

Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Domeniul Calculatoare și Tehnologia Informației
Specializarea Tehnologia Informației

**Aplicație bazată pe microservicii pentru
regăsirea în scopuri educaționale a unor
persoane cu interese comune**
Referat 1

Student:
Cornea Radu-Valentin, 1411A

Coordonator științific:
S.I. dr. inf. Dumitriu Tiberius

2022-2023

Introducere

Proiectul constă într-o aplicație pentru interconectarea oamenilor, în funcție de preferințele lor alese din cadrul aplicației, dar și în funcție de alte filtre precum distanța geografică. Scopul proiectului este de a uni oamenii mai ușor, folosindu-se algoritmi potriviți în acest sens.

Utilizatorul poate alege activități pe care vrea să le facă în acel moment, de exemplu, vizualizarea anime-urilor sau jocul de Monopoly cu alte persoane, activități care vor fi listate pe aplicație și celorlalți oameni. După acestea, utilizatorul apare disponibil cu acea activitate și celorlalți din proximitate. Iar ceilalți oameni care vor să facă aceeași activitate ca primul utilizator pot apăsa butonul "Connect" și să înceapă o conversație, pe baza căreia să discute următoarele detalii (cum vor realiza activitatea sau dacă ar prefera să facă altceva în schimb).

Aplicația se va folosi de modelul *client-server*, iar arhitectura va fi bazată pe servicii/microservicii. Un exemplu de serviciu util aplicației este cel pentru *identificarea locației*, altul este pentru *suport-ul utilizatorului* (în cazul în care acesta are nevoie de ajutor, să poată depune cereri). Pentru o bună funcționalitate a aplicației, părți funcționale de cod vor fi despărțite pe module, iar unele vor avea propriul lor proiect și server, tocmai pentru a mări decuplarea aplicației.

Tehnologiile planificate a fi utilizate pentru rezolvarea problemei sunt următoarele:

- Backend: Spring, REST, Kotlin, MariaDB și Spring Security;
- Frontend: React/Bootstrap, JS/TS/Python, CSS, HTML.

1. Fundamente teoretice

Kotlin este un limbaj de programare static, dezvoltat de JetBrains. Proiectul Kotlin a început în anul 2010 și, oficial, a fost lansat în 2016 [1]. În prezent, poate folosit nativ, dar și ținând JVM, Android și JavaScript.

Kotlin permite dezvoltatorilor să fie mai productivi, prin scrierea de cod mai puțin decât în limbaje precum Java, și de a obține același rezultat ca acesta [2]. Documentația pusă la dispoziție este foarte ușor de urmărit. Ce este iarăși util în contextul problemei este că limbajul Kotlin poate fi folosit și pentru aplicații Web, fiind o alternativă foarte bună pentru dezvoltarea serviciilor REST/RESTful.

Serviciile REST reprezintă un stil architectural, ce lucrează cu resurse și se folosește de verbe HTTP precum GET, PUT, POST, DELETE în scopul accesării acestora [3, 5]. Avantajele utilizării serviciilor REST sunt acelea că ușurează implementarea dezvoltatorilor, sunt și mai ușor

de înțeles și urmărit. Dacă sunt implementate corect, pot fi foarte utile și în a scala aplicația pe orizontală, cu ajutorul load balancer-elor. Altă funcție pe care o au aceste servicii sunt flexibilitatea, putându-se realiza servicii RESTful în funcție și de cum doresc dezvoltatorii.

Acestea pot fi dezvoltate cu ajutorul lui Spring Boot, un framework pentru realizarea aplicațiilor web, pentru backend. Este folosit și pentru a facilita conexiuni cu bazele de date, în cazul de față, MariaDB. Spring Boot pune la dispoziție și suport pentru securitate, cu ajutorul lui Spring Security. Prin intermediul lui Spring Security se vor folosi funcții de criptare și hashing, în scopul comparării algoritmilor oferiți de framework cu alte implementări, dar și de a securiza aplicația.

Pentru frontend, se va folosi React, împreună cu limbajele de programare și de scripting special concepute în acest sens. Framework-ul React nu doar că este cel mai des folosit pe piață, dar oferă foarte multe facilități precum flexibilitate, performanță, viteză și stabilitate.

Acestea sunt componentele prin care se va crea aplicația propusă. Scopul principal este acela de a evidenția performanțele diversilor algoritmi, de a-i compara și vedea cum funcționează, dar și acela de a introduce într-o aplicație de socializare inteligența artificială.

Proiecte similare

În urma unui studiu de piață, s-a constatat că, în ciuda faptului că există aplicații care încearcă să recomande oameni după anumite criterii, acestea nu realizează în totalitate obiectivul dorit. Multe dintre aplicațiile găsite pe Play Store includ opțiunea de alegere de preferințe (de exemplu, mâncare, hobby-uri, muzică), însă filtrele de utilizatori sunt inexistente.

