



GRÁFADATBÁZIS dokumentáció

Haladó adatbáziskezelő-rendszerek (VEMIVIM246)

Készítette:

Pócz-Nagy Dániel (I04AYS)

Vadász Csaba (D18ZGX)

**Programtervező informatikus
és Adattudomány mesterképzés**



ChillMate App



Tartalomjegyzék

1. Projekt rövid ismertetés..... 4

2. Csapat adatai 4

3. Használt technológiák 4

4. Architektúra..... 5

5. Funkcionális specifikáció..... 6

6. Logikai adatbázis modell (Neo4j séma)..... 6

7. Felhasználói felület (GUI) áttekintés 9

8. Működés bemutatása 11

9. Fejlesztési tapasztalatok 11

10. Konklúzió és továbbfejlesztési lehetőségek 12

Hivatkozások 13

1. Projekt rövid ismertetés

Projekt neve: *ChillMate (chill-mate) App*

A projekt célja, hogy emberek érdeklődési körök alapján találjanak egymásra. A rendszer gráfos adatbázison alapul, ahol a felhasználók és az érdeklődési területeik csomópontként jelennek meg, a köztük lévő kapcsolatok pedig élek formájában. Ez lehetővé teszi összetett kapcsolati hálókat hatékony megjelenítéssel és elemzéssel. Lehetőség van a kapcsolatok súlyozására is.

A fejlesztés során minden tervezési mintát, tiszta kód alapelveket, objektum-orientált programozási paradigmát és a SOLID elveket követve valósítottuk meg a rendszert, figyelembe véve a gráfadatbázis igényeit és az adatok hatékony kezelését.

2. Csapat adatai

Tagok:

- **Pócz-Nagy Dániel** (Neptun: *I04AYS*)
- **Vadász Csaba** (Neptun: *D18ZGX*)

3. Használt technológiák

Backend

- Java 21
- Spring Boot 3.4.3
- Spring Web
- Spring Data Neo4j
- Lombok

- ModelMapper
- Maven

Frontend

- React 19
- React Router DOM
- React Flow
- Axios
- React Modal
- Jest + React Testing Library

Adatbázis

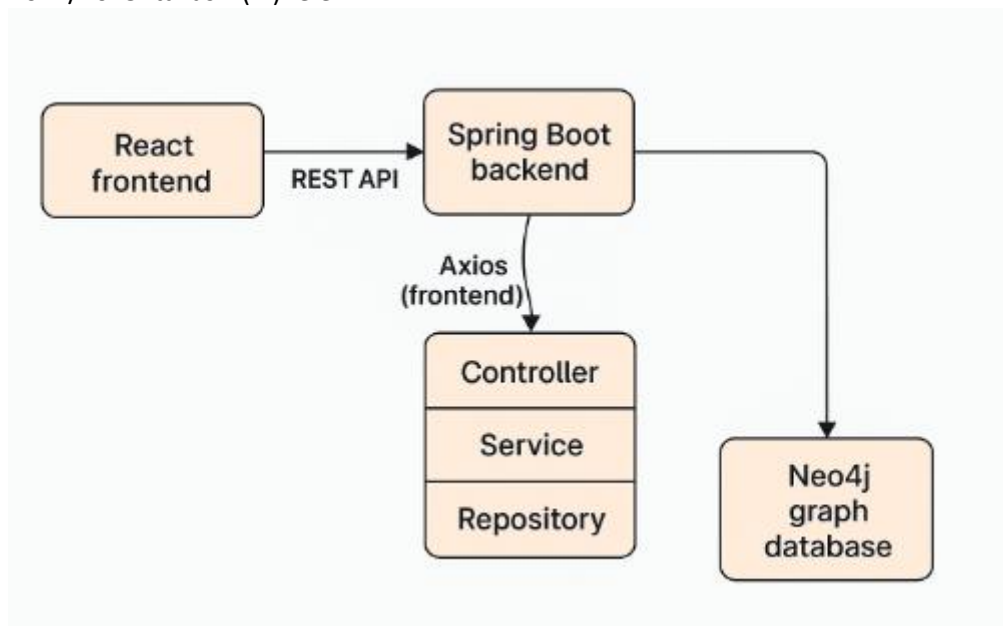
- Neo4j 5.x (Cypher + Spring Data Neo4j integrációval)

4. Architektúra

- **React frontend:** kapcsolat a backend REST API-n keresztül
- **Spring Boot backend:** üzleti logika, kapcsolatok kezelése
- **Neo4j gráfadatbázis:** felhasználók, érdeklődési körök, kapcsolatok tárolása

Kapcsolati útvonal:

Axios (frontend) → REST API → Controller → Service → Repository → Neo4j



5. Funkcionális specifikáció

- Felhasználó létrehozása, szerkesztése, törlése
- Érdeklődési körök hozzáadása a felhasználóhoz (zene, film, sport, hobbi, esemény, lakhely)
- Baráti kapcsolatok létrehozása és megjelenítése
- Kapcsolati gráf megjelenítése interaktív nézetrel (React Flow segítségével)
- Vizualizáció frissítése kapcsolat módosításakor
- Demó adatbázis inicializálás fejlesztéshez és teszteléshez

6. Logikai adatbázis modell (Neo4j séma)

Csomópont típusok (Nodes):

- User (pl. név, e-mail)
- City
- Hobby

- Sport
- MusicGenre
- Movie
- Activity (azaz, Event)



Kapcsolatok (Relationships):

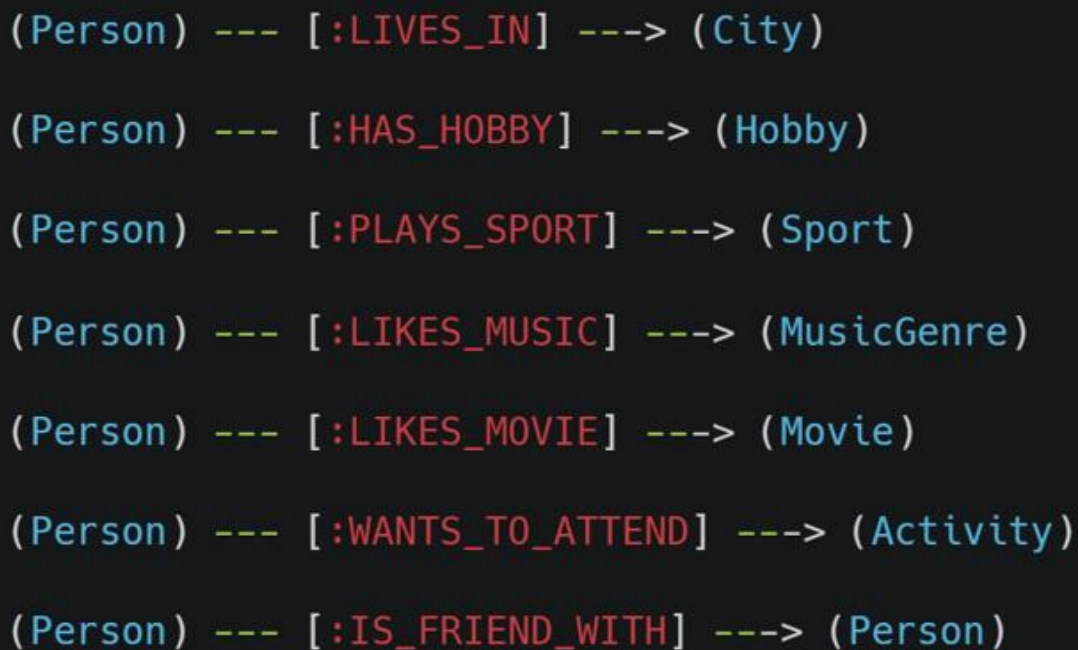
Egyirányú:

- LIVES_IN
- HAS_HOBBY
- PLAYS_SPORT
- LIKES_MUSIC
- LIKES_MOVIE
- WANTS_TO_ATTEND

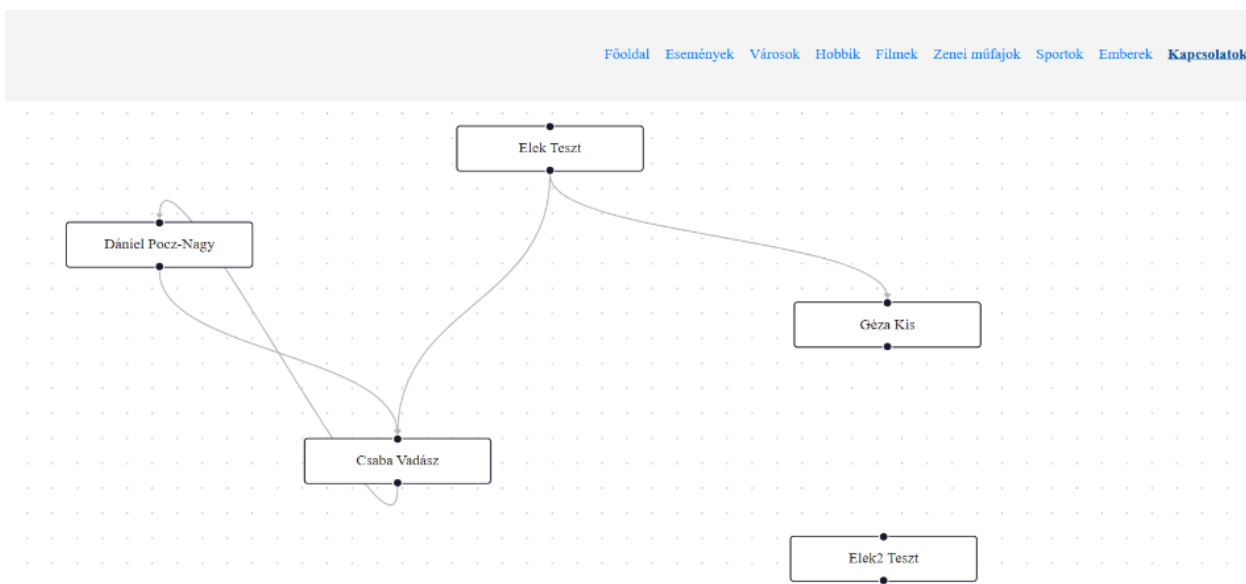
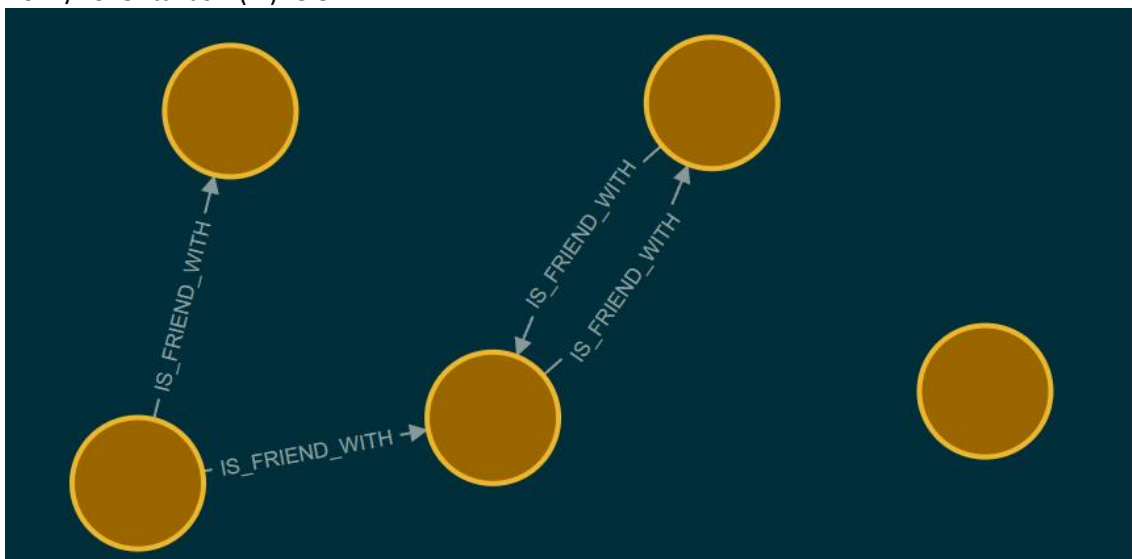
Kétirányú (szimmetrikus):

- IS_FRIEND_WITH (ahol egy user egy másik user felé tud kapcsolatot kialakítani)

A kapcsolat egy felhasználó (A) és egy másik felhasználó (B) között jön létre, ha A barátnak jelöli B-t. A rendszerünkben technikailag minden barátság kapcsolat egyirányúként kerül tárolásra. A valódi szimmetrikus kapcsolat (kölsönös barátság) csak akkor jön létre, ha mindkét fél (A és B is) kapcsolatot hoz létre egymás felé, tehát a gráfban két egyirányú él jelenik meg: $A \rightarrow B$ és $B \rightarrow A$. Ez a megközelítés lehetőséget ad például a „barátkérés függőben” állapot nyomon követésére is. A szimmetria tehát nem automatikus, és nem is követelmény — kizárólag akkor jön létre, ha mindkét fél kapcsolatot kezdeményez a másik irányába.



