

Volunteer Events System

Proiect PAD

Belde Mădălina Georgiana
Berzescu Maria Mirela
Chindriş Maria Cristina
Rotuna Răzvan Harald

Contribuțiile membrilor

- **Belde Mădălina Georgiana** - a creat serverul de notificări care se ocupă de notificarea diferitelor tipuri de clienți;
- **Berzescu Maria Mirela** - a creat aplicația web care le permite ONG-urilor să creeze și să își gestioneze evenimentele;
- **Chindriș Maria Cristina** - a creat aplicația mobilă care le permite voluntarilor să vadă evenimente și să își declare interesul pentru participarea la acestea;
- **Rotuna Răzvan Harald** - a creat serverul de backend, care prezintă un REST API și permite o comunicare uniformă cu diferitele tipuri de clienți.

Descrierea proiectului

Proiectul constă într-un sistem de evenimente conceput pentru a ușura colaborarea dintre voluntari și ONG-uri. Este compus din patru aplicații, două pentru clienți și două servere.

Una din aplicațiile pentru clienți este o aplicație mobilă, dedicată voluntarilor, unde aceștia pot vizualiza evenimente și marca faptul că doresc să participe la acestea. Cealaltă aplicație pentru clienți este o aplicație web, dedicată ONG-urilor; aceasta poate fi folosită pentru crearea și gestionarea de evenimente.

Aplicațiile de tip server furnizează funcționalitatea principală de comunicare a sistemului. Una din aceste aplicații oferă un REST API clienților, în timp ce a doua aplicație notifică clienții când sunt noutăți: voluntarii, când apar postări noi într-un eveniment, respectiv ONG-urile, când sunt participanți noi la un eveniment propriu.

Domeniul de aplicabilitate

Aplicația are ca public țintă tinerii voluntari și ONG-urile în căutare de oameni interesați de evenimentele organizate de acestea. Ea oferă un mediu online, confortabil, prin intermediul căruia să se creeze legătura între părțile interesate.

În contextul zilelor noastre, ușurința cu care se poate obține un serviciu, nu atât serviciul în sine, este de multe ori factorul decisiv.

Un caz evident este acela în care o organizație are nevoie de sprijin pe plan logistic într-un timp cât mai scurt. O platformă dedicată tinerilor ce doresc să participe la diferite evenimente (de ex. la un festival de muzică sau un hackathon) este poate cel mai bun mod de a disemina informația în rândul celor interesați. Se diminuează astfel nevoia ONG-urilor de a

face publicitate pe alte canale și posibilele costuri implicate, precum și timpul de recrutare al voluntarilor.

Din punctul de vedere al tinerilor, modul facil în care pot obține detalii și ușurarea procesului de participare la un eveniment de care sunt interesați reprezintă principalul avantaj al folosirii aplicației.

