionale documentata complet poate fi rulata dupa clonarea repository-ului. In total sunt **45** cazuri de testare acoperite de suita.

#### 4. Teste de acceptare (frontend)

Am inclus testarea de acceptare ca modalitate de testare a proiectului deoarece este critic sa se descopere orice problema care apare din perspectiva utilizatorului aplicatiei. Astfel, cazurile de testare sunt bazate pe explorarea interfetei grafice a jocului si sunt strans legate de functionarea corepunzatoare a componentelor de pe diversele pagini.

Nr. crt.	Pagina	Scenariu de utilizare	Rezultat (DA/NU/detalii)
1	Creare cont	Utilizatorul se afla pe pagina Register si isi completeaza credentialele pentru un nou cont	Se va putea realiza login cu email-ul si parola specificate anterior
2	Login	Utilizatorul se logheaza cu credentialele specificate la Register	Este permis accesul in aplicatie
3	Login cu credentiale invalide	Utilizatorul se logheaza cu cel putin un camp de credentiale invalid	Utilizatorul ramane pe pagina de login
4	Completare chestionar	Donatorul completeaza chestionarul referitor la donare de sange pe pagina specifica	Raspunsurile chestionarului sunt inregistrate
5	Vizualizarea donarilor anterioare	Donatorul se afla pe profilul sau si doreste sa vizualizeze un istoric al donarilor anterioare	DA
6	Vizualizarea datei urmatoarei donari	Donatorul se afla pe profilul sau si doreste sa vizualizeze urmatoarea data la care va putea dona sange	DA
7	Completarea cererilor pentru sange	Doctorul se afla logat in contul sau si completeaza detaliile unei cereri de donare sange pentru un pacient	Cererea este inregistrata si I se poate vedea statusul
8	Vizualizare status cereri	Doctorul se afla logat in contul sau si vrea sa vada statusul cererilor depuse in centrul de donare de care raspunde	DA
9	Expunere panou administratori	Panoul de administratori inglobeaza detalii despre toti utilizatorii (donatori, doctori, angajati ai centrului de transfuzii)	DA

## 5. Analiza si concluzii

Atat testele functionale, cat si cele de acceptare, au ajutat la rezolvarea anumitor inconsistente ale datelor si, cel mai important, au putut fi folosite in testarea de regresie. Astfel, suitele puteau fi rulate oricand dupa un update al codului pentru a se asigura faptul ca functionalitatile anterioare nu au fost afectate.

Printre principalele probleme(bugs/issues) descoperite (**si fixate pana la momentul prezentarii**) in procesul de testare se numara:

1. POST /register avea nevoie de un token de autentificare
2. GET /request nu continea toate obiectele de tip donation aferente
3. Exceptii MySQL la POST /register
4. Cereri blocate pentru ca continutul front-end-ului este livrat peste HTTPS dar call-urile la endpoint-uri se fac in HTTP
5. Internal Server Error (HTTP 500) la request-uri pe unele endpoints (/hospital, /transfusionCenter)
6. Eroare la serializarea tipului de utilizator (enum: DONOR, EMPLOYEE sau DOCTOR)
7. URL-uri pentru requesturi
8. Typos in legatura cu variabilele