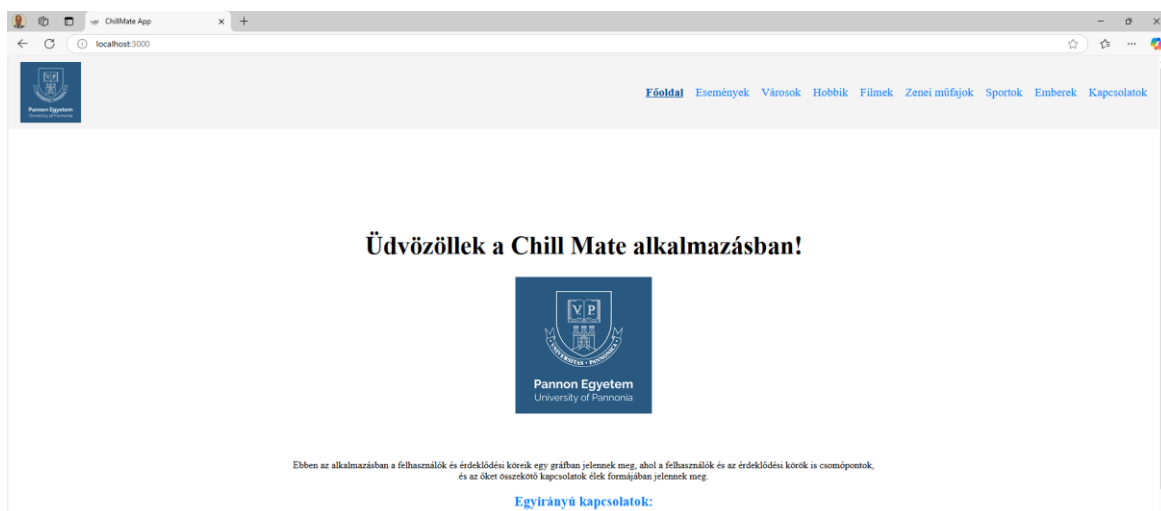
```
(Person) --- [:LIVES_IN] ---> (City)
(Person) --- [:HAS_HOBBY] ---> (Hobby)
(Person) --- [:PLAYS_SPORT] ---> (Sport)
(Person) --- [:LIKES_MUSIC] ---> (MusicGenre)
(Person) --- [:LIKES_MOVIE] ---> (Movie)
(Person) --- [:WANTS_TO_ATTEND] ---> (Activity)
(Person) --- [:IS_FRIEND_WITH] ---> (Person)
```

7. Felhasználói felület (GUI) áttekintés

- **Főoldal:** rövid projekt bemutatót tartalmazza
- **Események / Városok / Hobbik / Filmek / Zenei műfajok / Sportok:**
 - Minden szekcióban a keresési lehetőség mellett lehetőség van új elem hozzáadására vagy meglévő elem törlésére.
 - A felhasználói interakciók intuitívak, a felület egyszerű és gyors hozzáférést biztosít a keresési funkciókhoz.

- **Emberek**
 - Az embereknel az alap funkciók mellett egy modal ablak jelenik meg, amely lehetővé teszi a felhasználók és a többi csomópont közötti kapcsolat kezelést.
 - Egy felugró táblázatban jeleníthetők meg az egy felhasználóhoz tartozó kapcsolatokat.
- **Kapcsolatok**
 - Az emberek közötti kapcsolatokat gráf (React Flow) formájában jeleníti meg, így könnyen áttekinthetőek a különböző felhasználói kapcsolatok.
- **Dark/light mód támogatása**
- **Modal ablakok:** felugró űrlapok az interakciókhoz (pl. kapcsolat létrehozása)



[Főoldal](#) [Események](#) [Városok](#) [Hobbik](#) [Filme](#) [Zenei műfajok](#) [Sportok](#) [Emberek](#) [Kapcsolatok](#)

8. Működés bemutatása

Az alábbi funkciók kerülnek bemutatásra képernyővideó vagy élő demó formájában:

- Új felhasználó hozzáadása és érdeklődési körök megadása
- Kapcsolatok létrehozása más felhasználókkal
- Teljes gráf vizualizáció frissítése
- Két felhasználó közti barátság létrehozása és azonnali tükrözése a grafikonon

9. Fejlesztési tapasztalatok

- A gráfmodell tervezése eltért a hagyományos relációs gondolkodástól, új szemléletet igényelt
- A React Flow használata kezdetben kihívás volt, de vizuálisan nagyon erőteljes eszköz
- A Spring Data Neo4j és Cypher lekérdezések jól kombinálhatók, de figyelni kell a kapcsolatok irányára
- A vizuális és adatkezelő réteg szinkronizálása volt az egyik legösszetettebb kihívás

10. Konklúzió és továbbfejlesztési lehetőségek

A rendszer alkalmas érdeklődési körök alapján történő kapcsolatépítésre.

Továbblépési lehetőségek:

- Kapcsolatok súlyozása – például egy adott érdeklődési kör (pl. film) iránti kedveltség mértékének figyelembevételével
- Összetettebb keresések (pl. barátok barátai alapján)
- Jogosultságkezelés és bejelentkezés (auth modul)
- Mobil nézet / mobil applikáció támogatása
- Felhőalapú Neo4j Aura integráció
- Ütemezett mentések, naplózás a backend oldalon

Hivatkozások

1. számú melléklet: <https://github.com/Csaba79-coder/chill-mate-backend>
2. számú melléklet: <https://github.com/Csaba79-coder/chill-mate-frontend>