Arhitectura proiectului

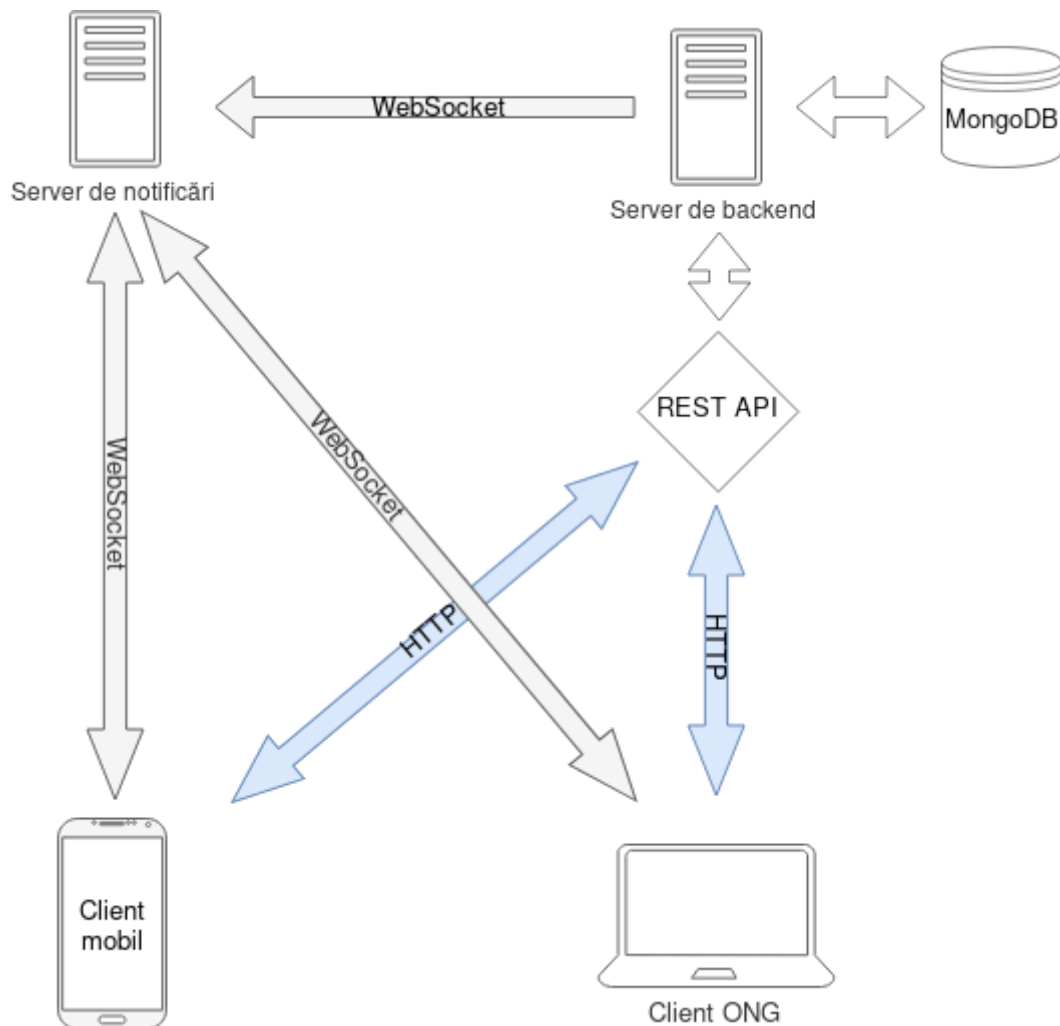


Diagrama de mai sus constă în patru componente majore: 2 aplicații server și 2 aplicații client.

Cele două aplicații de client au o conexiune bidirecțională cu serverul de backend. Ele trimit cereri de tip HTTP serverului, folosindu-se de REST API-ul oferit de acesta ca interfață. Aplicațiile de client au de asemenea conexiune bidirecțională cu serverul de notificări, pentru transmiterea datelor despre client, respectiv a notificărilor propriu-zise.

Serverul de backend are de asemenea conexiune cu serverul de notificări, pentru transmiterea datelor despre care vor fi notificați clienții, dar și conexiune cu baza de date, care este reprezentată de MongoDB.

Serverul de backend

Serverul de backend prezintă un REST API care oferă suport pentru majoritatea nevoilor pe care clienții le au, prin intermediul următoarelor rute:

- `/api/auth/register`
- `/api/auth/login`
- `/api/auth/me`
- `/api/users`
- `/api/users/:userId`
- `/api/events`
- `/api/events/:eventId`
- `/api/events/:eventId/participants`
- `/api/events/:eventId/posts`
- `/api/events/:eventId/posts/:postId`

Astfel, serverul le permite clienților să se înregistreze și să se autentifice, ceea ce le oferă acestora mai departe permisiunea de a executa alte acțiuni. Cererile de înregistrare și autentificare primesc ca și răspuns un token care identifică sesiunea utilizatorului timp de 24h. Acest token este ulterior folosit de către clienți, prin introducerea acestuia în header-ul mesajelor. Astfel, clienții vor avea permisiuni adiționale. De exemplu, doar utilizatorii de tip ONG au abilitatea de a crea evenimente prin trimiterea de cereri de tip POST către adresa `/api/events`. Un alt exemplu este adăugarea și ștergerea de postări la un eveniment, lucru permis doar prin cererile cu token ale ONG-ului care a creat evenimentul.

Serverul a scris în NodeJS și ExpressJS, ceea ce a facilitat crearea REST API-ului, iar baza de date a fost creată folosind MongoDB.

Serverul de notificări

Serverul de notificări are rolul de a notifica clienții în legătură cu apariția de schimbări relevante acestor. Astfel, ONG-urile care au evenimente create vor fi notificate când voluntarii își anunță dorința de participare la evenimentele proprii, în timp ce voluntarii interesați de un eveniment vor fi notificați când ONG-ul care a creat evenimentul postează un anunț în acesta.

Pentru a îndeplini această funcționalitate, serverul păstrează o conexiune bidirecțională cu clienții și o conexiune unidirecțională cu serverul de backend. O sesiune tipică s-ar desfășura astfel:

1. clienții de ambele tipuri se vor conecta la serverul de notificări și vor trimite imediat după un mesaj acestuia, specificând care sunt id-urile proprii de utilizatori ale clienților;

2. serverul de notificări va recepționa și memora aceste id-uri într-o listă locală (asociată socket-ului clientului emițător), pentru folosirea ulterioară;
3. serverul de backend va trimite un mesaj serverului de notificări în momentul în care o schimbare relevantă a avut loc (postare nouă sau participant nou), incluzând id-urile utilizatorilor afectați de acestea;
4. serverul de notificări primește acest mesaj și transmite clienților conectați în momentul respectiv un mesaj relevant schimbării.