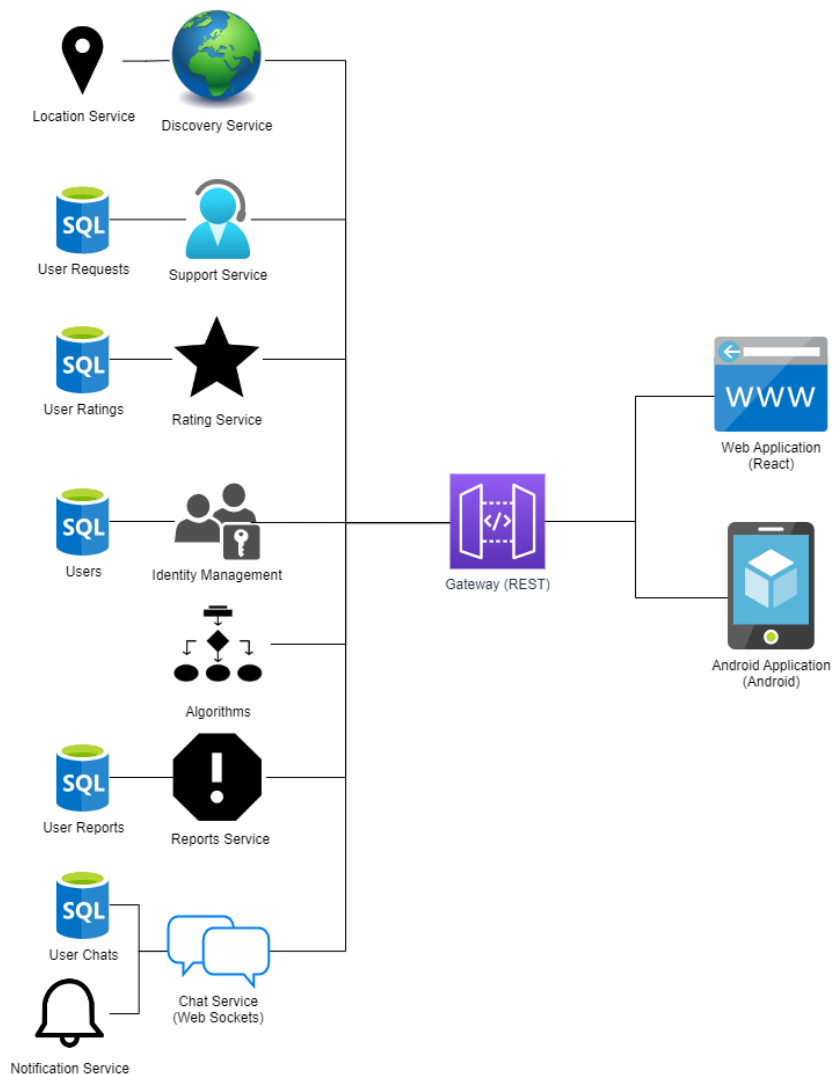
Singura aplicație care s-a constatat că ar face o parte dintre aceste funcționalități de recomandare este Meetup, dar acolo au loc recomandări de evenimente, nu de persoane. Mai de mult ar fi fost și Panion, dar acum, din păcate, nu mai funcționează publicului larg.

2. Proiectarea aplicației

Arhitectura aplicației este una bazată pe servicii. În cadrul proiectului, se vor folosi servicii precum: localizarea geografică, recomandarea utilizatorilor, sistemul de review, căutarea utilizatorilor în funcții de anumite criterii, suportul cu utilizatorii, raportarea lor și mesageria în timp real a acestora. Un alt serviciu, care este mai mult un utilitar în scopul demonstrării științifice, este serviciul pentru algoritmi, unde se vor măsura și compara cu ajutorul lui diverși algoritmi. S-a ales arhitectura de tip servicii tocmai pentru decuplarea acestora, dar și posibilitatea scalării lor în viitor.

Frontend-ul aplicației, creat cu ajutorul React, va comunica cu backend-ul prin serviciul Gateway, folosit pentru a redirecționa rutele [4]. Motivația din spate este ușurința de a securiza aplicația la nivelul Gateway-ului. Acolo se vor realiza request-uri HTTP către resursele dorite. Astfel, se pot adăuga mecanisme de securizare a aplicației precum JWT-urile, și poate chiar și din Spring Security. La nivelul backend se vor realiza implementările serviciilor menționate.

Funcționalitățile mai relevante pentru proiect sunt serviciul de recomandare și de algoritmi. Dar pentru o experiență mai bună cu utilizatorul ar fi de preferat să existe și celelalte servicii de mesagerie și de recomandări.



Concluzie

Există un serviciu pentru utilizator ce urmează standardul REST, precum și un serviciu pentru algoritmi la care s-a adăugat Spring Security, dar care încă a rămas la stadiul de dezvoltare.

Pe viitor, este de așteptat să finalizez serviciul pentru algoritmi, încât să nu mai existe probleme cu Spring Security, astfel încât să poată fi accesat de Gateway mai ușor. Apoi ar urma dezvoltarea serviciului de recomandări pentru utilizatori, în paralel cu dezvoltarea Frontend-ului aplicației Web.

Bibliografie

[1] Kotlinlang FAQ: <https://kotlinlang.org/docs/faq.html>, ultima accesare: decembrie 2022

[2] Nicolae Sfetcu, Limbajul de programare Kotlin pentru Android: <https://www.telework.ro/ro/limbajul-de-programare-kotlin-pentru-android>, ultima accesare: decembrie 2022

[3] Yiğit Kemal Erinc, The Benefits of Going RESTful – What is REST and Why You Should Learn About It: <https://www.freecodecamp.org/news/benefits-of-rest/>, ultima accesare: ianuarie 2023

[4] Ubaid Pisuwala, The benefits of ReactJS and reasons to choose it for your project: <https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html>, ultima accesare: ianuarie 2023

[5] MDN contributors, HTTP: <https://developer.mozilla.org/en-US/docs/Web/HTTP>, ultima accesare: ianuarie 2023