Acest server a fost scris în NodeJS și folosește Socket.io (protocolul WebSockets) pentru comunicare.

Aplicația mobilă pentru voluntari

Aplicația mobilă destinată voluntarilor este o aplicație de Android. Aceasta oferă utilizatorilor funcționalitățile de bază așteptate, precum înregistrare și autentificare, dar și funcționalități specifice sistemului în cauză. Astfel, utilizatorii au acces la o activitate de vizualizare de evenimente, unde își pot exprima dorința de participare la un eveniment prin apăsarea unui buton, dar și retragerea acesteia.

Aplicația se folosește de REST API-ul oferit de serverul de backend pentru realizarea majorității funcțiilor de comunicare, trimițând cereri de tip HTTP acestuia. Serverul menține de asemenea o conexiune constantă cu serverul de notificări, de unde primește înștiințarea de noi postări în evenimentele de care voluntarul este interesat.

Aplicația web pentru ONG-uri

Aplicația web pentru ONG-uri le oferă acestora un portal pentru gestionarea de evenimente. Aplicația le prezintă utilizatorilor formulare de înregistrare și autentificare, precum și formulare destinate creării, vizualizării și ștergerii de evenimente. ONG-urile dispun de asemenea de formulare pentru postarea (și ștergerea) de anunțuri în evenimentele proprii.

La fel ca și aplicația mobilă, comunicarea se realizează în principal prin REST API-ul serverului de backend, cu excepția notificărilor, care provin de la serverul de notificări. Aplicația a fost scrisă în Angular.

Alte tehnologii

Canal clienți - serverul de backend

- Diferența dintre REST (stil arhitectural) și SOAP (protocol) este poate una de prioritate și preferințe. Cu toate acestea, utilizarea SOAP ar putea fi caracteristică proiectelor de lungă durată, cu accentul pus pe securitate și fiabilitate, iar REST celor care pun pe primul plan simplitatea și performanța.
- În cazul gestionării sistemului de evenimente pentru voluntari, plusul pe care l-ar putea aduce SOAP este la nivelul securizării (ca și REST, suportă SSL - identitate point to point, dar și WS-Security - prin intermediari) și al extensibilității în anumite situații (WS-Addressing, WS-Coordination ș.a.). Comparativ, simplitatea, performanța (multe cereri) și, în cazul aplicației mobile, avantajele folosirii combinației REST-JSON sunt motivele alegerii celui de-al doilea în cadrul aplicației noastre.

Canal clienți - serverul de notificări

- În cadrul sistemului, am folosit tehnologia de WebSocket pentru a realiza o conexiune TCP între clienți și serverul de notificări. Aceasta a fost aleasă deoarece oferă comunicare în timp real cu clientul, odată ce conexiunea a fost stabilită, fără o cerere explicită de la unul dintre participanți. WebSocket oferă o conexiune de tip full-duplex între server și client, ceea ce permite comunicarea bidirecțională simultană, însă în cadrul sistemului nostru, doar o conexiune half-duplex este folosită în majoritatea timpului, clientul trimițând doar la conectare un mesaj către server.
- O alternativă pentru WebSockets reprezintă Server-Sent Events (SSE), o tehnologie care se folosește de HTTP pentru transmisia de date. Trimiterea de notificări reprezintă una din utilizările ideale ale acestei tehnologii, deoarece fluxul de date este unidirecțional de la server la client. De asemenea, are suport pentru reconectare automată și pentru trimiterea unui ID unic din partea clientului la realizarea conexiunii. Principalul dezavantaj al SSE este limita de conexiuni care pot fi deschise într-un moment, ceea ce ar împiedica ajungerea notificărilor la destinatari în anumite cazuri.

Canal server de backend - server de notificări

- La fel ca și pentru canalul clienți - server de notificări, și acest canal folosește WebSockets pentru comunicare. Astfel, se aplică aceleași avantaje și dezavantaje ca și pentru acel caz, iar o alternativă viabilă este SSE.
- O altă alternativă o reprezintă Polling. Serverul de notificări ar putea trimite cereri în mod regulat către serverul de backend, pentru a verifica dacă există date noi pentru el. Deși este o tehnică ușor de utilizat, această variantă este mai puțin eficientă decât WebSocket, ceea ce ar cauza o degradare de performanță pe partea serverului.